Solid Topology in Multiresolution

Esdras Soares de Medeiros Filho

October 17, 2011

Abstract: In this thesis we introduced a filtering method, by means of stellar and handle operators, for multiresolution representation of solid regions. In each level of resolution, we have a sampling by Poisson disks with particular characteristics. With this filtering, it is possible to generate adaptive meshes so that the resolution and topology are controlled in unified way.

Resumo: Nessa tese introduzimos um método de filtragem utilizando operadores *stelares* e de *handle* para representar sólidos em multiresolução. Em cada nível de resolução temos um amostragem por discos de Poisson com características particulares. Com esta filtragem é possível gerar malhas adaptativas de tal forma que a resolução e a topologia sejam controladas de forma unificada.

Agradecimentos

Agradeço primeiramente a Deus por ter me criado e por toda força necessária para essa desafiante jornada.

Aos meus queridos pais, Esdras e Rita, por todo carinho, atencão, educação e apoio.

Aos meus irmãos (Edgard, Alynne e Ylanne), padrinhos (Ênis e Cláudia) e familiares pelo incentivo constante.

À Comunidade Católica Shalom e seus membros por serem instrumentos de Deus que levam a um caminho seguro na busca da indentidade e da verdade.

Ao meus orientadores Prof. Luiz e Prof. Hélio pela responsabilidade e atencção em toda a trajetória deste trabalho iniciada desde o mestrado e ao Prof. Thomas pelas dicas elucidativas.

Aos colegas do Visgraf, Adelailson, Vinícius, Asla, Anderson, Ives, Emílio e tantos outros que passaram por este lar, agradeço pela amizade e trocas de idéias.

Por fim, agradeço ao IMPA e ao CNPq por toda a infra estrutura e pelo apoio financeiro.

Dedico esta tese especialmente ao
meu pai e amigo Esdras.

CONTENTS

1. INTRODUCTION

1.1 Motivation

In Computer Graphics most of the graphical objects [18] are representations of things from the real world such as humans, machine engines, water, buildings and cars. It is well known that many applications on visualization represent and render such objects by the external surface. Indeed, as nobody can see inside opaque objects, the interior part does not matter. In 3D Scanning, for example, the captured objects are represented by meshes with some attributes such as texture and reflectance.

In other applications the objects are carried with details inside that affect directly the results. These objects may contain some type of material with several attributes. We call them *Solid Objects*. In fluid simulation, for example, the fluid has attributes such as viscosity and gravity, and, in geological modelling, the rocks have porosity and permeability. Both the fluid and the rock are objects limited by a surface and may contain holes inside.

Mathematically a solid object is a compact manifold with the same dimension of the surrounding space. Its boundary is the most significant part because it determines the geometric shape and topology. The topology of a region in the plane can be roughly described as the number of holes and connected components. For solids in the space the topology is the number

of cavities and connected components. The main property of the topology is the invariance by homeomorphisms.

Most applications using solid objects are concentrated on $\mathbb{R}^2$ and $\mathbb{R}^3$ spaces. Here, we give examples in some active research areas.

Modelling: The Constructive Solid Geometry (CSG) is a well known technique. CSG allows designers create a complex surface by using boolean operators to combine primitive objects like spheres, cylinder, cones and tori.

Simulation: The finite element analysis (FEM) is a powerful tool for simulation of natural phenomena. In engineering, for example, the geomechanics simulates the behaviour of soil and rock. Actually, it is very helpful for oil exploitation. Fluid simulation and deformation are popular tools in computer graphics for generating realistic animations like water, smoke, explosions, and related phenomena. In computer graphics, such simulations range in complexity from extremely time-consuming high quality animations for film and visual effects, to simple real-time particle systems used in modern games.

Scientific Visualization : Scientific visualization focuses on the use of computer graphics to create visual images which aid in understanding of massive numerical representation of scientific concepts or results [27]. It is still a big challenge the visualization of complex solid objects.

Image Analysis: Segmentation refers to the process of partitioning a digital image into multiple regions (sets of pixels). The goal of segmentation is to simplify and/or change the representation of images into something that

is more meaningful and easier to analyse. Image segmentation is typically used to locate objects and boundaries (regions and lines) in images.

Image Compression: Image compression is the application of data compression on digital images. Indeed, the goal is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Some methods such as Demaret et al. [12] approximates the image by an adaptive triangulation for error minimization. Lewiner et al. [24] (GenCode) is a method that compress 2D and 3D solid objects in a very efficient way.

1.2 Related Subjects

This thesis on solid objects relates many subjects that are well known to the scientific community in Mathematics and Computer Graphics. In this section, we enumerate and summarize several keywords that we provide contributions. This document does not follow necessarily this sequence.

Sampling: Solid objects have an important advantage compared with general surfaces. Its maximal dimensionality allows the set of the interior points have a "natural metric", i.e., the same metric of the space where it is embedded. Therefore, samplings of solid objects can be extracted through Poisson distributions with a constant bound distance between sample points (Poisson disk sampling). This minimal distance constraint give us, in a natural way, a scaled representation of solid models.

(a) (b)

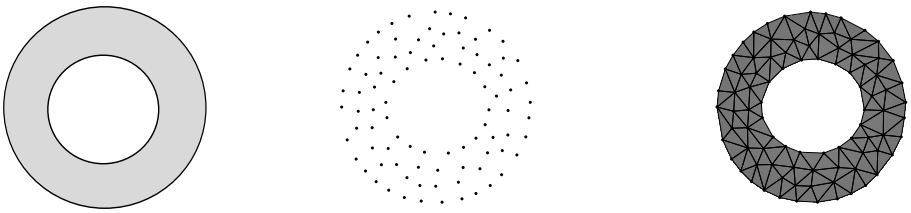
Fig. 1.1: (a) Original Solid Object. (b) Poisson disk sampling.

Interpolation: Sampling and reconstruction of Graphical Objects [18] are maxims in computer graphics. Thanks to the Shannon sampling theorem [39], signals may be fully represented by points uniformly distributed with a sampling rate greater than twice the bandwidth. These samples can be interpolated via *sinc* function to reconstruct exactly the original signal.

In the context of reconstruction of "surface signals" we highlight Amenta et al. [2] and Bernardini and Bajaj [5]. These papers determine sampling conditions that assure the existence of a simplicial reconstruction topologically equivalent to the original surface. Differently from the Shannon theorem related to general signals where the reconstruction is a exact one by means of interpolating *sinc* functions, the reconstruction of surface signals is just a linear approximation. In Bernardini and Bajaj [5], for example, the reconstruction results an *alpha solid*. In this case, it is possible to recover exactly the topology of the original surface signal using a predefined sampling condition.

In this thesis, we extend the alpha solid notion to *Solid Alpha Complexes* [28] for reconstruction of solid object instead of surfaces (see figure 1.2).

Multiresolution: The scale space theory models a signal $f : \mathbb{R}^d \to \mathbb{R}$ in a family of different scales as $L : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$, where L is a transformation



(a) (b) (c)

Fig. 1.2: (a) A ring image. (b) Poisson disk sampling. (c) The solid alpha complex reconstruction. The ring topology is recovered.

defined as the convolution of f with the Gaussian kernel. This transformation works as a low-pass filter. As we increase the variance of the Gaussian kernel we also reduce the bandwidth of the transformed signal, allowing the reduction of the sampling rate required to its reconstruction. Therefore, we may represent the scale space of f as a family of samplings of impulse functions with decreasing rates. The reconstruction of the family elements generates a sequence of infinite signals that, in the limit, converges to the original signal.

In *Multiresolution of geometric models* we have representations of geometric entities at different resolutions. Such models are useful to handle geometric data efficiently, depending on applications needs. In this thesis we provide a representation of solid objects "signals" in multiresolution. The resolution levels are represented by solid alpha complexes with nested Poisson disk samplings.

Topology: Topological questions arise frequently in many areas of computation. Recently several tools have been developed in topology to address these problems Edelsbrunner et al. [15], Sharf et al. [40].

We pose a question: The holes have size? The topology and geometry of

a space are intrinsically related, since both represents properties of the same space. Therefore, geometric modifications, such as decimation of the ring in figure 1.3, could modify the topology. Generally as we increase/decrease the sampling rate, two types of details are emphasized: the geometrical characterized by the shape of the holes and the topological characterized by the number of holes. Attempting to answer the question above this thesis addresses topology changes at different resolution representations. In a nutshell, the geometry and topology are correctly assigned to the model at a given scale.

(a) (b)

Fig. 1.3: Topology change in the decimation of the ring example. (a) Coarser ring. (b) Finer ring.

Adaptation: Level of Detail (LOD) operations extract from multiresolution models triangulations with localized details. This can be achieved by *Variable Resolutions Structures [37] (or Multitriangulation)*. The idea behind this structure is a partial order set represented in the form of a *Direct Acyclic Graph (DAG)*. The partial order is created between *local modifications* at different successive layers of the multiresolution model. Since all triangulations decompose the same domain, each of them can be obtained from the one preceding it (or following it) through refinement (respectively, through simplification) by a sequence of local modifications. In the context

of solid alpha complexes they are obtained through a *filtering by topological operators* which is based on the Delaunay point insertion/removal algorithm. The operators can be of two types: *combinatorial* modifications which change the structure of the triangulation internally and the *constructive* modifications which add or remove pieces of triangulations.

We abstract adaptations as *Geometrical and Topological queries operations (G/T-queries)*. An example of query operation is an adapted mesh extraction parametrized by an *adaptation function* and a *focus set* (figure 1.4). Queries are particularly important, because it appears in many applications, such as, real time visualization and interactive modelling.

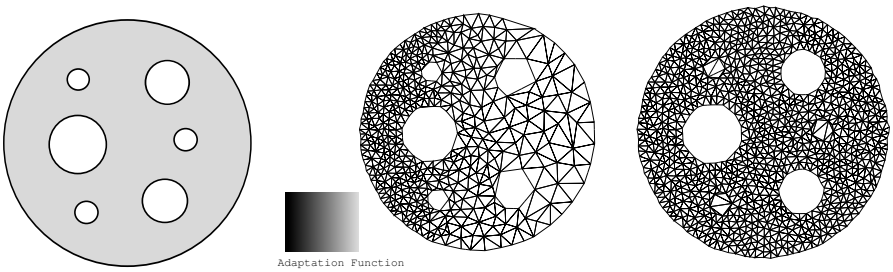


Fig. 1.4: Examples of query operations. (a) Original Solid. (b) Using an adaptation function. (c) Simplifying small holes.

1.3 Weak Points in Related Works

Topology DOES NOT Change in Multiresolution

There are many works related to representations of simplicial meshes in different scales. In Kobbelt [22], Velho and Zorin [42], the several resolution levels are generated by means of subdivision rules. In Puppo [37] a study is performed on simplicial multiresolution of planar domains. In Hoppe [21]

we find representations of meshes that progressively change resolution using *edge collapse* operations. The limitations of these mentioned works, besides of other similar ones, is that the topology is already previously determined by the represented object (or by the base mesh) in all levels of resolution. In short, there is no change of topology.

Resolution DOES NOT Change in Filtration

The topology treatment in a family of simplicial complexes in Edelsbrunner et al. [15] is very similar to ours. They define *topological persistence* by means of a *Filtration*. Their results have several applications. Among them, we point out the analysis of topological noise in protein samplings and the simplification of morse-smale complexes. One gap of this article is that there is not approach related to multiresolution.

Filling Gaps

We propose an unified geometric and topological framework for simplicial representations of bidimensional solid objects in different scale levels which allows the control of topology changes between them. In summary, we **can do** multiresolution in topological filtering and perform topological changes in multiresolution.

1.4 Organization

The thesis is organized as follows:

Chapter 2: We introduce and revisit concepts that will be used along the text. Among them, we point out the simplicial concepts, monotone family of

points, Delaunay triangulation, Solid Alpha Complex and Voronoi diagram. We also establish some notations.

Chapter 3: We present the mathematical framework and data structure adopted to represent the triangular meshes. We survey the topological operators that will be used to insert and to remove points into a mesh. These operators will provide the interactions between the resolution levels.

Chapter 4: We introduce the concepts Poisson disk samplings (PDS), we characterize the dual complex and exhibit conditions to obtain a topologically equivalent reconstruction of the original solid object.

Chapter 5: We will introduce a family of nested PDS's.

Chapter 6: This chapter describes how to use the topological operators in monotone family of meshes. Firstly, we analyse the intrinsic relationship between refinement (simplification) and topological junctions (disjunctions). Then we define *Filtering* as a sequence of topological junctions (disjunctions) and give algorithms to build them. The instances of the algorithms are the (α, β)-family of meshes studied in chapter 5.

Chapter 7: In this chapter we propose a multitriangulation data structure with topology changes. Our framework is obtained by abstracting on specific construction rules (local modifications), while considering only the intrinsic relationships among basic elements.

Chapter 8: In this chapter we consider the applications of (α, β)-Filtered

TMTs for managing level of detail (LOD) of solids. We define relevant operations and give some examples.

Chapter 9: We present a conclusion of the thesis. We explore the generalizations and limitations. Future works are proposed as well.

2. PRELIMINARY CONCEPTS AND NOTATIONS

This chapter describes some basic concepts in topology and in computational geometry that will be used along this thesis.

2.1 Basic Topological Concepts

A *simplex* σ_T^k of dimension k (k-simplex, for short) is the convex hull of $k+1$ points in general position $T = \{v_0, ..., v_k\}, v_i \in \mathbb{R}^n$, i.e., $\sigma_T^k = conv(T)$. The points $v_0, ..., v_k$ are called the *vertices* of σ. A face of σ_T is the convex span of some of the vertices of σ and therefore is also a simplex. The simplices of dimensions 2 and 1 will be called, respectively, *triangles* and *edges*. If σ is a face of a simplex τ then σ is said to be incident to τ. The boundary of a p-simplex σ, denoted by $\partial\sigma$, is the collection of all proper faces. Two k-simplices σ and $\rho \in K$ are *adjacent* when $\sigma \cap \rho \neq \emptyset$, and *independent* otherwise. The *valence* or *degree* of a vertex $v \in K$ is the number of edges which have v as a vertex, and is denoted by $\deg(v)$.

We will name $|\sigma|$ as the underlying subspace of $\mathbb{R}^n$ covered by the simplex σ. A *simplicial complex* K is a finite set of simplices together with all its subsimplices such that if σ and τ belong to K, then either $|\sigma|$ and $|\tau|$ meet at a common face λ or σ and τ are independent.

We will name $|K|$ as the subspace of $\mathbb{R}^n$ covered by the complex K. We say that two simplices σ_1, σ_2 are *commom* if $|\sigma_1| = |\sigma_2|$. A *subcomplex* L of

K is a simplicial complex such that $L \subset K$. The dimension of a simplicial complex K is given by the highest dimension among all simplices of K. We will represent by K^i the subset of the simplices of K with dimension i.

Let K_m and K_n be two complexes and $L_m \subset K_m$ and $L_n \subset K_n$ their commom simplices. We say that K_m and K_n are *compatible* if $|L_n| = |L_m| = |\{\text{common simplices of } K_m \text{ and } K_n\}|$.

A complex K is *connected* if it cannot be represented as a union of two non-empty disjoint subcomplexes L and M without common simplexes. A *component* of a complex K is a connected subcomplex that it is not contained in a larger connected subcomplex of K.

The *join* $\sigma \star \tau$ of independent simplices σ and τ is the simplex whose vertices are those of σ and τ. The join of complexes K and L, written $K \star L$, is $\{\sigma \star \tau : \sigma \in K, \tau \in L\}$ if the following holds:

1. If $\sigma \in K$ and $\tau \in L$, σ and τ are independent.

2. If $\sigma_1, \sigma_2 \in K$ and $\tau_1, \tau_2 \in L$, then $\sigma_1 \star \tau_1 \cap \sigma_2 \star \tau_2$ is either empty or a face of $\sigma_1 \star \tau_1$ and $\sigma_2 \star \tau_2$.

Consider a simplicial complex K and $\sigma \in K$. The local neighborhood of σ is described by the following elements:

- The *open star* of σ is

$$star(\sigma, K) = \{\tau \in K : \sigma \text{ is a face of } \tau\}.$$

- The *star* of σ is

$$\overline{\text{star}}(\sigma, K) = \{\tau \in K : \tau \text{ is a face of an element of } star(\sigma, K)\}.$$

- The *link* of σ is

$$link(\sigma, K) = \{\tau \in K : \tau \text{ and } \sigma \text{ are independent and } \sigma \star \tau \in K\}.$$

Definition 1. *A Solid Simplicial Complex is a complex that does not have isolated simplexes, i.e., k-simplexes that are not faces of a simplex of wider dimension.*

Given a simplicial complex K, the collection $\overline{K} \subset K$ is the maximal solid simplicial complex contained in K. The most important particular case of definition above is a *Combinatorial Surface*.

Definition 2. *A Combinatorial Surface is a solid simplicial complex K, $|K| \subset \mathbb{R}^2$: Every edge in S is bounding either one or two triangles and the link of a vertex in K is homeomorphic either to an interval or to a circle.*

Sometimes we will refer to combinatorial surfaces as *triangulations*, *triangle meshes* or, simply, as *meshes*. The edges in a combinatorial surface K incident to only one face are called *boundary edges*. A vertex incident to a boundary edge is called a *boundary vertex*. The subcomplex of K of those boundary simplices forms the *boundary* of S and is denoted by ∂K. The boundary of a combinatorial surface is a collection of closed curves. The edges and vertices that are not on the boundary are called, respectively, *interior edges*[1] and *interior vertices*.

The set of faces, edges and vertices of a surface K will be denoted, respectively, by K^2, K^1, and K^0.

Let $S \subset \mathbb{R}^n$ be a finite set of points and K a solid simplicial complex. If $K^0 \subset S$, by simplicity we will write (S, K) as a notation (read "pair of

[1] Observe that the *link* of an interior edge is the pair of opposite vertices.

points S and simplicial complex K" or simply "pair").

Let $\mathcal{F} = \{S_i\}_{i\in\{1,2,...n\}}$ be a *family of samplings*. We say that is *increasing* if $S_1 \subset S_2... \subset S_n$. Analogously we say that it is *decreasing* if $S_1 \supset S_2... \supset S_n$. Each number i represents a *level*. An *Increasing (Decreasing) Family of Pairs* $\mathcal{M}(\mathcal{F}) = \{(S_i, K_i)\}_{i\in\{1,2,...n\}}$ is family such that $\mathcal{F} = \{S_i\}_{i\in\{1,2,...n\}}$ is increasing (decreasing). We will speak generically *monotone*, meaning either increasing or decreasing.

Notice that in monotone family of pairs, the meshes have common vertices, however, they can increase or reduce the number of vertices as i increases. We will see ahead that this last definition is fundamental to formalize structures in multiresolution with topology changes.

Based on the structure of simplicial complexes in next section we will define some simplicial artifacts such as Delaunay Triangulations, Alpha Complexes and Alpha Shapes.

2.2 Triangulation and Voronoi Diagram

Delaunay Triangulation of a set of points in the plane is the unique set of triangles that connect such points and that satisfy the property of the "empty circle": the circumcircle of each triangle does not include any other point. In a certain sense, it is the most natural manner to triangulate a set of points. Below we will give a general definition based in simplicial complexes.

Definition 3. *Given a set $S \subset \mathbb{R}^n$ in general position, the Delaunay triangulation of S is the simplicial complex* $\mathrm{DT}(S)$ *that comprises only*

1. *all the n-simplices, $(0 \leq k \leq n)$, with vertices in S such that the*

circumsphere (the smallest sphere, such that all points are in its boundary) does not contain other point of S, and

2. *all the k-simplices that are faces of other simplices are also in* $\mathrm{DT}(S)$.

We will define the Voronoi diagram and will establish its relation with Delaunay triangulation.

Definition 4. *Let S be a set of n points in the plane. for each $s \in S$, the Voronoi region $V(s)$ is a set of points of the plane closest to p than other points of S. The Voronoi diagram $V(S)$ is the partition of the plane generated by the regions of Voronoi of S.*

We have then, the proposition below, which is well known in the literature.

Proposition 1. *The Delaunay triangulation of S is the dual graph of the Voronoi S diagram: two points of S are linked by an edge in the triangulation of the Delaunay triangulation if and only if its regions of Voronoi are incidents in the diagram of Voronoi of S.*

Proof. See [9]. □

2.3 Alpha Complexes and Alpha Shapes

Alpha Complexes are simplicial complexes that describe levels of detail of cluster points. Through the variation of a real positive number α we obtain different shapes, from the more refined to the more coarse. The more refined is the set of points itself, achieved when $\alpha = 0$. As α increases, the shape also increases by the addition of simplices developing cavities that may gather or split. The coarser form is the Delaunay triangulation which is obtained

for sufficient large values of α. More precisely, the Alpha Complex has the following definition:

Definition 5. *Let $S \subset \mathbb{R}^n$ be a set of points in a general position. For $T \subset S$ with $|T| \leq n$, let b_T and μ_T be the smallest ball that contains points of T and its radius, respectively. Given $0 \leq \alpha \leq \infty$, the alpha complex $\mathcal{C}_\alpha(S)$ of S is the subcomplex of $\mathrm{DT}(S)$ where the simplex $\sigma_T \in \mathrm{DT}(S)$ is in $\mathcal{C}_\alpha(S)$ if:*

1. *$\mu_T < \alpha$ and $b_T \cap S = \emptyset$, or*
2. *σ_T is a face of other simplex in $\mathcal{C}_\alpha(S)$.*

Observing the definitions of the Delaunay triangulation and Alpha Complex, the following properties are immediate:

P1. If $\alpha_1 \leq \alpha_2$ then $\mathcal{C}_{\alpha_1} \subset \mathcal{C}_{\alpha_2}$,

P2. $\mathcal{C}_\alpha \subset \mathrm{DT}(S)$, $\forall \alpha > 0$ and

P3. $\mathcal{C}_\infty = \mathrm{DT}(S)$.

The alpha shapes S_α is defined as $|C_\alpha(S)|$. Thus, as in alpha complexes we obtain a Delaunay triangulation for great α parameters also in alpha shapes we achieve precisely the convex hull. In fact, an alpha shape is adequate for the generalization of the concept of convex hull, being well adopted in many applications. See for instance [16].

2.4 Alpha Solid and Solid Alpha Complex

In general, the alpha complex and the alpha shape are respectively simplicial complexes and polytopes composed by simplexes of different dimensions.

Bernardini et al.[6] defined the *solid alpha shape* (or simply alpha solid) as the *alpha shape* without isolated k-simplices. In a similar way we define the Solid Alpha Complex as the solid alpha without k-simplexes isolated. It is a type of subcomplex which is a "regularized" version of the alpha complex. As we have seen before, it is a maximal solid simplicial complex.

In figure 2.1 we show the difference between the alpha complex and the solid alpha complex in the 2D case. We will denote the solid alpha complex of a set $S \subset \mathbb{R}^n$, given $0 \leq \alpha \leq \infty$, as $\overline{\mathcal{C}_\alpha(S)}$. Notice that the properties P1, P2 e P3 are still valid for solids alpha complexes.

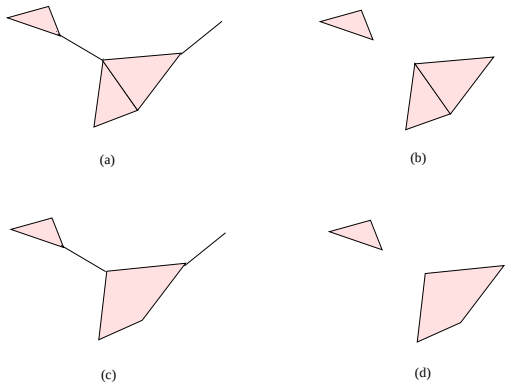


Fig. 2.1: The alfa complex (a) and its solid alpha complex (b). The *alpha shape* (c) and alpha solid (d).

3. TOPOLOGICAL OPERATORS

This chapter is based on Velho et al. [43] and introduces an unified framework for basic operations on combinatorial 2-manifolds with or without boundary. We present such operators and demonstrate that they provide a complete set of elementary operations for mesh modification. We also give a description of the algorithms and data structures for an efficient implementation of these operators. They will be strongly used in the next chapters. For instance, the operators link the different levels of a Increasing Family of Pairs (see chap. 2). More precisely, the "jump" between successive layers is a sequence of applications of them.

3.1 Mathematical Framework

We will see that basic concepts from topology is sufficient to define query operators. Operators that change the mesh topology are based on the Handlebody theory, while operators that alter the mesh structure are based on the Stellar theory.

3.1.1 Handlebody Theory

The topological setting applied to boundary representation of solids [3] has traditionally been the Euler-Poincaré theory, dated from the turn of the 19^{th} century [36].

The Handlebody theory [32] refines the Euler-Poincaré theory by bringing several new topological invariants for n-dimensional manifolds. The fundamental problem of Handlebody theory is to study the topological changes generated by handle attachments to manifolds with boundary.

In the surface case, three types of handles are to be defined and they will be distinguished by an index λ that varies from 0 to 2. Here, D^i denotes the i-dimensional disk and ∂P the boundary of a set P.

Definition 6. *A handle of index λ, denoted by H_λ, is a pair of topological spaces (A_λ, B_λ) such that $B_\lambda \subset A_\lambda$, $A_\lambda = D^\lambda \times D^{2-\lambda}$ and $B_\lambda = \partial D^\lambda \times D^{2-\lambda}$.*

According to this definition, one can observe that: 1) the set A_0 is a 2-disk and B_0 is the empty space; 2) the set A_1 is a square and B_1 is defined to be two of its opposite sides and 3) the set A_2 is a 2-disk and B_2 is its boundary (see Figure 3.1).

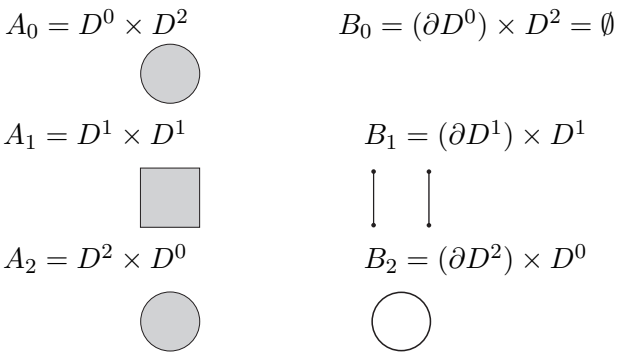


Fig. 3.1: The 2D Handles: $\mathbf{H}_0 = (A_0, B_0)$; $\mathbf{H}_1 = (A_1, B_1)$; $\mathbf{H}_2 = (A_2, B_2)$.

To attach a handle $H_\lambda = (A_\lambda, B_\lambda)$ to the boundary of a surface S means to identify by a homeomorphism the set B_λ with a subset I contained in the boundary of S that is homeomorphic to B_λ.

The next theorem is the main mathematical tool in which the Handlebody Theory is based.

$S_0 = \emptyset$

$S_1 = S_0 + \mathbf{H}_0$

$S_2 = S_1 + \mathbf{H}_1$

$S_3 = S_2 + \mathbf{H}_1$

$S_4 = S_3 + \mathbf{H}_2$

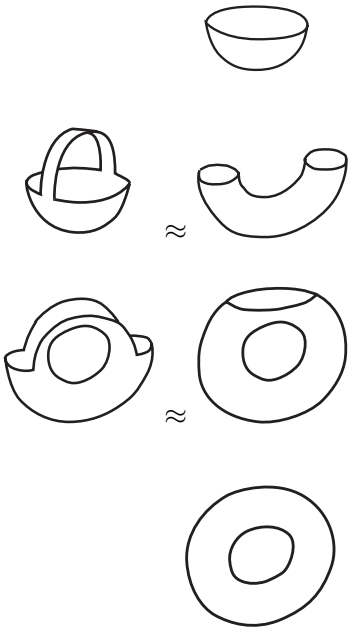


Fig. 3.2: Handlebody decomposition of a torus, $S_4 = (((S_0 + H_0) + H_1) + H_1) + H_2$.

Theorem 1. *(Handlebody Decomposition) For every surface S there is a finite sequence of surfaces $\{S_i\}$, $i = 0..N$, such that $S_0 = \emptyset$, $S_N = S$ and the surface S_i is obtained by attaching a handle $H_\lambda = (A_\lambda, B_\lambda)$ to the boundary of S_{i-1}. This sequence is called the Handlebody Decomposition of S.*

Figure 3.2 illustrates the handlebody decomposition of a torus, $S_4 = (((S_0 + H_0) + H_1) + H_1) + H_2$.

When a handle $H_\lambda = (A_\lambda, B_\lambda)$ is attached to the boundary of S_{i-1} to obtain S_i, a topological change is generated and such change depends *only* on the index λ.

Theorem 2. *If S_i is obtained by attaching the handle H_λ to S_{i-1}, then $\chi(S_i) = \chi(S_{i-1}) + (-1)^\lambda$.*

As a consequence, the Euler characteristic of a surface S provided with a handlebody decomposition $\{S_i\}$, $i = 0..N$ is

$$\chi(S) = |H_0| - |H_1| + |H_2|$$

where $|H_k|$, $k \in \{0, 1, 2\}$ corresponds to the number of handles of type k in $\{S_i\}$. For example, in the handlebody decomposition of the torus in Figure 3.2, there are one handle H_0, 2 handles H_1, and 1 handle H_2. The formula above is, then, verified, since the Euler characteristic of a torus is zero. This is a new topological invariant introduced by the Handlebody theory.

Handles can be attached to an orientable surface with boundary in such a way to preserve its orientability, i.e., the identification has to be coherent. If one starts with an orientable surface, then after attaching a handle coherently the surface is again orientable.

We observe that if we keep track the number of connected components and the number of boundary curves, we can easily calculate the number of genus on the surface and classify it whenever it is necessary. We are to present how to count those two numbers by studying the topological changes caused by a handle attachment that preserves the orientability.

The topological change generated by a handle attachment of index 0 is a creation of a new surface component (see S_1 in Figure 3.2). This handle attachment increases the Euler characteristic by one.

When the handle H_1 is coherently attached to a surface S_i, three situations can occur:

1. The set A_1 is attached to disjoint intervals on the same boundary curve component. In this case, the topological change is a inclusion of a new

boundary curve component in the surface (see S_2 in Figure 3.2).

2. The set A_1 is attached to intervals on different boundary curve components of a surface component. The topological change is here characterized by the creation of a new genus on the surface. In addition, the number of boundary curve components decreases (see S_3 in Figure 3.2).

3. The set A_1 is attached to intervals on different surface components. Here, a boundary curve component and a surface component is removed.

In these three situations, when a handle H_1 is attached coherently to S_{i-1} to obtain S_i, we have $\chi(S_i) = \chi(S_{i-1}) - 1$. Notice that, all of them alter the number of boundary curves. Moreover, the last one also changes the number of connected components on the surface.

Handles of index 2 close a boundary curve component (see S_4 in Figure 3.2).

Concluding, there are three types of handles and five different situations in which they can be attached to a boundary surface.

3.1.2 Stellar Theory

In the previous section, we have seen how to change the topology of a manifold. Now, we will see how to manipulate the structure of a combinatorial surface *without* modifying its topology, which is the main point of Stellar theory [1, 33, 34, 25].

As we have seen in chapter 2, the link and the star of a simplex σ provide a combinatorial description of the neighborhood of σ. We can use them to

define certain changes in a triangle mesh, without modifying essentially (i.e., "topologically") that neighborhood. That is, we do not want to change the topology of the realization of the surface in $\mathbb{R}^3$.

The *stellar operations* provide a such change. They comprise *bistellar moves* and *stellar subdivision*:

Definition 7. *Let K be an n-dimensional simplicial complex. Take an r-simplex $\sigma \in K$, and an $(n-r)$-simplex $\tau \notin K$, such that $link(\sigma, K) = \partial\tau$. Then, the operation $\kappa(\sigma, \tau)$, called* bistellar move*, consists of changing K by removing $\sigma \star \partial\tau$ and inserting $\partial\sigma \star \tau$.*

The bistellar moves are atomic operations that make local changes to the neighborhood of an simplex, while maintaining the integrity of its combinatorial structure.

In the case of combinatorial surfaces, there are three types of bistellar moves, for $\dim\sigma = 2, 1, 0$, called 2-move, 1-move, and 0-move. They are shown in figure 3.3.

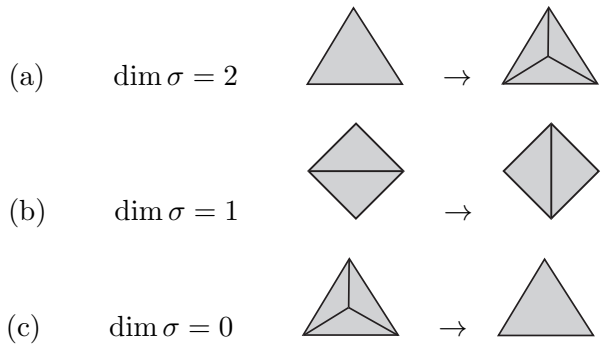


Fig. 3.3: Two dimensional bistellar moves.

The fundamental result of the Stellar theory is given by the following theorem:

Theorem 3. *([33], [34]) Two combinatorial surfaces are piecewise linearly homeomorphic if and only if they are bistellar equivalent.*

The above result guarantees that bistellar moves can change any triangulation of a closed piecewise linear manifold to any other.

A version of this theorem for manifolds with boundary uses all stellar operations, including stellar subdivision [34].

Definition 8. *Let K be a 2-dimensional simplicial complex, take an r-simplex $\sigma \in K$ and a vertex ν in the interior of σ. The operation (σ, ν) removes $\overline{star}(\sigma, K)$ and replaces it with $\nu \star \partial\sigma \star link(\sigma, K)$ is called a* stellar subdivision. *The inverse operation $(\sigma, \nu)^{-1}$ is called a* stellar weld.

Note that, some of the stellar subdivision and welds are also stellar moves. For example, in the two dimensional case, for $\dim \sigma = 2$, see $\kappa(\sigma, \nu)$ and $\kappa(\nu, \sigma)$ that are shown in the top and bottom rows of figure 3.3.

The new operation in two dimensions, is the stellar subdivision on edges, called 1-split. It is shown in figure 3.4 the interior edge case and in figure 3.5 the boundary edge case.

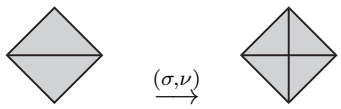


Fig. 3.4: Two dimensional stellar subdivision on interior edges.

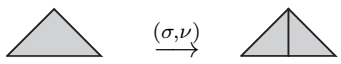


Fig. 3.5: Two dimensional stellar subdivision on boundary edges.

Stellar subdivision is a very powerful concept and it is the cornerstone of Stellar theory. Here, we will only mention some results of the stellar subdivision theory [1].

Proposition 2. *Any stellar move, $\kappa(\sigma, \tau)$, is the composition of a stellar subdivision and a weld, namely $(\tau, \nu)^{-1}(\sigma, \nu)$.*

This result can be easily seen through an example, shown in figure 3.6.

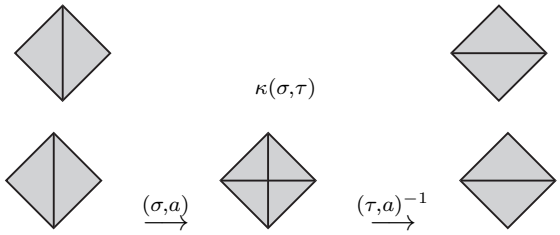


Fig. 3.6: A bistellar move on an edge can be decomposed into a subdivision and an weld.

Proposition 3. *Any stellar operation can be decomposed into a finite sequence of elementary stellar operations on* edges*.*

This result is even stronger than the previous one. It basically allows us to restate the main theorem of Stellar theory only in terms of operations on edges.

3.2 Computational Framework

The purpose of this section is to introduce a new representation for surfaces based on the concepts of Handlebody and Stellar theories. It consists of a topological data structure that describes the incidence and adjacency relations on a combinatorial surface with or without boundary. It also includes operators for building/unbuilding meshes and to change the structure and resolution of a mesh.

We remark that, although the Handlebody theory can be applied to general combinatorial manifolds, the Stellar theory is restricted to simpli-

cial complexes. Therefore, from now on, we will focus primarily on triangular meshes. This is not a limitation, since any manifold surface can be triangulated and, in practice, triangular meshes are a common choice in applications.

3.2.1 Handle Operators

The Handlebody theory presented in Section 3.1.1 studies the topological changes in a surface caused by a handle attachment. There are three types of handles to build a handlebody decomposition of a surface. From a combinatorial point of view, three types of operators are to be defined to represent the handle attachments:

- Handle operator of type 0 – This operator creates a new combinatorial surface component with only one triangle (see Figure 3.7).

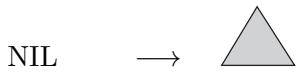


Fig. 3.7: Handle operator of type 0 (triangle creation).

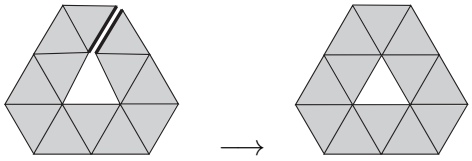
- Handle operator of type 1 – The purpose of this operator is to identify two given boundary edges *with no vertices in common*. There are three situations for this group:

  Case 1: the boundary edges are on different surfaces. In this case the operator attaches the surfaces and removes one boundary curve (see Figure 3.8(a)).

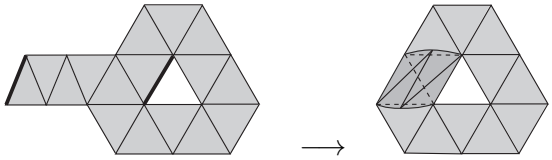
  Case 2: the given boundary edges are incident to the same boundary curve. The operator splits the boundary curve into two different components (see Figure 3.8(b)).

Case 3: the boundary edges are on different boundary curves on a surface component. It creates a new genus in the surface and reduce in one the number of boundary curve components of the surface (see Figure 3.8(c)).

(a) Boundary edges belong to different surfaces



(b) Boundary edges belong to the same boundary curve of a surface

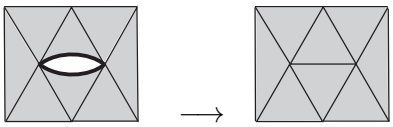


(c) Boundary edges belong to different boundary curves of a surface

Fig. 3.8: Handle operator of type 1 (joining boundaries).

- Handle Operator of type 2 – In this group the operator identify two given boundary edges *with two vertices in common*. The operator closes one boundary curve component and transform those boundary vertices into two interior vertices (see Figure 3.9).

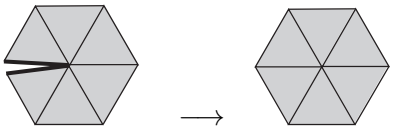
According to the definitions above, we notice that if a Handle operator of type λ is applied to a combinatorial surface S_1 to obtain S_2, then $\chi(S_2) = \chi(S_1) + (-1)^{\lambda}$. This is a direct consequence of theorem 2.



(a) Boundary edges have two vertices in common

Fig. 3.9: Handle operator of type 2 (closing boundaries).

One can observe that the Handle operators of type 1 and type 2 identify two boundary edges to make an interior edge. The first is applied when the edges have no vertices in common, and the second when the edges have two vertices in common. Thus, there is one missing case to consider: when the boundary edges have one vertex in common. So, it is suitable to define the Zip operator, which identifies two boundary edges with one vertex in common. This operator removes one edge and one vertex, then it doesn't change the Euler characteristic of the surface. Its main purpose is to close the vertex link (see Figure 3.10). In fact, such operator can be derived from the Handle operators together with their inverse. However, it is very convenient to have a direct implementation of it.



(a) Boundary edges have one vertex in common

Fig. 3.10: Zip operator.

Inverse Handle Operators

There is an inverse operator not only for each handle operator, but also for the Zip operator. The topological changes caused by their inverse operation are now described.

The inverse handle operator of index zero destroys a triangle. Inverse handle operators of index 1 and index 2 split an interior edge into two boundary edges. There are five cases to consider when splitting an interior edge. Such cases are distinguished according to the number of boundary vertices incident to the interior edge that will be operated, which could be 2,1 or 0. The inverse handle operator of type 1 is used when the incident vertices to the interior edge are both in the surface boundary. The inverse handle operator of type 2 is applied when the incident vertices of the interior edge are on the interior of the surface. In the last case, when the interior edge has one vertex in the boundary, one should use the inverse Zip operator.

The topological changes caused by an inverse handle operator of index 1 when applied to a given interior edge e, depend on the answer to the following question:

Are the incident boundary vertices to e on different boundary curve components?

If the answer is affirmative then the inverse handle operator will remove one boundary curve component (see the inverse operation in the Figure 3.8(b)). In contrary, the second question has to be answered.

Are those edges on the same boundary curve component?

When the vertices are incident to the same boundary curve, the inverse operation not only will add a new boundary curve component to the surface but also it will either remove a genus (see inverse of Figure 3.8(c)) or disconnect the surface (see inverse of Figure 3.8(a)).

Inverse operator of index 2 splits an interior edge with zero incident boundary vertices. The topological change in this situation is an addition of a new boundary curve to the surface.

The inverse Zip operator (the unzip op.) is applied when the interior

edge e has one incident vertex on the boundary. It simply splits an interior edge and transforms an interior vertex into a boundary vertex.

With the set of operators presented above one can easily build and unbuild all kinds of orientable combinatorial surfaces, with or without boundary. Handle operators shall be used to perform paste operations on the surface, while the inverse operators shall be used to make cut operations.

3.2.2 Stellar Operators

The Stellar theory presented in Section 3.1.2 studies structural modifications to the neighborhood of a simplex that do not alter the topology.

These modifications are the stellar moves, stellar subdivision and welds. They can be used to change the connectivity and the resolution of a mesh.

We classify the stellar operators in terms of their effect in the number of faces, $|F|$, in the mesh. Accordingly, there are three groups of operators: *isolevel*, *refinement*, and *simplification*.

The isolevel operators do not change $|F|$. The operator in this group is the bistellar 1-move, also called 1-flip (or edge flip). It simply exchanges two existing triangles by two new triangles. This operator is shown in Figure 3.3(b).

The refinement operators increase $|F|$, and the resolution of the mesh. The operators in this group are the 2-split (face split), and 1-split (edge split). The face split replaces one existing triangle with three new triangles, and thus, it increases $|F|$ by 2. This operator is shown in Figure 3.3(1). The edge split replaces two existing triangles sharing that edge with four new triangles, when the edge is an internal edge. When the edge is a boundary edge, it replaces one existing triangle with two new triangles. This operator

increases $|F|$, by 1 or 2, depending of whether the edge belongs to the boundary or not.

The simplification operators are the inverse of the refinement operators. The inverse of the face split is the face weld, and the inverse of the edge split is the edge weld. Observe that, weld operations $(\sigma, \nu)^{-1}$, are specified through a vertex ν, whose star defines the neighborhood to be changed.

At this point it is appropriate to note that stellar operators can be used as primitives to define other multiresolution operators. For example, edge collapse and its inverse, vertex split, can be decomposed into a sequence of elementary stellar operations. This is a natural consequence of Theorem 3. More specifically, the edge collapse is given by a composition of edge flips and a final edge weld, while the vertex split is given by an edge split composed wit a sequence of edge flips. This is shown in Figure 3.11.

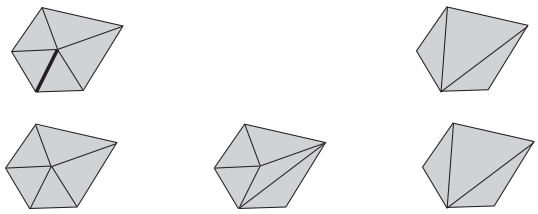


Fig. 3.11: Decomposition of an edge collapse (top) into an edge swap followed by an edge weld (bottom).

We remark that stellar operations are more flexible in general. In the case of edge collapse / vertex split, it is easy to see that there are many possible sequences of edge flips leading to the final edge weld. Therefore, these edges flips can be chosen such that the quality of the mesh is improved (for example, a measure of aspect ratio).

3.3 Mesh Representation

We will follow the same data structure by Velho et al. [43] to represent 2D meshes. This structure has the advantage of unifying the functionalities of the topological operators . We will rewrite them here.

A mesh is structured as $M = (V, E, F, B)$ where V, E, F, B are collections of vertices, edges, faces and boundary curves, respectively.

```
struct Surface {
  Container<Face*> faces;
  Container<Edge*> edges;
  Container<Vertex*> vertices;
  Container<Edge*> bndries;
}
```

The face stores a pointer for the first half-edge of the internal cycle.

```
struct Face {
  Half_Edge* he_ref;
}
```

A edge is formed by two half-edges. If it is representing a boundary edge, one of the half-edges points to a null face.

```
struct Edge {
  Half_Edge he[2];
}
```

The half-edge is the core element of the data structure. It stores a pointer for its initial vertex, a pointer for the next half-edge in the cycle of the face and pointers for the edge and face to which it belongs. Notice that the mate

half-edge may be accessed by the parent edge pointer.

```
struct Half_Edge {
  Vertex* org;
  Half_Edge* next;
  Face* f;
  Edge* e;
}
```

The vertex stores a pointer for the incident half-edge.

```
struct Vertex {
  Half_Edge* star_i;
  Data d;
}
```

In the collection of boundary curves, the representative of each element is an edge that belongs to such curve.

3.4 Application Program Interface (API)

The basic topological operators allow queries and navigation of the mesh structure. They are: c = link(s); and c = star(s). Note that they can take as arguments a simplex s of dimension 0 (vertex), 1 (edge) or 2(face). In our current implementation, we use only the vertex star, which returns an adjacency iterator object c, called *circulator* [30]. We also have the basic operators of the edge algebra [19]: v = org(e) (origin vertex v of a half edge e); f = left(e) (face f to the left of a half edge e); h = sym(e) (symmetric half edge h); and n = lnext(e) (next half edge n on left face). These operators are trivially computed from the data structure.

The handlebody operators allow cutting and pasting surfaces. They are:

- f = create(p0, p1, p2): creates a new face f from the points p0, p1 and p2 of S;
- (p0, p1, p2) = destroy(f): destroys an existing face and return its three points;
- e = glue(he0, he1): "identifies" two boundary half-edges and turns them into an interior edge which is returned;
- (he1, he2) = unglue(e): divides an interior edge in two boundaries and returns them.

The star operators allow us change the resolution (number of points in the mesh) and the mesh combinatorial structure. They are:

- e = flip(e): makes a swap in the edge e and returns the same edge. Notice that the flip operator is defined only for internal edges;
- v = split(f): trisects the face f and returns a new vertex which is added to S;
- f = weld(v): an inverse operator of the split operator, it returns one face and removes a vertex from S.

For convenience, as we will see later, we need operators that increase or decrease the number of points between the levels without changing the mesh structure. They actuate only on the set of points of S. They are: add(s) and remove(s), where $s \in S$.

Observe that among the topological operators, only two increase the resolution: create and split. The main difference between them is that create

operator sometimes has dependencies on the set S and needs to be preceded by add operator. On the other hand, split operator has a dependence on some face that should be in K. Analogously, we can compare destroy and weld as we simplify the mesh.

3.5 Composition of Operators

Let Δ be a topological operator and consider the pair (S, K). We denote by $\Delta(S, K)$ as the *action* of Δ over (S, K) that also generates a new pair (S', K').

We have nine topological operators over pairs and we can clearly observe that they are invertible. More precisely, we say that Δ^{-1} is the inverse of Δ if $\Delta^{-1} \circ \Delta = \Delta \circ \Delta^{-1} = Id$, where Id is the identity operator, that is, $Id(S, K) = (S, K)$. All topological operators are invertible. Thus, we have: $\mathsf{add}^{-1} = \mathsf{remove}$, $\mathsf{destroy}^{-1} = \mathsf{create}$, $\mathsf{glue}^{-1} = \mathsf{unglue}$, $\mathsf{flip}^{-1} = \mathsf{flip}$ e $\mathsf{split}^{-1} = \mathsf{weld}$.

The definitions below formalize the topological *junction* and *disjunction* operations over nested family of meshes.

Definition 9. *Let (S_0, K_0) and (S_1, K_1) be two distinct pairs. We say that $(S_1, K_1) = (S_0, K_0) \oplus s$ is a topological junction of a point s with (S_0, K_0) if $S_1 = S_0 \cup \{s\}$ and there exists a sequence of topological operations Δ_i, such that*

$$\Delta_n \circ \Delta_{n-1} ... \circ \Delta_1(S_0, K_0) = (S_1, K_1).$$

Definition 10. *Let* (S_0, K_0) *and* (S_1, K_1) *be two distinct pairs. We say that* $(S_1, K_1) = (S_0, K_0) \ominus s$ *is a topological disjunction of a point* s *with* (S_0, K_0) *if* $S_1 = S_0 - \{s\}$ *and there exists a sequence of topological operations* Δ_i *such that*

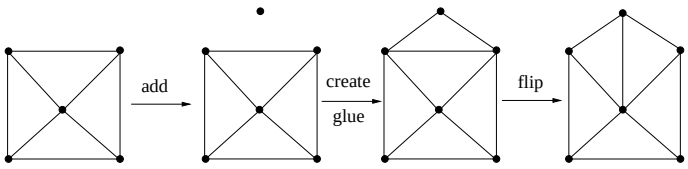
$$\Delta_n \circ \Delta_{n-1} ... \circ \Delta_1 (S_0, K_0) = (S_1, K_1).$$

It follows, directly from the operators inversion property that the topological junction and disjunction are also invertible. More precisely:

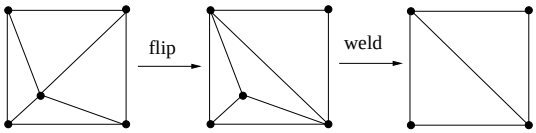
Property 1. $(S_1, K_1) = (S_0, K_0) \oplus s \Rightarrow (S_0, K_0) = (S_1, K_1) \ominus s$. *In another way:*

$$\Delta_n ... \circ \Delta_1 (S_0, K_0) = (S_1, K_1) \Rightarrow \Delta_1^{-1} ... \circ \Delta_n^{-1} (S_1, K_1) = (S_0, K_0).$$

Compositions of topological operators is relevant for generating sequences of operations from a base mesh. For instance in chapter 6 we will generate a particular sequence named *Filtering*. In figure 3.12 we exhibit examples of insertion and removal of points by means of a sequence of topological operators.



(a) Example of topological junction.



(b) Example of topological disjuntion.

Fig. 3.12: Examples of insertion and removal of points.

4. SOLID SAMPLING AND INTERPOLATION

In this chapter we present the Poisson Disk Sampling (PDS), an important class of stochastic sampling often used in Computer Graphics applications. We define PDS's for two different domains: firstly for the whole plane $\mathbb{R}^2$ and secondly for solids $R \in \mathbb{R}^2$ (open, limited and connected). We analyse their intrinsic relationship with Solid Alpha Complexes.

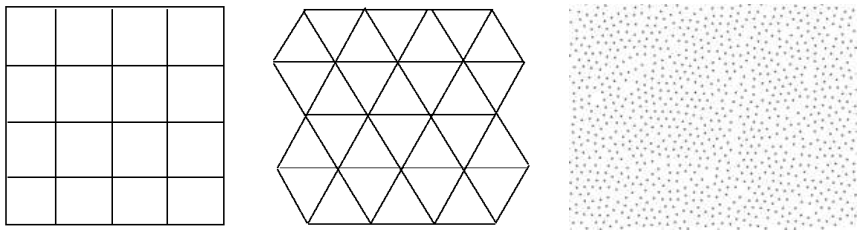
4.1 Poisson Disk Samplings...

... in the Plane

Definition 11. *Let $S_\alpha = \{s_1, s_2, ...\}$ be a sampling in the plane. We say that S_α is a Poisson Disk Sampling if (i) $\cup_{s_i \in S_\alpha} B_\alpha(s_i) = \mathbb{R}^2$ and, additionally, (ii)$S_\alpha \cap B_\alpha(s_i) = \{s_i\}$, $\forall i$. The condition (i) will be named covering condition and condition (ii) will be named Poisson condition.*

Proposition 4. *There exists a PDS in the plane.*

Proof. Trivial examples of PDS's are the regular *lattices* as the quadrilateral (figure 4.1.a) and the triangular (figure 4.1.b). Non-trivial examples examples may be created by means of the dart throwing algorithm [8]. In this approach, we have a random generator of samples in a given region and a validator that checks if they satisfy the geometric criteria expected. In our case we are considering the whole plane as the sampling region and the the



(a) (b) (c)

Fig. 4.1: PDS's examples: (a) quadrilateral lattice, (b) triangular lattice, (c) stochastic sampling.

Poisson condition as geometric criteria. If a sample is validated, then it is thrown to the output, otherwise, we discard it. The algorithm is interrupted when the coverage condition is satisfied. This is not feasible because there is an infinity number of points to be sampled in the plane. Therefore, it is only applicable for limited domains as we define soon. This algorithm is a typical example of stochastic sampling also known as blue noise [20]. See, for instance figure 4.1.c. □

Next we stablish the intrinsic relationship between a PDS and its solid alpha complex.

Proposition 5. *The Solid Complex Alpha of a PDS S_α of the plane, $\overline{C_\alpha(S_\alpha)}$, is a coverage of the plane.*

Proof. It is sufficient to show that every triangle $\sigma \in DT(S_\alpha)$ satisfies $r_\sigma < \alpha$ where r_σ is the circumscribed circle radius of σ. Let c_σ be the center of the circumscribed circle of σ. If, as an absurd, $r_\sigma > \alpha$ then directly from the Delaunay property we would have $d(c_\sigma, S_\alpha) > \alpha$ and, therefore, $c_\sigma \notin \cup_{s_i \in S_\alpha} B_\alpha(s_i)$. □

The proposition above has as consequence, two important facts. The first one is the quality of the planar subdivision in triangles. There is an

upper bound for the aspect-ratio(σ) = $\frac{L^2}{vol(\sigma)}$ where $vol(\sigma)$ is the triangle area of σ and L is the length of the longest edge of σ. In Medeiros et al. [29] it is evidenced that $\frac{L^2}{vol(\sigma)} \leq 4\sqrt{3}$ where the equality holds when the wider angle is $2\pi/3$. This material is also available in appendix A as a partial result of the thesis.

The second consequence is the scale control power of the simplicial elements. It is easy to see that by the sampling conditions, the radius of the circumscribed circles of the triangles and the edges lengths are varying in the interval $[\frac{\alpha}{2}, \alpha]$. Therefore, it is reasonable think in multiresolution representations, using these samplings, where the scales are controlled by the α parameter.

... in Bi-dimensional Solids

Next we define the Poisson disk samplings for a class of solids on the plane. In particular, for such solids, the boundary has a crucial importance: it defines both the shape and the topology of the solid.

Definition 12. *Let $S_\alpha = \{s_1, s_2, ..., s_n\}$ be a sampling of the solid region R ($R = \overline{A}$, A limited, open and connected). We say that S_α is a Poisson disk sampling (PDS) if $R \subset \cup_{s_i \in S_\alpha} B_\alpha(s_i)$ (covering condition) and, additionally, $S_\alpha \cap B_\alpha(s_i) = \{s_i\}$, $\forall i$ (Poisson condition).*

Observe that differently from the previous definition, the equality does not hold in the coverage condition. However, depending on the sampling radius[1] we can approximate the covering of balls to the region shape as much as intended. We will see this fact detailed in proposition 7.

[1] We will always be referring to the parameter α of Poisson disks sampling.

Analogously to $\mathbb{R}^2$ domains we have existence of PDS's for solid regions. The idea of generating them is to include a new validation criteria of the candidate sample points by discarding those which lies outside the domain. Therefore, we have also the existence of an infinity of them. We name these samplings as *approximating* sampling.

From here, whenever we refer to a solid R region, it will be open, connected and limited unless the contrary stated.

In the definition below, each sample point may be classified in accordance to its topology.

Definition 13. *Let S be a sampling of a solid region R. We say that $s \in S_\alpha$ is boundary (interior) if $s \in \partial R$ ($s \in int(R)$).*

From definition above we can obtain a new class of PDS's: *interpolating* samplings. The only difference from approximating samplings is that it takes in consideration the boundary at sampling time.

Definition 14. *Let $S_\alpha = \{s_1, s_2, ..., s_n\}$ be a sampling of R. We say that S_α is an interpolating sampling if exists $A \subset S_\alpha$ such that $A \subset \partial R$ and $\partial R \subset \cup_{s_i \in A} B_\alpha(s_i)$. We will denote an interpolating PDS by $\tilde{S}_\alpha$.*

A simple adaptation of the dart throwing algorithm gives an interpolating PDS sampling. We separate the algorithm in two steps. In the first one, we restrict the target sampling to the boundary of the region and later, we sample the subregion defined by its interior minus the covering of the disks in the boundary.

4.2 Shape Approximation

Definition 12 leads us to questions about shape approximations by means of solid alpha complexes. Firstly we state some relevant definitions to express and proof results.

Definition 15. *The weighted squared distance of one point $x \in \mathbb{R}^2$ from one ball b is given by $\pi_b(x) = ||x - c_b||^2 - r_b^2$ where c_b and r_b are the center and the radius of b, respectively.*

An important observation is that a point $x \in \mathbb{R}^2$ belongs to a ball if and only if $\pi_b(x) \leq 0$, and it belongs to the boundary of the ball if and only if $\pi_b(x) = 0$. Given a finite set of balls $\mathcal{B}$, we can divide the space into regions:

Definition 16. *The Voronoi region of a ball $u \in \mathcal{B}$ is the set of points of the plan for which u minimizes the weighted distance,*

$$V_u = \{x \in \mathbb{R}^2 | \pi_u(x) \leq \pi_v(x), \forall v \in \mathcal{B}\}$$

The diagram comprising the Voronoi regions is called, in the literature, a *power diagram*. It is not difficult to show that the set of points equally distant from two balls u and v is the hyper plane defined by $\pi_u = \pi_v$.

Notice that the power diagram of a PDS coincides to the Voronoi Diagram since the balls are Poisson disks with common radius.

Definition 17. *The dual complex K of $u \in \mathcal{B}$ is a collection of simplices*

$$K = \{conv\{T\} \mid T \subseteq \mathcal{B}, \bigcap_{u \in T} (u \cap V_u) \neq \emptyset\}.$$

This duality can be visualized in figure 4.2. Observe that the Voronoi potential regions decompose the union of balls in convex regions $V_u \cap u$.

Fig. 4.2: Dual complexes of three balls. (left) $(u \cap v \cap w) = \emptyset$, (right) $(u \cap v \cap w) \neq \emptyset$.

The proposition below characterize the dual complex of Poisson disks.

Proposition 6. *Let $\mathcal{S}_\alpha$ be a PDS of R. Then the dual complex of $\bigcup_{s \in S_\alpha} B_\alpha(s)$ is the Alpha Complex $C_\alpha(\mathcal{S}_\alpha)$.*

Proof. See [14]. □

This proposition give us a different manner of defining alpha complexes, differently from definition 5 in chapter 2. Depending on the objective to be achieved, one can use the more suitable one.

On geometric approximation of R by means of the complexes $C_\alpha(S_\alpha)$ we will establish a global convergence based on the parameter α.

Proposition 7. *Let $\alpha > 0$ and S_α be a PDS of R. The Hausdorff distance $d_H(R, |C_\alpha(S_\alpha)|)$ is lower than α.*

Proof. Notice that $\forall p \in R$ there exists $q \in S_\alpha$ such that $p \in B_\alpha(q)$. Therefore $d_H(R, \bigcup_{s \in S_\alpha} B_\alpha(s)) < \alpha$. Since $|C_\alpha(S_\alpha)| \subset \bigcup_{s \in S_\alpha} B_\alpha(s)$ then we get $d_H(R, |C_\alpha(S_\alpha)|) < \alpha$. □

Generally, algorithms that calculate alpha complexes use Delaunay triangulation as an intermediate stage. However, having in hands the next proposition, we can use the ball-pivoting algorithm restricted to the plane (RBPA) [28] to build the solid alpha complexes in linear time. Thus, precomputing Delaunay triangulation is not necessary.

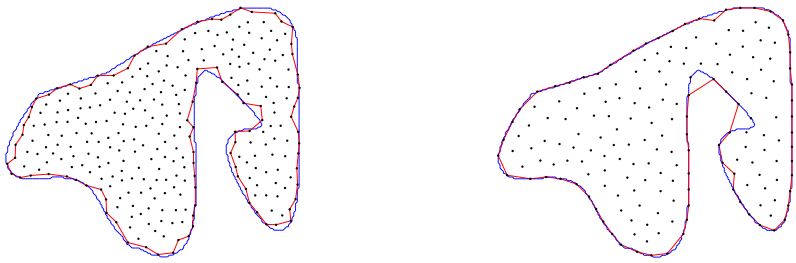
Proposition 8. *Let $S \subset \mathbb{R}^2$ be a set of points in general position and $\overline{C_\alpha(S)}$ its solid alpha complex. Consider $\mathcal{T}_\alpha$ as the output of RBPA being performed in the plane. Then $\overline{C_\alpha(S)} = \mathcal{T}_\alpha$.*

Proof. See Medeiros et al. [28] or appendix C. □

Next we define a new complex interpolation for Poisson disk samplings that assures topological correspondence at least for interior sample points, i. e., if $p \in S_\alpha$ is an interior sample point then $p \in int(\mathcal{C}_\alpha(S_\alpha))$.

Improving Boundary Approximation

The complex $\mathcal{C}_\alpha(S_\alpha)$ of approximating samplings S_α are noisier in the boundary than the complex $\mathcal{C}_\alpha(\tilde{S}_\alpha)$ of interpolating samplings $\tilde{S}_\alpha$ (see figure 4.3). Indeed, rarely the approximating samplings take representative points of the boundary whereas interpolating samplings take much more. Consequently the complex $\mathcal{C}_\alpha(\tilde{S}_\alpha)$ give better boundary approximations. However they have a small set of interior points that lies in the boundary (see figure 4.3.b). To avoid this problem we propose a new complex interpolation named *Quasi Alpha Complex*.



(a) (b)

Fig. 4.3: Examples of reconstructions of an approximating sampling (a) and an interpolating sampling (b).

First let us define *good* approximations of regions, reminding that the symbol ˜ means that the sampling is interpolating.

Definition 18. *Let $(\tilde{S_\alpha}, K_\alpha)$ be an α-pair of solid region R. Consider $F = \partial\overline{K_\alpha}$. We say that K_α is a good approximation of R iff all points of F are boundary.*

In order to assure topological correspondence of interior sample points the *Quasi-Alpha Complex* has characteristics combined from definitions of a Solid Alpha Complex and of a good approximation.

Definition 19. *Let S be any sampling with topological informations in the points (boundary or interior) of a region R. Given $\alpha > 0$, we denote $QC_\alpha(S)$, as a Quasi-Solid Alpha Complex (QSAC), the solid simplicial complex of S that satisfies the following properties:*

1. $QC_\alpha(S) \subset Del(S)$
2. $\sigma_{(s_i,s_j,s_k)} \in QC_\alpha(S) \iff$
   (a) *or $\mu_\sigma < \alpha$, where μ_σ is the circumscribed radius of σ;*
   (b) *or $\{s_i, s_j, s_k\} \cap int(R) \neq \emptyset$;*
   (c) *or $\exists\ B_l$ e B_m, distinct connected components of the boundary R, such that $B_l \cap \{s_i, s_j, s_k\} \neq \emptyset$ and $B_m \cap \{s_i, s_j, s_k\} \neq \emptyset$;*

Notice that by property 2(a) it follows that $\overline{C_\alpha(S)} \subset QC_\alpha(S)$. From definition above, we have the following lemma:

Lemma 1. *If S is an interpolating PDS of a region R then:*

1. *$QC_\alpha(S)$ is a good approximation;*

2. If B_l is a connected component of the boundary of $QC_\alpha(S)$ then all its points lies in the same connected component of the boundary of R.

Proof. The item (1) follows directly from property 2.b of definition 19. The item (2) is a direct consequence of property 2.c in definition 19. □

We conclude that the α-pair $(\tilde{S_\alpha}, QC_\alpha(\tilde{S_\alpha}))$ is a good approximation such that if $p \in \partial QC_\alpha(\tilde{S_\alpha})$ then $p \in \partial R$. Moreover the complexes $QC_\alpha(\tilde{S_\alpha})$ have the relevant property of being a subset of the Delaunay triangulation and a super set of the solid alpha complex.

Improving the Aspect Ratio

Unfortunately, we have a drawback from the fact that the complex $QC_\alpha(\tilde{S_\alpha})$ do not have an upper bound for the aspect ratio of the triangles. This problem takes place since interior points sampled too close to the boundary eliminate, in their neighbourhood, the possibility of sampling boundary points that would be more representatives (see figure 4.4). To solve this problem, we propose a new sampling strategy where interior points are not sampled in a tubular neighbourhood $\epsilon(\alpha)$ of the boundary. However, we need to use a weaker class of PDS's that we named (α, β)-*Poisson Disk Sampling*.

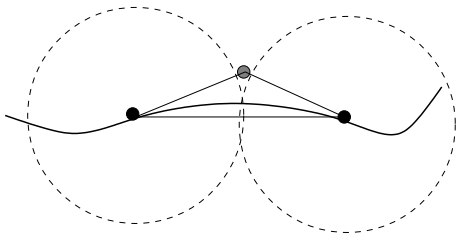


Fig. 4.4: In this example the light gray point is in the interior and is close to the boundary. The plotted triangle may have a very wide aspect ratio.

Definition 20. *Let $S_{\alpha\beta} = \{s_1, s_2, ..., s_n\}$ be a sampling of a solid region R ($R = \overline{A}$, A open, connected and limited) and $0 < \beta \leq \alpha$. We say that $S_{\alpha\beta}$ is a (α, β)-PDS if (i) $R \subset \cup_{s_i \in S_{\alpha\beta}} B_\alpha(s_i)$ (covering condition) and (ii) $S_{\alpha\beta} \cap B_\beta(s_i) = \{s_i\}$, $\forall i$ (Poisson condition).*

Notice that as parameter β approaches α, the sampling tends to be a PDS. The scale notion is implicit in parameter β. The following algorithm exhibits a way to generate a (α, β)-PDS assuring that $\beta = \alpha/2$.

Proposition 9. *Given $\alpha > 0$, we can generate the sample of points $S = I \cup B \cup P$, as an $(\alpha, \alpha/2)$-PDS of the solid region R from the following steps:*

1. *$I = \{$ PDS of $R - \cup_{s \in \partial R} B_\alpha(s)\}$;*
2. *$B = \{$ PDS of $\partial R\}$;*
3. *$P = \{$ "PDS" of $R - \cup_{s \in B \cup I} B_\alpha(s)$ with conditional projection in the boundary$\}$. The conditional projection is performed after the generation and the validation of a sampling s. If $dist(s, \partial R) < \frac{\alpha}{2}$ then select s' as the nearest point of ∂R, else, select s.*

Proof. By construction it is clear that $R \subset \cup_{s \in I \cup B \cup P} B_\alpha(s)$. We will show that $\beta = \alpha/2$. Let s' be the projection of s in ∂R. Let $p \in \partial R$ be the sampling point nearest to s. We know that $ss' < \alpha/2$ and that $ps > \alpha$. By the triangular inequality, we have that $ps' + ss' > ps \Rightarrow ps' > ps - ss' > \alpha - \alpha/2 > \alpha$. Then $\beta = \alpha/2$. □

Observe that if $s \in int(R)$ from steps 2 and 3 we conclude that $dist(s, \partial R) > \alpha/2$. Then, there exists a neighbor $\epsilon(\alpha) = \alpha/2$ of the boundary that does not contain interior points. As mentioned before this improve the aspect ratio of the triangles along the boundary of an α-par $(\tilde{S_{\alpha,\beta}}, QC_\alpha(\tilde{S_\alpha}))$. At

this moment we are not able to show an upper bound but we conjecture that it is ≤ 8 (we have generated some examples and all are satisfactory and below of this limit).

To complete this section we define the class of interpolating (α, β)-PDS's.

Definition 21. *Let $S_{\alpha,\beta} = \{s_1, s_2, ..., s_n\}$ be a sampling of a solid region R. We say that $S_{\alpha,\beta}$ is a interpolating (α, β)-PDS if exists $A \subset S_{\alpha,\beta}$ such that $A \subset \partial R$ and $\partial R \subset \cup_{s_i \in A} B_\alpha(s_i)$. Let us denote a interpolating (α, β)-PDS as $\tilde{S}_{\alpha,\beta}$.*

Notice that according to definition above the (α, β)-PDS, from proposition 9, is always an interpolating.

4.3 Topology Approximation

Now, we provide some tools that allow we make conclusions about the topology of the approximated triangulation. Actually, we are looking for the topological relationship between the union of the balls of an PDS and its dual alpha complex.

Definition 22. *A deformation retraction of a space $\mathbb{X}$ onto a subspace $\mathbb{A}$ is a family of maps $f_t : \mathbb{X} \to \mathbb{A}, t \in [0, 1]$ such that f_0 is the identity map, $f_1(\mathbb{X}) = \mathbb{A}$ and $f_t|\mathbb{A}$ is the identity, for all t. The family should be continuous, in such a way that the associated map $\mathbb{X} \times [0, 1] \to \mathbb{X}$, $(x, t) \mapsto f_t(x)$ is continuous.*

In other words, starting from the original space $\mathbb{X}$ in time 0, we continuously deform the space to transform it in subspace $\mathbb{A}$ on time 1. A retraction deformation is a particular case of homotopy.

Definition 23. *A homotopy is a family of maps $f_t : \mathbb{X} \to \mathbb{Y}, t \in [0, 1]$, such that its associated map $F : \mathbb{X} \times [0, 1] \to \mathbb{Y}$ given by $F(x, t) = f_t(x)$ is*

continuous. Then , $f_0, f_1 : \mathbb{X} \to \mathbb{Y}$ are homotopic via the homotopy f_t. We denote this as $f_0 \simeq f_1$.

Let us suppose that we have a retraction as in definition 22. we consider $i : \mathbb{A} \to \mathbb{X}$ an inclusion, we have that $f_1 \circ i \simeq id$ and $i \circ f_1 \simeq id$. This will allows us to classify $\mathbb{X}$ and its subspace $\mathbb{A}$ as having the same connectivity. This is a special case of homotopic equivalence.

Definition 24. *A map $f : \mathbb{X} \to \mathbb{Y}$ is called a homotopy equivalence if there is a map $g : \mathbb{Y} \to \mathbb{X}$, such that $f \circ g \simeq id$ and $g \circ f \simeq id$. Then, $\mathbb{X}$ and $\mathbb{Y}$ are homotpy equivalent and have the same homotopy type. This fact is denoted as $\mathbb{X} \simeq \mathbb{Y}$.*

Now we can enunciate an important result about homotopy between two spaces.

Proposition 10. *Let S_α be a PDS of a region R and $\mathcal{B} = \cup_{s_i \in S_\alpha} B_\alpha(s_i)$. Then $\mathcal{B} \simeq |C_\alpha(S_\alpha)|$.*

Proof. The idea of demonstration is to exhibit a retraction that takes space $\mathcal{B}$ in space $|C_\alpha(S_\alpha)|$. See Edelsbrunner [14] for more details. □

The relevance of the proposition is the invariance of the homology between spaces. This relation does not preserve the intrinsic dimension since the alpha complexes may have isolated simplexes of smaller dimensions. However the *homeomorphism* is stronger than homotopy because it preserves the topology.

Definition 25. *A homeomorphism $f : \mathbb{X} \to \mathbb{Y}$ is a bijection, such that f and f^{-1} are continuous. We say that $\mathbb{X}$ is homeomorphic in relation to $\mathbb{Y}$, $\mathbb{X} \approx \mathbb{Y}$ and that $\mathbb{X}$ and $\mathbb{Y}$ have the same topology type.*

To assure topological equivalence between the region and the solid alpha complex, we use *Medial Axis* and *LFS (Local Feature Size).*

Definition 26. *The medial axis of a curve F is the closing of a set of points in the plane that has two or more closest points in F.*

Definition 27. *The Local Feature Size, $LFS(p)$, of a point $p \in F$ is the Euclidean distance of p to the nearest point m from the medial axis.*

In the proposition below we give a condition for PDS's of regions with smooth boundaries[2] so that its dual complex has the same topology of the region.

Proposition 11. *If the radius of a PDS S_α (with smooth boundary) is lower than $\frac{1}{2} inf_{p \in \partial R} LFS(p)$ then $|\overline{C_\alpha(S_\alpha)}| \approx R$.*

Proof. See Medeiros [29]. □

We add smoothness restriction to the boundary to avoid sharp features. Those features give $\inf_{p \in \partial R} LFS(p) = 0$ and there is no manner to determine an upper bound for α. Our plan is to generalize proposition 11 for regions with a finite set of sharp features.

[2] This condition is to assure that exists $\epsilon > 0$ such that $LFS(p) > \epsilon$, $\forall p \in \partial R$.

5. MULTI-SAMPLINGS

After the analysis of one single (α, β)-PDSs of a solid in last chapter, we will interact different (α, β)-PDSs using the scales given by parameters α and β.

5.1 (α, β)-Family

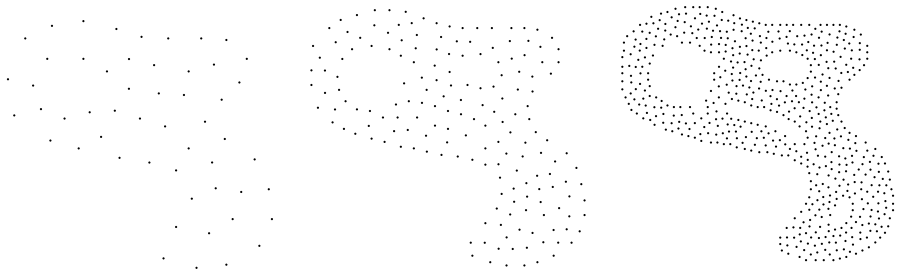
For a given solid there is infinite sets of PDS's depending on parameter α and β and on the randomness of the generating algorithm. We will introduce the concept of (α, β)-*Family* which is a particular case of increasing (decreasing) family of points. Such concept has the purpose of structuring a family of (α, β)-PDS's in such a way that they are nested. This will give us the basic notions of multiresolution representation of solids.

Definition 28. *We say that the monotone family of points $\mathcal{F} = \{S_{\alpha_i \beta_i}\}$ is an (α, β)-Family if its elements are (α_i, β_i)-PDSs of the same solid. We call the indices i's as the scaling levels and the β_i's the scales. If $\alpha_i = \beta_i$ we say simply α-Family.*

Next, we will propose algorithms to build such families. Firstly, we attempt to build (α, β)-families where $\alpha_i = \beta_i$. Then, we attempt to generate another particular family such that $\beta_i = \frac{\alpha_i}{2}$ and whose resolution rate between successive levels is a power of 2.

Building α-Families

For a set of positive reals $\alpha_1 > \alpha_2 > ... > \alpha_n$, we can generate an increasing α-family by the following way: given the sampling S_{α_i} it is sufficient to apply the dart throwing algorithm in $(R - \bigcup_{s \in S_{\alpha_i}} B_{\alpha_i}(s))$ to obtain the next sampling $S_{\alpha_{i+1}}$. In figure 5.1 we have an example of an increasing scaling family with three levels of scale. The algorithm 1 shows this process:



(a) (b) (c)

Fig. 5.1: Example of α-family with three levels.

Algorithm 1: *scaling_family*

Input : Solid R, $\alpha_1 > \alpha_2 > ... > \alpha_n$
Output: $\mathcal{F} = \{S_{\alpha_i}\}$

$S_{\alpha_0} = \emptyset$;
for *each* i **do**
  $S_{temp} = \mathsf{pds}(R - \bigcup_{s \in S_{\alpha_i}} B_{\alpha_i}(s))$;
  $S_{\alpha_{i+1}} = S_{temp} \cup S_{\alpha_i}$;

Building (α, β)-families

Below we present the scheme of an algorithm that builds a particular (α, β) - family.

Proposition 12. *Given $\alpha_1 > 0$, there exists an increasing (α, β)-family $\mathcal{F} = \{\tilde{S}_{\alpha_i\beta_i}\}$ of a solid R such that $\alpha_{i+1} = \alpha_i/2$, $\beta_{i+1} = \beta_i/2$ and $\beta_i = \alpha_i/2$.*

Proof. The existence is given by the algorithm below:

1. Do $i = 1$;
2. Build $S_{\alpha_i\beta_i}$ using proposition 9;
3. Do $i = i + 1$ and $\alpha_i = \alpha_i/2$;
4. Use proposition 9 to generate

$$S_{\alpha_i\beta_i} - S_{\alpha_{i-1}\beta_{i-1}} \subset (R - \bigcup_{s_j \in S_{\alpha_{i-1}\beta_{i-1}}} B_{\beta_i}(s_j));$$

5. Do $\alpha_{i+1} = \alpha_i/2$; $i = i + 1$;
6. Go back step 1 until $i = n$.

Correctness of the algorithm: According to proposition 9 (chapter 4) it follows that $\beta_i = \alpha_i/2$ for each i. □

5.2 Graded Family

A special case of monotone family takes place when the difference of successive levels differs by exactly one point.

Definition 29. *Let $\mathcal{F} = \{S_i\}$ be an increasing (decreasing) family. We say that $\mathcal{F}$ is graded if $S_{i+1} = S_i \cup \{s_i\}$ ($S_i = S_{i+1} \cup \{s_{i+1}\}$).*

Next we will study how to generate two classes of graded families. They are: graded α-families and graded (α, β)-families.

Graded α-Families

We propose two algorithms that generate a graded α-family. Algorithm **A** attempts to generate it with pre-determined sub-scales ($\alpha_1 > \alpha_2 > ... > \alpha_n$) whereas algorithm **B** attempts to generate it without requirements.

Algorithm A: The sub-scales $\{\alpha_i\}$ define a α sub-family that can be generated through the algorithm 1. Now the problem is: Is it possible to gradate such sub family? The answer is negative since we have a counter example in figure 5.2. In figure 5.2.a we have a coarser PDS level formed by the vertices of a square with side 1 so that $\frac{\sqrt{2}}{2} < \alpha < 1$. In 5.2.b we have a finer level where the points are over a spacing grid $\frac{1}{3}$ with a small perturbation. We point out the best candidate to the gradation. In figure 5.2.c we exhibit geometrically that it is impossible the new sampling satisfy the coverage condition since there exists multiple empty spaces and we are not able to cover then with only a single Poisson disk. Although it is not possible to gradate an α-family, we will show that it holds for the (α, β)-family.

Algorithm B: The algorithm generates a family by decreasing the scale parameter continuously in a given interval $[a, b]$. It begins with one PDS with scale radius b. Then, define $f : [a, b] \to \mathbb{R}$ such that $f(t) = \text{area}(R - \mathcal{B}_t)$ and $\mathcal{B}_t = \cup_{s \in S_{\alpha_t}} B_{\alpha_t}(s)$. We have $f(b) = 0$ until $t = t_0$ where the function f

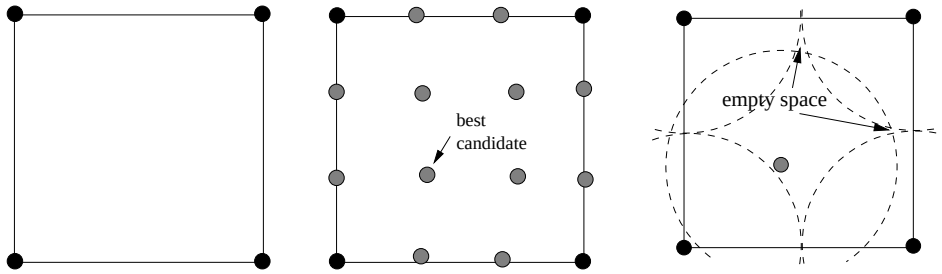


Fig. 5.2: Counter example for grading process.

assume increasing values. At time t_0, infinitesimal holes are opened and the coverage condition becomes invalid for disks of radius t_0. Hence we insert a new point p in each hole. We repeat the same process until we reach the scale a. This process is described in algorithm 2.

Algorithm 2: *graded_αβ_family*

Input : Solid R, $[a, b]$
Output: graded familly $\mathcal{F} = \{S_{\alpha_i}\}$

$S_{\alpha_0} = \emptyset$;
$t = b$;
$i = 1$;
while $t > a$ **do**
  if $f(t) > 0$ **then**
    for *each small hole* ρ **do**
      $\alpha_i = t$;
      $p_i = \mathsf{dart\_throw}(\rho)$;
      $S_{\alpha_i} = S_{\alpha_{i-1}} \cup p_i$;
      $i = i + 1$;
  decrease t;

Graded (α, β)-families

We exposed the difficulties of grading an α-family in algorithm **A**. Surprisingly, for an (α, β)-family it is possible due its generality.

Theorem 4. *Let $\mathcal{F} = \{S_{\alpha_i\beta_i}\}$ be an (α, β)-family of the solid R. Then there exists an (α, β)-family $\mathcal{F}_g$ that is a gradation of $\mathcal{F}$.*

Proof. We will prove only for an increasing family since the decreasing one can be transformed into an increasing just inverting its order. It is also sufficient to prove the gradation between two levels.

We give two algorithms. The first one gradates following the order given by the Housdorff distance. The second way gradates following the order given by the smallest edge of the Delaunay triangulation.

Let $D = S_{\alpha_{i+1}\beta_{i+1}} - S_{\alpha_i\beta_i} = \{s_1, s_2, ..., s_k\}$, then we generate an order $D_\sigma = \{s_{\sigma(1)}, s_{\sigma(2)}, ..., s_{\sigma(k)}\}$ such that there exists a unique $s_{\sigma(l)} \in D$ such that:

$$\max\{d(s, A_{l-1}) \quad | \quad s \in D_{l-1}\} = \beta_{i_l}$$

where, $D_l = D - \{s_{\sigma(1)}, s_{\sigma(2)}, .., s_{\sigma(l)}\}$ e $A_l = S_{\alpha_i\beta_i} \cup \{s_{\sigma(1)}, s_{\sigma(2)}, .., s_{\sigma(l)}\}$.

The value of α_{i_l} can be chosen as the lowest real positive that satisfies the covering condition:

$$\alpha_{i_l} = \inf\{\alpha \in \mathbb{R} \,|\, R \subset \bigcup_{s \in A_l} B_\alpha(s)\}.$$

Given $S_{\alpha_i,\beta_i} \cup \{\sigma_1, \sigma_2, ..., \sigma_k\} = S_{\alpha_{i+1},\beta_{i+1}}$, the grading algorithm of $\mathcal{F}$ is initialized from the Delaunay triangulation of $S_{\alpha_{i+1},\beta_{i+1}}$. The algorithm

orders the points from $S_{\alpha_{i+1},\beta_{i+1}}$ to S_{α_i,β_i} by removing points.

Algorithm 3: Gradation using DT

Input : $S_{\alpha_{i+1},\beta_{i+1}}$.

Output: Gradation of $S_{\alpha_{i+1},\beta_{i+1}} - S_{\alpha_i,\beta_i}$.

$A = \emptyset$;

init L_e;

while $A \neq (S_{\alpha_{i+1},\beta_{i+1}} - S_{\alpha_i,\beta_i})$ **do**
- $e = L_e.top$;
- **if** $lv(p) > lv(q)$ **then**
  - delete p from $DT(S_{\alpha_{i+1},\beta_{i+1}} - A)$;
  - $A = A \cup \{p\}$;
- **if** $lv(q) > lv(p)$ **then**
  - delete q from $DT(S_{\alpha_{i+1},\beta_{i+1}} - A)$;
  - $A = A \cup \{q\}$;
- **if** $lv(q) = lv(p)$ **then**
  - $e_p = \max\{m(w) \quad | \quad w \in link(p)\}$;
  - $e_q = \max\{m(w) \quad | \quad w \in link(q)\}$;
  - **if** $e_p > e_q$ **then**
    - delete q from $DT(S_{\alpha_{i+1},\beta_{i+1}} - A)$;
    - $A = A \cup \{q\}$;
  - **else**
    - delete p from $DT(S_{\alpha_{i+1},\beta_{i+1}} - A)$;
    - $A = A \cup \{p\}$;
- update L_e;

In the beginning we initialize the priority queue L_e that contains all edges of the Delaunay triangulation such that the top is the smallest one. Let $lv(w)$ be the level of the point w and link(w) be the set composed by

the neighbor edges to w in the mesh. Let e be the edge at the top of the priority queue L_e. The point to be removed is chosen between the adjacent vertices of e. Let p and q be such vertices. Then if $lv(p) > lv(q)$ we delete p as described in [13]. Analogously, if $lv(q) > lv(p)$ then we remove q. If $lv(p) = lv(q)$, we decided for removing the one that is nearest to the neighbor vertices, not considering edge e. In more detail, consider e_p and e_q edges such that $e_p = \max\{m(w)|w \in \text{link}(p)\}$ and $e_q = \max\{m(w)|w \in \text{link}(q)\}$, where $m(w)$ is the length of w. If $m(e_p) > m(e_q)$ then we remove q otherwise, removes p. The queue L_e is updated and the algorithm proceeds in the same way. The value of β_i corresponds to the length of the top edge in L_e and α_i to the any value covering the solid. See algorithm 3.

□

5.3 (α, β)-Family of Meshes

The (α, β)-*Family of Meshes* is directly associated to (α, β)-family. They are particular cases of a increasing (decreasing) family of meshes.

Definition 30. *Let $S_{\alpha\beta}$ be an (α, β)-PDS of a solid and $K_{\alpha\beta}$ a solid simplicial complex. If $K^0_{\alpha\beta} \subset S_{\alpha\beta}$ we say that $(S_{\alpha\beta}, K_{\alpha\beta})$ is an $\alpha\beta$-pair.*

Definition 31. *Let $\mathcal{F} = \{S_{\alpha_i\beta_i}\}$ be a monotone (α, β)-family. Then $\mathcal{M}(\mathcal{F}) = \{(S_{\alpha_i\beta_i}, K_{\alpha_i\beta_i})\}$ is an monotone (α, β)-Family of Meshes if $(S_{\alpha_i\beta_i}, K_{\alpha_i}\beta_i)$ is an $\alpha\beta$-pair, $\forall i$. For each i we denote scale level and for each β_i we denote scale.*

Example 1. *Let $F = \{S_{\alpha_i}\}$ be a α-family. We say that $C_\alpha\mathcal{F} = \{(S_{\alpha_i}, \overline{C_{\alpha_i}(S_{\alpha_i})})\}$ is a α-family of Solid Alpha Complexes.*

Example 2. *Let* $F = \{\tilde{S}_{\alpha_i\beta_i}\}$ *be a* (α, β)*-family. We say that* $C_{\alpha\beta}\mathcal{F} = \{(\tilde{S}_{\alpha_i\beta_i}, QC_{\alpha_i}(\tilde{S}_{\alpha_i\beta_i}))\}$ *is a* (α, β)*-family of Quasi-Solid Alpha Complexes.*

In examples above we have a much stronger structure: besides vertices in common, through properties there is also a subset of coincident faces between successive meshes. We will exploit this fact to construct filterings in next chapter.

6. FILTERING BY TOPOLOGICAL OPERATORS

This chapter describes how to use the topological operators to build or unbuild monotone family of meshes. First we analyze the intrinsic relationship between refinement (simplification) and topological junctions (disjunctions). Then we define *Filtering* as a sequence of topological junctions (disjunctions) and give algorithms to build them. The instances of the algorithms are the (α, β)-family of meshes studied last chapter.

6.1 Refinement (Simplification) with Topological Operators

In topology, a *refinement (or subdivision)* of a cover C of a topological space X is a new cover D of X such that every set in D is contained in some set in C. In the case of meshes we distinguish the boundary and the interior as topological spaces. The only topological operator that refines boundary is edge split on boundary edges and the topological operators that refines interior are edge split on interior edges and face split on triangles. The edge split can be substituted by an *atomic operation* composed by the other operators. In figure 6.1 we show examples of two atomic operations of refinement performed on the boundary.

Analogously, we can reason for simplification of topological spaces. The corresponding operators are face weld, edge weld on boundary edges and edge weld on interior edges. We can substitute edge weld by the other operators.

In the remaining of this thesis we opted to not use both edge split and edge weld. Therefore, we simplify the adaptation structure explained in next chapter.

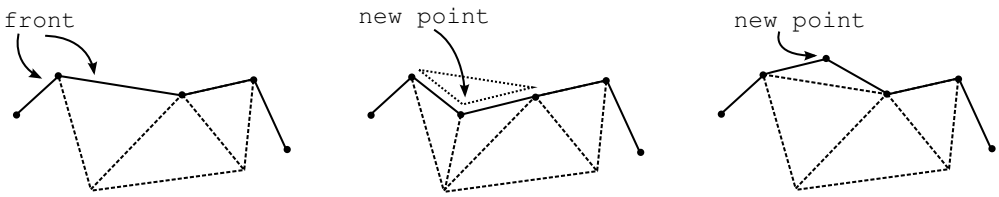


Fig. 6.1: . Boundary refinement. (Left) Original mesh; (Middle) Using split and destroy operators. (Right) Using create operator.

What does topological junction (disjunction) really mean?

For our purposes, a topological junction (disjunction) is the combination of two characteristics:

Topological: means refinement (simplification) of topological spaces. It can perform either in the interior of the mesh or in its boundary. It also means homology change. For example, the Solid Alpha Complex is a subset of Delaunay triangulations then all triangles have empty circumcircle and radius less then a given parameter α. In this case holes can open, close, merge or unmerge by the use of handle operators.

Geometrical: means addition of details. If the point falls at the boundary, we get a better approximation of the front and if it falls in the interior, we get a better resolution of the solid. Analogously a topological disjunction consists in loss of details. If the point is on the boundary, we have a lower approximation of the front and, if it is on the interior, the solid resolution is decreased.

6.2 Filtering

We are interested in a particular family of meshes: graded families such that successive elements are common except at small pieces of the mesh. The difference may correspond exactly to the application of a sequence of stellar and handle operators so that they compose a topological *junction* or a topological *disjunction*. To these operations incorporated to the graded family we call *Filtering*.

Definition 32. *A Filtering is an increasing (decreasing) family of meshes* $\mathcal{M}(\mathcal{F}) = \{(S_i, K_i)\}$ *such that* $\{S_i\}$ *is graded and* $(S_{i+1}, K_{i+1}) = (S_i, K_i) \oplus s_i$ $((S_i, K_i) \ominus s_i)$ *with* $s_i = S_{i+1} - S_i$ $(s_i = S_i - S_{i+1})$.

When the filtering is based on (α, β)-family of meshes we call (α, β)-filtering or simply α-filtering as $\alpha_i = \beta_i$. When the family is increasing then we call *filtering by refinement* and when it is decreasing we call *filtering by simplification*

The notion of filtering is constructive. Indeed, by definition, the successor element is obtained from the previous by a sequence of topological operators.

The main problem we study in this chapter is: *Given a monotone family of meshes* $\mathcal{M}(\mathcal{F}) = \{(S_i, K_i)\}$*, transform it into a filtering.* This problem is too general, however, we are able to solve it among four instances. Next we enumerate them:

I_1: Graded increasing α-family of Solid Alpha Complexes;

I_2: Graded decreasing α-family of Solid Alpha Complexes;

I_3: Increasing (α, β)-family of Quasi-Alpha Complexes;

I_4: Decreasing (α, β)-family of Quasi-Alpha Complexes.

In all instances we give *algorithmic proofs*. For each one we propose a different algorithm and we name as A_i the algorithm for instance I_i and its output as $A_i(I_i)$. Therefore $A_1(I_1)$ and $A_2(I_2)$ are α-filterings and $A_3(I_3)$ and $A_4(I_4)$ are (α, β)-filterings.

We classify A_1 and A_3 as *filtering algorithms by refinement* and A_2 and A_4 as *filtering algorithms by simplification*.

Symmetry

Let $\mathcal{M}(\mathcal{F})$ be an increasing (decreasing) family of meshes. We can reverse the order of $\mathcal{M}(\mathcal{F})$ so that we obtain a decreasing (increasing) family of meshes. Lets denote such family as $(\mathcal{M}(\mathcal{F}))^{-1}$.

The proposed algorithms A_1 and A_2 are strongly related by *symmetry*. They are constructed in such a way that $A_1(I_1) = A_2(I_1^{-1})$ and $A_2(I_2) = A_1(I_2^{-1})$. Analogously, A_3 and A_4 are symmetric.

6.3 Strategies and Algorithms

In all algorithms we maintain two meshes: M_f is the current mesh of the filtering whereas M is the Delaunay triangulation of the current points. Both meshes are related so that $M_f \subset M$. The main reason we use mesh M in the algorithms is that it guarantees the desired symmetry property.

We also have a list of operators L_f that store the output of the filtering. The operators of this list applied sequentially results in the current mesh M_f. This means that whenever L_f is updated then M_f is modified and vice-versa.

Flipping Order in M

The main idea of the filtering algorithms by refinement is to exploit the Delaunay triangulation construction algorithm based on insertion of points (see [23]). We use a priority queue of half-edges L_{he} that keeps the order in which they will be flipped. The order of L_{he} is given by the power (with inverted signal) of the new point to be inserted s_i with the circumcircle of he.mate.face, for all he $\in L_{he}$.

In the filtering algorithms by simplification we use the deletion of points for Delaunay triangulations (see [13]) and modify slightly the code to what we intend. Let s_i be the candidate point to be removed. The removal algorithm must take out all triangles incident to s_i and retriangulate in the "Delaunay sense" the star-shaped polygon $H = \{q_0, q_1, ..., q_{k-1}, q_k = q_0\}$ created by these removals. Three consecutive vertices $q_iq_{i+1}q_{i+2}$ along the boundary of H are said to form an *ear* if the segment q_iq_{i+2} lies in H. The algorithm uses the following lemma:

Lemma 2. *Consider polygon $H = \{q_0, q_1, ..., q_k = q_0\}$ and a point p such that the edges q_iq_{i+1} lies in the Delaunay triangulation of $\{q_0, q_1, ..., q_{k-1}, p\}$. If $|power(p, circle(q_i, q_{i+1}, q_{i+2}))|$ is minimal, then q_iq_{i+2} is in the Delaunay triangulation of $\{q_0, q_1, ..., q_{k-1}\}$.*

Proof. See Devillers [13]. □

For filtering algorithms by simplification we also have a priority queue L_{ear} of *ears* such that their elements have an augmented structure of type candidate_ear composed by three half-edges: s_iq_i, s_iq_{i+1}, s_iq_{i+2} (see below the definition of the structure that will be used in the algorithms). Then L_{ear} keeps the order that the edges s_iq_{i+1} will be flipped. The order is given

by the power of s_i (point to be removed) with the circumcircle of $q_i q_{i+1} q_{i+2}$. The lemma above assures that the top of the priority queue will contain an ear that belongs to the Delaunay triangulation.

```
struct candidate_ear {
  Hedge*     he1, he2, he; }
```

The basis of symmetry property comes from the order of queues L_{he} and L_{ear}. Since they are ordered by the power of the point to be inserted or removed, symmetry holds by construction.

Flip generation

To "generate" a topological operation X stands for "to insert operation X into L_f". We decide for generating a flip operation of an edge e according to pertinence of the adjacent faces to M_f. For instance, if both faces belong to M_f then we generate flip and if both faces do not belong to M_f we decide to not generate it. But what do we do in the other cases? We give a rule for filtering algorithms by refinement and then we deduct the symmetric rule for the filtering algorithms by simplification.

Given an instance $I = \mathcal{M}(\mathcal{F}) = (S_i, K_i)$, $i = 1, ..., n$, the rule for the filtering algorithms by refinement follows the pertinence of the *pivot triangle* to M_f. This triangle is opposite to the inserting point s_i (see on the left of figure 6.2). Then we generate a flip whenever the pivot triangle belongs to the current mesh.

Now lets infer the corresponding rule for filtering algorithms by simplification with instance I^{-1}. The order of $s'_j = s_i$ in the filtering algorithm by simplification is such that $s'_j = s_{n-i}$. As we know, the ear $t = q_k q_{k+1} q_{k+2}$

being Delaunay (i.e. the triangle has empty circumcircle) assures that it will belong to M. However it will belong to M_f if and only if it pass the validation test function. This test consists on verify if t satisfies the corresponding properties of the instance I^{-1} at the level $(j+1)$. It has two versions depending on the input: for instance of type I_2 it compares the circumradius of t with the scale α_{j+1}; whereas the test for instance of type I_4 is given by definition 19 in chapter 4. The function of this test is described in algorithms 4 and 5.

Algorithm 4: function validation_test_A

Input : (t, α)

if (circunradius of t is lower than α) **then**
  return true;
else
  return false;

Algorithm 5: function validation_test_B

Input : (t, α)

if (circunradius of t is lower than α and
vertices of t belong to boundary and
to same connected component) **then**
  return true;
else
  return false;

Then the corresponding pivot triangle of filtering algorithms by simplification is t (see on the right of figure 6.2).

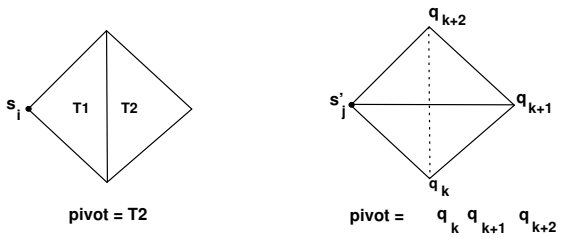


Fig. 6.2: Pivot of A_1 (LEFT) and pivot of A_2 (RIGHT).

After all flips in the Filtering Algorithms by Simplification

In the filtering algorithms by simplification we have two possible configurations in the mesh M after all possible flip operations in the adjacent edges of the candidate point s_i to be removed. In the first configuration there are exactly three adjacent triangles to s_i whereas in the second there is at least one adjacent triangle to s_i. In each configuration the point s_i is respectively interior and boundary. Figure 6.3 illustrates these examples. We will name them as configurations C1 and C2, respectively.

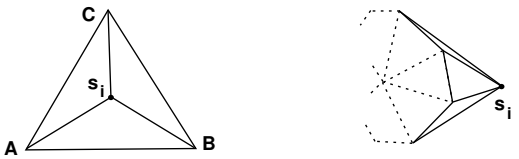


Fig. 6.3: Configurations after all flips in A_2. (Left) C1, (Right) C2.

Let T be the set of adjacent triangles to s_i in any configuration. In the configuration C2 all triangles in T will be removed from M. Given $t \in T$ we must generate the operation destroy whenever $t \in M_f$. After all triangles been removed from M we generate the operation remove(s_i). In configuration C1 there are two sub cases depending on the validation test of triangle ABC. This test corresponds to the function validation_test_X,

where X is either A or B, depending on the input of the algorithm. If the triangle do not pass to the test it means that there is a hole in the mesh K_{i+1} where ABC is placed. We name this sub case as C1.1. Therefore we must also generate the operation destroy for each $t \in T$ such that $t \in M_f$. After weld operation in M on point s_i we generate remove(s_i). In case it pass to the test, this means that ABC is entirely covered by K_{j+1}. We name this sub case as C1.2. Then we must generate the operation create for each $t \in T$ such that $t \notin M_f$. After weld operation in M on point s_i we generate weld(s_i). The algorithm 6 describes this process.

Algorithm 6: function after_all_flips

```
if (s_i ∈ ∂M) then
    for (each t ∈ T) do
        if (t ∈ M_f) then
            L_f.insert(destroy);
    remove(s_i);
    L_f.insert(remove(s_i));
else
    T = initialize (A, B, C);
    if validation_test_X(ABC, α_{i+1}) then
        for (each t ∈ T) do
            if (t ∈ M_f) then
                L_f.insert(destroy);
        weld(s_i);
        L_f.insert(remove(s_i));
    else
        for (each t ∈ T) do
            if (t ∉ M_f) then
                L_f.insert(create);
        weld(s_i);
        L_f.insert(weld(s_i));
```

The order that we sweep the triangles in set T is decreasing in the size of their circumradius. The glue operations are immediately generated after each face creation whereas unglue operations are immediately generated before each face destruction. These operations follows the decreasing order of the edges length.

Before all flips in the Filtering Algorithms by Refinement

The symmetric steps of algorithm 6 in the filtering algorithms by refinement holds before all flips. We construct this part of the algorithm and show its symmetry to the last one in all cases described.

Since in the last step of algorithm by simplification we remove the point s_i from the mesh M, the symmetric operation corresponds to its insertion in M. There are two cases depending on the place where s_i falls. If it falls outside M we generate the operation $\mathsf{add}(s_i)$. This case is symmetric C2. Then we create the set T of all visible faces from s_i to the mesh M. After that we must prepare the triangles in T according to the pivot triangle. For each $t \in T$ we generate the operation create if and only if its pivot triangle belongs to the mesh M_f. If it falls inside a face ABC in M then a face split is performed and a new set T of triangles is created in M. This case is symmetric to C1. The set T has exactly three triangles. Now we have two sub cases depending on the triagle ABC pertinence in M_f. This sub case si symmetric to C1.1. If ABC belongs to M_f we generate the operation split and we prepare the set T according to their pivot triangles. For each triangle in the set T we generate the operation destroy if and only if its pivot triangle do not belong to M_f. If ABC do not belong to M_f then we generate

the operation create for each triangle in T if and only if its pivot triangle belongs to the mesh M_f. This sub case is symmetric to C1.2 and algorithm 7 describes this process.

Algorithm 7: function before_all_flips

```
if (s_i falls outside M) then
    for (each visible edge e_jk
    in ∂M from s_i) do
        create(s_i, s_j, s_k) ;
        if (pivot triangle of s_i s_j s_k is in M_f) then
            L_f.insert(create);
else
    T=split(ABC);
    if (ABC ∉ M_f) then
        for (each t ∈ T) do
            if (pivot triangle of ABC
            belongs to M_f) then
                L_f.insert(create);
    else
        L_f.insert(split(s_i));
        for (each t ∈ T) do
            if (pivot triangle of ABC
            do not belong to M_f) then
                L_f.insert(destroy);
```

The order that we sweep the triangles in set T is increasing in the size of their circumradius. The glue operations are immediately generated after each face creation whereas unglue operations are immediately generated before each face destruction. These operations follows the increasing order of the edges length. Notice that this canonic order is also symmetric to the order of algorithm 6.

After each flip in Filtering Algorithms by Refinement

Let s_i be the inserted point as in last section before all flips. After each flip performed in mesh M in the filtering algorithms by refinement, we analyze the new triangles T_1 and T_2. Let $t \in \{T_1, T_2\}$, we first verify the power of the point s_i with its the pivot triangle. If it is lower than zero we enqueue the half-edge of t opposite to s_i in L_{he} and prepare it to the next flip. Then we generate the operation create (destroy) if and only if $t \notin M_f$ ($t \in M_f$) and its pivot triangle belongs (do not belong) to M_f. If the power the pivot triangle of t is greater than zero we just verify the pertinence of the triangle to the complex K. Therefore we generate the operation create (destroy) if and only if the triangle $t \in K_i$. This test consists exactly to validation_test_X (see algorithms 4 and 5). The algorithm 8 describes totally this process. We sweep the set $\{T_1, T_2\}$ following the counter clockwise order.

Algorithm 8: function after_each_flip

```
Input : T_1, T_2, s_i
for (t ∈ {T_1, T_2}) do
    if (power(t, s_i) < 0) then
        enqueue in L_he the half-edge in t opposite to s_i ;
        if (pivot triangle of t belongs to M_f and
        t ∉ M_f) then
            L_f.insert(create(t));
        if (pivot triangle of t do not belong to M_f and
        t ∈ M_f) then
            L_f.insert(destroy(t));
    else
        if (validation_test_X(t, α_i)) then
            L_f.insert(create(t));
        else
            L_f.insert(destroy(t));
```

Before each flip in Filtering Algorithms by Simplification

Let s_i be the candidate point to be removed. Before each flip performed in mesh M in the filtering algorithms by simplification, we analyze the candidate ear $q_k q_{k+1} q_{k+2}$ in the top of the heap L_{ear}. Let $\{T_1, T_2\}$ be the adjacent faces of the edge to be flipped. We verify the pertinence of the ear $q_k q_{k+1} q_{k+2}$ in the complex K_{i+1} performing the test of the pivot triangle as the algorithm validation_test_X. In case it pass to the test then for each $t \in \{T_1, T_2\}$ we generate an operation create if and only if $t \notin M_f$. In case it do not pass to the test we generate the operation destroy if and only if $t \in M_f$. The algorithm 9 describes this process. We sweep the set $\{T_1, T_2\}$ following the clockwise order. Notice that this algorithm is symmetric to the last one.

Algorithm 9: function before_each_flip

```
Input : T_1, T_2, s_i
if (validation_test_X(q_k q_{k+1} q_{k+2}, α_{i+1})) then
    for (t ∈ {T_1, T_2}) do
        if (t ∉ M_f) then
            L_f.insert(create(t));
else
    for (t ∈ {T_1, T_2}) do
        if (t ∈ M_f) then
            L_f.insert(destroy(t));
```

After Insertion and Before Removal

After the insertion of a point s_i and all flips have been performed in mesh M we have to update M globally. This means that we must verify all triangles outside the link of s_i and verify if they belong to the complex K_{i+1} by the test function validation_test_X. The order we sweep the faces is increasing in the size of their cincumcircles.

Before the removal of a point s_i in mesh M we also update M_f globally by following the last algorithm in a symmetrical way. Then we verify all triangles outside the link of s_i by performing the test of the function validation_test_X. The order we sweep the faces is decreasing in the size of their cincumcircles.

Surprisingly both process above have the same algorithms. The algorithm 10 summarizes them.

Algorithm 10: functions after_insertion and before_insertion

```
Input  : s_i, α
for (each triangle t in M − link(s_i)) do
    if (validation_test_X(t, α) and t ∉ M_f) then
        L_f.insert(create(t));
    if (not validation_test_X(t, α) and t ∈ M_f) then
        L_f.insert(destroy(t));
```

Filtering Algorithms: joining pieces

We recall the symmetry correspondence between algorithms described in this chapter until now:

- The order of flip operations is between the filtering algorithm by refinement and the filtering algorithm by simplification are symmetric;

- after_all_flips is symmetric to before_all_flips;
- after_each_flip is symmetric to before_each_flip;
- after_removal is symmetric to before_removal;

Next we describe the filtering algorithms for each point insertion or removal. The symmetry holds by the correspondence established above.

Algorithm 11: Filtering algorithm by refinement: inserting a point

```
before_all_flips(s_i, α_{i+1});
initialize(L_he);
while (L_he ≠ ∅) do
    e = L_he.top;
    if (T_1 ∈ M_f and T_2 ∈ M_f) then
        L_f.insert(flip(e));
    flip(e);
    after_each_flip(T_1, T_2);
    update L_he;
after_insertion(s_i,α_i);
```

Algorithm 12: Filtering algorithm by simplification: removing a point

```
before_removal(s_i, α_{i+1});
initialize(L_ear);
while (L_ear ≠ ∅) do
    q_k q_{k+1} q_{k+2} = L_ear.top;
    {T_1, T_2} = adjacent faces of s_i q_{k+1};
    before_each_flip(T_1, T_2);
    if (T_1 ∈ M_f and T_2 ∈ M_f) then
        L_f.insert(flip(s_i q_{k+1}));
    flip(e);
    update L_ear;
```

Next we establish the theorem based on the results we obtained.

Theorem 5. *Let $\mathcal{M}(\mathcal{F})$ be an increasing graded α-family of solid alpha complexes (or a quasi solid alpha complex). Then there exist a filtering algorithm by refinement A_r and a filtering algorithm by simplification A_s such that $A_f(\mathcal{M}(\mathcal{F})) = A_r(\mathcal{M}(\mathcal{F})^{-1})$.*

6.4 Generalizations

The filtering algorithms may be easily extended for non decreasing monotonic functions $g : \mathbb{R} \to \mathbb{R}$ on the radius of the samplings graded family at each level. The only thing we need to do is change the validation_test_X function to compare the radius of the circumcircles of the triangles according to $g(\alpha)$. This allows us to make a **homology control in the filtering**, i.e., the function g controls all homology changes between the filtering levels. Then we obtain have the following theorem for α-families (for (α, β)-families is analogous):

Corollary 1. *Let $C_{g(\alpha)}(\mathcal{F})$ be an increasing graded α-family of solid alpha complexes and $g : \mathbb{R} \to \mathbb{R}$ be a non decreasing monotonous function. Then there exist a filtering algorithm by refinement A_r and a filtering algorithm by simplification A_s such that $A_f(C_{g(\alpha)}(\mathcal{F})) = A_r(C_{g(\alpha)}(\mathcal{F})^{-1})$.*

Notice that the Delaunay triangulation is equivalent to the corollary above considering the $g(x) = \infty$, $\forall x$.

We can also generalize for non graded families. However the only possible instances are the $(\alpha, \beta)-$family of quasi alpha complexes.

Theorem 6. *Let $C_{\alpha\beta}(\mathcal{F}) = \{(\tilde{S}_{\alpha_i\beta_i}, QC_{\alpha_i}(\tilde{S}_{\alpha_i\beta_i}))\}$ be an increasing (α, β)-family of meshes. For any non decreasing monotonous function $g : \mathbb{R} \to \mathbb{R}$ there exist a filtering algorithm by refinement A_r and a filtering algorithm by simplification A_s such that $A_f(C_{g(\alpha)\beta}(\mathcal{F})) = A_r(C_{g(\alpha)\beta}(\mathcal{F})^{-1})$.*

Proof. The proof is the composition of two algorithms. First we apply the algorithm of theorem 4 to grade the intermediate levels of $\mathcal{F}$ in a new graded family $\mathcal{F}_c$. Then, we apply the algorithm of theorem 1 to $\mathcal{F}_c$. □

On the left of figure 6.4 we have four levels of a non-graded (α, β)-family of meshes. On the right we have four sub levels, after gradation, between the third and the fourth levels.

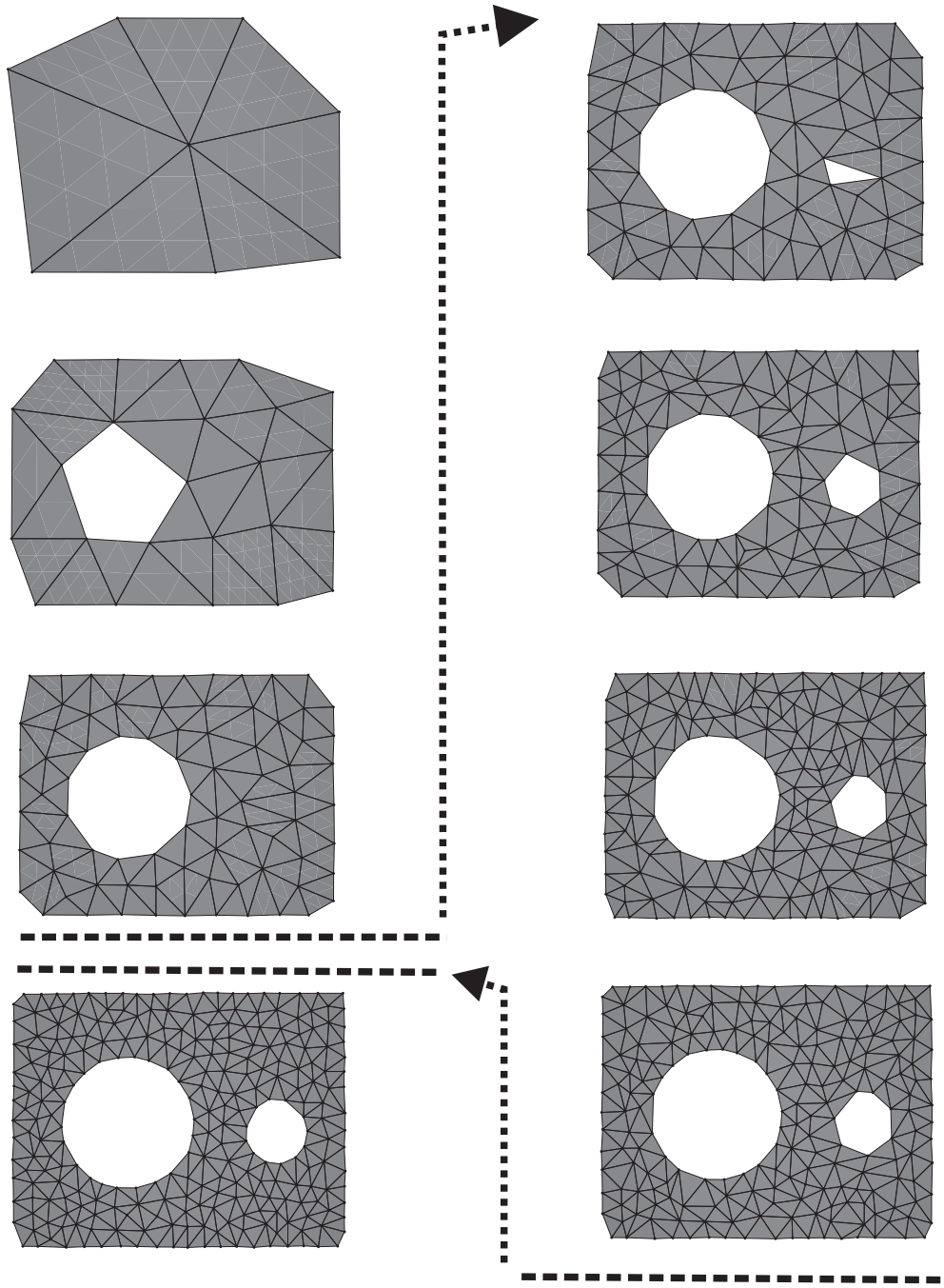


Fig. 6.4: Four levels (LEFT) and four sublevels (RIGHT).

7. FILTERED MULTITRIANGULATION

In this chapter we propose a multitriangulation data structure with topology changes. Our framework is obtained by abstracting on specific construction rules (local modifications), while considering only the intrinsic relationships among basic elements.

So far, we have a sequence of topological operators with symmetry and uniqueness properties. The sequence is generated by a filtering algorithm. Now, we make the following question: what can we do with such filtering? Before we answer this question, let us explore new sequences of operations at three stages: pre-filtering, at filtering time and post-filtering.

Pre-filtering: We recall that given a quasi scaling-family $QC_{g(\alpha)}\mathcal{F}$ the filtering is deterministic. However, we can generate new filterings at sampling time. Indeed, we have a new sampling whenever the seed sampling changes and, consequently, we have a new filtering.

At filtering time: We can obtain new sequences by varying the function g in the filtering algorithm which changes the α-complex parameter at each point insertion/removal. This function is responsible by all topological changes at each filtering step.

Post-filtering: We can create new sequences by means of permutations

forming a compatible sequence of meshes. However to perform such permutations we must know when two successive operators can be commuted. When they commute we say that they are **independent**, otherwise they are **dependent**. Independency comes from the local property of the topological operation, i.e., it changes only a small piece of the region considered. On the other hand dependency takes place where the operations are too close. Take as example split ∘ create(p0, p1, p2). There is no way the operator split commute with create because of the dependency between arguments. In this case, the face must pre-exists to the split operator act.

Goals

The answer to the question posed before is: adaptation. It can be obtained simply by truncating possible permutations. To form them, any model may focus on structuring dependency and independency relations between *modifications*. In this chapter we propose a model which comprises **stellar** and **handle** codifications. Then, we can perform *queries*, i.e., the extraction of geometric and topological details by means of an *adaptation function*.

The approach has a potentially strong adaptation mechanism as we will demonstrate in next chapter. Additionally, to the best of our knowledge, *multitriangulations* with topological changes support is at the cutting edge of geometric modeling and any similar work have not yet been published.

The data structure representation we design has the following desired properties:

- It represents the filtering globally;
- It represents all interdependencies keeping topological consistence;

- It allows permutations of topological operations, i.e., adaptability;
- It represents all possible [1] meshes.

The graph structure is the most suitable for these requirements. We follow formalizations of Velho and Gomes [41], Puppo [37] to represent the data structure.

7.1 Topological Multi-Triangulations

7.1.1 Sequential Structures

A Mesh Sequence, H, is a sequence of meshes $H = \{K_j\}_{j=1,2,\ldots,n-1}$ with dependency relations at two subsequent levels j and $j+1$. Such relations can be between faces or boundary edges with overlapping support.

A mesh sequence is constructed by *local modifications* from some initial mesh. They are either *combinatorial* or *constructive* modifications.

(I.) Combinatorial

(a) Edge flip: The edge e and its adjacent faces f_l and f_r are replaced by two new adjacent faces f_u and f_d and another edge linking the new faces.

(b) Face split: The face f is replaced by three new vertex v, new faces f_1, f_2 and f_3 and three new edges e_1, e_2 and e_3.

(c) Face weld: The face f replaces three faces f_1, f_2 and f_3 and three edges e_1, e_2 and e_3.

(II.) Constructive

[1] All possible subsets of the sequence of operations with considering permutations and dependency relations.

(a) Edge glue: Two boundary edges are replaced by one interior edge.

(b) Edge unglue: One interior edge is replaced by two boundary edges.

(c) Face create: One isolated face is added to the mesh.

(d) Face destroy: One isolated face is removed from the mesh.

Given a local modification $\mathcal{M}(K_i)$, its *pre-image* are those simplices that will be replaced and its *image* are the replaced and added simplices in K.

Formally, a mesh sequence, $H = (K_0, K_1, ..., K_n)$, is generated by the application of a sequence of local modifications $(\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_{n-1})$ starting with the mesh K_0. This produces the sequence $(K_0, \mathcal{M}_1(K_1), ..., \mathcal{M}_{n-1}(\mathcal{K}_{n-1}))$ where:

$$K_j = \mathcal{M}_{j-1}(\mathcal{M}_{j-2}(...\mathcal{M}_1(K_1))), \text{ for } j > 0.$$

The purpose of a multi-triangulation with topology changes (TMT) is to encode all possible mesh sequences that can be generated from a sequence of local modifications $(\mathcal{M}_i)_{i=1,...,n-1}$. In order to achieve this goal, we need to distinguish between dependent and independent modifications.

A TMT, $V = (K_0, \mathcal{M}, \prec)$ is defined by an initial mesh K_0, a set of compatible local modifications $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_{n-1}\}$, and a partial order relation $\prec$ on $\mathcal{W}$, satisfying the following conditions:

i Dependency: $\mathcal{M}_i \prec \mathcal{M}_j$, iff there is a simplex σ in the pre-image of $\mathcal{M}_i$ that belongs to the image $\mathcal{M}_j(K_j)$ of $\mathcal{M}_j$.

ii Non-redundancy: $\sigma \in \mathcal{M}_i(K_i)$ implies that $\sigma \notin \mathcal{M}_j(K_j)$ for all $j \neq i$.

7.1.2 Representation

Posets can be described by a direct acyclic graph (DAG), where the nodes are associated with modifications $\mathcal{M}_i$, and there is an arc from $\mathcal{M}_i$ to $\mathcal{M}_j$ whenever $\mathcal{M}_j$ is the successor of $\mathcal{M}_i$ according to the partial order $\prec$.

The TMT is complemented by adding a source and a drain which give us a *lattice*. A source node is associated with a constructor of the initial mesh K_0, and the drain node is associated with the application of all modifications $\mathcal{M}_i$, $i = 1, ..., n-1$, to K_0, that produces the final mesh K_n. Appropriate arcs are added to and from these special nodes.

The node having a simplex σ in its pre-image is called *successor* of σ, and the node having σ in its image is called *predecessor* of σ.

A *cut* of a DAG consists of a set of nodes disconnecting it. A *front* in a lattice is a cut which contains exactly one arc for each path from the source to the drain. Figure 7.1 illustrates the Lattice Representation of a *TMT* with a front.

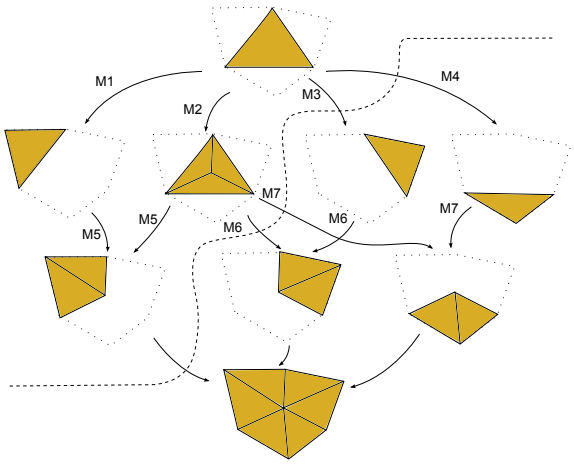


Fig. 7.1: Lattice representation of a TMT.

7.1.3 Data Structure

Basically, we have a half-edge data structure, just as described in the topological operators [**?**], except by the pointers to encode the partial order.

The flip modification is easily encoded in the edge with a pointer to the flipped edge. It has pointers (2) to the adjacent faces and pointers (2) to the adjacent faces of the flipped edge. (see figure 7.2.a).

The face split and face weld modifications are encoded in the vertex with a pointer to each child face (3) and its parent face. (figure 7.2.b).

Since the create operator is immediately followed by at most three glue operator we group implicitly the edge glue to the face create modification. Analogously we group the edge unglue to the face destroy modification. The create, destroy, glue and unglue modifications are encoded in the edge with pointer: to parents (2) and children (2) edges. (figure 7.2.c).

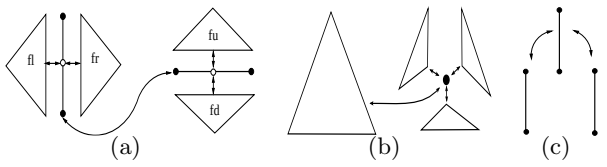


Fig. 7.2: Diagram representations of the data structure.

The specification of these structures in pseudo-C is given in figure 7.3. To simplify the algorithms described in this chapter each structure is labeled by the modification type used to include it in the mesh[2].

Each modification has a representative simplex σ in the pre-image such that it is capable to recover the successor of σ and all elements in the pre-

[2] The modification type can also be identified by the non null attributes.

```
Struct Face {          Struct Edge {          Struct Vertex {
Vertex *vparent;       Edge *eparent[2];      Face *fparent;
Vertex *vchildren;     Edge *echildren[2];    Face *fchildren[3];}
Edge *eparent;         Face *fparent[2];      int m; }
Edge *echildren;       Face *fchildren[2];
int m; }               int m; }
```

Fig. 7.3: Pseudo-C code of data structures.

image. For example, the representative simplex of the edge flip is just the edge to be flipped. In the face weld, the vertex is the most representative. The create operator, is represented by a boundary edge. Face split, and create are represented by a face. Such simplices will be useful to the algorithms described later.

7.1.4 Construction

There are two basic techniques for construction of such models: simplification methods start from the full resolution and progressively reduce the number of vertices on which the model is based, in order to coarsen the resolution; refinement methods start from coarse approximation, and progressively refine it by inserting new vertices, in order to improve resolution.

7.1.5 Compatible Subsequences

The TMT is able to obtain different triangulations not only by truncating the mesh sequence H at some time, but also by considering any subsequence that is compatible with the partial order. In figure 7.4 we show examples of meshes obtained from the TMT represented in figure 7.1. Notice that each mesh is generated with a different sequence of compatible modifications.

Such subsequences can be obtained as cuts in the lattice representation of the TMT. The following proposition shows how fronts are related to the

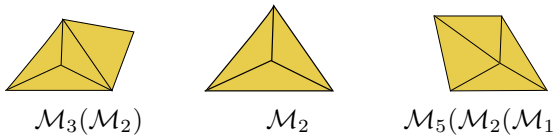


Fig. 7.4: Three out of possible 33 combinations that can be extracted from TMT of figure 7.1.

level of detail extracted from the lattice.

Proposition 13. *([37]) Let $\mathcal{T}$ be a TMT, and let $\mathcal{T}_\mathcal{E}$ be the sequence of $\mathcal{T}$ formed by the nodes in the upper part of a given front $\mathcal{E}$ (with the natural order). Then $\mathcal{T}_\mathcal{E}$ is a compatible sequence.*

The triangulation on the right side of figure 7.1 corresponds exactly to front of the TMT. The previous theorem says that performing a sequence of modifications is equivalent to sweeping a front downwards through the lattice. This process is initiated with a front that lies immediately below the source and moving it below a node when we perform the corresponding modification. Notice that the fronts define all possible compatible sequences in a TMT, i.e., the number of different triangulations that we can build from compatible sequences of local modifications is equal to the number of different fronts.

Let $U_\mathcal{E}$ be the upper region of a front $\mathcal{E}$ in a lattice $\mathcal{T}$. We wish advance $\mathcal{E}$ below a node (modification) $\mathcal{M}$. This means that there are arcs (dependecies) cutting the front $\mathcal{E}$ and going to $\mathcal{M}$. In order to advance the front we must include recursively in $U_\mathcal{E}$ all nodes going directly to the node $\mathcal{M}$.

The recursive function advance_node in algorithm 13 describes this process. It receives as input the representative simplex of the corresponding

topological operator. The modification_type is a function that receives a simplex σ and returns the modification and its representative simplex. Depending on the dimension of σ it has three versions. They are described in algorithms 15, 16, 17. The modify performs the corresponding topological operator modification.

In the algorithm 16 conventions are adopted in lines 2, 4 and 10. Lines 2 and 4 returns the adjacent face with greater natural order[3]. Analogously, line 10 returns the adjacent vertex with greater natural oder.

Algorithm 13: advance_node

Input : representative simplex σ

```
1 for each τ ∈ image(successor(σ)) do
2     for each ρ ∈ pre-image(predecessor(τ)) do
3         if ρ ∉ mesh then
4             advance_node(ρ.representative_simplex);
5 modify(modification_type(σ));
```

[3] For each simplex is assigned a natural order given by the filtering sequence.

Algorithm 14: modify

Input : $\mathcal{M}$, σ

```
1  if M == CREATE then
2      create(σ);
3      for each e_ij ∈ σ do
4          if e_ji ∈ M then
5              glue(e_ij, e_ji);
6  if M == DESTROY then
7      for each e_ij ∈ σ do
8          if e_ij ∉ ∂M then
9              unglue(e_ij, e_ji);
10     destroy(σ);
11 if M == FLIP then
12     flip(σ);
13 if M == SPLIT then
14     split(σ);
15 if M == WELD then
16     weld(σ);
```

Algorithm 15: modification_type_v

Input : Vertex v

Output: $\mathcal{M}$, σ

```
1  if v.m == SPLIT then
2      return (SPLIT, v.fparent);
3  if v.m == CREATE then
4      return (CREATE, v.fparent);
```

Algorithm 16: modification_type_e

Input : Edge e
Output: $\mathcal{M}$, σ

```
1  if e.m == GLUE then
2      return (CREATE, e.eparent[0].face);
3  if e.m == UNGLUE then
4      return (DESTROY, e.eparent[0].face);
5  if e.m == CREATE then
6      return (CREATE, e.face);
7  if e.m == FLIP then
8      return (FLIP, e.eparent[0]);
9  if e.m == SPLIT then
10     return (SPLIT, e.org);
```

Algorithm 17: modification_type_f

Input : Face f
Output: $\mathcal{M}$, σ

```
1  if f.m == SPLIT then
2      return (SPLIT, f.vparent.fparent);
3  if f.m == WELD then
4      return (WELD, f.vparent);
5  if f.m == FLIP then
6      return (FLIP, f.eparent.eparent[0]);
7  if f.m == CREATE then
8      return (CREATE, f);
```

7.2 Analisys

In this section we analyze three criterias, as discussed in Puppo [37], that express the effectiveness of a DAG in geometric modeling applications. They are *Growth Rate*, *Expressive Power* and *Depth*. Particularly, we are interested in investigate them in (α, β)-Filtered-TMTs.

7.2.1 Growth Rate

We consider a special class of TMTs, corresponding to either refinement or simplification sequences. In the following proposition, we will refer to the *size* of a set of simplices as the the number of triangles.

Definition 33. *A TMT is increasing (decreasing) if and only if the size of each image is larger (smaller) than the size of its pre-image.*

It is sufficient to study monotonicity in the case of increasing TMTs, while a decreasing TMT can be dealt with by working on its reverse.

We speak generically of *monotone* TMT, meaning either increasing or decreasing TMT.

According to Puppo [37], the growth ratio of a TMT is the ratio between the size of the sequence of modifications and the size o its cumulative application to a mesh. When this rate is bounded by a constant, the growth is liner. This result provides a sufficient (but not necessary) condition to linear growth that is easy to verify in practical cases. Below we rewrite the proposition in our context:

Proposition 14. *Let $\mathcal{T}$ be an increasing TMT. If the size of each image is larger then the size of its pre-image for at least a constant factor, then $\mathcal{T}$ has linear growth.*

Linear growth is a desirable property since it allows to achieve optimal output time complexity in visiting the structure.

We wish to analyse linear growth in (α, β)-Filtered TMTs, however, there are two drawbacks corresponding to the existence of non monotonic modifications. The first one is the destroy modification, which is decreasing, and the the other one is the flip modification, which is non increasing.

We will first analyze a subclass of the (α, β)-Filtered TMTs that do not contain such modifications. As we know, whenever α is sufficiently high, the (α, β)-Filtered TMT is the lattice corresponding to the Delaunay triangulation by either inserting or removing points at each step. Let us call its particular lattice as D-Filtered TMT. Notice that it does not contain the destroy modification, but, it may contain flip modifications. We will consider an *induced* TMT that is monotone (i.e. without flip or destroy modifications) that can be extracted from a D-Filtered TMT.

Induced Lattice

The D-Filtered TMT has an induced lattice where the nodes are atomic operations corresponding to the sequence of topological operations at each point insertion of the Delaunay triangulation. For instance, in the figure 7.1, the destination nodes of the arcs $M2$, $M5$, $M6$ and $M7$ can be replaced by just one node. More precisely, it can be defined by the following way:

Definition 34. *Let $\mathcal{T}$ be a D-Filtered TMT. The induced TMT $\mathcal{T}^*$ is a DAG where each node is the cumulative application of all local modifications derived from a unique point insertion and two nodes have an arc if and only if their associated subgraphs in $\mathcal{T}$ are dependent.*

Notice that the set of nodes of $\mathcal{T}$ labeled with each associated node of $\mathcal{T}^*$

comprises a *partition*. Moreover, this induced lattice has relevant property: it is *strictly monotonic*. It can be shown that the growth rate g of the induced D-Filtered TMT is bounded by $g = \frac{n+2}{n+1}$ which corresponds to the attachment of a triangle to the mesh. To complete our investigation we give the following proposition:

Proposition 15. *Let $\mathcal{T}$ be a D-Filtered TMT lattice with a bounded number of flips at each point insertion. Let $\mathcal{T}^*$ be its induced monotone lattice. Then if $\mathcal{T}^*$ has linear growth then $\mathcal{T}$ has also linear growth.*

Proof.

Firstly, let us define a *fragment* T as the set of triangles which is the images of a given local modification $\mathcal{M}$ and $|T|$ as its size, i.e., number of triangles.

Let $T = (T_1, ..., T_k)$ be a compatible sequence of fragments of $\mathcal{T}$ and $T^* = (T_1^*, ..., T_l^*)$ be its corresponding compatible sequence of fragments of the induced lattice $\mathcal{T}^*$. Let $\oplus T_i$ and $\oplus T_i^*$ be the resulting mesh of the applications of the sequence of modifications from T and T^*, respectively. Notice that $|\oplus T| = |\oplus T^*|$ but $\oplus T \neq \oplus T^*$. Indeed $\oplus T$ and $\oplus T^*$ differs only by a set of flip operations.

Let f be the bound of the number of flips of the subgraphs. Then, for each T_i^* we have that

$$\sum_{T_j \in T_i^*} |T_j| < 2f + |T_i^*|$$

and

$$\frac{|T|}{|\oplus T|} = \frac{|T|}{|\oplus T^*|} = \frac{\sum_{i=1}^{i=l} \sum_{T_j \in T_i^*} |T_j|}{|\oplus T^*|} < \frac{\sum_i (2f + |T_i^*|)}{|\oplus T^*|} = \frac{2f}{|\oplus T^*|} + \frac{|T^*|}{|\oplus T^*|}$$

Let $|\oplus T^*| = n$ and g be the growth rate of the lattice $\mathcal{T}^*$. Then,

$$\frac{|T|}{|\oplus T|} < \frac{2f + ng}{n}$$

and, as a consequence, $\mathcal{T}$ has a linear growth.

□

In a graded (α, β)-Family $\mathcal{F}$, the points are uniformly spread over the solid object at each point insertion. Then, it is reasonably to assume that its D-Filtered TMT has a bounded number of flips. As a consequence, by the proposition 15, all D-Filtered TMT of the family $\mathcal{F}$ has linear growth.

Going back to the general (α, β)-Filtered TMT, we recall that they contain the destroy modifications which are decreasing. Therefore, we cannot claim about linear growth using the proposition above. However, we know that it is a subset of the composition of a sequence of modifications of the Delaunay triangulation. Then, based on this underlying triangulation, we can argue about the linear growth of a general Filtered TMT using again the proposition above.

7.2.2 Depth

The depth of a lattice corresponds to the number of levels of the longest path from source to drain in the lattice representation. This property is relevant to structure traversal operations such as point location.

We can estimate easily the depth of a lattice generated by subdivision process such as the regular $\sqrt{3}$ Subdivision, Kobbelt [22], and the semi-regular $4-8$ Subdivision, Velho and Zorin [42], since their subdivision process are recursive. Unfortunately, the general TMTs does not have such recursiveness and, therefore, its depth becomes difficult to compute. However,

we are able to estimate it based on the the characteristics of the (α, β)-family of points and on the properties of the filtering.

The (α, β)-family we will analyze is the one proposed in the proposition 12, chapter 5. As we know, the resolution rate between successive levels is 2. Then, at each level, the corresponding size of the Delaunay triangulation also grows exponentially. The amount of topological operations necessary to add or remove each point is bounded by a constant, since the points are uniformly distributed in the plane and the triangulation is Delaunay. By the same reason, the depth between each level is bounded by a constant k. Therefore if an (α, β)-family $\mathcal{F}$ has q levels then the depth of its corresponding TMT is bounded by kq.

7.2.3 Expressive Power

The expressive power of α-Filtered TMT is the number of different triangulations that one can build from compatible sequences of modification. It is equal to to the number of different fronts. This property is relevant to the adaptiveness of the mesh. The (α, β)-Filtered TMT being based on a large number of small fragments, it has high expressive powerPuppo [37].

8. APPLICATIONS

In this chapter we consider the applications of (α, β)-Filtered TMT's for managing level of detail (LOD) of solids. We define relevant operations and give some examples. The main concepts of LOD operations follow De Floriani et al. [10].

8.1 Filtering, Refinement and Resolution

The filtering by refinement could refer to the increase of the resolution of the mesh (number of points per unit length). Recall that in the filtering algorithm by refinement as α-parameter decreases at the same time that a new point is inserted to the mesh, a collection of modifications is globally performed to compose a new approximating complex of the solid. Therefore, the global resolution is strongly influenced by the α-parameter, i.e.,the edges and triangles have circumradius less than α. Consequently the corresponding filtered TMT is composed by simplexes in multiple resolutions and LOD operations can be applied.

8.2 Extracting Meshes at Variable Resolution

Assume that a boolean condition $c : \mathcal{T} \to \{0, 1\}$ is defined on the simplexes of a filtered TMT $\mathcal{T}$, such that for a given simplexes σ, $c(\sigma)$ is true if and only if the resolution of σ is acceptable. Based on this function we consider

the following *selective problem*:

Problem: Find the smallest (i.e. the one with fewest number of triangles) triangulation T made of simplexes of $\mathcal{T}$ such that $c(\sigma) = 1$, $\forall \sigma \in \mathcal{T}$.

The search process of this problem is a *level of detail (or query) operation* and its resulting mesh is a *triangulation at variable resolution*.

In order to exploit as much as possible the capability of the Filtered TMS's, we propose a framework that formalizes the boolean function $c()$. Then we give an algorithm to solve the posed problem above.

Framework

In addition to other frameworks for LOD extraction we propose a new topological parameter over the focus set which comprises the homology group (boundary curves). The boolean function $c()$ can be specified through the following three parameters:

- A *adaptation function* $\tau : \mathbb{R}^2 \to \mathbb{R}$ to bound mesh approximation at each point in space (e.g. a density function);
- A *focus set* $F_i \subset M_i$ representing the sub complex of interest in the query (e.g. boundary, interior, simplexes intersected by a line, point or region etc.).
- A *criteria function* $cr : F_i \to \mathbb{R}$ that gives the approximation error of the focus set F_i (e.g. edge length, triangle area, radius of the circumcircle, size of the boundary etc).

It follows that the parameters above give us the behavior of the function $c()$, for a simplex σ:

$$c(\sigma) = \begin{cases} 0, & \text{if } \sigma \in F_i \text{ and } cr(\sigma) < \frac{\int_\sigma \tau(p)dA}{A(\sigma)} \\ 1, & \text{otherwise} \end{cases}$$

We call σ a *valid* simplex if and only if $c(\sigma) = 1$; we call *active* if and only if it intersects the focus set F_i.

Depending on the focus set F_i we distinguish three types of queries: geometric, topological and improvement. They can also be combined to form new queries. As follows, we characterize each one:

Geometric: The focus set is given by a subspace $A \subset R^2$. At each adaptation step only simplexes which lies over A are considered.

Topological: The focus set is composed by topological elements of M_i. For instance, it can be the homology group (boundary curves) or interior vertices.

Improvement: The focus set is composed by triangles with high aspect ratio. The purpose of this query is to improve the quality of adapted mesh extracted.

The query is a refinement when the source of the filtered TMT is a coarse mesh and the drain an fine mesh. Analogously, the query is a simplification if the source is a fine mesh and the drain is a coarse mesh.

We remark that the mesh of a general query could be either a mesh representing the whole surface, or a sub-mesh containing just the elements

inside the region of interest. In first case, the query is called *globally defined* and in the second one, *locally defined*.

The globally defined result of the query is given simply by suitable cut of the Filtered TMT. The result of a local query can be regarded as a cut in a clipped DAG, obtained from the DAG describing the TMT by deleting all non-active simplexes, and all arcs labeled by non-active simplexes only.

In this thesis we address globally defined LOD queries. We left locally defined as a future work.

Algorithm

The algorithms for solving the global queries on an Filtered TMT combine a DAG traversal with local interference: the DAG traversal addresses the LOD issue and is parametric over the given *adaptation function* and the *geometric criteria*; the interference test addresses the *geometric*, *topological* and *improvement* issues and are parametric over the given *focus set*.

Our algorithm is based on an breadth-first traversal of the DAG: starting from the source, it progressively visits all arcs and nodes above the solution. A priority queue $\mathcal{Q}$ of the nodes that must be visited and the current mesh M is maintained. Each element of the queue is the representative simplex of the corresponding modification. For instance, in create, destroy and split the representatives are faces[1]; in flip the representative is an edge and in weld is a vertex. This gives an equivalence relation which generates a partition of the mesh at a given time of the algorithm. In other words, the queue $\mathcal{Q}$ compose a covering of the mesh M at each iteration (see figure 8.1).

[1] Actually the create operator has one of its three adjacent edges to the current mesh as representative since it does not belong to the current mesh.

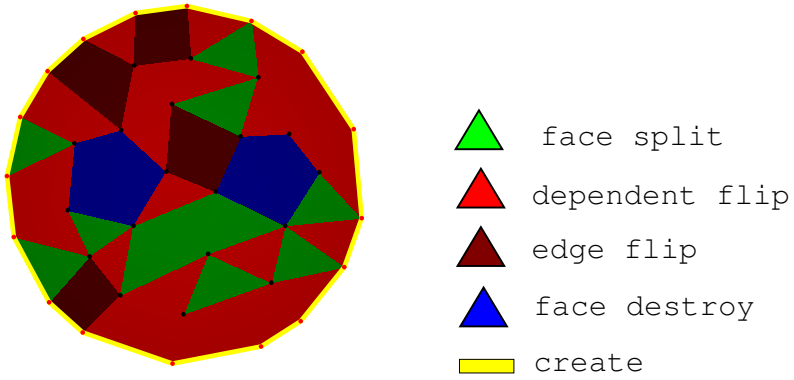


Fig. 8.1: Partition representation of the current mesh in $\mathcal{Q}$. The query is a refinement.

The key of the elements in the queue $\mathcal{Q}$ is the geometric criteria $cr()$. The algorithm consists of a main loop that iterates until the queue is empty. A node is added to the queue if either one of the nodes (pre-images) of the arcs (modification) entering it is in the current solution, but does not satisfy the required accuracy or one of the nodes below it must be added to the queue (dependent node). When a node is visited, all simplexes are added to the current mesh. On exit of the loop the mesh M will contain all simplexes of the solution.

The current solution is initialized with the source of the Filtered MTM $\mathcal{T}$. For each simplex σ that does not satisfy the boolean function $c()$, the representative simplex τ of the modification corresponding to arc emanating from the node containing σ is added to the queue. At each iteration the simplex σ_i which is the first element of the queue is advanced below destination node of modification $\sigma_i.m$. This function was demonstrated in the algorithm 13, chapter 7, and we call it inside the query algorithm 18.

Algorithm 18: query

```
Input : T, c()
1 M = source(T);
2 while Q ≠ ∅ do
3     σ = Q.pop;
4     advance_node(σ, σ.m);
5     /*update Q*/ ;
6     for each τ ∈ image(σ.m) do
7         if τ is representative and is not valid then
8             Q.insert(τ, cr(τ));
```

8.3 Examples

In this section we show examples of queries. Firstly, we describe implementation details of the *adaptation functions* and the *criteria functions*.

Adaptation Functions: In our implementation the adaptation functions are represented by images in the gray scale ranging from 0 to 255. Most of the test images are the *uniform* functions at a given gray level, and the *ramp* functions in a predefined direction.

Criteria Functions: We have implemented two generic *criteria functions* τ_1 and τ_2 which are based geometrically on the area and on the circumradius, respectively. However, these functions are not well defined for some representative simplexes. For example, the edge e of a flip operation does not have area. In this case we have decided to use the average of the areas of the adjacent faces of the edge e. In tables 8.1 and 8.2, we describe for each criteria function the corresponding value of the representative simplex of the modification. When the focus set is composed by the boundary curves

(homology group) we have decided to use another criteria function τ_3 that returns the area of a single hole.

Modification Type	**Representative Simplex**	**Value**
split	face f	Area of f.
flip	edge e	Average area of the adjacent faces that belong to the pre-image of modification and to the current mesh.
weld	vertex v	Ditto.
destroy	face f	Area of f.
create	half edge he	Area of the equilateral triangle with side length he.

Tab. 8.1: The criteria function τ_1.

Modification Type	**Representative Simplex**	**Value**
split	face f	Circumradius of f.
flip	edge e	In the refinement query we take the highest circumradius of the adjacent faces that belong to the pre-image of modification and to the current mesh. In simplification we take the smallest one.
weld	vertex v	Ditto.
destroy	face f	Circumradius of f.
create	half edge he	Area of he.

Tab. 8.2: The criteria function τ_1.

Next, for each example, we will give the parameters of the selective function $c()$, the input mesh and show the resulting queries. According to the *focus set* we can classify the examples as geometrical, topological and improvement.

8.3.1 Geometric

Example 1: We explore **uniform** adaptation functions on a solid region without boundaries in its interior. The type of the queries is by refinement and, the base mesh, is only one triangle. The *criteria function* is given by the circumradius of the simplexes (function τ_2) and we show three queries at different gray levels (see figure 8.2.).

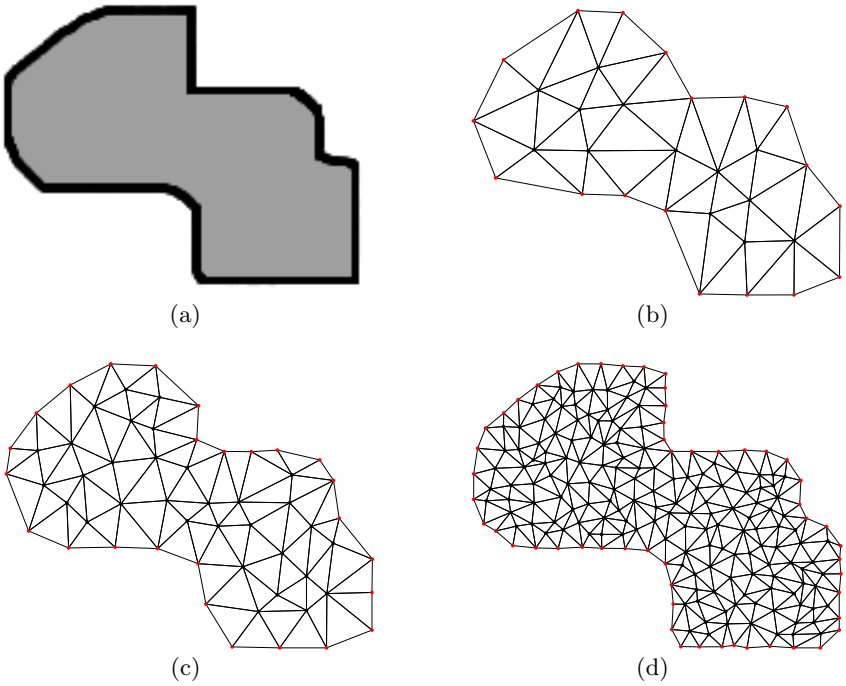


Fig. 8.2: (a) Original Solid. (b), (c) and (d) queries at different gray levels.

Comments on example 1: This is the simplest query we can perform and the meshes are uniformly distributed, as we expected. Similarly to filterings, we can make a question about the symmetry of queries. More specifically, given an uniform adaptation function at a given gray level x

and given a criteria function, according to the query types by simplification and by refinement, are such queries similar meshes? We concluded that they are slightly different and the reason for it is that, in the end of the query by refinement, for any simplex σ evaluated by selective function $c()$, we have that $c(\sigma) < x$, whereas, in the end of the query by simplification, we have that $c(\sigma) > x$. In summary, the queries by refinement tends to be more refined than the queries by simplification. In figure 8.3 we illustrate such difference.

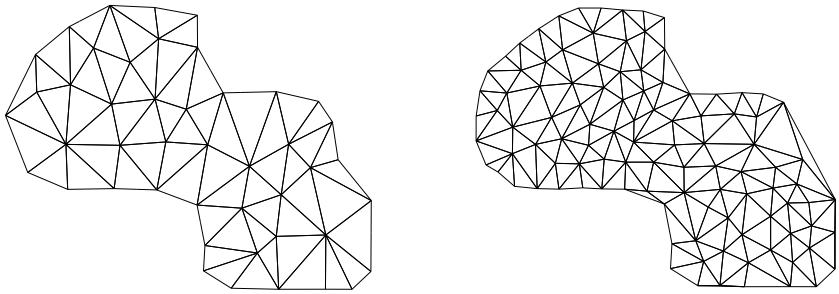


Fig. 8.3: Comparison of simplification and refinement using uniform distribution and the area as criteria function (a) Simplification. (b) Refinement.

Example 2: In this second example we explore **ramp** adaptation functions in a solid region with five holes where the type of the queries is by simplification. Moreover, the criteria function is given by the area of the simplexes (function τ_1). We show three queries in figure 8.4 where the ramp function in 8.4.b is directed horizontally, increasing smoothly from left to right and, the function in 8.4.c is oriented diagonally, increasing smoothly from top to bottom. In the figure 8.4.d the adaptation function has only two constant levels (255 and 0) such that the separation line of them is directed diagonally.

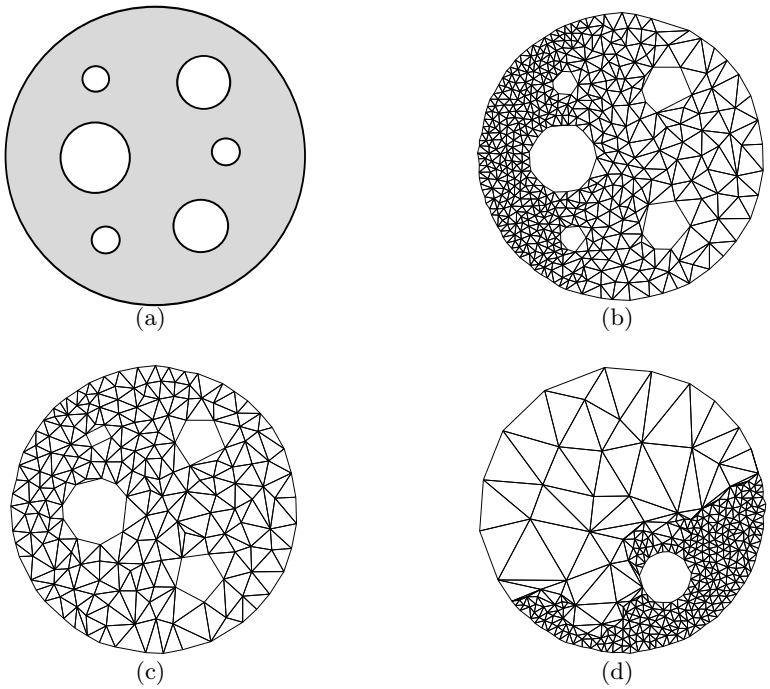


Fig. 8.4: (a) Original solid. (b),(c) and (d) queries at different ramp funcions.

Comments on example 2: The meshes in figures 8.4.a and 8.4.c behave as we expected. Notice that the smallest circle holes in the original solid, exist according to the resolution of mesh in the corresponding region of the hole. In 8.4.d, the part of mesh in the vicinity of transition region of the adaptation function is very instable since the transition between the gray levels is abrupt and, consequently, the triangles size do not variate smoothly. Moreover, surprisingly, there are big triangles flooding the part of mesh that would be supposedly more refined. The dependent modifications are the main reason for it.

8.3.2 Topological

Example 1: In this first example the focus set is the boundary elements of the solid (see figure 8.5.). The queries are performed by refinement and we use the circumradius as criteria function (τ_2) and the uniform distribution as adaptation function.

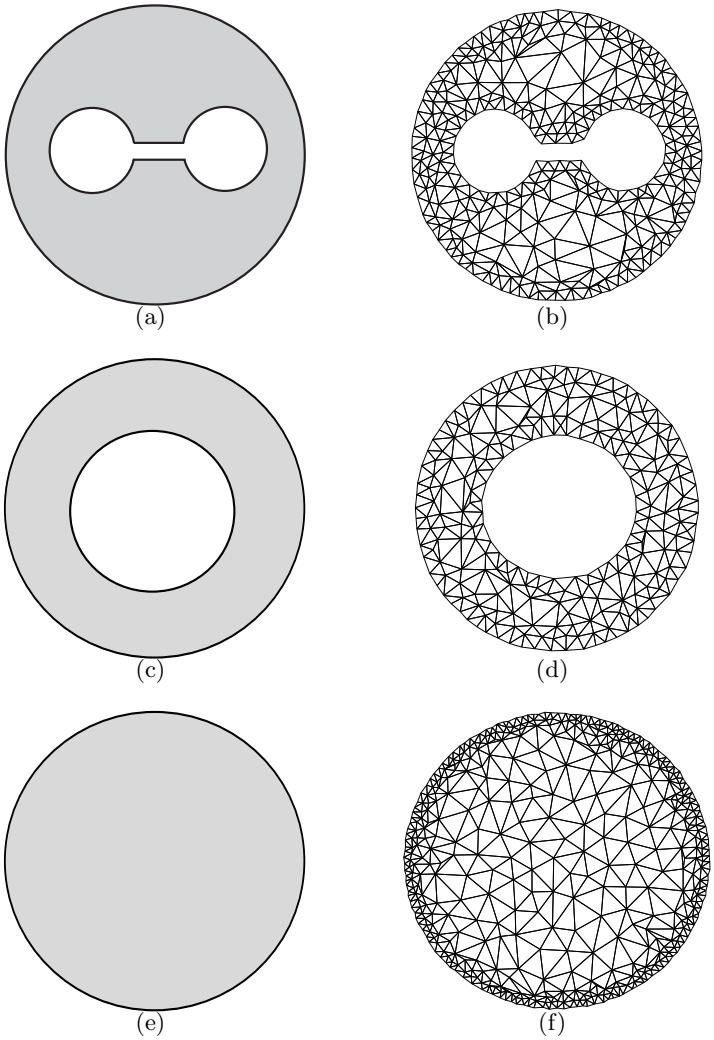


Fig. 8.5: (a), (c), (e) Characteristic function. (b), (d) and (d) query results.

Comments on example 1: We have been constructed the examples in two stages. The first one adapts the mesh uniformly until a predefined resolution and, except the solid 8.5.e which does not have holes, the meshes are refined uniformly until a hole been opened. In the second stage we apply the topological query, i.e., refinements in the boundary elements. As we expected, the boundaries in the final mesh are more refined than in the interior.

Example 2: In this example the *focus set* is the homology of the solid, i.e., each hole represents a unique element (see figure 8.6.). The queries are performed by simplification, the criteria function is the area of the holes (τ_3) and the adaptation function is the uniform distribution.

Comments on example 2: Since we evaluate each hole globally (the area of the hole is a global property), this example is slightly different from the last one where each boundary element is evaluated individually. The simplification is performed on the simplicial elements that have at least one subsimplex intersecting the hole. The solid in figure 8.6.a has four holes with areas 313.15, 314.61, 422.05 and 545.32. In the mesh 8.6.c we have a query which closes holes with area less than 320.0, i.e., only two of them. In figure 8.6.b the limit area is 430.0, as a consequence, three holes are closed. In the solid of figure 8.6.b, we have three holes having different sizes. Following the same scheme of the previous example, we obtain two meshes adapted topologically: the one in figure 8.6.d has two holes closed whereas 8.6.d has

all the holes closed. The largest hole[2] has the resolution[3] smaller than the others, then, to close it, we need large triangles. Consequently, the adapted mesh has a discontinuity in the boundary of the holes, i.e., in its vicinity, there are large triangles closer to small triangles.

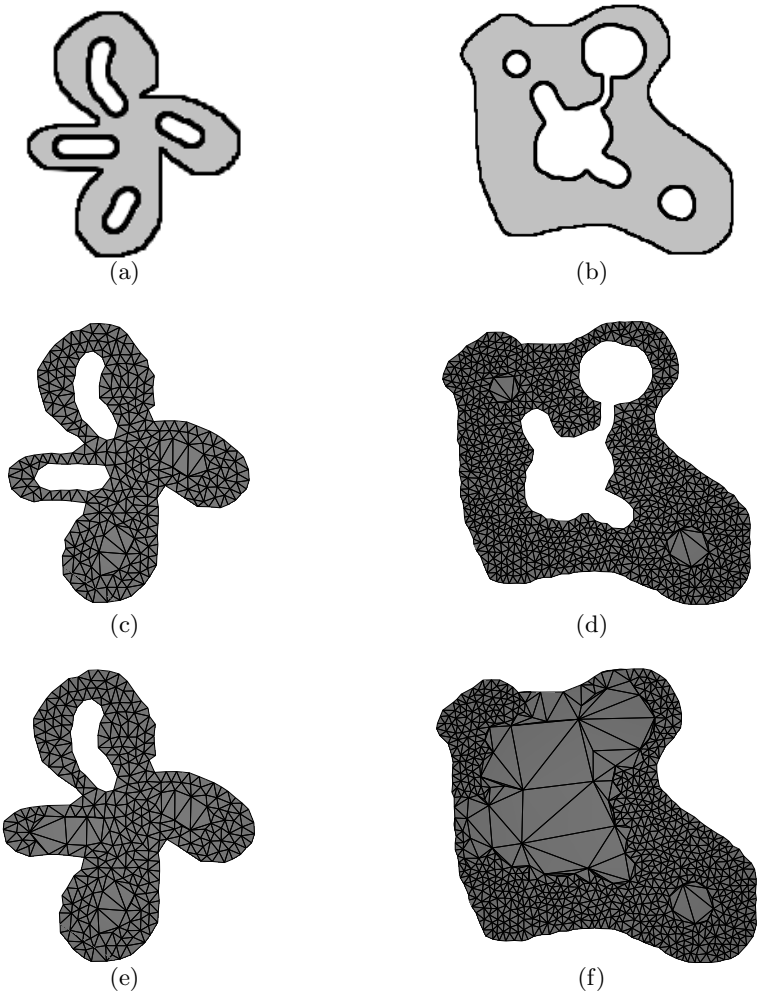


Fig. 8.6: (a), (b) characteristic functions. (c), (d) closing two holes. (e), (f) closing three holes.

[2] The most precise notion of the size of is the *local feature size*.
[3] The level that it appears in the first time of filtering.

8.3.3 Improvement

Example: The improvement is applied after a geometrical or topological query. The elements of the focus set are composed by the triangles having high aspect ratio that belong to the pre-image of a flip operation. In this case, both triangles of the pre-image must belong to the current mesh. The adapted meshes we show are based on the **uniform** functions. In the first example (figure 8.7) we set up the circumradius as criteria function whereas in the second one (figure 8.8) we use the area. The flips are *conservative*, which means that we always go down in the DAG. The regions in dark red refers to triangles that can be directly flipped.

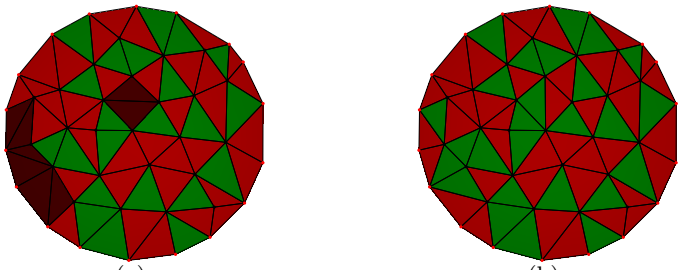


Fig. 8.7: (a) uniform query using circumradius. (b) mesh improved.

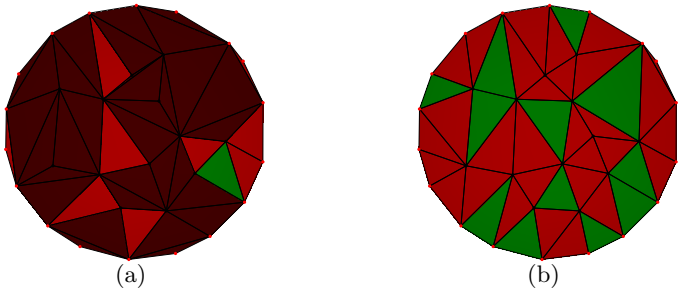


Fig. 8.8: (a) uniform query using area. (b) mesh improved.

Comments on example: Although this improvement query be quite simple and having good results, it does not take into account triangles with small aspect ratio that are pre-images of a flip. Other disadvantage is that a flip operation does not always mean improvement. In the simplification, for example, the flips are performed to remove a point through a weld operation. Such point has valence three and, as we know, it tends to have bad shaped triangles in its vicinity.

9. CONCLUSION

In this chapter we present the conclusion of the thesis. We give show main contributions and propose future works where limitations are discussed and generalizations are exposed as well.

9.1 Main Contributions

We introduced a framework for multiresolution of two dimensional solids with topology change support. Below we point out the main contributions:

- The analysis of geometry and topology of interpolated Poisson Disks Sampling (and variants) by means of *S*olid Alpha Complexes (and variants);
- A combinatorial data structure which supports *stellar* and *handle* operations. This structure gives the basis for building filterings by topological operators;
- Algorithms for construction of filterings. The instances are families of interpolations of Poisson Disk Samplings (and variants) and the outputs have symmetry properties;
- A data structure for multitriangulation of solids with topology support. The corresponding DAG is builded from filterings;

- Examples of applications for solid modeling that are queries performed on multitriangulations and have three qualitative behaviors, depending on the focus set parameter (geometrical, topological and improvement). Such examples do not focus on a specific scientific area but illustrate how the results can be applied in different domains. For instance, we point out the field of *Computational Structural Biology* which explores the structural properties of molecules using combinatorial and numerical algorithms on computers. Understanding topologies of proteins through homology is a common process. Then, modeling proteins using our framework may be helpful for proteins identification.

We also obtained some relevant partial results, as described in the appendixes, that contributed qualitatively to our work:

- An upper bound for the aspect ratio of solid α-complexes triangulations of Poisson Disk Samplings.
- A sampling condition of the Poisson Disk sampling that guarantees the topological equivalence of the solid with its reconstructed solid α-complex.
- The Restricited BPA, a new 2D triangulation method based on the ball-pivoting algorithm (BPA) which is based. This algorithm achieves optimal performance when the points are uniformly distributed.

9.2 Future Works

The directions for future works are improvement of examples illustrated in chapter 9 and generalizations.

9.2.1 Improving Results

Back to the Aspect Ratio

In a finite element analysis (FEM), it is desirable to avoid elements with high aspect ratio. Such elements can influence analysis results, and lead to misleading and inaccurate results, which are dependent on the mesh. Although the query examples illustrated in chapter 8, section 8.3.3, have presented some satisfactory results, the approach has problems. It arises from the restriction of hierarchy since we perform only conservative flip operations to improve the mesh quality. An idea to get better examples is to create an strategy that perform a combination of the all topological operations *dynamically*, i.e., going either up or down in the DAG instead of being conservative.

Gradation Control

In FEM applications it is also very important to control the mesh gradation smoothness. The example illustrated in figure 8.2.d has a serious gradation problem. As discussed before, its cause stems from the discontinuity of the adaptation function. Another example where gradation fails is illustrated in figure 8.6 where the coarser triangles covering the hole are neighbors of refined triangles on the boundary.

We intend to implement a control that alleviates such gradation problems and the level restrictions between neighbors triangles is commonly used in multiresolution representations. However, we have a trade-off: if the level difference between two triangles is at most one, we can obtain a very smooth gradation and, consequently, we lose the adaptation function properties.

9.2.2 Generalizations

n-Dimensional Solids

Many generalizations problems to n-dimensional spaces in geometric modeling are not so trivial. I think that the best strategy to generalize the problems is following linearly the order of this thesis. Most of the mathematical background such as, the stellar and the handlebody theory for n-dimensional manifolds, are already developed in the literature. The software implementations are more difficult since both data structures and algorithms increase enormously in complexity, and mistakes may occur frequently.

As predicted in appendix A, a partial result is not possible to generalize. In this case, the *slivers* are tetrahedrons with high aspect ratio that may appear in an interpolation of a Poisson disk sampling, however, we can go deeper making some investigations. For example, we can make statistical analysis on incidence frequency of slivers compared with the Poisson parameters or we can create a Poisson disk sampling algorithm capable to avoid them.

Hierarchy of homology

Hierarchy of homology is the correspondence of holes between successive resolution levels in a monotone family of meshes. Since the holes may split or merge forming new holes, the hierarchy (DAG) arises from these events (see for instance figure 9.1). However, some formulations are not so clear. For example, what is the best way to represent the homology set? As we know, the hole changes itself at each handle modification by adding or removing triangles lying on its boundary. Therefore it cannot have a unique representative simplex during the filtering process. As a consequence, the

construction of the corresponding DAG becomes difficult.

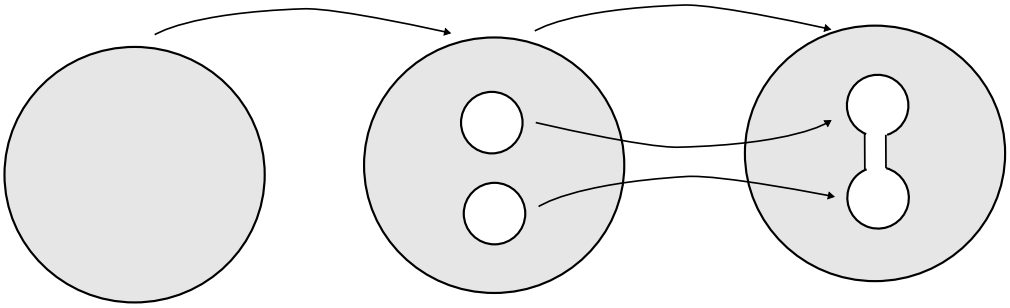


Fig. 9.1: An example of homology DAG.

APPENDIX

A. AN ASPECT RATIO UPPER BOUND

To study the quality of the mesh generated by computing the solid α-complex of a Poisson disc sampling, we adopt a triangle shape metric [17]. A popular one for a triangle τ is its aspect ratio $\frac{L^2}{\text{Area}(\tau)}$ [7], where L is the longest edge of τ.

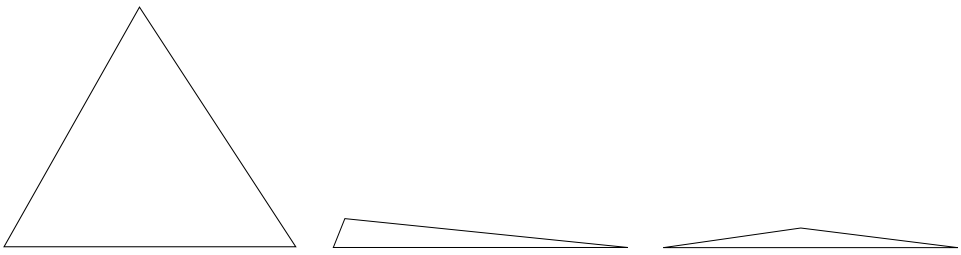


Fig. A.1: Triangle aspect-ratio: well-shaped (left) and degenerate triangles (middle and right).

The aspect ratio of an equilateral triangle is $\frac{4}{3}\sqrt{3}$ and that is the minimal one. This corresponds to the intuition that the equilateral triangle is the best shaped one. Then, the lower the aspect ratio is, the closer to an equilateral triangle it is (cf. Figure A.1).

A.1 Intuitive bound

Triangles with big aspect ratio are degenerated, and they contributes to numerical instabilities in simulations. In the center of Figure A.1, the triangle is degenerated since two of its vertices are too close. This case cannot occur

in an solid α-complex of a PDS because of the Poisson condition.

Therefore in our context, the only possibility for degenerate triangle is the one at the right of Figure A.1. The thinner this triangle is, the higher its circumradius. Since the parameter α of the PDS also serves for the solid α-complex, its construction removes such triangles with big circumradius from the Delaunay triangulation. It is therefore expected that the solid α-complex of a PDS has an upper bound for the aspect ration of its triangles. It is also intuitive from previous results that well spaced points imply bounded aspect ratio [38, 44].

A.2 Main theorem

In this section, we will show the main result of the paper, i.e. an aspect ratio upper bound in an solid α-complex of a PDS. This bound is $4\sqrt{3}$, which is only three times the aspect ratio of an equilateral triangle. Moreover, we prove that this bound is tight.

Theorem 7. *Let P_α be a PDS sampling of a region R. The aspect ratio of the triangles of $\overline{C_\alpha(P_\alpha)}$ is bounded by $4\sqrt{3}$.*

Proof. Let τ be a triangle and $b \leq a \leq L$ its three edge lengths. Let $S(\tau)$ be the area of the triangle τ and R its circumradius. Using, $S(\tau) = \frac{abL}{4R}$, we get aspect-ratio$(\tau) = \frac{L^2}{S(\tau)} = \frac{4RL}{ab}$. We will look for the worst case, i.e. the triangle that have the biggest possible aspect ratio, with the restriction that $L > a > b \geq \alpha$ (PDS condition) and $R \leq \alpha$ (solid α-complex condition).

Worst case: $b = \alpha$

We first check that the smallest edge b may actually reach the smallest

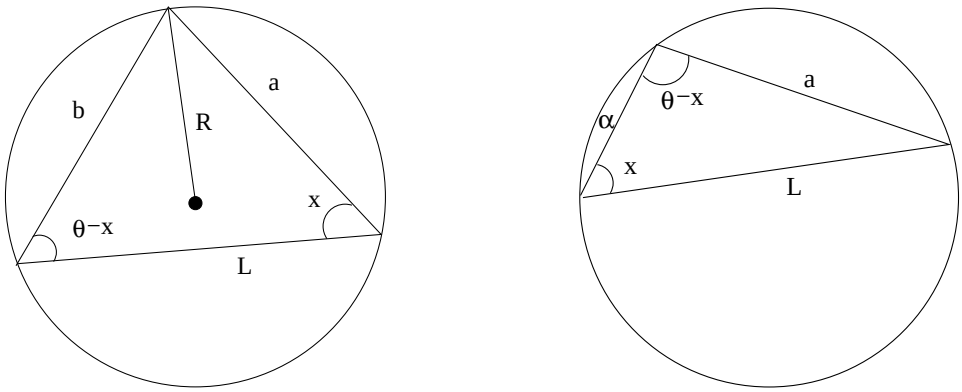


Fig. A.2: Minimizing ab with R and L fixed (left) and maximizing $\frac{L}{a}$ with R and $b = \alpha$ fixed (right).

distance between points, i. e., α. Fix R and L, the worst aspect ratio will be obtained minimizing ab. We have, with the notation of Figure A.2(left):

$$\frac{a}{\sin(\theta - x)} = \frac{b}{\sin x} = \frac{L}{\sin(\pi - \theta)} = k,$$

where k is a constant fixed by R and L. This implies that:

$$ab = k^2 \sin(x) \sin(\theta - x).$$

Differentiating $f(x) = \sin(x)\sin(\theta - x)$ we obtain:

$$f'(x) = \cos(x)\sin(\theta - x) - \cos(\theta - x)\sin(x) = \sin(\theta - 2x).$$

Since b is the smallest edge, $\sin(\theta - x) \geq \sin(x)$, with $\frac{\pi}{2} \geq \theta - x \geq x$, and thus $\sin(\theta - 2x) \geq 0$. We conclude that f is a non decreasing function, and the minimum for the product ab then occurs at the minimal value for b. However, the PDS condition ensures that $\alpha \leq b$. Then, the lowest value

of ab occurs when $b = \alpha$.

Worst case: $a = b = \alpha$

Now lets check that a can also reach α, the minimal value for ab being α^2. The worst aspect ration will occur when the ratio $\frac{L}{a}$ is maximal. Fix R and $b = \alpha$, we have, with the notation of Figure A.2(right):

$$\frac{L}{a} = \frac{\sin(\theta - x)}{\sin x} = \sin\theta \cot x - \cos\theta.$$

Since $0 \leq x \leq \frac{\pi}{2}$ and since the cotangent function is decreasing, the maximal value of the ratio occurs when a is minimal, i.e. $a = \alpha$ for a PDS.

Worst case: $a = b = R = \alpha$

Now lets derive the main result. To get the worst aspect ratio we maximize RL, fixing $a = b = \alpha$. We have, with the notation of Figure A.2(left):

$$\cos\theta = \frac{L}{2\alpha} \quad , \quad \sin\theta = \frac{\alpha}{2R}. \text{ then } \frac{L^2}{4\alpha^2} + \frac{\alpha^2}{4R^2} = 1.$$

We get:

$$(4\alpha^2 - L^2) \cdot R^2 = \alpha^4.$$

Since RL is maximal when R is maximal, i.e. $R = \alpha$, since $R \leq \alpha$ from the solid α-complex condition. The above relation then defines $L = \alpha\sqrt{3}$. □

The maximal aspect ratio is $4\sqrt{3}$ which is the configuration of the triangle of the Figure A.2(left). In this case the greatest angle is $\frac{3\pi}{2}$. This proves that the upper bound is tight.

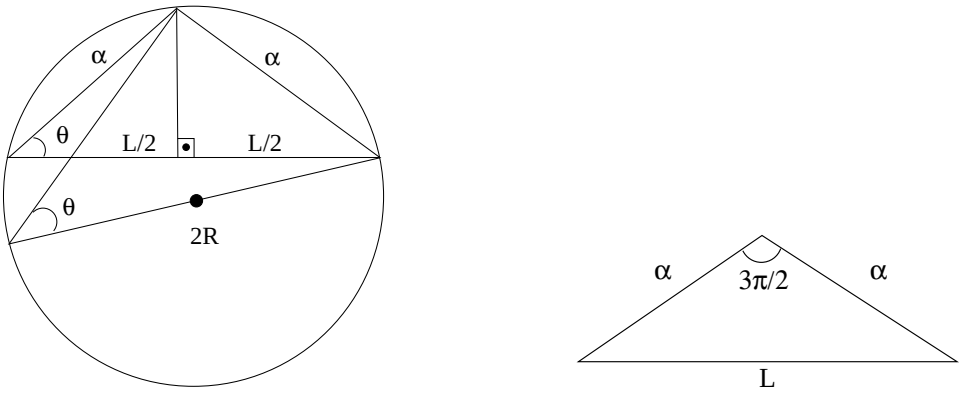


Fig. A.3: Maximizing RL with $a = b = \alpha$ fixed (left) and the worst case configuration (right).

A.2.1 An Analytical Proof

We have verified the upper bound by optimizing an objective function using the Maple software. The problem can be modeled as:

$$\min_{a,b,L}(-\frac{L^4}{A^2}) \tag{A.1}$$

s.t.

$$a \leq \alpha,\ b \leq \alpha \text{ and } R^2 \leq \alpha^2.$$

where by the known Heron's formula we have $A^2 = p(p-a)(p-b)(p-L)$, with $p = \frac{a+b+L}{2}$. For parameter radius R we find again $R = \frac{abL}{4A}$. Our problem can be solved using Lagrangian multipliers with KKT conditions. For $x = (a, b, L)$ and $\mu = (\mu_1, \mu_2, \mu_3)$, the Lagrangian of the constrained optimization problem is:

$$f(x,\mu) = -\frac{L^4}{A^2} + \mu_1 \cdot \underbrace{(\alpha - a)}_{h_1} + \mu_2 \cdot \underbrace{(\alpha - b)}_{h_2} + \mu_3 \cdot \underbrace{(\alpha^2 - \frac{a^2 b^2 L^2}{16A^2})}_{h_3} \quad \text{(A.2)}$$

which gives optimality conditions for $x^* = (a^*, b^*, L^*)$ and $\mu^* = (\mu_1^*, \mu_2^*, \mu_3^*)$ as a local minimum:

i $\bigtriangledown_x(L(x^*, \mu^*)) = 0$

ii $\mu^* \geq 0$

iii $\mu_i^* \cdot h_i(x^*) = 0$, where h_i is defined in equation A.2.

Since there are three complementary conditions, there are eight cases to check:

i $\mu_1 = 0$, $\mu_2 = 0$, $\mu_3 = 0$: gives $\{(a = a, b = b, c = 0), (a = 0, b = 0, c = 0)\}$ which are not feasible.

ii $\mu_1 = 0$, $\mu_2 = 0$, $\alpha^2 - R^2 = 0$: gives no solution.

iii $\mu_1 = 0$, $\alpha - b = 0$, $\mu_3 = 0$: gives $\{(a = a, b = \alpha, c = 0)\}$ which is not feasible.

iv $\mu_1 = 0$, $\alpha - b = 0$, $\alpha^2 - R^2$: gives no solution.

v $\alpha - a = 0$, $\mu_2 = 0$, $\mu_3 = 0$: is symmetric to item 3.

vi $\alpha - a = 0$, $\mu_2 = 0$, $\alpha^2 - R^2$: symmetric to item 4.

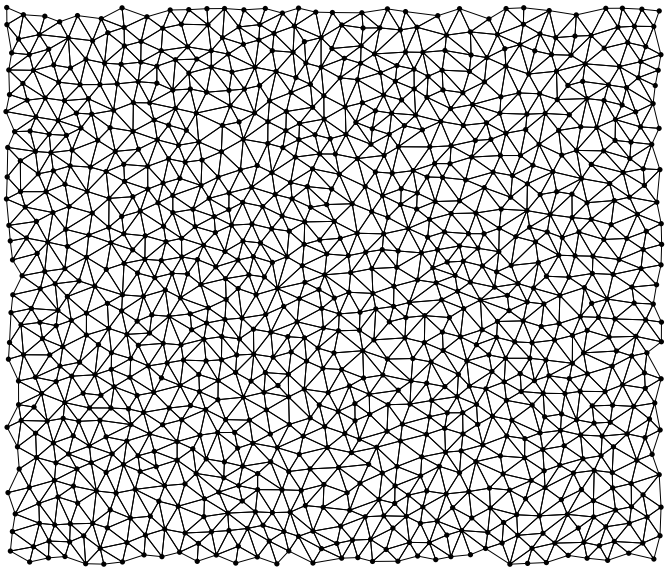
vii $\alpha - a = 0$, $\alpha - b = 0$, $\mu_3 = 0$: gives no solution.

viii $\alpha - a = 0$, $\alpha - b = 0$, $\alpha^2 - R^2$: gives $\{(a = \alpha, b = \alpha, c = \alpha\sqrt{3})\}$

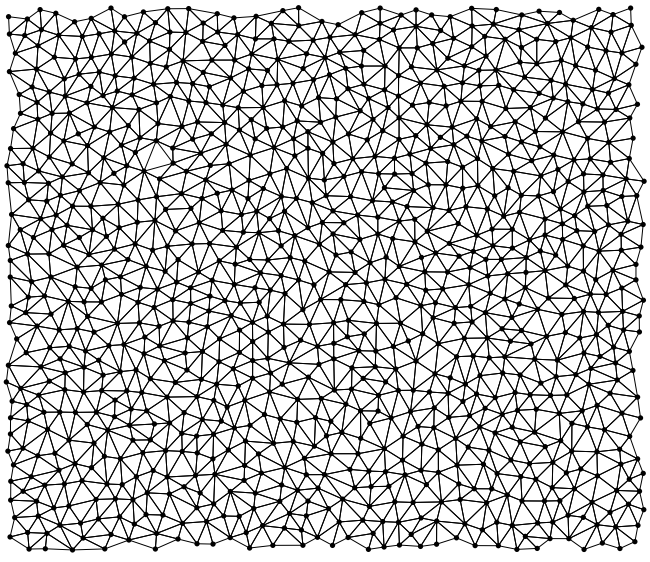
In conclusion, the solution of optimization problem A.1 is $\{(a = \alpha, b = \alpha, c = \alpha\sqrt{3})\}$.

A.3 Conclusion

In Figure A.5, we compare the quality of two triangulations generated from two different PDS samplings of the same rectangular region. We check that the quality of the triangles does not depend on a specific PDS and that the aspect ratio is more concentrated on the left of the histogram which is fair. In figure A.6 we have the numerical statistics of the average, standard deviation, minimum and maximum. The low standard deviation indicates that the data is little spread around the average as well the minimum and maximum validate the aspect ratio interval bounds argued in this article.

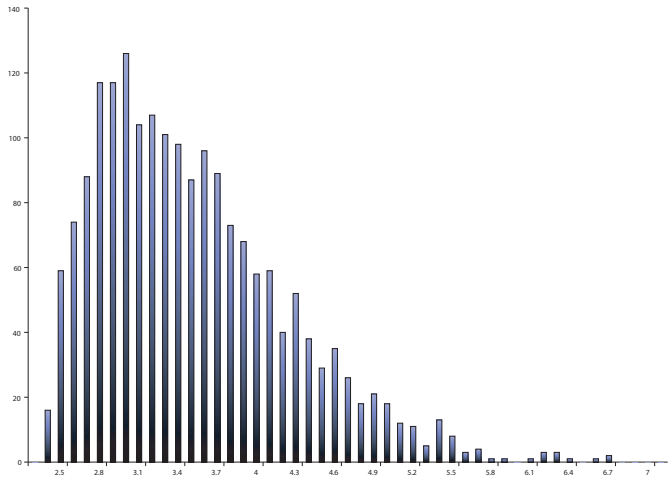


(a)

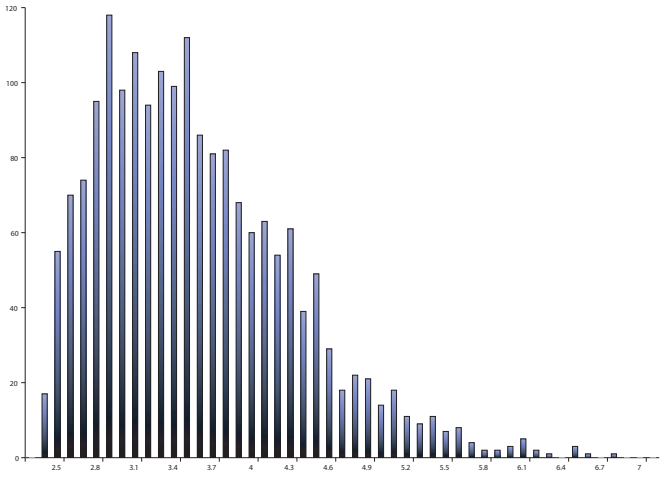


(b)

Fig. A.4: Two Poisson disc samples with the same parameter α.



(a)



(b)

Fig. A.5: The corresponding aspect ratios distributions of figure A.4.

Example	Average	Standard Deviation	Minimum	Maximum
Top	3.46	0.73	2.31	6.68
Bottom	3.52	0.75	2.32	6.70

Fig. A.6: Data table of figure A.5.

B. A SAMPLING CONDITION

Consider a region R and its PDS P_α. In this appendix we will prove that for a sufficiently small parameter radius α the polytope $|C_\alpha(P_\alpha)|$ is topologically equivalent to R. Now we derive we will enunciate and prove five lemmas.

Lemma 3. *A disk containing a point $p \in \partial R$, with diameter at most $LFS(p)$, intersects ∂R in a topological disk.*

Proof. See Amenta et. al[2]. □

Lemma 4. *Let R be a region and P_α a PDS such that $\alpha \leq \frac{1}{2} inf_{p\in\partial R} LFS(p)$ then $C_\alpha(P_\alpha) = \overline{C_\alpha(P_\alpha)}$.*

Proof. According to definition of solid simplicial complexes we have two cases. They correspond to the property that $C_\alpha(P_\alpha)$ does not contain isolated simplices.

First case: $C_\alpha(P_\alpha)$ does not have isolated points.

Suppose that p is an isolated point. Then we have that:

$$B_\alpha(p) \cap (\cup_{q\in P_\alpha q\neq p} B_\alpha(q)) = \emptyset$$

Let $w \in P_\alpha - \{p\}$. As R is connected there exists a path pw contained in R. But this is an absurd because there is a ϵ ring neighborhood of $B_\alpha(p)$ that does not cointain any point of R and pw must pass through this ring.

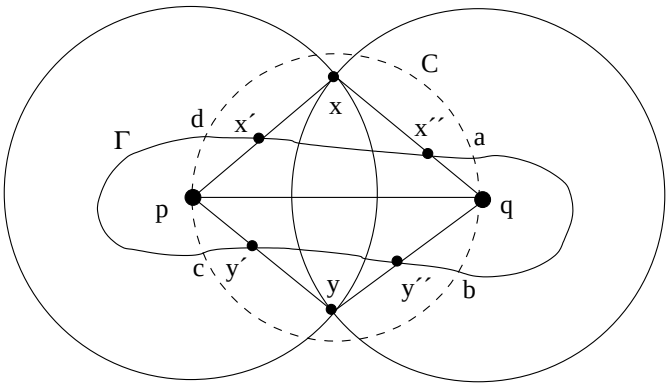


Fig. B.1: The isolated edge pq and the boundary configuration.

Note that we proved this first result by using only connectivity arguments which means that, independently of α $C_\alpha(P_\alpha)$, is always a connected graph.

Second case: $C_\alpha(P_\alpha)$ does not have isolated edges.

Suppose that e is an isolated edge. Looking at the figure B.1 we consider p and q the vertices of e. Let $\{x, y\} = \partial B_\alpha(p) \cap \partial B_\alpha(q)$. We will show that $\{x, y\}$ are outside. Suppose $x \in R$ then by the coverage condition there exists a point $r \in P_\alpha$ such that $x \in B_\alpha(r)$. Then we have that $B_\alpha(r) \cap B_\alpha(p) \cap B_\alpha(q) \neq \emptyset$ and we conclude that the circunscruibed circle of the triangle pqr has radius less or equal than α and it does belong to $C_\alpha(P_\alpha)$. This is an absurd because we have supposed that e is isolated. Hence $x \notin R$. Using the same arguments we can prove that $y \notin R$.

As a consequence of the intermediate value theorem the line segment px intersects ∂R in some point x'. Analogously for the line segments qx, py and qy we have the boundary points x'', y' and y''. See figure B.1.

Take a circle with diameter $x'x''$. It is easy to see that size $|x'x''| < 2\alpha$. Using the fact that $2\alpha < LFS(z)$ for all z in ∂R and using the lemma 3

we have that this circle intersects ∂R in a topological disk. Lets denote this interval as $[x', x'']$. Analogously we get the intervals $[x'', y'']$, $[y'', x']$ and $[y', x']$. All these interval joined compose the closed boundary Γ. Since the interval $[x'', y'']$ is also part of the interval of the disk $B_\alpha(q)$ it separates this disk into two regions, one in interior to R and other exterior to R. As p is a interior point then the interval $[x'', y'']$ must intersect the circle C with diameter pq at least in two distinct points a and b in order to contain the point q. Analogously $[x', y']$ intersects C in more two points b and c. We have that $pq < 2\alpha$ and, again, using the lemma 3, C intersects ∂R in an interval. This is an absurd because all interval has only two extremes and we found at least four, that is a, b, c and d.

This lemma show us that $C_\alpha(P_\alpha)$ is a solid object.

□

Lemma 5. *Let R be a region and P_α a PDS such that $\alpha \leq \frac{1}{2} inf_{p \in \partial R} LFS(p)$. If p and q are two boundary points such that $B_\alpha(p) \cap B_\alpha(q) \neq \emptyset$ then $(B_\alpha(p) \cup B_\alpha(q)) \cap \partial R$ is a topological disk.*

Proof. Suppose that $B_\alpha(p) \cap B_\alpha(q) \cap \partial R$ is not a topological disk. Then we have the configuration of the figure B.2(a). We will take an argument similar to the one used in the preceding lemma. The interval $I = B_\alpha(p) \cap \partial R$ splits the disk $B_\alpha(p)$ into two regions, one interior of R and other exterior. As p is inside R then the boundary of the dashed circle C with diameter pq intersects I in two points a and b. In the same way the interval $I' = B_\alpha(q) \cap \partial R$ intersects the boundary of C in two points c and d. The circle C has diameter less than α and it intersects ∂R in a topological disk. This is an absurd because ∂C has four intersection points with ∂R, that is a, b, c and d.

In conclusion, given B_i a boundary component of R then it is covered by

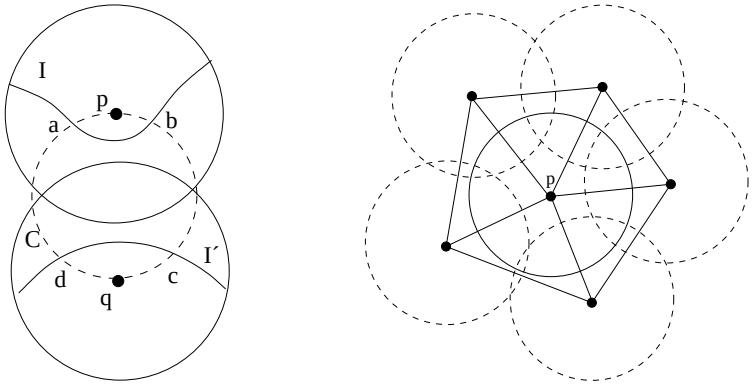


Fig. B.2: (a)Configuration of two boundary disks with no empty intersection and disjoint intervals. (b)Interior point p and its neighbor sampling points.

a closed chain of circles, that is, its dual graph is homeomorphic to B_i. □

The next defintion will assign topology information to sampling points. It will be used to prove the following lema.

Definition 35. *Let P_α be a PDS of a region R and $p \in P_\alpha$ a sample point. We say that p is boundary if $B_\alpha(p) \bigcap \partial R \neq \emptyset$. On the contrary we say that p is interior.*

Lemma 6. *Let R be a region and P_α a PDS such that $2\alpha \leq inf_{p \in \partial R} LFS(p)$. Given $p \in P_\alpha$, if p is interior then p is in the interior of $C_\alpha(P_\alpha)$.*

Proof. As p is a interior point then the disk $B_\alpha(p)$ is all contained in R. Let $A \subset P_\alpha$ be a set of points such that $\partial B_\alpha(p) \subset \cup_{q \in A} B_\alpha(q)$. The points in A falls exactly in the link of p and they are connected. Then p falls in the interior of $C_\alpha(P_\alpha)$. See figure B.2(b).

□

Lemma 7. *Let R be a region and P_α a PDS such that $\alpha \leq \frac{1}{2} inf_{p \in \partial R} LFS(p)$. The region R and $|C_\alpha(P_\alpha)|$ have the same number of boundary components.*

Proof. We will establish a bijection between the boundary components of R and $|C_\alpha(P_\alpha)|$.

Consider B_i a boundary component of ∂R. By lemma 5 there exists a chain of circles covering this component such that its dual graph C_i is homeomorphic to B_i.

Let C_i be a boundary component of $|C_\alpha(P_\alpha)|$. By lemma 6 a interior point $p \in P_\alpha$ is in the interior of $|C_\alpha(P_\alpha)|$ then we conclude that C_i has only boundary points of the same connected component. Hence C_i is associated directly with the boundary such that its points belong. □

Theorem 8. *Let R be a region and P_α a PDS such that $\alpha \leq \frac{1}{2} inf_{p \in \partial R} LFS(p)$. Then $|C_\alpha(P_\alpha)| \sim R$. The symbol $\sim$ means that the two spaces are topologicaly equivalent.*

Proof. Since R and $|C_\alpha(P_\alpha)|$ are solid objects then they have the same number of connected components. It is well known that this result implies they are topologically equivalents. □

Conclusion

In figure B.3, we have two regions sampled in two different resolutions. The first region B.3(a) has the topology defined by two holes. They are two circles such that the biggest circle's radius is approximately the quadruple of the lesser circle's radius. Having the first sampling parameter α as half of the biggest circle's radius, we observe in B.3(c) that the α-complex, as expected by theorem 8, captures only the topology of this circle. As we decrease the parameter α to half of the lesser circles radius then, as we can see in figure B.3(e), the α-complex captures the whole topology. In the

second region B.3(b) there is one hole such that its shape is composed by two identical circles joined by a rectangular pipe. In figure B.3(d), when we choose the parameter α as half of the minor circle's radius the α-complex captures only the topology of the circles. As we decrease the parameter by half of the height of the pipe, then in figure B.3(f) we check that the whole topology is recovered. These examples make us reason in multiresolution modeling of solid objects. As we increase the sampling rate, we are able to identify two types of details that are emphasized: the geometrical details characterized by the shape of the holes and the detail characterized by the number of holes. The last one is what we called "topological" detail. This is what we will go deep in future works.

The aspect ratio result of this paper is not directly extendible for dimension three. A simple counter-example involves a sliver tetrahedron, generated by perturbing four samples on the vertices of a square. The aspect ratio of this tetrahedron has an aspect ratio arbitrarily big and locally satisfies the PDS condition.

Although there is no guarantee of limited aspect ratio for higher dimensions we can measure them by means of probability distribution. The problem can be formulated in stochastic terms in a future work.

We hope that the result of sampling condition can be generalized for dimension three. The **inf** constraint for local feature size of the boundary is very strong because it turns the sampling condition a global parameter. Indeed, the Poisson Disc Sampling is also a global parameter. In a future work we plan to create a less constraint condition based on a formalization of adaptive Poisson disc samplings. This adaptiveness can also be obtained by others metrics assigned to the plane. It will give us a local sampling condition instead of a global one on uniform poison disc samplings.

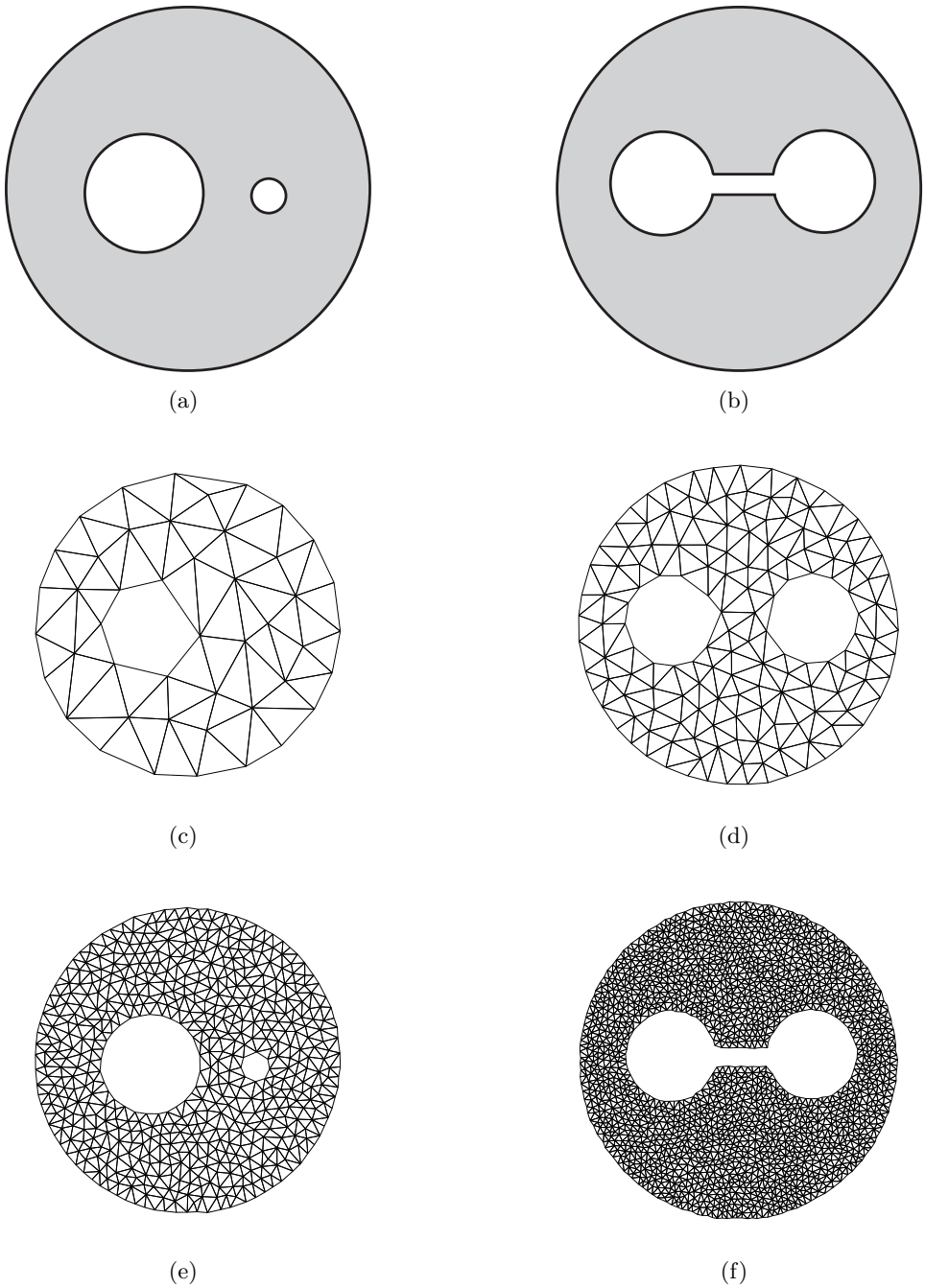


Fig. B.3: Examples of Poisson disc samples in multiresolution on two different regions and their solid α-complexes.

C. RBPA: RESTRICTED BPA

In this apendix we propose a new 2D triangulation method based on the ball-pivoting algorithm (BPA). The BPA is an interesting advancing front approach for surface reconstruction that uses a ball of fixed radius traversing the 3D sample points by pivoting front edges and attaching triangles to the mesh. Given a set of 2D points, our method applies the BPA on them assuming that they have a constant third coordinate. We show that such geometrical restriction implies in several simplifications on the original BPA implementation. We demonstrate that the resulted triangulation is a solid alpha complex, a special subset of Delaunay Triangulations that is closely related to alpha shapes. The BPA efficency is extremely dependent on the uniformity of the sampling and on the ball radius. We also present an efficient generalization of our method to obtain, in an adaptive way, 2D solid alpha complexes of generic samplings (uniform or non-uniform) free from the influence of ball size.

C.1 introduction

The Ball-Pivoting Algorithm is an advancing front Lo [26], Paul Louis George [35] algorithm for surface reconstruction, one of the most powerful among the incremental surface reconstruction methods. BPA is based on growing a surface by moving its boundary curves until the geometry and topology of the

whole object is captured. For algorithms of this class it is necessary a criteria to choose a new element to be assigned to the mesh de Medeiros Filho [11]. In the case of BPA, the criteria is a ball of fixed radius traversing the sample points by pivoting front edges of a current active boundary. Next, we summarize the geometric step, data struture involved and the mesh construction (algorithm outline). For more details Bernardini et al. [4], de Medeiros Filho [11].

Geometric step. The input of the algorithm are three dimensional set points $P = \{p_1, p_2, ..., p_n\}$, their normals and a fixed positive real parameter α. The geometric step of BPA takes a boundary edge $e_{ij} = \{p_i, p_j\}$ (pivot) and the sphere S of fixed radius α which has e_{ij} as a chord. The ball is turned around e_{ij} until it touches a point p_k. This point will be the only candidate to compose a new triangle with p_i and p_j (see Figure C.1). For this cadidate edge a test for normal consistence is performed and non-manifold cases are verified. To start the mesh construction there is another geometric routine, that find the seed triangle, i.e, three points such that the simplex they compose is not in the current mesh and it is α-exposed.

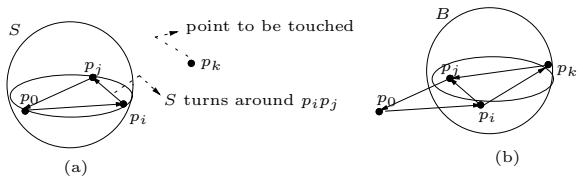


Fig. C.1: Ball Pivoting intuition. In the begining the front is composed by the polygonal $p_1p_2p_3$ (a). After pivoting the new polygonal of the front is $p_1p_2p_3p_4$ (b).

From the above description of the algorithm we have the following remark:

Observation 1. *The point p_k returned by the geometric step is such that σ_T, $T = \{p_k, p_i, p_j\}$, is α-exposed.*

Data structure. In general, BPA implementations have a graph G which represents point conectivity and two more fundamental data structures (e.g. half edge Baumgart [3]). The first one is the front F, a collection of connected boundary curves that stores the contour of the current mesh front. The second fundamental data structure is a uniform 3D grid that takes advantage of the local property of the geometric step to speed up the algorithm. We define the size of the voxel in this grid as 2α. The reason for that is: a canditate to be the first touched point is the one whose distance from it to the center of the pivoting chord is less than 2α. Notice that the grid takes linear time to be constructed. We will call the 27 neighbors voxels of a point p in the grid as V_p and denote $\#V_p$ as the number of points in V_p.

Algorithm outline. Initially the graph G contains only the first seed triangle and the front F corresponds to its three edges. The algorithm goes updating G by performing ball pivoting steps (candidate points are searched in V_p) once in each edge of the current front F until there is no more seed triangles. In each pivoting step this collection is updated by performing glue operations that join or separate boundary curves (for more details [11]).

C.2 Our method

Though BPA algorithm be usefull for 3D models, in general for range scans, we want to investigate what happen when we apply it in a set points restricted to a plane. More precisely, take an isometric embbeding application $i : \mathbb{R}^2 \hookrightarrow \mathbb{R}^3$, say $i(x, y) = (x, y, 0)$ and apply BPA. This is what we call RBPA, a particular case of the BPA. We are now to discuss some particular

features of RBPA which simplifies its implimentation, when compared to BPA:

- it use a 2D uniform grid: the neighbors of a point are on the 9 pixels around it instead of the 27 voxels of the 3D grid;
- there is neither normals consistence test (normals information is not necessary) nor non-manifold cases verification for candidate triangles;
- it is not necessary to treat the glue case when two boundary curves in the same connected component are joined into one (i. e., a genus could not be created in a planar surface); and
- the geometric step is simplified observing that to compute the smallest pivoting angle of the ball on some edge e_{ij} is equivalent to compute the biggest angle composed by the candidate point with e_{ij} (see fig. C.2).

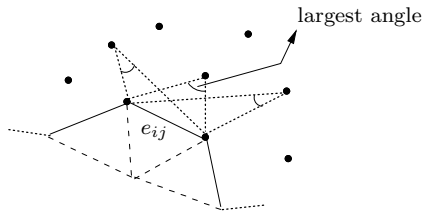


Fig. C.2: Geometric sipmlification of the candidate point selection.

In its essence RBPA does not differ from BPA (note that observation 1 is also valid for RBPA). However, we are interested in its output and how it can contribute to 2D triangulation problems and sampling analysis in a qualitative and practical way. In the next section we show the RBPA relation with 2D solid alpha-complex.

C.3 2D Solid Alpha-Complexes and RBPA

Now, we go to an important point of this paper by asking the following question:

How does the triangulation of RBPA look like, given $0 \leq \alpha \leq \infty$?

The answer is: it looks like exactly to the 2D solid alpha-complex of the points. Indeed, let $\mathcal{T}_\alpha$ be the set of triangular faces built by RBPA. From observation 1 we have that each $\sigma_T \in \mathcal{T}_\alpha$ is α-exposed by a 3D-ball. Therefore, in the restricted plane its circunscribed circle b_T with radius μ_T is empty with $\mu_T \leq \alpha$. So, by definition 2, we have that $\mathcal{T}_\alpha \subset \widehat{\mathcal{C}}_{\alpha(S)}$. To prove the converse we must suppose that the seed triangle selection is *ideal* in a sense that it returns a simplex iff there exists a set of three points $T = \{p_1, p_2, p_3\}$ such that σ_T is α-exposed and there is no segment of this triangle in the mesh. Let $\sigma_T \in \widehat{\mathcal{C}}_\alpha(S)$. Since the seed selection is ideal, one segment of T, say p_1p_2 belongs to the mesh. At some moment runnning the RBPA, the segment p_1p_2 was in the front and one of the two cases could occur: (i) a pivoting was perfomed on p_1p_2 or (ii) it was glued to another boundary edge. Clearly, in the second case we have got to the conclusion. Since $\mu_T \leq \alpha$, in the first case p_3 is V_m, where m is the mid point of p_1p_2, and geometrically it is the unique cadidate point to be retuned by the ball pivoting step. As a conclusion, in both cases we have $\sigma_T \in \widehat{\mathcal{C}}_\alpha(S)$. More precisely, we have the following theorem:

Theorem 9. *Let $S \subset \mathbb{R}^2$ be a set of points in general position and $\widehat{\mathcal{C}}_\alpha(S)$ its solid alpha-complex. Consider $\mathcal{T}_\alpha$ as the triangulation output of the RBPA running on the image of the embbeding i with the "ideal" seed triangle selection. Then $\widehat{\mathcal{C}}_\alpha(S) = \mathcal{T}_\alpha$.*

Note: Though in our implementation we do not have an "ideal" seed selection, to speed up the algorithm we adopted an heuristic that takes linear time independent on the number of calls. The approach looks at each pixel grid once and search for candidate triangles in a constant time. As we have a linear non empty number of pixels the total seed selection cost is linear.

Theorem 1 give us an important insight: it generates the 2D solid alpha complex without the computation of Delaunay triangulation. In the next section we will show that this computation is very efficient if, considering a suitable radius, we apply RBPA on uniform samplings. Indeed it has linear behaviour.

C.4 RBPA and uniform samplings

It is well-known that scan images generated by the points capture of object surfaces in 3D fotography are very dense and are huge. This implies that surface reconstruction methods [31] should be efficient in order to deal with such type of data. The BPA uses a uniform 3D grid to accelerate the search. This strategy is very suitable because, in each ball pivoting step, only the points on the neighboorhood V_p of the rotation certer p are candidates to build a new triangle in the mesh. Moreover, if the object is uniformly sampled and the value of α is the less possible in such a way that the entire surface is reconstructed, then $\#V_p$ is very small and practically constant at any point of the surface. That is the reason why the BPA is very fast: the querying time at each pivoting step is constant and small. In other words, we can affirm that an optimal performance of the algorithm is achieved when such conditions of sampling characterize an "ideal" instance and the

parameter α is sufficiently small. The same situation holds for RBPA.

We observe two disadvantages of the BPA, also present in the RBPA. When α is big, then $\#V_p$ is also big. Thus, at each pivoting step the algorithm spend more time searching for candidates in V_p. As a consequence, the algorithm slowdown at a point that is inneficient for large quantity of points. The other disadvantage occurs when the sampling is non-uniform. Beyond the same side effect cited above on uniform samplings, the values of $\#V_p$ are unbalanced at several regions. This fact also makes the algorithm inefficient.

In order to solve those problems, we are to propose a new methodology that accelerates the RBPA, iteratively running it with different radius in increasing scales of samplings (in our case we use the diadic scale) until we get to a desired radius. This new method will be denoted by HRBPA (Hierarchical RBPA). It calls RBPA several times in an adaptive way using the most suitable radius for the different sampling scales of the set of points. Moreover, it takes the advantage of the property that the complex generated at a iteration is enclosed on the complex of the previous iteration (property P1 in section 2.3). This allows us to eliminate points that are already on the interior of the mesh. Next section presents this method in detail.

C.5 HRBPA: Hierarchical RBPA

As we presented in the last section, HRBPA is in some sence a "multiscale" version of RBPA. This method speeds up the RBPA itself to compute solid alpha complexes given any positive real α and a generic sampling.

The methodology of HRBPA exploits the efficience of RBPA with a suitable radius on uniform samplings combined with the solid alpha com-

plex property that if $\alpha_1 \leq \alpha_2$ then $\widehat{\mathcal{C}}_{\alpha_1} \subset \widehat{\mathcal{C}}_{\alpha_2}$. More preciselly, it applies iteratively RBPA in an adaptative way with a diadic resolution, i.e, from a "minimal" radius estimation α_0 in each iteraction step, say i, we run RBPA with ball radius $2^i\alpha_0$. The total number of steps is $\log(\alpha/\alpha_0) + 1$. Next we explain how to estimate the minimal radius and how the iterative triagulation goes from one level step to the next one. Consider as input of HRBPA the real positive parameter α and the set of points S.

Minimal Radius Estmation. We tackled it by looking for small clusters of k-points in a quadtree constructed on the set of points such that each leaf has at most k points, where k is an integer controlled by the user. Then, we set α_0 as a half of the smallest leaf resolution (see fig C.3.).

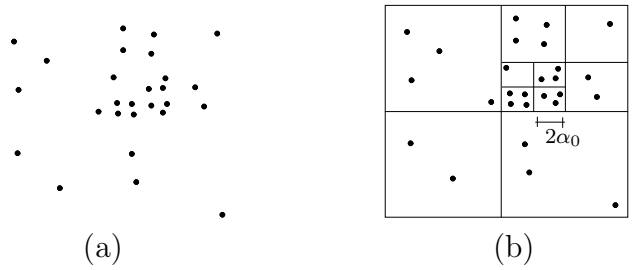


Fig. C.3: Computing the minimal radius resolution.

Iterative Triangulation After the minimal radius α_0 be estimated we evaluate $\alpha < 2\alpha_0$ (*). If (*) is true we apply RBPA with radius α and stop triangulation. Since $\widehat{\mathcal{C}}_{\alpha_0} \subset \widehat{\mathcal{C}}_{2\alpha_0}$, in the case of (*) be false, we apply again RBPA with radius $2\alpha_0$ only on the points lying in the set $S - int(\widehat{\mathcal{C}}_{\alpha_0})$ by advancing the boundary of $\widehat{\mathcal{C}}_{\alpha_0}$, which composes the front F. Therefore, to go from a lower to the higher resolution, we need only transfer boundary and isolated points (see fig. C.4) and continue advancing the front F. Analogously, we apply the same process to the others levels.

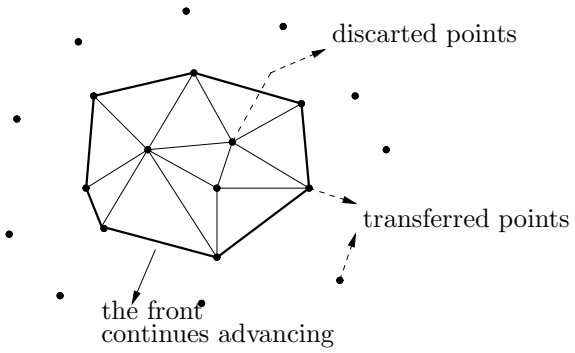


Fig. C.4: Iterative triangulation step.

Summarizing HRBPA, for $0 \leq \alpha \leq \infty$ we have the following steps:

i Compute minimal radius α_0, set $\alpha_i = \alpha_0$ and $i = 0$;

ii Transfer boundary and isolated points as input;

iii If $\alpha < \alpha_i$ then apply RBPA with radius α and stop, else apply it with radius α_i;

iv Set $i = i + 1$, $\alpha_i = 2\alpha_{i-1}$ and go back to step 2.

Complexity. The HRBPA complexity for $\alpha = \alpha_0$, without considering the complexity of the quadtree construction, corresponds to the "ideal" RBPA case on regular samplings and it has linear behaviour because in each pivoting step we have that $\#V_p$ is constant. Most part of the linear constant depends on data structure implementation [1] and on the parameter k as the maximum number of points in each leaf. In figure C.5 we show a graphic to illustrate RBPA linearity on uniform samplings.

[1] We used a generic data structure [43] which comprises geometry and topology in a unified framework for representations of meshes with or without boundaries. It is a very flexible framework that makes easier future advanced studies on topology and sampling analyses we plan as a future work.

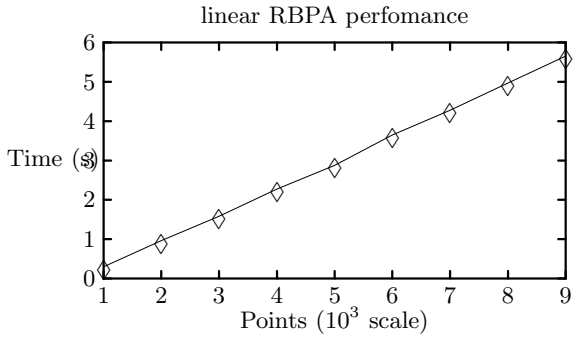


Fig. C.5: Timing graphic.

For generic samplings we do not have a global complexity bound on the number of points but we can analyse and evaluate it in parts.

The preprocessing cost to build the quadtree depends on the depth d of the quadtree which can be shown that $d = \log(s/c) + 3/2$ where s is the side length of the initial square and c is the diameter of the smallest cluster with k points (adaptaded from [9] cap 14 pg. 293). Therefore building the quadtree takes $O((d+1)n)$ time, where n is the number of points.

There is also the triangulation cost in each iteraction step considering only the boundary and isolated points of the previous triangulation step to continue advancing the front. Since at each level we apply RBPA with the appropriate radius (i. e. we fall in the ideal case of RBPA) thus we have a linear bound time. In conclusion, if there are k_i boundary and isolated points in the level step i, RBPA takes $O(k_i)$ time cost to transfer these points, build the grid data structure and perform triangulation. Therefore the total cost of HRBPA is $O((d+1)n + \sum k_i)$, where $k_j \leq k_i$ for $i < j$.

C.6 Examples

In figures C.6, C.7 and C.8 we summarize applications with examples.

Figure C.6 shows in the first picture a uniform sampling of the alpha symbol and its quadtree structure detecting a minimal resolution of the points. Applying HBPA in the ideal case ($\alpha = \alpha_0$), we obtain a nice result by recovering the actual shape of the samples. As we depicted in (a) and (b) the recognized shape can be represented through the solid alpha complex or through its alpha solid, respectively.

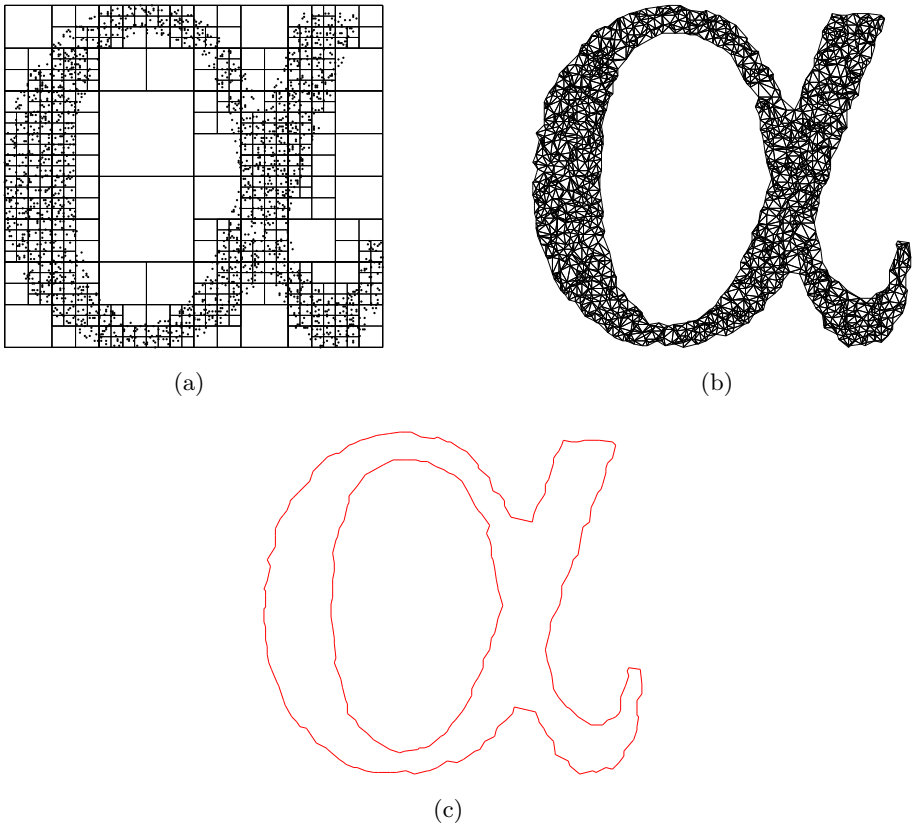
Figure C.7 shows an example which is very applicable to computational biology analysis and related domains. In picture (a) we have a preprocessed image of a molecule and its quadtree structure detecting minimal resolution. Running HRBA with the minimal radius it did not capture the whole topology (i.e. atoms) of the molecule (see picture (b)). However, by increasing slightly the radius we obtain the correct one (see picture (c)).

In figure C.8 we show an example that illustrate how HBPA improves RBPA performance. In the first picture we have a non uniform sampling containing 700 points and its two resolution levels (α_0 and $2\alpha_0$) captured by the quadtree structure. Applying RBPA with radius size in the largest level ($\alpha = 2\alpha_0$) we obtain, as a result, the alpha complex depicted in (c) at time running of 50ms. Applying HRBPA, which first compute the alpha complex depicted in (b), we obtain for the same radius the time running of 25ms (more preciselly 15ms in the first level and 8ms in the second one).

C.7 Conclusion

Based on BPA principles, in this paper we introduced the concept of *solid alpha complexes*. We present a new method for the construction of this kind

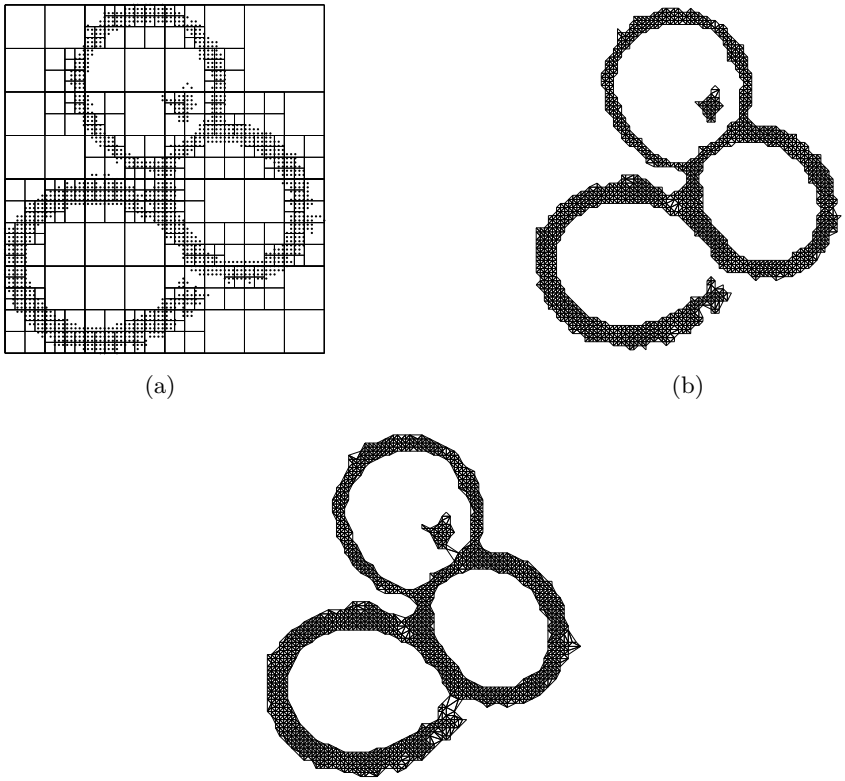
of object, the RBPA. We showed that RBPA is an algorithm whose efficiency is dependent on the uniformity of the sample points and on the ball size. To solve this problem we proposed the hierarchical RBPA (HRBPA), which generalizes the RBPA method. It obtains, in an adaptive way, 2D solid alpha complexes of generic samplings (uniform or non-uniform) free from the influence of ball size.



(a) (b)

(c)

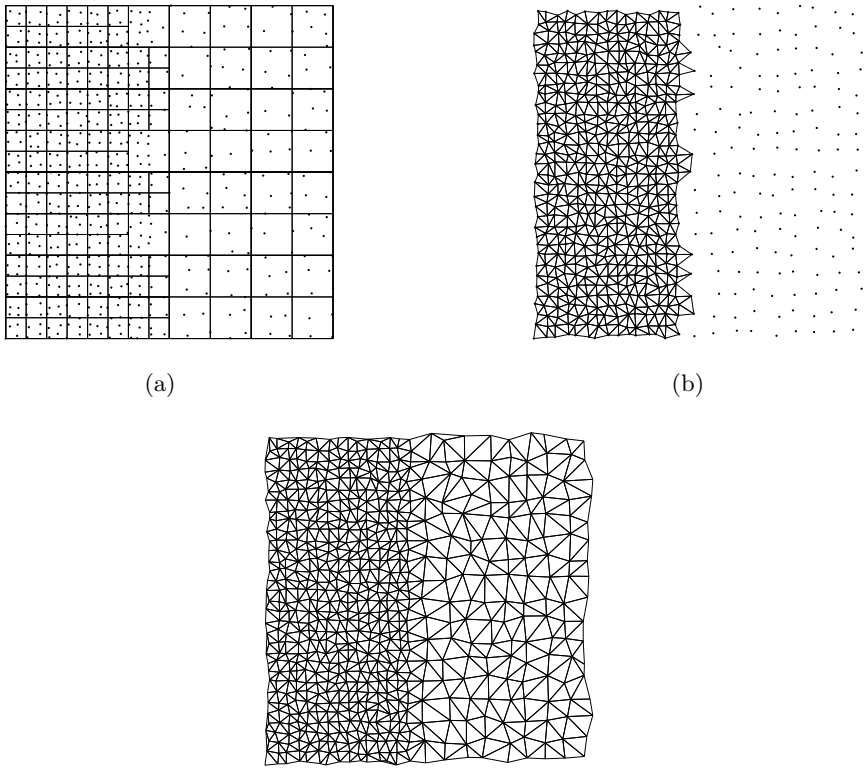
Fig. C.6: (a) sampling of the alpha symbol, its (b) solid alpha complex and its (c) alpha solid.



(a) (b)

(c)

Fig. C.7: Preprocessed image of a molecule (a), its solid alpha complex for $\alpha = \alpha_0$ (b) and its topology recovered with a slight increase in the ball size (c).



(a) (b)

(c)

Fig. C.8: Multiscale sampling with two levels (a) and its two reconstructed levels of detail, (b) and (c).

BIBLIOGRAPHY

[1] J. Alexander. The combinatorial theory of complexes. *Ann. Math.*, 31: 294–322, 1930.

[2] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP*, 60(2):125–135, 1998.

[3] B. G. Baumgart. A polyhedron representation for computer vision. *AFIPS National Computer Conference*, (44):589–596, 1975.

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, /1999.

[5] Fausto Bernardini and Chandrajit L. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proc. 9th Canadian Conf. Computational Geometry*, pages 193–198, 1997.

[6] Fausto Bernardini, Chandrajit L. Bajaj, Jindong Chen, and Daniel Schikore. Automatic reconstruction of 3d CAD models from digital scans. *International Journal of Computational Geometry and Applications*, 9(4/5):327–369, 1999.

[7] Siu-Wing Cheng. On the sizes of delaunay meshes. *Comput. Geom. Theory Appl.*, 33(3):130–138, 2006. ISSN 0925-7721.

[8] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, 1986. ISSN 0730-0301.

[9] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer-Verlag, 1997.

[10] L. De Floriani, P. Magillo, and E. Puppo. Efficient implementation of multi-triangulations. *Visualization '98. Proceedings*, pages 43–50, Oct. 1998. ISSN 1.

[11] Esdras Soares de Medeiros Filho. Algoritmo ball-pivoting: Contextualização e estado da arte. Master's thesis in portuguese, IMPA, August 2003.

[12] Laurent Demaret, Nira Dyn, and Armin Iske. Image compression by linear splines over adaptive triangulations. *Signal Process.*, 86(7):1604–1616, 2006. ISSN 0165-1684.

[13] Olivier Devillers. On deletion in delaunay triangulations. pages 181–188, 1999.

[14] H. Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry (DCG)*, 13:415440, 1995.

[15] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. page 454, 2000.

[16] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, 1994. ISSN 0730-0301.

[17] P. Frey and H. Borouchaki. Surface mesh evaluation. In *In 6th International Meshing Roundtabl*, pages 363–374, 1997.

[18] Jonas Gomes, Bruno Costa Silva, Lucia Darsa, and Luiz Velho. Graphical objects. *The Visual Computer*, 12(6):269–282, 1996.

[19] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.

[20] S. Hiller, O. Deussen, and A. Keller. Tiled blue noise samples. In *Proceedings of Vision, Modeling, and Visualization 2001*, pages 265–271. IOS Press, 2001.

[21] Hugues Hoppe. Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4.

[22] Leif Kobbelt. $\sqrt{3}$-subdivision. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 103–112, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5.

[23] C. L. Lawson. Software for c^1 surface interpolation. In John R. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, New York, 1977.

[24] Thomas Lewiner, Marcos Craizer, Hélio Lopes, Sinésio Pesco, Luiz Velho, and Esdras Medeiros. Gencode: geometry–driven compression in arbitrary dimension and co–dimension. In *18th Brazilian Symposium*

on Computer Graphics and Image Processing, pages 249–256, Natal, RN, October 2005.

[25] W. B. R. Lickorish. Simplicial moves on complexes and manifolds. In *Proceedings of the Kirbyfest*, volume 2, pages 299–320, 1999.

[26] S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21(8): 1403–1426, 1985. doi: 10.1002/nme.1620210805.

[27] B. H. McCormick. Visualization in scientific computing. volume 10, pages 15–21, New York, NY, USA, 1988. ACM.

[28] E. Medeiros, L. Velho, and W. Lopes. Restricted bpa: applying ball-pivoting on the plane. *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, pages 372–379, Oct. 2004. ISSN 1530-1834.

[29] Esdras Medeiros, Luiz Velho, Hélio Lopes, and Thomas Lewiner. On 2d solid alpha-complexes of poisson disc samplings. *International Journal of Shape Modeling*, 15(1-2):77–91, 2009.

[30] Kurt Mehlhorn, Stefan Naher, and Christian Uhrig. The LEDA platform of combinatorial and geometric computing. In *Automata, Languages and Programming*, pages 7–16, 1997.

[31] Robert Mencl and Heinrich Müller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *Dagstuhl '97, Scientific Visualization*, pages 223–232, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0505-8.

[32] J. Milnor. *Morse Theory*. Annals of Mathematics Study 51. Princeton University Press, 1963.

[33] M. H. A. Newman. On the foundations of combinatorial analysis situs. *Proc. Royal Acad.*, 29:610–641, 1926.

[34] U. Pachner. Pl homeomorphic manifolds are equivalent by elementary shellings. *Europ. J. Combinatorics*, 12:129–145, 1991.

[35] Eric Seveno Paul Louis George. The advancing-front mesh generation method revisited. *International Journal for Numerical Methods in Engineering*, 37(21):3605–3619, 1994. doi: 10.1002/nme.1620372103.

[36] H. Poincaré. Sur la géneralisation d'un théoréme d'euler relatif aux poliédres. *C. R. Acad. Sci. Paris*, (117):437–464, 1893.

[37] Enrico Puppo. Variable resolution triangulations. *Computational Geometry*, 11(3-4):219–238, 1998.

[38] Jim Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 83–92, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics. ISBN 0-89871-313-7.

[39] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 72(9):1192–1201, Sept. 1984. ISSN 0018-9219.

[40] Andrei Sharf, Thomas Lewiner, Gil Shklarski, Sivan Toledo, and Daniel Cohen-Or. Interactive topology-aware surface reconstruction. page 43, 2007.

[41] Luiz Velho and Jonas Gomes. Variable resolution 4-k meshes: Concepts and applications. *Computer Graphics Forum*, 19(4):195–212, 2000.

[42] Luiz Velho and Denis Zorin. 4–8 subdivision. *Computer-Aided Geometric Design*, 18(5):397–427, 2001. Special Issue on Subdivision Techniques.

[43] Luiz Velho, Hélio Lopes, Esdras Medeiros, Thomas Lewiner, and Geovan Tavares. Topological mesh operators. Technical Report - VISGRAF Laboratory TR-2005-03, IMPA, October 2005.

[44] Zhu, Sundaram, Snoeyink, and Mitchell. Generating random polygons with given vertices. Technical report, 1996.