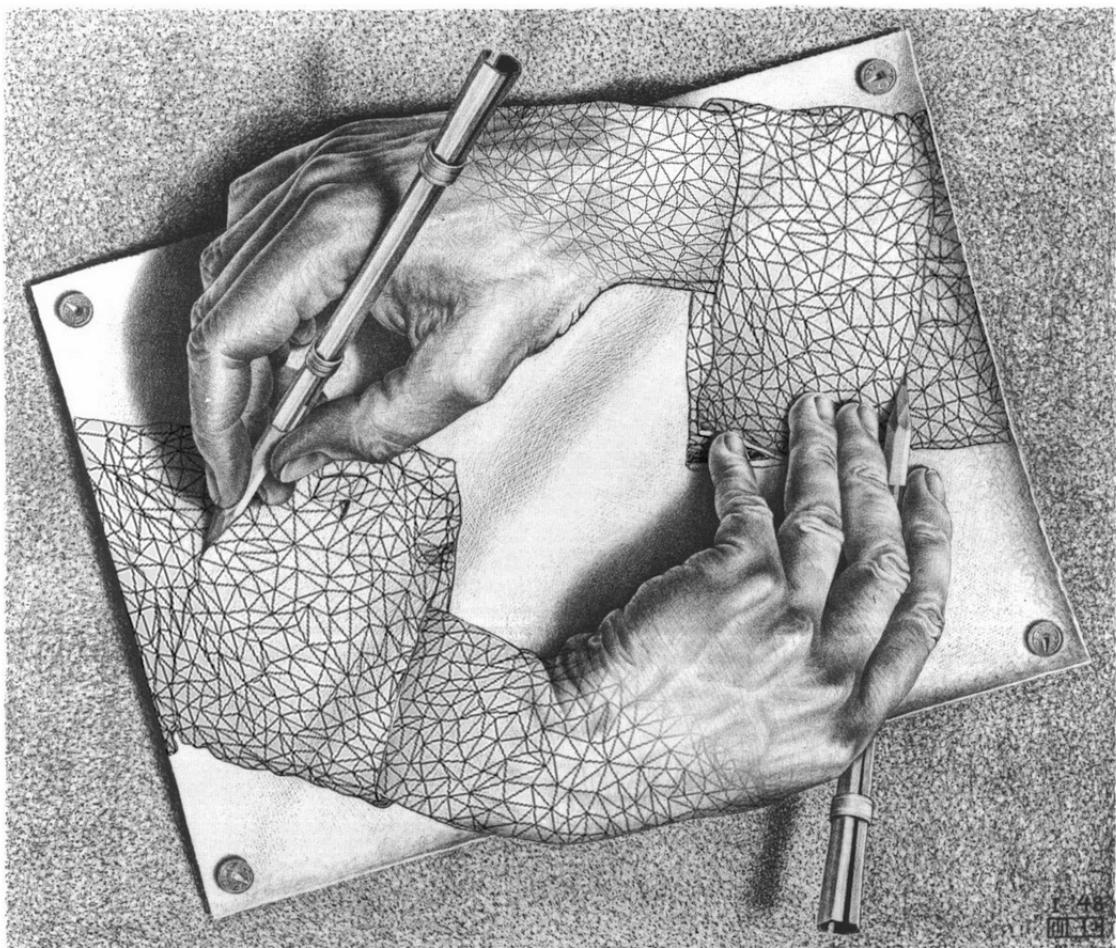


# Fast User-Guided Mesh-Simplification with Topology Control





---

# Fast User-Guided Mesh-Simplification with Topology Control

*Diploma Thesis*  
Department of Computer Graphics  
TU-Berlin

penned by: Lukas N.P. Egger  
mail address: [mail@lnpe.at](mailto:mail@lnpe.at)  
matriculation nr.: 22 48 48  
area of studies: Industrial Engineering  
& Philosophy  
censor: Prof. Dr. Marc Alexa  
second censor: Prof. Dr. Bernd Bickel  
supervisor: Dipl. Inf. Ronald Richter  
filing date: 31<sup>st</sup> of July, 2012





---

## **Abstract**

This thesis focuses on the user-guided simplification of geometric models, i.e. the decimation of 3D embedded triangle meshes. Within this area, research spends its attention mostly on deterministic and fully automated algorithms. The goal of this work however is not to add a new part to the standard ensemble, but instead look into the gains that can be achieved by user interaction combined with non-deterministic approaches. Thus not only the possibility for artistic control changes, but also the benchmarks for various aspects of the problem, e.g. speed has to be gauged with the requirements for fluid interactive work in mind. Additionally we show how techniques derived from the field of topological persistence can be used to effectively measure and control the topology of a mesh.

Apart from the obligatory discussion of results there is also a brief historical outline of mesh simplification, a primer to the mathematical skill-set necessary to tackle the emerging phenomena, as well as an outlook on further research questions.

## **Zusammenfassung**

Diese Arbeit widmet sich der nutzergestützten Vereinfachung von geometrischen Modellen. Innerhalb dieses Bereiches liegt der Fokus gewöhnlich auf deterministischen und vollautomatischen Algorithmen. Ziel dieser Arbeit ist es, nicht dem Kanon an Standardansätzen einen neuen beizufügen, sondern stattdessen die Vorteile von Benutzerinteraktion in Kombination mit nicht-deterministischen Ansätzen zu beleuchten. Dadurch verschiebt sich nicht nur die artistischen Kontrollmöglichkeiten, sondern auch der Maßstab, bspw. muss Geschwindigkeit fortan anhand direktem und flüssigen Arbeitens bewertet werden. Zusätzlich zeigen wir, wie mit Techniken der topologischen Persistenz die Eigenschaften eines Modells ermittelt und verändert werden können.

Neben dem obligatorischen Auswertungsteil, gibt es auch einen kurzen geschichtlichen Abriss der Entwicklung dieses Spezialgebietes, als auch eine Einführung in die Teile der Topologie, welche zur Beschreibung der auftretenden Phänomene benutzt werden, sowie einen Ausblick auf weiterführende Fragestellungen.

---

## Used Notations

Definitions of mathematical terms are generally given within paragraphs of text, rather than displayed separately like theorems<sup>0</sup>.

Table 0.1.: Definitions

$\mathbb{R}, \mathbb{Z}$ , etc.:	rationals, integers and so forth.
$\mathbb{R}^n$ :	$n$ -dimensional Euclidean space, in particular $\mathbb{R}^0 = \mathbb{C}^0 = \{0\}$ .
$S^n$ :	the unit sphere in $\mathbb{R}^{n+1}$ i.e. all points of distance: $ d  = 1$ , from the origin.
$D^n$ :	the unit disk or ball in $\mathbb{R}^n$ i.e. all points of distance: $ d  \leq 1$ , from the origin.
$\partial D^n = S^{n-1}$ :	the boundary of the $n$ -disk, respectively $\partial$ the boundary operator.
$e^n$ :	an $n$ -cell, homeomorphic to the open $n$ -disk: $D^n - \partial D^n$ , for instance $e^0$ consists of a single point, whereas $S^0 = \partial D^1$ of two.
$\mathcal{M}$ :	orientable manifold mesh with $\mathcal{V}$ & $\mathcal{F}$ being the list of vertices & faces i.e. a PLC and $ \cdot $ denoting the the cardinality of a subcomplex, respectively $\ \mathcal{M}_A - \mathcal{M}_B\ $ being the distance between to meshes.
$\cong, \approx, \simeq$ :	isomorphism, homeomorphism and homotopy, with $\approx \Rightarrow \simeq$ but not the other way around.
$\mathbb{X}$ :	topological space with sets $A_i$ .
$\pi_n(\mathbb{X})$ :	the fundamental group of the topological space $\mathbb{X}$ , note that $\pi_n(\mathbb{X}_1) = \pi_n(\mathbb{X}_2)$ does not imply $\mathbb{X}_1 \approx \mathbb{X}_2$ .
$\equiv$ :	congruence relation, i.e. equivalence relation on an algebraic structure.
$K$ :	simplicial complex built from a finite set of simplices: $\{\sigma_{nr}^{dim}\}$ $\sigma_{nr}^0 \rightarrow$ vertices, $\sigma_{nr}^1 \rightarrow$ edges and so forth, together with an associated filtration: $\emptyset = K_{-1} \subset K_0 \subset K_1 \subset \dots \subset K_n = K$ .

There are also a few notations used in this thesis that are not completely standard but increase readability: The union of a set  $A$  with a family of sets  $B_i$ , with  $i$  ranging over some index set, is written simply as  $A \cup_i B$  rather than something more elaborate such as  $A \cup (\cup_i B)$ . Other similar operations are treated in the same way.

---

<sup>0</sup> I follow the style recommendations used by *Allen Hatcher* [cf. [Hatcher, 2002](#)].

# Contents

<b>Abstract &amp; Zusammenfassung</b>	II
<b>Used Notations &amp; Symbols</b>	III
<b>1. Introduction</b>	<b>1</b>
1.1. The Complexity Conundrum . . . . .	1
1.2. The Case for Highly Detailed Geometry . . . . .	3
1.3. Reducing Complexity . . . . .	5
1.4. Design Principles & Workflow Regimes . . . . .	7
1.5. The Outline of this Thesis . . . . .	9
<b>2. The Concept and History of Mesh Simplification</b>	<b>11</b>
2.1. Problem Statement . . . . .	11
2.2. Chronology of Simplification Designs . . . . .	12
2.3. Taxonomies . . . . .	19
<b>3. Topology Control</b>	<b>23</b>
3.1. Mathematical Primer . . . . .	25
3.1.1. Topology . . . . .	26
3.1.2. Manifolds . . . . .	26
3.1.3. Simplicial Complexes . . . . .	28
3.1.4. General Euler Characteristic . . . . .	30
3.1.5. Homotopy . . . . .	31
3.1.6. The Fundamental Group . . . . .	32
3.1.7. Chains and Circles . . . . .	33
3.1.8. Homology Groups & Betti Numbers . . . . .	35
3.1.9. The Euler-Poincaré Formula . . . . .	36
3.2. Handles and Tunnels . . . . .	37

3.2.1. Previous Work . . . . .	38
3.2.2. Definition & Existence of Handle and Tunnel Loops . . . . .	39
3.2.3. Topological Persistence . . . . .	41
3.2.3.1. Filtration . . . . .	41
3.2.3.2. Incremental Betti Numbers . . . . .	42
3.2.3.3. Persistent Homology . . . . .	44
3.2.4. Computing Geometry-aware Loops . . . . .	46
3.2.4.1. Pairing Algorithm . . . . .	47
3.2.4.2. Refining the Loops . . . . .	49
<b>4. User-Guided Mesh Simplification</b>	<b>51</b>
4.1. The TopStoc Algorithm . . . . .	52
4.1.1. Fast Topological Clustering . . . . .	53
4.1.2. Adaptive Density Control . . . . .	56
4.1.3. Saving Sample Data . . . . .	58
4.2. Simplifying Topology . . . . .	59
4.2.1. Information & Feature Detection . . . . .	59
4.2.2. Mesh Editing . . . . .	62
4.2.3. Limitations . . . . .	64
4.3. Drawing on Meshes . . . . .	65
4.4. Back-face Subsampling & Silhouette Preservation . . . . .	67
4.5. Textures . . . . .	69
4.6. Bounding Boxes & Materials . . . . .	71
4.7. Measurements . . . . .	72
4.7.1. Hausdorff Distance . . . . .	73
4.7.2. Multiview . . . . .	75
<b>5. Conclusion</b>	<b>77</b>
5.1. Summary . . . . .	77
5.2. Future work . . . . .	78
5.2.1. Adaption for Quad-Meshes . . . . .	78
5.2.2. Fast $\beta^1$ Calculation via Decimation & Segmentation . . . . .	79
<b>List of Figures</b>	<b>83</b>

<b>List of Tables</b>	<b>87</b>
<b>Bibliography</b>	<b>89</b>
<b>A. Appendix</b>	<b>i</b>
A.1. Moore's law . . . . .	i
A.2. RenderMen timeline . . . . .	ii
A.3. The Utah Teapot . . . . .	iii
A.4. Proof – Homeomorphism of regular quad meshes and tori . . . . .	iv
A.5. Hausdorff distance . . . . .	v
A.6. The house with two rooms . . . . .	vi
A.7. Proof – Why P2 is not embeddable . . . . .	vii
A.8. Glossary – Topology . . . . .	ix
<b>B. Affidavit / Eidesstattliche Erklärung</b>	<b>xv</b>









---

# 1. Introduction

## 1.1. The Complexity Conundrum

*“Reality is 80 million polygons per second”<sup>1</sup>*

– Alvey Ray Smith [co-founder Pixar, cited in: [Rheingold, 1991](#), p.168]

Computer Graphics, much as related fields in Computer Science, has seen an awe-inspiring rate of progress over the last decades. Yet it seems to be a peculiar fact that advances in Computer Graphics at times appear to be invisible, albeit being often directly translated to prettier pictures and more visceral applications.

Normally at this point most people reference the infamous success story of increased computational power, known as *Moore’s Law*<sup>2</sup>. Less widely understood, but even more remarkable is the fact that in most areas, performance gains due to improvements in algorithms have vastly exceeded the performance gains due to increased processor speed<sup>3</sup>. When asked about the significant advances over the last 20 years *Ed Catmull* explained how sometimes “*we felt it was at an impossible level*” what they were trying to reach in terms of quality, but now years later “*we have exceeded that level in terms of scene and render complexity – we have gone beyond impossible*”<sup>4</sup> [*Edwin „Ed“ Catmull*, president of Walt Disney & Pixar Animation Studios, citet in: [Seymour, 2008](#)].

---

<sup>1</sup> “There used to be this quipped saying back in the early days of 3D rendering. It was a great catchphrase at the time because during those days, it was an obscene number – ridiculously beyond the technology of the day. But nowadays, … , even cheap console systems can crank out that kind of performance. More importantly, today’s technology can render that many polygons in realtime with tons more features than was ever available before, such as programmability or multiple passes. And the high-end keeps getting higher.” [John Carmack, co-founder of id Software, on that very quote, cited in: [Kosak, 2005](#)]

<sup>2</sup> The prediction was initially stated in 1965 by Intel co-founder Gordon Moore and held up with physical law-like precision for the last half-century (for the original reference and its context see Appendix A.1).

<sup>3</sup> It’s difficult to quantify the improvement, though, because it is as much in the realm of quality as of execution time. In the field of numerical algorithms, however, the improvements can be quantified more straightforwardly. One example, provided by Prof. Martin Grötschel: A benchmark production planning model solved using linear programming improved by a factor of roughly 43 million between 1991 and 2008. Of this, a factor of roughly 1,000 was due to increased processor speed, whereas a factor of roughly 43,000 was due to improvements in algorithms [[Holdren et al., 2010](#), cf. p.71].

<sup>4</sup> See Appendix A.2 for a timeline of important technological advances of RednerMan®.

Despite the industries incredible success, *Catmulls* remark about the complexity as being “*beyond impossible*” appears to contradict the notion of super-exponential technology growth in the field of Computer Graphics. For instance, rendering times have been constant<sup>5</sup> for animation films over the last two decades, although algorithms and hardware got so much better during the same time. If the movie business holds as an example<sup>6</sup>, it begs the question: How is this possible?

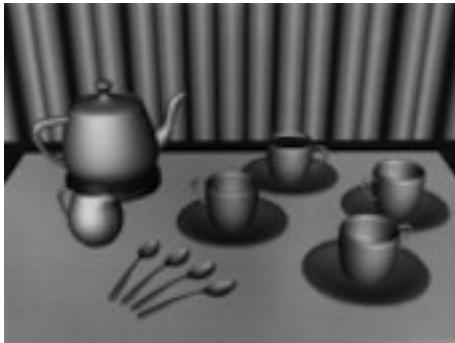


Figure 1.1.: Teapot rendering scene as presented in *Martin Newell's* PhD dissertation, published in 1975 [cf. [Torrence, 2006](#)].

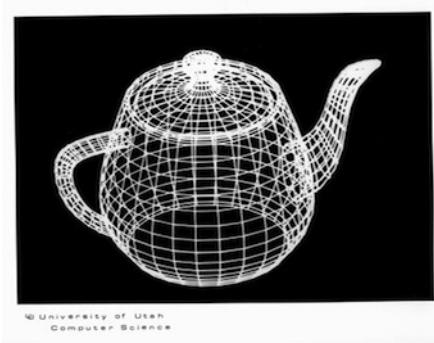


Figure 1.2.: The original wireframe 'Utah Teapot' (28 bicubic Bézier patches, with 16 control points each) [cf. [Newell, 1975a](#)].

The answer is ever increasing complexity! Increasing complexity regarding: Lighting techniques, phenomena that are simulated, and many other domains but most importantly the increased complexity of the assets used for rendering. Every studio struggles with heavier and heavier assets, i.e. the sheer size in geometry information used for models: “*Our typical rendered scenes [for work on Marvels 'The Avengers' (2012)] contain over 140 million triangles across thousands of objects though we have pushed over 1 billion polygons*” [Votch Levi, CTO Whiskytree, cited in: [Seymour, 2012](#)].

<sup>5</sup> For Industrial Light & Magic’s visual effects work on the 1993 film Jurassic Park, typical overnight render times were 4 hours per frame. When the same company worked on the 1997 movie sequel, using both faster computers and more advanced software, frame times were typically an identical 4 hours. Each small speed-up, tends to be used for greater input complexity rather than greater interactivity. While the precise value of the “threshold of pain” for render time varies by institution, this effect has been widely observed. Jim Blinn famously remarked, “All frames take 45 minutes,” hence the effect is known as Blinn’s Law [[Enderton and Wexler, 2011](#), cf. Chapter 3].

<sup>6</sup> Personally, I think that animation and special effects driven movies are vibrant examples for the tremendous progress, that the field of Computer Graphics has seen over the last decades. Although I am sure, other interesting references can be made, most of the examples I will give are related to the *VFX-Industry*. For one because I find them to be compelling and vivid, but more so because it is the one field I have personal experience in, due to my internship at *Weta Digital* in 2011.

To illustrate this hasty evolution, one simply has to look at the stark contrast between the models used for the top-of-the-line renderings of 1975, namely the 'Teapot scene' and a scene from the 2009 movie 'UP', as shown in the pictures 1.2 and 1.4. Whereas all the information needed to describe the models in the 'Teapot scene' could be literally fit on one page of paper<sup>7</sup>, the models currently in production regularly max out in the domain of several giga-bytes of raw data per asset.



Figure 1.3.: Example of highly detailed surface – note that the houndstooth pattern of the jacket is genuine geometric data [Credit: Pixar 2009].

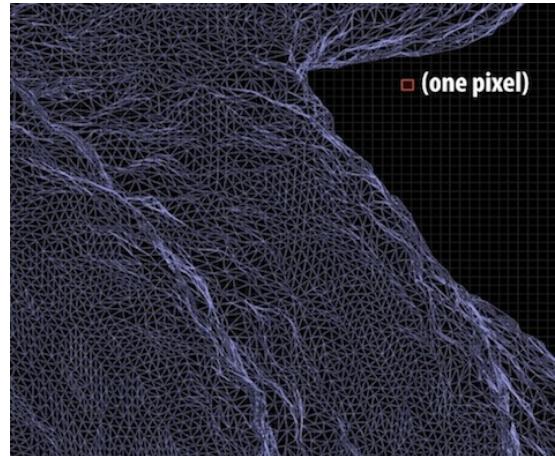


Figure 1.4.: To leave no evidence that surfaces are discretely approximated, film models consist of micropolygons, about a pixel in area [Fatahalian, 2011, p.7].

## 1.2. The Case for Highly Detailed Geometry

Regardless of the factual costs and critical handling issues when working with this kind of scope and complexity, the trend appears to be unbroken<sup>8</sup>. The reason for this is, that in order to render richly detailed and artifact-free scenes, complex models are inevitable. Over the time there have been numerous ingenious strategies to cut corners, such as pre-computed static illumination, bump-, normal- and occlusion-mapping, etc. to compensate

<sup>7</sup> For more information about the original Teapot renderings and a facsimile of the model description, see Appendix A.3.

<sup>8</sup> The importance of highly accurate modeled surfaces has always been an crucial factor for (offline) rendering systems [cf. Cook et al., 1987], but especially with tent-pole movie productions, this paradigm seems to know only one direction – that is – bigger meshes: "the complexity continues to scale on each project we do, the amount we need to render continues to multiply..." – 'Avatar' (2009), was the first movie project that exceed 2 petabyte of storage data [Martin Hill, head of shading at Weta Digital, cited in: NVidia, 2009].

## 1. Introduction

---

for the lack of geometric detail, especially in interactive applications [cf. [Akenine-Möller et al., 2008](#)]. These techniques have comparatively low rendering cost and are used effectively in games. However, they are prone to artifacts.



High-Resolution Triangle Mesh



Low-Resolution Triangle Mesh

Figure 1.5.: Renderings of a stone house using a high-resolution (top) and low-resolution (bottom) triangle mesh (house image by Unigine Engine).

The stone house pictured in 1.5 features many complex surfaces like bumpy cornerstones and curved red roof-tiles. Accurate shadows (self-shadowing) are cast and self occlusion can be seen. These details can be rendered because the house is represented using a high-resolution triangle mesh. The lower image, uses the same texturing and soft shadowing techniques, but with scene geometry represented using a low resolution triangle mesh. The

model holds up quite well, but unfortunately, the illusion breaks down eventually. Especially along the house's silhouettes, artifacts are spotted easily. Shadows are rendered but on close examination they also give away the flat nature of the low resolution stand-in. Similarly, the roof tiles and windows on the left side of the image appear distorted. These artifacts become even more noticeable when objects move [cf. [Fatahalian, 2011](#), p.2-4]. The industry will not cease its pursuit for visual brilliance and together with the growing trend for more physically correct rendering, the polygon counts will be constantly pushed for the next years to come. With this scaling of complexity also scales the need for tools that facilitate the work with the created data<sup>9</sup>. Thus robust and capable techniques are essential and at the forefront of fundamental techniques for geometry processing is mesh simplification.

### 1.3. Reducing Complexity

Mesh decimation or simplification is both, a very current, and a very old idea in computer graphics. As early as 1976 *James Clark* described the benefits of representing objects within a scene at several resolutions [cf. [Clark, 1976](#)].

Dozens and dozens of simplification algorithms have been developed in the meantime. Even a number of excellent surveys reviewing the field of polygonal simplification have been published<sup>10</sup>. Yet, no algorithm today excels at simplifying all models equally good. Some approaches best suit curved organic forms, while others work best at preserving hard planar objects with sharp corners, flat faces, and regular curves. This is probably due to the fact that, we still have limited understanding about what really determines perceptual fidelity. We do have a good understand of geometric and volumetric fidelity but as it turns out, the supposedly easy question: Does the simplification look good? Is hard to answer [[Luebke, 2001](#), cf. p.25].

At this point it seems that the research community tends to be of the opinion, that until the question of perceptual metrics is solved, the research area of mesh simplification is in

---

<sup>9</sup> During my internship I personally asked a modeler, at what point they would normally stop adding geometric detail. He noted that everyones biggest fear is to get ones' work rejected because it does not hold up to a supervisors expectations. So what they typically will do is to work until their tools – their computers and software – give in and make it unbearable to work any further.

<sup>10</sup> For example, Cignoni et al. supplied comparative performance statistics for over 30 different decimation strategies [cf. [Cignoni et al., 1998b](#); [Heckbert and Garland, 1997](#)].

## 1. Introduction

---

a state of arrested development<sup>11</sup>. As much as we agree that it would be fantastic to get a sound scientific grasp of human perception, we reject the notion that exciting research topics aren't to be found.

At the beginning of most journal papers concerning mesh decimation, the same emblematic application examples are listed: meshes generated by marching-cubes, CAD/CAM models, oversampled 3D scan data, etc. – but with the advent of image based automatic geometry capture techniques and their impressive results [cf. [Pollefeys et al., 2004](#); [Snavely, 2008](#)], arguably the scope of the field has changed<sup>12</sup>. Not only is there a demand for highly detailed meshes but also a plethora of source material to work with.

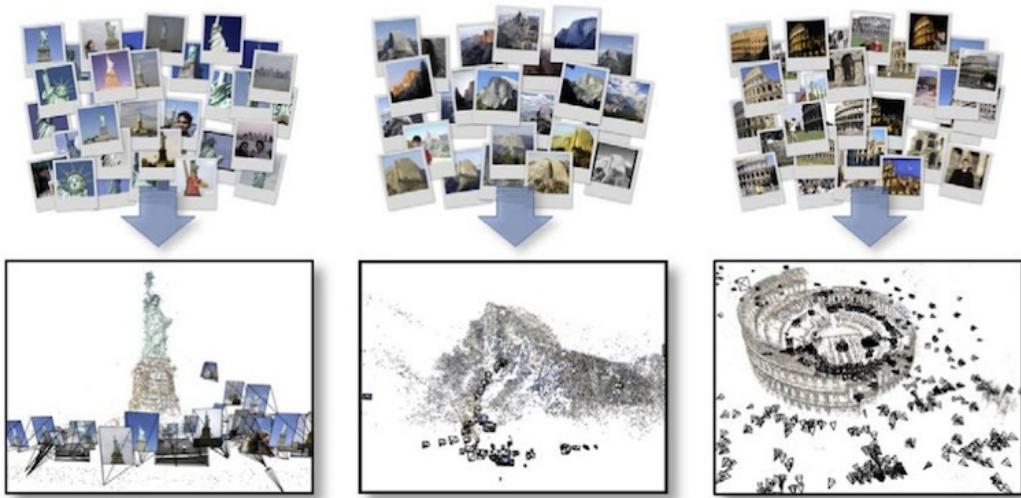


Figure 1.6.: 3D mesh reconstructions from Internet photo collections, that typically can have dozens of millions of triangles [[Snavely, 2008](#), p.6].

Still, as of lately, the research community does not emphasise on mesh decimation. One of the reason for this, is the set of expectations that govern academic zeitgeist. For example, “good” solutions are considered to be solutions that have been completely automated

<sup>11</sup> Only few ameliorations have been proposed, mostly at the cost of near to prohibitively expense execution times, like Cohen’s “Appearance-Preserving Simplification” [cf. [Cohen et al., 1998](#)] and Lindstrom’s “Image-Driven Simplification” [cf. [Lindstrom and Turk, 2000](#)] – both of which aiming at per-pixel comparisons of multiple rendered views as “weights” or upper bounds for the priority queue of an incremental decimation process.

For an comprehensive discussion about the difficulties faced when trying to define perceptually salient descriptors see “Perceptual metrics for static and dynamic triangle meshes” [cf. [Corsini et al., 2012](#)], respectively “Perceptually Driven Simplification for Interactive Rendering” for further information on psychophysical models of visual perception [[Luebke and Hallen, 2001](#)].

<sup>12</sup> Another branch of research that will affect the evolution of geometry processing tools in a profound way is 3D printing and rapid prototyping – even if only for the scale of deployment, once 3D printing is widely adopted [cf. [Vilbrandt et al., 2008](#); [Bickel et al., 2010](#)].

with little or none required input from the user. However there is much to be gained and researched, once the user is seen as integral part of the process.

*“Decimation of generated models, often reduces the precision ... and while the generated meshes may have the correct level of detail, it is not represented in a structured form, usable in subsequent stages ... In many cases it would be more useful to rely on user input to generate a simplified, yet structured mesh that can be further manipulated.”* [Sylwan, 2011, p.195]

Bringing together the arguments of the last pages, the important question to ask now is, what kind of requirements should a modern mesh decimation algorithm focus on?

## 1.4. Design Principles & Workflow Regimes

The first step in deciding upon the design features of an application that simplifies meshes, is to acknowledge that there will never be one perfect solution to fit all possible scenarios. Not only because there are always trade-offs to be made in engineering, but because there are conflicts of objectives that oppose each other diametrically. For instance, it is not possible to ensure quasi ‘Delaunay-Triangulation’ in order to cater for the needs of a finite elements simulation and at the same time aggressively reduce the vertex count without loosing geometric detail<sup>13</sup>.

Instead, it shows more promise to put emphasis on a few selected, crucial aspects. Thus our work focuses on two major principles that were decided beforehand:

- Topological control of the decimation process.
- Artistic freedom by the means of user-guidance.

As for the first aspect – topology control – most algorithms can ensure the preservation of the topological genus of a given model<sup>14</sup>, but normally topological features by themselves are not driving criteria for decimation. Accounting for this poses some interesting research questions that have not yet been answered satisfactorily. It is also the more theoretical and abstract part of the thesis.

---

<sup>13</sup> Arguably it will be always a choice of just two out of the following three: good triangle shape / high decimation rates / optical fidelity – and many other examples for such conflicts can be found.

<sup>14</sup> With the notably exception of the class of vertex clustering algorithms, that by design, are agnostic to topological changes [cf. the original paper by Rossignac and Borrel, 1993].

The second aspect – artistic freedom, i.e. user-guidance – has some further ramifications that need to be pointed out. Unquestionably it appears to be a good thing to give additional control to the user. Not only can the hard problem of perceptual fidelity be handed over to the one instance best suited for the decision, but also there can be direct feedback incorporated to ensure the ideal outcome.

*“Often researchers are biased towards a fully automated solution or assume the user interaction needs to be targeted to a novice user.”* [Sylwan, 2011, p.195]

Just adopting some measure of control not automatically yields better results. In order to leverage the outlined advantages, the entire system has to meet the needs of direct user interaction. There is a balance between interaction and computation.

If a process gives perfectly accurate results automatically, it is a candidate to run unsupervised in an multiply hour offline scenario. Whereas an algorithm that requires artist intervention, must run relatively quickly so artists get immediate feedback and iterate on it [cf. Hillman et al., 2010, p.20]. This sounds like an obvious observation, but research has shown, that missing certain levels of interactivity, ultimately decides upon how a tool is viewed as a whole and thereby determines the success and adoption rate of this technology [cf. Enderton and Wexler, 2011]. In summary, what we want is a very fast, hence smoothly interacting application that gives good results but invites the user to tweak any aspect of it until the best possible outcome is reached. This is in short is what this thesis’ goal is!

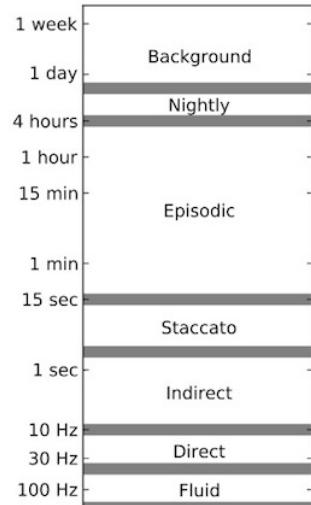


Figure 1.7.: Response time to user input vs. workflow.

*“A workflow regime is defined as a range of system response times in which the artist’s relationship to the task is qualitatively similar ... technology is much more likely to succeed when it brings an artist’s workflow into a new regime ... we feel physically connected, albeit indirectly. This connection is critical for focused and rapid task achievement.”* [Enderton and Wexler, 2011, p.2]

## 1.5. The Outline of this Thesis

The introduction started with a general observations on trends: The complexity from the point of view of Computer Graphics has increased dramatically over the last decades and is still on the rise. This is especially true for geometric complexity, since even the best short-cuts suffer from undesired artifacts. With models getting more and more detailed, the need for good tools that can simplify meshes likewise increases. But instead of adding a new algorithm to the standard repertoire, we propose to focus on adopting techniques for user-guidance and topology control. After this widely argument for our motivation, the rest of the thesis is structured in a straightforward manner.

The next section, chapter 2, will very briefly discuss the historical development of the field of mesh decimation. Not so much in order to give an exact overall account, but rather to give an overview of the principle techniques and the ideas behind them. Bringing these ideas together we will try to formulate a taxonomy that structures the known approaches. This is followed by chapter 3, that functions as a primer for mathematical concepts which are crucial for the understanding and control of topology. Retrieving correct topology information of a mesh and controlling it, is one of the key contributions of our work and although the comprehension of the involved mathematics is important, we try to not make it a mandatory prerequisite for the results section of the text.

Chapter 4 finally details the contribution of our work in three sections. Firstly some minor adaptions to the 'TopStoc' algorithm that we used as the foundation in section 4.1. Then in 4.2 we deal with the topological aspects, namely computing the precise Betti numbers for arbitrary meshes. Finding all topological relevant features such as handles and tunnels, which is achieved by the means of gathering special groups of homologous edge loops. And of course the ability to cut and delete parts of the mesh as well as mending boundaries. Section 4.3 finally describes the tools that give artistic freedom to the artist. Among these are sketch tools to define resolution, point of view constraints for decimation plus visual feedback in form of Hausdorff visualizations and multiple instances of decimation results to easily make side-by-side comparisons.

The last part, chapter 5, then summarizes the work and gives an outlook on further research topics.



---

## 2. The Concept and History of Mesh Simplification

### 2.1. Problem Statement

*“The essence of mathematics is ... to make complicated things simple.”*

– Stan Gudder

Before considering any concrete approaches or talking about historical developments of mesh simplification, we want to give a formal definition of the problem. For a given mesh  $\mathcal{M} = \{\mathcal{V}, \mathcal{F}\}$  where  $\mathcal{V}$  is the list of vertices and  $\mathcal{F}$  the list of faces indexed over  $\mathcal{V}$ , we search for  $\mathcal{M}' = \{\mathcal{V}', \mathcal{F}'\} \subset \mathcal{M}$  so that either:  $|\mathcal{V}'| < |\mathcal{V}|$  with  $|\mathcal{V}'| \in [n_{min}, n_{max}]$ , and  $\|\mathcal{M} - \mathcal{M}'\|$  is minimal – the so called budget-based approach, or  $\|\mathcal{M} - \mathcal{M}'\| < \epsilon$  and  $|\mathcal{V}'|$  is minimal – a fidelity-based simplification, led by some measure of difference  $\epsilon$ . Most of the time this is done together with further constraints like globally enforced fairness criteria<sup>15</sup> to evaluate the quality of the mesh  $\mathcal{M}'$ : *“the use of a fairness oracle turns the plain downhill decision of the greedy algorithm into an ‘educated guess’”* [Kobbelt et al., 1998, p.46]. This means that mesh simplification describes a class of algorithms that transform a given polygonal mesh into another with fewer faces, edges and vertices. Thus a mesh simplification scheme can be viewed as a decomposition operator to obtain a low frequency component  $\mathcal{M}'$  and a high frequency component, i.e. the difference  $(\mathcal{M} - \mathcal{M}')$ . The decimation process is usually aimed at a specific target size and controlled by a set of user-defined quality measures that aim to preserve specific properties and salient features of the original mesh as much as possible [Shene, 2005, cf. pp.9-10].

---

<sup>15</sup> What is typically used, are geometrical analogies of the first and second fundamental form of differential geometry, transferred to the discrete setting of triangle meshes. For example [cf. Schröder, 2003]:

Order 0: distance vertex and plane, parametric or geometric distance

Order 1: local distortion, triangle shape or inradius

Order 2: local or mean curvature, sum dihedral angles

No order: e.g. valence balance or scalar attributes that can account for other arbitrary restrictions

## 2.2. Chronology of Simplification Designs

The idea to use different resolutions of the same model can be traced back, as far as 1976 [cf. [Clark, 1976](#)<sup>16</sup>] but at the time all low-resolution proxies of models were created manually. Only in the early 1990s the first algorithms to automate this process appeared. The new field instantly spiked interest in the research community and a flurry of papers were published in the following years. Some of those first algorithms, such as vertex decimation [[Schroeder et al., 1992](#)] and vertex clustering [[Rossignac and Borrel, 1993](#)], remain viable and useful solutions until today.

Today there is an abundance of designs for algorithms to choose from, literally hundreds of different solutions for mesh decimation. Therefore we can not and do not want to discuss every single algorithm ever conceived, but instead we give a chronology of very influential works with a short description of the idea behind it. Thereby showing the numerous ways in which decimation can be done. For a more detailed discussion of the history of the field, see [[Díaz-Goano, 1998](#), cf. p.4 ff.] and [[Luebke et al., 2002](#), cf. p.8 ff.]. Almost every paper today can be referenced to one of the following works or rather these designs:

- Selectively removing vertices [cf. [Schroeder et al., 1992](#)].

The algorithm makes multiple passes over the set of vertices of the triangle mesh and removes vertices according to predefined criteria. As a result of a removal a hole is created which a triangulation process then fills<sup>17</sup> (also see figure 2.1).

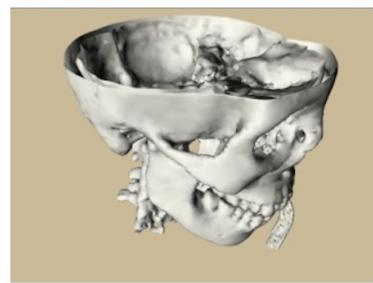
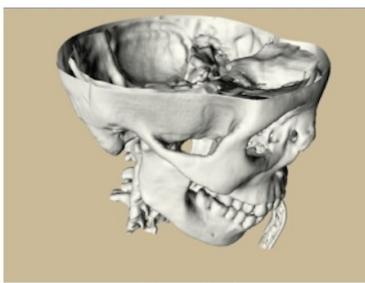


Figure 2.1.: Example from the original paper, left full resolution (569k triangles) and right the 90% decimated version (142k triangles) [[Schroeder et al., 1992](#), p.68].

<sup>16</sup> Clark's paper "Hierarchical Geometric Models for Visible Surface Algorithms" also described hierarchical scene graph structures and by todays standards mundane techniques, such as view-frustum culling and out-of-core simplification. Nevertheless it is interesting to read for its value as a foundational paper of Computer Graphics.

<sup>17</sup> The number of ways to triangulate a regular polygon with  $n + 2$  sides, is given by the Catalan numbers. One can think of choosing from among these possible ways as a discrete optimization problem:

$$C_n = \frac{1}{n+1} \binom{2n}{n} = \frac{(2n)!}{(n+1)! n!} = \prod_{k=2}^n \frac{n+k}{k} \text{ for } n \in \mathbb{N}$$

- Redistributing vertices over a surface [cf. [Turk, 1992](#)].

The idea is to reduce the number of polygons by adding new points on the surface of the model, connecting these new vertices into triangles forming a mutual tessellation. Thus creating an intermediate polygonal surface that incorporates both the old vertices and the newly placed ones and then deleting the old vertices, i.e. re-tiling the surface. In order to maintain the features of the surface accurately, the point distribution is modified so that more new vertices are placed in regions of higher curvature (also see figure 2.2).

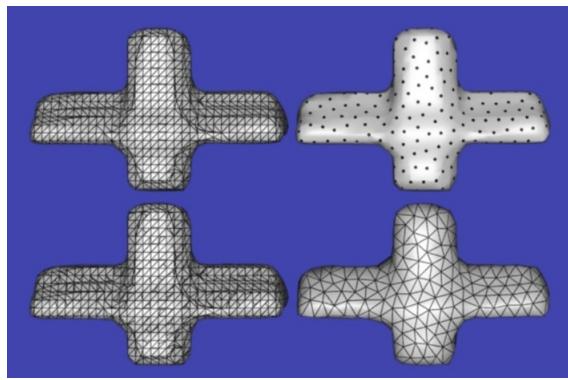


Figure 2.2.: Re-tiling of a radiation iso-dose surface. Upper left: Original surface. Upper right: Candidate vertices after point-repulsion. Lower left: Mutual tessellation. Lower right: Final tessellation [[Turk, 1992](#), p.58].

- Clustering vertices [cf. [Rossignac and Borrel, 1993](#)].

Is done by merging vertices based on their spatial proximity. All vertices within each cell of the subdivided 3D are collapsed into a single representative vertex for the voxel. Hence the resolution of the grid determines the level of detail achieved. The

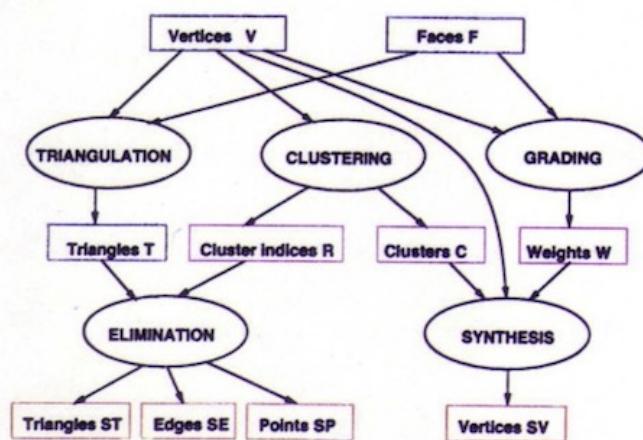


Figure 2.3.: Overview of the simplification process [[Rossignac and Borrel, 1993](#), p.458].

algorithm is very robust as it can handle degenerated models, but it is sensitive to the orientation of the clustering grid. It can be implemented quite efficiently. Unfortunately topology is not preserved, and it is very hard to give guaranteed explicit error bounds (also see figure 2.3).

- Merging nearly coplanar polygons [cf. [Hinkler, 1993](#)].

This method obtains impressive reductions for large models with nearly flat surfaces. It identifies coplanar or nearly coplanar polygons, merging them together into a larger complex polygon and finally re-triangulating them into fewer polygons than the original description. However, this approach offers only small gains for surfaces of high curvature (also see figure 2.4).

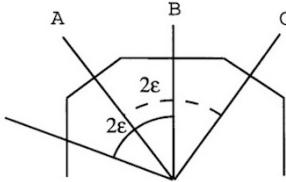


Figure 2.4.:  $2\epsilon$  is the user specified angle describing the maximum angular difference between coplanar sets selected to be merged [[Hinkler, 1993](#), p.192].

- Removing polygons in order of curvature [cf. [Hamann, 1994](#)].

Given a surface triangulation, each triangle is weighted according to the principal curvature at its vertices. These curvature values are pre-computed based on a parametric representation of the surface. A triangle is associated with a surface region of low curvature if the sum of the absolute curvatures at its vertices is low. The lower the curvature value at the vertices of a triangle the lower its weight. The triangle with lowest weight is removed. The region affected by the removal is re-triangulated and new weights are computed for all affected triangles (also see figure 2.5).

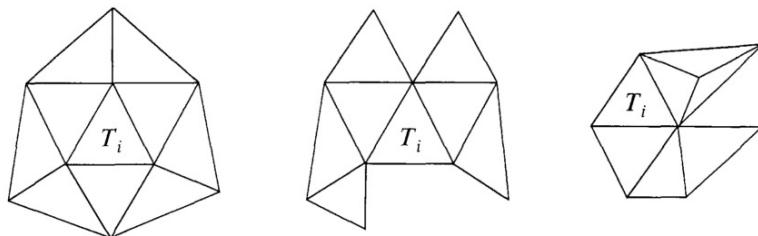


Figure 2.5.: The different cases for a triangle platelet  $P_i$ , i.e. the data stencil in the surface triangulation affected by the removal of triangle  $T_i$  – platelets with cyclic, disconnected and connected corona [[Hamann, 1994](#), p.200].

→ Using Wavelet theory [cf. [Eck et al., 1995](#)].

A mesh is converted into a sample mesh together with a sequence of local correction terms, i.e. wavelet coefficients that capture the detail present in the model at various resolutions. The algorithm uses enough wavelet coefficients to meet a user specified error bound. An important contribution of this method is that it overcomes subdivision connectivity restrictions. It constructs a continuous parametrization of an arbitrary mesh over a simple domain mesh, thus all arbitrary meshes can be converted to multiresolution form (also see figure [2.6](#)).

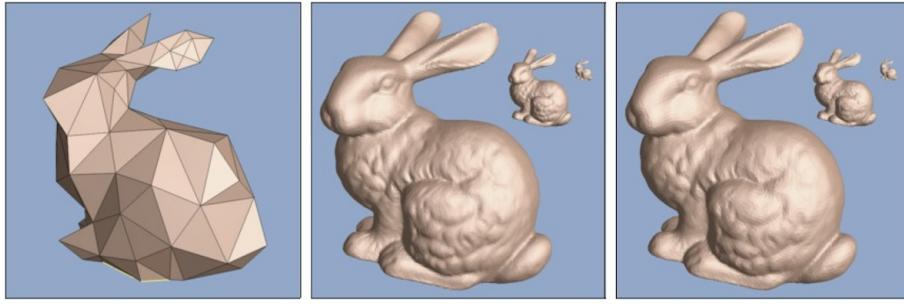


Figure 2.6.: Left: Base mesh (162 triangles). Middle: Original mesh (69,437 triangles). LOD using multiresolution approximation [[Eck et al., 1995](#), p.181].

→ Minimizing an energy function [cf. [Hoppe, 1996](#)].

Hoppe presents a framework, called 'Progressive Meshes', which consist of a base mesh created by a sequence of edge collapses and vertex splits together with a sequence of detailed records that indicate how to incrementally refine the base mesh exactly back to the original mesh. The sequence of collapses during a simplification

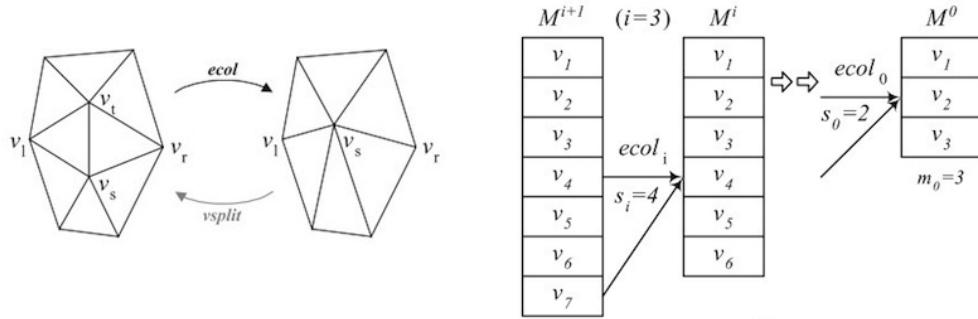


Figure 2.7.: Original illustration of the edge collapse transformation respectively the vertex split and the resulting vertex correspondences [[Hoppe, 1996](#), p.100].

is determined explicitly as an energy function to be minimized. All edges to be collapsed are evaluated according to the energy function and sorted in a priority queue.

'Progressive Meshes' also introduced a very efficient data structure that can be used to represent a triangle mesh at multiple levels of detail, i.e. progressive refinement (also see figure 2.7). This technique has been widely used for view-dependent simplifications, as well as progressive transmissions [cf. [Bajaj et al., 1999](#)].

- Using bounding surfaces [cf. [Cohen et al., 1996](#)].

A proposed method called 'Simplification envelopes'. The approach guarantees an error bound so that all points of an approximation are within a user-specifiable distance from the original model, also it enforces global and local topology preservation. Simplification envelopes of a surface consists of two offset surfaces. The outer and inner envelope are generated by displacing each vertex of the original mesh along its normal by a defined distance  $\pm\epsilon$ , which then guides the process. A strong suit of this method is that it can gracefully deal with sharp edges (also see figure 2.8).

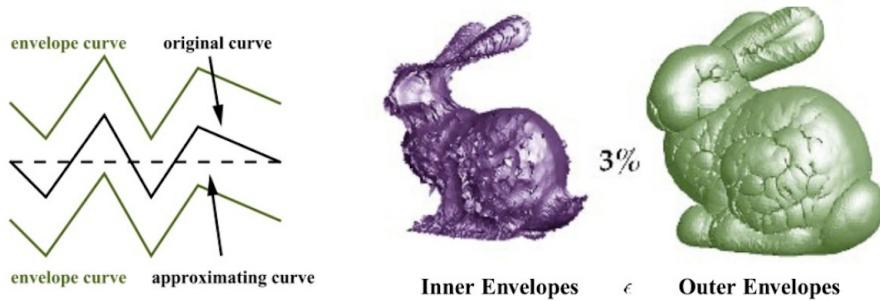


Figure 2.8.: Approximation with inner & outer envelopes [[Cohen et al., 1996](#), pp.123-124].

- Quadric Error Metrics [cf. [Garland and Heckbert, 1997](#)].

The algorithm uses iterative contractions of vertex pairs to simplify models and maintains surface error approximations using quadric matrices. By contracting arbitrary vertex pairs and not just edges, the algorithm can change the topology of the mesh and does not need a manifold mesh to begin with. As the algorithm proceeds, a geometric error approximation is maintained at each vertex  $\mathbf{v}$  of the decimated model:  $\Delta(\mathbf{v}) = \mathbf{v}^T \mathbf{Q} \mathbf{v}$ , for a given contraction  $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \mathbf{v}_{12}$  the new matrix  $\mathbf{Q}_{12}$  can be easily derived by simply adding  $\mathbf{Q}_1 + \mathbf{Q}_2 = \mathbf{Q}_{12}$ . Simplification with quadric error metrics possesses much of the generality of vertex clustering as well as the quality and control of iterative contraction algorithms. The only major downside is its relative slowness compared to vertex clustering (also see figure 2.9).



Figure 2.9.: 98,5% decimation of the stanford bunny. The error ellipsoids for each vertex are shown in green [Garland and Heckbert, 1997, p.215].

→ Vertex placement for collapsed edges [cf. [Lindstrom and Turk, 1998](#)].

Using edge collapse as the method for simplifying, edges are incrementally replaced with a single vertex. The new vertex is placed so it preserves the location and shape, and does not require a retriangulation. The decision of how to position and order the edges builds upon volume and surface information. Since these conditions do not fully determine the decimation process, an optimization stage is added. Per triangle volume and area differences are used to create the priority queue (also see figure [2.10](#)).

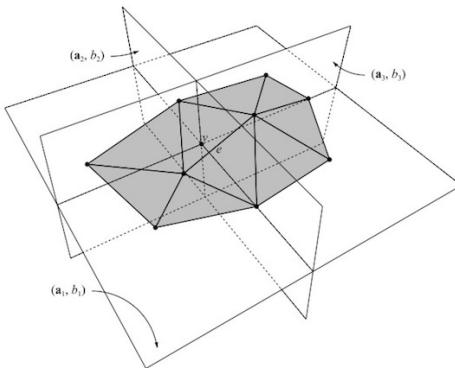


Figure 2.10.: The optimal vertex position  $v$  expressed as the intersection of three planes that ensure volume preservation [Lindstrom and Turk, 1998, p.284].

→ User-assistance for mesh simplification [cf. [Cignoni et al., 1998a](#)].

The first system that takes advantage of users-guidance for simplification. “Zeta: A Resolution Modeling System”, takes a precomputed sequence of simplifications as input, and utilizes an layer of abstraction to work with different local mesh reduction schemes. Especially noteworthy is that the paper explicitly mentions response times and the role of interactivity (also see figure [2.11](#)).

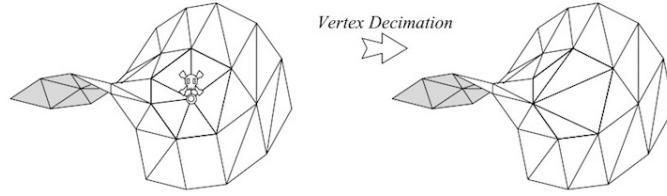


Figure 2.11.: Explicit deletion of a vertex on the mesh [Cignoni et al., 1998a, p.312].

→ Using image compression for geometry [cf. Gu et al., 2002].

Most of the time surface geometry is modeled with irregular triangle meshes, i.e. not constant valency. In their paper, the authors devise a method to remesh an arbitrary surface onto a completely regular structure they call a geometry image, enabling them to represent geometry as a simple 2D array of quantized points. Surface signals like normals and colors are stored in similar 2D arrays using the same implicit surface parametrization. To create this geometry image, the mesh is cut along a network of edge paths, and the resulting single chart is parametrized onto a square<sup>18</sup>. The ingenious twist is, that in this form geometry images can be encoded using traditional lossy image compression algorithms, such as JPEG. A compelling example of how lifting one problem into another domain can lead to unexpected connections and new ideas (also see figure 2.12).

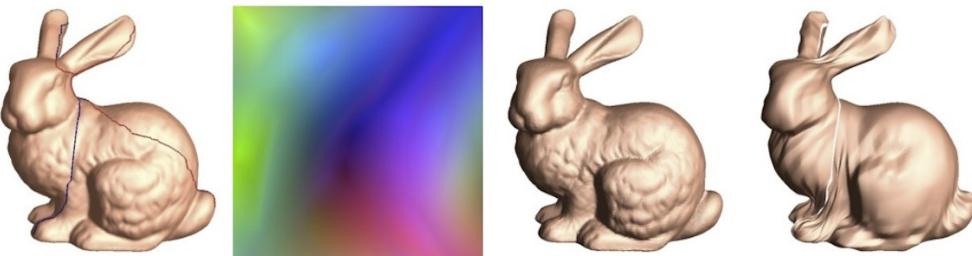


Figure 2.12.: Left: Original mesh and its geometry image. Right: Reconstruction of the original and a 1,5kB compressed version mesh [Gu et al., 2002, p.356].

This section can not give an exhaustive nor overall account of mesh simplification, but hopefully by example it illustrated the richness of the solutions and the progress already made. The next section will try to bring together the various ideas presented here in order to give a general frame of reference, that is to say principles for taxonomies.

<sup>18</sup> More precisely the geometry is mapped onto a torus, since it is the only 3D manifold that can be a completely regular quad mesh and accordingly a regular triangle mesh with valence 6. With two cuts the torus then is equivalent to a 2D array (for a proof of this statement see Appendix A.4).

## 2.3. Taxonomies

Besides listing important ideas and trying to get a feel for viable solutions, there is a second important step to get a firm grasp of a topic. Namely by aggregating the ideas in question, grouping the similar ones and putting them into a relational structure, i.e. building a taxonomy. This way, one does not only inductively broaden his knowledge one case at a time, but also deductively establish an overall framework for the principle problems in question. Good taxonomies give the impression of necessity, as insofar as the taxa used to create the divides seem to be without alternatives. Sometimes this happens at the cost of blocking the sight for alternatives<sup>19</sup>.

Most taxonomies concerning mesh simplification, that can be found today, seem to be derived from the seminal work of Kobbelt [Gotsman et al., 2002, cf. p.6]. As it were, the prototypical taxonomy is as follows [Shene, 2005, cf. p.11]:

- **Vertex Clustering** – Usually very efficient, robust and typically with linear complexity:  $\mathcal{O}(|\mathcal{V}|)$ . However, the quality of the resulting mesh is not always satisfactory and it is hard to guarantee the preservation of topological features.
- **Incremental Decimation** – Delivers higher quality meshes in most cases and can take arbitrary user-defined criteria into account, according to how the next removal operation is chosen. Though, complexity is generally higher:  $\mathcal{O}(|\mathcal{V}| \log_2 |\mathcal{V}|)$ , and can go up to  $\mathcal{O}(|\mathcal{V}|^2)$  especially when a global threshold has to be met.
- **Resampling** – Most general approach. The major motivation for building a completely new mesh is to maintain a special connectivity structure like subdivision connectivity. Problematic is its higher proneness to aliasing errors.

---

<sup>19</sup> It is worth mentioning that taxonomies are normally presented as prefaces to discussions of individual solutions. By doing this, the view on the actual evolution gets biased. Research is a fundamentally inductive endeavor.

*“The view is often defended that sciences should be built up on clear and sharply defined basal concepts. In actual fact no science, not even the most exact, begins with such definitions. The true beginning of scientific activity consists rather in describing phenomena [and coming up with heuristics, only then] to group, classify and correlate them.”* [cf. Freud, 1963]

The three cases are obviously very broad and each of them can be further characterized by classifying more details. For example, 'Vertex Clustering' algorithms can be sorted whether they apply a hierarchical or top-down approach, or how they compute representative vertices: average, median, error quadrics and so forth. Likewise 'Incremental Decimation' algorithms can be organized by the decimation operators they use: Topology-changing vs. topology-preserving, subsampling vs. filtering, existence of inverse operations, etc. – for an exhaustive discussion of these aspects, we recommend the paper “A General Framework for Mesh Decimation” [cf. [Kobbelt et al., 1998](#)].

The standard taxonomy is practical and easy to understand but it does not seem to follow any specific principle or foundation. This is especially obvious when confronted with algorithms that overstep the traditional scheme – such as probabilistic optimization techniques of multiple-choice algorithms [cf. [Wu and Kobbelt, 2002](#)], or the 'TopStoc' algorithm that will be discussed in detail later [[Boubekeur and Alexa, 2009](#), cf.].

	Clustering	Incremental
Deterministic	Spatial clustering	Priority queue
Stochastic	TopStoc	Multiple choice

Table 2.1.: Taxonomy from Boubekeur & Alexa [[Boubekeur and Alexa, 2009](#), p.2].

A more general classification can be set up when comparing mesh simplification to a method from signal processing, that is, vector quantization. Vector quantization has traditionally been used for lossy data compression, but has also applications for lossy data correction and density estimation. The following description adopts the ideas described in the paper “Model Simplification Through Refinement”, which first made the argument of familiarity [[Brodsky and Watson, 2000](#), cf. pp.221-222]:

Vector quantization is the process of mapping a vector in a large set  $\mathcal{S} \subset \mathbb{R}^n$  into a smaller set  $\mathcal{T} \subset \mathbb{R}^n$  with  $\mathcal{T}$  partitioning the set  $\mathcal{S}$ , hence  $|\mathcal{S}| > |\mathcal{T}|$ . This is done by a so called quantizer function  $\mathcal{Q} : \mathbb{R}^n \rightarrow \mathcal{T}$ . Where  $\mathcal{T} = \{\vec{v}_i \in \mathbb{R}^n | 1 \leq i \leq N\}$  represents all vectors in  $\mathcal{S} \subset \mathbb{R}^n$ . The goal is to find the best  $\mathcal{T}$  to represent all vectors in  $\mathcal{S}$ . When the distortion between an input vector is minimal for all  $\vec{v} \in \mathcal{S}$ , then  $\mathcal{T}$  is called optimal.

This definition looks familiar to the one given at the beginning of the chapter [2.1](#), with:  $\mathcal{S} \propto \mathcal{M}$  the base mesh, and  $\mathcal{T} \propto \mathcal{M}'$  the decimated mesh, respectively  $\mathcal{Q}$  representing a

decimation algorithm. Gersho and Gray list the four basic types for vector quantization algorithms [Gersho and Gray, 1991, cf. pp.358 ff.]:

1. So called '**product code**' algorithms use scalar quantizers that are independently applied to each input vector element. The main difference to other methods is, that the partitioning set  $\mathcal{T}$  is built once and not iterated upon.

An analogy for this type are clustering algorithms, where partitions are formed with a uniform voxelization.

2. In '**pruning algorithms**',  $\mathcal{T}$  initially contains all the vectors of the input set  $\mathcal{S}$ . The entry  $\vec{v}_{mindis} \in \mathcal{T}$  that increases distortion least is removed and removals continue until the desired size of  $\mathcal{T}$  reached.

Simplification algorithms taking the pruning approach are for instance algorithms, that growing coplanar patches, or that remove and pruning away single vertices.

3. Pairwise '**nearest neighbor**' algorithms also set  $\mathcal{T}$  to contain all the vectors in  $\mathcal{S}$ . All possible pairs  $(\vec{v}_i, \vec{v}_j) \in \mathcal{T}$  with  $i \neq j$  are considered and the pair that introduces the least distortion is merged. Merging continues until the desired size for  $\mathcal{T}$  or distortion tolerance is reached.

Vertex merge or edge collapse algorithms are of this kind. They merge the vertices, recompute the affected vertex pairs, and iterate.

4. In '**splitting algorithms**',  $\mathcal{T}$  initially contains a single partition of  $\mathcal{S}$ . The partition with the most distortion is located and then split. Splitting continues until the required distortion or size is reached.

Only very few actual decimation techniques use this quasi inverse method of starting with a base primitive and then adaptively subdividing it until the desired decimation level is reached.

In other words, the logic of the classification is to look at possible dynamics for partitioning, since there are just three principle ways to form  $\mathcal{T}$ : One can set it statically ('**product code**'), or produce it step by step, either via a top-down ('**pruning algorithms**' and '**nearest neighbor**') or bottom-up approach ('**splitting algorithms**').

Besides having a complete taxonomy, the last point to discuss is, upon which differentiations any simplification algorithm should be distinguished and designed. We compiled the following checklist to catalog the most important attributes to characterize any mesh

decimation algorithm. Thereby representing the question one should answer before implementing code in order to ensure to end up with an effective and not only an efficient solution<sup>20</sup>:

The decimation process is ...	<b>automated</b>	↔	<b>with user input</b>
The result is measured ...	<b>perceptual</b>	↔	<b>analytical</b>
The algorithm works ...	<b>deterministic</b>	↔	<b>stochastic</b>
	<b>incremental</b>	↔	<b>static</b>
The input data is .....	<b>constrained</b>	↔	<b>arbitrary</b>
... and must be topologically ...	<b>transformed</b>	↔	<b>preserved</b>
The work-flow regime is ...	<b>off-line</b>	↔	<b>realtime</b>

Table 2.2.: Checklist for simplification algorithms.

We have ordered the checklist in so far, as the the work presented in this thesis, aims to meet the description on the right hand side of table 2.3: We use a very fast, user-guided decimation technique with topology analyzation and control, that can handle any manifold triangle mesh. In addition to that, our approach addresses a number of problems that are pivotal for a practical solution: Texture handling, reproducibility, etc.

Before describing the details of our solution and discussing the results we achieved, the next chapter 3 will set a solid mathematical foundation for describing the topological challenges we encountered during our work.

---

<sup>20</sup> In economics the neglectance of tackling this issue is known under the name of: “The efficiency trap”.

---

### 3. Topology Control

*“If it’s just turning the crank it’s algebra, but if it’s got an idea in it, it’s topology.”*

– Solomon Lefschetz

Topology began with the study of curves, surfaces and other objects in the plane and 3D space. One of the central ideas in topology is that spatial objects like circles and spheres can be treated as objects in their own right, and knowledge of objects is independent of how they are embedded in space. Topology can be used to abstract the inherent connectivity of objects while ignoring their detailed form.

For example, the figures shown in picture 3.1 illustrate the connectivity of a number of topologically distinct surfaces and at the same time links them to a common denominator – solid parallel edges join one another with the orientation indicated by arrows, dashed lines show edges that remain free [Weisstein, 2012, cf. p.1]:

00

Figure 3.1.: Connectivities of prominent 2D manifolds.

Although topology is a comparatively young field in the history of mathematics, it would be futile trying to give a comprehensive explanation of the entire field<sup>21</sup>.

For our purposes we split this chapter in two parts, firstly we introduce the most important definitions in section 3.1. Many of the presented concepts will be already known, at least in an intuitive way, but in order to proceed to more demanding theories, it is necessary to formulate them in an concise and unifying manner. Key ingredient is the notion of ‘Simplicial Complexes’ in 3.1.3, a powerful abstraction for spaces with triangulations, subsequently constructed by gluing together series of lower-dimensional simplices.

---

<sup>21</sup> The beginnings of Topology are closely related to advances in set theory at the end of the 19<sup>th</sup> century. An iconic date for the commencement is the formal definition of metric spaces by Maurice Fréchet in 1906. Today topology can be divided into algebraic topology (which includes combinatorial topology), differential topology, geometric topology and low-dimensional topology known as point-set topology [Weisstein, 2012, cf. p.2].

The second chapter 3.2 then will introduce a set of mathematical tools to describe and compute geometric features like handles and tunnels in a precise and rigid way, see figure ???. As a means to do this, it will be necessary to introduce advanced theories like 'Filtration' and 'Topological Persistence', first described by Edelsbrunner [cf. [Edelsbrunner et al., 2000](#)]. Some of the definitions might seem overly abstract at times, but as it turns out, describing and finding handles or tunnels in a mathematically sound manner is a challenging task.

What we ultimately want to achieve, by mobilizing all the mathematical machinery, is best expressed by the relation expressed in equation 3.18:

$$\chi = \underbrace{\sum (-1)^n \beta^n}_{Euler-Poincaré} \quad \rightarrow \quad \underbrace{\beta^n}_{n-Betti} = \underbrace{rank Z^n - rank B^n}_{Homology\ groups} = \underbrace{|{}^+ \sigma^n| - |{}^- \sigma^{n+1}|}_{Filtration} \quad (3.1)$$

In a nutshell what it says, is that the topological important information is connected to the filtration of simplices. The Euler-Poincaré characteristic representing the topological important information can be calculate with Betti numbers. This conclusion can be further used because the Betti numbers in turn, represent ranks of homology. In other words, at the end we want to find groups of chains and circles that are either trivial or bounding. All of that and more will be covered in this chapter.

Note that the following expositions are somewhat informal in so far that they focus on conveying the concepts rather than building a strict series of deductions. Lemmata or proofs will only be given if they are of importance for the understanding, further details and side-notes generally refer to sections in the appendix.

### 3.1. Mathematical Primer

Before we introduce any concepts it is important to stress that the field of topology is of axiomatic nature. Therefore it can appear that already known structures are convoluted by cryptic terminology, but as often with axiomatic fields we have to accept these definitions as new. For example, we mostly imply metric spaces when talking about “a space”, when the differences are actually quite profound, see figure 3.2:

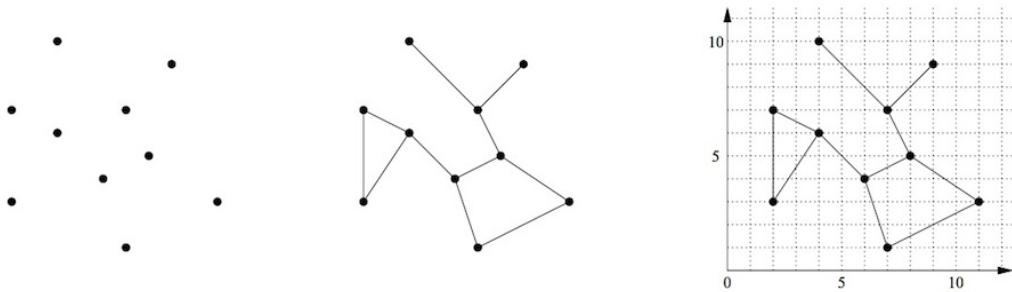


Figure 3.2.: A point-, topological- and metric-space.

A space can be a set of points without any structure. For a topological space we need knowledge of the connectivity of the space, each point *knows* which points are near it, that is its neighborhood. To form a metric space we additionally need an associated metric, which enables us to measure distances and in turn, implicitly defines neighborhoods. Consequently any metric space is a topological one. This means the geometry and topology of a space are fundamentally related, as they are both properties of a certain space. However, the questions we are interested in are often topological in nature, and we may solve them easier by focusing on the topology and not on its geometry. To do so, we need to identify intrinsic properties of spaces, especially the ones that are not changing – so called invariants of the space<sup>22</sup> [Zomorodian, 1996, cf. pp.1-5].

<sup>22</sup> This definition, to classifying geometries by their underlying symmetry groups, was coined by Felix Klein (1849-1925) under the name “Erlanger Programm”.

“... Euclidean metric geometry can be described as the study of properties unchanged by the group of all rigid motions. ... All of these characteristics are dependent on two fundamental invariants in terms of which that can be expressed, namely, distance and angle. Euclid’s pure geometry of measurement assumed that the effects of motion must leave size invariant. The issue of whether this is a good assumption for motion in the real world arises in twentieth-century physics.” [Kramer, 1982, p.405]

### 3.1.1. Topology

A topology is simply a system of sets that describe the connectivity of a set. Formally a topology on a set  $X$  is a subset  $T \subseteq 2^X$  such that<sup>23</sup>:

1. If  $A_1, A_2 \in T$ , then  $A_1 \cap A_2 \in T$ . (3.2)

2. If  $\{A_j | j \in J\} \subseteq T$ , then  $\cup_j A_j \in T$ . (3.3)

3.  $\emptyset, X \in T$ . (3.4)

We endow a set with structure to get the pair  $\mathbb{X} = (X, T)$ , called a topological space with any  $A_i \in T$  an open set and its complements in  $X$  closed. A continuous function between two topological spaces  $f : \mathbb{X}_1 \rightarrow \mathbb{X}_2$ , is called a map if for every open set  $B \in \mathbb{X}_2$ ,  $f^{-1}(B)$  is also open in  $\mathbb{X}_1$ , i.e. for every object in  $\mathbb{X}_2$  exists an unique object in  $\mathbb{X}_1$ .

A homeomorphism  $f : \mathbb{X}_1 \rightarrow \mathbb{X}_2$  is a 1-1 onto function such that both  $f$  and  $f^{-1}$  are continuous, i.e. they have the same topological type  $\mathbb{X}_1 \approx \mathbb{X}_2$ . Homeomorphisms partition the class of topological spaces into equivalence classes and a fundamental problem in topology is characterizing these classes of homeomorphic spaces.

### 3.1.2. Manifolds

A topological space may be viewed as an abstraction of a metric space. Similarly, manifolds generalize the connectivity of  $n$ -dimensional Euclidean spaces  $\mathbb{R}^n$  by being locally similar, but globally different, thus allowing to describe more complicated structures in terms of well-understood properties of Euclidean space. A homeomorphism mapping the local “area”  $U \subseteq \mathbb{X}$  to an open subset of  $\mathbb{R}^n$  is called a chart:  $\varphi : U \rightarrow \mathbb{R}^n$ . Formally a manifold is a topological space with a chart defined for every  $x \in \mathbb{X}$ .

For example, all neighborhoods on the 2-sphere  $S^2$  are homeomorphic to open disks. Likewise the circle or 1-sphere  $S^1$  is a 1-manifold as every point has a neighborhood homeomorphic to an open interval in  $\mathbb{R}^1$ . It is also the only compact 1-dimensional manifold. Manifolds are generally the only topological spaces we are interested in.

---

<sup>23</sup> Definitions and figures in this chapter stem from mainly three resources and won't be cited individually:

- [Hatcher, 2002] is a freely available textbook on algebraic topology and very often cited.
- [Zomorodian, 1996] the PhD thesis of Prof. Afra Zomorodian, which features an extensive introductory chapter on topology (often citing Hatcher).
- [Weisstein, 2002] the online resources of MathWorld, written and curated by Eric W. Weisstein, hosted/sponsored by and licensed to 'Wolfram Research, Inc'.



Figure 3.3.: Complete homeomorphism list for compact, boundaryless 2-dim. manifolds.

Another important attribute is compactness: A compact manifold can be embedded in  $\mathbb{R}^n$  if it has finite extent, i.e. it can be covered by finitely many charts  $\varphi_n$  – thus all compact surfaces are triangulable [which was first shown by Radó, 1925]. Compact manifolds in two dimensions are completely classified by two further attributes, namely orientation and genus, see picture 3.3 and table 3.1:

genus	0	1	2
<b>orientable</b>	sphere	torus	double torus
<b>nonorientable</b>	-	cross-cap	Klein bottle

Table 3.1.: Classes of 2-dimensional compact manifolds with low genus.

The genus is maybe the best known invariant property, as it intuitively describes the number of holes in a surface and we will see its connection to the more general notion of the Euler characteristic<sup>24</sup> later in section 3.1.4. The genus can also be seen as the maximum number of cuttings along non-intersecting closed simple curves that can be drawn on the surface without disconnect the manifold into two.

The second invariant, orientability, describes whether it is possible to give all local charts  $\varphi_n$  an ordering of either right- or left-handedness, without any two neighboring charts disagreeing globally. Formally we will define this concept in the next section 3.1.3.

Using the connected sum, the homeomorphism problem is fully resolved for 2-manifolds<sup>25</sup>, as every compact surface is homeomorphic to a sphere, the connected sum of tori, or the connected sum of cross surfaces<sup>26</sup>!

<sup>24</sup> In higher-dimensional manifolds, the Euler characteristic is replaced by the concept of Betti numbers and homology/cohomology, as it gives rise to a broader framework of invariants, called functors. For definitions of homology and Betti numbers, see section 3.1.8, respectively the glossary in appendix A.8.

<sup>25</sup> Grigori Perelman showed that 3-manifolds can also be decomposed into pieces with uniform geometry, whereas the problem is proven to be undecidable for higher dimensions [Morgan and Tian, 2007].

<sup>26</sup> Hence for any closed 2-manifold embedded in 3-dimensional space that can not be oriented, it follows, that it must have at least one self-intersection and any orientable 2-manifold without boundary is homeomorphic to a sphere with  $n$ -handles, i.e. its genus.

### 3.1.3. Simplicial Complexes

At the beginning of the chapter in picture 3.1 we have seen that a torus, like a projective plane<sup>27</sup> and a sphere, all can be obtained from a square by identifying the edges with certain directions and gluing them together accordingly. Cutting a square trivially along a diagonal produces two triangles that can be easily further subdivided, so each of these surfaces in turn can be reduced to triangles. Together with the result from the last section 3.1.2, that all compact 2-manifolds can be assembled by combining spheres, tori and cross-caps, we have proven that in fact all closed surfaces can be constructed from triangles.

Thus we have a single building block to construct a large class of 2-dimensional spaces and the idea now is to generalize this for any dimension: The  $n$ -dimensional analog of the triangle is the convex hull of  $n+1$  affinely independent points  $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ , it is called the  $n$ -simplex:  $\sigma_i^n$ , with  $n$  and  $i$  denoting the dimension and index:  $i, n \in \mathbb{N}$ . Simplices have familiar names for low dimensions, see figure 3.4:

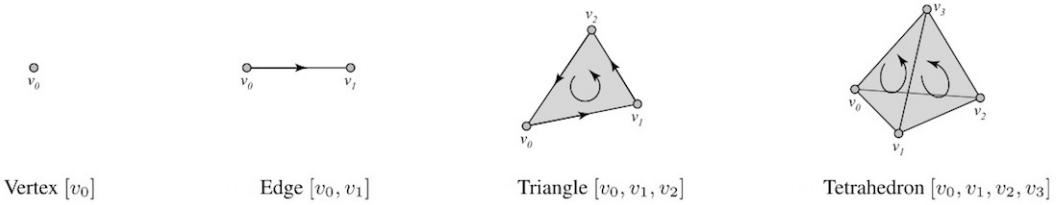


Figure 3.4.: Oriented  $n$ -simplices,  $0 \leq n \leq 3$ .

Now a simplicial complex  $K$ , is a finite set of simplices such that every subset  $T \subseteq \mathcal{V}$  is also in  $K$ , and the nonempty intersection of any two simplices, again is a subset in  $K$ . The dimension  $n$  of  $K^n$  is defined as the maximum dimension of its simplices:  $\max_n \{\sigma_i^n\}$ .

That is to say, that a simplicial Complex is a set of elements, i.e. simplices of different dimension, that built up and represent a manifold. The cumbersome definition stems from the fact that we want to exclude certain ill-formed sets, shown in figure 3.5.

To put this into context, a triangulation of a topological space  $\mathbb{X}$  is a simplicial complex  $K$ . Formally this is the case if the underlying space of the simplicial complex is isomorphic to the topological space:  $\|K\| \approx \mathbb{X}$ . Thus, two simplicial complexes  $K$  and  $L$  are isomorphic if and only if their topological spaces are:  $\mathbb{X}_K \approx \mathbb{X}_L$ .

<sup>27</sup> A sphere with one cross-cap has traditionally been called a real projective plane. The cross-cap is one of the three possible surfaces obtained by sewing a Möbius strip to the edge of a disk – the other two are the Boy surface and Roman surface, see picture 3.7. A sphere with two cross-caps having coinciding boundaries is topologically equivalent to a Klein bottle.

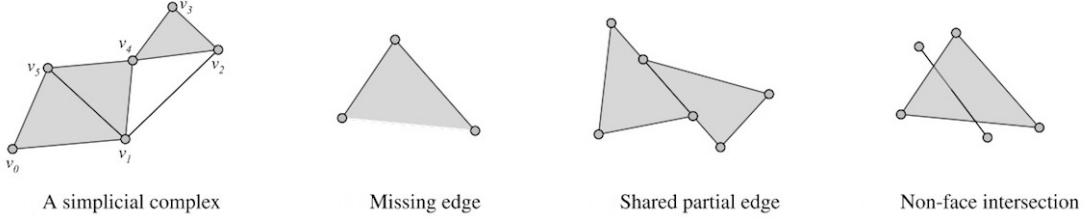


Figure 3.5.: One simplicial complex and three unions that are not complexes.

For purposes of homology it is important to keep track of the order of the simplices, so  $n$ -simplex really means  $n$ -simplex with ordering, which as a by-product, determines orientation: Two  $n$ -simplices sharing a  $(n-1)$ -simplex are consistently oriented if they induce different orientations. A triangulable  $n$ -manifold is orientable if all simplices can be oriented consistently, otherwise it is nonorientable, see figure 3.7. Orientations are normally graphically indicated by using arrows, as shown in figure 3.6. Note that in three dimensions, there is no unbounded non-orientable surface which does not intersect itself<sup>28</sup>.

We can generalize the idea now for any dimension  $n \geq 1$ :

Then an orientation of a  $n$ -simplex  $\sigma^n$ , is an equivalence class of orderings, written  $[\sigma]$ , of its vertices:  $\sigma = \{v_0, v_1, \dots, v_n\}$ , where:  $(v_0, v_1, \dots, v_n) \sim (v_{\tau(0)}, v_{\tau(1)}, \dots, v_{\tau(n)})$ , are equivalent orderings if the parity of the permutation  $\tau$  is even<sup>29</sup>.

It is worth noting that simplicial complexes, by definition, are combinatorial objects with topology and don't necessarily have to utilize geometry. If the topology gets separated,  $K^*$  is called an abstract simplicial complex<sup>30</sup>.

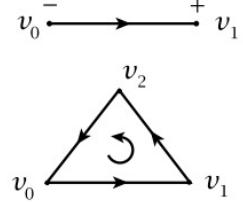


Figure 3.6.: Orientation.



Figure 3.7.: Basic non-orientable surfaces: Cross-cap, Roman and Boy surface.

<sup>28</sup> See appendix A.7, for a very elegant proof of this statement.

<sup>29</sup> For any finite set  $X$  with an ordering and at least two elements, the bijective mapping  $\tau : X \rightarrow X$ , falls into two classes of equal size, i.e. the even and odd permutations. The parity of a permutation is then defined as the number of inversions for  $N(\tau)$ , of pairs of elements  $x_i, x_j \in X$ , such that  $x_i < x_j$  but  $\tau(x_i) > \tau(x_j)$ , intuitively this denotes the number of occurred swaps.

<sup>30</sup> Actually written down for simplicial complexes, the mathematical definition of this vertex scheme, strikingly resembles the connectivity list of an .obj, .ply or .off file.

### 3.1.4. General Euler Characteristic

Next to orientability the genus is the second most well known invariant for classifying 2-manifolds. We already have an intuitive notion for it, as it represents the number of holes<sup>31</sup>, but with simplicial complexes at our disposal we can formulate this concept in a generalized and formal adequate way for any  $n$ -manifold.

The Euler characteristic is defined for a simplicial complex  $K$  with  $|\sigma^n|$  finitely many  $n$ -simplices as the number of even- minus the number of odd-dimensional simplices:

$$\chi(K) = \sum_{n=0}^{\dim K} (-1)^n |\sigma^n| \quad (3.5)$$

Which in the 2-dimensional case leads to the well known relation:

$$\chi(M_2) = |\sigma^0| - |\sigma^1| + |\sigma^2| = \text{vertices} - \text{edges} + \text{faces}$$

And we get for the tetrahedron  $\sigma^3$  in picture 3.5:  $\chi(\sigma^3) = 4 - 6 + 4 = 2$ , which by definition is a triangulation of the sphere  $S^2$ , thus defining the homeomorphism for any convex polyhedron  $M$  as  $\chi(M) = 2$ , respectively  $\chi(M) \simeq \chi(S^2)$ .

The Euler characteristic of a  $K$  depends only on its homotopy type, no matter how it is geometrically represented, i.e. it is an integer invariant for  $|K|$ , the underlying space. We already know that any compact orientable surface is homeomorphic to a sphere or  $n$  connected tori. Thereby we can give an atomic formula for any connected sums of orientable  $M_{sum}$  or nonorientable  $N_{sum}$  manifolds<sup>32</sup>:

$$\chi(M_{sum}) = 2 - 2g, \text{ respectively: } \chi(N_{sum}) = 2 - g \quad (3.6)$$

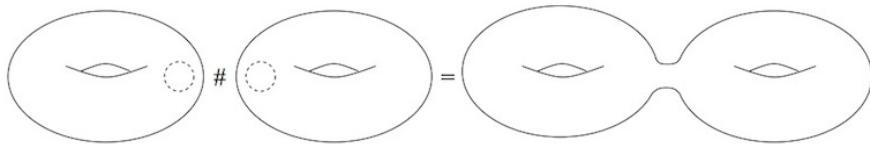


Figure 3.8.: A double torus, as the connected sum of two tori.

<sup>31</sup> Though even for non pathological cases it can be very hard to “see” the genus of a given surface. A famous example is the so called “House with two rooms”, illustrated in appendix A.6.

<sup>32</sup> An informal explanation for the added  $(-2g)$  and  $(-g)$  terms is, that, to connect the surfaces one either has to cut out a disk on either tori  $(-2)$ , or cut two disks and insert a projective plane  $(-2+1)$ ,  $g$ -times.

### 3.1.5. Homotopy

Two continuous functions from one topological space to another are called homotopic, if one can be continuously deformed into the other. The deformation then is called the homotopy between the two functions and an equivalence relation on topological spaces<sup>33</sup>. More precisely, two continuous maps  $f, g : \mathbb{X}_1 \rightarrow \mathbb{X}_2$  are homotopic, if a continuous function exists  $F : \mathbb{X}_1 \times [0, 1] \rightarrow \mathbb{X}_2$ , so that:

$$F(x, 0) = f(x) \text{ and } F(x, 1) = g(x), \text{ for all } x \in \mathbb{X}_1 \quad (3.7)$$

To give an example: Any two continuous real functions  $f, g : \mathbb{R} \rightarrow \mathbb{R}$  are trivially homotopic via  $F(x, t) = (1 - t) \cdot f(x) + t \cdot g(x)$ . If  $f$  is homotopic to a constant map, i.e.  $f \simeq \text{const.}$ , then  $f$  is called nullhomotopic. As with functions, two spaces  $\mathbb{X}_1$  and  $\mathbb{X}_2$  are homotopy equivalent if they can be transformed into one and another by continuous bending, shrinking and expanding operations, see picture 3.9.

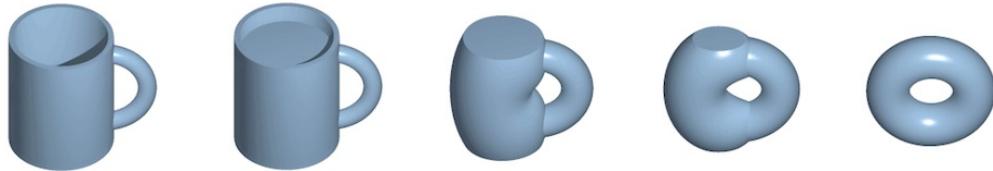


Figure 3.9.: A famous example for homotopic equivalence are a coffee cup and a torus.

A topological space  $\mathbb{X}$  is said to be contractible or nullhomotopic if  $\mathbb{X}$  is homotopy equivalent to a point, i.e.  $\mathbb{X} \simeq \sigma^0$ .

It is important not to mix up homeomorphism with homotopy. For example, a cylinder is not homeomorphic to a circle, but homotopy equivalent to it, as it may continuously be shrunken via a deformation retraction<sup>34</sup>. Thus, two spaces with a different topological type can still have the same homotopy type, but all homeomorphic spaces are homotopic:  $\mathbb{X}_1 \approx \mathbb{X}_2 \Rightarrow \mathbb{X}_1 \simeq \mathbb{X}_2$  but not the other way around.

<sup>33</sup> Equivalence classes under  $\simeq$  are called homotopy types. Like any equivalence relation they partition a set so that every element of the set is a member of one and only one cell of the partition. The intersection of any two different cells must be empty and the union of all the cells equals the original set. They are reflexive:  $X \simeq X$ , symmetric:  $X \simeq Y \Rightarrow Y \simeq X$  and transitive:  $X \simeq Y \& Y \simeq Z \Rightarrow X \simeq Z$ .

<sup>34</sup> Deformation retraction is a special case of a homotopy, with the requirement that the target space is a subspace. The retraction of a space  $\mathbb{X}$  onto a subspace  $A$  is a family of continuous maps  $f_t : \mathbb{X} \rightarrow A, t \in [0, 1]$ , so that  $f_0$  is the identity map:  $f_0(\mathbb{X}) = \mathbb{X}$ , and  $f_1(\mathbb{X}) = A$ . In other words, starting from the original space  $\mathbb{X}$  at time 0, we continuously deform the space until it becomes the subspace  $A$  at time 1. This is done without ever moving the subspace  $A$  in the process.

### 3.1.6. The Fundamental Group

We saw in the last section that two maps are homotopic if one can be continuously deformed into the other. Now this idea gets applied to maps on surfaces, called paths. Paths that begin and start at the same basepoint  $x_{bp}$ , namely loops, are of special concern and used to characterize the so called fundamental group.

Loops that are boundaries of a disk are contractible and belong to the same class as the trivial loop, which never leaves its basepoint. If we find non-bounding loops, they form a different class, see picture 3.10.

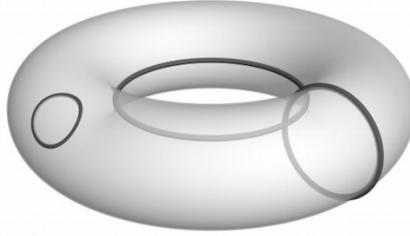


Figure 3.10.: One boundary loop (on the left) and two non-bounding loops on a torus.

$$f_{loop} : [0, 1] \rightarrow \mathbb{X} \quad \text{with} \quad f_{loop}(0) = x_{bp} = f_{loop}(1) \quad (3.8)$$

The non-bounding loops define the fundamental group. Formally the fundamental group of a topological space is the group formed by the set of equivalence classes of all loops with the same initial and final basepoint, under the equivalence relation of homotopy:

$$\pi(\mathbb{X}, x_{bp}) = \{f_{loop} : [0, 1] \rightarrow \mathbb{X} \mid f_{loop}(0) = x_{bp} = f_{loop}(1)\}_h \quad (3.9)$$

For a non-pathological spaces, i.e. path-connected spaces, the basepoint is irrelevant and we refer to the group as  $\pi_1(\mathbb{X})$ . For example, any loop drawn on the sphere is bounding, so  $\pi_1(S^2) \cong \{0\}$  where  $\cong$  denotes group isomorphism<sup>35</sup> and  $\{0\}$  is the trivial group.

Each homotopy class consists of all loops which wind around the circle a given number of times. So the fundamental group of the circle is isomorphic to the additive group of integers  $\mathbb{Z}$ . Intuitively then,  $\pi_1(S^1) = \mathbb{Z}$  and similarly the two non-bounding loops in picture generate the fundamental group of a torus:  $\pi_1(\text{Torus}) \cong \mathbb{Z} \times \mathbb{Z}$ .

<sup>35</sup> Admittedly, it is hard not to get confused by all the introduced -morphisms (left alone, that there are even more esoteric ones). Isomorphism is a bijective group homomorphism, which in turn must not be confused with homeomorphism. Contrary to homeomorphism, homomorphism is structure-preserving.

The fundamental group, in fact, measures the 1-dimensional hole structure of a space. Following is table 3.2 of the fundamental group for some common spaces:

Space	Symbol	$\pi_1$
Circle	$S^1$	$\mathbb{Z}$
Torus	$T^2$	$\mathbb{Z}^2$
Projective plane	$\mathbb{RP}^2$	$\mathbb{Z}$
Sphere	$S^2$	0

Table 3.2.: Fundamental group for various spaces.

The fundamental group is one in a series of homotopy groups  $\pi_n(\mathbb{X})$  that study higher dimensional holes. The homotopy groups  $n > 1$  extend the notion of a loop to  $n$ -dimensional circles and capture the homotopy classes of these circles.

### 3.1.7. Chains and Circles

As before, we can now generalize the concept of paths and loops with the help of simplicial complexes and use it to define another topological invariant, homology.

A path  $c$  constructed with simplices of dimension  $n$  is called a  $n$ -chain of the simplicial complex  $K$ . With  $\sigma_i^n \in K$ ,  $k_i \in \mathbb{Z}$  and the brackets  $[\cdot]$  indicating orientation as introduced in section 3.1.3, we define a  $n$ -chain as the sum of  $i$  connected simplices:

$$c^n = \sum_i k_i [\sigma_i^n] \text{ respectively the chain group of all } n\text{-chains: } C^n = \{\cup_K c^n\} \quad (3.10)$$

We also define a commutative addition for chains with integer coefficients modulo 2. In other words, the sum of two  $n$ -chains  $c$  and  $d$  is the symmetric difference of the two sets:  $c + d = (c \cup d) - (c \cap d)$ , preventing multiple instances of an element  $\sigma^n$  in the set.

The chain group  $C^n$  is the set<sup>36</sup> of all chains with dimension  $n$ , hence  $K$  has a chain group in every dimension. These chain groups  $C^n$  are structurally related in the way that

<sup>36</sup> To be precise, this set forms a free Abelian group on the oriented  $n$ -simplices:  $(C^n, +)$ . Which means, they form a basis and the entirety of the elements can be written as linear combinations of the basis. Abelian groups are structures of abstract algebra, studied in group theory. As we are interested in characterizing the topology of spaces, group theory is a related classification systems that yields many interconnections. Although promising powerful tools to define equivalence relations, the exploration of this relation is beyond the scope of this thesis and my knowledge thereof [for an in depth introduction to the field of geometric group theory, see [Stillwell, 1993](#)].

$(n-1)$ -chains form the boundaries of  $n$ -chains, i.e. every tetrahedron  $\sigma^3$  is bounded by four triangles  $\sigma_{\{1-4\}}^2$  and every triangle  $\sigma^2$  is bounded by three edges  $\sigma_{\{1-3\}}^1$  and so forth. We now define this relation formally via the boundary operator:

$$\partial \sigma^n = \partial \{v_0, \dots, v_n\} = \sum_i^n \{v_0, \dots, \hat{v}_i, \dots, v_n\} \quad (3.11)$$

where  $\hat{v}_i$  is the simplex that gets deleted from the sequence. We extend this operator to entire chain groups,  $\partial_n : C^n \rightarrow C^{n-1}$  and interpret  $\partial_0 \equiv \emptyset$ . The entire series of the groups that are connected via the boundary operator is then called a chain complex  $C^K$ :  $C^K = C^n \xrightarrow{\partial_n} C^{n-1} \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_1} C^0 \xrightarrow{\partial_0} \emptyset$

This gives us the definition of a  $n$ -circles as  $n$ -chains that have no boundary. As an example consider the triangle in figure 3.6, with the obvious 1-circle, formed by its edges:  $\partial_2 \sigma^2 = \partial_2 \{\sigma_0^1, \sigma_1^1, \sigma_2^1\} = \{v_1, v_2\} + \{v_0, v_2\} + \{v_0, v_1\}$ , now taking the boundary of the boundary we can see:  $\partial_1(\{v_1, v_2\} + \{v_0, v_2\} + \{v_0, v_1\}) = 2v_2 + 2v_1 + 2v_0$  and with the addition defined modulo 2, the results is:  $2v_2 + 2v_1 + 2v_0 = \emptyset$ . This is intuitively correct, as the boundary of a triangle is a cycle, and a cycle does not have a boundary. In fact, this intuition generalizes to chain groups of any dimension<sup>37</sup>:  $\partial_n(\partial_{n-1}) = \emptyset$ . In other words, the  $n$ -cycles are the kernel<sup>38</sup> of  $\partial_n$  and thereby form a subgroup of the chain group  $C^n$ , which is called the cycle group:  $Z_n = \ker \partial_n = \{c \in C^n \mid \partial_n c = 0\}$ .

If we recall the three circles on the torus shown in figure 3.10, we see that apart from the boundary loops, there are also non-bounding loops. The entirety of loops can be defined by the image of all  $(n+1)$ -boundaries:  $B_n = \text{im } \partial_{n+1} = \{c \in C^n \mid \exists d \in C^{n+1} : c = \partial_{n+1}(d)\}$ .

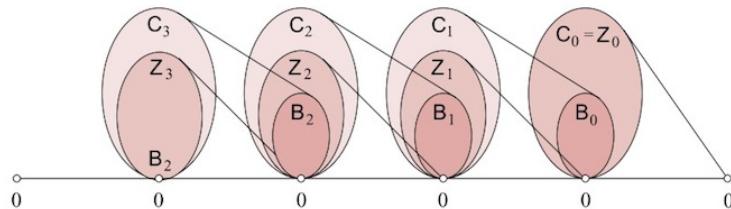


Figure 3.11.: The chain complex for a 3-dimensional simplicial complex.

In other words, for any dimension  $n$  of  $K$ , there exist chains of simplices:  $C^n$ , of which some form loops:  $Z^n$  and of those some are bounding:  $B^n$ , hence:  $C^n \supseteq Z^n \supseteq B^n$ , see 3.11.

<sup>37</sup> We omit the proof, but it is rather straight forward, as  $\partial_n \partial_{n-1} = \sum \sum [\dots] \bmod 2$  always cancels out.

<sup>38</sup> The boundary operator  $\partial_n$  is a group homomorphism, hence the kernel is defined as the set of all elements which are mapped to the identity element of  $\partial_{n-1}$ . The image, analogous to real functions, is the range of all mapped elements, i.e. all chains for which  $\partial_{n+1}$  is defined.

### 3.1.8. Homology Groups & Betti Numbers

We have  $n$ -chains and  $n$ -cycles, which are simplicial analogs of paths and loops for  $n=1$  in the continuous domain of topological spaces. Following the construction of the fundamental group, we now need the simplicial version of homotopy to form equivalent classes of cycles on triangulated spaces, that is the homology group<sup>39</sup>:  $H^n = Z^n / B^n = \ker \partial_n / \text{im } \partial_{n+1}$ . The homology groups are invariants for  $|K|$  and for homotopy equivalent spaces. This means formally,  $X_1 \simeq X_2 \Rightarrow H^n(X_1) \cong H^n(X_2)$  for all  $n$ . However, for the rest of the thesis, we will only consider the first homology group  $H^1$ , as it represents the circles we are interested in, especially the ones which constitute handle and tunnel loops.

One might ask, why we defined simplicial complexes and for that matter homology groups when they just resemble topological spaces and homotopy. The crucial difference is, that homology is defined for non-continuous spaces, built out of simplices and therefore lends itself to feasible methods for computation. In fact, if we extend homology groups with the definition of  $\text{rank}(H^n)$ , as the number of independent subsets, we get a full series of invariants, called the  $n^{th}$  Betti number, with intuitive meanings for low dimensions:

$$\beta^n = \text{rank } H^n = \text{rank } Z^n - \text{rank } B^n \quad (3.12)$$

$\beta^0$  counts the number of connected components of the space.

$\beta^1$  is the dimension of any basis for the loops.

$\beta^2$  counts the number of enclosed spaces or voids.

For example, the torus is one connected component, has two loops and encloses one void:  $\beta^0 = 1$ ,  $\beta^1 = 2$ ,  $\beta^2 = 1$ , and  $\beta^n = 0$  for all  $n > 2$ . Note that the Betti numbers for the projective plane are defined, although it is not embeddable in  $\mathbb{R}^3$  without intersecting itself<sup>40</sup>, see table 3.3:

Surface	$H^0$	$H^1$	$H^2$
Torus	$\mathbb{Z}$	$\mathbb{Z}^2$	$\mathbb{Z}$
Sphere	$\mathbb{Z}$	$\{0\}$	$\mathbb{Z}$
Projective plane	$\mathbb{Z}$	$\{0\}$	$\{0\}$

Table 3.3.: Homology groups for the basic surfaces.

<sup>39</sup> With  $/$  the quotient set, which for  $H^1$  is the class of circles that are non-bounding, analogous to  $\pi_1$ .

<sup>40</sup> Hiroshi Maehara conceived an elegant proof, showing the impossibility of:  $P^2 \hookrightarrow \mathbb{R}^3$ , see appendix A.7.

### 3.1.9. The Euler-Poincaré Formula

The relevance of Betti numbers goes further than just applying the concept of homotopy to simplices. To end this section, we derive the invariance of the Euler characteristic of section 3.1.4 from the invariance of homology<sup>41</sup>. By doing this we introduce a bit more algebra and complexity than we might otherwise would have needed, but it will yield a reference of justification for the next chapter.

Recall that a simplicial complex  $K$ , as a constructed object, breaks down into a chain complex  $C^K$  of finite length. The Euler characteristic  $\chi(K)$ , as defined in equation (3.5), is the alternating sum of the number of used simplices  $|\sigma^n|$ . However, this is also the rank of the corresponding  $C^n$ , as it is the group on oriented  $n$ -simplices that get generated by these very simplices. Thus we can define the Euler characteristic in terms of the chain complex:

$$\chi(K) = \chi(C^K) = \sum_n^{\dim K} (-1)^n \text{rank}(C^n) \quad (3.13)$$

Maybe surprisingly, it can be shown<sup>42</sup> that  $\chi(C^K) = \chi(H(C^K))$ , which means that the homology functor preserves the Euler characteristic of a chain complex. With the equation (3.12):  $\text{rank } H^n = \beta^n$ , we get the Euler-Poincaré formula:

$$\chi(K) = \sum_n^{\dim K} (-1)^n |\sigma^n| = \sum_n^{\dim K} (-1)^n \text{rank } H(C^n) = \sum_n^{\dim K} (-1)^n \beta^n \quad (3.14)$$

This relation, besides its intrinsic mathematical appeal, has important implications for us. It means that we can derive the Euler characteristic by computing the Betti numbers. More importantly it shows that Betti numbers are intimately related to the topological concept of certain paths, namely circles. And since tunnels and handles are described by circles, we have a connection between the former and the latter. Hence, in order to find special geometric structures, we can study specific groups of circles that are by themselves distinguishable via the decomposition of their simplices!

<sup>41</sup> Historically this idea gave rise to a famous conjecture, namely the so called Hauptvermutung. The hope was, that if two different simplicial complexes  $K$  and  $L$  are bound by a homeomorphism, i.e.  $|K| \approx |L|$ , it would translate to the homology of the spaces itself. In other words, if true, any two triangulations of a topological space have at least one common refinement, a single triangulation that is a subdivision of both of them. The general version of the conjecture was proven to be false, but the manifold version holds true for dimensions  $\dim \leq 3$  [for an extensive discussion, see: [Ranicki et al., 1996](#)].

<sup>42</sup> For the proof, see [\[Hatcher, 2002, pp.155-156\]](#).

## 3.2. Handles and Tunnels

The last section ended with the insight that circles, or rather loops as their 2-dimensional manifestations, are valid constructs to describe topological features. However, this is not enough, not only would it be much easier to compute topological properties via the simple Euler characteristic, but moreover the description of topological features in itself is “blind”. What we are really interested in, after all, is the identification of specific geometric details. Geometric details that come with certain topological traits, but in order to exploit circles for feature detection we need to employ the theory of topological persistence, which is a substructure of the more general Morse theory<sup>43</sup>. The theory identifies critical points at which level-sets of a function undergo topological changes, and relates these points via a complex – note that:  $\chi = 2 - 2g = \text{minima} - \text{saddles} + \text{maxima}$ , see figure 3.12:

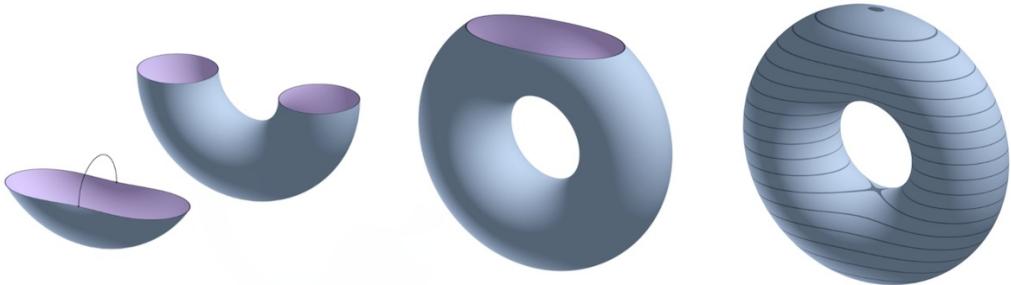


Figure 3.12.: Morse theory – starting from the bottom the homotopy type of the torus changes at its critical points from: point or disk, to cylinder, continues to a torus with a removed disk boundary, until it is finally a full torus.

The rest of this chapter is split into the following sections: First we will survey previous work in 3.2.1, then in section 3.2.2 we will proof that there is in fact, always a distinct class of circles that describe the geometric features we are interested in. Afterwards we define topological persistence in section 3.2.3 and see how it may help us, and finally in 3.2.4 we explain how to find the loops computationally.

<sup>43</sup> Morse Theory provides an analysis of the relationship between the geometry and the topology of a space. Though it was developed for smooth domains, it can be translated to simplicial complexes. It is a generalization of calculus of variations which seeks to find the path, curve, surface, etc. for which a given function has a stationary value. It draws on the relationship between the stationary points of a smooth real-valued function on a manifold and the global topology of the manifold. For example, if a compact manifold admits a function whose only stationary points are a maximum and a minimum, then the manifold is a sphere [for an introduction, see: [Kitagawa, 2006](#)]. Morse theory is related to the topology of Lie groups and has received much attention in the last decades, as it is important for the description of quantum field theory [cf. [Witten, 1982](#)].

### 3.2.1. Previous Work

Numerous papers have been published dealing with non-trivial loops on surfaces, as they hold important topological information. For a good overview, see the references in [Ni et al., 2004, especially section 2]. For our thesis, we will focus on preceding work with a explicit context in simplification. Below are some references with descriptions:

→ [El-Sana and Varshney, 1997] “*Controlled Simplification of Genus ...*”

They identify holes and the concavities by extending the concept of  $\alpha$ -hulls<sup>44</sup> to polygonal meshes under the  $L_\infty$  distance metric and then triangulate parts of the surface that are not accessible for a ball of a defined radius.

→ [Guskov and Wood, 2001] “*Topological Noise Removal*”

Propose a surface growing strategy to find features and remove unnecessary non-trivial topology. They use a local wave front traversal to discover and identify small tunnels. Then they find and select non-separating cuts to sever and seal the mesh, thus reducing the genus.

→ [Nooruddin and Turk, 2003] “*Simplification ... Using Volumetric Techniques*”

Describe a method for converting polygonal models to a volumetric representation. The topology altering part is based on 3D morphological operators that work in the volume domain.

→ [Wood et al., 2004] “*Removing Excess Topology From Isosurfaces*”

They find handles by incrementally constructing and analyzing Reeb graphs<sup>45</sup>. Handles are removed robustly by modifying the volume in a disk-filling procedure.

→ [Dey et al., 2007] “*On Computing Handle and Tunnel Loops*”

Using the concept of topological persistence, they filter a manifold to get a set of simplices that are linked to edges that form handle and tunnel loops. The main advantage of this approach is, that unlike others, it can provide a mathematical guarantee to find relevant topological features without introducing artifacts.

Hence we follow this idea<sup>46</sup> and built upon their work as described in section 3.2.4.

<sup>44</sup> Generalization for the concept of convex hulls as a set of points associated with disks that represent simple curves [introduced by Edelsbrunner et al., 1983].

<sup>45</sup> Named after Georges Reeb, these graphs of a function describe the connectivity of its level sets – for figure 3.12 the graph would be formed by two edges connected bottom and top of a circle.

<sup>46</sup> The work of Tamal K. Dey was first brought to our attention in a discussion with Tammy Boubekeur.

### 3.2.2. Definition & Existence of Handle and Tunnel Loops

One goal for this thesis is, not only to preserve, but to control topology. This entails mathematically having to define even very intuitive concepts like handles and tunnels. Formally this was first done in reference to mesh simplification by [Dey et al., 2007].

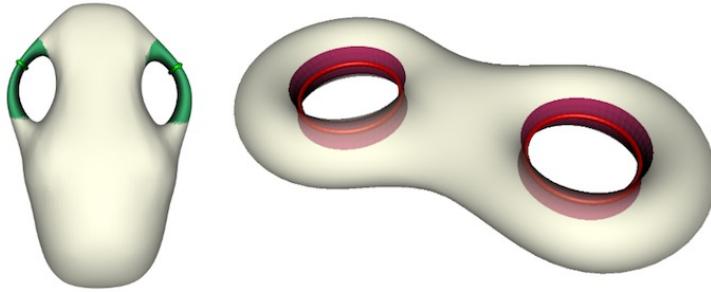


Figure 3.13.: Handles (green) and tunnels (red) with their defining loops [Dey et al., 2012].

A loop defines a handle if it spans a disk in the bounded space bordered by the surface. If one cuts the surface along such a loop and fills the boundaries with two disks, the handle is eliminated. Similarly, a tunnel loop spans a disk in the unbounded space bordered by the surface, see figure 3.13.

Formalizing this description, let  $\mathcal{M}$  be a compact, orientable surface without boundary, i.e. closed and embeddable. If  $\mathcal{M}$  reside inside a 3-sphere<sup>47</sup>, it separates  $S^3$  into two parts  $\{(S^3 \setminus \mathcal{M})\}$ , namely the inside  $\mathbb{I}$  and the outside  $\mathbb{O}$ . Note that  $\mathcal{M}$  is part of both  $\mathbb{I}$  and  $\mathbb{O}$ , analogous to two adjacent triangles that share an edge.

- A tunnel loop is a loop of which the homology class is trivial in  $H^1(\mathbb{O})$  and non-trivial in  $H^1(\mathbb{I})$ .
- Accordingly a handle loop has a trivial homology class in  $H^1(\mathbb{I})$  and a non-trivial in  $H^1(\mathbb{O})$ .

In other words, handles and tunnels are characterized by the fact that they can be contracted to a point on the inside but are incontractible from the outside and vice versa. By definition, the set of tunnel loops are disjoint from the set of handle loops, i.e. no handle can be at the same time a tunnel and conversely. However, not every non-trivial loop on  $\mathcal{M}$  is either a handle or a tunnel and both conditions have to be fulfilled, see figure 3.14.

<sup>47</sup> The reason why  $\mathcal{M}$  gets embedded in  $S^3$ , which is a compactification of  $\mathbb{R}^3$ , is to make notations of proofs easier since the space in consideration is finite.

It can be shown that for any connected closed surface  $\mathcal{M} \subset S^3$  of genus  $g$ , there exist exactly  $g$  handle and tunnel loops:  $\{c_i^{handle}, c_i^{tunnel}\}$ , with  $0 < i \leq g$  forming a basis for  $H(\mathbb{O})$ , respectively  $H(\mathbb{I})$ . Furthermore the homology class of circles, that is represented by  $[c_i^{handle}]$  and  $[c_i^{tunnel}]$  form a basis for  $H^1(\mathcal{M})$ , which relates them directly to the Betti number  $\beta^1$ , as expected, and will be of great importance in subsection 3.2.3.3.

The proof involves more algebraic topology than what was already covered, especially Mayer–Vietoris sequences, therefore we omit the technicalities but instead describe the idea behind it.

We saw that  $\mathcal{M}$  bisects the space in which it is embedded,  $S^3 = \mathbb{I} \cup \mathbb{O}$ . Hence it follows that:  $H^1(\mathcal{M}) = H^1(\mathbb{I}) \oplus H^1(\mathbb{O})$ , with the direct sum of abelian groups. More importantly it also holds that:

$$rank(H^1(\mathcal{M})) = rank(H^1(\mathbb{I})) + rank(H^1(\mathbb{O})) = 2g \quad (3.15)$$

For symmetry reasons it can be inferred that actually:  $rank(H^1(\mathbb{O})) = rank(H^1(\mathbb{I}))$ , thus proving that a 2-manifold of genus  $g$ , will always have  $g$  tunnel and handle loops.

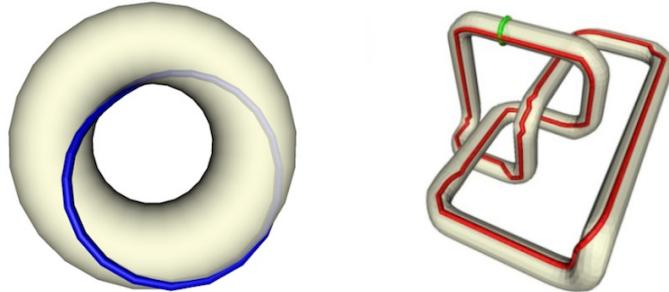


Figure 3.14.: Left: A non-trivial loop that is neither a handle nor a tunnel, Right: Loops on knotted surface reject the divide in  $H^1(\mathbb{I})$  and  $H^1(\mathbb{O})$  [Dey et al., 2012].

Although this theorem assures the existence of handle and tunnel loops on all connected closed surfaces, they do not bear intuitive meaning on knotted surfaces. Figure 3.14 shows such a surface, which is obtained by thickening a trefoil knot. Contrary to the natural intuition the red loop is not a tunnel loop. It is not trivial in  $H^1(\mathbb{O})$  yet it can be shown that it generates  $H^1(\mathbb{I})$ . This problems can be ameliorated but it introduces unnecessary complexity and consequently we restrict any manifold from here on to be knot free, which is not a problematic restriction in practice.

### 3.2.3. Topological Persistence

The concept of persistent homology has been introduced, independently by *Robins* [Robins, 1999] and *Edelsbrunner* [cf. Edelsbrunner et al., 2000]. It provides a mathematical tool, along with a combinatorial algorithm, to capture geometric shapes and is closely related to spectral sequences, mathematical ideas that have been developed 50 ago<sup>48</sup>.

The main idea behind it, is to study the topological changes of a mesh during an induced growth process and to keep track of how long a topological change lasts, that was induced by a certain simplex. This is done by monitoring the homology groups and Betti numbers. In order to measure the longevity of a topological feature, we need two further ingredients, one geometric, assigning a function to a space, the other algebraic, turning the function into measurements. The measurements make sense, only if the function does, which is studied by substituting an ordering of the simplices for the function<sup>49</sup>.

#### 3.2.3.1. Filtration

Let  $K$  be a simplicial complex. A filtration is a nested sequence of subcomplexes and we may think of the filtration as a description of how to construct  $K$  by adding individual parts of it, one after another:  $\emptyset = K_{-1} \subset K_0 \subset K_1 \subset \dots \subset K_n = K$

More than in the sequence of complexes, we are interested in their topological evolution, expressed by the corresponding sequence of homology groups. This is possible, since for every  $K_{n-1} \subset K_n$  with  $n \geq 1$ , the inclusion map induces a homomorphism between the homology groups:  $f : H(K_{n-1}) \rightarrow H(K_n)$ . Hence, the nested sequence of complexes corresponds to sequences of connected homology groups:

$$0 = H(K_0) \rightarrow H(K_1) \rightarrow \dots \rightarrow H(K_n) = H^p(K) \quad (3.16)$$

Where  $p$  represents the homomorphism for each dimension, corresponding to the type of  $n$ -simplices. As a consequence the filtration defines a partial ordering on the simplices with:  $\sigma_{early} \in (K_{n-1} - K_n)$  preceding  $\sigma_{late} \in (K_n - K_{n+1})$ .

<sup>48</sup> See the glossary in A.8 and chapter 1.4 of [Zomorodian and Carlsson, 2005] for more information.

<sup>49</sup> This section and its figures are compiled, from the papers: [cf. Delfinado and Edelsbrunner, 1995], [cf. Edelsbrunner et al., 2000], [cf. Edelsbrunner, 2001], [cf. Zomorodian and Carlsson, 2008] and [cf. Edelsbrunner, 2006]. It seems one scholar focused, but not only did *Edelsbrunner* lay the foundations of the field, he is also one of the most highly cited researchers, see: ISI Highly Cited. For a complete introduction, a thorough survey of the entire field and the current research, see [Kozlov, 2008].

This total ordering can be extended by deciding on the ordering of the simplices within each  $K_{n-1} - K_n$ . For simplicity one can assume that this ordering consists in the addition of a single simplex at a time. In other words, the simplices of  $K$  are ordered as  $\sigma_0, \sigma_1, \dots$  such that  $K_i = \{\sigma_0, \sigma_1, \dots, \sigma_i\}$ , for each  $i \leq n$ , avoiding ties that result from the otherwise simultaneous creation of simplices, e.g. when a new edge creates a triangle.

For example, a 2-manifold filtration is executed by first adding all the 0-simplices (vertices) to the set, then including the 1-simplices (edges) and finally the 2-simplices (triangle planes), see also figure 3.15 and the description in the next subsection.

### 3.2.3.2. Incremental Betti Numbers

The ordering of simplices in a filtration permits a simple algorithm for computing Betti numbers. Suppose the sequence of  $K_i = \{\sigma_j \mid 0 \leq j \leq i\}$ , for  $i \leq n$ . The initial Betti numbers for the set of the empty simplicial complex is obviously  $\beta^0 = \beta^1 = \beta^2 = 0$ , and the algorithm then works as follows:

	BETTI-NUMBERS ()
1	for $i = 0$ to $n - 1$ do
2	$k = \dim(\sigma_i) - 1$
3	if $\sigma_i$ belongs to a $(k+1)$ -cycle in $K_i$
4	then $\beta^{k+1} = \beta^{k+1} + 1$
5	else $\beta^k = \beta^k - 1$
6	endif
7	endfor
8	return $(\beta^0, \beta^1, \beta^2)$

Table 3.4.: Algorithm – Betti numbers of simplicial complexes.

The only question remaining is, how to decide whether a simplex  $\sigma_i$  belongs to a cycle in  $K_i$ , or not? For vertices  $\dim(\sigma_i) = 0$  it is defined, since every connected vertex belongs automatically to a 0-cycle<sup>50</sup>. To decide this for edges, we have to maintain a representation of the connected components, since an edge belongs to a 1-cycle if and only if its two endpoints belong to the same component, i.e. if it closes a triangle. Triangles  $\dim(\sigma_i) = 2$  are treated similarly, checking correspondence with 2-cycles.

<sup>50</sup> We imply that there are no “free floating” unconnected vertices allowed in the set  $K$ .

This definition sounds rather abstract, but can be easily made clear by showing a filtration. We call a simplex positive  ${}^+\sigma$ , if it belongs to a cycle and negative  ${}^-\sigma$ , if it doesn't. In figure 3.15, a filtration is depicted that builds a tetrahedron with a flag. Each step, shown by the number in the lower left box, adds one simplex to the set  $K_i$  with  $0 \leq i \leq 17$ .

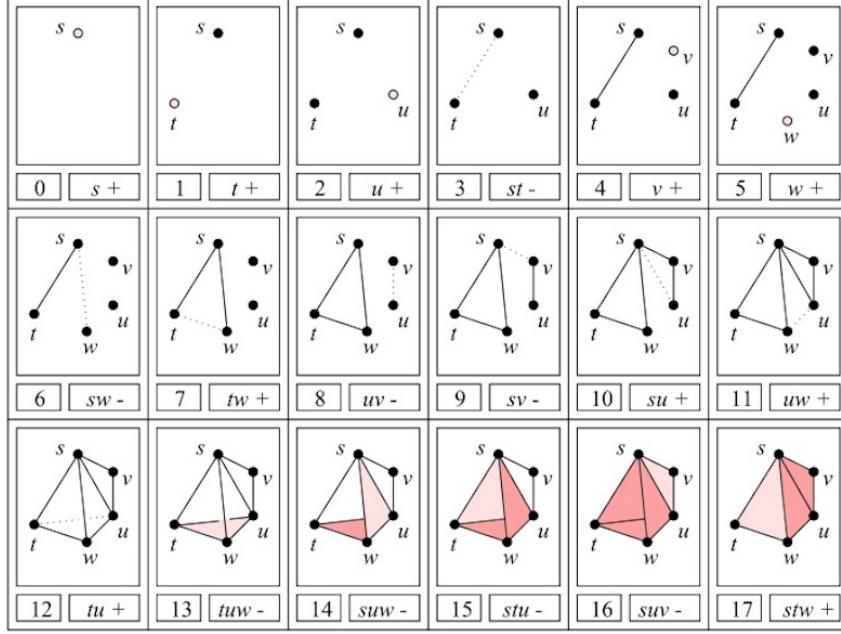


Figure 3.15.: Example of a filtration sequence, building a tetrahedron with flag triangle.

- Vertices  $\{s, t, u, v, w\}$  increment  $\beta^0$  – steps: 0, 1, 2, 4, 5 →  $\beta^0 = +5$
- Edges  $\{st, sw, uv, sv\}$  decrement  $\beta^0$  – steps: 3, 6, 8, 9 →  $\beta^0 = +5 - 4$
- Edges  $\{tw, su, uw, tu\}$  increment  $\beta^1$  – steps: 7, 10, 11, 12 →  $\beta^1 = +4$
- Faces  $\{tuw, suw, stu, suv\}$  decrement  $\beta^1$  – steps: 13, 14, 15, 16 →  $\beta^1 = +4 - 4$
- Face  $\{stw\}$  increments  $\beta^2$  – step: 17 →  $\beta^2 = +1$

We can now summarize the incremental steps and get:  $\beta^0 = 1$ ,  $\beta^1 = 0$ ,  $\beta^2 = 1$  meaning that there is one component with no non-trivial loops, encapsulating one void. Also, since  $\chi(K) = \sum_i (-1)^i \beta^i = +1 - 0 + 1 = 2$ , we verify the result with what we already know about compact and orientable 2-manifolds that have no holes, namely their Euler characteristic of  $\chi = 2$ . The correctness of the incremental algorithm implies:

$$\beta^i = |{}^+\sigma^i| - |{}^-\sigma^{i+1}| \text{ for } i \geq 0 \quad (3.17)$$

In other words, the  $i^{th}$ -Betti number  $\beta^i$  is the number of  $i$ -simplices that create  $i$ -cycles minus the number of  $(i+1)$ -simplices that destroy  $i$ -cycles by creating  $i$ -boundaries.

### 3.2.3.3. Persistent Homology

The idea of persistent homology can be described as a measure that ranks attributes by their lifetime in a filtration. To assess the persistence for a topological feature in the face of growth, positive and negative simplices are being paired<sup>51</sup>. Although this is only the tip of the iceberg of the theory, for our purposes, the concept of pairing is the only aspect we are interested in<sup>52</sup>.

Figure 3.16 visualizes two ways of illustrating the pairing of simplices from the filtration process in figure 3.15. Either with half open intervals  $[+\sigma_i, -\sigma_j)$ , or by the two dimensional representation that spans triangles. The light triangles represent 0-cycles, the dark ones 1-cycles and the area indicates the perseverance of a simplex:

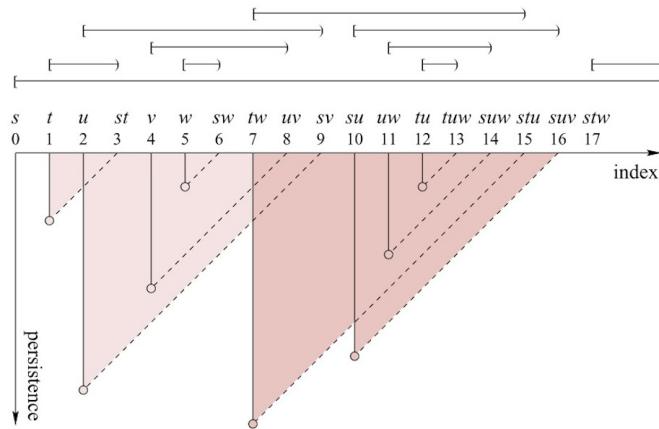


Figure 3.16.: Persistence of the simplices.

One might note that the vertex  $\{s\}$  and the face  $\{stw\}$  are unbounded and not drawn. This is a consequence of equation (3.17) and means that for every increment of  $\beta^i$  an unpaired positive simplex  $+\sigma^i$  gets associated that belongs to the homology group of the generating set, i.e. the basis. In other words, each simple connected manifold  $\beta^0 = 1$ , has a representative vertex  $\sigma^0$  and for every non-bounding 1-cycle, represented by  $\beta^1$ , there is an edge that belongs to a collection of non-contractible closed curves, and so forth.

<sup>51</sup> We haven't explicitly discussed the relation between positive and negative simplices and omit a proof, but since every Betti number  $\beta^i \in \mathbb{N}$ , it must follow that for every negative simplex  $\dim(i+1)$  exists a positive simplex  $\dim(i)$  that it pairs:  $|+\sigma^i| \geq |-\sigma^{i+1}|$ . Note that while the pairing depends on the filtration, the number of paired tuples and unpaired positive simplices does not. For a discussion of tracking generating cycles if the simplicial complex or the filtration changes, see [Busaryev et al., 2010].

<sup>52</sup> Not only can persistent homology be used as an invariant that captures the homological history of an arbitrary space, it also has applications in topological approaches for separating signals from noise and has been utilized in various fields. For example to measure structural changes in membrane fusion, which describe key stages in cellular processes such as virus infections [cf. Kasson et al., 2007].

Formally the reasoning behind this observation combines and concludes the work of the entire chapter and will finally explain why we are interested in all of this, as we initially set out to identify certain geometrical features, specifically handles and tunnels. Combining the relations of  $\beta$  as introduced in equations 3.14, 3.12 and 3.17, we get:

$$\underbrace{\chi = \sum (-1)^n \beta^n}_{\text{Euler-Poincaré}} \rightarrow \underbrace{\beta^n}_{n\text{-Betti}} = \underbrace{\text{rank } Z^n - \text{rank } B^n}_{\text{Homology groups}} = \underbrace{|^+ \sigma^n| - |^- \sigma^{n+1}|}_{\text{Filtration}} \quad (3.18)$$

Euler-Poincaré: Without a question the genus, respectively the Euler characteristic is the most prominent topological invariant. As intuitive as it might be, it is also very limited in its explanatory reach. For instance, just judging by  $\chi$ , there is no way to discriminate whether a tunnel of a manifold was closed or new objects added to the set. Therefore we refine the concept with the help of Betti numbers.

n-Betti: In this way, controlling the topology translates to tinkering with Betti numbers. Since we are generally not interested in modeling but simplifying the surfaces, we focus on decreasing Betti numbers. For the most part we deal with just one simple connected manifold, hence  $\beta^0 = 1$  is unchangeable and of minor importance. Neither is the amount of enclosed voids, denoted by  $\beta^2$ , as it pertains to the characteristics of the interior. Surfaces do not have  $\beta^i \neq 0$  with  $i \geq 3$ , so by exclusion the only really importance falls onto  $\beta^1$ , describing the amount of disjoint non-contractible circles.

Homology groups: The integer that represents  $\beta^1$  actually represents an entire group of circles that belong to the same homologic group. Consequently we have to expand the concept as we did with the genus that was refined into the Euler-Poincaré formula. This is achieved by examining the relation of chains in general and specifically the subgroup of chains that bind cycles<sup>53</sup>, also see figure 3.11.

Filtration: The final step is finding one representative circle from the entire homologic group. To narrow down the selection, we use filtration, as it identifies edges that are boundaries for the circles in question. That is to say, that after all the mathematical machinery that was put into motion we have firmly proven that for every tunnel or handle of a manifold we can compute an edge that is part of the defining loop. What is left now, is an exact description of these loops and the algorithm, given in section 3.2.2 and 3.2.4.

---

<sup>53</sup> Hinted by the fact that we introduced *ranks*, *kernels* and *images*, it is not surprising that persistence can also be nicely explained in terms of matrix operations, which is a field of study in itself [see chapters IV-3 and VI-1 in Edelsbrunner, 2006].

### 3.2.4. Computing Geometry-aware Loops

At the end of the last subsection we concluded that if we pair simplices during a filtration, we acquire a set  $\mathbb{U} = \{{}^+\sigma\}$  of unpaired positive simplices. We are interested in the subset of simplices  ${}^+\sigma^p$  of dimension  $p = 1$ :  $\mathbb{U}^1 \subset \mathbb{U}$  with  $|\mathbb{U}^1| = 2g$ , as this subset consists of edges that represent the tunnel and handle loops of the surface.

We now describe the principal steps<sup>54</sup> necessary to generate this set and afterwards discuss the two building algorithms in more detailed in subsection 3.2.4.1 and 3.2.4.2.

1. The input is a simplicial complex  $K_S$  that constitutes a closed, orientable surface  $\mathcal{M}$ . By using a Delaunay meshing algorithm<sup>55</sup>, one obtains the tessellated convex hull of  $\mathcal{M}$ , i.e. the simplicial representations for the inside  $K_{\mathbb{I}}$  and the outside  $K_{\mathbb{O}}$ , that together form  $K = K_S \cup K_{\mathbb{I}} \cup K_{\mathbb{O}}$ . Although we are only interested in the circles on  $K_S$ , we need triangles for the inside and the outside in order to add them to our filtration and find not only unpaired edges but also their corresponding loops.
2. The simplices of the surface  $K_S$  are added to filtration in an arbitrary order. Paring each of them with the algorithm described in 3.2.4.1, gives  $2g$  unpaired edges  $\mathbb{U}^1$ .
3. The simplices of  $K_{\mathbb{I}}$  are added to the filtration. Which pairs half of the edges in  $\mathbb{U}^1$  with negative triangles  ${}^-\sigma_i^2$  and associates an explicit circle to each edge, i.e. the one that got killed by  ${}^-\sigma_i^2$ , which by definition is a handle loop. The filtration with  $K_{\mathbb{I}}$  can be stopped as soon as  $g$  edges of  $\mathbb{U}^1$  are paired with their circles  $\{c_{\text{handle } i}^1\}_{i=1}^g$ , see figure 3.17 illustrating the steps 2-4.
4. Lastly the simplices of  $K_{\mathbb{O}}$  are included into the filtration. The remaining positive edges in  $\mathbb{U}^1$  get paired as before, thereby obtaining  $g$  more circles that are by definition tunnel loops  $\{c_{\text{tunnel } i}^1\}_{i=1}^g$ . Note that loops that get killed by negative triangles in  $K_{\mathbb{I}}$  or  $K_{\mathbb{O}}$  have all their edges lying on  $K_S$  because the simplices of the inside and outside only get added after all simplices of  $K_S$  are in the filtration.
5. The computed loops are undoubtedly correct as far as homology is concerned, but as can be seen in figure 3.17, they are not necessarily appealing in a geometrical sense. For that reason we refine the loops as described in 3.2.4.2.

<sup>54</sup> The following algorithms were first described and proven in [Dey et al., 2007] and later refined and sped-up in [Dey et al., 2008; Dey and Li, 2009]. Basically we follow the ideas for pairing simplices, but deviate in the latter stages when optimizing the found loops.

<sup>55</sup> We used the freely available implementation named `TetGen` for that purpose [cf. Si, 2010].

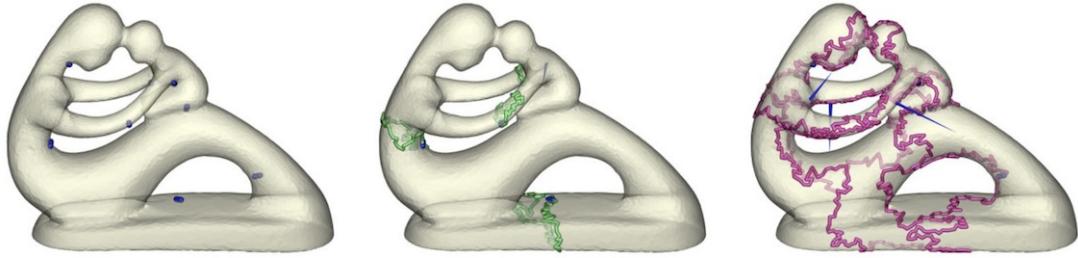


Figure 3.17.: Left to right: Unpaired edges, handle loops and tunnel loops [Dey et al., 2012].

### 3.2.4.1. Pairing Algorithm

Pairing is the essential concept of persistent homology, as it determines the lifetime, i.e. persistence of  $n$ -cycles. In order to pair the simplices, we have to determine whether they are negative or positive. Recall that a simplex  $\sigma^n$  of dimension  $n$ , is positive  ${}^+\sigma^n$ , if it creates a non-bounding  $n$ -cycle and that it is negative  ${}^-\sigma^p$ , if it kills an existing  $(n-1)$ -cycle. In other words, every new edge  $\sigma^1$  kills a 0-cycle, unless it closes a chain  $c = \{\sigma_0^1, \sigma_1^1, \dots, \sigma_i^1\}$  so that  $\partial(c + \sigma^1) = \emptyset$ . In the exemplary case shown in figure 3.15, it is not hard to figure the sign out  $\pm\sigma$ , but for an arbitrarily large simplicial complex  $K$ , we need a more elaborate approach using the notion of temporality – introduced as a consequence of equation (3.16), we defined that for every step in a filtration, exactly one additional simplex gets added with that the latest addition being the youngest simplex of the complex. Generally speaking  $\sigma_j = K_j - K_{j-1}$  is younger than  $\sigma_i = K_i - K_{i-1}$ , if it appears later in the filtration, i.e.  $K_j \supset K_i$ .

We analyze parts of the filtration from example 3.15 to see how the algorithm works before we give a formal description in 3.5.

In the stage  $K_3$  of the process the first edge gets added, before that only the vertices  $\sigma_0^0$  up to  $\sigma_2^0$  populate the simplicial complex. All vertices are trivially positive, as there are no  $(p-1)$ -cycles that could be closed. This can be shown formally:  $\partial\sigma^0 = \partial\{v\} = \emptyset \rightarrow {}^+\sigma^0$ , since the boundary operator, applied to any vertex, results in the empty set. Now to pair the new edge  $\sigma_3^1 = \{s, t\}$ , we check its boundary:  $c = \partial\sigma_3^1 = \partial\{s, t\} = (s + t)$ . No circle was closed, which would have left the the boundary empty, thus must be negative and we can pair the simplex with the youngest candidate. By convention we always assign the youngest unpaired positive simplex to the pair, in this case this is vertex  $t$  as it is younger than  $s$ :  $\sigma_1^0 > \sigma_0^0$  and we get the pairing:  $\langle t, st \rangle = \langle {}^+\sigma_1^0, {}^-\sigma_3^1 \rangle$ , which is visualized as the first

triangle in figure 3.16. Now we look at step K<sub>7</sub> that introduces the edge  $\sigma_7^1$ , that closes a triangle. More precisely:  $c = \partial\sigma_7^1 = \partial\{t, w\} = (t+w)$ , but since  $t$  and  $w$  are already paired, we have to add their pairs respectively their boundaries:  $(t+w) = (t+\{s, t\}+w+\{s, w\}) \rightarrow (t+\partial\{s, t\}+w+\partial\{s, w\}) = (t+s+t+w+s+w) = (2s+2t+2w)$ , note that we defined the addition modulo 2:  $(2s+2t+2w) = \emptyset \rightarrow {}^+\sigma_7^1$ , meaning that the edge is positive. Generalizing this for the entire simplicial complex, gives us the pairing algorithm:

	PAIRING ( $\sigma$ )
1	$c = \partial\sigma^p = \{\sigma_i^{p-1}, \dots, \sigma_j^{p-1}\}$
2	let $\sigma_{last}$ be the youngest ( $p-1$ )-simplex in $c$
3	while $\sigma_{last}$ is paired and $c \neq \emptyset$ do
4	let $c_{kill}$ be the cycle which was killed by $\sigma_{last}$
5	$c = c + c_{kill}$
6	update $\sigma_{last}$ to the youngest ( $p-1$ )-simplex in $c$
7	end while
8	if $c \neq \emptyset$ then
9	the simplex is negative ${}^-\sigma$ , killing $c$
10	pair $(\sigma_{last}, {}^-\sigma)$
11	associate $c$ as $c_{kill}$ with the simplex $(c_{kill}, {}^-\sigma)$
12	else
13	the simplex is positive ${}^+\sigma$
14	end if

Table 3.5.: Algorithm – Simplex pairing.

The algorithms spends most of its time expanding  $c = \partial\sigma$  in order to decide whether at the end it is empty or not. We can speed the process up by introducing another condition for the **while** loop. It relies on the observation that, at no time during the expansion of  $c$ , the youngest simplex in  $c$  can be older than the oldest simplex in  $\mathbb{U}^1$ , without it automatically being a positive simplex. This means that if  $\sigma_{last} < {}^+\sigma_{last \mathbb{U}} \rightarrow {}^+\sigma_{last}$ . The comparison is easy to do, as we only have to check the indices  $j$  and  $i$  of the corresponding stage of the filtration:  $\sigma_{last} = K_j - K_{j-1}$  and  ${}^+\sigma_{last \mathbb{U}} = K_i - K_{i-1}$ .

The observation can be formally proven but it is explicitly clear, considering the fact that after the filtration of  $K_S$ , only positive simplices are unpaired and that it is not possible to be left with unpaired negative simplices at that point<sup>56</sup>.

<sup>56</sup> See the first footnote on page 44 for more explanation.

### 3.2.4.2. Refining the Loops

After the entirety of this chapter we are now finally at the point where we actually have the loops representing the topological features we wanted to find, namely handles and tunnels. To be precise, we have a number of homologous circles that we acquire in the while loop of algorithm 3.5. During the process of finding the paired positive edges for a negative triangle we get a series of them and now have to choose one among them.

The previous work of [Dey et al., 2007, 2008; Dey and Li, 2009] has devoted considerable effort to perfect the selection by two methods: One is to assign geodesic sizes to the triangles in  $K_{\mathbb{I}}$  and  $K_{\mathbb{O}}$ , respectively the projection of their edges onto the surface. This is done to then order them according to size and introduce them, smallest triangle first to the filtration, thereby ensuring to pair smaller loops first. The other method drops the idea to connect the handle and tunnel loops to their unpaired positive edges all together. Instead the shortest circle in the homology class is searched, broadening the search immensely.

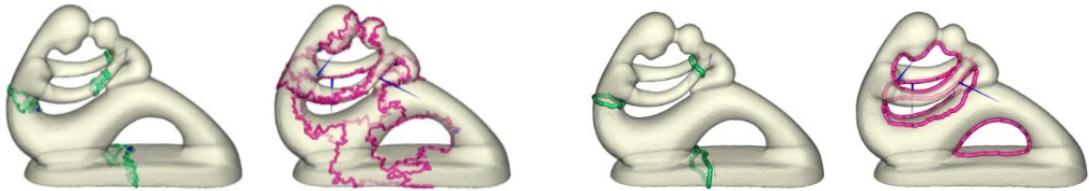


Figure 3.18.: Initial results and refined loops on the right [Dey and Li, 2009].

These two refinements undoubtedly improve the geometrical appearance greatly, as seen in figure 3.18. However we find the prize in computational expenses to be too high.

The only costly stage in the entire sequence thus far has been the Delaunay triangulation, which we consider to be a pre-process that can be automated and done in advance. One advantage of the persistence based algorithm is that it is combinatorial in nature and thereby avoids costly, error-prone numerical computations. Unlike many previous methods, the algorithm does not require computing any extra data structures such as Reeb graphs, medial axes, or curve skeletons<sup>57</sup>. We find it of pivotal importance to try and keep the

<sup>57</sup> In 2D, the medial axis of a shape is a set of curves defined as the locus of points that have at least two closest points on the boundary of the shape. In 3D, the corresponding object is also called the medial surface because in addition to curves, it can contain surface patches.

Likewise a curve skeleton is an abstract 1D geometrical representation of 3D shapes that considers topological features. By segmenting the object it can help spatial understanding, providing a higher level description of the object.

workflow as fast as possible, not changing workflow regime. As mentioned in section 1.4, we firmly believe that interactivity trumps automation and legitimates then necessary user interaction to obtain similar results.

Accordingly our solution for refining the computer loops is to substitute the geodesic metric by a simple edge-count. This is reasonable given the fact that for most triangle surfaces the edge size does not vary much in local neighborhoods. To further improve the results we give the user the freedom to manually tweak circles be changing parts of the path. Not only does this ameliorate the quality chasm between the reference implementation and our solution but also lends itself nicely to the simplification step where the user defines which loops to kill by triangulating them, see section 4 for the details.

This concludes this chapter 3, we have talked about basic topology in section 3.1, as well as more advanced concepts like filtration in section 3.2 and found a way to leverage the theory to describe and find topological features. Having described the algorithms involved we now turn to the actual application of the techniques. We will discuss in the next chapter 4, among other principles, how to put them use. Resulting in a direct and useful workflow to simplify not only fast and robustly, but also with numerous levers that can be tweaked directly by the user – one being the control over topology.

---

## 4. User-Guided Mesh Simplification

*“The original motivation to do research [that led to technological discoveries] was to expand the range of possibilities for storytellers.”*

– Ed Catmull [at the 73rd Scientific and Technical Academy Awards, 2001]

In this section we finally want to connect the theoretical part with the practical application. We will illustrate the workings of our reference implementation<sup>58</sup> and discuss our results. First, we will briefly describe the so called “TopStoc” algorithm in section 4.1. It serves as the core decimation method to which the other tools were added. The following three subsections 4.1.1, 4.1.2 and 4.1.3, discuss three useful adaptations of “TopStoc” that we made use of.

The next section 4.2 then lays out the topological options for the user. 4.2.1 shows what type of information the user gets and how it is presented to her. For instance, open boundaries get treated much like handle or tunnel loops, as they are also topologically important circles. Besides the tools to change the topology, described in 4.2.2, we will also talk about the limitations of our approach in 4.2.3.

The rest of the chapter from section 4.3 to 4.6 then illustrates further ways of valuable user-interaction and how to leverage guidance as well as practical considerations like support for textures in 4.5. Last but not least, we will come to talk about how to measure and evaluate results of any decimation and the two principle ways 4.7.1 and 4.7.2, we found most suitable to work with.

---

<sup>58</sup> Our reference implementation was written in STL conform C++. The only dependancies are [OpenMesh](#) which provides the data structure for the mesh operations, [Qt](#) for the GUI and [libQGLViewer](#) that facilitates the incorporation of [OpenGL](#) for the view-port. Although the core algorithms are very lean, the entire code-basis has grown to well over 4000 lines of code over the course of the thesis. Right now, most parts are not appropriately documented, but after some annotations are added, it will be made available – feel free to inquire via [mail@lnpe.at](mailto:mail@lnpe.at).

## 4.1. The TopStoc Algorithm

The basis for our work are the two algorithms described in the paper “Stochastic Sampling and Topological Clustering” [cf. [Boubekeur and Alexa, 2009](#)]. The main advantage over other solutions, undoubtedly is its speed, typically surpassing the quasi decimation standard “QSlim” by an order of magnitude. Consequently it is a natural fit for any system that wants to offer smooth interaction for user-guidance.

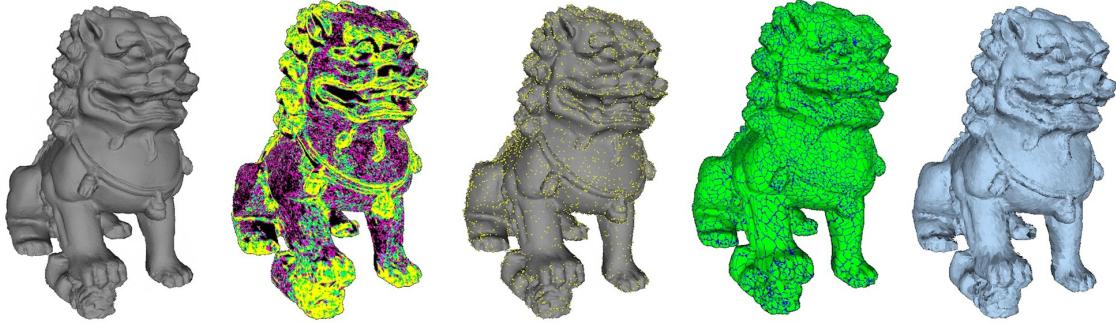


Figure 4.1.: Results from our TopStoc implementation, left to right: The base mesh; visualization of the local normal deviation; sampled vertices that define the low-resolution mesh; visualization of the topological clustering; the final mesh.

Before we can treat any modifications we give an ‘as short as possible’ rundown of the individual steps, as laid out in the original paper:

- *Stochastic Vertex Selection* – to select the “best” vertices for a low resolution model, every vertex is assigned a value  $x$  of geometric importance, we refer to as weight:

$$x(v) = \frac{|N_v| - \sum_{u \in N_v} (\mathbf{n}_v^T \mathbf{n}_u)}{2 \cdot |N_v|} \quad (4.1)$$

In other words, for every vertex the normal deviation of its immediate surroundings, i.e. the so called one-ring of vertex normals is calculated. Depending on the weight of each vertex and the target size, a subsample is selected. Note that the probability distribution must be tweaked, so that subsampling in flat areas is avoided. The second and third dragon in figure 4.1 show the vertex weight and an example of sampled vertices.

- *Topological Clustering* – the sampled vertices are used to partition the mesh via a breadth-first flood-filling. This way every vertex of the base mesh is allotted to one representative. Following the idea of vertex clustering each of the sub-meshes is used

to retriangulate the surface, but instead of spatial distance, the topological proximity defines the grid cells. Thus, contrary to normal clustering, the homotopy between the base and the decimated mesh can be ensured. The penultimate dragon in figure 4.1 shows the result of a flood-filling. Green areas mark topological cells, blue triangles the border between any two of them and red faces indicate an intersection of three different cells that determine the connectivity of the new mesh.

→ *Density Control* – not part of the basic decimation suit, rather an option, is this method a way to improve the shapes of the resulting triangles. There are already many concepts to refine triangles, but most of them are very costly. Density control avoids this relapse to high complexity by using a strategy inspired by “Poisson disk rejection sampling”. Before the sampled vertices are used in the topological cluster stage, each neighborhood of a sampled vertex gets rid of other samplings within a given distance.

#### 4.1.1. Fast Topological Clustering

As already mentioned, the stochastic vertex selection as well as the topological clustering are both very fast. However, of these two, the latter is considerably slower than the first. Not only is the clustering more algorithmically complex, but also is there no immediate way to quickly accelerate it. The calculation of weights and the selection of vertices can be trivially sped up by implementing a parallelized version of the algorithm, as every individual decision is completely independent of the other calculations<sup>59</sup>. For this reason we are more interested in improving the clustering stage and propose a small modification that pays off for cases in which the target sizes is very small, i.e. below ca. 5%.

The algorithm in its original version only needs the set of sampled vertices  $\bar{V}$  and a queue  $Q$  to store vertex references, besides the means to navigate the neighborhood of any vertex on the surface of course. During the flood-fill every vertex of the base mesh gets touched once to assign the mark of the cell it belongs to, see algorithm 4.1. Only to then iterate one more time over the entire mesh to check whether three different cell marks coincide on a given face.

In the modified version, we use knowledge about the connectivity of the decimated mesh

---

<sup>59</sup> Given that every SIMD thread has access, not only to one vertex, but also the adjacent vertices.

	BREADTH-FIRST FLOOD-FILLING ( $\bar{\mathcal{V}}$ )
1	for each vertex $v$ in $\bar{\mathcal{V}}$ do
2	mark $v$ with $v$
3	append reference of $v$ to $Q$
4	end for
5	while $Q$ is not empty do
6	$v :=$ head of $Q$
7	for each vertex $u$ in $N_V[v]$ do
8	if $u$ is not marked then
9	mark $u$ with mark of $v$
10	append $u$ to $Q$
+11	else if mark is different to $v$ then
+12	append the associated face $f_u$ to $L$
13	end if
14	end for
15	pop head of $Q$
16	end while

Table 4.1.: Algorithm – Breadth-first flood-filling.

to discard faces early on that can not qualify as candidates later. In the case that vertex  $u$  can't be marked by  $v$  because it was previously marked, there are two scenarios: Either it was already marked by  $v$  during an earlier step of the flood-fill, or it was marked by a different vertex  $s$ . Note that in the end, what we are ultimately interested in, is finding only the faces where all three vertices were marked by separate vertices. Hence, if vertex  $v$  encounters an already marked vertex  $u$  that bears the same mark as itself, it is impossible for the two adjacent faces – of the edge connecting  $v$  and  $u$  – to be one of these interesting faces.

Understanding this, we append faces to an unique list  $L$ , i.e. without allowing multiple entries, if during the flood-fill two different marks where found. The new steps in the algorithm are marked with '+' in 4.1. Note that we only add one face  $f_u$  to  $L$ , not both, this is because the check will happen two times from both sides starting with either vertex once. For an improved runtime it is important to avoid multiple insertions, consistently attaching either the face of the outgoing or the incoming half-edge only.

Although the final list of faces to check holds far less references, a speed-up manifest itself only for high decimation rates. This is due to the fact that inserting faces to the list  $L$  is very costly.

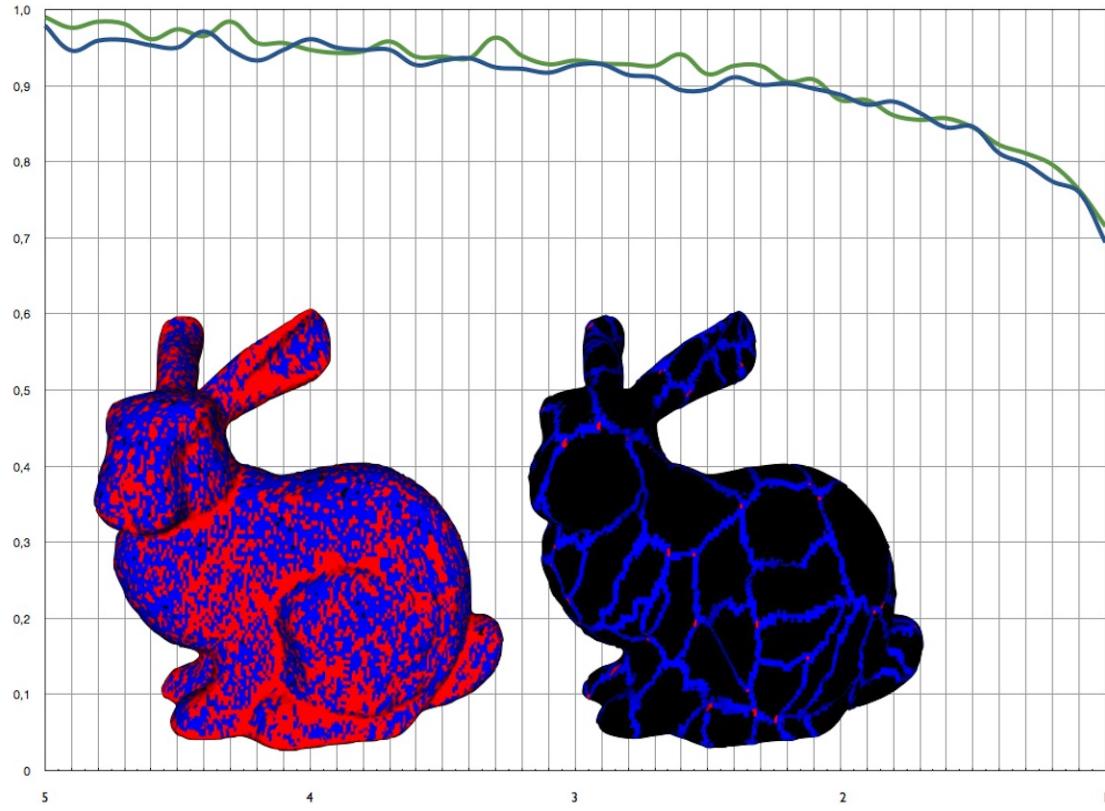


Figure 4.2.: The Y-axis shows the relative improvement in % of the runtime for two models (green/blue), for decimation ratios between 5% and 1% on the X-axis. To demonstrate the effect, the two bunnies show the discarded faces in black, for a very high and a very low sampling.

For example, decimating the *Chinese Dragon* Model, from initially over 655,000 vertices to about 3%, the runtime decreased at about 8%. For target sizes around 1% the speed-ups can be far greater, succeeding 30%. In figure 4.2 we plotted the relative runtime  $\frac{t_{fast}}{t_{normal}}$  for two models, namely the *Stanford Bunny* in green and the *Chinese Dragon* in blue<sup>60</sup>. As the target size is known before the flood-fill occurs, we automatically use this feature in case the the target size is smaller than 5%<sup>61</sup>.

<sup>60</sup> Each measurement was taken multiple times in 0.1% steps for the interval of 5% down to 1%.

<sup>61</sup> Note that “TopStoc” in general is better suited for high decimation ratios, as it is especially advantages over other solutions in these cases.

### 4.1.2. Adaptive Density Control

Density control as introduced in the original paper can greatly improve the quality of decimated mesh triangulation. Having said that, it suffers two drawbacks.  $\epsilon$  defines the minimum distance between two sampled vertices, as  $\epsilon$  increases the resulting mesh resembles more and more what one would expect from a uniform sampling. Thereby negating the intention and design of the stochastic selection phase of the vertices. The second problem is that the clean-up process can increase the runtime noticeably, up to 50%.

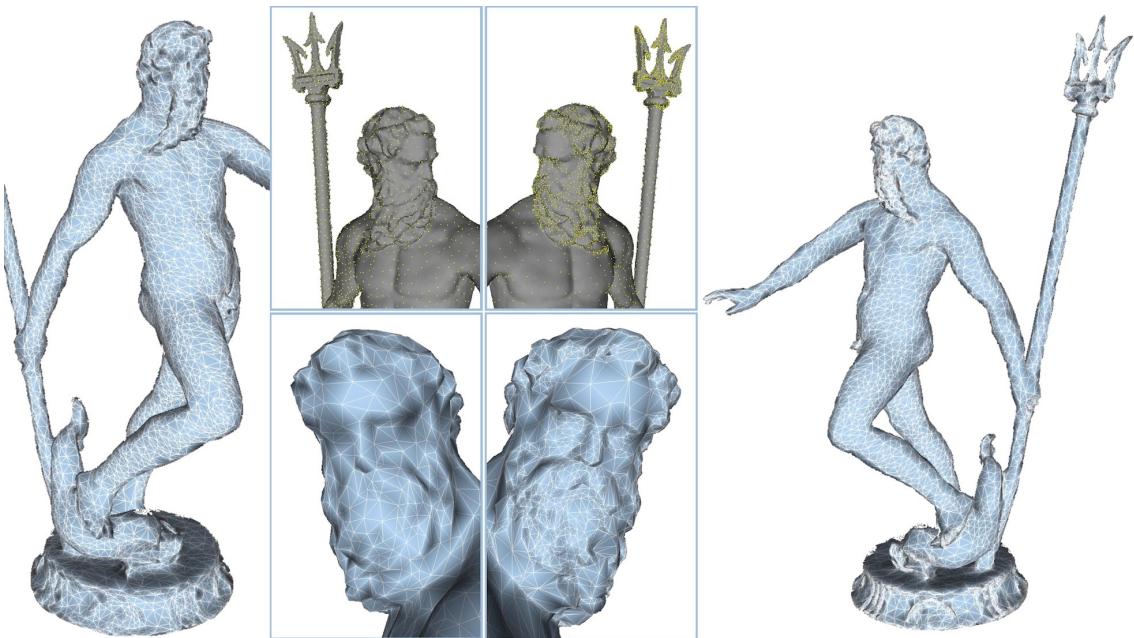


Figure 4.3.: Normal vs. adaptive density control. Whereas the sampling left shows almost uniform distribution, the adaptive version retains higher vertex counts in important regions without loosing good aspect ratios of the triangles. Note how the triangle size changes between forehead and beard.

We propose two simple modifications to mitigate these effects, which we call *Adaptive Density Control*. Ideally we want good triangle shapes without loosing too much geometric detail. To solve this problem we simply vary the distance  $\epsilon_v$  according to the weight of the vertex  $v$  it is associated with. In figure 4.3 are the results for the *Neptune* Model. On the left one can see the decimated mesh after isotropic or normal density control was used. For the mesh on the right, we adapted the distance  $\epsilon$  to just two values.  $\epsilon_{big}$  if the vertex weight was below the mean vertex weight:  $x(v) < \bar{x}$  and to  $\epsilon_{small}$  if the vertex weight was above the mean vertex weight:  $x(v) > \bar{x}$ , with  $\epsilon_{small} < \epsilon_{big}$ . Note that the value  $\bar{x}$  gets calculated during the sampling phase and is already given at that point.

We did not specify exact values in our example, neither for  $\epsilon_{big}$ , nor for  $\epsilon_{small}$ . The reason for this is, that instead of Euclidian distances as originally proposed, we opted for measuring the distances in terms of so called 1-rings<sup>62</sup>. This means for instance, that every sampled vertex with a weight  $x(v) > \bar{x}$  only evaluates its direct neighboring vertices  $\epsilon_{small} = 1$ , whereas a vertex with  $x(v) < \bar{x}$  will spread out much like the cells during flood-fill. We achieved good results for  $\frac{\epsilon_{big}}{\epsilon_{small}} \geq 2$ .

Not having to calculate Euclidian distances greatly improves the runtime, unfortunately one problem still retains that hasn't been mentioned at all, so far. While tinkering with adaptive density control, we calculated the aspect ratios of the triangles to have quantitative benchmarks. The aspect ratio of a triangle  $f$  is a dimensionless number  $AR \in [1, \infty)$ , that is 1 if and only if for all edges  $\|e_i\| = \|e_j\|$ :

$$AR(f) = \frac{\|e_0\| \cdot \|e_1\| \cdot \|e_2\|}{8 \cdot (s - \|e_0\|) \cdot (s - \|e_1\|) \cdot (s - \|e_2\|)}, \quad s = \frac{\|e_0\| + \|e_1\| + \|e_2\|}{2} \quad (4.2)$$

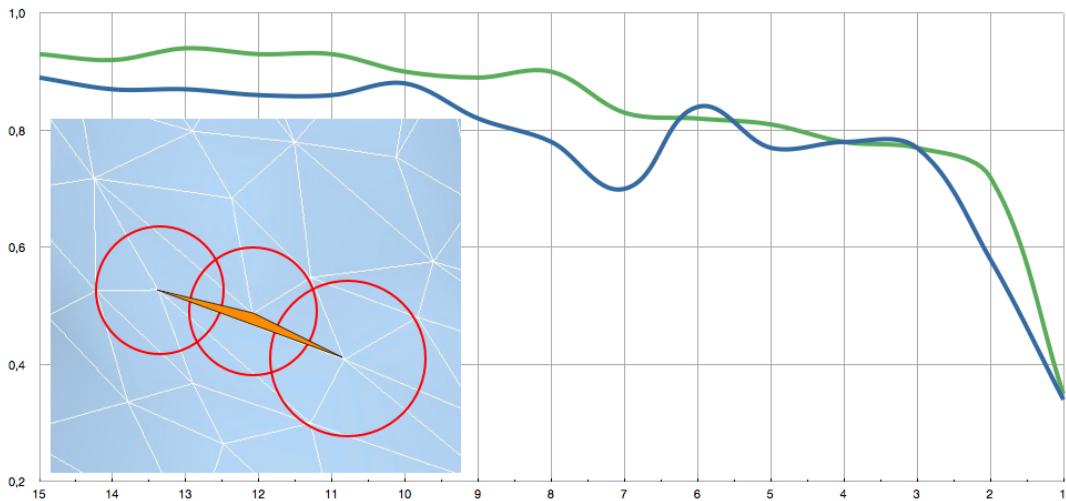


Figure 4.4.: Relative triangle quality, Y-axis, of normal decimation (blue) vs. density control (green), over small decimation ratios X-axis and an example of a bad triangle shape that can't be caught by density control.

After looking at the data we notices the the ratio between the mean and maximum value for all faces:  $AR(\mathcal{M}) := \frac{AR_{mean}}{AR_{max}}$  did not turn out as expected. First and most importantly,

<sup>62</sup> This concept depends to some extend on the assumption that edge lengths will not drastically vary in local neighborhoods. We think this is a sound assumptions, for some applications like marching cubes or Delaunay generated surfaces this holds by definition. However should the user see that a given surfaces is not suitable the Euclidean distance measure with an continuous  $\epsilon \in [d_{min}, d_{max}]$  can always be used as a fall-back alternative.

they weren't as good as hoped. Drilling down on the issue showed that only  $AR_{mean}$  improved and that  $AR_{max}$  still remained high. This is especially worrisome as for example, simulations are not resilient against single bad triangles no matter how good the rest are. The second finding was that the ratio  $AR(\mathcal{M})$  is highly volatile and can vary greatly from sample to sample.

In figure 4.4 we plotted  $AR(\mathcal{M})$  on the Y-axis, over the target size of the decimation, X-axis. The blue curve is the result for decimations without any density control, and green shows the results with normal density control. Although generally better, judging only by the diagram, density control hardly merits its expenses. Yet, optically the results are convincing. The phenomenon can be explained with the example from the triangulation in 4.4. All three vertices of the badly shaped, orange triangle have a big surrounding that was cleared by the density control step, but still this does not guarantee good shapes, as they nevertheless can be poorly arranged.

In order to circumvent this problem one can run a conventional clean-up step at the very end, which also means that the runtime drastically increases. Up to this point we do not have a viable solution that securely rids every single bad triangles from the mesh without breaking time complexity of the otherwise fast density control.

#### 4.1.3. Saving Sample Data

By no means a big addition, but a helpful supplement, is the option to save the references of sampled vertices. This is possible because the topological clustering will result in the exact same decimated mesh, as long as the order of the vertex queue  $Q$  is preserved and the underlying mesh does not change.

This enables the user to save any specific decimation at almost no memory cost. So instead of saving the entire meshes for various discrete LODs, the user can carefully edit each version, including any manual tweaks like setting control points etc. and save only the short and efficient list of references. Once the geometry of a mesh is loaded and only different versions are needed, this strategy can lead to huge difference in loading times, especially when dealing with big models.

## 4.2. Simplifying Topology

### 4.2.1. Information & Feature Detection

Our goal is to provide the user with clear and unambiguous information, so that she can easily decide whether to dismiss or keep a topological feature. It took the entire chapter 3, to lay down the mathematical foundations and to introduce the algorithms needed, especially the pairing of simplices 3.5. Now we want to show the results.

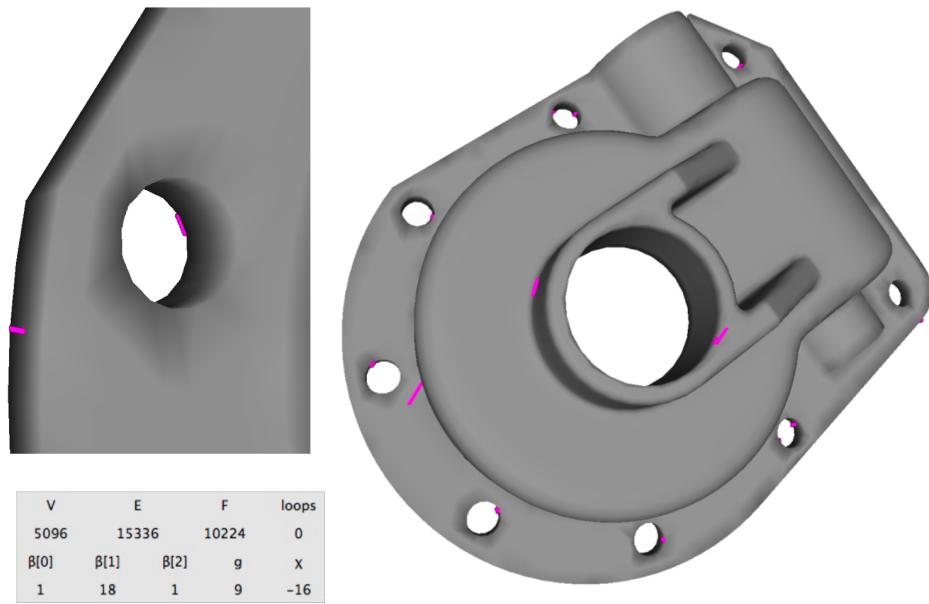


Figure 4.5.: The model *Casting*, of genus 9 with the handle and tunnel loops inducing edges visualized.

At any time the user is presented with a control panel displaying the following topological relevant information:

- $V, E, F$  – the total number of vertices, edges and faces, i.e. the number of simplices.
- $\beta[0] / \beta[1] / \beta[2]$  – the Betti numbers, which are determined once all simplices of the surface mesh have been paired. As soon as the filtration finishes, the unpaired edges that define handles and tunnel can be visualized. In figure 4.5 the mesh is of genus 9, producing 18 edges, which are colored in pink.

The exact location of these edges depends on the filtration, however we generally observed that they mostly stick very closely to “their” features. The further an unpaired edge is away, the more likely it is to get paired by a positive face of low

persistence as a positive face will pair with the first negative edge it encounters during the expansion of its boundary  $\partial^+ \sigma^2$ .

- $g \not\in \chi$  – the genus respectively the Euler–Poincaré characteristic. If a boundary edge, i.e. an edge only adjacent to one face, is encountered while the mesh is loaded, the exact number of boundaries gets determined. This is necessary in order to calculate the genus:  $\chi = 2 - 2g - \text{boundaries}$ . Note that this does not apply if the mesh is already filtered, as the Betti numbers are accurate regardless whether the simplicial complex  $K$  is closed or not, see also equation 3.14.
- *loops* – shows either the number of open boundary loops or the sum of the handle and tunnel loops. The latter can only be detected once the mesh is closed, as their definition relies on the distinction between the inside  $K_{\mathbb{I}}$  and outside volume  $K_{\mathbb{O}}$  of the mesh, see subsection 3.2.4.

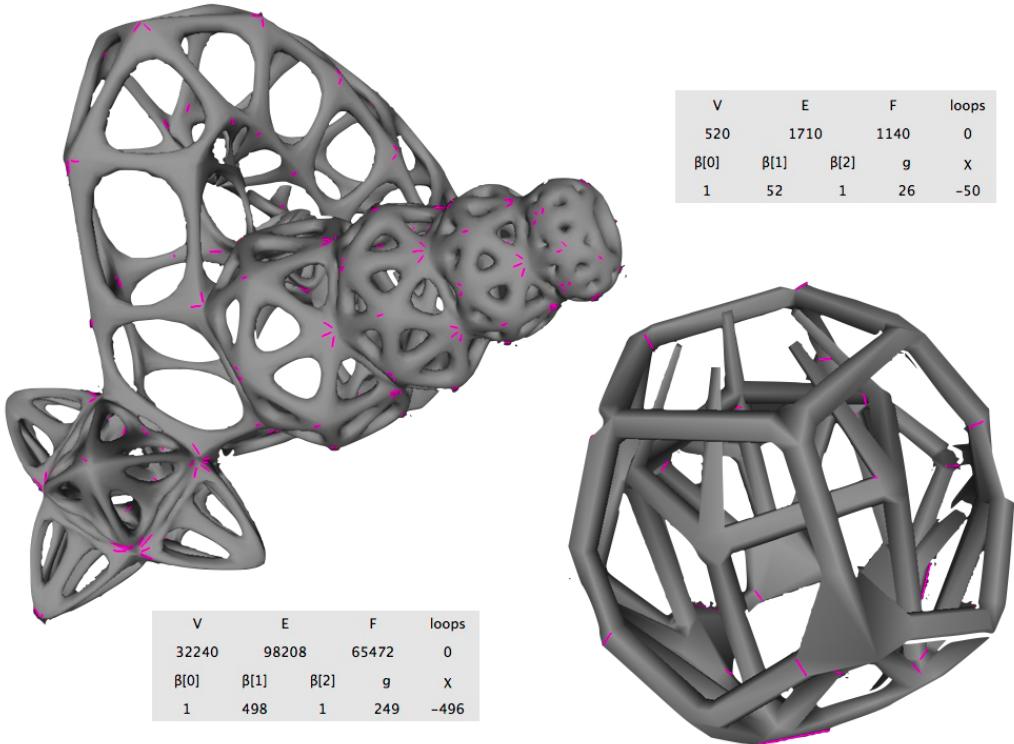


Figure 4.6.: Two ornamental models with high genera and their unpaired edges colored.

The decision if a particular unpaired edge gives rise to a handle or a tunnel can only be made after the simplices of the inside, respectively outside have been added to the filtration.

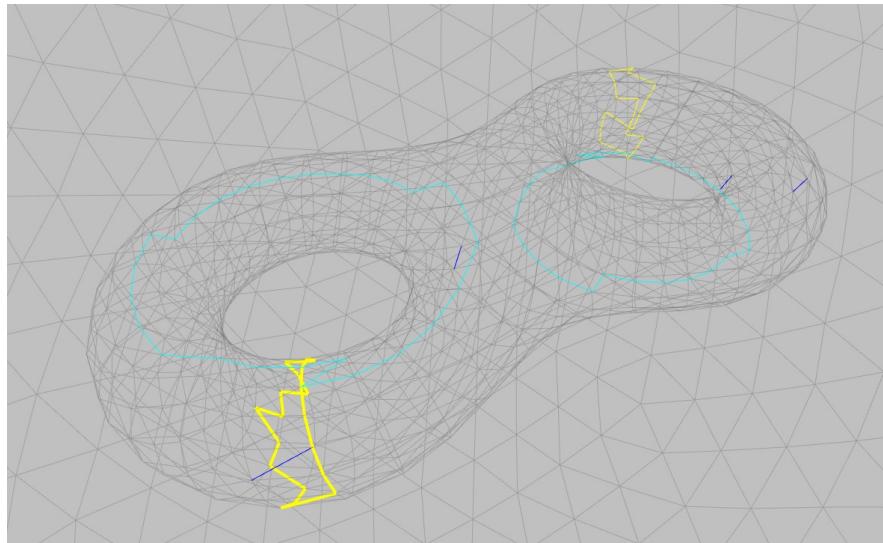


Figure 4.7.: A tunnel (turquoise) and handle (yellow) loop of a simple double torus with its paired edges in dark blue.

In figure 4.7 and 4.8 the found handle and tunnel loops are shown. Note that the refined loops generally do not share an edge with the generating edges they are paired with. In order to get geometrically better loops, we only save the shortest loop, in terms of the edge count, that is entirely on the surface. Once the loops are found, the user can cycle through them and adjust the edges individually. Thereby adjusted handles or tunnels can be used to quickly delete unimportant parts of the mesh.

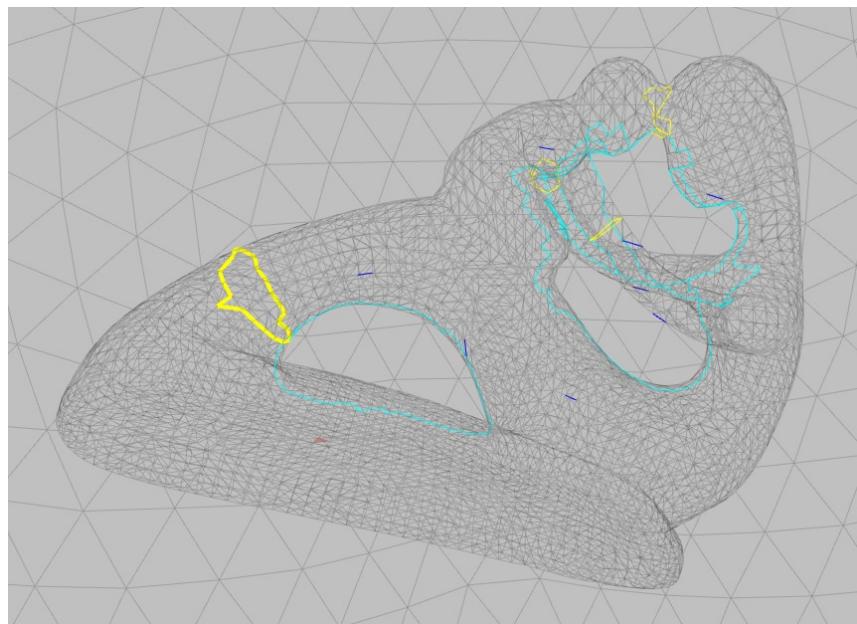


Figure 4.8.: Model *Mother* with its tunnel (turquoise) and handle (yellow) loops.

Note that the complexity of the pairing algorithm does not depend on the genus, only on the total number of simplices  $|\sigma^n|$  with  $n > 0$ , that constitute  $K$ . It is further noteworthy that the time needed to pair all simplices of a dimension  $n > m$  gets exponentially bigger as not only the size of the boundary  $\partial_n > \partial_m$  increase, but also with it the number of chain elements that get introduced during every step of the expansion.

Just pairing the simplices of the surface, typically takes only a matter of seconds for meshes with less than 100,000 faces and is negligible for meshes below 25,000 faces. Unfortunately, the pairing of faces from the inside and the outside scales far worse than we initially expected. It exceeds the minute mark even for meshes with less than 50,000 faces. This is due to the tetrahedralization, as the number of total simplices increases with a factor of about five. During our tests, the pairing of vertices and edges normally took less than 15% of the total time. This is concerning, mainly because we stopped pairing faces from the inside and the outside once the handle and tunnel loops where found, i.e. only paring less than half of the entire amount.

This compels us to acknowledge that finding topological accurate loops via filtration, is not a good fit with “TopStoc”. It could be argued, that this step should be skipped, pairing only the generating edges. This sounds unintuitive, but it is a valid point, especially since the unpaired edges are enough to indicate topological features and most importantly guaranteed to be mathematical correct. The user has enough information to decide if she wants to alter the topology or not and that is what we set out to achieve. So one possible direction to pursuit is to filter the surface mesh, thus ensuring that all important features get represented, but then switching to a cheaper heuristic to find loops.

#### 4.2.2. Mesh Editing

Independent of feature detection, the user always has the option to manually select any face of the mesh and delete it. Besides the deletion tool, the users second instrument to change topology is the retriangulation of a boundary loop.

In figure 4.9 this is shown exemplarily for a double torus that gets dissected by hand in two places. The deletion on the left kills a handle loop and the deletion on the right kills a tunnel loop. After the cuts, the mesh gets filtered again to update the Betti numbers – note that  $\chi$  remains -2, this is not an error, it is correct for the reason that we introduced 4 new boundaries. To indicate the boundaries, the vertices on them are colored red. As

long as the mesh has open boundaries, the user can cycle through them via buttons and retriangulate them<sup>63</sup>. The selected boundary is highlighted in yellow.

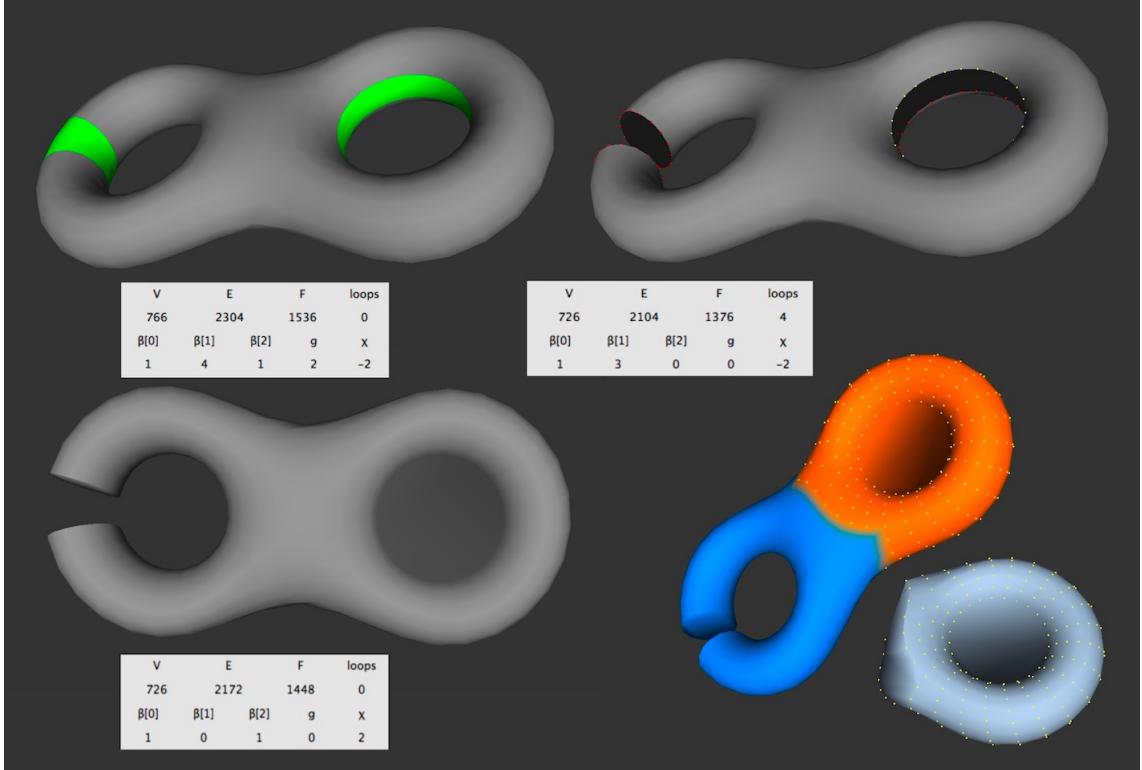


Figure 4.9.: Topological operations on a double torus.

After the four boundaries are retriangulated the mesh is topologically equivalent to a sphere as indicated by the invariants. To further prove the successful transformation we set the sampling probability for half of the mesh to zero and to one for the other half – the blue region and orange regions. On the lower right of figure 4.9 is the outcome, a decimated mesh with genus null.

<sup>63</sup> Retriangulating holes in surface meshes, is by no means a trivial task and many papers have been published, devoted to answering this question [cf. Zhao et al., 2007]. Yet, we assume that any hole filling happens at places of minor importance, typically to mend the mesh after an unimportant feature has been deleted. Additionally we don't care for the quality as long as it produces a water-tight sealing. Consequently we do not deploy a sophisticated scheme but instead project the boundary onto a plane and use a 2D triangulation algorithm to solve the problem.

### 4.2.3. Limitations

There are two limitations to consider when working with the current implementation. The first is, that it was designed to work with one single connected component at a time. If a mesh gets partitioned into two individuals parts, the program has no way to display the topological invariants individually. Reckoning the change, however is trivial as  $\beta^0 > 1$ . Interestingly enough though, the pairing algorithm does still work and the values for the Betti numbers represent the summed up results, see figure 4.10. Note that although the Betti numbers are correct, the displayed genus for the second mesh is wrong. At this point, reconnecting parts that were split before, is not supported.

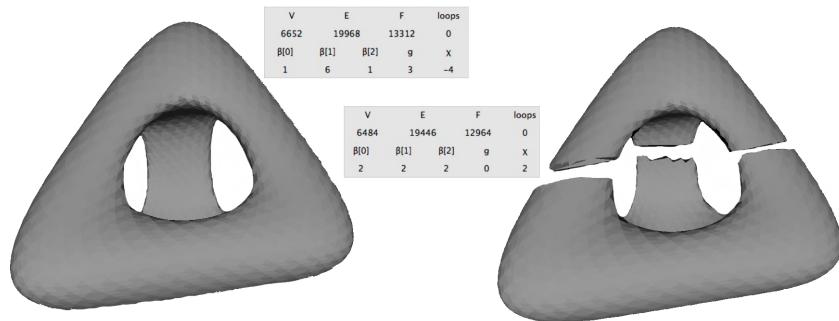


Figure 4.10.: A mesh  $g = 3$  that gets severed into two parts, with  $g_1 = 0$  and  $g_2 = 1$ .

The second restrictions concerns not orientable surfaces. This is more of a barrier, as the pairing algorithm completely fails and even displays boundary vertices for the completely closed mesh. Even our viewport fails in the face of this challenge, when asked to render triangles, as seen on the right sight of figure 4.11. For that matter, the TopStoc algorithm can't be convinced to work with this type of meshes either.



Figure 4.11.: The Klein bottle withstands filtration and decimation by our methods.

### 4.3. Drawing on Meshes

We now turn to the very heart and epitome of user-guidance, that is the explicit control over the decimation process<sup>64</sup>.

Although a major aspect of our work and an extremely powerful tool to selectively increase the fidelity of a mesh, the concept and implementation with “TopStoc” is strikingly unproblematic. This is due to the fact that other decimation techniques are based on intangible error-bounds or priority queues, whereas our approaches rests on stochastics which are intuitively clear. This way, instead of just attributing certain regions of a mesh with immunity to decimation, or fiddling with abstract rules and weights, the user can just literally ‘sketch probabilities’ onto the base mesh. In the example of figure 4.12 the

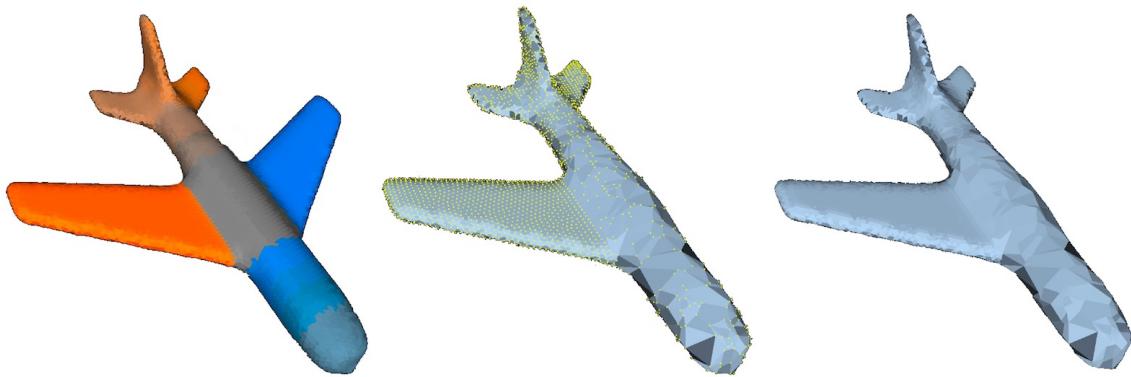


Figure 4.12.: Vertices on the wings of the plane are either kept or dismissed entirely (bright orange or dark blue). The body gradually gains geometric detail from top to back. On the plane in the middle, the then sampled vertices are shown and on the right is the final mesh alone.

various parts of the plane model were painted with different probability levels. Orange depicts values of relatively higher probability and blue levels of lower probability in respect to grey, which always conforms to the global target size that was set by the user. Having the same effect as if each part of the model was processed individually with its own decimation ratio.

This means that, for example, if the user adjusts the slider to a 0.5 the orange parts cover decimation ratios between [0.5, 1.0] and the likewise the blue areas cover [0.5, 0.0]. This

<sup>64</sup> User-guidance, somewhat unexpectedly, hasn't been of much interest in the last years as far as research papers are concerned. The work that still gets published, mostly reprises unimaginative adaptations of known decimation schemes to work with user input [cf. Ho et al., 2006]. This is astounding as the very first work already set a different trajectory by discussing among other things response times, i.e. work-flows [cf. Cignoni et al., 1998a].

seems to be problematic if the overall decimation ratio closes in on either boundary 1.0 or 0.0 and the colors encode less and less equal parts of the entire decimation range. However, the design was deliberately chosen, since this way one has a unambiguous and obvious feedback which parts were tweaked by the user to use less geometric detail and which parts needed to be boosted. Furthermore the application offers the user a finely partitioning of each side, orange and blue, so that even in the most extreme cases the adjustability is ensured.

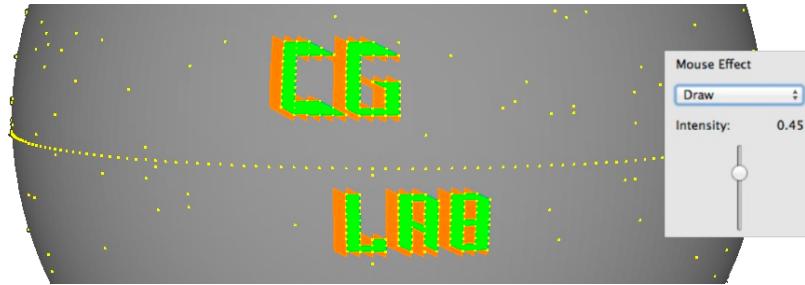


Figure 4.13.: Besides the fuzzy painting of entire regions, the user has also the option to select a set of single elements and set their probability individually.

Note that the incorporation user-guidance does not impact the speed of the algorithm, as it only adds additional information for each vertex but does not require additional processing. It is important to mention though, that once a region is augmented with user-set probabilities, the resulting mesh size after the decimation will deviate from the target size. User-input always trumps automatic adjustment and enforcing the precise target size would lead to higher runtimes.



Figure 4.14.: Another example of how user-assistance can lead to better results. Left is the unaltered result, right the improved version, note that both versions were sampled to the same vertex count  $\pm 1\%$ .

## 4.4. Back-face Subsampling & Silhouette Preservation

View-dependent decimation reckons that most of the time, only a fraction of the geometric detail is visible to us and that a good decimation scheme should exploit this. To leverage this fact, at least the direction of the camera is needed, achieving its full potential if a complete frustum is defined, as model parts outside the frustum can be dismissed entirely. Together with the view-direction comes the concept of silhouettes, which are of pivotal importance for the perception of detail and geometric fidelity. As early as 1996 appeared papers dealing with simplification methods that paid special attention to these details [cf. [Xia and Varshney, 1996](#), in which, not only view-dependencies but also decimation adapting to lighting conditions, gets discussed].

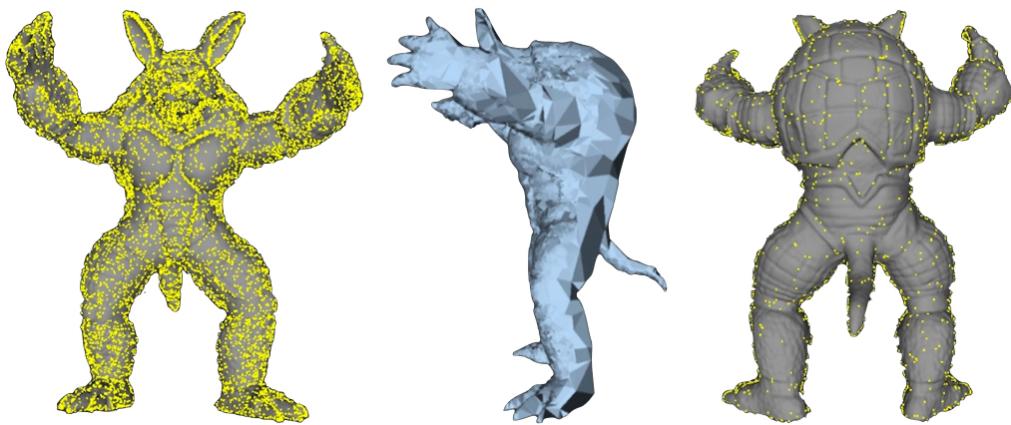


Figure 4.15.: The *Armadillo* model, showing the sampled vertices on the front and the back, with the resulting mesh in the middle.

As in the case with user set decimation ratios, once again the “TopStoc” algorithm lends itself almost effortless to an adaption for silhouette preservation and back-face culling. Remember that the weight for vertices, as introduced in equation 4.1 already uses normals. Therefore this technique more or less inevitably suggests itself to be adapted, as soon as a view-vector  $\mathbf{v}_{cam}$  is defined:  $\mathbf{x}_{view}(\mathbf{v}) = (\mathbf{n}_v^T \mathbf{v}_{cam})/2$

Of course it is possible to introduce  $\mathbf{v}_{cam}$  directly to equation 4.1, not adding it as an extra value to each vertex like we propose. This undeniably would lessen the computational impact, offering back-face subsampling and silhouette preservation at no additional cost. However this also means that all given weights have to be recalculated every time  $\mathbf{v}_{cam}$  changes, even if only a normal decimation is desired.

In figure 4.15 the effect for a considerable lower sample rate on the back-face is shown. Making use of a point of view aware decimation immediately drops the geometric complexity nearly by half, for the same accuracy of visible parts<sup>65</sup>. To give a further impression of the results, we visualized the errors per triangle for the decimated mesh from figure 4.15, see figure 4.16.

Another interesting result is that since the “TopStoc” algorithm preserves the topology of an object we can cull faces on the backside of models altogether, still ending with a water-tight mesh at the end. However, it is desirable to preserve vertices that lie on the silhouette to avoid artifacts that could otherwise result from the culling.

Silhouette preservation in itself is not an independent step as exactly the same normal calculation takes place. The difference is that values close to zero are dealt with more attention. Also for the practical applications it is necessary to define an interval for a graceful diminish, see the model on the right in figure 4.17. This amends the problem that the area at the silhouette is not necessarily friendly towards such a treatment, unless the body is perfectly concave, obviously.

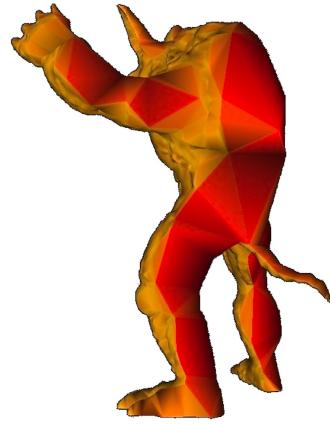


Figure 4.16.: Errors per Triangle.

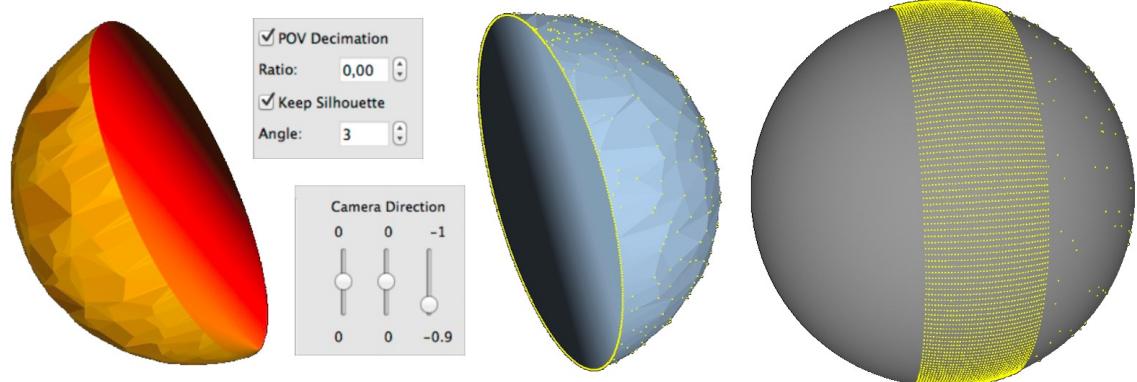


Figure 4.17.: Back-face culling and silhouette preservation are demonstrated on a sphere. On the left is the error visualization, next to it the GUI panel for the feature, in the middle is the decimated mesh with the sampled vertices colored and on right, the same model with a bigger aperture of the silhouette.

---

<sup>65</sup> Depending on the geometry and the extend to which the object is concave – with most organic models being very suitable, but hard surface models posing a bigger challenge.

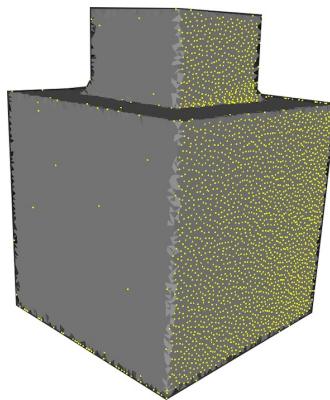


Figure 4.18.: Oversampling on ill-formed silhouette areas.

Unfortunately there are bad cases in which the preservation of the silhouettes not necessary fails, but yields really unwanted results. Figure 4.18 shows such a situation in which the silhouette coincides with a planar region, preserving way too many vertices. This problem can be solved though by user intervention. Painting with value -1, i.e. the dark blue, overrides any sampling decisions by the the algorithm. All in all are the tools connected with the point of view a very helpful addition at negligible costs.

## 4.5. Textures

A key component in making any decimation method feasible for the actual adaptation in other field besides academia is the support for texture handling. The role that textures play in fields like *Computer Games* or the *VFX* industry can not be overstated. Actually the success of wrapping geometry with 2D images for various scenarios like bump-mapping, displacement information, dirt layers, etc. has to some extend fallen victim to its own success, but although more sophisticated solutions are available, like Ptex<sup>66</sup>, conventional textures will be around for years to come [cf. Burley and Lacewell, 2008, Ptex opposed to MeshColors is a fully-fledged concept that has been already successfully used].

Because of this, it was important for us to include texture handling and support. The problem when decimating meshes that normally hold textures is that highly visible artifacts appear if the borders of the so called UV patches are not respected.

<sup>66</sup> The idea behind this concept proposed by Burley and Lacewell, is to assigns an entirely separate texture per face, skipping the texture unwrap entirely. The technique uses adjacency data to filter across faces, removing otherwise visible seams.

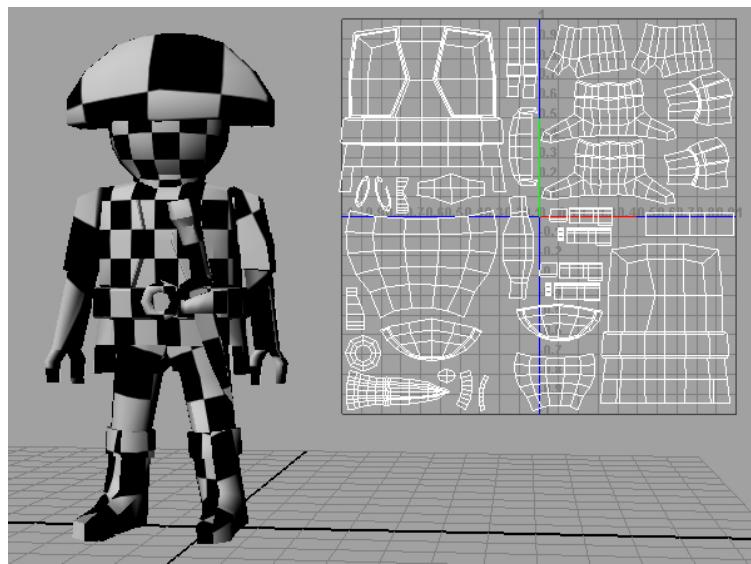


Figure 4.19.: Model with UV layout, image courtesy of [World Of RaftCraft](#).

Figure 4.19 shows a model together with its UV layout. Note that color between the patches is normally black, so what one will typically see if a UV border vertex gets deleted is that the model has strange dark patches on its surface.

The easiest remedy to tackle this problem is to check which vertices have multiple texture coordinates as this defines the vertices on the border of UV patches. Any of these can subsequently set to “untouchable” to ensure that no border gets lost. This is what we implemented and is shown in figure 4.20.



Figure 4.20.: Modell “Mini Cooper” courtesy of [Oyonale](#).

The Mini on the left is the base mesh, on the right is the decimated version, note the high vertex density around the doors where a UV border resides. A more sophisticated approach is to flood-fill the vertices of each patch to know, not only if a vertex lies on

the boundary but also which specific patch it belongs to. This in turn makes it possible to redraw the texture and avoid artifacts altogether, well besides artifacts of distortion which would require even more subtle handling<sup>67</sup>.

## 4.6. Bounding Boxes & Materials

Last but not least is the support for bounding boxes and materials definitions. These two items are not orthogonal to the rest of the tool set, so to say. They *only* facilitate the work for the user when dealing with models, but the effects could also be achieved if only using the previously described methods.

Support for bounding boxes entails the option to specify a series of cuboids. Vertices that happen to be in any of such a volume get assigned one common weight. This for instance is helpful to define frustums that cull an object.

Considering materials has a similar effect only that the familiarities of vertices are not defined by proximity but instead by the material definition given in the source file. Almost all standards for geometry representation include such material definition and are normally used for rendering and shader binding.

---

<sup>67</sup> At this moment our reference implementation supports the preservation of vertices that have multiple texture coordinates, but we haven't pursued any further so far.

## 4.7. Measurements

The evaluation of the results falls into two domains, on the one hand there is the individual perception of the user and on the other hand there are precise mathematical error bounds. The user will typically make the final decision whether a certain asset holds up to his or her standards judging by the optical impression. Yet, it is also important to be able to get unequivocal data to rely on, for instance if simulations are involved.

We want to cater both scenarios as they are equally important. Consequently we implemented one distinct tool for each one of them, not only to give feedback, but also to help guiding an iterative workflow.

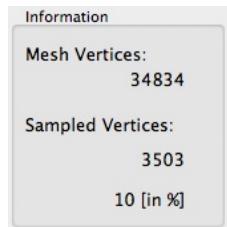


Figure 4.21.: Basic information, vertex count of the meshes.

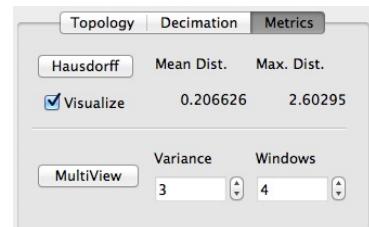


Figure 4.22.: Metrics tab with the 'Hausdorff' and 'Multiview' section.

The basic information of the amount of vertices in the base mesh and the decimated mesh, is of course always shown. Additionally, the tab labeled 'Metrics' offers the calculation and visualization of the Hausdorff distance, as well as the 'Multiview' tool which spawns new instances of the current decimation settings with limited variation, which will be both described in sections 4.7.1 and 4.7.2.

Another useful utensil in assessing the decimation quality, is the turntable button. When pressed, the mesh is placed in the center of the screen and the camera then gyrates two times around the model in an upwards spiral to show all the mesh from all direction.

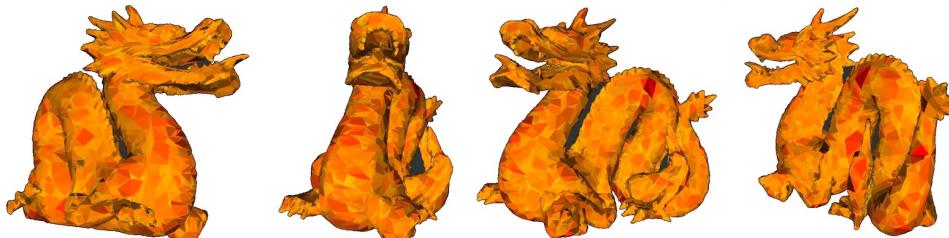


Figure 4.23.: Turntable of the decimated dragon mesh, with Hausdorff distance visualized.

### 4.7.1. Hausdorff Distance

In the case of 1D-signals and images, many distortion measurements have been studied. They range from simple analytic methods such as the mean square error to very elaborate techniques based on the characteristics of human perception [cf. [Winkler, 2001](#)]. Despite the constantly growing usage of 3D models, the distortion measurements for such data seem to be less thorough. Still the predominant approach to compare two meshes is the *Hausdorff distance*<sup>68</sup>, defined as:

$$d_H(\mathcal{M}, \mathcal{M}') = \sup_{p \in \mathcal{M}} \inf_{p' \in \mathcal{M}'} d(p, p') \quad (4.3)$$

respectively the symmetric version:

$$d_{H_{sym}}(\mathcal{M}, \mathcal{M}') = \max\left\{\sup_{p \in \mathcal{M}} \inf_{p' \in \mathcal{M}'} d(p, p'), \sup_{p' \in \mathcal{M}'} \inf_{p \in \mathcal{M}} d(p', p)\right\} \quad (4.4)$$

Where the distance  $d(p, p')$  between two points  $p$  and  $p'$  on the surfaces of the mesh  $\mathcal{M}$  and it's simplified version  $\mathcal{M}'$  is defined as:  $d(p, p') = \|p - p'\|$  with  $\|\cdot\|$  denoting the usual Euclidean norm. We will refer to  $d_H(\mathcal{M}, \mathcal{M}')$  as the forward distance, and to  $d_H(\mathcal{M}', \mathcal{M})$  as the backward distance.

The distance between any point  $p$  belonging to  $\mathcal{M}$  and  $\mathcal{M}'$  can be computed analytically, since it can be reduced to the minimum of the distances between  $p$  and all the triangles  $\mathcal{F}_{\mathcal{M}'}$ . It is worth noting that  $p$  might not be a vertex of the list  $\mathcal{V}_{\mathcal{M}}$  but any point on the surface. Hence, although straightforward, the algorithm becomes too complex if implemented naively<sup>69</sup>. To achieve reasonable running times we followed the implementation strategy described in the paper “*Mesh: Measuring errors between surfaces using the hausdorff distance*”, where a cell partitioning strategy is used to greatly reduces the number of point-triangle distance evaluations. Each triangles is sampled via a regular grid according to a length criterion, thus avoiding uniform sampling [for more details see [Aspert et al., 2002](#), especially chapter 3].

<sup>68</sup> First introduced in the book *Grundzüge der Mengenlehre*, published in 1914 and named after the German mathematician *Felix Hausdorff* (1868 – 1942), one of the founders of modern topology. Technically the Hausdorff distance it is not a real distance function since it lacks symmetry. It is better to think of it in terms of set containment, for more information and examples see Appendix A.5.

<sup>69</sup> In fact, for each sample point  $p$  it would be necessary to calculate the distance to all triangles  $\mathcal{F}_{\mathcal{M}'}$  in order to find the minimum. This leads to a complexity  $\mathcal{O}(|p| \cdot |\mathcal{F}_{\mathcal{M}'}|)$ , where  $|p|$  is the number of sample points taken on  $\mathcal{M}$  and  $|\mathcal{F}_{\mathcal{M}'}|$  is the number of triangles.

By pressing the 'Hausdorff' button the backward distance is calculated and the numerical values for mean and maximum distances are shown. Since the algorithm maintains a 1-1 correspondence between the sampled points and the base mesh, calculating the forward distance would yield little information and is omitted to save time.

More important for the practicality though, is that the normalized error values can be used to colorize the decimated mesh. Thereby giving the user a clear representation of errors in the form of darkened triangles. In image 4.24 the upper sphere shows the gradually loosening of fidelity on the base mesh in colors. Starting at the orange section on the upper left, where almost all vertices are sampled, the sampling rate steadily diminishes until it is below 1% in the dark blue area. The second lower sphere shows the decimated mesh with the visualization of the Hausdorff distance at each triangle.

The user can then switch between the two visualization modes and selectively set control points on the base mesh to improve the quality in certain regions that would be otherwise undersampled. In figure 4.25 the mesh is sampled at the same decimation rate of about 9,000 vertices two times, only deviating at less than 1% of the total number. However, for the second sampling, a series of control points were inserted in the areas of the cheek, forehead and the throat. Giving a better overall result for the mean error of the Hausdorff distance.

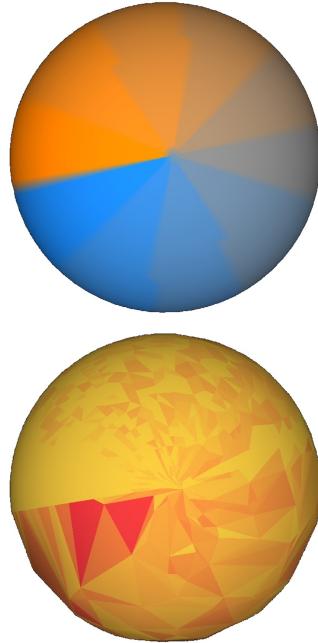


Figure 4.24.: Visualization  
Hausdorff.

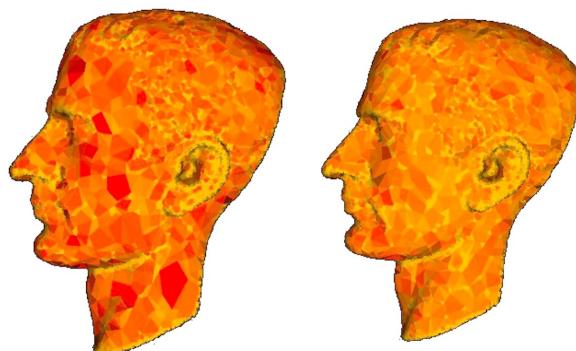


Figure 4.25.: Two versions of a bust, with the same vertex count, but on the left hand side without control points to avoid undersampling.

### 4.7.2. Multiview

With the Hausdorff distance at his disposal, the user is equipped to make very precise augmentations to a mesh and evaluate them quickly. This complements the principle gauge tool for any decimation, that is the target size. Even so, we think, that another tool has its place in this ensemble.

Multiview rests on the idea that it can be very beneficiary to get quick iterations with some variation of one and the same result. The sampling algorithm is inherently stochastic and therefore will produce different versions albeit respecting constrains like user-set control points and the target size of the mesh. We use this fact to offer the user the chance to create multiple instances of his decimation with just one single click.

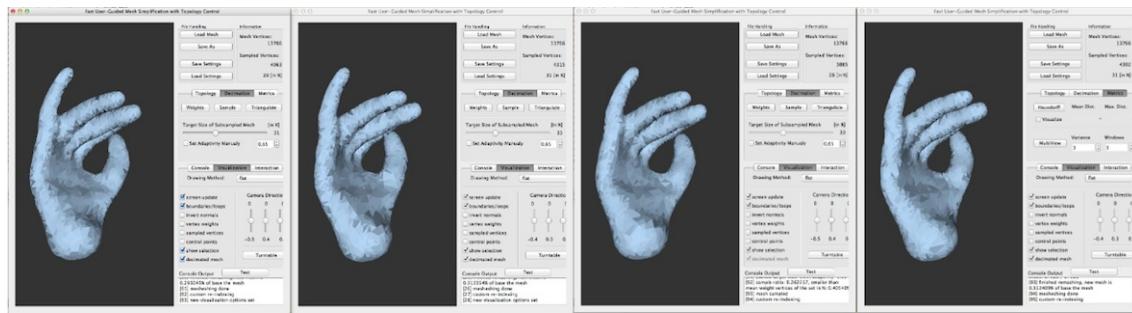


Figure 4.26.: Four versions of a mesh with 3% variance of the number of vertices.

There are only two input variables that have to be defined, that is the maximal allowed variation in terms of % mesh size and the number of instances that should be generated. Using Multiview is possible because sampling and remeshing are comparatively inexpensive. However with very large models it is advisable to restrict the newly instantiated windows to a small number.

Once the different models are created the user immediately sees whether it makes sense to persuit even lower decimation rates or if he already is at a threshold of accuracy he does not want to cross. Besides giving fast visual feedback, Multiview can also be used to rapidly save multiple LOD versions if the variance is set to high percentages.



---

## 5. Conclusion

*“... in life we ultimately pursue, not conclusions, but beginnings.”*

– Sam Tanenhaus [American historian, in: [Tanenhaus, 1986](#)]

### 5.1. Summary

In the beginning we tried to clarify why mesh decimation is and will continue to be an important issue, as model complexity is still on the rise. We also mentioned the importance of work-flows and time regimes that often get overlooked but fundamentally determine how a tool is seen and used.

In the second chapter, we looked at the spectrum of available concepts to simplify geometry and ways to categorize them. This lead to a taxonomie and a checklist in which we formulated our goals. But before we could evaluate our results, we gave an introduction to the mathematical concepts that were necessary in order to master the topological challenges. In contrast to most of the previous work, we decided to implement a mathematical more demanding approach that relies on persistent homology but guarantees to find all topological relevant features.

Although this gave us very good results, it turned out that calculating the defining loops, breaks the otherwise fast decimation pipeline. Fortunately we are still equipped with all the necessary information, as single edges take on the role of indicating topology. Nevertheless we think that the processes can still be improved and might turn out to be viable in terms of time complexity after all, see [5.2.2](#).

Besides the topology we also tried to make sensible additions to the base algorithms and extend the scope of application to take full advantage of user input. Additionally we addressed some practical issues like how to best measurement decimation, support

textures, or in cooperative point of view decimation – which fit really nice to the decimation pipeline.

As a result, “TopStoc” in its current iteration not only offers fast and robust automatic decimation, but can take full advantage of user-guidance. In the interplay between the program and the user, the user is not restricted to some selected features for improving the results but instead really has full control over the entire decimation process, as well as the topology!

## 5.2. Future work

One can think of numerous little improvements and extras, that would help the workflow. Among them are additional tools for topology control or mesh manipulation. The optimization and the parallelization of the algorithms, also seem to pose legitimate research question.

Having said that, we want to focus the discussion on two entirely different issues. The first question in section 5.2.1 is motivated by the standards used in the industries, which are currently dealing with biggest assets and therefore would be the most likely candidates to adopt any of the presented results. In short, the problem is, that for various reasons quad meshes are the quasi norm for many applications. Yet it is not clear how the remeshing part could be adopted in a smart manner to provide for quad meshes.

The second question in section 5.2.2, reviews ideas of how the calculation of the Betti numbers could be sped up. The main idea being, that a locally restricted algorithm would not only decrease running times but also improve the geometrical significance of our results.

### 5.2.1. Adaption for Quad-Meshes

The problem of generating high quality quad meshes from triangle meshes or point cloud data has been fairly well researched. Including several surveys on the topic [cf. [Alliez et al., 2005](#); [Hormann et al., 2007](#)]. The reason for the interest in quad meshes is, that it converts raw geometric data into a higher representation that is suitable for sophisticated operations like texturing, subdivision and compression.

Constructing good meshes is still hard task though, not only are there a series of quality criteria like ‘edge-flow’, ‘individual element quality’, etc. but their optimization often

requires the consideration of global dependencies. Therefore to implement a naïve quad meshing scheme after the stochastic vertex selection phase should be rather straight forward [cf. [Li et al., 2009](#)], but doing it with good results and some flexibility in terms of what criteria to optimize isn't [[Bommes et al., 2009](#)]. More importantly, whatever strategy gets deployed, it should take full advantage of the levers that are already in place. What we mean by that is, that there are parts like loop computation that could be exploited to guide the meshing scheme.

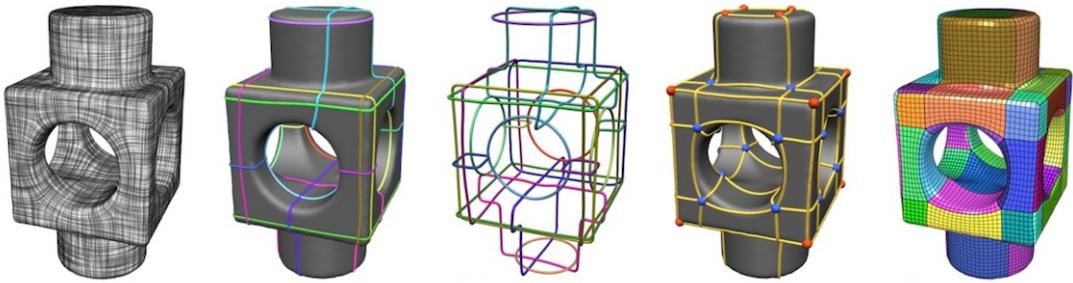


Figure 5.1.: BLA [cf. [Campen et al., 2012](#)].

Just now, [Campen et al.](#) presented a novel approach, called “Dual Loops Meshing”, to get high quality, coarse, all-quadrilateral patch layouts on manifold surfaces [[Campen et al., 2012](#)]. Their method uses a field of principal curvature directions, to get a collection of transversally crossing loops on the surface. These loops partition the surface into polygonal regions whose valences are connected to their total curvature, see figure 5.1. And although we haven't analyzed the actual familiarities between their concept and the techniques we use, it is clear that they are related and could generate synergies.

### 5.2.2. Fast $\beta^1$ Calculation via Decimation & Segmentation

Throughout the thesis we calculated all the Betti numbers for entire meshes, yet this seems to be unnecessary wasteful since large parts of the model are just not important, i.e. bearing  $\beta^1 = 0$ . Especially as the used pairing algorithm, described in [3.5](#), does not scale very well. The reason for that is, that the expansion of chain complexes in the `while` loop gets considerably more computational expensive as the size of the simplicial complex  $K$  increases and with it, its chains. We think that there could be ways to exclude the pricy calculation of  $\beta^2$  in favor of smart heuristics, thereby gaining huge speed-ups. Even if that should turn out, not to work, two further ideas could yield valuable improvements.

First it could be tried to quickly segment the mesh into topological meaningful parts. Ideally one would try to dismiss as much segments, with genus null, as possible. Thus, leaving only a smaller subset for the more involved computations. This should be feasible since checking the topological importance is very cheap and calculated via the linear sum over the simplices. Breaking up the problem into smaller parts provides further advantages for later stages of the process. In case of searching for handle and tunnel loops, this would also make the tetrahedralization faster, and place the generating edges closer to the features they represent. On top of all of that, could the smart segmentation of a mesh lead to a parallelization of the algorithm that might even lends itself to a GPU implementation – altering the running times of these computations rather orders of magnitude than just some percentages.

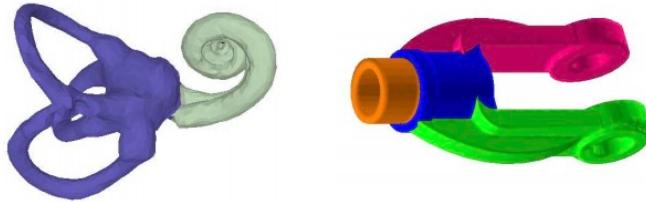


Figure 5.2.: Example of topological meaningful mesh segmentations [Attene et al., 2006, Figure 7 & 9].

The second idea plays on the fact that topology does not care about the actual size of simplices but rather their connectivity. Consequentially all invariants are retained even if the mesh has undergone drastic simplifications, as long as the simplification process preserves the topology. On that account, accelerating the computation can be achieved by adding a step in the algorithm that strongly decimates the mesh upfront. Equipped with a homeomorphic low polygonal representation, the actual computation of the invariants can be done much faster. This could make it even practically relevant to drop the tetrahedralization part of the handle and tunnel loop computation all together, which in turn, could lead to better results for knotted surfaces which present a problem for the current approach. The interesting features of the low poly model, of course, would have to be mapped to the highly detailed original mesh afterwards. Possibly this can result in the necessity to do a further clean-up process of the mapped data, but even so this route seems to be very promising.

A preliminary search of the topic showed that, fast algorithms for the calculation of Betti numbers have been of interest for researchers since at least the early nineties [cf. [Delfinado and Edelsbrunner, 1993](#)]. However, simplification or segmentation as a preprocess hasn't been prominently featured in any of the works that we found so far. Which is somewhat striking, for the reason that Betti numbers are often used in context of decimation. It seems that papers in this research area are either more concerned about other aspects, like for instance 'Contour Trees', or restrict their applications to isosurfaces, alpha shapes and so on [cf. [Pascucci and Cole-McLaughlin, 2003](#); [Konkle et al., 2003](#)].



## List of Figures

1.1.	Teapot rendering scene as presented in <i>Martin Newell's</i> PhD dissertation, published in 1975 [cf. <a href="#">Torrence, 2006</a> ]. . . . .	2
1.2.	The original wireframe 'Utah Teapot' (28 bicubic Bézier patches, with 16 control points each) [cf. <a href="#">Newell, 1975a</a> ]. . . . .	2
1.3.	Example of highly detailed surface – note that the houndstooth pattern of the jacket is genuine geometric data [Credit: Pixar 2009]. . . . .	3
1.4.	To leave no evidence that surfaces are discretely approximated, film models consist of micropolygons, about a pixel in area [ <a href="#">Fatahalian, 2011</a> , p.7]. . . . .	3
1.5.	Renderings of a stone house using a high-resolution (top) and low-resolution (bottom) triangle mesh (house image by Unigine Engine). . . . .	4
1.6.	3D mesh reconstructions from Internet photo collections, that typically can have dozens of millions of triangles [ <a href="#">Snavely, 2008</a> , p.6]. . . . .	6
1.7.	Response time to user input vs. workflow. . . . .	8
2.1.	Example from the original paper, left full resolution (569k triangles) and right the 90% decimated version (142k triangles) [ <a href="#">Schroeder et al., 1992</a> , p.68]. . . . .	12
2.2.	Re-tiling of a radiation iso-dose surface. Upper left: Original surface. Upper right: Candidate vertices after point-repulsion. Lower left: Mutual tessellation. Lower right: Final tessellation [ <a href="#">Turk, 1992</a> , p.58]. . . . .	13
2.3.	Overview of the simplification process [ <a href="#">Rossignac and Borrel, 1993</a> , p.458]. . . . .	13
2.4.	$2\epsilon$ is the user specified angle describing the maximum angular difference between coplanar sets selected to be merged [ <a href="#">Hinkler, 1993</a> , p.192]. . . . .	14
2.5.	The different cases for a triangle platelet $P_i$ , i.e. the data stencil in the surface triangulation affected by the removal of triangle $T_i$ – platelets with cyclic, disconnected and connected corona [ <a href="#">Hamann, 1994</a> , p.200]. . . . .	14
2.6.	Left: Base mesh (162 triangles). Middle: Original mesh (69,437 triangles). LOD using multiresolution approximation [ <a href="#">Eck et al., 1995</a> , p.181]. . . . .	15
2.7.	Original illustration of the edge collapse transformation respectively the vertex split and the resulting vertex correspondences [ <a href="#">Hoppe, 1996</a> , p.100]. . . . .	15
2.8.	Approximation with inner & outer envelopes [ <a href="#">Cohen et al., 1996</a> , pp.123-124]. . . . .	16

2.9.	98,5% decimation of the stanford bunny. The error ellipsoids for each vertex are shown in green [Garland and Heckbert, 1997, p.215]. . . . .	17
2.10.	The optimal vertex position $v$ expressed as the intersection of three planes that ensure volume preservation [Lindstrom and Turk, 1998, p.284]. . . . .	17
2.11.	Explicit deletion of a vertex on the mesh [Cignoni et al., 1998a, p.312]. . . . .	18
2.12.	Left: Original mesh and its geometry image. Right: Reconstruction of the original and a 1,5kB compressed version mesh [Gu et al., 2002, p.356]. . . . .	18
3.1.	Connectivities of prominent 2D manifolds. . . . .	23
3.2.	A point-, topological- and metric-space. . . . .	25
3.3.	Complete homeomorphism list for compact, boundaryless 2-dim. manifolds. . . . .	27
3.4.	Oriented $n$ -simplices, $0 \leq n \leq 3$ . . . . .	28
3.5.	One simplicial complex and three unions that are not complexes. . . . .	29
3.6.	Orientation. . . . .	29
3.7.	Non-orientable surfaces . . . . .	29
3.8.	A double torus, as the connected sum of two tori. . . . .	30
3.9.	A famous example for homotopic equivalence are a coffee cup and a torus. . . . .	31
3.10.	One boundary loop (on the left) and two non-bounding loops on a torus. . . . .	32
3.11.	The chain complex for a 3-dimensional simplicial complex. . . . .	34
3.12.	Morse theory – starting from the bottom the homotopy type of the torus changes at its critical points from: point or disk, to cylinder, continues to a torus with a removed disk boundary, until it is finally a full torus. . . . .	37
3.13.	Handles (green) and tunnels (red) with their defining loops [Dey et al., 2012]. . . .	39
3.14.	Left: A non-trivial loop that is neither a handle nor a tunnel, Right: Loops on knotted surface reject the divide in $H^1(\mathbb{I})$ and $H^1(\mathbb{O})$ [Dey et al., 2012]. . . . .	40
3.15.	Example of a filtration sequence, building a tetrahedron with flag triangle. . . . .	43
3.16.	Persistence of the simplices . . . . .	44
3.17.	Left to right: Unpaired edges,handle loops and tunnel loops [Dey et al., 2012]. . . .	47
3.18.	Initial results and refined loops on the right [Dey and Li, 2009]. . . . .	49
4.1.	Results from our TopStoc implementation, left to right: The base mesh; visualization of the local normal deviation; sampled vertices that define the low-resolution mesh; visualization of the topological clustering; the final mesh. . . . .	52
4.2.	The Y-axis shows the relative improvement in % of the runtime for two models (green/blue), for decimation ratios between 5% and 1% on the X-axis. To demonstrate the effect, the two bunnies show the discarded faces in black, for a very high and a very low sampling. . . . .	55

4.3. Normal vs. adaptive density control. Whereas the sampling left shows almost uniform distribution, the adaptive version retains higher vertex counts in important regions without loosing good aspect ratios of the triangles. Note how the triangle size changes between forehead and beard. . . . .	56
4.4. Relative triangle quality, Y-axis, of normal decimation (blue) vs. density control (green), over small decimation ratios X-axis and an example of a bad triangle shape that can't be caught by density control. . . . .	57
4.5. The model <i>Casting</i> , of genus 9 with the handle and tunnel loops inducing edges visualized. . . . .	59
4.6. Two ornamental models with high genera and their unpaired edges colored. . . . .	60
4.7. A tunnel (turquoise) and handle (yellow) loop of a simple double torus with its paired edges in dark blue. . . . .	61
4.8. Model <i>Mother</i> with its tunnel (turquoise) and handle (yellow) loops. . . . .	61
4.9. Topological operatons on a double torus. . . . .	63
4.10. A mesh $g = 3$ that gets severed into two parts, with $g_1 = 0$ and $g_2 = 1$ . . . . .	64
4.11. The Klein bottle withstands filtration and decimation by our methods. . . . .	64
4.12. Vertices on the wings of the plane are either kept or dismissed entirely (bright orange or dark blue). The body gradually gains geometric detail from top to back. On the plane in the middle, the then sampled vertices are shown and on the right is the final mesh alone. . . . .	65
4.13. Besides the fuzzy painting of entire regions, the user has also the option to select a set of single elements and set their probability individualy. . . . .	66
4.14. Another example of how user-assistance can lead to better results. Left is the unaltered result, right the improved version, note that both versions were sampled to the same vertex count $\pm 1\%$ . . . . .	66
4.15. The <i>Armadillo</i> model, showing the sampled vertices on the front and the back, with the resulting mesh in the middle. . . . .	67
4.16. Errors per Triangle. . . . .	68
4.17. Back-face culling and silhouette preservation are demonstrated on a sphere. On the left is the error visualization, next to it the GUI panel for the feature, in the middle is the decimated mesh with the sampled vertices colored and on right, the same model with a bigger aperture of the silhouette. . . . .	68
4.18. Oversampling on ill-formed silhouette areas. . . . .	69
4.19. Model with UV layout, image courtesy of <a href="#">World Of RaftCraft</a> . . . . .	70
4.20. Modell "Mini Cooper" courtesy of <a href="#">Oyonale</a> . . . . .	70
4.21. Basic information, vertex count of the meshes. . . . .	72
4.22. Metrics tab with the 'Hausdorff' and 'Multiview' section. . . . .	72

4.23. Turntable of the decimated dragon mesh, with Hausdorff distance visualized. . . . .	72
4.24. Visualization Hausdorff. . . . .	74
4.25. Two versions of a bust, with the same vertex count, but on the left hand side without control points to avoid undersampling. . . . .	74
4.26. Four versions of a mesh with 3% variance of the number of vertices. . . . .	75
5.1. BLA [cf. <a href="#">Campen et al., 2012</a> ]. . . . .	79
5.2. Example of topological meaningful mesh segmentations [ <a href="#">Attene et al., 2006</a> , Figure 7 & 9]. . . . .	80
A.1. Microprocessor transistor counts 1971-2011 & Moore's Law . . . . .	i
A.2. 20 Years of Pixar's RenderMan - 1 . . . . .	ii
A.3. 20 Years of Pixar's RenderMan - 2 . . . . .	ii
A.4. Facsimile of the original 'Utah Teapot' drawing. . . . .	iii
A.5. The 'Teapotahedron' as the sixth platonic solid. . . . .	iii
A.6. Homology cycles on a torus (image copyright wikipedia). . . . .	iv
A.7. Asymmetry of the one-sided Hausdorff distance $d(\mathcal{S}, \mathcal{S}') \ll d(\mathcal{S}', \mathcal{S})$ . . . . .	v
A.8. Decomposition of the retractable surface into its parts. . . . .	vi
A.9. Left: A complete 6-graph; Right: Möbius band with dotted meridian. . . . .	vii
A.10. A Möbius band built from a complete 6-graph, K6. . . . .	viii

## **List of Tables**

0.1. Definitions . . . . .	II
2.1. Taxonomy from Boubekeur & Alexa [ <a href="#">Boubekeur and Alexa, 2009</a> , p.2]. . . . .	20
2.2. Checklist for simplification algorithms. . . . .	22
3.1. Classes of 2-dimensional compact manifolds with low genus. . . . .	27
3.2. Fundamental group for various spaces. . . . .	33
3.3. Homology groups for the basic surfaces. . . . .	35
3.4. Algorithm – Betti numbers of simplicial complexes. . . . .	42
3.5. Algorithm – Simplex pairing. . . . .	48
4.1. Algorithm – Breadth-first flood-filling. . . . .	54



---

## Bibliography

Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-time Rendering*. A K Peters, Wellesley, 3rd. edition, 2008. ISBN 978-1568814247. [1.2](#)

Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. Recent Advances in Remeshing of Surfaces. Technical report, AIM@SHAPE Network of Excellence, 2005. URL [http://graphics.stanford.edu/courses/cs164-10-spring/Handouts/recent\\_advances.pdf](http://graphics.stanford.edu/courses/cs164-10-spring/Handouts/recent_advances.pdf). [5.2.1](#)

Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *ICME*, pages 705–708, Lausanne, 2002. IEEE Computer Society. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1035879](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1035879). [4.7.1](#), [A.5](#)

M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh Segmentation - A Comparative Study. In *International Conference on Shape Modeling and Applications*, page 12. Ieee, 2006. ISBN 0-7695-2591-1. doi: 10.1109/SMI.2006.24. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1631191>. [5.2](#), [5.2.2](#)

Chandrajit L Bajaj, Valerio Pascucci, and Guozhong Zhuang. Progressive Compression and Transmission of Arbitrary Triangular Meshes. In *VIS*, VISUALIZATION ’99, pages 307–316, Los Alamitos, 1999. IEEE Computer Society. ISBN 0-7803-5897-X. URL <http://dl.acm.org/citation.cfm?id=832273.834164>. [2.2](#)

Michael Bartoň, Iddo Hanniel, Gershon Elber, and Myung-Soo Kim. Precise Hausdorff Distance Computation between Polygonal Meshes. *Computer Aided Geometric Design*, 27(8):580–591, 2010. URL <http://www.sciencedirect.com/science/article/pii/S016783961000049X>. [A.5](#)

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *Transactions on Graphics*, 29(4):1, July 2010. ISSN 07300301. doi: 10.1145/1778765.1778800. URL <http://portal.acm.org/citation.cfm?doid=1778765.1778800>. [12](#)

David Bommes, Henrik Zimmer, and Leif P. Kobbelt. Mixed-Integer Quadrangulation. *Transactions on Graphics*, 28(3):1, July 2009. ISSN 07300301. doi: 10.1145/1531326.1531383. URL <http://portal.acm.org/citation.cfm?doid=1531326.1531383>. [5.2.1](#)

Tamy Boubekeur and Marc Alexa. Mesh Simplification by Stochastic Sampling and Topological Clustering. *Shape Modeling International*, 2009. [2.3](#), [2.1](#), [4.1](#), [5.2.2](#)

## Bibliography

---

- Dmitry Brodsky and Benjamin Watson. Model Simplification Through Refinement. In *Graphics Interface*, pages 221–228, Montreal, 2000. University of British Columbia. URL <http://people.cs.ubc.ca/~dima/pubs/gi00.pdf>. 2.3
- Brent Burley and Dylan Lacewell. Ptex: Per-Face Texture Mapping for Production Rendering. *Computer Graphics Forum*, 27(4):1155–1164, June 2008. ISSN 01677055. doi: 10.1111/j.1467-8659.2008.01253.x. URL <http://doi.wiley.com/10.1111/j.1467-8659.2008.01253.x>. 4.5
- Oleksiy Busaryev, Tamal K. Dey, and Yusu Wang. Tracking a Generator by Persistence. *Computing and Combinatorics*, 2(4):539, 2010. ISSN 1793-8309. doi: 10.1142/S1793830910000875. URL <http://www.worldscinet.com/dmaa/02/0204/S1793830910000875.html>. 51
- Marcel Campen, David Bommes, and Leif P. Kobbelt. Dual Loops Meshing: Quality Quad Layouts on Manifolds. *Transactions on Graphics*, 31(4), 2012. 5.1, 5.2.1, 5.2.2
- P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno. Zeta: A Resolution Modeling System. *Graphical Models and Image Processing*, 60(5):305–329, September 1998a. ISSN 10773169. doi: 10.1006/gmip.1998.0477. URL <http://linkinghub.elsevier.com/retrieve/pii/S1077316998904774>. 2.2, 2.11, 64, 5.2.2
- P. Cignoni, C. Montani, and R. Scopigno. A Comparison of Mesh Simplification Algorithms. *Computers & Graphics*, 22(1):37–54, 1998b. URL <http://www.sciencedirect.com/science/article/pii/S0097849397000824>. 10
- James H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, October 1976. ISSN 00010782. doi: 10.1145/360349.360354. URL <http://portal.acm.org/citation.cfm?doid=360349.360354>. 1.3, 2.2
- Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simplification Envelopes. In *SIGGRAPH*, SIGGRAPH ’96, pages 119–128, New York, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237220. URL <http://doi.acm.org/10.1145/237170.237220>. 2.2, 2.8, 5.2.2
- Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-Preserving Simplification. *SIGGRAPH*, 25, 1998. doi: 10.1145/280814.280832. URL <http://dl.acm.org/citation.cfm?id=280832>. 11
- J. H. Conway and C. McA. Gordon. Knots and Links in Spatial Graphs. *Journal of Graph Theory*, 7(4):445–453, 1983. ISSN 1097-0118. doi: 10.1002/jgt.3190070410. URL <http://dx.doi.org/10.1002/jgt.3190070410>. A.7
- Robert L. Cook, Loren Carpenter, and Edwin Catmull. The Reyes Image Rendering Architecture. *Computer Graphics*, 21(4):95–102, 1987. 8
- Massimiliano Corsini, Mohamed-Chaker Larabi, Guillaume Lavoué, Oldřich Petřík, Libor Váša, and Kai Wang. Perceptual metrics for static and dynamic triangle meshes. *State of The Art Report*, 2012. URL <http://liris.cnrs.fr/publis/?id=5451>. 11

- Hrvoje Crvelin. Moore's Law limit reached, 2012. URL <https://community.emc.com/people/ble/blog/2012/02/21/moores-law-limit-reached>. 4.1
- Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes. In *Proceedings of the ninth annual symposium on Computational geometry*, SCG '93, pages 232–239, New York, 1993. ACM. ISBN 0-89791-582-8. doi: 10.1145/160985.161140. URL <http://doi.acm.org/10.1145/160985.161140>. 5.2.2
- Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12:771–784, 1995. URL <http://www.sciencedirect.com/science/article/pii/016783969500016Y>. 49
- Tamal K. Dey and Kuiyu Li. Persistence-based Handle and Tunnel Loops Computation Revisited for Speed Up. *Computers & Graphics*, 33(3):351–358, June 2009. ISSN 00978493. doi: 10.1016/j.cag.2009.03.008. URL <http://www.cse.ohio-state.edu/~tamaldey/paper/handle/holes.pdf>. 54, 3.2.4.2, 3.18, 5.2.2
- Tamal K. Dey, Kuiyu Li, and Jian Sun. On Computing Handle and Tunnel Loops. In *International Conference on Cyberworlds*, pages 357–366, Washington, 2007. IEEE Computer Society Press. doi: 10.1109/CW.2007.12. URL <http://www.cse.ohio-state.edu/~tamaldey/paper/handle/holes.pdf>. 3.2.1, 3.2.2, 54, 3.2.4.2
- Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner. Computing Geometry-aware Handle and Tunnel Loops in 3D Models. *Transactions on Graphics*, 27(3):8, 2008. doi: 10.1145/1360612.1360644. URL [http://www.geomtop.org/paper\\_ppt/hantun-sig.pdf](http://www.geomtop.org/paper_ppt/hantun-sig.pdf). 54, 3.2.4.2
- Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner. HanTun - Computing Geometry-aware Handle and Tunnel Loops in 3D Models, 2012. URL <http://www.cse.ohio-state.edu/~tamaldey/handle/hantun.html>. 3.13, 3.14, 3.17, 5.2.2
- Carolina Díaz-Goano. Analysis of surface curvature for polygonal models. Technical report, Desig Graphics Lab North Carolina State University, Raleigh, 1998. URL <http://koolhaas.csc.ncsu.edu/pages/docs/theses/carolina.final.rpt.pdf>. 2.2
- Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. *SIGGRAPH*, 1:173–182, 1995. doi: 10.1145/218380.218440. URL <http://dl.acm.org/citation.cfm?id=218440>. 2.2, 2.6, 5.2.2
- Herbert Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, Cambridge, 2001. ISBN 978-0511530067. doi: 10.1017/CBO9780511530067. URL <http://ebooks.cambridge.org/ref/id/CBO9780511530067>. 49
- Herbert Edelsbrunner. Computational Topology, 2006. 49, 53
- Herbert Edelsbrunner, David G. Kirkpatrick, and Raimund Seidel. On the Shape of a Set of Points in the plane. *Transactions on Information Theory*, 29(4):551–559, July 1983. ISSN 0018-9448. doi: 10.1109/TIT.1983.1056714. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1056714>. 44

## Bibliography

---

- Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological Persistence and Simplification. In *FOCS*, page 13, Washington, 2000. IEEE Computer Society Press. doi: 0-7695-0850-2. URL <http://www.springerlink.com/index/pdf/10.1007/s00454-002-2885-2>. 3, 3.2.3, 49
- Jihad El-Sana and Amitabh Varshney. Controlled Simplification of Genus for Polygonal Models. In *Visualization'97*, pages 403–410, Los Alamitos, 1997. IEEE Computer Society. ISBN 0-8186-8262-0. doi: 10.1109/VISUAL.1997.663909. URL <http://dl.acm.org/citation.cfm?id=266989.267109&coll=DL&dl=GUIDE&CFID=90718611&CFTOKEN=54765010>. 3.2.1
- Eric Enderton and Daniel Wexler. The Workflow Scale. Why 5x Faster Might Not Be Enough. In *Computer Graphics International*, page 3, Ottawa, 2011. Workshop on VFX, Computer Animation, and Stereo Movies. URL <http://enderton.org/eric/>. 5, 1.4
- Kayvon Fatahalian. Graphics and Imaging Architectures. Technical report, Carnegie Mellon, Pittsburgh, 2011. URL <http://www.cs.cmu.edu/afs/cs/academic/class/15869-f11/www/>. 1.4, 1.2, 5.2.2
- Sigmund Freud. *General Psychological Theory: Papers on Metapsychology*. Touchstone, New York, reprint edition, 1963. ISBN 978-0684842929. 19
- Michael Garland and Paul S. Heckbert. Surface Simplification Using Quadric Error Metrics. In *SIGGRAPH*, pages 209–216, New York, 1997. ACM. doi: 10.1145/258734.258849. 2.2, 2.9, 5.2.2
- Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Springer, Berlin, 1991. ISBN 978-0792391814. 2.3
- Craig Gotsman, Stefan Gumhold, and Leif Kobbelt. Simplification and compression of 3D meshes. In *Tutorials on Multiresolution in Geometric Modelling*, page 44, Berlin, 2002. Springer. URL [http://www.cs.sfu.ca/~haoz/teaching/cmpt464/references/compression\\_tutorial.ps](http://www.cs.sfu.ca/~haoz/teaching/cmpt464/references/compression_tutorial.ps). 2.3
- Xianfeng Gu, Seven Gortler, and Hugues Hoppe. Geometry Images. *Transactions on Graphics*, 21(3):355–361, 2002. URL <http://www.cs.princeton.edu/courses/archive/fall10/cos526/papers/gu02.pdf>. 2.2, 2.12, 5.2.2
- Igor Guskov and Zoë J. Wood. Topological Noise Removal. In *Graphics Interface*, pages 19–26, Ottawa, 2001. ACM Press. doi: 0-9688808-0-0. URL [http://digitalcommons.calpoly.edu/csse\\_fac/208/](http://digitalcommons.calpoly.edu/csse_fac/208/). 3.2.1
- Bernd Hamann. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11(2):197–214, April 1994. ISSN 0167-8396. doi: 10.1016/0167-8396(94)90032-9. URL [http://dx.doi.org/10.1016/0167-8396\(94\)90032-9](http://dx.doi.org/10.1016/0167-8396(94)90032-9). 2.2, 2.5, 5.2.2
- Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002. ISBN 978-0521795401. URL <http://books.google.com/books?hl=en&lr=&id=h-wc3TnZMCcC&oi=fnd&pg=PR16&dq=Algebraic+Topology&ots=LfTd96BX5P&sig=0acLn0IHB3QpWsTlVKkayU0dD2Q>. 0, 23, 42, A.6

- Paul S. Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. *SIGGRAPH*, 25(May), 1997. URL <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA461098>. 10
- Peter Hillman, J.P. Lewis, Sebastian Sylwan, and Erik Winquist. Issues in adapting research algorithms to stereoscopic visual effects. In *Image Processing (ICIP)*, pages 17–20, Hong Kong, 2010. IEEE Computer Society Press. ISBN 9781424479948. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5653542](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5653542). 1.4
- Paul Hinken. Geometric Optimization. In *Proceedings Visualization*, pages 189–195, Washington, 1993. IEEE Computer Society. ISBN 0-8186-3940-7. URL <http://dl.acm.org/citation.cfm?id=949882>. 2.2, 2.4, 5.2.2
- TC Ho, YC Lin, and JH Chuang. User-Assisted Mesh Simplification. In *VRCA*, pages 59–66, New York, 2006. ACM. URL <http://dl.acm.org/citation.cfm?id=1128934>. 64
- John P. Holdren, Eric Lander, and Harold Varmus. Designing a Digital Future. Technical report, Executive Office of the President, Washington, 2010. URL <http://www.whitehouse.gov/administration/eop/ostp/pcast>. 3
- Hugues Hoppe. Progressive Meshes. In *SIGGRAPH*, SIGGRAPH ’96, pages 99–108, New York, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237216. URL <http://doi.acm.org/10.1145/237170.237216>. 2.2, 2.7, 5.2.2
- Kai Hormann, Konrad Polthier, and Alla Sheffer. Mesh Parameterization: Theory and Practice. In *SIGGRAPH Asia - Course Notes*, New York, 2007. ACM. URL <http://www.inf.usi.ch/hormann/parameterization/SphericalParameterization.pdf>. 5.2.1
- Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–9, July 2007. ISSN 1367-4811. doi: 10.1093/bioinformatics/btm250. URL <http://www.ncbi.nlm.nih.gov/pubmed/17488753>. 52
- Takuya Kitagawa. The Idea of Morse Theory. Technical report, Harvard University, Physics Department, Boston, 2006. URL <http://www.people.fas.harvard.edu/~tkitagaw/others.html>. 43
- Leif Kobbelt, Swen Campagna, and Hans-Peter Seidel. A General Framework for Mesh Decimation. In *Graphics Interface*, pages 43–50, New York, 1998. ACM. URL <http://vims.cis.udel.edu/~jeab/research/papers/simplification/kobbelt98general.pdf>. 2.1, 2.3
- Shirley F. Konkle, Patrick J. Moran, Bernd Hamann, and Kenneth I. Joy. Fast Methods for Computing Isosurface Topology with Betti Numbers. In *Data Visualization: The State of the Art-Proceedings Dagstuhl Seminar on Scientific Visualization*, pages 363–375. Kluwer Academic Publishers, Norwell, 2003. 5.2.2
- Dave Kosak. John Carmack on Next-Gen Consoles and Game Development, 2005. URL <http://www.gamespy.com/articles/641/641662p1.html>. 1

## Bibliography

---

- Dmitry Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and Computation in Mathematics*. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-71961-8. doi: 10.1007/978-3-540-71962-5.  
URL <http://www.springerlink.com/index/10.1007/978-3-540-71962-5>. 49
- Edna Ernestine Kramer. *The Nature and Growth of Modern Mathematics*. Princeton Paperbacks. Princeton University Press, Princeton, 1982. ISBN 9780691023724. URL <http://books.google.de/books?id=LLEZQC74gVcC>. 22
- Er Li, XiaoPeng Zhang, WuJun Che, and WeiMing Dong. Global parameterization and quadrilateral meshing of point cloud. *ACM SIGGRAPH ASIA 2009 Posters on - SIGGRAPH ASIA '09*, pages 1–1, 2009. doi: 10.1145/1666778.1666832. URL <http://portal.acm.org/citation.cfm?doid=1666778.1666832>. 5.2.1
- Peter Lindstrom and Greg Turk. Fast and Memory Efficient Polygonal Simplification. In *VIS*, pages 279–286, Los Alamitos, 1998. IEEE Computer Society. ISBN 0-8186-9176-X. doi: 10.1109/VISUAL.1998.745314. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=745314> [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=745314](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=745314). 2.2, 2.10, 5.2.2
- Peter Lindstrom and Greg Turk. Image-driven simplification. *Transactions on Graphics*, 19(3):204–241, July 2000. ISSN 07300301. doi: 10.1145/353981.353995. URL <http://dl.acm.org/citation.cfm?id=353995>. 11
- David Luebke and Benjamin Hallen. Perceptually Driven Simplification for Interactive Rendering. In Steven Gortler and Karol Myszkowski, editors, *Rendering Techniques 2001: Proceedings of the Eurographics*, pages 223–234. Springer, Wien, 2001. ISBN 978-3211837092. URL [http://books.google.com/books?hl=en&lr=&id=EAudgTi-0mEC&oi=fnd&pg=PA223&dq=Perceptually+Driven+Simplification+for+Interactive+Rendering&ots=HU9fD9b9YF&sig=EqgqoMe9Z4wSl9vkXL09xdv7\\_8w](http://books.google.com/books?hl=en&lr=&id=EAudgTi-0mEC&oi=fnd&pg=PA223&dq=Perceptually+Driven+Simplification+for+Interactive+Rendering&ots=HU9fD9b9YF&sig=EqgqoMe9Z4wSl9vkXL09xdv7_8w). 11
- David P. Luebke. A Developer’s Survey of Polygonal Simplification Algorithms. *Computer Graphics and Applications, IEEE*, 21(3):24–35, 2001. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=920624](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=920624). 1.3
- David P. Luebke, Martin Reddy, and Jonathan D. Cohen. *Level of Detail for 3D Graphics*. Morgan Kaufmann, San Francisco, 2002. ISBN 978-1558608382. URL <http://books.google.com/books?hl=en&lr=&id=uiz6IKAVxP8C&oi=fnd&pg=PP2&dq=Level+of+Detail+for+3D+Graphics&ots=lMuKjLMwMc&sig=DpIUVAID3V-h7ICX4iE9yh0oLp0>. 2.2
- Hiroshi Maehara. Why is P2 Not Embeddable in R3? *The American Mathematical Monthly*, 100(9):862–864, 1993. URL <http://www.jstor.org/stable/10.2307/2324664>. A.7
- Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8):114–117, 1965. A.1

- John Morgan and Gang Tian. *Ricci Flow and the Poincaré Conjecture*. American Mathematical Society, Cambridge, volume 3 edition, 2007. URL <http://www.claymath.org/library/monographs/cmim03.pdf>. 25
- Martin Newell. Wire-frame model of the Utah Teapot, 1975a. URL <http://archive.computerhistory.org/resources/still-image/UtahTeapot/102672453.03.01.1g.jpg>. 1.2, 5.2.2
- Martin Newell. Martin Newell's drawing of the Utah Teapot, 1975b. URL <http://www.computerhistory.org/revolution/computer-graphics-music-and-art/15/206/556>. A.3
- Xinlai Ni, Michael Garland, and John C. Hart. Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh. *Transactions on Graphics*, 23(3):613–622, 2004. URL <http://dl.acm.org/citation.cfm?id=1015769>. 3.2.1
- Fakir S. Nooruddin and Greg Turk. Simplification and Repair of Polygonal Models Using Volumetric Techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, April 2003. ISSN 1077-2626. doi: 10.1109/TVCG.2003.1196006. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1196006>. 3.2.1
- NVidia. NVIDIA Collaborates With Weta to Accelerate Visual Effects for Avatar, 2009. URL [http://www.nvidia.com/object/wetadigital\\_avatar.html](http://www.nvidia.com/object/wetadigital_avatar.html). 8
- Valerio Pascucci and Kree Cole-McLaughlin. Parallel Computation of the Topology of Level Sets. *Algorithmica*, 38(1):249–268, November 2003. ISSN 0178-4617. doi: 10.1007/s00453-003-1052-3. URL <http://www.springerlink.com/index/10.1007/s00453-003-1052-3>. 5.2.2
- D. J. Paul. Moore's Law, 2006. URL <http://www.sp.phy.cam.ac.uk/~SiGe/Moore'sLaw.html>. A.1
- Marc Pollefeys, Luc Van Gool, and Maarten Vergauwen. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, September 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000025798.50602.3a. URL <http://www.springerlink.com/openurl.asp?id=doi:10.1023/B:VISI.0000025798.50602.3a>. 1.3
- Tibor Radó. Über den Begriff der Riemannschen Fläche. *Acta Scientiarum Mathematicarum Universitatis Szegediensis*, 2(2):101–121, 1925. URL <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Über+den+Begriff+der+Riemannschen+Fläche#0>. 3.1.2
- A. A. Ranicki, A. J. Casson, D. P. Sullivan, M. A. Armstrong, C. P. Rourke, and G. E. Cooke. *The Hauptvermutung Book*. Springer, Berlin, Heidelberg, 1996. ISBN 978-0792341741. URL <http://www.maths.ed.ac.uk/~aar/books/haupt.pdf>. 41
- Howard Rheingold. *Virtual Reality*. Summit Books, New York, hardcover edition, 1991. ISBN 978-0671693633. 1.1
- V Robins. Towards computing homology from finite approximations. *Topology Proceedings*, 24:503–532, 1999. URL <http://at.yorku.ca/b/a/a/k/28.htm>. 3.2.3

## Bibliography

---

- Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. *Geometric Modeling in Computer Graphics*, pages 455–465, 1993. URL <http://www.cs.uu.nl/docs/vakken/ddm/slides/papers/rossignac.pdf>. 14, 2.2, 2.2, 2.3, 5.2.2
- Peter Schröder. Fairness Criteria, 2003. URL <http://users.cms.caltech.edu/~cs175/>. 15
- William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, July 1992. ISSN 00978930. doi: 10.1145/142920.134010. URL <http://portal.acm.org/citation.cfm?doid=142920.134010>. 2.2, 2.1, 5.2.2
- Mike Seymour. Siggraph Day One, 2008. URL [http://www.fxguide.com/featured/Siggraph\\_Day\\_One/](http://www.fxguide.com/featured/Siggraph_Day_One/). 1.1, A.2
- Mike Seymour. The Art of Rendering, April 2012. URL <http://www.fxguide.com/featured/the-art-of-rendering/>. 1.1
- Ching-Kuang Shene. Mesh Basics & Simplification, 2005. URL <http://www.cs.mtu.edu/~shene/COURSES/cs3621/SLIDES/Simplification.pdf>. 2.1, 2.3, A.4
- Hang Si. Constrained Delaunay Tetrahedral Mesh Generation and Refinement. *Finite Elements in Analysis and Design*, 46(1-2):33–46, 2010. doi: 10.1016/j.finel.2009.06.017. 55
- Keith N. Snavely. *Scene reconstruction and visualization from internet photo collections*. PhD thesis, University of Washington, 2008. URL <http://books.google.com/books?hl=en&lr=&id=QpxgFDvfSD4C&oi=fnd&pg=PR6&dq=Scene+Reconstruction+and+Visualization+from+Internet+Photo+Collections&ots=oFUzbsByMX&sig=A2fVL4u1UaopzQmlcHs4iCGaCUU>. 1.3, 1.6, 5.2.2
- John Stillwell. *Classical Topology and Combinatorial Group Theory*. Springer, Berlin, 1993. ISBN 978-0387979700. 36
- Sebastian Sylwan. The Application of Vision Algorithms to Visual Effects Production. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, volume 6492 of *Lecture Notes in Computer Science*, pages 189–199. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-19314-9. URL [http://dx.doi.org/10.1007/978-3-642-19315-6\\_15](http://dx.doi.org/10.1007/978-3-642-19315-6_15). 1.3, 1.4
- Sam Tanenhaus. *Literature Unbound*. Ballantine Books, New York, 1986. ISBN 978-0345332974. 5
- Ann Torrence. Martin Newell’s original teapot. *ACM SIGGRAPH*, 29(1180128), 2006. doi: 10.1145/1180098.1180128. URL <http://www.siggraph.org/s2006/main.php?f=conference&p=teapot&s=20>. 1.1, 5.2.2
- Greg Turk. Re-tiling Polygonal Surfaces. *SIGGRAPH*, 26(2):55–64, July 1992. ISSN 00978930. doi: 10.1145/142920.134008. URL <http://portal.acm.org/citation.cfm?doid=142920.134008http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Re-Tiling+Polygonal+Surfaces#0>. 2.2, 2.2, 5.2.2

- T. Vilbrandt, E. Malone, H. Lipson, and A. Pasko. Universal Desktop Fabrication. In *Heterogeneous objects modelling and applications*, pages 259–284. Springer, Berlin, 2008. ISBN 978-3-540-68441-1. 12
- Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman and Hall, Boca Raton, 2nd edition, 2002. ISBN 978-1584883470. URL <http://mathworld.wolfram.com/>. 23
- Eric W. Weisstein. Topology, 2012. URL <http://mathworld.wolfram.com/Topology.html>. 3, 21
- Stefan Winkler. Visual Fidelity and Perceived Quality: Towards Comprehensive Metrics. *Proc. SPIE Human Vision and Electronic Imaging*, 4299:114–125, 2001. ISSN 0277786X. doi: 10.1111/12.429540. URL <http://link.aip.org/link/?PSI/4299/114/1&Agg=doi&http://citeseervx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.5663&rep=rep1&type=pdf>. 4.7.1
- Edward Witten. Supersymmetry and Morse Theory. *J. Differential Geometry*, 17(4):661–692, 1982. URL <http://intlpress.com/JDG/archive/1982/17-4-661.pdf>. 43
- Zoë Wood, Hugues Hoppe, Mathieu Desbrun, and Peter Schröder. Removing Excess Topology From Isosurfaces. *Transactions on Graphics*, 23(2):190–208, April 2004. ISSN 07300301. doi: 10.1145/990002.990007. URL <http://dl.acm.org/citation.cfm?id=990007>. 3.2.1
- Jianhua Wu and Leif Kobbelt. Fast Mesh Decimation by Multiple-Choice Techniques. In *Vision, Modeling, Visualization*, pages 241–248, New York, 2002. ACM. URL <http://books.google.com/books?hl=en&lr=&id=mjpBpwXTFocC&oi=fnd&pg=PA241&dq=Fast+Mesh+Decimation+by+Multiple-Choice+Techniques&ots=rn46C15kF0&sig=sr-3Zles5RmZ4MiiJ05tEei50-8>. 2.3
- Julie C. Xia and Amitabh Varshney. Dynamic View-Dependent Simplification for Polygonal Models. In *Annual IEEE Visualization*, pages 327–334, New York, 1996. ACM. ISBN 0-89791-864-9. doi: 10.1109/VISUAL.1996.568126. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=568126>. 4.4
- Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, August 2007. ISSN 0178-2789. doi: 10.1007/s00371-007-0167-y. URL <http://www.springerlink.com/index/10.1007/s00371-007-0167-y>. 63
- Afra Zomorodian. *Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes*. Phd thesis, Stanford University, 1996. URL <http://www.cs.dartmouth.edu/~afra/thesis/onesided1.pdf>. 3.1, 23
- Afra Zomorodian and Gunnar Carlsson. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005. 48
- Afra Zomorodian and Gunnar Carlsson. Localized Homology. *Computational Geometry*, 41(3):126–148, 2008. URL <http://www.sciencedirect.com/science/article/pii/S0925772108000187>. 49

---

## A. Appendix

### A.1. Moore's law

*"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year."* [Moore, 1965, p.115]

Moore's law is an economic law stating that the density of transistors must double every 18 to 24 months to enable an economic return from the investment in semiconductor foundries (see figure A.1 below for historical data). Thus it would be wrong to state that the performance of a chip doubles at this rate. However, the more widely it became

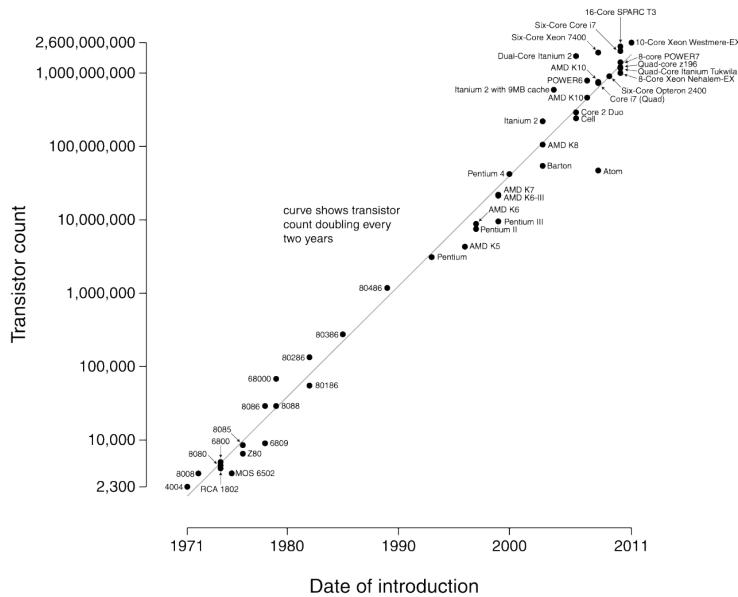


Figure A.1.: Microprocessor transistor counts 1971-2011 & Moore's Law

accepted, the more it served as a goal for an entire industry. This drove both marketing and engineering departments of semiconductor manufacturers to focus enormous energy aiming for the specified increase in processing power that it was presumed one or more of their competitors would soon actually attain. In this regard, it can be viewed as a self-fulfilling prophecy [cf. Paul, 2006; Crvelin, 2012].

## A.2. RenderMen timeline

During his keynote speech, to kick off Siggraph 2008, Ed Catmull presented the audience with the following timeline detailing the critical advances in rendering technology over the last 20 years at Pixar [cf. Seymour, 2008].

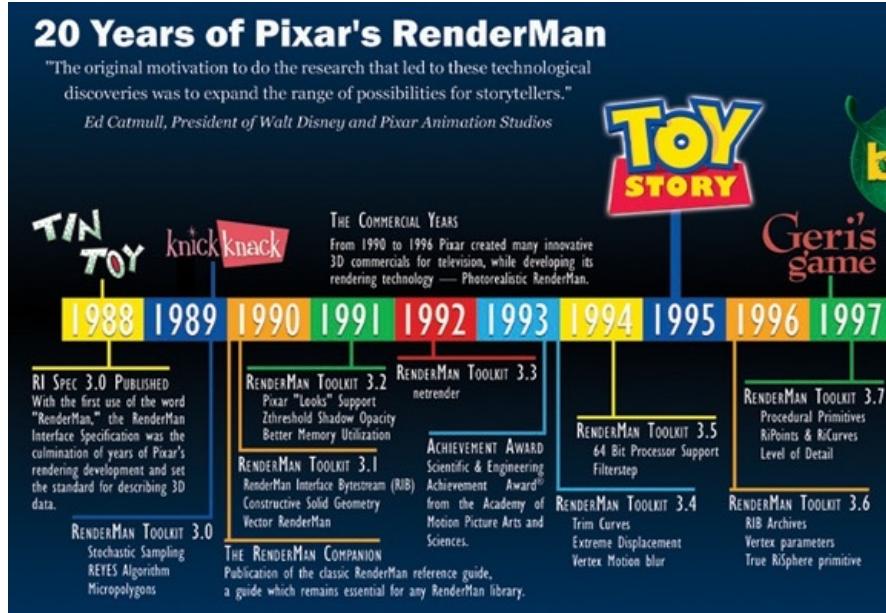


Figure A.2.: 20 Years of Pixar's RenderMan - 1

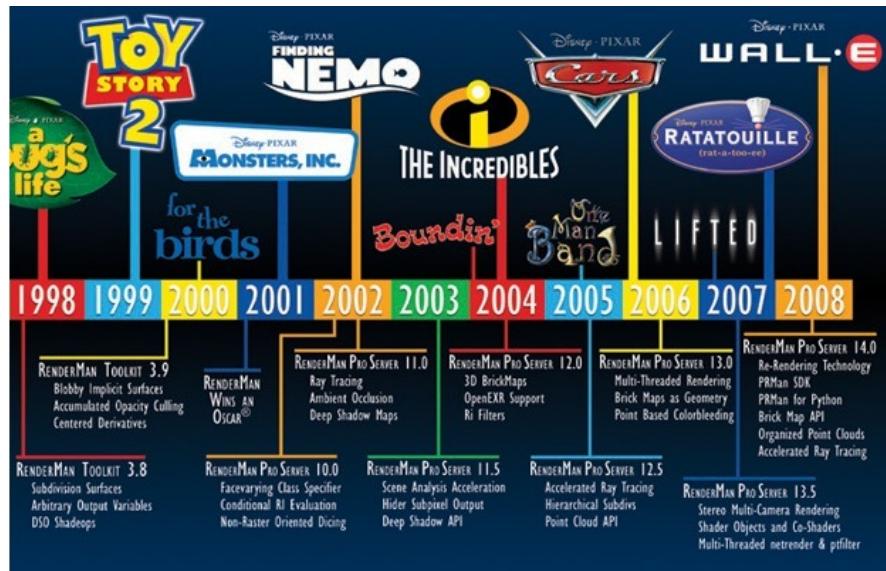


Figure A.3.: 20 Years of Pixar's RenderMan - 2

### A.3. The Utah Teapot

*Martin Newell's* PhD dissertation, was published as a DARPA research technical report in 1975. All of the images were photographic prints.

The teapot with its lid, handle, and spout comprise only 28 bicubic Bézier patches, each with 16 control points in a 4x4 grid. In the circular directions of the teapot body and lid, each patch covers 1/4 of a circle. The control points on and next to the edges of all adjacent patches are collinear, so that there are no sharp edges between the patches. The objects were modeled on paper first, see picture A.4.

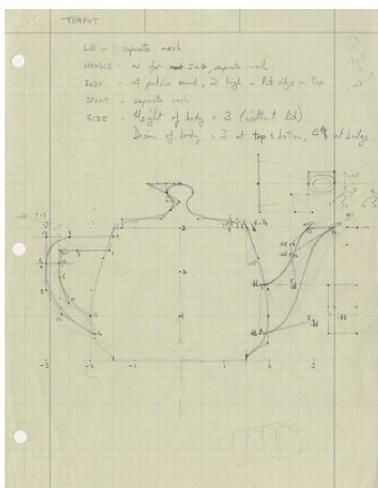


Figure A.4.: Facsimile of the original 'Utah Teapot' drawing.

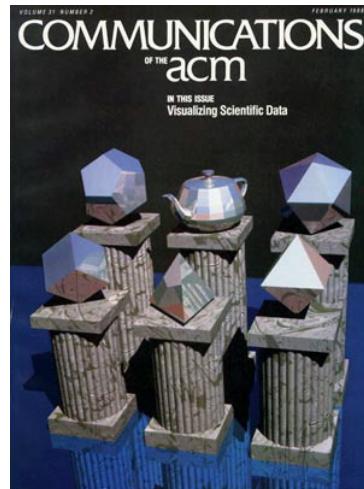


Figure A.5.: The 'Teapotahedron' as the sixth platonic solid.

One famous tribute by Jim Arvo and Dave Kirk, from their '87 SIGGRAPH paper "Fast Ray Tracing by Ray Classification" shows six stone columns five of which are surmounted by the platonic solids (tetrahedron, cube, octahedron, dodecahedron, icosahedron) – and the sixth has a teapot, see picture A.5.

It is said that *Newell* mentioned at a talk, that of all the things he has done for the world of Computer Graphics, the only thing he will be remembered for is: "That damned teapot" [cf. [Newell, 1975b](#), Copyright Computer History Museum]

#### A.4. Proof – Homeomorphism of regular quad meshes and tori

Meshes  $\mathcal{M}_{reg}$  are regular if and only if all faces  $\mathcal{F}_i$  have the same number of edges:  $|E_{\mathcal{F}}| = const.$  and all vertices  $\mathcal{V}$  have the same valence:  $val(\mathcal{V}_i) = const.$  – this also means that every regular mesh must be manifold and thus satisfy the Euler-Poincaré characteristic, with  $g$  being the genus of the mesh:

$$\chi(\mathcal{M}) = \mathcal{V} - E + \mathcal{F} = 2(1 - g) \quad (\text{A.1})$$

A regular quad mesh  $\mathcal{M}_{reg4}$  hence is a quadrilateral and each vertex is incident to exactly four edges, but since we count each edge twice, we get:  $2\mathcal{V} = E$ , accordingly each face has four edges, of which each edge is counted twice, so we have:  $2\mathcal{F} = E$

$$\mathcal{V} - E + \mathcal{F} = 2(1 - g) \quad (\text{A.2})$$

$$\frac{E}{2} - E + \frac{E}{2} = 2(1 - g) \quad (\text{A.3})$$

$$E - E = 2(1 - g) \quad (\text{A.4})$$

Thus  $\chi(\mathcal{M}) = \mathcal{V} - E + \mathcal{F} = 0$ , which means  $g = 1$ , i.e. a torus. The argument for a regular triangle mesh is the same, with  $3\mathcal{V} = E$  and  $3\mathcal{F} = 2E$  [cf. [Shene, 2005](#)].

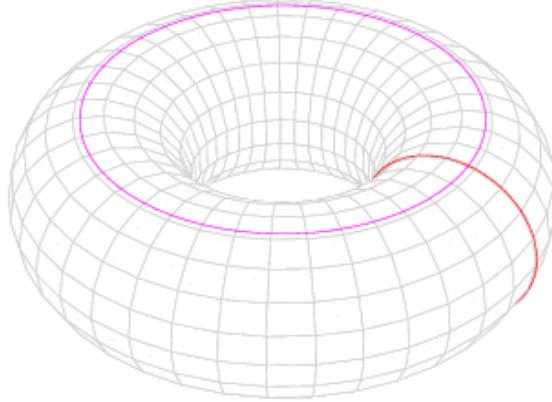


Figure A.6.: Homology cycles on a torus (image copyright wikipedia).

## A.5. Hausdorff distance

To be precise the Hausdorff distance is not a real distance, since any metric on a set must fulfill the requirement of symmetry, i.e.  $d(x, y) = d(y, x)$  whereas  $d_H(x, y) \neq d_H(y, x)$ , so instead of a distance between  $X$  and  $Y$ , the Hausdorff distance is more a distance from  $X$  to  $Y$  (sometimes also referred to as directed Hausdorff distance, but the terminology is not consistent). For example in the case depicted in picture A.7,  $d(\mathcal{S}, \mathcal{S}')$  of two open curves will remain much smaller than  $d(\mathcal{S}', \mathcal{S})$ , since here  $d(A, \mathcal{S}) \ll d(B, \mathcal{S}')$ . Thus a small one-sided distance does not necessarily imply a small overall distortion [Aspert et al., 2002, cf. pp.705-706]. This asymmetry is a property of *maximin* functions, while *minimin* functions are generally symmetric.

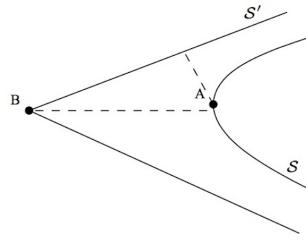


Figure A.7.: Asymmetry of the one-sided Hausdorff distance  $d(\mathcal{S}, \mathcal{S}') \ll d(\mathcal{S}', \mathcal{S})$

An important fact to reckon is that the Hausdorff distance is very sensitive to even one single outlying point  $p_d$ . For example, consider  $\mathcal{M}_A = \mathcal{M}_B \cup p_d$ , where the point  $p_d$  is at a large distance  $d$  from any point of  $\mathcal{M}_A$ . In this case  $d_H(\mathcal{M}_A, \mathcal{M}_B) = d$ , solely determined by the point  $p_d$ .

In our application scenario this is not of pivotal consideration, as every vertex of a decimated mesh has a 1-1 correspondent in the original mesh, but in general it is advisable to use a generalization of the Hausdorff distance. It is given by taking the  $k^{th}$  ranked distance rather than just the maximum:

$$d_H^k(\mathcal{M}_A, \mathcal{M}_B) = k^{th} \left\{ \inf_{p \in \mathcal{M}} \inf_{p' \in \mathcal{M}'} d(p, p') \right\} \quad (\text{A.5})$$

For a good overview of the research that has been done in the recent years on the subject, as well as best practices for the implementation, see [Bartoň et al., 2010].

## A.6. The house with two rooms

A space having the homotopy type of a point is called contractible, this is trivially true for any unit ball  $\mathcal{D}^n$  in  $\mathbb{R}^n$ . Here is an example of a 2-dimensional subspace of  $\mathbb{R}^3$ , known as “The house with two rooms” or “Bing’s house”, which is contractible but not in any obvious way.

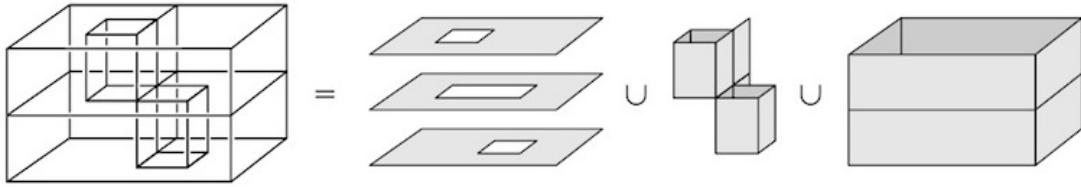


Figure A.8.: Decomposition of the retractable surface into its parts.

To build this space, we have to start with a box divided into two chambers by a horizontal rectangle. Access to the two chambers from outside the box is provided by two vertical tunnels. The upper tunnel is made by punching out a square from the top of the box and another square directly below it from the middle horizontal rectangle, then inserting four vertical rectangles, the walls of the tunnel. This tunnel allows entry to the lower chamber from outside the box. The lower tunnel is formed in similar fashion. Finally, two vertical rectangles are inserted to form ‘support walls’ for the two tunnels. The resulting space is homeomorphic to the unit ball:  $X \simeq \mathcal{D}^3 \simeq \text{point}$ . However, it is quite a challenging exercise to see exactly what this deformation retraction looks like.

To see this, imagine forming it from a ball of clay by pushing a finger into the ball to create the upper tunnel, then gradually hollowing out the lower chamber, and similarly pushing a finger in to create the lower tunnel and hollowing out the upper chamber.

The house has a visually obvious 2-simplex structure. The  $\sigma^0$  are the vertices where three or more of the depicted edges meet, and the  $\sigma^1$  are the interiors of the edges connecting these vertices and the  $\sigma^2$  are the components of the remainder of the space. If counted, one finds there are 29  $\sigma^0$ , 51  $\sigma^1$ , and 23  $\sigma^2$ , with the alternating sum  $29 - 51 + 23 = 1$ , the Euler characteristic [Hatcher, 2002, cf. pp.4-6].

## A.7. Proof – Why P2 is not embeddable

The following proof was included, simply, for its intrinsic beauty. The projective plane  $P^2$ , sometimes called a twisted sphere, is the closed surface obtained by pasting a 2-cell  $e^2$  and a Möbius band  $\mathcal{M}$  together along their boundaries.

Recall that a trivial circle bounds a 2-cell in  $\mathbb{R}^3$  that is disjoint from all other curves, otherwise it is non-trivial. Now, consider a set of six points in  $\mathbb{R}^3$ , and assume that each pair of these points is connected by a simple curve such that the curves meet only at their endpoints. Such a figure is called a complete 6-graph, denoted K6, see figure A.9:

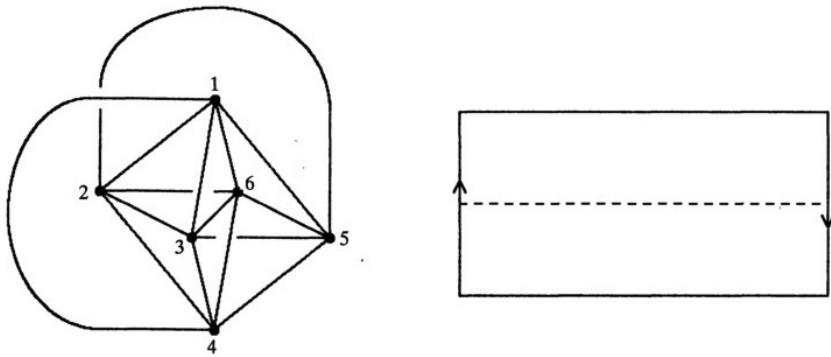


Figure A.9.: Left: A complete 6-graph; Right: Möbius band with dotted meridian.

To proof the theorem we need one corollary, the so called 'Link Appearing Theorem': *Any complete 6-graph in  $\mathbb{R}^3$  contains a pair of disjoint cycles that form a non-trivial link* [proven by: [Conway and McA. Gordon, 1983](#)]. In figure A.9 the pair of cycles [1, 3, 5] and [2, 4, 6] form such a non-trivial link<sup>70</sup>. Furthermore we define the meridian C of a Möbius band  $\mathcal{M}$ , as the line segment connecting the midpoints of the to-be-identified sides, see the dotted line in A.9. It represents a simple closed curve and for any embedding of  $\mathcal{M}$  in  $\mathbb{R}^3$ , the pair  $\{\partial\mathcal{M}, C\}$  form a non-trivial link<sup>71</sup>.

<sup>70</sup> Initially I found it hard to “see” these cycles, it helps to picture the two intertwined rings like two links of a chain.

<sup>71</sup> A fun way to practically test this lemma, is to glue a paper strip so it forms a Möbius band and then cut it along the meridian. Thus getting an even more twisted band, cutting its meridian one more time, separates the two non-trivial links, which results in two independent but chained up bands of paper:  $\beta^0 = 1 \xrightarrow{\text{double meridian cut}} \beta^0 = 2$ .

## A. Appendix

---

Consider the Möbius band represented in figure A.10. Each pair of the six points  $\{1, 2, \dots, 6\}$  is indeed, connected by a simple curve, making it a representation of  $K_6$ . It contains ten pairs of disjoint cycles, each bounding a 2-cell that is disjoint from the others. For example, the cycle  $[1, 3, 5]$  bounds the 2-cell shaded in figure A.10. Hence, as nine pairs of cycles of  $K_6$ , other than  $[1, 3, 4]$  and  $[2, 5, 6]$  are trivial links, it follows with the 'Link Appearing Theorem' that these two must be non-trivial, i.e. the meridian and the boundary.

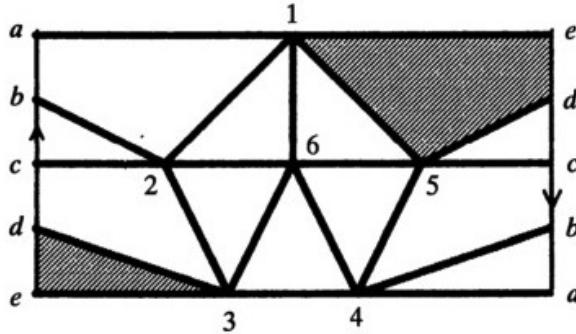


Figure A.10.: A Möbius band built from a complete 6-graph,  $K_6$ .

Now briefly suppose, that it was possible to embed  $P^2$  in  $\mathbb{R}^3$ . This would mean, that by removing an open 2-cell from the surface we get, by definition, a Möbius band represented by  $\mathcal{M}$ . Then, the boundary  $\partial\mathcal{M}$  together with the meridian  $C$  form a non-trivial link, like two links of a chain. But in consequence this would mean that  $C$  and the 2-cell, as it is bound by  $\partial\mathcal{M}$ , must intersect each other at some point. This is a contradiction, as the meridian is not intersected, as can be seen in the figure. Therefore it is only possible to immerse  $P^2$  but impossible to embed it in  $\mathbb{R}^3$ , as it will always self intersect<sup>72</sup>, see also the original publication [Maehara, 1993, pp.862-864].

---

<sup>72</sup> Basically the difference between an immersion and an embedding is that the immersion allows self-intersections. See figure 3.7 for examples of a  $P^2$  immersion.

## A.8. Glossary – Topology

During my work on this thesis, I often found myself checking back with online resources to clarify certain expression. This stems from the fact that on the one hand, topology is axiomatic in its nature and therefore relies on exact definitions rather than intuition, and on the other hand that there are deviant usages of nomenclature throughout the different fields that concern themselves with the topic. Following is a short glossary that was a handy companion during the reading sessions. I compiled it mainly from these resources: [Wolfram MathWorld](#) – which is an extensive pool of definitions, and some dedicated glossaries: [UC Davis – Topology Index](#), [Topology Atlas Glossary](#) and [Wikipedia – Glossary of topology](#):

**absolut retract** Let  $\mathcal{K}$  be a class of topological spaces that is closed under homeomorphism, and let  $\mathbb{X}_1, \mathbb{X}_2 \in \mathcal{K}$ . If  $\mathbb{X}_1 \subseteq \mathbb{X}_2$ , is a retract for every element, i.e. there is a continuous map  $f : \mathbb{X}_2 \rightarrow \mathbb{X}_1$  such that for all elements in  $\mathbb{X}_1 : f(x) = x$ , then  $\mathbb{X}_1$  is a retract for the class  $\mathcal{K}$ .

**algebraic geometry** Traditionally, the geometry of solutions in the complex numbers to polynomial equations. Modern algebraic geometry is also concerned with algebraic varieties, which are a generalization of such solution sets, as well as solutions in fields other than complex numbers, for example finite fields.

**algebraic topology** The branch of topology concerned with homology and other algebraic models of topological spaces.

**approach space** An approach space is a generalization of metric space based on point-to-set distances, instead of point-to-point.

**arc** Homeomorphic image of a closed line interval.

**automorphism** An isomorphism of a group with itself.

**basis** A basis for a topological space  $\mathbb{X}$  is a collection of open sets that contain "arbitrarily small" neighborhoods of every point of  $\mathbb{X}$ . Specifically, for every point  $x \in \mathbb{X}$ , and for every open set  $\mathbb{U}$  containing  $x$ , the collection must include a neighborhood of  $x$  lying within  $\mathbb{U}$ .

**boundary** The boundary (or frontier) of a set is the set's closure minus its interior. Equivalently, the boundary of a set is the intersection of its closure with the closure of its complement. The boundary of a set  $A$  is denoted by  $\partial A$  or  $bd A$ . The boundary  $\partial M$  of a manifold is the set of points of it that have boundary patches. The boundary of an  $n$ -manifold  $M$  is an  $(n-1)$ -manifold. A  $n$ -boundary must be a  $n$ -cycle.

**bounded** A set in a metric space is bounded if it has finite diameter. Equivalently, a set is bounded if it is contained in some open ball of finite radius. A function taking values in a metric space is bounded if its image is a bounded set.

**characteristic class** A kind of homological model for a decoration or property of a manifold or other topological space. The simplest characteristic class describes how a manifold fails to be orientable, that is, in which directions a being can travel in the manifold and reverse its handedness.

**circle packing** An arrangement of round disks in the Euclidean or hyperbolic plane or on the round sphere such that no two disks overlap with non-zero area. Depending on the context, the circles may or may not be the same size. A theorem of Koebe, revived by Thurston, states that given any planar graph, there is a circle packing with a circle for each vertex of the graph and kissing circles for each edge.

**classification** The goal in a branch of mathematics of providing an exhaustive list of some type of mathematical object with no repetitions. Example: The classification of 3-manifolds is one of the outstanding problems in topology. With the advent of computers, one weak but precise way to state a classification problem is to ask whether there is an algorithm to determine whether two given objects are equivalent.

**codimension** In general, if a mathematical object sits inside or is associated to another object of dimension  $n$ , then it is said to have codimension  $k$  if it has dimension  $n - k$ .

## A. Appendix

---

**cohomology** Cohomology is an invariant of a topological space, formally dual to homology, and so it detects holes in a space. Cohomology has more algebraic structure than homology, making it into a graded ring (with multiplication given by the so-called 'cup product'), whereas homology is just a graded Abelian group invariant of a space.

**combinatorial geometry** The visual study of discrete and finite structures, and the study of discrete and finite possibilities for the arrangement or features of geometric objects.

**compact** A topological space is compact if every collection of open sets that covers the space has a finite subset that also covers the space. The compact subspaces of  $\mathbb{R}^n$  are the closed and bounded sets.

**complex manifold** A manifold with complex coordinates; its ordinary or real dimension is then twice its complex dimension.

**completely regular** A space is completely regular if, whenever  $C$  is a closed set and  $x$  is a point not in  $C$ , then  $C$  and  $x$  are functionally separated.

**conformal** Angle-preserving or angle-defining. The Mercator map is a conformal map of the Earth because angles are true. A conformal structure on a manifold defines angles between curves segments on the manifold but not their lengths.

**connected sum** 1. A manifold formed from two others by removing balls and gluing along the resulting spherical boundary. 2. The analogous operation for knots; a band-connected sum in which the band connects the knots in the simplest possible way (by piercing the separating sphere only once)..

**contractible** A space  $X$  is contractible if the identity map on  $X$  is homotopic to a constant map, i.e. a space is contractible if it can be shrunk to a point within itself. The homotopy that does this is called a 'contraction'. Every contractible space is simply connected.

**convolution** A convolution of two planar regions is the set of all vector sums of a point in one region with a point in the other.

**curvature** In Riemannian geometry, usually means the intrinsic curvature of a manifold with a Riemannian metric. The curvature at a point is positive (negative) if the sum of the angles of a small approximate triangle at that point is greater than (less than)  $180^\circ$  degrees.

**cyclically symmetric, self-complementary plane partition** A plane partition in a cube which is symmetric under sending the  $x$  axis to the  $y$  axis,  $y$  to  $z$ , and  $z$  to  $x$ , and at the same time is equal to its complement; the set of cubes in the box not in the plane partition considered again as a plane partition by inverting all three axes. Equivalently, a lozenge tiling of a regular hexagon invariant under 60 degree rotation.

**deformation retract** A subspace  $A \subset X$  is a deformation retract if  $X$  can be shrunk down to  $A$  without moving any point of  $A$ . The homotopy that does the shrinking is called a 'deformation retraction'.

**degenerate diffusion** A diffusion process that only allows diffusion in prespecified directions instead of in every direction.

**diffeomorphism** A bijection between two manifolds that preserves all smooth structure.

**differential geometry** The general study of smooth manifolds decorated by continuous structures such as foliations, Riemannian metrics, and symplectic structures. Riemannian geometry is a disproportionate part of differential geometry.

**discrete space** A space  $X$  is discrete if every subset of  $X$  is open. We say that  $X$  carries the discrete topology.

**embedding** A mapping into a space whose image is homeomorphic to the domain. The parametrization of a submanifold by means of a standard model. A knotted sphere in 4-space is an embedding of the familiar round sphere.

**foliation** A decoration of a manifold in which the manifold is partitioned into sheets of some lower dimension, and the sheets are locally parallel. (More technically, the foliated manifold is locally homeomorphic to a vector space decorated by cosets of a subspace).

**functor** A correspondence from one category to another mapping objects to objects and preserving morphisms.

**fundamental group** A group, often non-abelian, that rather fully describes the periodicity or the 1-dimensional holes in a topological space.

**geodesic** On a Riemannian manifold or some other metric space, a curve which is the shortest path between any two points on it that are sufficiently close together.

**geometric function theory** The study of the geometry in the complex plane of complex analytic functions, in particular the relation between the image of the unit disk of an analytic function and its power series.

**Geometrization Conjecture** The conjecture of Thurston that, after cutting along essential spheres and tori, every compact 3-manifold admits a special Riemannian metric known as a geometry, usually hyperbolic geometry. The conjecture subsumes the Poincaré conjecture and many other standard conjectures about 3-manifolds, and constitutes a classification of 3-manifolds.

**Gromov norm** An invariant associated with the homology of a topological space that measures how many simplices are needed to represent a given homology class.

**group** A mathematical system consisting of elements from a set and a binary operation such that they obey: Associativity and closure for the elements, as well as having an identity element and an inverse element.

**homeomorphism** A bijection between two topological spaces that preserves all continuous structure; the basic notion of equivalence in topology. If  $\mathbb{X}_1$  and  $\mathbb{X}_2$  are spaces, a homeomorphism from  $\mathbb{X}_1$  to  $\mathbb{X}_2$  is a bijective function  $f : \mathbb{X}_1 \rightarrow \mathbb{X}_2$  such that  $f$  and  $f^{-1}$  are continuous. The spaces  $\mathbb{X}_1$  and  $\mathbb{X}_2$  are then said to be homeomorphic. From the standpoint of topology, homeomorphic spaces are identical.

A space  $\mathbb{X}$  is homogeneous if, for every  $x$  and  $y$  in  $\mathbb{X}_1$ , there is a homeomorphism  $f : \mathbb{X} \rightarrow \mathbb{X}$  such that  $f(x) = y$ . Intuitively, the space looks the same at every point. Every topological group is homogeneous.

**homological algebra** The algebraic study of the homology and cohomology of manifolds and other mathematical objects. Homological algebra is a grand generalization of linear algebra.

**homology/singular homology** Homology is an algebraic object associated to a manifold or another mathematical object which gives a measure of the number of holes in the object – manifolds form a homology when they form the boundary of a higher-dimensional manifold inside the manifold in question. The homology of a topological space has a relatively technical definition, but it is relatively easy to compute and study with tools from linear algebra.

**homotopy/homotopic maps** Two continuous maps  $f, g : \mathbb{X}_1 \rightarrow \mathbb{X}_2$  are homotopic (in  $\mathbb{X}_2$ ) if there is a continuous map  $H : \mathbb{X}_1 \times [0, 1] \rightarrow \mathbb{X}_2$  such that  $H(x, 0) = f(x)$  and  $H(x, 1) = g(x)$  for all  $x$  in  $\mathbb{X}_1$ . Here,  $\mathbb{X}_1 \times [0, 1]$  is given the product topology. The function  $H$  is called a homotopy (in  $\mathbb{X}_2$ ) between  $f$  and  $g$ .

**hyperbolic geometry** In modern terms, hyperbolic geometry is the study of manifolds with Riemannian metrics with constant negative curvature. The hyperbolic plane is a particular hyperbolic manifold which is in a sense universal among hyperbolic surfaces; similarly there is also hyperbolic n-space. Escher's circle limit prints are excellent illustrations of the hyperbolic plane.

**immersion** A locally but not globally smoothly invertible mapping of one manifold into another. The image may have self-intersections. For example, the figure-8 is an immersion of the circle in 2D.

**invariant** In topology, a number, polynomial, or other quantity associated to a topological object such as a knot or 3-manifold which depends only on the underlying object and not on its specific description or presentation. A topological invariant is a property which is preserved under homeomorphism. For example, compactness and connectedness are topological properties, whereas boundedness and completeness are not. Algebraic topology is the study of topologically invariant abstract algebra constructions on topological spaces.

**interior** The interior of a set is the largest open set contained in the original set. It is equal to the union of all open sets contained in it. An element of the interior of a set  $A$  is an interior point of  $A$ .

**isometric isomorphism** If  $\mathbb{M}_1$  and  $\mathbb{M}_2$  are metric spaces, an isometric – a mapping which preserves the metric – isomorphism from  $\mathbb{M}_1$  to  $\mathbb{M}_2$  is a bijective isometry  $f : \mathbb{M}_1 \rightarrow \mathbb{M}_2$ . The metric spaces are then said to be isometrically isomorphic. From the standpoint of metric space theory, isometrically isomorphic spaces are identical.

**Jacobian variety** A space associated to a Riemann surface defined most succinctly as the complex cohomology (as a vector space) divided by the integral cohomology (as a lattice). It is simultaneously a complex manifold, an algebraic variety, and a Lie group.

**Jones polynomial** A famous invariant of knots and links discovered by Vaughan Jones. It has several extremely elementary definitions and at the same time involves deep mathematics.

**kernel** In algebra, the set of vectors annihilated (sent to zero) by a matrix, linear operator, homomorphism, or any similar function. In analysis, a continuous analogue of a matrix. Given a vector space of functions of a parameter or functions on a manifold, an operator may have a kernel or matrix whose rows and columns are indexed by the parameter or by points on the manifold.

**knot/link** A link is a collection of disjoint circles lying in a 3-manifold, often but not always Euclidean 3-space or the 3-sphere. A knot is link which happens to be a single circle. Manifold topologists usually study tame knots and links, which can be represented by smooth or polygonal curves, but there are also wild links which are infinitely knotted. Ordinary knots and links were topologically classified (in a certain sense) by Haken and Thurston.

**lamination** A decoration of a manifold in which some subset is partitioned into sheets of some lower dimension, and the sheets are locally parallel. It may or may not be possible to fill the gaps in a lamination to make a foliation.

**lattice** A periodic arrangement of points such as the vertices of a tiling of space by cubes or the positions of atoms in a crystal. More technically, a discrete abelian subgroup of an n-dimensional vector space which is not contained in an n-1-dimensional vector space. Lattices play a central role in the theory of Lie groups, in number theory, in error-correcting codes, and many other areas of mathematics.

## A. Appendix

---

**Lie algebra** An algebraic structure on a vector space which describes multiplication of elements of a Lie group which are very close to the identity (infinitesimal transformations). Lie algebras are almost as important as their comrades-at-arms, Lie groups.

**Lie group** A group (in the sense of abstract algebra) which is at the same time a manifold. Example: The group of rotations in n dimensions is a Lie group of dimension  $n(n-1)/2$ . Lie groups are fundamental objects in mathematics and physics, especially quantum field theory.

**loop space** Given a topological space  $\mathbb{X}$ , its loop space is the topological space of all continuous functions from a circle to  $\mathbb{X}$ . Loop spaces are important examples of new topological spaces formed from old ones, as well as examples of infinite-dimensional spaces in mathematics.

**loop and sphere theorems** Two foundational theorems in 3-manifold topology, proved by Papakyriakopoulos in the 1950's. The loop and sphere theorems generalize Dehn's lemma, which asserts that if a knot in three dimensions bounds a disk which intersects itself in its interior, the disk can be simplified to remove all self-intersection and the knot must be trivial.

**loop, trivial** The trivial loop is the loop that takes every point to its basepoint. Formally, if  $\mathbb{X}$  is a topological space and  $x \in \mathbb{X}$ , the trivial loop based at  $x$  is the map  $f : [0, 1] \rightarrow \mathbb{X}$  given by  $f = x$  for all arguments – therefore (trivially) fulfilling the requirement of a loop: A map of the unit interval  $[0, 1]$  into a space with the same beginning and end points, i.e.  $f(0) = f(1)$ .

**metrizable** For a topological space, the property that there exists a metric compatible with the topology. To say that a topological space is metrizable is to treat it as a metric space, but without distinguishing any specific or preferred distance function.

**Moore space** A Moore space is a developable regular Hausdorff space.

**neighbourhood** A neighbourhood of a point  $x$  is a set containing an open set which in turn contains the point  $x$ . More generally, a neighbourhood of a set  $S$  is a set containing an open set which in turn contains the set  $S$ . A neighbourhood of a point  $x$  is thus a neighbourhood of the singleton set  $\{x\}$ . (Note that under this definition, the neighbourhood itself need not be open. Many authors require that neighbourhoods be open; be careful to note conventions.)

**non-abelian** Non-commutative or order-dependent. For example, the group of manipulations of the Rubik's cube is non-commutative because the state of the cube depends greatly on the order that moves are performed on it.

**orbifold** A topological object defined by Thurston which is locally modelled by Euclidean spaces divided by finite groups of symmetries. Orbifolds are manifolds with singularities such as reflection surfaces, where they resemble manifolds with boundary, and cone lines, where they are modelled (in the direction perpendicular to the cone line) by a cone with an angle of  $360/n$  degrees for some  $n$ .

**partition of unity** A partition of unity of a space  $\mathbb{X}$  is a set of continuous functions from  $\mathbb{X}$  to  $[0, 1]$  such that any point has a neighbourhood where all but a finite number of the functions are identically zero, and the sum of all the functions on the entire space is identically 1.

**path-connected component** A path-connected component of a space is a maximal nonempty path-connected subspace. The set of path-connected components of a space is a partition of that space, which is finer than the partition into connected components. The set of path-connected components of a space  $\mathbb{X}$  is denoted  $\pi_0(\mathbb{X})$ .

**path homotopy** A homotopy between paths that fixes their endpoints; or the relation of being path-homotopic. Two paths are path-homotopic if there is a path homotopy between them.

**PL flow** A motion on a space or a manifold, akin to a flow given by a vector field, in which every particle in a given simplex of some triangulation moves with constant velocity and in the same direction, so that the particle trajectories are polygons.

**pleated surface** A surface in Euclidean or hyperbolic space which resembles a polyhedron in the sense that it has flat faces that meet along edges. Unlike a polyhedron, a pleated surface has no corners, but it may have infinitely many edges that form a lamination.

**Poincaré conjecture** The conjecture that a closed, simply-connected 3-manifold must be homeomorphic to the 3-sphere. Many mathematicians, including Poincaré himself, have presented incomplete and incorrect proofs of the conjecture. It was finally proven by Grigori Perelman who was awarded the Field Medal for it, but declined it. It is the only solved Millennium problem.

**polyhedron** A region in Euclidean space which consists of flat facets with flat edges. More technically, a polyhedron must locally be a cone over a lower-dimensional polyhedron. It is sometimes but not always implicitly assumed that a polyhedron is a manifold, a topological sphere or ball, or a convex set.

**polytope** The n-dimensional generalization of a polygon or a polyhedron.

**projective plane** A 2-manifold obtained by gluing a Möbius strip and a disk along their circle boundaries.

**quasicompact** Some authors define "compact" to include the Hausdorff separation axiom, and they use the term quasicompact to mean what we call in this glossary simply "compact" (without the Hausdorff axiom). This convention is most commonly found in French, and branches of mathematics heavily influenced by the French.

**quotient map** If  $\mathbb{X}_1$  and  $\mathbb{X}_2$  are spaces, and if  $f$  is a surjection from  $\mathbb{X}_1$  to  $\mathbb{X}_2$ , then  $f$  is a quotient map (or identification map) if, for every subset  $U$  of  $\mathbb{X}_2$ ,  $U$  is open in  $\mathbb{X}_2$  if and only if  $f^{-1}(U)$  is open in  $\mathbb{X}_1$ . In other words,  $\mathbb{X}_2$  has the  $f$ -strong topology. Equivalently,  $f$  is a quotient map if and only if it is the transfinite composition of maps  $\mathbb{X}_1 \rightarrow \mathbb{X}_1/\mathbb{X}_3$ , where  $\mathbb{X}_3 \subset \mathbb{X}_1$  is a subset. Note that this doesn't imply that  $f$  is an open function.

**quotient space** If  $\mathbb{X}_1$  is a space,  $\mathbb{X}_2$  is a set, and  $f : \mathbb{X}_1 \rightarrow \mathbb{X}_2$  is any surjective function, then the quotient topology on  $\mathbb{X}_2$  induced by  $f$  is the finest topology for which  $f$  is continuous. The space  $\mathbb{X}_1$  is a quotient space or identification space. By definition,  $f$  is a quotient map. The most common example of this is to consider an equivalence relation on  $\mathbb{X}_1$ , with  $\mathbb{X}_2$  the set of equivalence classes and  $f$  the natural projection map. This construction is dual to the construction of the subspace topology.

**random triangulation** A triangulation of a surface or other manifold chosen by a random process. In a particularly interesting case, one considers all triangulations with a fixed, large number of triangles to be equally likely.

**representation/linear representation** A realization of a group, Lie group, or Lie algebra by matrices or linear transformations. More technically, a homomorphism from a group to a group of matrices.

**representation theory** The study of linear representations of groups, Lie groups and Lie algebras.

**resolvable** A topological space is called resolvable if it is expressible as the union of two disjoint dense subsets.

**Reuleaux polytope** A convex body in the plane or in higher dimensions which, like a Reuleaux triangle, consists of pieces of round spheres, each centered at one of the corners of the convex body.

**Riemann sphere** A topological sphere consisting of the complex plane and the point at infinity; an example of a Riemann surface.

**Riemannian geometry** The study of curvature and other properties of Riemannian metrics on manifolds. Sometimes also called differential geometry.

**round** In topology, the terms circle and sphere refer to topological objects and not geometric ones, so that the surface of an egg shape is also considered a sphere. A round sphere, then, is a sphere with constant curvature, i.e. a sphere in the normal sense of geometry.

**simplex** The  $n$ -dimensional generalization of the triangle and the tetrahedron; a polytope in  $n$  dimensions with  $n+1$  vertices.

**simplicial** Made up of simplices. For example, a simplicial polytope has simplices as faces and a simplicial complex is a collection of simplices pasted together in any reasonable vertex-to-vertex and edge-to-edge arrangement.

**smooth** Generally meaning, possessing infinitely many derivatives. For example,  $\sin(x)$  is a smooth function, while  $|x|^3$  is not. Manifolds are smooth if they can be defined or described by smooth functions, which implies the possession of continuous tangents.

**spectral sequence** Is a means of computing homology groups by taking successive approximations. Spectral sequences are a generalization of exact sequences, and since their introduction in the 1940s by Jean Leray, they have become an important research tool, particularly in homotopy theory. Roughly speaking, a spectral sequence is a system for keeping track of collections of exact sequences that have maps between them, thus describing intricate relationships among homology and cohomology groups coming from geometric situations such as fibrations and from algebraic situations involving derived functors.

**stratifiable** A topological space is stratifiable if it can be decomposed into shells which are similar to the shells that are a fixed distance from a point in a metric space.

**SU(2)** The 3-dimensional Lie group of  $2 \times 2$  unitary matrices; the most common Lie group in mathematics and physics after the circle.

**topological complexity** A lower bound on computational complexity introduced by Smale that involves topology. A task with high computational complexity requires a computer to make many decisions, sometimes arbitrary decisions, to untangle the topology of the space of possible inputs or outputs or the space of possibilities for an intermediate quantity.

**topological space** The basic object of topology; a formal model of the qualitative way in which something is connected to itself. The technical definition is that a topological space is a set  $X$  together with distinguished subsets called open sets or open neighborhoods, such that the empty set and  $X$  are open, the intersection of two open sets is open, and the union of an arbitrary collection of open sets is open. See also the definition of space.

**train track** A graph drawn on a surface such that every vertex has degree three, and such that all three edges meeting at a vertex have a common tangent, two edges on one side and one on the other. Every lamination of a surface is approximately parallel to, or carried by, a train track.

**weak topology** The weak topology on a set, with respect to a collection of functions from that set into topological spaces, is the coarsest topology on the set which makes all the functions continuous.

**weight** The weight of a space  $X$  is the smallest cardinal number  $K$  such that  $X$  has a base of cardinal  $K$ . (Note that such a cardinal number exists, because the entire topology forms a base, and because the class of cardinal numbers is well-ordered.)



---

## B. Affidavit / Eidesstattliche Erklärung

I hereby declare that I wrote this thesis with the titel:

***“Fast User-Guided Mesh-Simplification with Topology Control”***

on my own and without the use of any other than the cited sources and tools. All explanations that I copied directly or in their sense are marked as such. This text has not yet been handed in, neither in this nor in equal form, at any other official commission, nor has it been published.

Ich versichere hiermit, dass ich meine Diplomarbeit mit dem Titel:

***“Fast User-Guided Mesh-Simplification with Topology Control”***

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder dem Sinn nach auf Publikationen oder Werke anderer Autoren beruhen, sind als solche kenntlich gemacht. Die Arbeit wurde in gleicher oder ähnlicher Form bisher bei keinem anderen Prüfer als Prüfungsleistung vorgelegt und ist auch noch nicht als solche veröffentlicht worden.

Mir ist bewusst, dass Verstöße gegen die Grundsätze der Selbstständigkeit als Täuschung betrachtet und entsprechend der Prüfungsordnung und/oder der Allgemeinen Satzung für Studien- und Prüfungsangelegenheiten geahndet werden.

Berlin, den 31. Juli 2012

---

*Lukas N.P. Egger*