A Practical Guide to Polygon Mesh Repairing

Marcel Campen[1] and Marco Attene[2] and Leif Kobbelt[1]

[1]RWTH Aachen University, Germany
[2]IMATI-GE, Consiglio Nazionale delle Ricerche, Italy

Abstract

Digital 3D models are key components in many industrial and scientific sectors. In numerous domains polygon meshes have become a de facto standard for model representation. In practice meshes often have a number of defects and flaws that make them incompatible with quality requirements of specific applications. Hence, repairing such defects in order to achieve compatibility is a highly important task – in academic as well as industrial applications. In this tutorial we first systematically analyze typical application contexts together with their requirements and issues, as well as the various types of defects that typically play a role. Subsequently, we consider existing techniques to process, repair, and improve the structure, geometry, and topology of imperfect meshes, aiming at making them appropriate to case-by-case requirements. We present seminal works and key algorithms, discuss extensions and improvements, and analyze the respective advantages and disadvantages depending on the application context. Furthermore, we outline directions where further research is particularly important or promising.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—

1. Introduction

Nowadays, digital 3D models are key components in many industrial and scientific sectors, such as product design and manufacturing, gaming, simulation, cultural heritage, archaeology, medicine and bioinformatics. Due to their flexibility, expressiveness and hardware support, polygon meshes have become a de facto standard for model representation in many of these domains. Each application, however, has its own quality requirements that restrict the class of acceptable and supported models. In practice *real* meshes often have a number of defects and flaws that make them incompatible with such requirements. Hence, repairing these defects in order to achieve compatibility is a highly important task – a task whose complexity and level of difficulty is not uncommonly underestimated by non-experts in the field.

This importance is in place for both, academic and industrial applications: researchers in all areas of Computer Graphics want (and not rarely have) to assume a certain level of quality and integrity of the meshes they work with (to avoid unnecessarily complex algorithms or to make concepts work out), whereas practitioners have to reliably deal with real-world meshes in demanding industrial workflows which similarly rely on certain assumptions.

Thus, this tutorial has a twofold objective: first, we show how to exploit state-of-the-art techniques to solve the mesh repair problem in various scenarios; second, we describe the existing repairing methodologies and outline the directions where further research is particularly important. We systematically analyze the application contexts that deal with polygon meshes together with the requirements they pose and the problems they provoke, as well as the various types of defects that typically play a role and may make a mesh unsuitable. Subsequently, we consider existing techniques to process, repair, and improve the structure, geometry, and topology of an imperfect mesh to make it appropriate to case-by-case requirements. We describe seminal works and key algorithms, discuss extensions and improvements, and analyze the respective advantages/disadvantages while taking various key application contexts into account. Where available, we refer to existing implementations.

The tutorial is based on a recent extensive survey by the presenters [ACK], which is about to appear in ACM Computing Surveys. An accompanying website featuring freely obtainable implementations of several of the presented methods is available at `www.meshrepair.org`. There we also provide further material and updates.

2. Outline

The Application Perspective

The tutorial provides a useful and handy overview of mesh repair techniques from a practical application perspective, by considering the 3D model lifecycle from production to exploitation. Thus, we first discuss upstream applications (that *create* a mesh) based on the typical characteristics/defects of the meshes they produce, and then provide a classification of downstream applications (that *use* the model) based on the requirements they typically impose on their input meshes. By looking at the combinatorics of upstream application, repair method, and downstream application based on these criteria, we derive practical guidelines to decide which repair approaches are well suited for the data-link between any particular upstream-downstream pair – bridging the corresponding compatibility gap.

Overview and Problem Definition

We can define a mesh repairing algorithm to be a process that takes as input a surface mesh M and produces a modified version M' where some specific defects or flaws are removed or alleviated. This loose definition intentionally does not exclude methods that, while fixing specific defects, may newly introduce other flaws that again need to be fixed by subsequently applied methods – as it is often the case with available algorithms.

In general, it can be useful to investigate the context as follows:

1. What is the upstream application?
   → Determines characteristics of M
2. What is the downstream application?
   → Determines requirements on M'
3. Based on this information:
   → Is it necessary to repair M?
4. If repairing is necessary:
   → Is there an algorithm that does it directly?
5. If direct repair is not possible:
   → Can several algorithms be used in sequence?
6. If not:
   →There is room for further research.

When defining the goal of mesh repair, the problem's inherent ill-posedness must be taken into accout. Imperfect meshes with defects quite often represent an object ambiguously or incompletely and, without additional information (e.g. context, semantics), it can be impossible to decide how a certain defect is to be repaired in the right way. Depending on the types of defects, it can even be impossible to decide whether a mesh actually *contains* defects or flaws which need to be repaired. Hence, we also take a closer look at algorithms that accept additional information as input or allow for user-interaction in order to deal with this general problem.

Defect Categories

Most file formats that are used to represent polygon meshes are not guaranteed to represent only defect-free models, as they may easily encode non-manifold and/or non-orientable sets of polygons, isolated elements, intersections and a number of other defects that often are the source of problems in several contexts. We provide a categorization of all the issues that may need treatment – specifically, we distinguish among issues about local connectivity, global topology, and geometry. The following is a list of individual types of defects and flaws treated in the tutorial: isolated/dangling elements, singular edges/vertices, holes, gaps/overlaps, intersections, degeneracies, noise, aliasing, topological noise, inconsistent orientation.

Upstream Applications

Common mesh sources (i.e. upstream applications) can be characterized based on the *nature* of the data modeled (i.e. (physical) real-world data vs. (virtual) concepts) and on the *approach* employed to convert such data into polygon meshes (e.g. patch tessellation, raster data contouring, point cloud reconstruction). Both, nature and conversion approach, can be the source of defects in a mesh. In essence, to identify all the potential defects of a mesh based on the upstream application that produced it, it is often sufficient to identify the nature as well as the approach employed. In the tutorial we determine the specific properties of both aspects.

Downstream Applications

We provide an overview of the prototypical requirements of key application contexts. For instance, for the purpose of mere *visualization*, only the existence of significant holes is generally deemed unacceptable – all other types of defects can often be neglected. Other applications, e.g. *modeling*, demand at least topological manifoldness, for instance in order to be able to apply discrete differential operators. Even stricter requirements are to be fulfilled for, e.g., *rapid prototyping* purposes: the mesh model naturally needs to be convertible to a solid model, i.e. it has to well-define an interior and exterior volume. For this purpose the mesh definitely has to be closed and free of intersections and singular non-manifold configurations that would prevent an unambiguous volume classification.

Repair Algorithms

On the highest level we distinguish between methods that use a local approach (modifying the mesh only in the vicinity of the individual defects and flaws) and methods that employ a global strategy (typically based on remeshing of the input, which allows to more easily achieve robustness and global correctness guarantees).

Since we are interested in identifying repair algorithms suitable for specific contexts, we do not only explain the individual algorithmic approaches, but also, for each discussed method, consider the requirements the repair method itself poses on its input mesh, guarantees of success, accuracy of the results, possible defects newly introduced, as well as required or allowed user interaction.

For each category of defects and flaws we explain major results, seminal works, and key algorithms in detail and further discuss valuable extensions and improvements that have been proposed. We provide pointers to available implementations and tools that can readily be employed to fix mesh defects.

Outlook

One insight that can be gained is that some repair tasks are *significantly* more challenging than others. While some problems can be easily formalized and unambiguously solved, non-trivial interpretations are necessary to provide robust and intelligent algorithms for, e.g., hole filling, gap closing, and intersection removal. We discuss the gaps in the available range of repairing methods and show up possible avenues for future research that could provide further valuable contributions in the field. Promising research directions include hybrid methods which are minimally invasive and still provide global guarantees, the high-level incorporation of meta-knowledge, and the vertical integration of multiple repair techniques to pratical workflows.

3. Target Audience

The tutorial is targeted at both, researchers and practitioners with a Computer Science or Geometric Modeling background. Instead of listing the existing algorithms based on their methodology, the tutorial presents the mesh repair problem from an application perspective that is naturally helpful for both developers of 3D applications and researchers that make use of meshes in their activity. In particular, researchers from the wide field of Computer Graphics constitute one of the main targets of this tutorial, since they quite often work with polygon meshes and (often implicitly) make assumptions about their integrity. Furthermore, after having discussed what can be done today (and how it can be done), we provide an analysis of gaps in the state-of-the-art and we show fruitful avenues for future research. Thus, also researchers in the more specific field of Geometry Processing can take advantage of this tutorial.

4. References

The following is a list of all the works covered in the tutorial:

[ABA02] ANDÚJAR C., BRUNET P., AYALA D.: Topology-reducing surface simplification using a discrete solid representation. *ACM Transactions on Graphics 21*, 2 (2002), 88–105.

[ACK] ATTENE M., CAMPEN M., KOBBELT L.: Polygon mesh repairing – an application perspective. *ACM Computing Surveys to appear*.

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 587–594.

[AF06] ATTENE M., FALCIDIENO B.: ReMESH: An interactive environment to edit and repair triangle meshes. In *Shape Modeling and Applications* (2006), pp. 271–276.

[AFRS05] ATTENE M., FALCIDIENO B., ROSSIGNAC J., SPAGNUOLO M.: Sharpen&Bend: Recovering curved sharp edges in triangle meshes produced by feature-insensitive sampling. *IEEE Trans. Vis. and Comp. Graph. 11*, 2 (2005), 181–192.

[AGFF09] ATTENE M., GIORGI D., FERRI M., FALCIDIENO B.: On converting sets of tetrahedra to combinatorial and pl manifolds. *Computer-Aided Geometric Design 26*, 8 (2009), 850–864.

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. *ACM Transactions on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 408–416.

[Att10] ATTENE M.: A lightweight approach to repair polygon meshes. *The Visual Computer* (2010), 1393–1406.

[AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*, DeFloriani L., Spagnuolo M., (Eds.). Springer, 2008, ch. 2, pp. 53–82.

[BDK98] BAREQUET G., DUNCAN C., KUMAR S.: RSVP: A geometric toolkit for controlled repair of solid models. *IEEE Transactions on Visualization and Computer Graphics 4*, 2 (1998), 162–177.

[BF05] BRECKON T. P., FISHER R. B.: Non-parametric 3D surface completion. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 573–580.

[BF09] BERNSTEIN G., FUSSELL D.: Fast, exact, linear booleans. *Computer Graphics Forum 28*, 5 (2009), 1269–1278.

[BK97] BAREQUET G., KUMAR S.: Repairing CAD models. In *VIS '97: Proceedings of the 8th conference on Visualization '97* (Los Alamitos, CA, USA, 1997), IEEE Computer Society Press, pp. 363–370.

[BK01] BOTSCH M., KOBBELT L.: A robust procedure to eliminate degenerate faces from triangle meshes. In *Vision, Modeling and Visualization* (2001), pp. 283–290.

[BK05] BISCHOFF S., KOBBELT L.: Structure preserving CAD model repair. *Computer Graphics Forum 24*, 3 (2005), 527–ï£¡536.

[BMVS04] BLANZ V., MEHL A., VETTER T., SEIDEL H.-P.: A statistical method for robust 3D surface reconstruction from sparse data. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004)* (2004), pp. 293–300.

[BNK02] BORODIN P., NOVOTNI M., KLEIN R.: Progressive gap closing for mesh repairing. In *Advances in Modelling, Animation and Rendering* (2002), pp. 201–213.

[BPB06] BRANCH J., PRIETO F., BOULANGER P.: Automatic hole-filling of triangular meshes using local radial basis function. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 727–734.

[BPK05] BISCHOFF S., PAVIC D., KOBBELT L.: Automatic restoration of polygon models. *ACM Transactions on Graphics 24*, 4 (2005), 1332–1352.

[BR02] BERNARDINI F., RUSHMEIER H.: The 3D model acquisition pipeline. *Computer Graphics Forum 21*, 2 (2002), 149–172.

[BS95] BAREQUET G., SHARIR M.: Filling gaps in the boundary of a polyhedron. *Computer-Aided Geometric Design 12*, 2 (1995), 207–229.

[BSK05] BENDELS G. H., SCHNABEL R., KLEIN R.: Detail-preserving surface inpainting. In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2005), Eurographics Association, pp. 41–48.

[BTD08] BAC A., TRAN N.-V., DANIEL M.: A multistep approach to restoration of locally undersampled meshes. In *Advances in Geometric Modeling and Processing* (2008), pp. 272–289.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM, pp. 187–194.

[BW92] BØHN J. H., WOZNY M. J.: A topology-based approach for shell-closure. In *Selected and Expanded Papers from the IFIP TC5/WG5.2 Working Conference on Geometric Modeling for Product Realization* (Amsterdam, The Netherlands, 1992), North-Holland Publishing Co., pp. 297–319.

[BWS*10] BRUNTON A., WUHRER S., SHU C., BOSE P., DEMAINE E.: Filling holes in triangular meshes using digital images by curve unfolding. *International Journal of Shape Modeling 16*, 1–2 (2010), 151–171.

[CC08] CHEN C.-Y., CHENG K.-Y.: A sharpness dependent filter for recovering sharp features in repaired 3D mesh models. *IEEE Trans. Vis. and Comp. Graph. 14*, 1 (2008), 200–212.

[CCR08] CIGNONI P., CORSINI M., RANZUGLIA G.: Meshlab: an open-source 3D mesh processing system. *ERCIM News 73* (2008), 45–46.

[CK10] CAMPEN M., KOBBELT L.: Exact and robust (self-)intersections for polygonal meshes. *Computer Graphics Forum 29*, 2 (2010), 397–406.

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM, pp. 303–312.

[CS00] CURLESS B., SEITZ S.: 3D Photography. *Course Notes for SIGGRAPH 2000* (2000).

[DMGL02] DAVIS J., MARSCHNER S. R., GARR M., LEVOY M.: Filling holes in complex surfaces using volumetric diffusion. In *1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002)* (2002), pp. 428–438.

[DMP03] DE FLORIANI L., MORANDO F., PUPPO E.: Representation of non-manifold objects through decomposition into nearly manifold parts. In *ACM Solid Modeling* (2003), pp. 304–309.

[ESV97] EL-SANA J., VARSHNEY A.: Controlled simplification of genus for polygonal models. In *IEEE Visualization 97 proceedings* (1997), pp. 403–410.

[Far99] FAROUKI R.: Closing the gap between CAD model and downstream application. *SIAM news 32*, 5 (1999), 303–319.

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 950–953.

[FIMK07] FURUKAWA R., ITANO T., MORISAKA A., KAWASAKI H.: Improved space carving method for merging and interpolating multiple range images using information of light sources of active stereo. In *ACCV 2007, 8th Asian Conference on Computer Vision* (2007), pp. 206–216.

[FLD01] FISCHL B., LIU A., DALE A. M.: Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging 20*, 1 (2001), 70–80.

[FYP10] FAN H., YU Y., PENG Q.: Robust feature-preserving mesh denoising based on consistent subneighborhoods. *Visualization and Computer Graphics, IEEE Transactions on 16*, 2 (2010), 312–324.

[GHH*03] GRANADOS M., HACHENBERGER P., HERT S., KETTNER L., MEHLHORN K., SEEL M.: Boolean operations on 3D selective Nef complexes: Data structure, algorithms, and implementation. *Algorithms-ESA 2003* (2003), 654–666.

[GLWZ06] GUO T.-Q., LI J.-J., WENG J.-G., ZHUANG Y.-T.: Filling holes in complex surfaces using oriented voxel diffusion. In *Proc. 2006 International Conference on Machine Learning and Cybernetics* (2006), pp. 4370–4375.

[GTLH01] GUÉZIEC A., TAUBIN G., LAZARUS F., HORN B.: Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (2001), 136–151.

[GW01] GUSKOV I., WOOD Z.: Topological noise removal. In *Proceedings of Graphics Interface* (2001), pp. 19–26.

[HHVW96] HE T., HONG L., VARSHNEY A., WANG S. W.: Controlled topology simplification. *IEEE Transactions on Visualization and Computer Graphics 2* (1996), 171–184.

[HK06] HORNUNG A., KOBBELT L.: Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Proc. Eurographics Symposium on Geometry Processing* (2006), pp. 41–50.

[HP04] HILDEBRANDT K., POLTHIER K.: Anisotropic filtering of non-linear surface features. *Computer Graphics Forum 23* (2004), 391–400.

[HRA*11] HÉTROY F., REY S., ANDUJAR C., BRUNET P., VINACUA A.: Mesh repair with user-friendly topology control. *Computer-Aided Design 43* (01 2011), 101–113.

[HXBNP02] HAN X., XU C., BRAGA-NETO U., PRINCE J. L.: Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Transactions on Medical Imaging 21*, 2 (2002), 109–121.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3D freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM New York, NY, USA.

[JDD03] JONES T. R., DURAND F., DESBRUN M.: Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics 22* (July 2003), 943–949.

[JT04] JIA J., TANG C.-K.: Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 6 (2004), 771–786.

[Ju04] JU T.: Robust repair of polygonal models. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 888–895.

[Ju09] JU T.: Fixing geometric errors on polygonal models: A survey. *Computer Science and Technology 24*, 1 (2009), 19–29.

[JZH07] JU T., ZHOU Q.-Y., HU S.-M.: Editing the topology of 3D models by sketching. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (2007), 42–1–42–9.

[KB03] KOBBELT L., BOTSCH M.: Feature sensitive mesh processing. In *SCCG ï£¡03: Proceedings of the 19th Spring Conference on Computer Graphics* (2003), pp. 17–22.

[KHYS02] KÄHLER K., HABER J., YAMAUCHI H., SEIDEL H.-P.: Head Shop: generating animated head models with anatomical structure. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2002), ACM, pp. 55–63.

[KS05] KRAEVOY V., SHEFFER A.: Template-based mesh completion. In *Eurographics Symposium on Geometry Processing* (2005), pp. 13–22.

[KSI*07] KUMAR A., SHIH A. M., ITO Y., ROSS D. H., SONI B. K.: A hole-filling algorithm using non-uniform rational b-splines. In *Proc. 16th International Meshing Roundtable* (2007), pp. 169–182.

[L03] LÉVY B.: Dual domain extrapolation. *ACM Transactions on Graphics 22*, 3 (2003), 364–369.

[Lie03] LIEPA P.: Filling holes in meshes. In *Proc. Eurographics Symposium on Geometry Processing* (2003), pp. 200–205.

[Lue01] LUEBKE D.: A developer's survey of polygonal simplification algorithms. *Computer Graphics and Applications 21*, 3 (2001), 24–35.

[M88] MÄNTYLÄ M.: *Introduction to Solid Modeling*. WH Freeman & Co. New York, NY, USA, 1988.

[Mas04] MASUDA T.: Filling the signed distance field by fitting local quadrics. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 1003–1010.

[MD93] MÄKELÄ I., DOLENC A.: Some efficient procedures for correcting triangulated models. In *Proc. Symp. Solid Freeform Fabrication* (1993), pp. 126–134.

[MF96] MORVAN S. M., FADEL G. M.: IVECS, interactively correcting STL files in a virtual environment. In *Solid Freeform Fabrication Symposium* (1996), pp. 491–498.

[MF97] MURALI T., FUNKHOUSER T.: Consistent solid and boundary representations from arbitrary polygonal data. In *Proceedings of Symposium on Interactive 3D Graphics* (1997), pp. 155–162.

[NHR99] NIELSON G., HOLLIDAY D., ROXBOROUGH T.: Cracking the cracking problem with Coons patches. In *IEEE Visualization '99 proceedings* (1999), pp. 285–290.

[NT03] NOORUDDIN F., TURK G.: Simplification and repair of polygonal models using volumetric techniques. *ACM Transactions on Visualization and Computer Graphics 9*, 2 (2003), 191–205.

[NYC05] NGUYEN M. X., YUAN X., CHEN B.: Geometry completion and detail generation by texture synthesis. *The Visual Computer 21*, 8-10 (2005), 669–678.

[OSD97] OOMES S., SNOEREN P., DIJKSTRA T.: 3D shape representation: Transforming polygons into voxels. In *Proc. 1st Intnl. Conf. on Scale-Space Theory in Computer Vision* (1997), pp. 349–352.

[PGSQ06] PARK S., GUO X., SHIN H., QIN H.: Surface completion for shape and appearance. *Vis. Comput. 22*, 3 (2006), 168–180.

[PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L. J.: Example-based 3D scan completion. In *Eurographics Symposium on Geometry Processing* (2005), pp. 23–32.

[PMR05] PATEL P. S., MARCUM D. L., REMOTIGUE M. G.: Stitching and filling: Creating conformal faceted geometry. In *Procs of 14th International Meshing Roundtable* (2005), pp. 239–256.

[PMV06] PERNOT J. P., MORARU G., VERON P.: Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Comput. Graph. 30*, 6 (2006), 892–ï£¡902.

[PR05] PODOLAK J., RUSINKIEWICZ S.: Atomic volumes for mesh completion. In *Eurographics Symposium on Geometry Processing* (2005), pp. 33–42.

[PS96] PFEIFLE R., SEIDEL H.-P.: Triangular b-splines for blending and filling of polygonal holes. In *GI '96: Proceedings of the conference on Graphics interface '96* (Toronto, Ontario, Canada, 1996), Canadian Information Processing Society, pp. 186–193.

[RC99] ROSSIGNAC J., CARDOZE D.: Matchmaker: manifold BReps for non-manifold r-sets. In *Procs of 5th ACM symposium on Solid modeling and applications* (1999), pp. 31–41.

[RW92] ROCK S., WOZNY M. J.: Generating topological information from a bucket of facets. In *Solid freeform fabrication symposium proceedings* (1992), pp. 251–259.

[RW97] ROTH G., WIBOWOO E.: An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In *Proceedings of the conference on Graphics interface '97* (Toronto, Ont., Canada, Canada, 1997), Canadian Information Processing Society, pp. 173–180.

[SACO04] SHARF A., ALEXA M., COHEN-OR D.: Context-based surface completion. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 878–887.

[SCD*06] SEITZ S. M., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), 519–526.

[She02] SHEWCHUK J. R.: Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications 22*, 1-3 (2002), 21–74.

[SI03] SAGAWA R., IKEUCHI K.: Taking consensus of signed distance field for complementing unobservable surface. *International Conference on 3D Digital Imaging and Modeling* (2003), 410–417.

[SI08] SAGAWA R., IKEUCHI K.: Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 4 (2008), 686–699.

[SK02] SAVCHENKO V., KOJEKINE N.: An approach to blend surfaces. In *Advances in Modeling, Animation and Rendering* (2002), pp. 139–150.

[SL01] SHATTUCK D. W., LEHAY R. M.: Automated graph based analysis and correction of cortical volume topology. *IEEE Transactions on Medical Imaging 20*, 11 (2001), 1167–1177.

[SLS*07] SHARF A., LEWINER T., SHKLARSKI G., TOLEDO S., COHEN-OR D.: Interactive topology-aware surface reconstruction. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (2007), 43–1–43–9.

[SM95] SHENG X., MEIER I. R.: Generating topological structures for surface models. *IEEE Comput. Graph. Appl. 15*, 6 (1995), 35–41.

[SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 896–904.

[SRML07] SUN X., ROSIN P., MARTIN R., LANGBEIN F.: Fast and effective feature-preserving mesh denoising. *IEEE transactions on visualization and computer graphics 13*, 5 (2007), 925–938.

[SV03] SZYMCZAK A., VANDERHYDE J.: Extraction of topologically simple isosurfaces from volume datasets. In *IEEE Visualization* (2003), pp. 67–74.

[SWT06] SPILLMANN J., WAGNER M., TESCHNER M.: Robust tetrahedral meshing of triangle soups. In *Proc. Vision, Modeling, Visualization (VMV)* (2006), pp. 9–16.

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM, pp. 351–358.

[TC04] TEKUMALLA L. S., COHEN E.: *A hole-filling algorithm for triangular meshes*. Tech. rep., School of Computing, University of Utah, 2004.

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM, pp. 311–318.

[TO02] TURK G., O'BRIEN J.: Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics 21*, 4 (2002), 855–873.

[VCBS03] VERDERA J., CASELLES V., BERTALMIO M., SAPIRO G.: Inpainting surface holes. In *Int. Conference on Image Processing* (2003), pp. 903–906.

[VMM99] VOLLMER J., MENCL R., MUELLER H.: Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum 18*, 3 (1999), 131–138.

[VPK05] VARNUSKA M., PARUS J., KOLINGEROVA I.: Simple holes triangulation in surface reconstruction. In *Proc. Algorithmy* (2005), pp. 280–289.

[Wan06] WANG C.: Incremental reconstruction of sharp edges on mesh surfaces. *Computer-Aided Design 38*, 6 (2006), 689–702.

[WHDS04] WOOD Z., HOPPE H., DESBRUN M., SCHROEDER P.: Removing excess topology from isosurfaces. *ACM Transactions on Graphics 23*, 2 (2004), 190–208.

[WLG03] WAGNER M., LABSIK U., GREINER G.: Repairing non-manifold triangle meshes using simulated annealing. In *Proc. of The 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics* (2003), pp. 88–93.

[WO07] WANG J., OLIVEIRA M. M.: Filling holes on locally smooth surfaces reconstructed from point clouds. *Image Vision Comput. 25*, 1 (2007), 103–113.

[WWP10] WEI M., WU J., PANG M.: An integrated approach to filling holes in meshes. In *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence* (Washington, DC, USA, 2010), IEEE Computer Society, pp. 306–310.

[XGR*06] XU S., GEORGHIADES A., RUSHMEIER H., DORSEY J., MCMILLAN L.: Image guided geometry inference. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 310–317.

[XPR*02] XU C., PHAM D., RETTMANN M., YU D., PRINCE J.: Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging 18*, 6 (2002), 467–480.

[XZM*07] XIAO C., ZHENG W., MIAO Y., ZHAO Y., PENG Q.: A unified method for appearance and geometry completion of point set surfaces. *Vis. Comput. 23*, 6 (2007), 433–443.

[ZGL07] ZHAO W., GAO S., LIN H.: A robust hole-filling algorithm for triangular mesh. *The Visual Computer 23*, 12 (2007), 897–ï£¡997.

[ZJH07] ZHOU Q., JU T., HU S.: Topology repair of solid models using skeletons. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 675–685.

[ZRP02] ZHANG Y., ROHLING R., PAI D.: Direct surface extraction from 3D freehand ultrasound images. In *IEEE Visualization* (2002), pp. 45–52.

[ZTCS99] ZHANG R., TSAI P., CRYER J., SHAH M.: Shape-from-Shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 8 (1999), 690–706.

Eurographics 2012 Tutorial

A Practical Guide to
POLYGON MESH REPAIRING

Marcel Campen
Leif Kobbelt

RWTH Aachen University,
Germany

Marco Attene

IMATI-GE, CNR,
Italy

Eurographics 2012 Tutorial

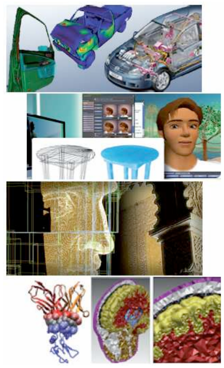
A Practical Guide to
POLYGON MESH REPAIRING

The most up-to-date slides are available at:

> www.meshrepair.org

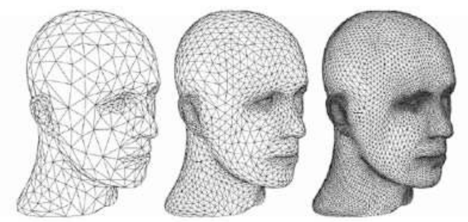
OVERVIEW

- Digital 3D models in widespread and ubiquitous use.
  - Key components in numerous industrial and scientific sectors:
    - Product design and manufacturing
    - Simulation
    - Gaming
    - Cultural heritage and archaeology
    - Medicine
    - Bioinformatics and pharmaceutical sciences



OVERVIEW

- Depending on the application, 3D models need to be:
  - visualized,
  - analyzed,
  - processed,
  - converted.



- Advanced algorithms in these contexts often have strict requirements on model quality and integrity.

OVERVIEW

- Polygon meshes are a de facto standard in numerous domains.
  - Extremely flexible and expressive,
  - Supported by hardware acceleration.

- Their versatility, at the same time, allows for a variety of representational defects and flaws.

OVERVIEW

- *Real world* meshes often contain various defects, depending on their origin.
- But many applications assume *ideal* meshes free from defects or flaws.

- *Mesh Repairing* adapts raw mesh models to specific application requirements.

OVERVIEW

- Complexity of the repair task is often underestimated by non-experts.
  - A large difference between „looks good" and „is good"
  - Most repair algorithms focus on certain defect types and ignore or even introduce others.
  - „The devil is in the details!"

THE APPLICATION PERSPECTIVE

- *The* optimal mesh repair method does not exist:
  - Each has advantages and disadvantages.
  - Some defects are repaired, others introduced.
  - The input needs to meet certain requirements.
  - Only certain guarantees about the output are provided.
- Hence, the application context needs to be considered to make the best trade-off.

THE APPLICATION PERSPECTIVE

- Categorization of:
  - Defect types
  - Upstream applications
    - based on typical characteristics/defects of produced meshes.
  - Downstream applications
    - based on typical requirements on consumed meshes.
  - Repair approaches
    - along with specific requirements and guarantees

THE APPLICATION PERSPECTIVE

- Based on these criteria, by looking at the combinatorics of
  - upstream application,
  - repair algorithm,
  - downstream application,

  guidelines can be derived to find methods well-suited for a specific data-link problem at hand.

TUTORIAL OUTLINE

- Defect types
- Upstream applications
- Downstream applications
- Repair approaches
  - local <-> global
  - mesh-based <-> volumetric
  - geometrical <-> topological
- Discussion & Open Problems

PROBLEM STATEMENT

- Mesh Repairing Algorithm:
  - Input surface mesh M → modified output M′
    - some specific defects removed
    - others possibly introduced
- Selection Procedure:
  1. What is the upstream application?
     → determines characteristics and defects of M
  2. What is the downstream application?
     → determines requirements on M′

PROBLEM STATEMENT

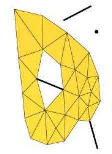
3. Based on this information:
   → is it necessary to repair M?
4. If repairing is necessary:
   → Is there a suitable direct algorithm?
5. If direct repair is not possible:
   → can several methods be used in sequence?
6. If not:
   → There is a gap in the state of the art …

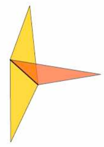
DEFECT TYPES

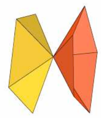
- We distinguish issues about:
  - Local connectivity
    - „The set of polygons does not represent a combinatorially manifold simplicial complex“
  - Global topology
    - „The overall topological structure (number of components, genus, orientability) is wrong“
  - Geometry
    - „The geometric realization is flawed (holes, gaps, noise, …)“

DEFECT TYPES

- Local connectivity
  - Isolated vertices
    - „A vertex that is not a face of any other simplex"
  - Dangling edges
    - „Edges without any incident triangles"
  - Singular edges
    - „Edges with more than two incident triangles"
  - Singular vertices
    - „Vertices with a non-disc neighborhood"

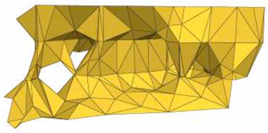


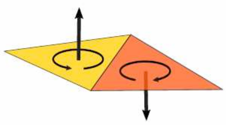




DEFECT TYPES

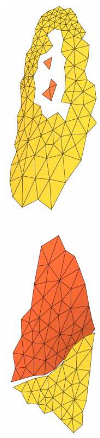
- Global topology
  - Topological noise
    - „Tiny spurious handles or tunnels"
    - „Tiny disconnected components"
    - „Unwanted cavities"
  - Orientation
    - „Incoherently oriented faces"





DEFECT TYPES

- Geometry
  - Holes
    - „Missing pieces within a surface“
      - e.g. due to occlusions during capturing
  - Gaps
    - „Missing pieces between surfaces“
      - e.g. due to inconsistent tessellation routines
  - Cracks / T-Junctions

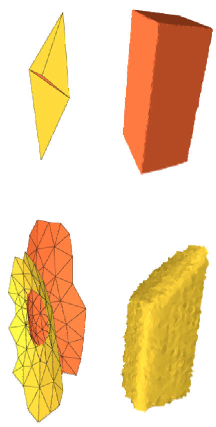


→ Inherently ill-posed

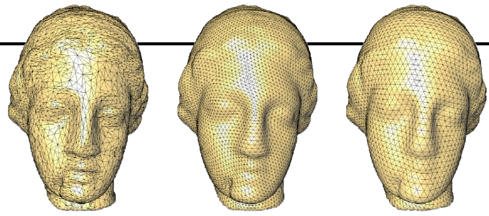
→ Plausible geometry needs to be conceived

DEFECT TYPES

- Geometry
  - Degenerate elements
    - „Triangles with (near-)zero area“
  - Self-intersections
    - „Non-manifold geometric realization“
  - Sharp feature chamfering
    - „Aliasing artifacts due to sampling pattern“
  - Data noise
    - „Additive noise due to measurement imprecision“



DEFECT TYPES



- Geometry
  - Besides the absence of (near-)degeneracies, the general element quality is an important characteristic in several applications.

    → Conversion of meshes to meet such „continuous quality criteria“ is the scope of „surface remeshing“.

UPSTREAM APPLICATIONS

- Upstream applications (or *sources*) characterized by:
  - *Nature*
    - (physical) real-world data <-> (virtual) concepts
  - *Approach*
    - *… employed to convert data to polygon mesh*
- Both aspects can be the source of defects and flaws.

UPSTREAM APPLICATIONS

- Nature
  - *Designed*
    - Basic concept is an abstraction
    - Problems due to:
      - Inaccuracies in the modeling process
      - Inconsistencies in the description/representation
  - *Digitized*
    - Measurement of real-world phenomenon
    - Problems due to:
      - Measurement inaccuracies
      - Measurement limitations

UPSTREAM APPLICATIONS

Nature	noise	holes	gaps	intersections	degeneracies	singularities	topolog. noise	aliasing
Digitized (physical)	**X**	**X**					**X**	**X**
Designed (virtual)			**X**	**X**	x	**X**		

UPSTREAM APPLICATIONS

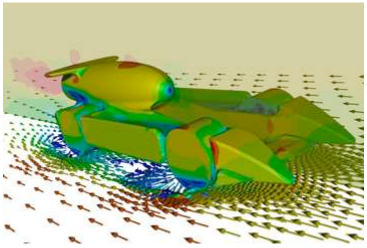
- Approach
  - *Tessellation*
  - *Depth image fusion*
  - *Raster data contouring*
  - *Implicit function contouring*
  - *Reconstruction from points*
  - *Height field triangulation*
  - *Solid model boundary extraction*

UPSTREAM APPLICATIONS

Approach	noise	holes	gaps	intersections	degeneracies	singularities	topolog. noise	aliasing
Tessellation			**X**	**X**	x			
Depth image fusion				**X**	x	x		
Raster data contouring						**X**		
Implicit function contouring					**X**		x	**X**
Reconstruction from points		x	x				x	x
Height field triangulation								
Solid model boundary extract.						**X**		

DOWNSTREAM APPLICATIONS

- We consider prototypical requirements of a sample of the wide application spectrum
  - *Visualization*
  - *Modeling*
  - *Rapid Prototyping*
  - *Processing*
  - *Simulation*



DOWNSTREAM APPLICATIONS

Application Group	noise	holes	gaps	intersections	degeneracies	singularities	topolog. noise	aliasing
Visualization	x	X	x					x
Modeling		X	X		X	x	x	
Rapid Prototyping		X	X	X		X		
Processing	X	X	X	x	X	X	x	x
Simulation	X	X	X	X	X	X	X	x

REPAIR APPROACHES

- We distinguish between two types:
  - Local:
    - handling defects individually by local modifications.
    - Low invasiveness, but only few guarantees
  - Global:
    - typically based on a complete remeshing.
    - High robustness, but often loss of detail

LOCAL APPROACHES
Gap Closing

Gaps – Nature and origin

- Mostly between connected components of a mesh
- Produced by tessellation, round-off, conversion errors, inaccurate trimming, …
- Made of separated chains of edges
- Long and narrow

- Most methods match gap boundaries by considering their spatial proximity

Proximity-based approaches

- Merge vertices within a prescribed distance [RW92]
  - to re-unite equivalent vertices that are slightly displaced due to round-off
- Proceed edge-by-edge and progressively “zip” pairs of boundary edge chains [SM95],[BK97]
  - Start from closest pairs to resolve ambiguities
- Consider also “negative gaps”, by clipping and merging [TL94]

Issues of pairwise boundary stitching

- Pairwise processing does not introduce singular edges
- Some gaps remain when only reasonably resolvable into a non-manifold mesh
- Thus, methods are allowed to produce non-manifolds to be able to close all the gaps [BNK02]
- When gap borders are too far away, add a triangle strip inbetween [PMR05]

Advanced Gap Closing

- Only a portion of an edge-chain may be the boundary of a gap, several gaps may cross and meet at common points
  - globally consistent matching of (parts of) boundary curves [BS95]
- Gaps may separate patch boundaries from non-boundary parts of other patches, or even inner parts of different patches
  - Hybrid approach – global remeshing, but only within voxels surrounding the gaps [BK05]

LOCAL APPROACHES
Hole Filling

Earlier Hole Filling

- Holes are larger than gaps - fixing them by simple zipping is too coarse
- Earlier methods detect holes by looking for closed loops of boundary edges
- These "simple" holes can be patched by triangulating their boundary loops
- Efficient solution, but still too coarse for large holes

[BW92; MD93; VPK05; RW97]

3D Polygon Triangulation

- Need to parameterize the boundary loop over the plane
- Might be a hard task → simulated annealing to "unfold" complex boundaries [BWS*10]
- Some 3D polygons cannot be triangulated without self-intersections [BS95]

Beyond Triangulation

- For large holes, insert additional vertices within the triangulation while trying to:
  - meet the Delaunay criterion [PS96]
  - reproduce the sampling density of the surrounding mesh [Lie03]
  - consider internal angles, dihedral angles, and areas simultaneously [WWP10]
- Dynamic programming makes these algorithms very inefficent for larger holes

Other approaches

- Advancing front with Poisson equation for smoothness [ZGL07]
- Radial Basis Functions [BPB06]
- NURBS fitting [KSI*07]
- Curvature energy minimization [Lev03; PMV06]
- Moving Least Squares projection [WO07; TC04]

Self-intersections

- When additional vertices are inserted, methods may try to create intersection-free patches
  - [TC04] – Insert one triangle and check for intersections. Might fail in producing the patch
  - [WLG03] – Randomized optimization by simulated annealing. Less failures, but still not guaranteed

Issues

- Holes might have complex topologies
- The algorithms cited consider one loop at a time. Not suitable for e.g. holes with “islands”

Exploit the volume

- If input is free of self-intersections, singularities and degeneracies:
  - Input can be used to build a Constrained Delaunay Tetrahedrization
  - Using graph-cut techniques, proper facets of tetrahedra can be selected to fill holes with multiply-connected boundaries
  - Algorithm due to [PR05]

LOCAL APPROACHES
Mesh Completion

Beyond smooth patches

- Reproducing morphological details within the patch might be useful
- Mesh completion algorithms attempt to solve this problem
- Try to copy structure, textures and features from intact parts

Types of completion algorithms

- Mainly two classes:

- Modification of template shapes
  - Use available geometry to select a proper "template" from a DB and adapt the latter
- Creation of the missing geometry
  - Use available geometry to create detailed patches that complete the missing parts

Template-based completion

- Often require user suggestions to start the alignment, e.g. correspondences, feature markers, ...
- Useful just for objects that can be clearly classified into one of few categories
  - human head scans [BV99; KHYS02; BMVS04]
  - bodies [ACP03; ASK*05]
  - teeth [KHYS02; SK02]

Creation of missing geometry

- Patches can be easily copied, but it's hard to merge them continuously with mesh
- Many algorithms work on point-based representations instead ([SACO04; BSK05; BF05; PGSQ06; XZM*07]
  - Need to appropriately sample if input is mesh
  - Need to triangulate the resulting patches

Inter vs. Intra-shape similarities

- The necessary patches can be
  - copied from other parts of the same model [SACO04]
    - Useful if textures and features must be replicated
  - synthesized from the available geometry of a set of meshes of the same class as the input [PMG*05]
    - To achieve correct global structure and topology

LOCAL APPROACHES
Degeneracy and Self-Intersection Removal

Types of input

- Degeneracies (or near-degeneracies) are often the source of instabilities
- Algorithms that fix them might need to use robust geometric predicates
- Two types of input
  - Tessellated CAD models
  - Digitized models

Slicing CAD models

- Needle-like triangles are simply removed by collapsing the edge opposite to the degenerate corner
- Caps can be split into needles
  - To avoid loops, [BK01] employ a slicing technique
- When done, iterative edge collapses can simplify the model while removing all the needles

Treating raw digitized meshes

- Here we can count on a rather uniform and dense sampling [Att10]
- Needles can still be removed by collapsing the opposite edge
- Caps can be resolved by “swapping” the edge opposite to the flat corner
- Guaranteed to converge for exact degeneracies, not for near-degeneracies

Manifold meshes

- Keeping the mesh manifold while removing the needles might be impossible for non-exact degeneracies
- E.g. a long and thin cylinder might be tessellated with only triangles with angles below the given threshold

Self-intersections

- Two problems
  - Detection
  - Resolution
- All-with-all intersection tests lead to quadratic complexity → unaffordable
- Need spatial subdivision to reduce the search space

Local remeshing

- [BK05] use a voxel grid to locate self-intersections efficiently
- If a voxel contains intersecting triangles, the surface within the voxel is remeshed
- Same process to locate and fix small gaps
- Modifications occur only near the flaws, thus the approach is local
- Useful to fix tessellated CAD patches with approximated trimming curves

Removing triangles

- In digitized meshes intersecting triangles are small and can be simply removed, and the resulting holes filled
- Approach used in [Att10], where several repairing tasks are performed in sequence
- No parameters are necessary

Robustness issues

- Finite precision may be insufficient to represent the intersection points
- [CK10] use an intermediate BSP representation
  - Fast and robust
- [GHH*03] use arbitrary precision arithmetic
  - More precise
  - Slower; requires more resources

LOCAL APPROACHES
Sharp Feature Restoration

Interactive approach

- In [KB03], for each corrupted sharp edge, the user draws a “fishbone” structure (spine and orthogonal ribs)
- A tessellation of this structure replaces the original chamfer with a sharp patch
- Useful also to model arbitrary profiles to be swept along the edge

The EdgeSharpener approach

- Detect smooth regions by analyzing the dihedral angle at mesh edges
- Create sharp features as intersections of planar extrapolations of smooth regions
- Suitable for meshes interpolating points of feature-insensitive sampling patterns
- Automatic algorithm presented in [AFRS05]

Other automatic approaches

- In their hole-filling algorithm [CC08] include a sharpness-dependent filter to reconstruct features
- In [Wan06] both sharp features and smooth blends between smooth regions are reconstructed
  - To differentiate between these two cases, the user is required to specify a parameter

Newly-introduced flaws

- All the methods discussed “add material” to reconstruct the features → potential self-intersections
- The EdgeSharpener method might produce degenerate triangles while splitting the chamfers

LOCAL APPROACHES
Mesh Denoising

Laplacian smoothing

- Iterative algorithm
- For each iteration, compute the eventual position of each vertex as the center of mass of the neighbors
- Tends to “shrink” the shape

Non-shrinking methods

- λ/μ algorithm [Tau95] - a modification of the Laplacian smoothing
- For each iteration, two sub-iterations are performed
  - One inward diffusion, controlled by λ
  - One outward diffusion, controlled by μ
- Alternatively, [VMM99] push vertices towards original surface after each Laplacian iteration

Morphology-aware denoising

- Previous algorithms smooth everything
- If morphological features are important, [FDCO03] propose to use a bilateral filter as done in image processing
  - User need to set two parameters
- [JDD03] propose an alternative approach which is non-iterative
  - Can treat polygon soups as well

Denoising and sharpening

- In both [HP04] and [SRML07], smooth regions are denoised while potential sharp edges are actually sharpened
- For the case of mechanical/man-made objects, [FYP10] propose a specific approach that is more accurate

Newly introduced flaws

- All the described methods move vertices to new positions
- Some of them reconstruct sharp features, thus add material to the object
- Typically, no checks that these modifications do not produce self-intersections or degeneracies are performed.

LOCAL APPROACHES
Topology Correction

Types of approaches

- Based on prior knowledge of the topology
  - E.g. Reconstruction of human cortex from MRI [XPR*02]: known to be genus 0
- Involving user interaction
  - [SLS*07] – Ask the user to resolve possible ambiguities
- Based on threshold parameters
  - [GW04] – Remove all handles smaller than a threshold size

Genus-0 surfaces

- [FLD01] inate the input mesh (brain cortex) by alternating steps of Laplacian smoothing and radial projection
- In such a spherical paramaterization handles are actual mesh folds
- Folds are replaced by disk-like patches, and the parameterization is reversed
- Other methods: [SL01], [HXBNP02] (genus-N)

Removing small handles

- [ESV97] roll a sphere of radius α over the mesh and fill up all the regions that are not accessible to the sphere
- This removes tiny handles and tunnels, but also spoils concave edges and is unsuitable for meshes with boundary
- Appears to be extremely difficult to implement; robustness issues may arise

Topological Noise Removal

- [GW04] use a wavefront traversal to find if the mesh has local handles or tunnels (user-defined size)
- Non-separating cuts are identified and the mesh is cut and sealed along them
- [AF06] propose an accelerated method for digitized meshes which exploits wavefront splitting points

Newly introduced flaws

- All these methods add material
- Typically, no checks that these modifications do not produce self-intersections are performed
- Other algorithms exploit explicit definition of the volume to avoid this problem
  - If you have a mesh, voxelization modifies it everywhere, so these must be considered global approaches

Voxel-based topology correction

- [WHDS04] use Reeb graphs to locate handles in voxelized shapes
- If handle is small (measured by short non-separating cycles), volumetric data is processed to remove it
- If model has numerous handles, topology-sensitive carving [SV03] is faster though less precise

Voxel-based topology correction (2)

- For huge voxelizations (e.g. 4096^3), [ZJH07] use discrete curve skeletons
- [JZH07] make it possible to actually edit the topology of an object so as to make it equivalent to that of a given target shape

GLOBAL APPROACHES

GLOBAL APPROACHES

- Approaches discussed so far are local
  - remove single defects (holes, singularities, self-intersections, ...) mainly individually.
- Absence of individual defects not required for their own sake:
  - part of greater requirement for manifoldness.
- Achieving this by sequential local operations is extremely difficult:
  - new defects can be introduced
  - ambiguities are hard to resolve in a local manner.

GLOBAL APPROACHES

- Global repair methods can be advantageous in this regard.
  - consider mutual relation of defects for better ambiguity arbitration.
  - possibly employ intermediate volumetric representation:
    - Guarantees that the result is a manifold surface of some solid.
    - Disadvantage: often complete conversion and remeshing necessary → invasive, loss of detail

GLOBAL APPROACHES

- Intermediate volumetric representation
  → the repair task boils down to deciding which parts of the volume are inside and outside.
- We can group the global methods by how this decision is performed and by their input requirements
  - Input without significant gaps and holes
  - Input with normal or orientation information
  - Arbitrary input

GLOBAL APPROACHES

- Input without significant gaps and holes
  - Rasterization into voxel grid representation,
  - Determination of inside/outside volume by flood-filling,
    - from given seed points [OSD97]
    - from a point at infinity [ABA02]
  - Reconversion to polygon mesh by contouring.

  - Not possible for gaps or holes beyond voxel size.

GLOBAL APPROACHES

- Input with orientation information
  - often available due to the acquisition process (e.g. line-of-sight of the laser scanner).
  - Inside/outside decision by:
    - Line-of-sight carving [CL96]
      + line-of-light carving [FIMK07]
    - Diffusion-based propagation [DMGL02]
      + feature sensitivity [GLWZ06] [Mas04]
    - Moving Least Squares [SOS04]
    - Surface area minimization [SI03] [SI08]

GLOBAL APPROACHES

- Arbitrary Input
  - Various inside/outside decision principles:
    - Parity Counting
    - Ray Stabbing
    - Morphology & Flooding
    - Graph Cutting
    - Membrane Shrinking
    - Optimization of global consistency functional

GLOBAL APPROACHES

- Parity Counting
  - Consider intersections of rays with the object [NT03] [Ju04] [SWT06].
- Ray Stabbing
  - Combine findings from multiple ray directions to be less affected by holes and larger gaps [NT03].

GLOBAL APPROACHES

- Morphology
  - Local closing operations to fill holes and gaps, flood-filling to determine outside [BPK05].
  - Global membrane shrinking [HRA*11].
  - Improve final surface smoothness using graph-cut [HK06].

GLOBAL APPROACHES

- Global Optimization [MF97]
  - Surface-aligned volumetric representation (BSP-based)
  - Determine optimal inside/outside labels for cells such that the output surface maximally conforms with the (partial) input.
  - The geometry of the resulting hole-filling patches is rather random and can be unpleasing in case of larger holes.

DISCUSSION & OUTLOOK

- Widely varying hardness of repair tasks depending on the defects involved.
  - e.g. consistently orienting faces is easily formalized and solved,
  - but filling complex holes or plausibly removing intersections requires a non-trivial approach and intricate case-by-case study due to ambiguities.
- Despite the vast number of existing techniques, there is room for future investigation, especially for the *hard* cases.

DISCUSSION & OUTLOOK

- Minimally Invasive, but Guaranteeing and Global
  - Hybrid methods that are as accurate as local methods but ensure correctness like global ones.
- High-Level Interaction incorporating Meta-Knowledge
  - Missing data always implies ill-posedness, automatic heuristics have limits → exploit qualified knowledge of the user through intuitive interaction metaphors.
- Vertical Integration to Repair Workflows
  - Local methods often treat one defect type → suitable sequencing of methods necessary.

AVAILABLE REPAIR TOOLS

- Several repair tools are freely available
  - Implementing one or multiple of the covered methods.
- Up-to-date information and references to these tools are available at:

  http://www.meshrepair.org