Considerations toward a Dynamic Mesh Data Structure

Sebastian Pena Serna, Andre Stork, Dierter W. Fellner

Fraunhofer-Institut für Graphische Datenverarbeitung
IGD
Fraunhoferstraße 5
64283 Darmstadt

Tel +49 6151 155 – 468
sebastian.pena.serna@igd.fraunhofer.de
www.igd.fraunhofer.de

Fraunhofer
IGD

Motivation

- 3D shapes is essential for representing 3D physical reality in different domains
- Meshes are a versatile and common representation for the 3D reality
  - The mesh generation process does not always produce quality results
  - The domain application demands special requirements
  - The processes applied on the meshes need to support operations

- Objective: analyze considerations toward designing a data structure for dynamic meshes in a generic and robust manner

Introduction

- Dynamic mesh:
  - Dynamic changes in the geometry of the mesh
  - Dynamic changes in the topology of the mesh

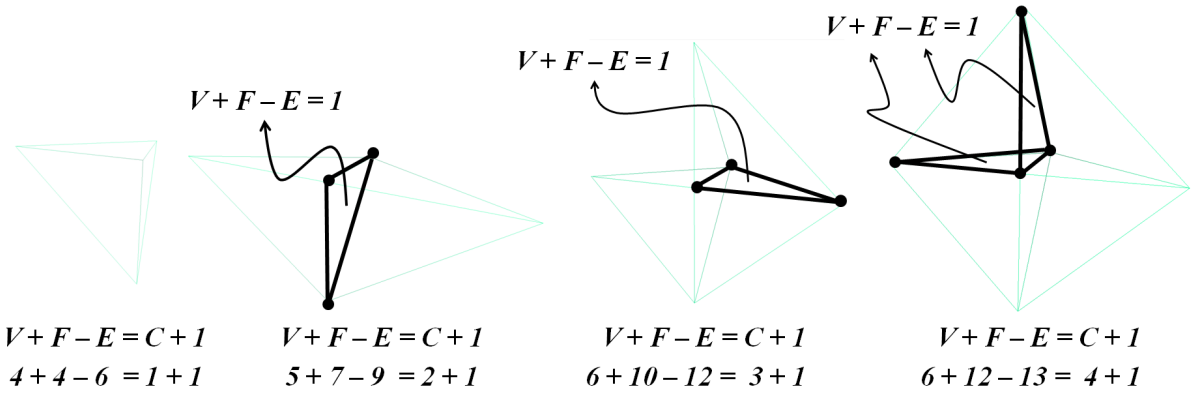
- Considerations:
  - Memory and Performance
  - Neighboring Information
  - Mesh Modifications

Memory and Performance

- Data structures for static meshes compactly encode the topology
- These achieve minimal memory consumption and maximal performance

- Data structures for dynamic meshes cannot encode the topology
- The memory consumption is difficult to optimize
- A mechanism to rapidly update the neighboring information and to increase or decrease the number of entities is needed
  - Memory buffers

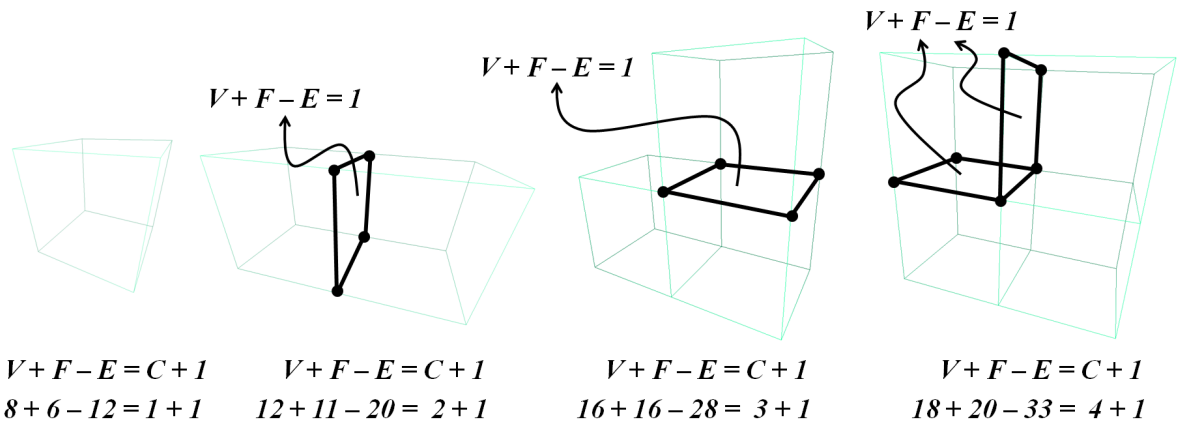
Memory and Performance



Euler Formula for 2-manifold with genus 0

$$F + V - E = 2$$

$$E = F + V - 2$$



For 3-manifold with genus 0

$$V + F - E = C + 1$$

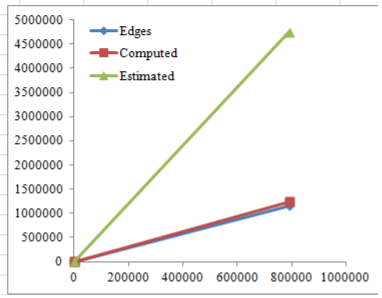
$$E \approx (2 \times V) + C$$

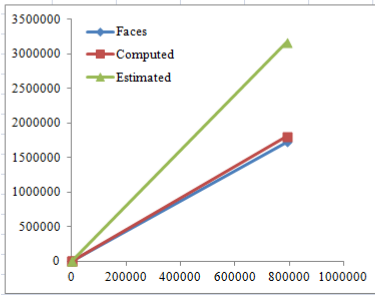
$$F \approx V + (2 \times C)$$

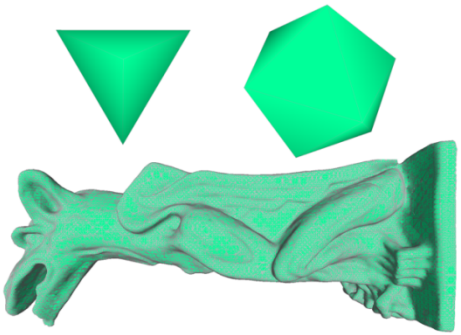
Fraunhofer IGD

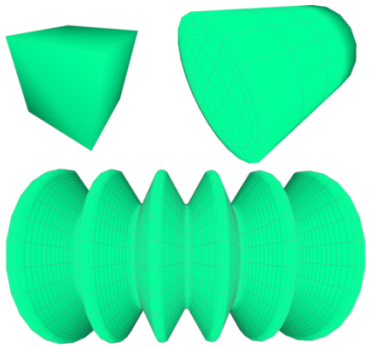
Memory and Performance

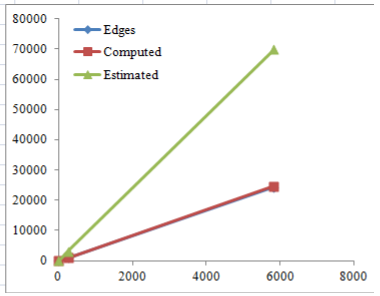
Mesh	V	E	F	C	$V+F-E$	$E \approx (2\times V)+C$	$F \approx V+(2\times C)$
Tetrahedron	4	6	4	1	2	9	6
Sphere	13	42	50	20	21	46	53
Gargoyle	221039	1152799	1723886	792125	792126	1234203	1805289
Hexahedron	8	12	6	1	2	17	10
Cylinder	380	1004	885	260	261	1020	900
Rubber	9367	24354	20812	5824	5825	24558	21015

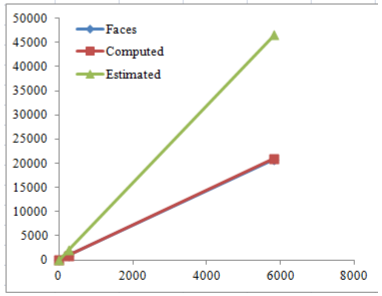








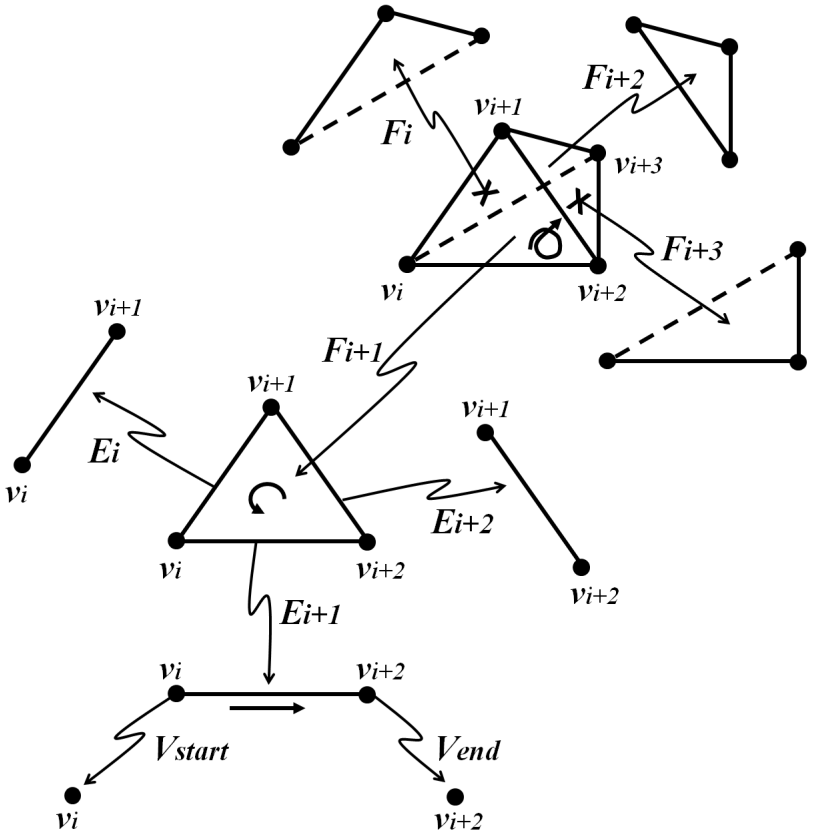




Fraunhofer IGD

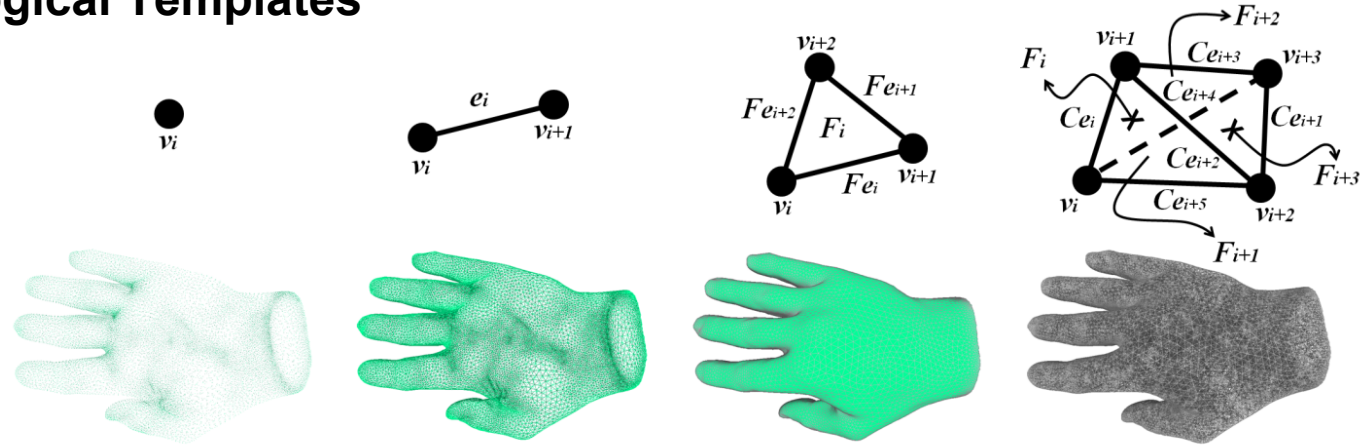
Neighboring Information

- The initialization builds relationships between topological entities
- Hierarchical decomposition: *Cell*, *Face*, *Edge*, and *Vertex*
- n-dimensional entity is decomposed into its n-1-dimensional entities
- The hierarchical decomposition follows always the right-hand rule



Neighboring Information

Topological Templates



F_i	Fe_i	Fe_{i+1}	Fe_{i+2}
V_i, V_{i+1}, V_{i+3}	V_i–$V_{i+1} \rightarrow Ce_i$	V_{i+1}–$V_{i+3} \rightarrow Ce_{i+3}$	V_{i+3}–$V_i \rightarrow Ce_{i+4}$
V_{i+1}, V_i, V_{i+2}	V_{i+1}–V_i	V_i–V_{i+2}	V_{i+2}–V_{i+1}
$V_{i+2}, V_{i+3}, V_{i+1}$	V_{i+2}–$V_{i+3} \rightarrow Ce_{i+1}$	V_{i+3}–V_{i+1}	V_{i+1}–$V_{i+2} \rightarrow Ce_{i+2}$
V_{i+3}, V_{i+2}, V_i	V_{i+3}–V_{i+2}	V_{i+2}–$V_i \rightarrow Ce_{i+5}$	V_i–V_{i+3}

Neighboring Information

- External query functions:
  - *Vts(Ei), Eds(Ei), Star(ei), Star(Vi)*
- Internal query functions:
  - *Eds(Vi), Fcs(ei), Vts(ei)*
- Pre-computed or computed on demand:
  - memory consumption
  - querying performance
  - updating performance
- Topological templates enable both alternatives

Mesh Modifications

- Dynamic meshes deal with changes in the geometry and the topology of the mesh

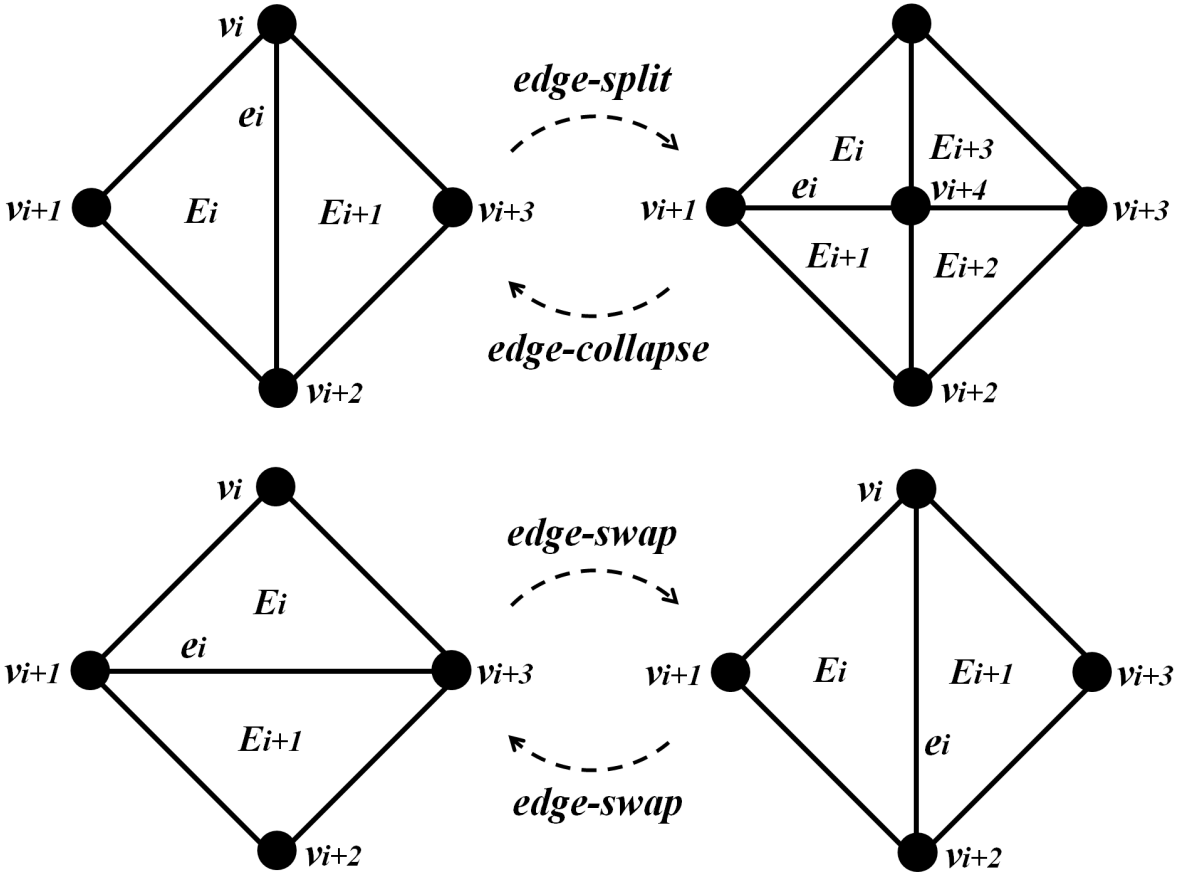
- Geometry remains constant and the topology is changed
- Geometry is changed, invoking modification in the topology

- Typical actions on the mesh are called topological operations
  - Correct degeneracies on the mesh
  - Improve the quality of the mesh

Mesh Modifications

Topological Operations:

- edge-split
- edge-collapse
- edge-swap



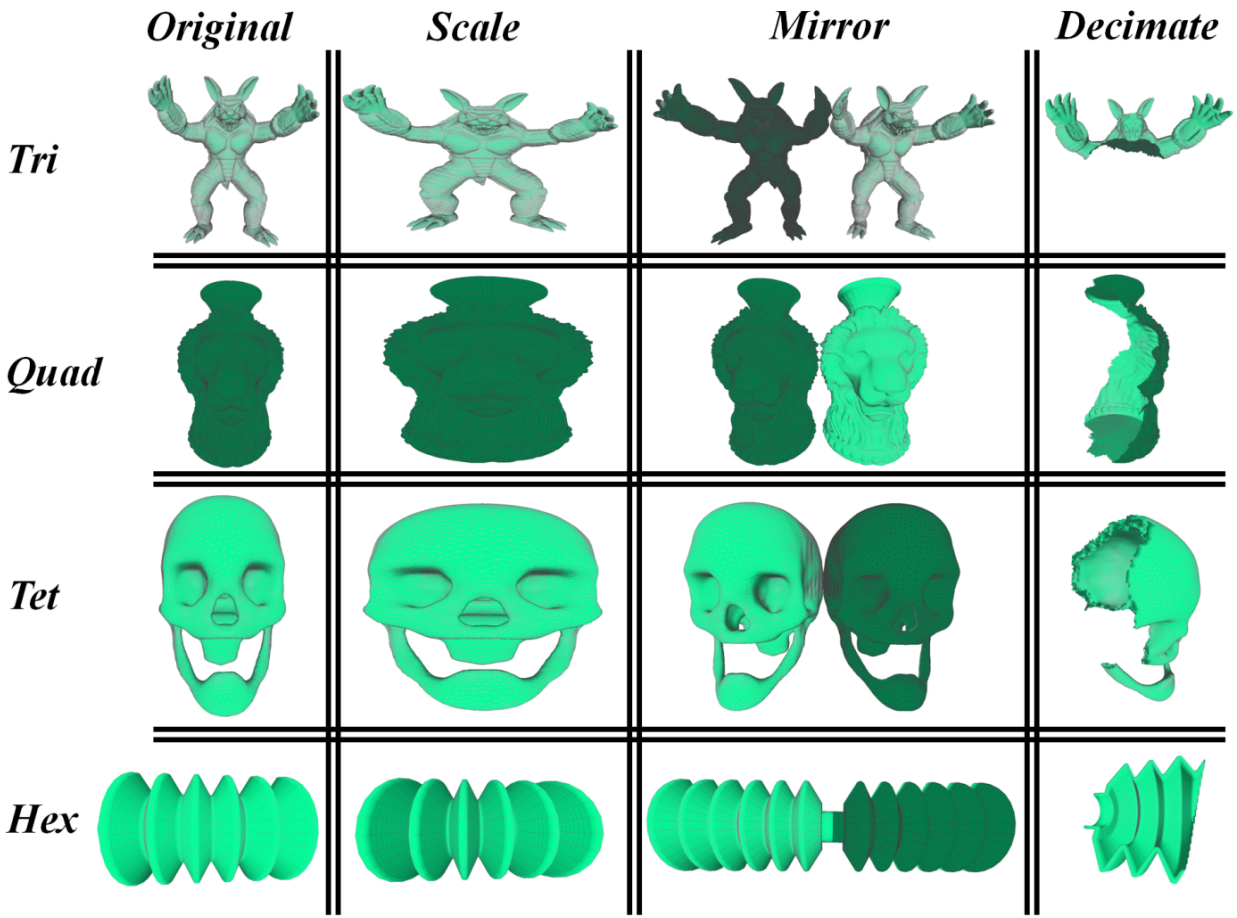
Fraunhofer IGD

Mesh Modifications

- Abstraction from the application
  - *Move(Vi)*
  - *Add(Vi)*
  - *Kill(Vi)*
  - *Add(Ei)*
  - *Kill(Ei)*

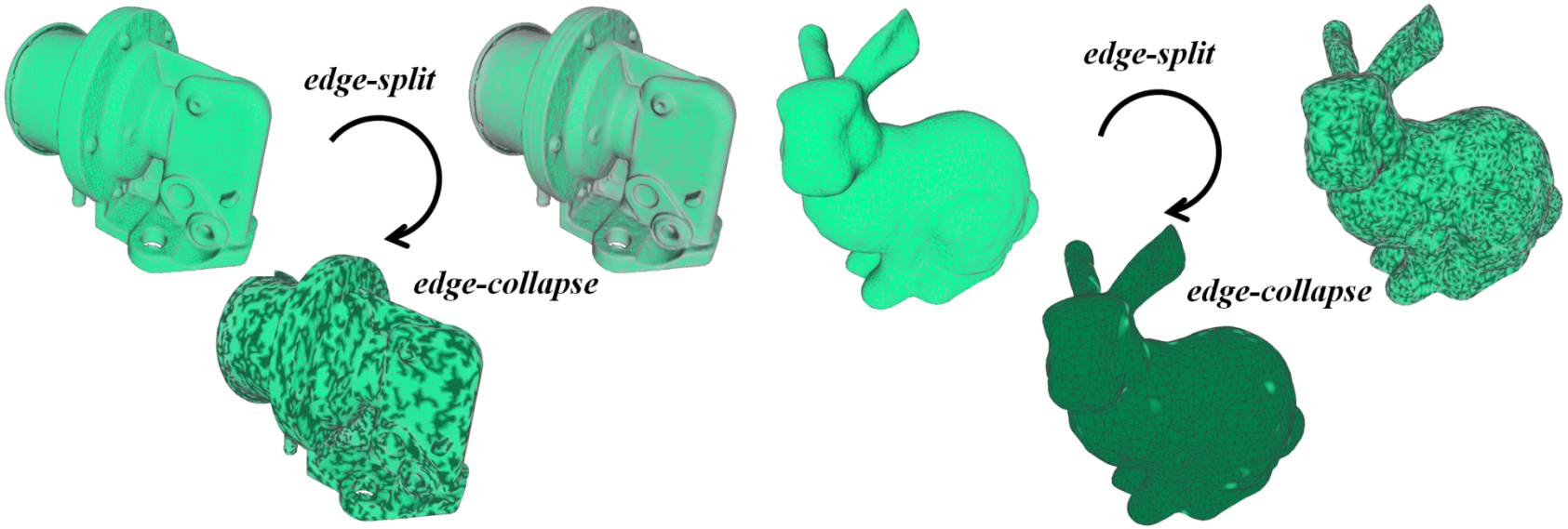
Implementations

- Basic Operations
  - Scale
  - Mirror
  - Decimate



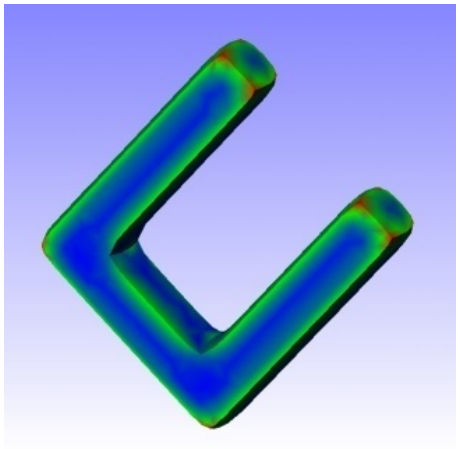
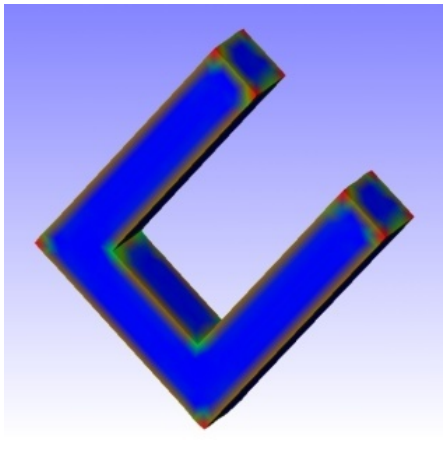
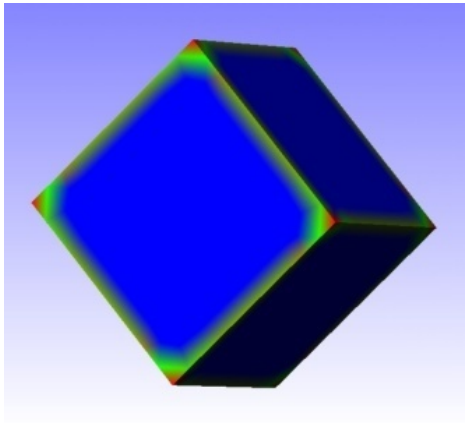
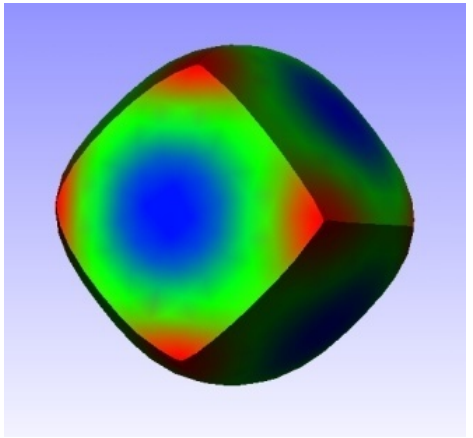
Implementations

- Topological Operations
  - edge-split
  - edge-collapse



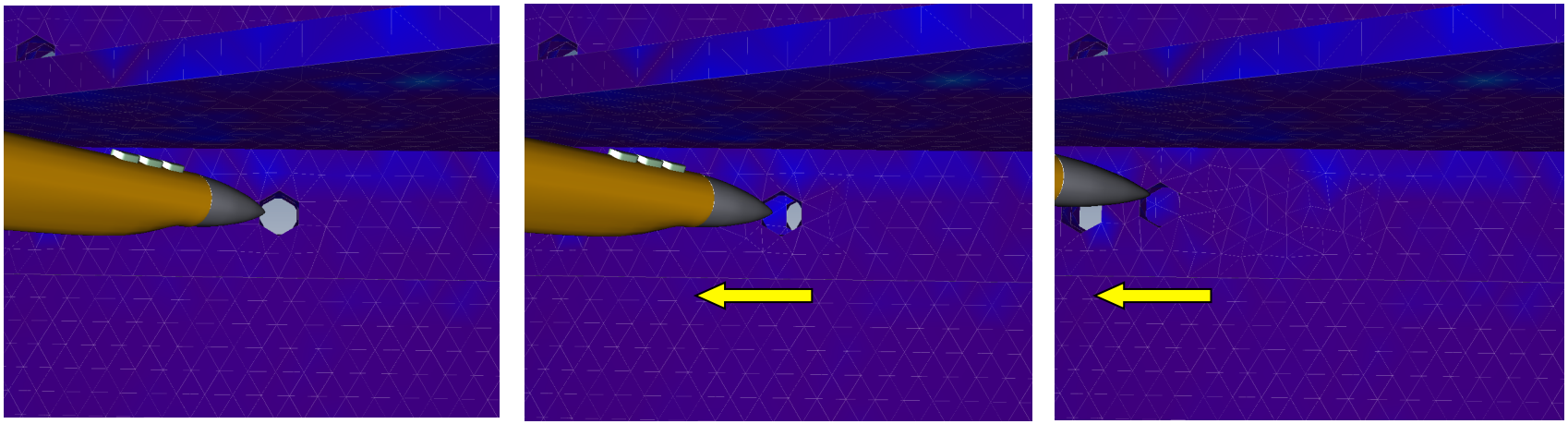
Implementations

- Decreasing and increasing the radius of features

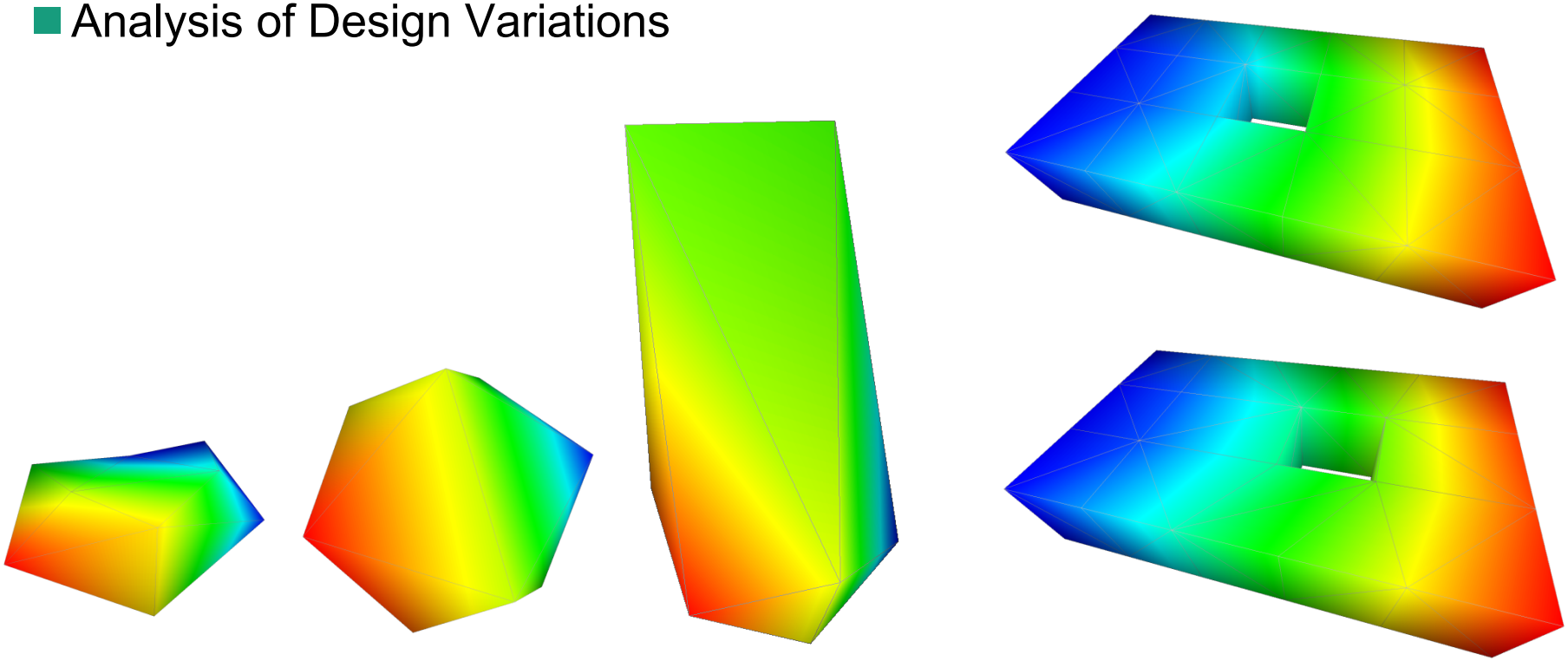
Implementations

- Dragging holes (semantic features)



Implementations

- Analysis of Design Variations



Fraunhofer IGD

Conclusions

- Several mesh data structures for triangular meshes and some for tetrahedral meshes
- Quadrangular and hexahedral data structures are very limited
- Many data structures are designed for minimizing the memory consumption for specific domain applications
- There are not enough data structures, which robustly represent 3D shapes and for supporting dynamic meshes
- Considerations in terms of memory and performance, neighboring information, and mesh modifications
- We will investigate the most appropriate trade-off between memory and performance

Thank You!

Sebastian Pena Serna
Fraunhofer-Institut für Graphische Datenverarbeitung IGD
Fraunhoferstraße 5
64283 Darmstadt

Tel +49 6151 155 – 468
sebastian.pena.serna@igd.fraunhofer.de
www.igd.fraunhofer.de

Fraunhofer IGD