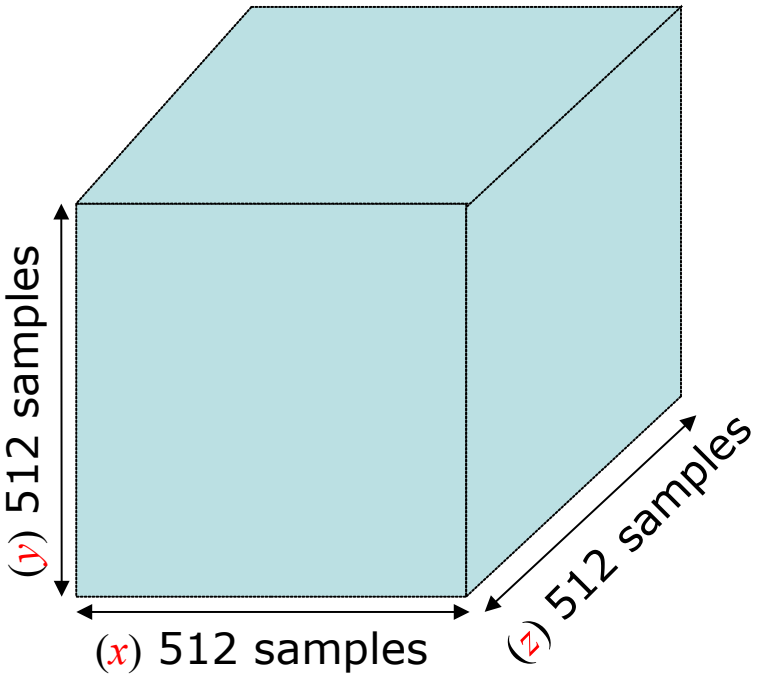
GPU Isosurface Extraction

Leonardo A. Schmitz
laschmitz@inf.ufrgs.br
In cooperation with Carlos A. Dietrich
cadietrich@inf.ufrgs.br
Advised by João Luiz Dihl Comba
comba@inf.ufrgs.br

CNPq CAPES UFRGS

Motivation: Isosurface Extraction

- 3D Volumes: large and increasing.



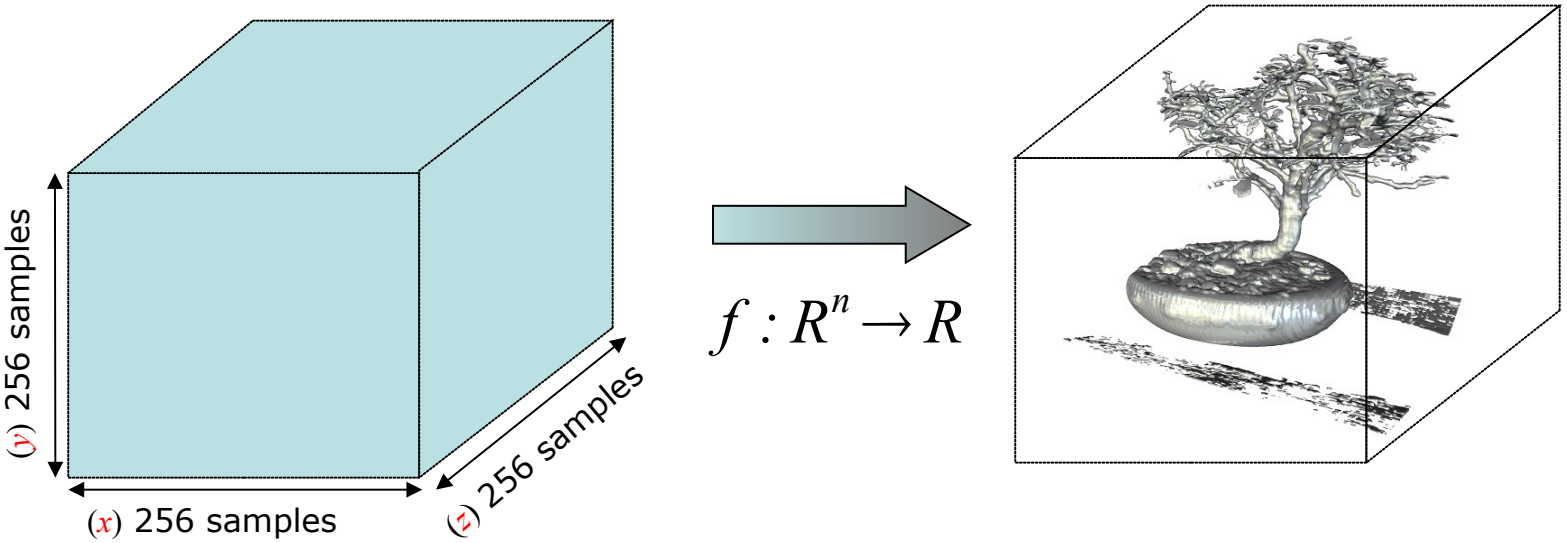
Ex.: www.volvis.org

- ***Head Aneuyrism***
  - *512 x 512 x 512;*
  - *256 MBytes (16 bits);*
- ***Future?***
  - *2048 x 2048 x 2048;*
  - *Available already;*

UFRGS

Motivation: Isosurface Extraction

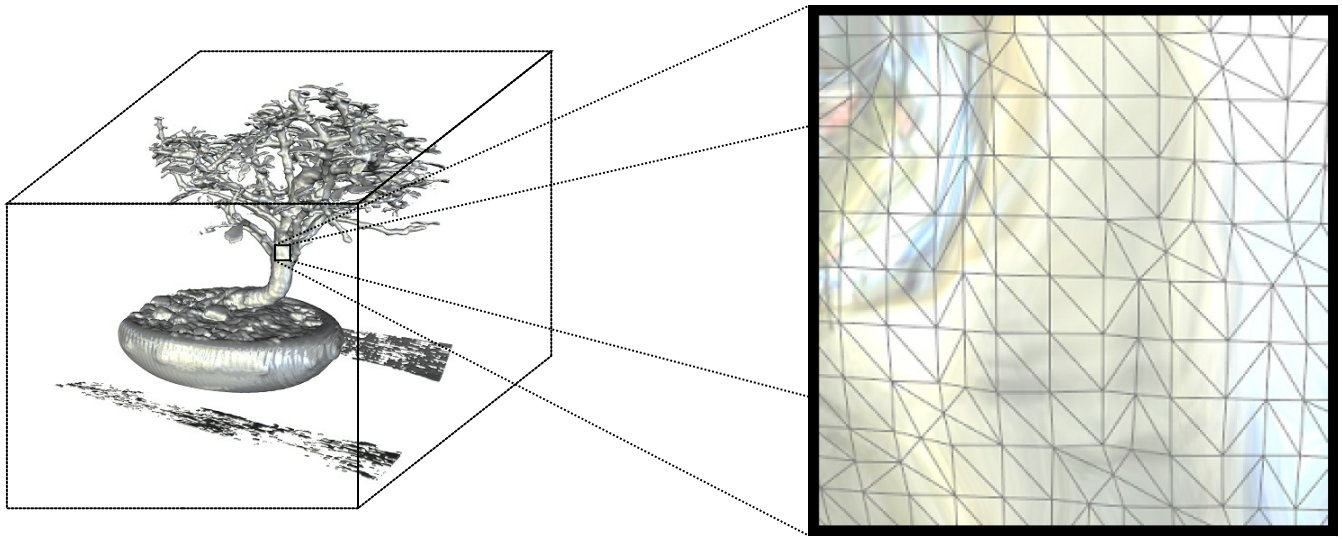
- 3D Volumes: large and increasing.
  - Fast polygonization of isosurfaces.
    - Real-time visualization.
    - On-the-fly input for applications.



UFRGS

Motivation: Isosurface Extraction

- 3D Volumes: large and increasing.
  - Fast polygonization of isosurfaces.
  - High quality triangle meshes (triangle aspect and distance).



UFRGS

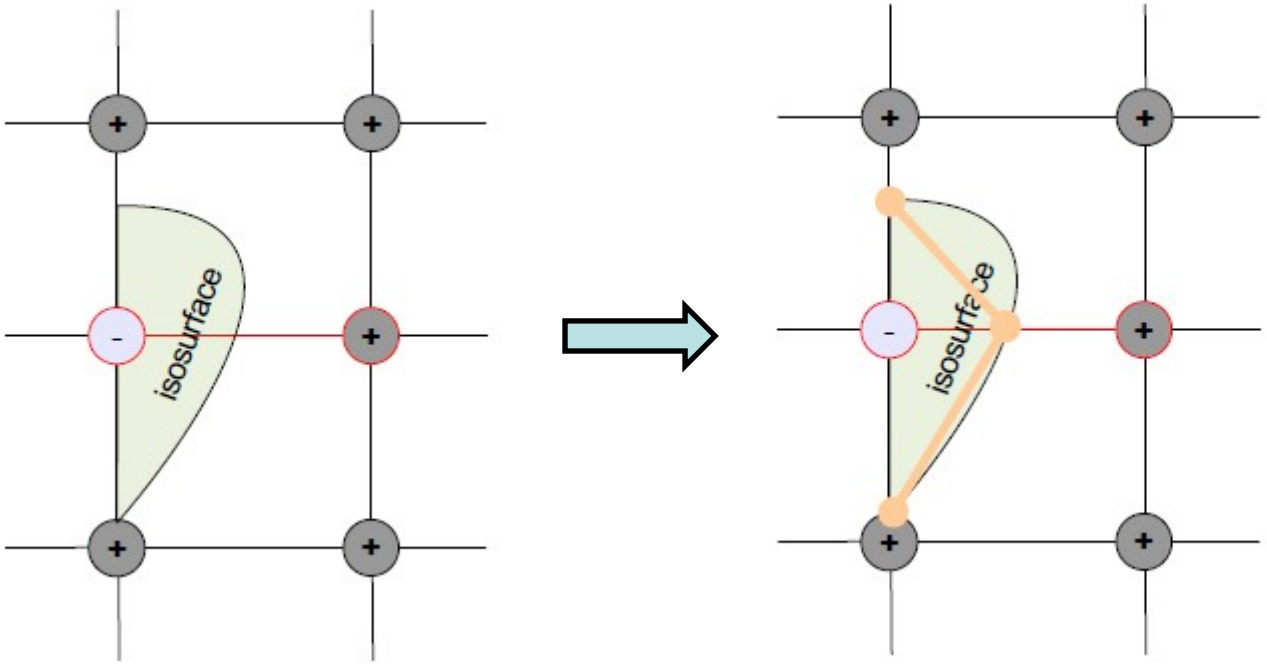
Summary

- Motivation.
- Related Work.
  - Efficiency.
  - Quality.
- GPU Programming.
- Improvement Attempts.
- Results.
- References.

UFRGS

Related Work

- Isosurface extraction with Marching methods.
  - Simple and parallel.
  - Ex.: Marching Cubes [L&C 1988].



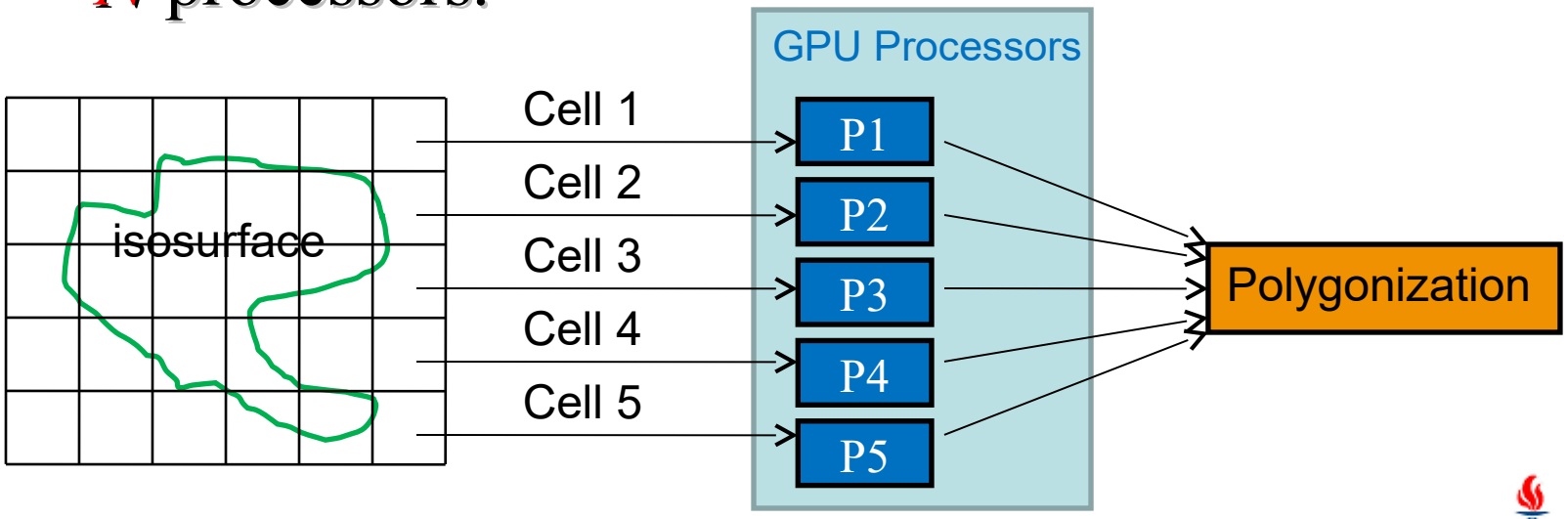
Detection of cut points

Polygonization

UFRGS

Related Work – Efficiency

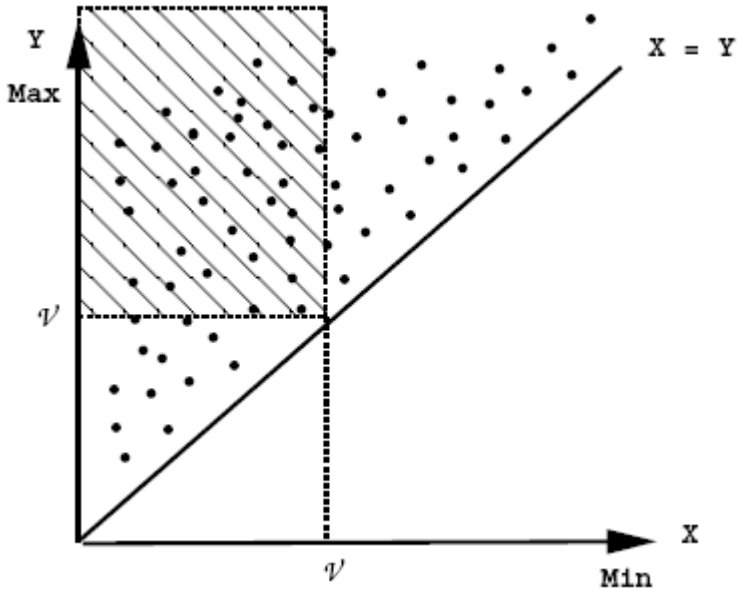
- GPU Parallelism (brute force acceleration).
  - Divide-and-conquer strategy.
    - Marching Cubes [L&C 1988].
    - Dual Contouring [Ju2002].
  - *N* processors.



UFRGS

Related Work – Efficiency

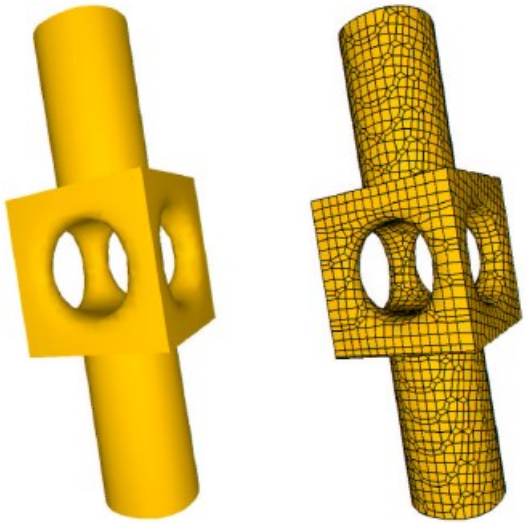
- Span Space (algorithmic acceleration).
  - Early detection of cut points.
  - Separation of maximum and minimum per cell.

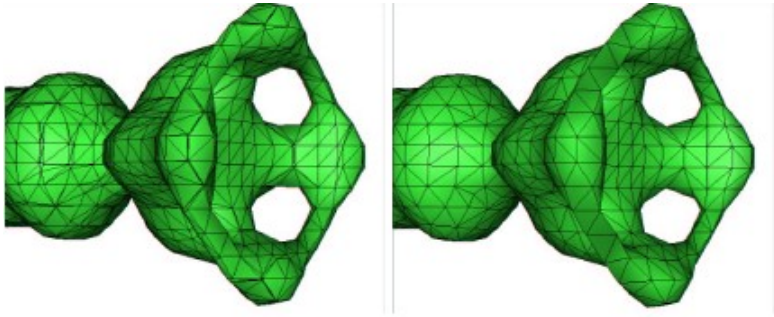


UFRGS

Related Work – Quality

- Isosurface Approximation.
  - Dual Contouring [Ju2002].
  - Extended MC [Kobbelt2001].
- Triangle Quality.
  - MACET [Dietrich2008].
  - Stuffing [Labelle2007].
  - SnapMC [Ramam2008].





UFRGS

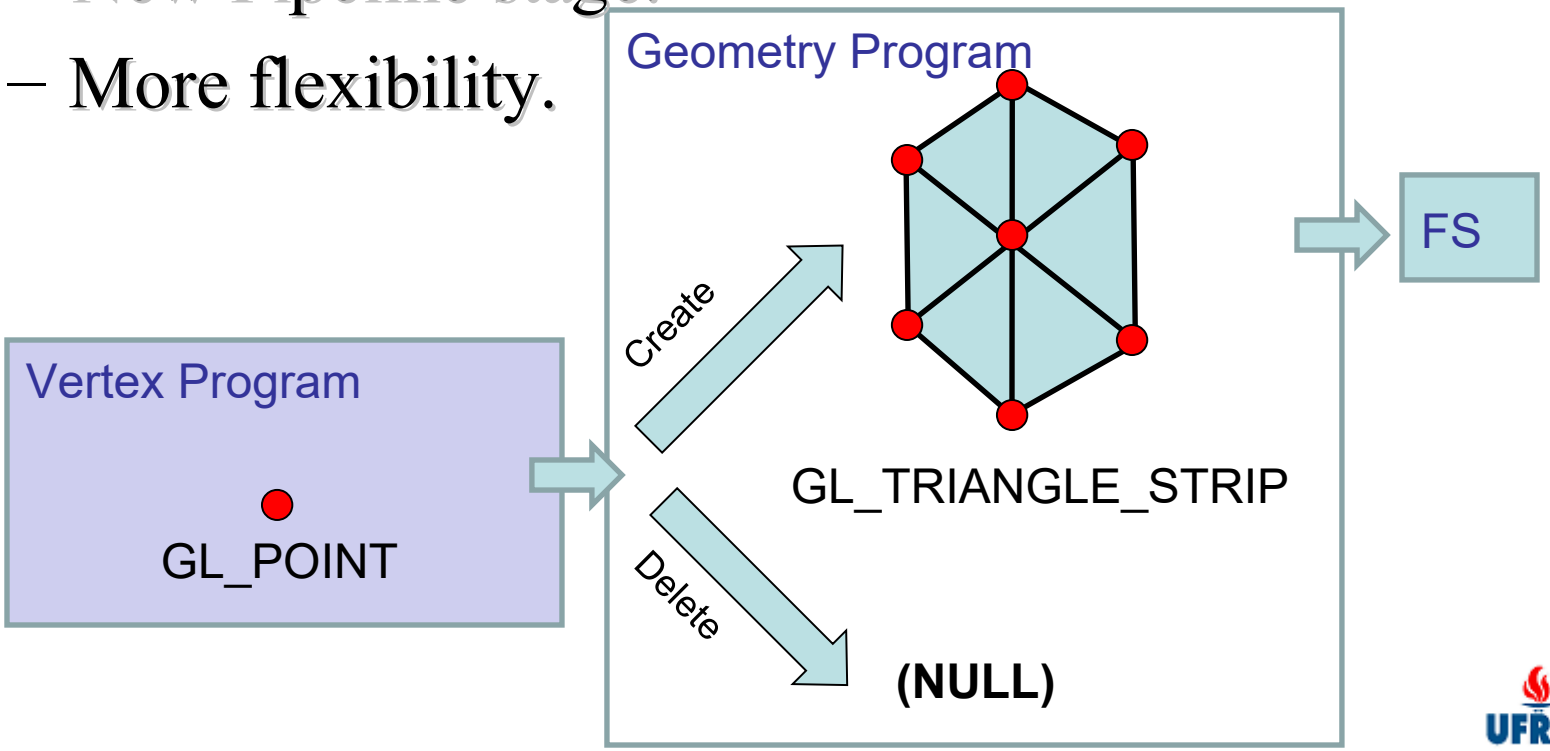
Summary

- Motivation.
- Related Work.
  - Efficiency.
  - Quality.
- GPU Programming.
- Improvement Attempts.
- Results.
- References.

UFRGS

GPU Programming

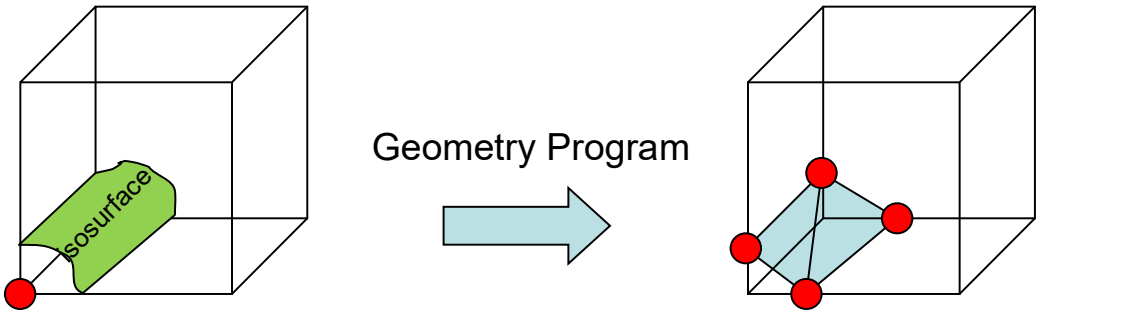
- Geometry Shader (GS).
  - Creation/deletion of primitives.
  - New Pipeline stage.
  - More flexibility.



UFRGS

GS & Marching Cubes

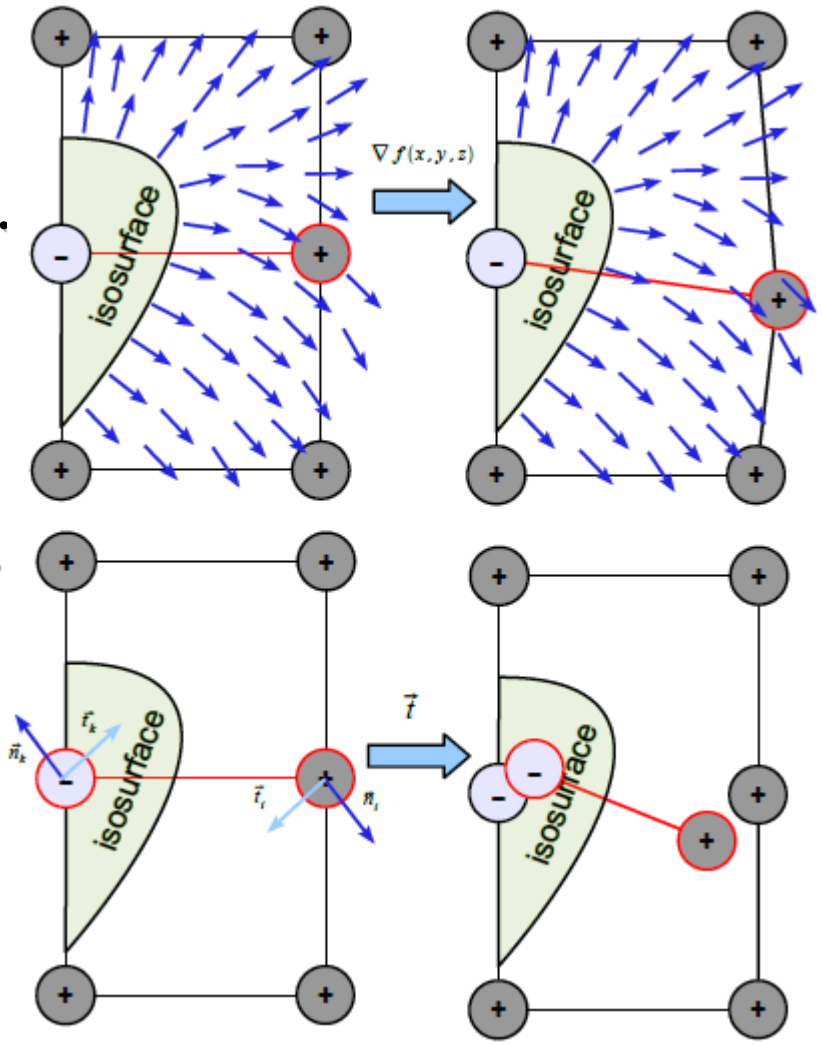
- Information inside GPU.
  - 3D Volume (GL_TEXTURE_3D).
  - Topology table.
- Linear interpolation on active edges.



UFRGS

GS & MACET & Span Space

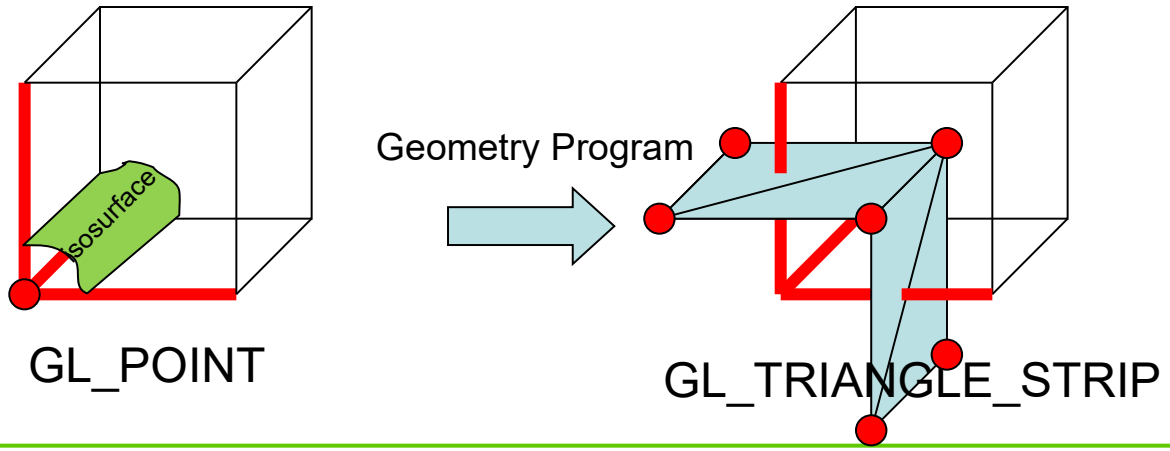
- Algorithm speed.
  - Span Space on CPU.
- Triangle quality.
  - Tangent Transform.
  - Gradient Transform.



UFRGS

GS & Dual Contouring

- Topology table specified.
  - Each cell has a unique x, y and z axis.
  - Each axis can produce at most one quad.
- Particle based feature approximation.



UFRGS

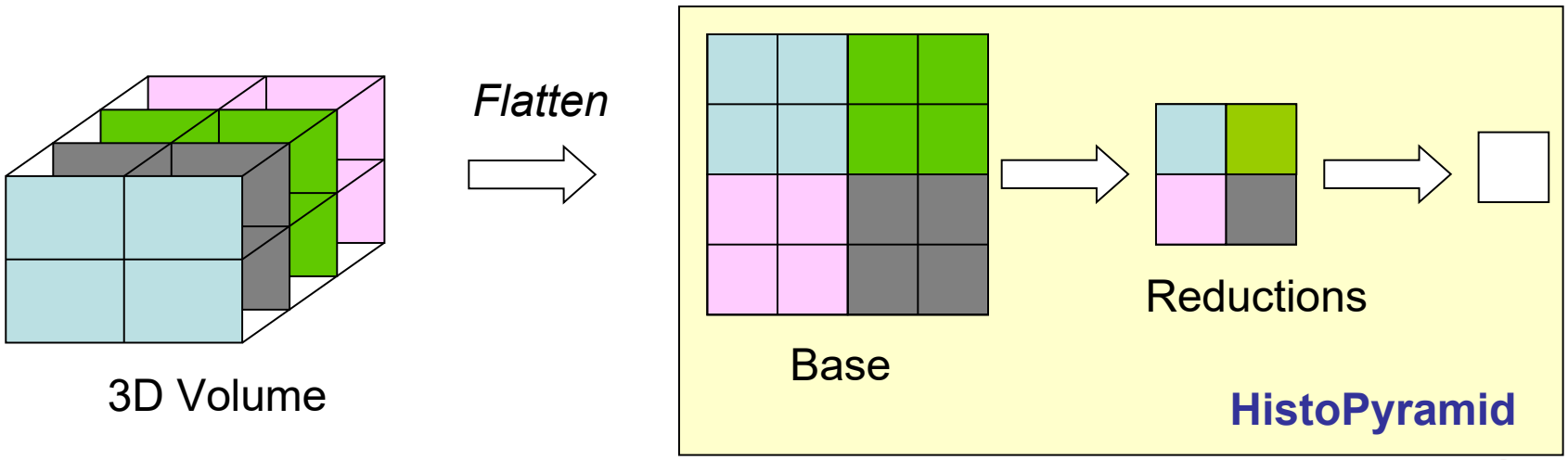
Geometry Shader

- Pros.
  - Allows full isosurface extraction inside GPU.
  - Allows easy Marching implementations.
  - Only needs memory for dataset and topology table.
- Cons.
  - Is much slower than HistoPyramids (multipass technique).
  - Does not allow accessing neighbor triangles after creation.

UFRGS

Histogram Pyramids.

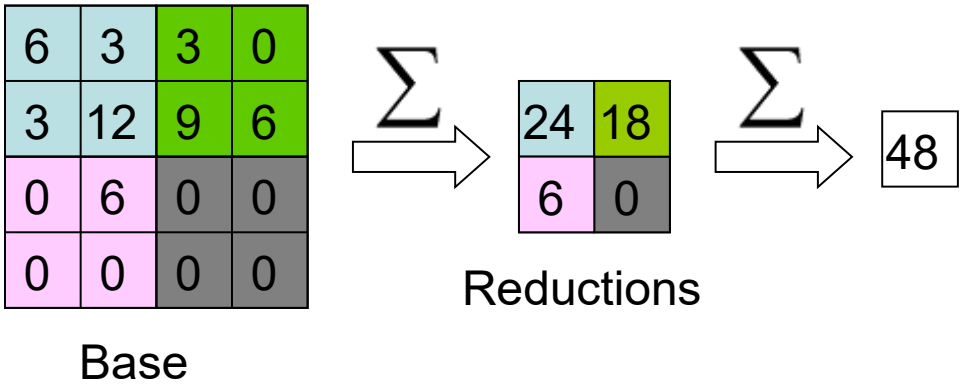
- Creates geometry with multipass rendering.
  - *Frame Buffer Objects*.
  - *Flat3D* [Harris2003].



UFRGS

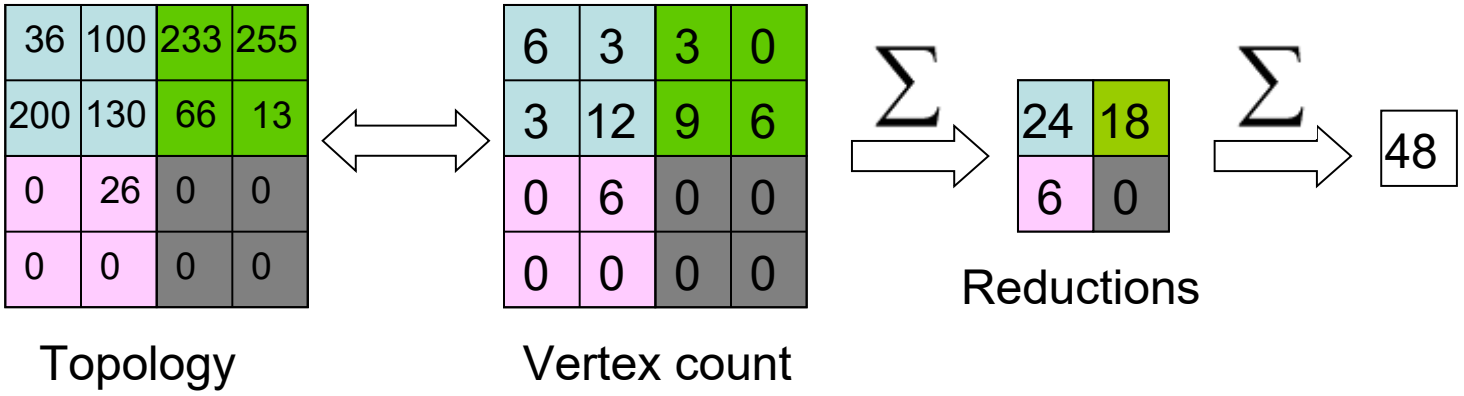
Histogram Pyramids.

- Each element is uniquely identified in base.
  - Top-down traversal with *Vertex Buffer Object*.
  - *Id's* from $i = 0$ to 47 (in the example).



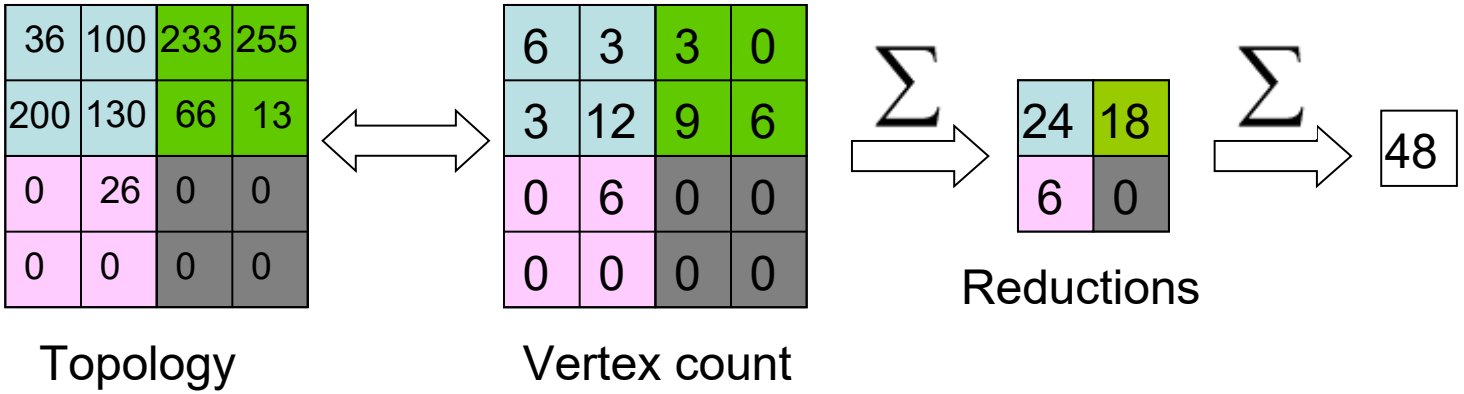
HistoPyramids & Polygonizers.

- Base texture.
  - Vertex count and corresponding topology.
  - Each cell represented in one pixel.
  - Cut points determined in pixel program.



HistoPyramids & Polygonizers.

- Fragment Shader defines geometry which is accessed through the *VBO's id*.
- Ex.: 0x40H (topology) creates 1 triangle (3 vertices).



Histogram Pyramids

- Pros.
  - Extremely fast (33 FPS; bonsai; isovalue *40.5*).
  - It is possible to access the whole mesh within each vertex.
- Cons.
  - Excessive memory use.
  - Square base texture.
  - Very complex to implement.

UFRGS

Summary

- Motivation.
- Related Work.
  - Efficiency.
  - Quality.
- GPU Programming.
- Improvement Attempts.
- Results.
- References.

UFRGS

Improvement Attemps

- How to accelerate even more HistoPyramids?
  - Instead of using *flat3d*, re-order volume in order to reach only cut-points.
  - Fixed Bucket Span Space [Waters2005].

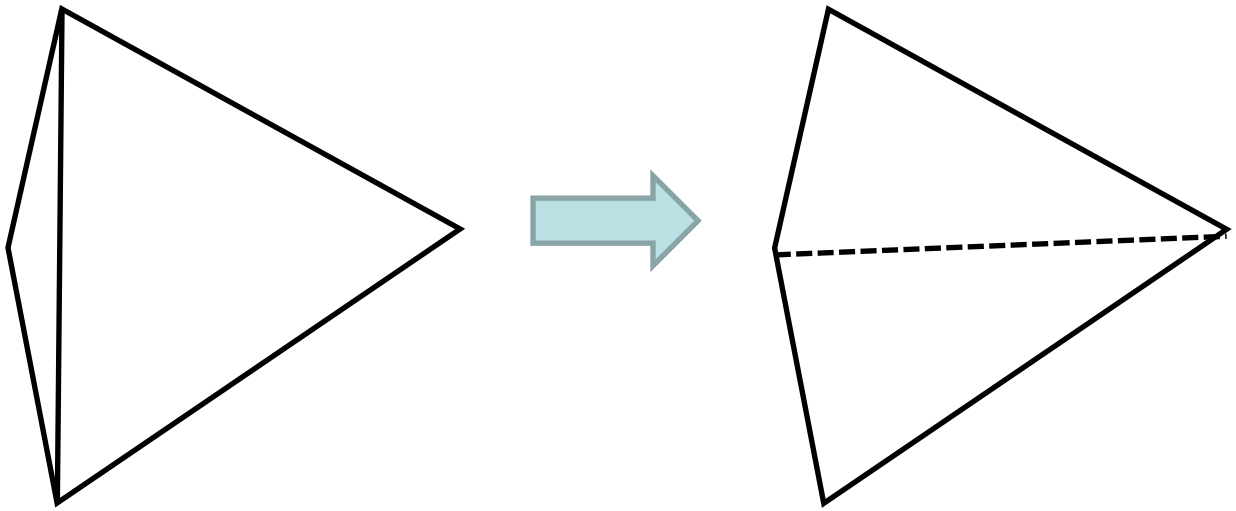
Span Space Texture

Red – active cells;
Black – inside cells;
Gray – outside cells;

UFRGS

Improvement Attemps

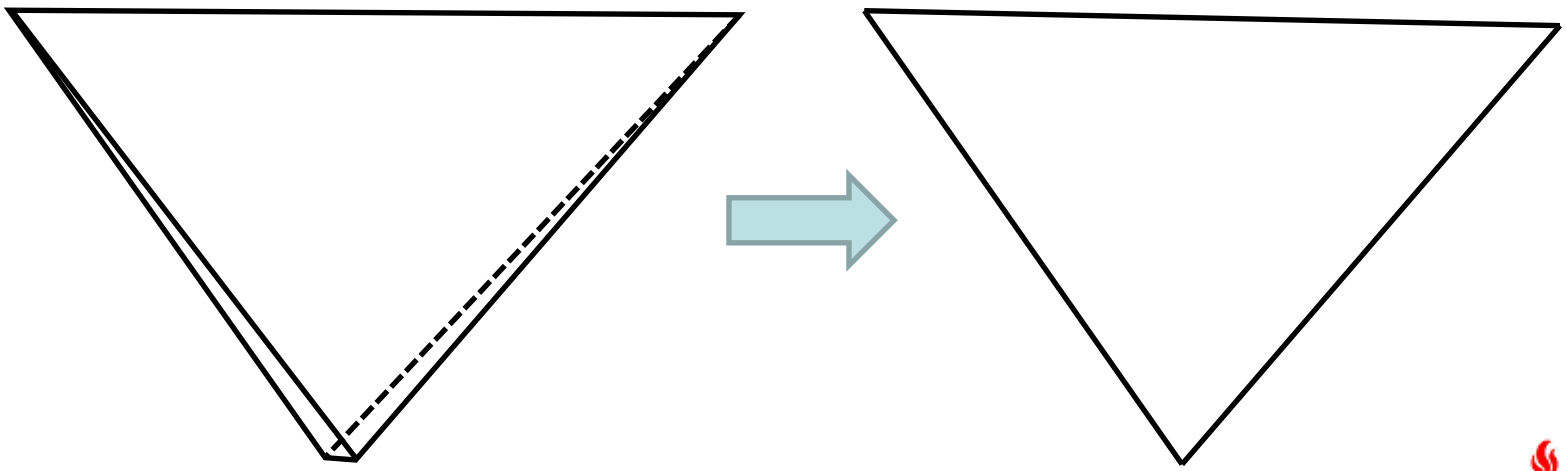
- How to avoid needle-like triangles in DC?
- Two problem cases:
  - Finding Delauney edge.



UFRGS

Improvement Attemps

- How to avoid needle-like triangles in DC?
- Two problem cases:
  - Avoiding useless QEFs computations (with edge collapsing).



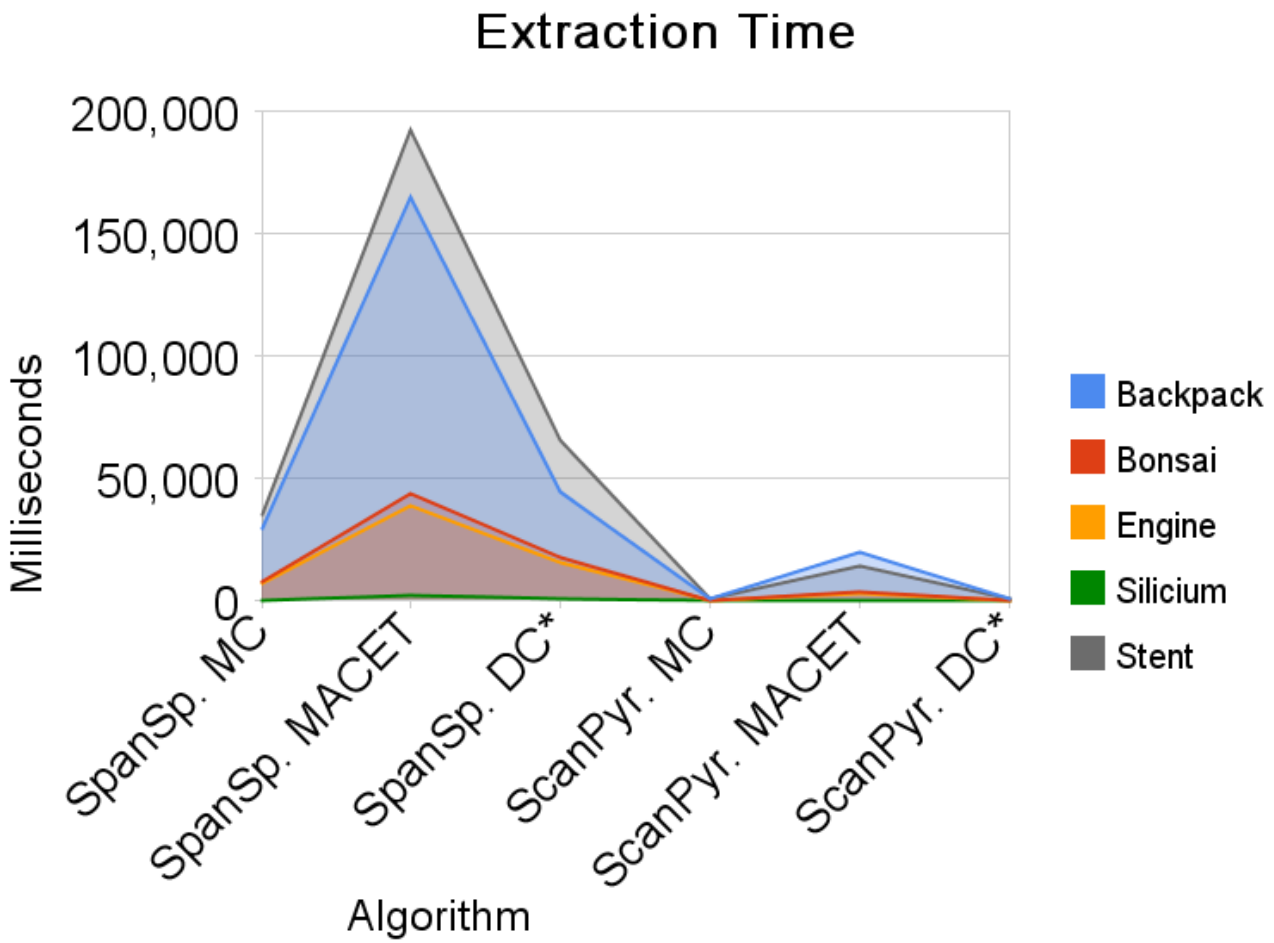
UFRGS

Summary

- Motivation.
- Related Work.
  - Efficiency.
  - Quality.
- GPU Programming.
- Improvement Attempts.
- Results.
- References.

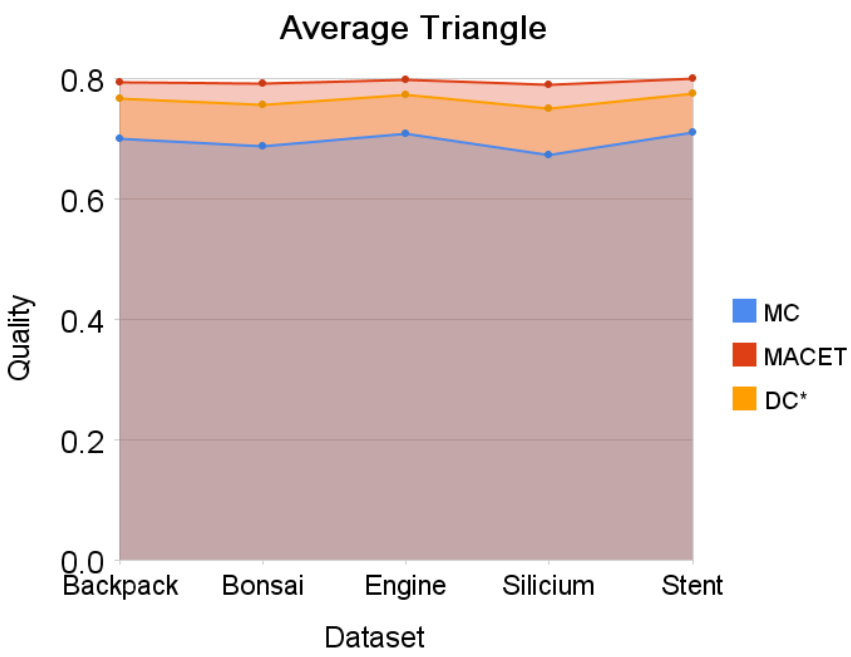
UFRGS

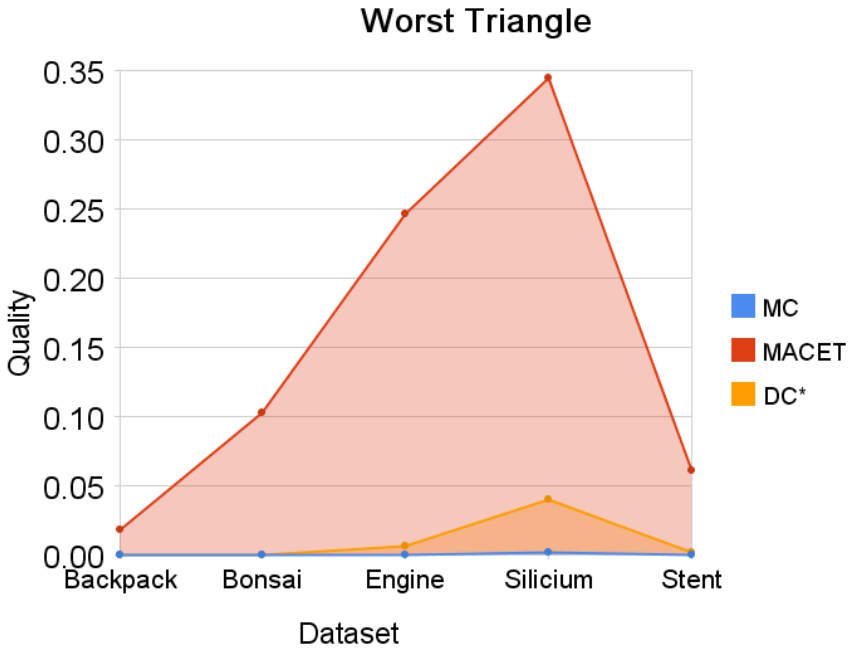
Performance Results (Old)



UFRGS

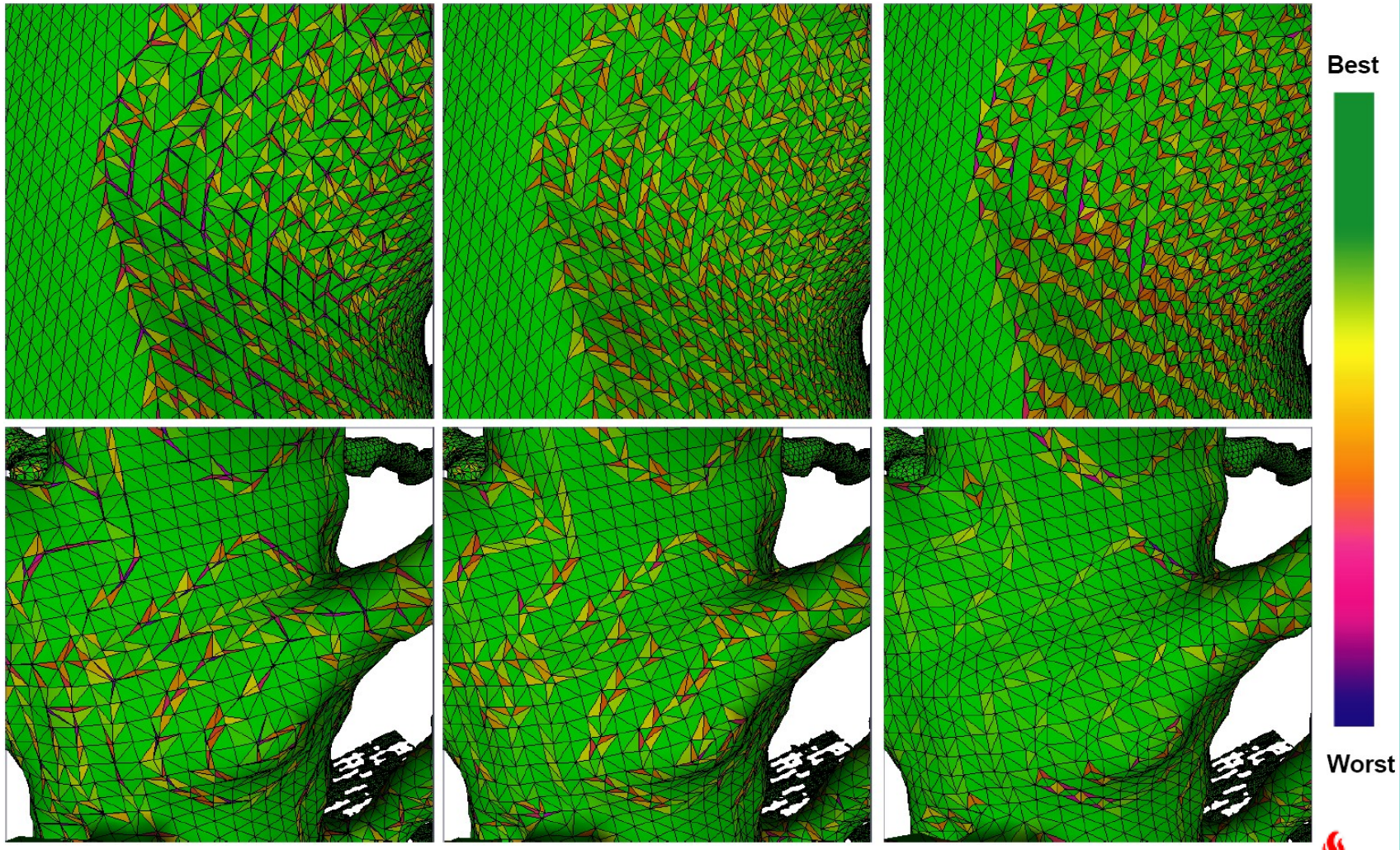
Quality Results





UFRGS

Quality Results



(a) MC (a) MACET (c) DC

UFRGS

References

- Dietrich, C. A., Comba, J. L. D., Nedel, L. P., Scheidegger, C., Schreiner, J., and Silva, C. T. (2007). Edge transformations for improving mesh quality of marching cubes. IEEE Transactions on Visualization and Computer Graphics.
- Dyken, C., Ziegler, G., Theobalt, C., and Seidel, H.-P. (2007b). Histopyramids in isosurface extraction. Technical Report MPI-I-2007-4-006, Max-Planck-Institut für Informatik.
- Ju, T., Losasso, F., Schaefer, S., and Warren, J. (2002). Dual contouring of hermite data. ACM Trans. Graph., 21(3):339–346.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 163–169, New York, NY, USA. ACM Press.

UFRGS

GPU Isosurface Extraction

Questions?

Leonardo A. Schmitz
laschmitz@inf.ufrgs.br
In cooperation with Carlos A. Dietrich
cadietrich@inf.ufrgs.br
Advised by João Luiz Dihl Comba
comba@inf.ufrgs.br

CNPq CAPES UFRGS