A Fast Method to Calculate the Volumetric Divergence Metric for Evaluating the Accuracy of the Extracted Isosurface

Cuilan Wang

Georgia Gwinnett College

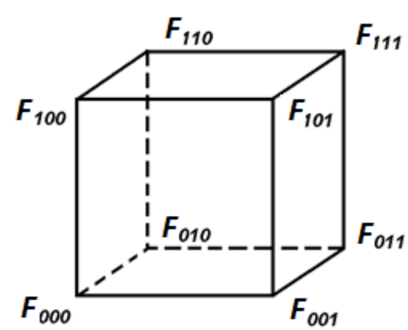
High-Performance Graphics 2018

August 10-12, Vancouver, BC

Isosurface Extraction

- Usually: 3D scalar rectilinear volumes
- Trilinear interpolation
  - Commonly used to model the interior of each dataset cell especially when the underlying data function is unknown

$$
\begin{aligned}
F(x,y,z) = {} & F_{000}(1-x)(1-y)(1-z) \\
& + F_{100}x(1-y)(1-z) + F_{010}(1-x)y(1-z) \\
& + F_{110}xy(1-z) + F_{001}(1-x)(1-y)z \\
& + F_{101}x(1-y)z + F_{011}(1-x)yz + F_{111}xyz,
\end{aligned}
$$



where F_{ijk} denotes the value at cube corner, i,j,k = 0,1, and (x, y, z) are local coordinates in a cube, $0 \le x \le 1$, $0 \le y \le 1$, $0 \le z \le 1$.

Isosurface Extraction Algorithms

- Example: Marching Cubes
- Produce a triangular mesh to approximate the isosurface given by trilinear interpolation
- Accuracy
    - Closeness of the mesh and the trilinear interpolation isosurface

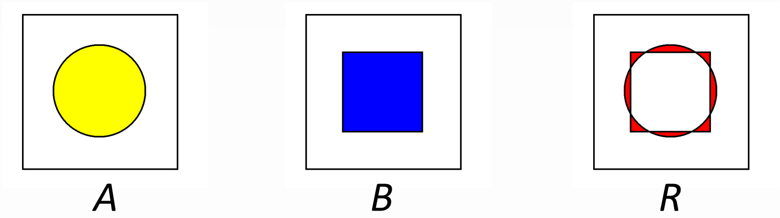
Estimating Accuracy of the Produced Mesh

- Benefits
  - Researchers: design new isosurfacing algorithms
  - The extracted mesh with high accuracy are useful:
    - Users need to substantially zoom into the data.
    - Medical applications where thin specimens or internal small cavities need to be viewed.

Volumetric Divergence (VD) Metric

- Volumetric divergence of two closed surfaces A and B:

  $V(R)$, where $R = (A_{in} \cap B_{out}) \cup (A_{out} \cap B_{in})$.

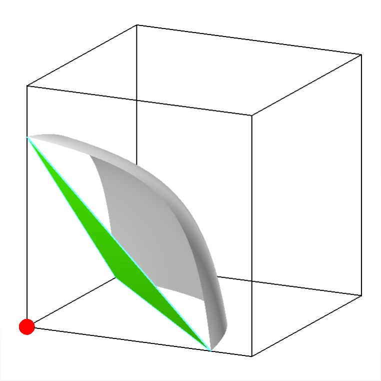


- It is a natural and accurate metric to measure the closeness between two surfaces.

- To measure the accuracy of the extracted isosurface

  A: extracted triangular mesh

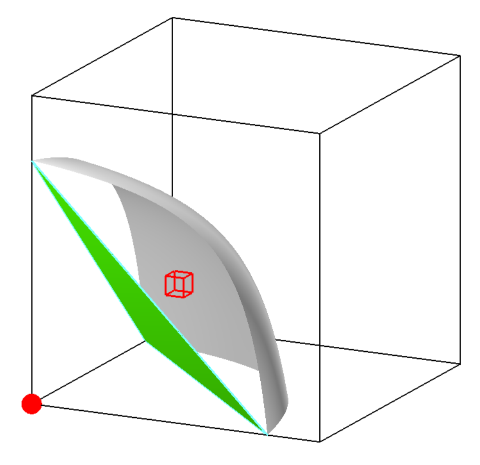
  B: trilinear interpolation isosurface



Volumetric Divergence Calculation

- The previous method (i.e., VDC method) [1] is a numerical approach
    1. Recursively subdivide a cube into 8 subcubes until a certain subdivision depth is reached or other termination criteria are satisfied.
    2. Determine each subcube's contribution to the volumetric divergence and sum the contributions.

- When the subdivision depth is high, the produced VD metric is very accurate, but the execution time is very long



Goal

- Develop a new VD calculation method faster than the previous method when both methods achieve the same level of accuracy
  - Make the VD metric used in real time to benefit isosurface extraction algorithms, such as Multi-Resolution MC [2]
- Solution
  - Combine an analytical approach and a numerical approach

Related Work

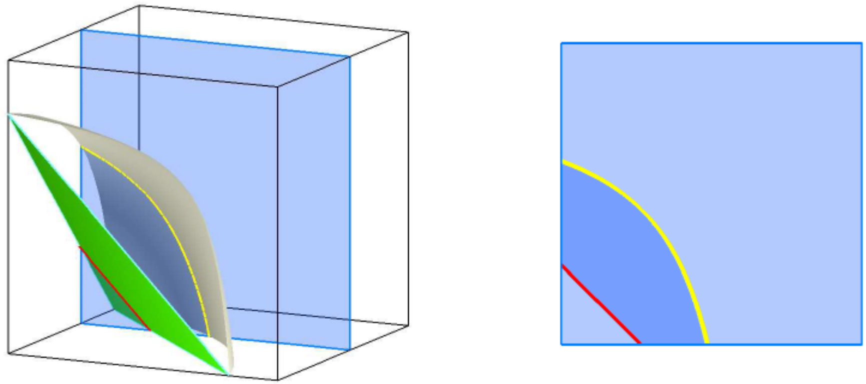
Measure Representational Accuracy

- Global geometric metrics
  - Surface area, volume inside the meshes [3]
  - Do not measure local deviation well
- Distance based metrics
  - Quadric error metric [4], Hausdorff distance [5]
    - Consider only some points (e.g., mesh vertices) of the mesh
  - Metro Scheme [6]
    - At densely sampled locations
    - Shortcomings
      - Not exhaustive
      - One-sided (i.e., not symmetric)
      - Approximate measures of mesh distance

The Slicing Method

The Overall Method

- Analytical approach
  - On each slice, the cross section of the VD region is enclosed by the isocontour hyperbola and the iso-polyline. Its area can be calculated analytically.
- Numerical approach
  - Similar to integrating areas to find volume



(a) (b)

The Overall Method

- Suppose the cube is evenly sliced along the *y*-axis. The volume of the VD region is approximated by

$$\sum_{i=1}^{n} A(y_i)\Delta d$$

, where *n* is the number of slices, $\Delta d = 1/n$, and $A(y_i)$ is the cross-sectional area of the VD region on the slice $y = y_i$.

- The more that the cube is sliced, the more accurate the result is.

Area Under Hyperbola

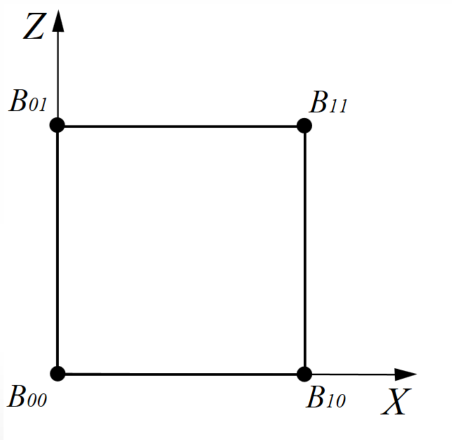
- The bilinear interpolation function over a slice is

$$B(x,z) = B_{00}(1-x)(1-z) + B_{10}x(1-z) + B_{01}(1-x)z + B_{11}xz$$

- It can be rewritten as

$$B(x,z) = axz + bx + cz + d,$$

where $a = B_{00} - B_{10} - B_{01} + B_{11}$, $b = B_{10} - B_{00}$, $c = B_{01} - B_{00}$, and $d = B_{00}$.



Area Under Hyperbola

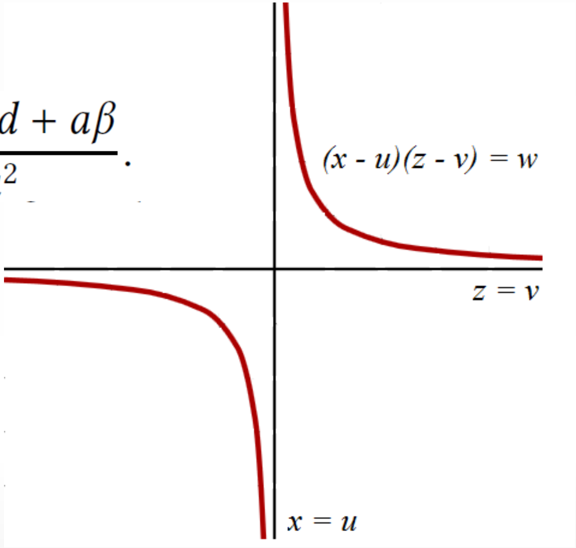
- Let β be the isovalue.
- The bilinear interpolation isocontour is defined as

$$C_B \equiv \{(x, z) : B(x, z) = \beta\},$$

- It can be rewritten as

$$(x - u)(z - v) = w,$$

$$\text{where } u = -\frac{c}{a},\ v = -\frac{b}{a},\ \text{and } w = \frac{bc - ad + a\beta}{a^2}.$$



- It can be expressed as z being a function of x :

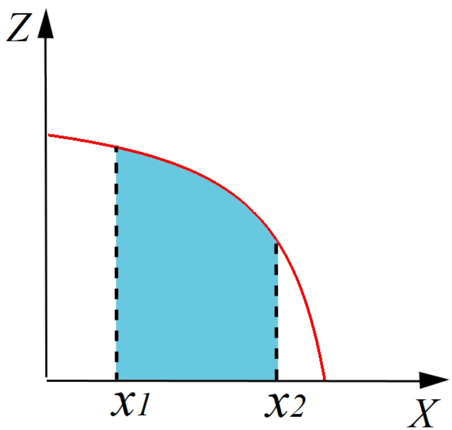
$$z = f(x) = \frac{-bx - d + \beta}{ax + c}$$

Area Under Hyperbola

- Integrating $z = f(x)$ between the limits of x_1 and x_2 to get the area under hyperbola:

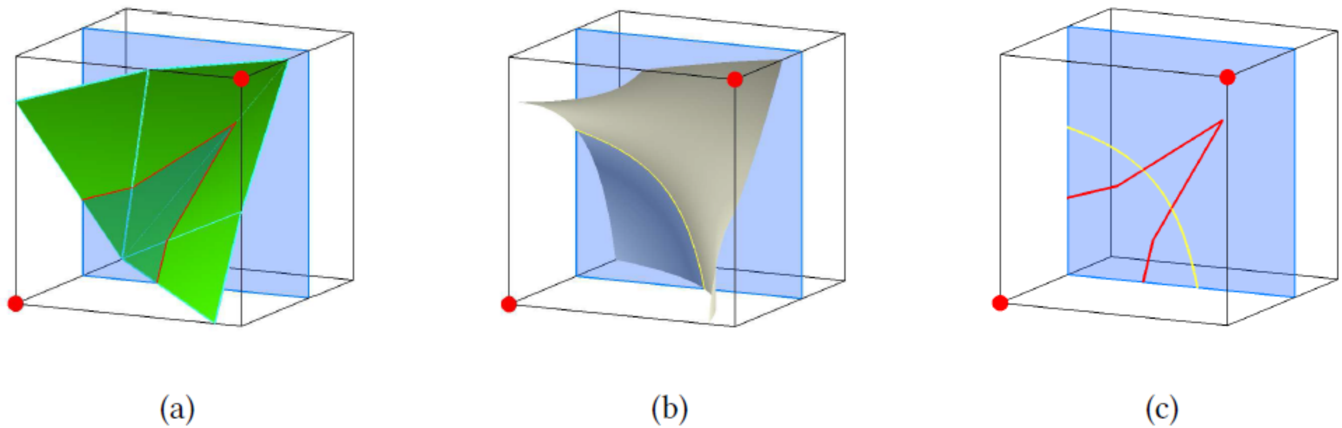
$$area = \int_{x_1}^{x_2} f(x)dx$$

$$= v(x_2 - x_1) + w \ln \left| \frac{x_2 - u}{x_1 - u} \right| .$$



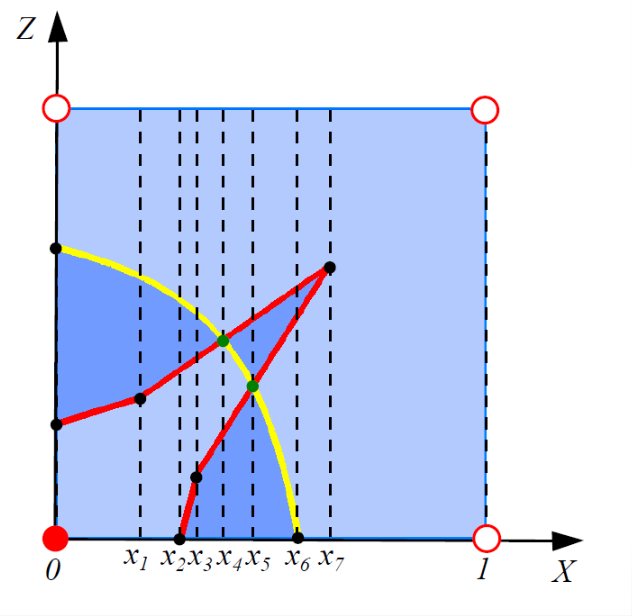
Slice Partition

- The iso-polyline may intersect with the isocontour hyperbola



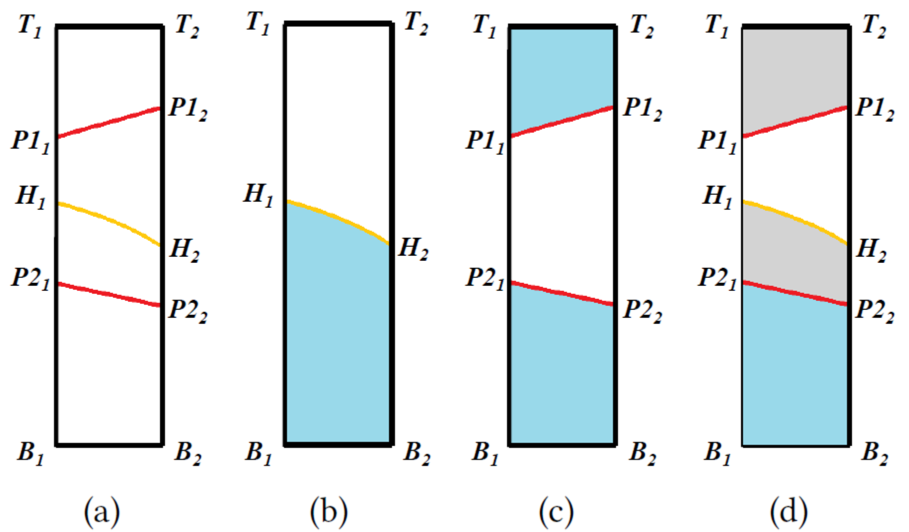
Slice Partition

- To partition, use the x values of the following points:
  - Iso-polyline vertices
  - intersection points of isocontour hyperbola and slice edges
  - Intersection points of iso-polylines and isocontour hyperbola



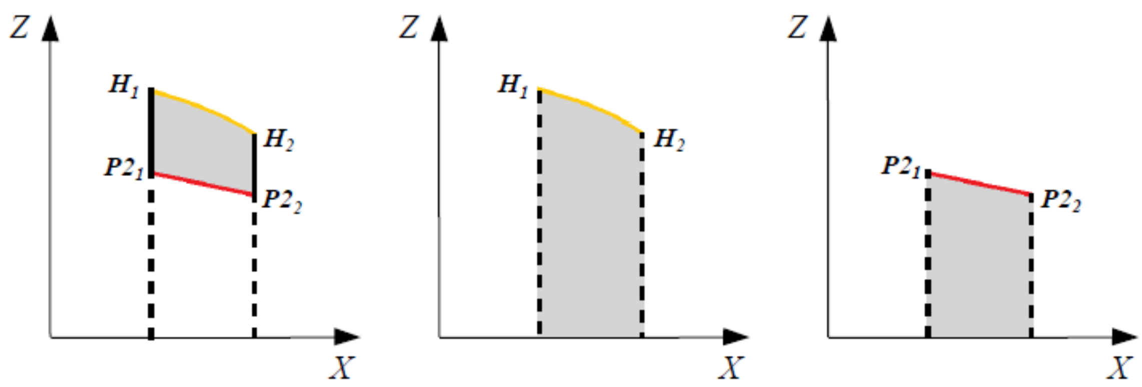
Subregion Division

- The iso-polylines and the isocontour hyperbola divide a subregion into several segments.
- The segments that are inside one type of the surface but outside the other belong to the VD region.



Calculating the Area of a Subregion Segment

- Area under the upper boundary (b) subtracting the area under the lower boundary (c) yields the subregion segment area (a).



(a) (b) (c)

Results

Experimental Settings

- The unit of the VD metric is the volume of a unit cube, which is set to 1.
- Two depths
  - VDC method – subdivision depth s
  - Slicing method – partition depth p
    - Number of slices = 2^p

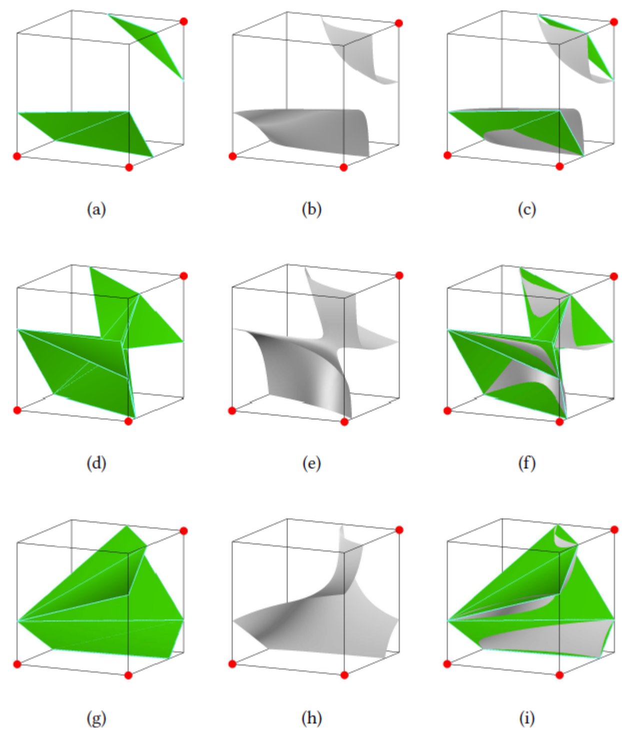
Individual Cube Instances

- Three instances of base case 6. 1. 1, 6. 1. 2, and 6. 2

6.1.1

6.1.2

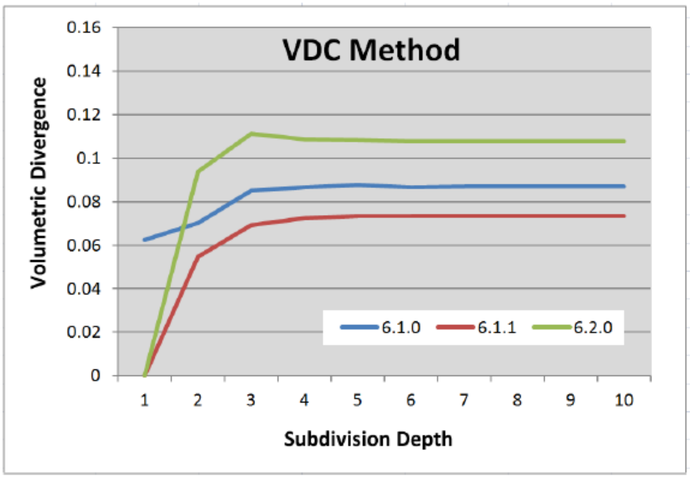
6.2



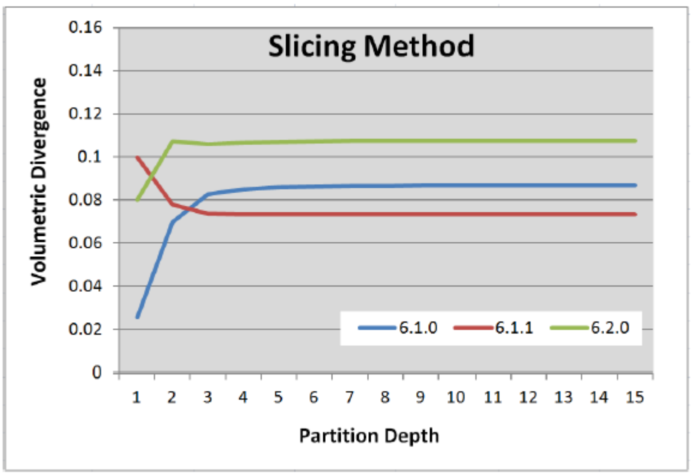
Extracted Isosurface | Tri. Inter. Isosurface | overlay

Individual Cube Instances

- Both methods converge and they converge to the same set of values



(a) VDC



(b) Slicing

Randomly Generated Instances

- Test 1000 randomly generated instances for each of three isovalues: 63, 127, and 191
- Use the VDC method with subdivision depth 10 as the baseline

Randomly Generated Instances

Partition Depth	Isovalue = 63			Isovalue = 127			Isovalue = 191		
	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)
3	0.001995	0.021705	0.003	0.003361	0.024784	0.009	0.001887	0.025700	0.003
4	0.000953	0.009922	0.006	0.001539	0.010775	0.017	0.000904	0.011719	0.008
5	0.000468	0.005099	0.012	0.000745	0.005969	0.031	0.000439	0.004845	0.013
6	0.000233	0.002569	0.022	0.000367	0.003512	0.063	0.000217	0.002340	0.023
7	0.000116	0.001295	0.047	0.000182	0.001226	0.109	0.000109	0.001203	0.047
8	0.000058	0.000647	0.078	0.000091	0.000562	0.218	0.000054	0.000610	0.093
9	0.000029	0.000323	0.171	0.000045	0.000267	0.452	0.000027	0.000305	0.172
10	0.000014	0.000162	0.344	0.000023	0.000133	0.905	0.000014	0.000153	0.358
11	0.000007	0.000081	0.670	0.000011	0.000067	1.794	0.000007	0.000080	0.734
12	0.000004	0.000041	1.326	0.000006	0.000033	3.588	0.000003	0.000057	1.466
13	0.000002	0.000021	2.652	0.000003	0.000019	7.191	0.000002	0.000045	2.932

The Slicing Method

Randomly Generated Instances

The VDC Method									
Subdivision Depth	Isovalue = 63			Isovalue = 127			Isovalue = 191		
	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)	Avg. Abs. VD Error	Max. Abs. VD Error	Time (sec.)
3	0.002245	0.025474	0.090	0.004327	0.029965	0.220	0.002336	0.021588	0.089
4	0.000602	0.007285	0.283	0.001167	0.010405	0.805	0.000649	0.013962	0.295
5	0.000156	0.002632	0.961	0.000306	0.004454	2.692	0.000174	0.005890	1.015
6	0.000039	0.000562	3.291	0.000074	0.001412	9.355	0.000042	0.002079	3.475
7	0.000010	0.000143	11.801	0.000020	0.001265	33.762	0.000010	0.000286	12.307
8	0.000002	0.000038	43.817	0.000005	0.000477	125.556	0.000003	0.000259	45.755

Randomly Generated Instances

- Map the VDC method and the Slicing method with the same level of accuracy.

Randomly Generated Instances											
Isovalue = 63				Isovalue = 127				Isovalue = 191			
VDC		**Slicing**		**VDC**		**Slicing**		**VDC**		**Slicing**	
Subd. Depth	Avg. Abs. VD Error	Partition Depth	Avg. Abs. VD Error	Subd. Depth	Avg. Abs. VD Error	Partition Depth	Avg. Abs. VD Error	Subd. Depth	Avg. Abs. VD Error	Partition Depth	Avg. Abs. VD Error
3	0.002245	3	0.001995	3	0.004327	3	0.003361	3	0.002336	3	0.001887
4	0.000602	5	0.000468	4	0.001167	5	0.000745	4	0.000649	5	0.000439
5	0.000156	7	0.000116	5	0.000306	7	0.000182	5	0.000174	7	0.000109
6	0.000039	9	0.000029	6	0.000074	9	0.000045	6	0.000042	9	0.000027
7	0.000010	11	0.000007	7	0.000020	11	0.000011	7	0.000010	11	0.000007
8	0.000002	13	0.000002	8	0.000005	13	0.000003	8	0.000003	13	0.000002

Randomly Generated Instances

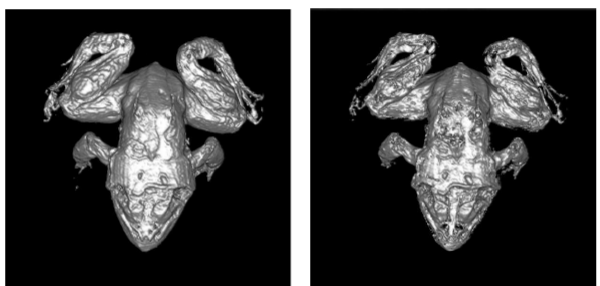
- Performance comparison

Randomly Generated Instances														
Isovalue = 63					Isovalue = 127					Isovalue = 191				
VDC		**Slicing**		Speedup	**VDC**		**Slicing**		Speedup	**VDC**		**Slicing**		Speedup
Subd. Depth	Time (sec.)	Partion Depth	Time (sec.)	Factor	Subd. Depth	Time (sec.)	Partion Depth	Time (sec.)	Factor	Subd. Depth	Time (sec.)	Partion Depth	Time (sec.)	Factor
3	0.090	3	0.003	30.0	3	0.220	3	0.009	24.4	3	0.089	3	0.003	29.7
4	0.283	5	0.012	23.6	4	0.805	5	0.031	26.0	4	0.295	5	0.013	22.7
5	0.961	7	0.047	20.4	5	2.692	7	0.109	24.7	5	1.015	7	0.047	21.6
6	3.291	9	0.171	19.2	6	9.355	9	0.452	20.7	6	3.475	9	0.172	20.2
7	11.801	11	0.670	17.6	7	33.762	11	1.794	18.8	7	12.307	11	0.734	16.8
8	43.817	13	2.652	16.5	8	125.556	13	7.191	17.5	8	45.755	13	2.932	15.6

Real Datasets

- Performance comparison

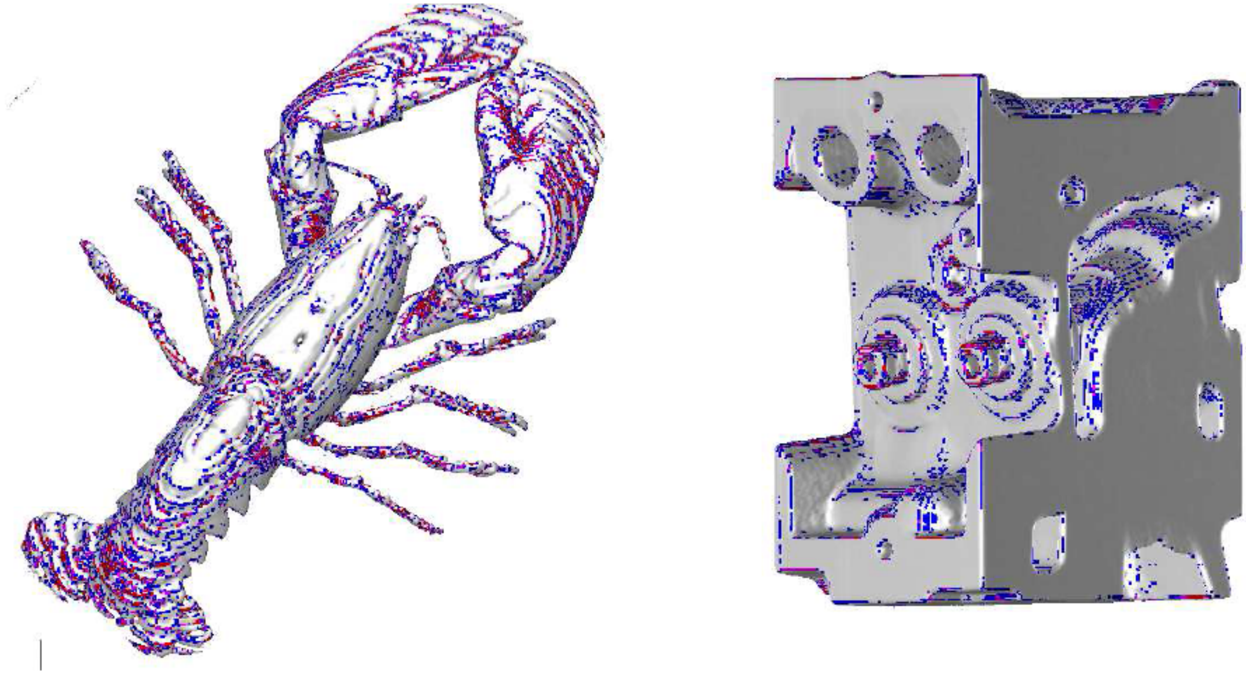
The Frog Dataset														
Isovalue = 45.5 # of active cubes = 114371, avg. VD = 0.023200 MC time = 0.20 sec.								Isovalue = 85.5 # of active cubes = 164229, avg. VD = 0.029655 MC time = 0.27 sec.						
VDC			**Slicing**			Speedup		**VDC**			**Slicing**			Speedup
Subd. Depth	Avg. Abs. VD Error	Time (sec.)	Part. Depth	Avg. Abs. VD Error	Time (sec.)	Factor		Subd. Depth	Avg. Abs. VD Error	Time (sec.)	Part. Depth	Avg. Abs. VD Error	Time (sec.)	Factor
3	0.002581	10.595	2	0.002401	0.159	66.6		3	0.002768	16.868	2	0.003189	0.258	65.4
4	0.000715	36.808	4	0.000519	0.522	70.5		4	0.000756	58.346	4	0.000679	0.868	67.2
5	0.000183	124.792	6	0.000130	1.889	66.1		5	0.000191	195.977	6	0.000168	3.086	63.5
6	0.000043	430.097	8	0.000037	7.130	60.3		6	0.000045	675.075	8	0.000046	11.847	57.0



(a) isovalue = 45.5 (b) isovalue = 85.5

Real Datasets

- Rendering of two real datasets where the mesh has large error is rendered in color

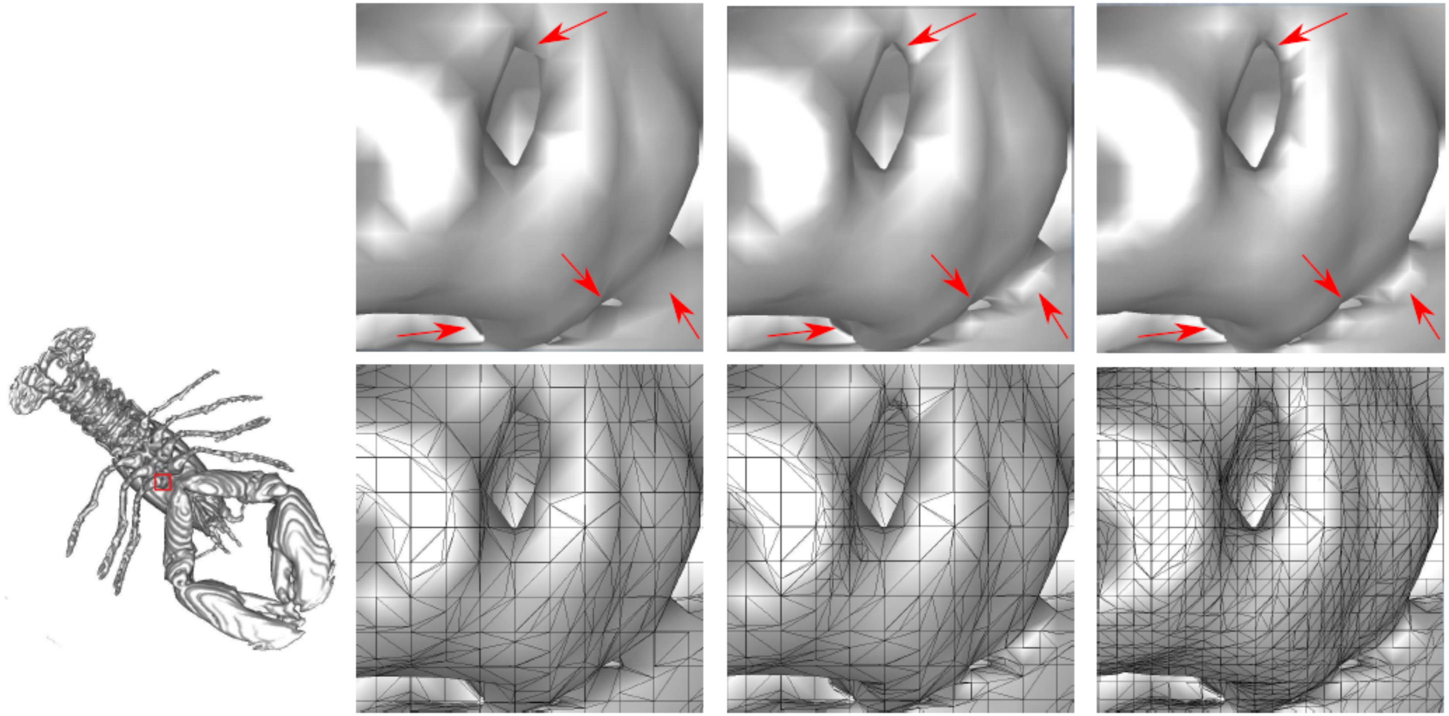


(a) Lobster (b) Engine

Application of the Slicing Method in Adaptive Isosurface Extraction

- Three methods are compared
  - Marching Cubes (MC)
  - Multi-Resolution Marching Cubes (MRMC) in [2]
    - Use the Slicing method to calculate the VD metric values
    - Mesh at the cubes where the VD metric values are larger than a threshold is refined.
  - Dividing Cubes with subdivision depth 1 (DC1)

Application of the Slicing Method in Adaptive Isosurface Extraction



(a) (b) MC (c) MRMC with $\varepsilon = 0.1$ (d) DC1

Conclusion and Future Work

Conclusion

- Both methods are correct and are capable of producing highly accurate VD metric values.
- The new method is about 15-70 times faster than the previous method.

Potential Future Work

- Modify the method to compute
  - Volume between two meshes
    - Check the closeness of the two meshes
  - Volume between two isosurfaces
    - Reveal the change rate of the isosurfaces
      - May be used to guide the selection of the representative isovalues of a dataset
  - Volume between a mesh and an isosurface where the underlying data function is known, but not trilinear interpolation isosurface

Questions?

Thank you!

References

- [1] C. Wang, T. Newman, J. Lee: On accuracy of marching isosurfacing methods, *In Proceedings of the Eurographics/IEEE VGTC Workshop on Volume Graphics '08*, Los Angeles, CA, USA, pp. 49-56, 2008
- [2] C. Wang and S. Lai, Adaptive Isosurface Reconstruction Using a Volumetric-Divergence-Based Metric, *In Proceedings of 12th International Symposium on Visual Computing (ISVC '16)*, Las Vegas, NV, USA, pp. 367–378, 2016.
- [3] G. Treece, R. Prager, A. Gee, Regularised Marching Tetrahedra: Improved Iso-surface Extraction, *Computers & Graphics*, Vol. 23, pp. 583-598, 1999.
- [4] M. Garland and P. Heckbert, Surface Simplification Using Quadric Error Metrics. *Computer Graphics (SIGGRAPH '97 Proceedings)*, vol. 31, pp. 209-216, 1997.
- [5] R. Klein, G. Liebich, and W. Straber, Mesh Reduction with Error Control, *Proceedings of Visualization '96*, San Francisco, pp. 311-318, 1996.
- [6] P. Cignoni, C. Rocchini, R. Scopigno, Metro: Measuring error on simplified surfaces. *Comp. Graphics Forum 17*, Vol. 2, pp. 167–174, 1998