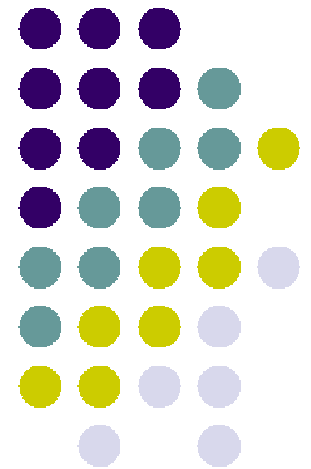


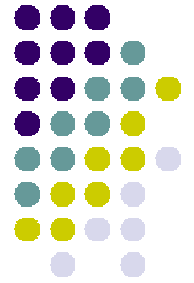
ME 290-R:
General Purpose Computation
(CAD/CAM/CAE) on the GPU
(a.k.a. Topics in Manufacturing)

Sara McMains
Spring 2009
Lecture 20

DiFi: Fast 3D Distance Field Computation Using Graphics Hardware

A.Sud, M. Otaduy and
D. Manocha





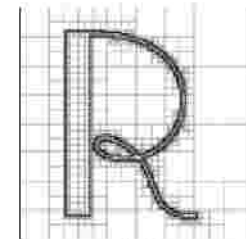
Distance Fields

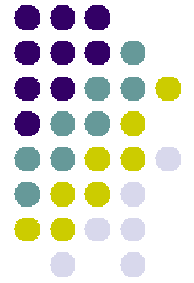
| Definition:

- | Smallest distance at each point (of a grid) to a set of objects

| Applications:

- | Collision detection, proximity queries
- | Model simplification
- | Geometric modeling (sculpting, CSG operations)
- | Medial Axis Transforms

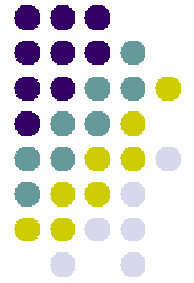




Main Contributions

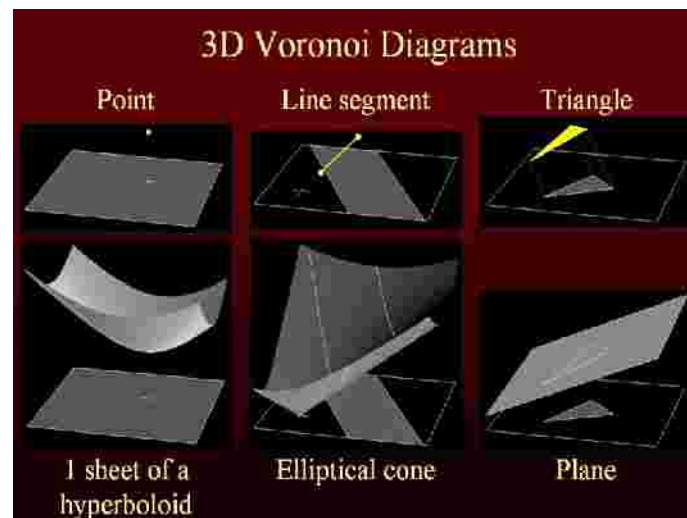
- | Fast algorithm to compute uniformly sampled distance fields.
 - | With novel primitive culling
 - | With distance function clamping

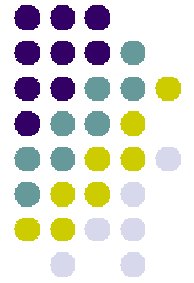
- | Huge improvement of the previous algorithm based on Voronoi Diagram approach.



Related Work

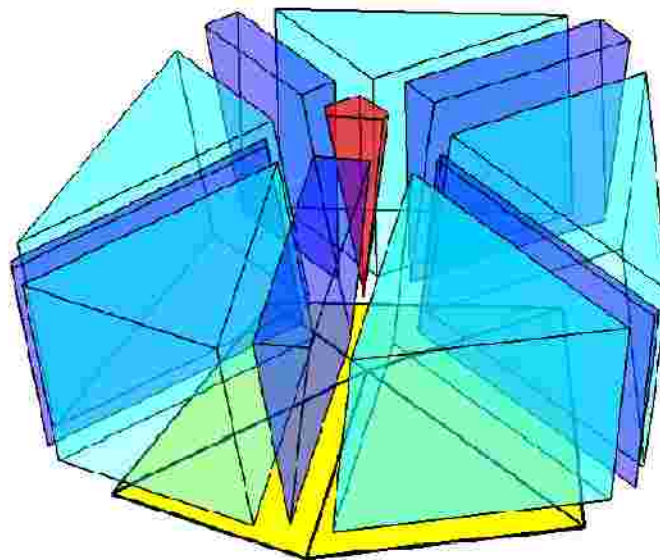
- | Hoff et al.
 - | 3D VD using GPU
 - | Graphics hardware can render polygonal approximations of distance function (distance mesh) for each primitive to depth buffer (one slice) (example: [hyperboloid for point](#))
 - | Sweep along z-axis





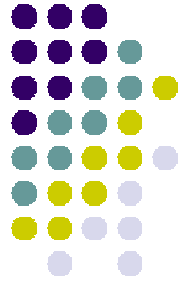
Related Work

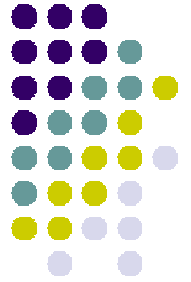
- | CSC (Characteristic Scan Conversion) by Mauch
 - | Scan convert polyhedra that are supersets of Voronoi cells for each primitive (in voxel space)



Related Work

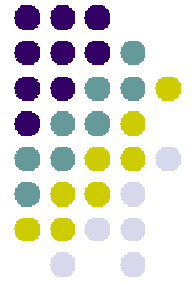
- | Volumetric (by Sethian, Maurer et al.)
 - | Wave propagation





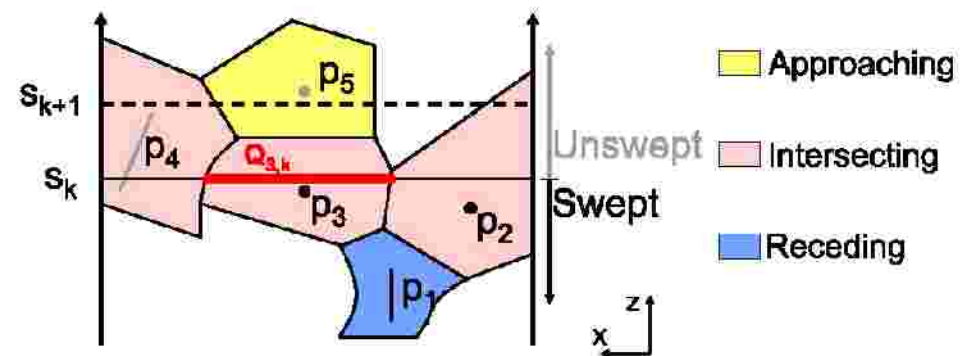
Authors' approach

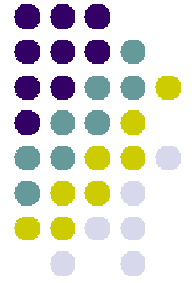
- | Speed up the distance field computation using the following properties
 - | Monotonicity of Distance Functions:
 - | Connectivity of VD regions:
 - | Spatial Coherence:



Region (Cell) Classification

- | By the z-max and z-min of VD regions
 - | Intersecting, I_k^+
 - | Receding R_k^+
 - | Approaching A_k^+
- | However, VD computation is expensive. Instead of exact VD, they used...





Intersecting Swept Sites

S_k^+ : swept

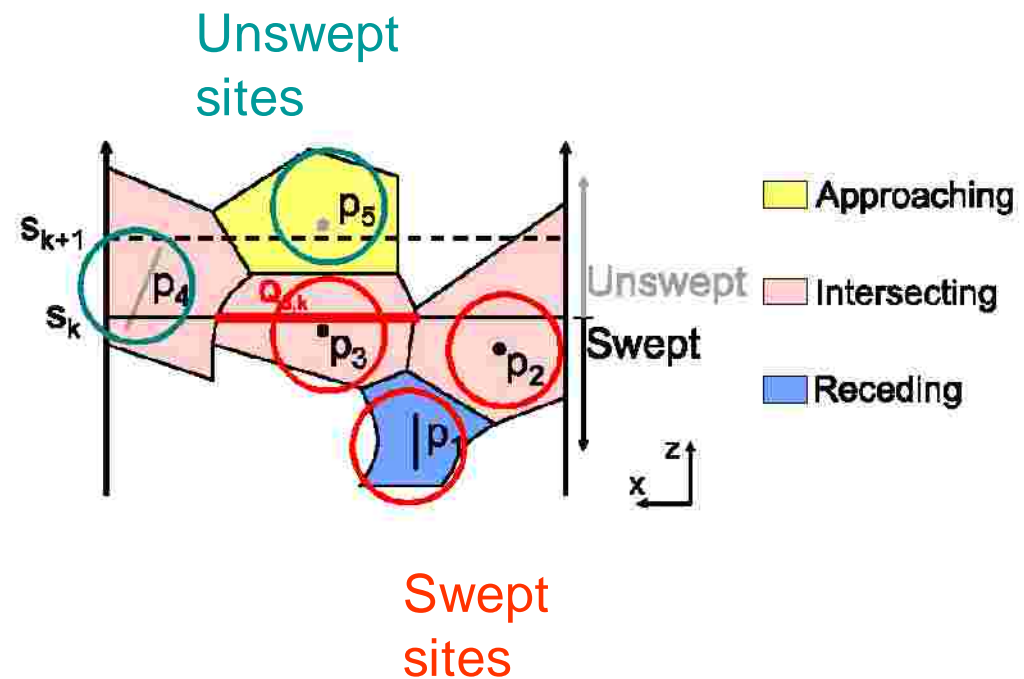
U_k^+ : unswept

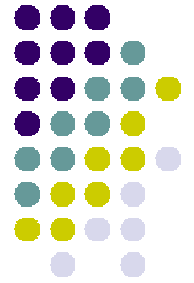
$$I_k^+ = IS_k^+ \cup IU_k^+$$

$$IU_k^+ = IS_k^-$$

$$I_k^+ = IS_k^+ \cup IS_k^-$$

$$D_k(I_k) = D(IS_k^+ \cup IS_k^-) = \min(D_k(IS_k^+), D_k(IS_k^-))$$

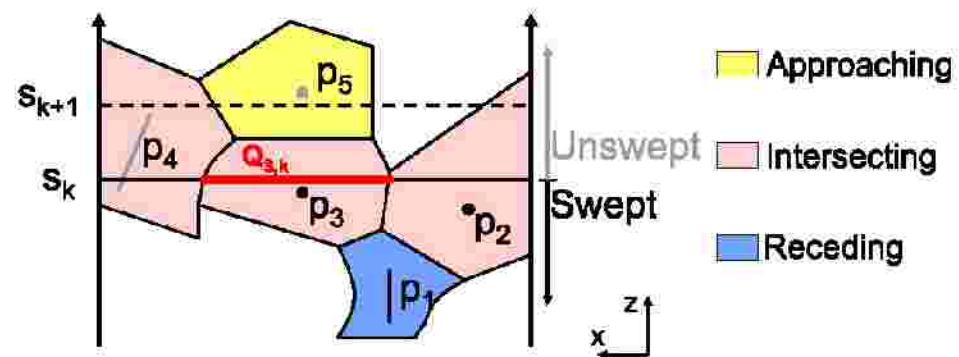


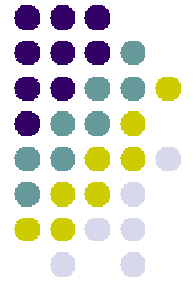


Spatial Coherence

- | Most of sites in IS_k^+ are in IS_{k+1}^+
- | $IS_{k+1}^+ = IS_k^+ \cup (S_{k+1} - S_k) - (R_{k+1} - R_k)$
- | $IS_k^+ (= I_k^+ \cap S_k^+)$ is still hard to compute
- | Instead, use “potentially intersecting swept sites”

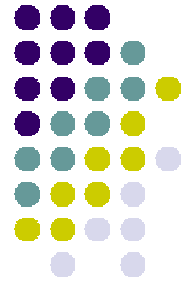
$$\hat{IS}_k \supseteq IS_k$$





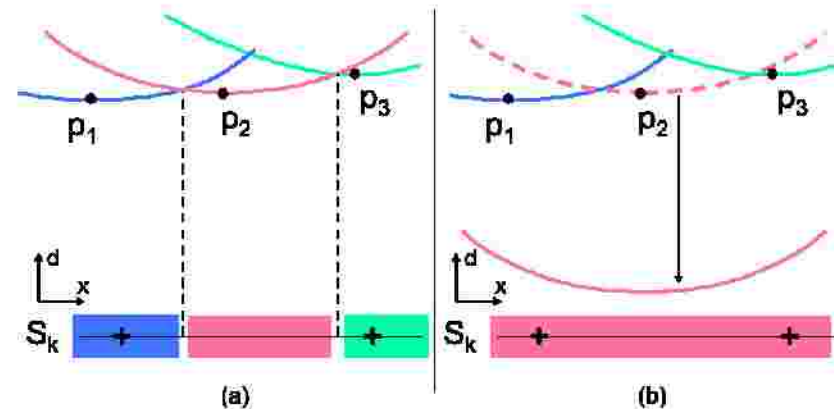
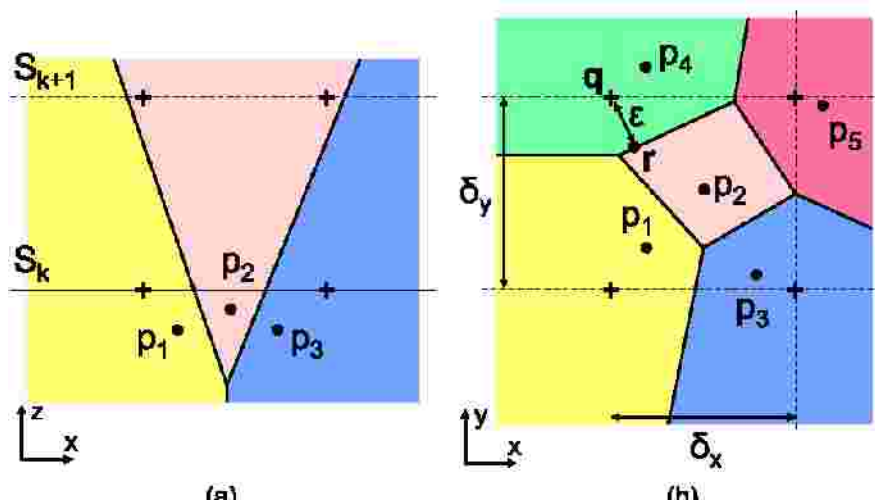
Main Algorithm

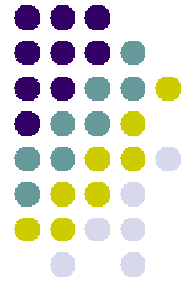
- | 1. Initialize $I\hat{S}_{k+1}^+ = I\hat{S}_k^+, D_{k+1}^+ = \infty$
- | 2. Update (Add fresh swept sites, $S_{k+1}^+ - S_k^+$)
- | 3. Compute distance at slice (k+1)
 - | Use depth buffer
- | 4. Compute receding sites $R_{k+1}^+ - R_k^+$
 - | Check whether each site contributes to D_{k+1}^+
- | 5. Update $I\hat{S}_{k+1}^+$
 - | Remove receding sites that don't contribute to min distances
 - | If a site doesn't contribute, remove it from $I\hat{S}_{k+1}^+$



Sampling errors

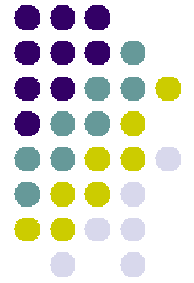
- To ensure a site gets at least one pixel, before removing from $I\hat{S}_{k+1}^+$ translate $D_k(p_i)$, then update distance fields.





Main Algorithm (modified)

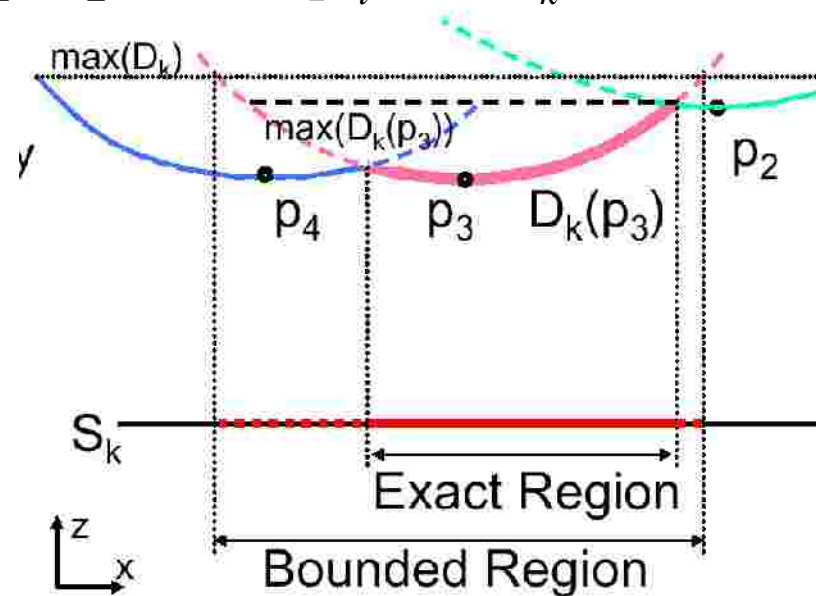
- | 1. Initialize $I\hat{S}_{k+1}^+ = I\hat{S}_k^+, D_{k+1}^+ = \infty$
- | 2. Update (Add fresh swept sites, $S_{k+1}^+ - S_k^+$)
- | 3. Compute distance at slice (k+1)
 - | Use depth buffer
- | 4. Compute receding sites $R_{k+1}^+ - R_k^+$
 - | Check whether each site contributes to D_{k+1}^+
- | 5. Update $I\hat{S}_{k+1}^+$
 - | (Remove receding sites that don't contribute to min distances)
 - | If a site doesn't contribute, reduce distance by d
 - | If still not contributing, remove it from $I\hat{S}_{k+1}^+$

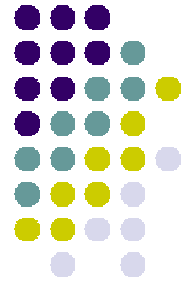


Distance Function Clamping

- | Rasterization is expensive.
 - | For a site p_i , rather than draw the distance function to the entire depth buffer, only draw on

$$Q_{i,k} = \{q \mid q \in V(p_i) \mid s_k\}$$





Distance Function Clamping

| Rasterization is expensive.

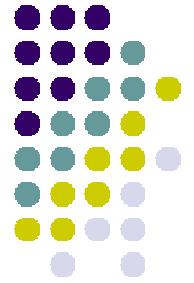
However, exact $Q_{i,k}$ is impractical. Therefore, superset

$$\hat{Q}_{i,k} = \{q \mid \text{dist}(q, p_i) \leq \max D_k\}$$

| Then, how do we know $\max D_k$?

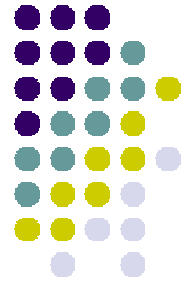
$$\max D_{k+1} \leq \max D_k + d_z$$

Main Algorithm (final version)

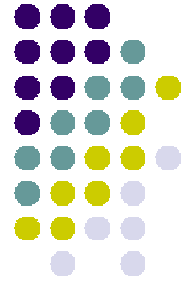


- | 1. Initialize $I\hat{S}_{k+1}^+ = I\hat{S}_k^+, D_{k+1}^+ = \infty, \max D_{k+1}^+ = \max D_k^+ + d_z$
- | 2. Update (Add fresh swept sites, $S_{k+1}^+ - S_k^+$)
- | 3. Compute distance at slice (k+1)
 - | Compute $\hat{Q}_{i,k+1}$ using $\max D_{k+1}$
 - | Draw distance function of p_i in $\hat{Q}_{i,k+1}$
- | 4. Compute receding sites $R_{k+1} - R_k$
 - | Check whether each site contributes to D_{k+1}
- | 5. Update
 - | (Remove receding sites that don't contribute to min distances)
 - | If a site doesn't contribute, reduce distance by d
 - | If still not contributing, remove it from $I\hat{S}_{k+1}^+$

Results

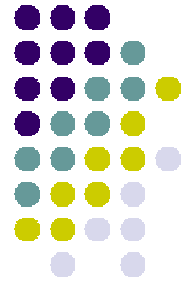


Model	Polys	Resolution	CSC	HAVOC	HAVOC+CSC	DiFi
Rotor	4736	4x128x128	59.22	6.33	3.98	0.61
Rotor	4736	8x254x254	424.89	18.73	12.12	1.16
Triceratops	5660	128x56x42	127.81	2.11	1.10	0.79
Triceratops	5660	254x111x84	990.48	6.33	3.65	1.92
Hugo	17000	73x45x128	X	30.55	19.24	4.22
Hugo	17000	145x90x254	X	108.84	75.85	8.63
Head	21764	78x105x128	201.12	37.89	16.76	4.98
Shell	22598	254x252x252	X	162.97	95.31	7.79
Cassini	93234	186x254x188	X	356.03	298.55	29.86
Dragon	108926	57x90x128	X	171.13	95.69	24.76



Limitations

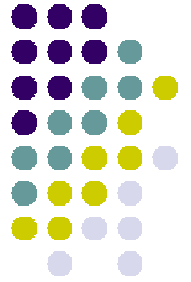
- | 4K*4K pixel resolution
- | The algorithm is bounded by the cost of rasterization and fill rate
- | Not applicable for adaptive grids



Summary

- | Fast computation of distance fields by GPU
 - | Incremental 2-pass approach
 - | Optimize at bottleneck: rasterization and z-test
 - | Site culling
 - | VD is not computed
 - | Distance functions for all m primitives not computed at each of n slices
 - | Distance function clamping
 - | Further reduces rasterization cost
 - | Minimizes sampling error using offsets
 - | Calculates supersets
 - | Ensures all sites tested by at least one pixel

Acknowledgements



- | Young Shon put together original version of these slides