

First International Workshop on Implicit Surfaces

**Adaptive enumeration of implicit surfaces
with
affine arithmetic**

LUIZ HENRIQUE DE FIGUEIREDO
JORGE STOLFI

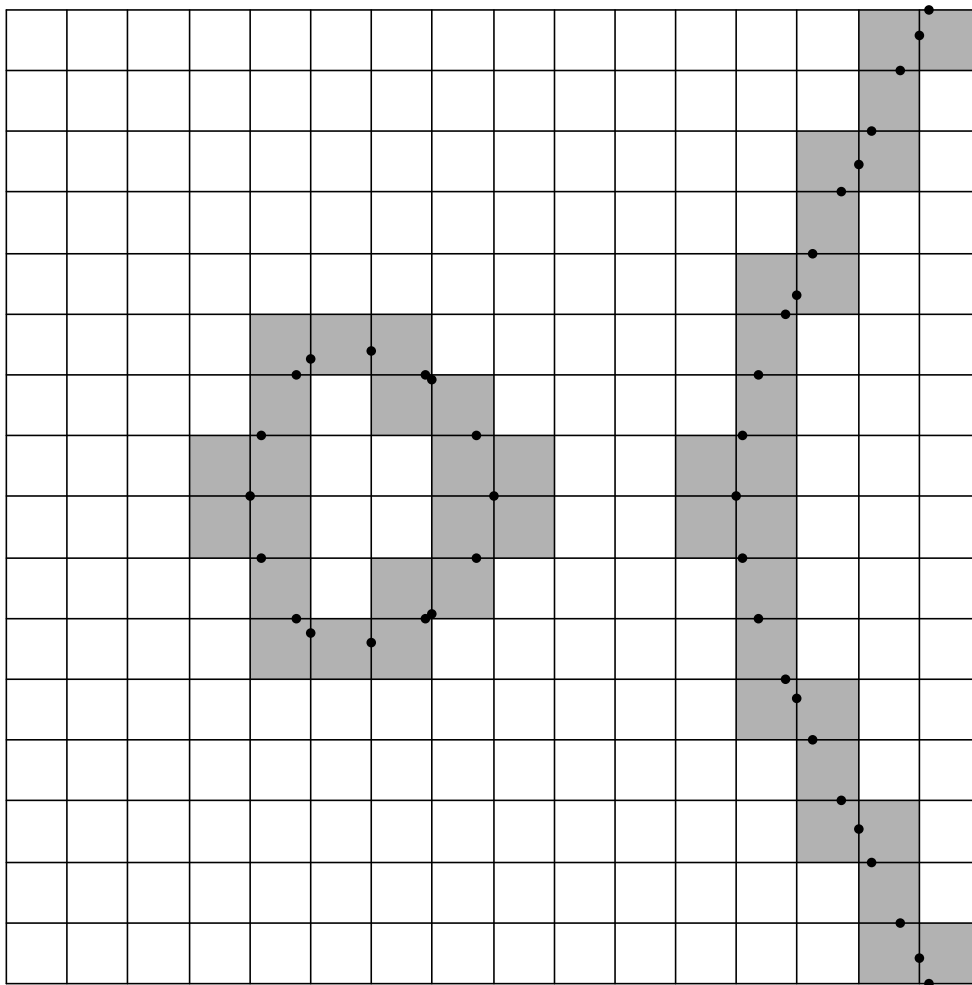
Computer Graphics Technology Group (TeC_{Graf})
Computer Science Department, PUC-Rio, Brazil

Computer Science Department
State University of Campinas (UNICAMP), Brazil

April 1995

Enumeration of implicit surfaces

- simple and general algorithm for approximating surface S given by $h(x, y, z) = 0$ in domain Ω :
 - ▷ decompose Ω into small cells
 - ▷ identify which cells intersect S
 - ▷ approximate S within each intersecting cell

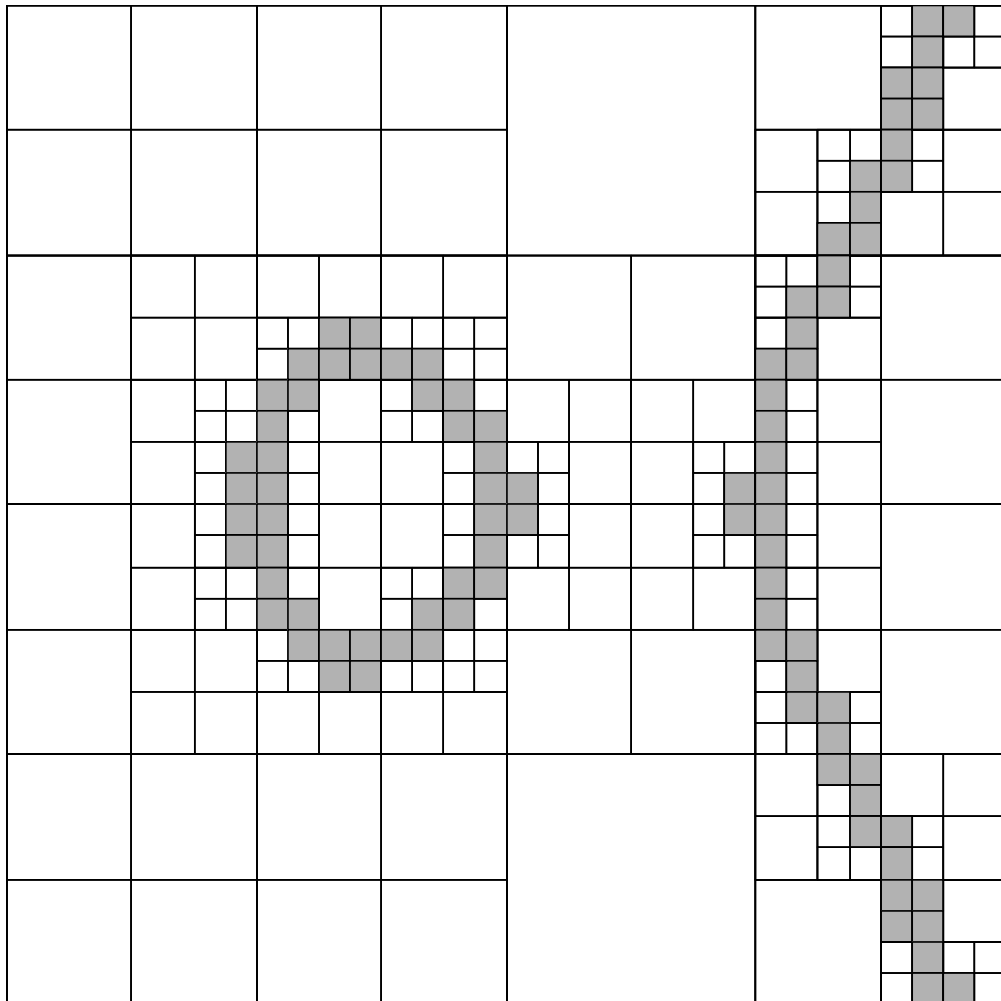


Problems with full enumeration

- cost is enumeration of intersecting cells
 - ▷ sampling (vulnerable to aliasing)
- simplest decomposition is regular
 - ▷ n^3 cubes in total
 - ▷ only $O(n^2)$ cubes intersect S
 - ▷ how to avoid scanning “useless” cells?
- efficient enumeration based on sampling
 - ▷ continuation methods
 - ▷ adaptive refinement
 - ▷ need one cell in each connected component

Hierarchical decomposition methods

- rely on test to prove absence of S
- recursive exploration of Ω
- range analysis
 - ▷ interval arithmetic



Interval arithmetic

- quantities represented by intervals of floating-point numbers
- computed intervals contain results
 - ▷ too conservative
 - ▷ error explosion
- assumes operands are independent

$$\bar{x} = [4 .. 6]$$

$$10 - \bar{x} = [10 .. 10] - [4 .. 6] = [4 .. 6]$$

$$\bar{x}(10 - \bar{x}) = [4 .. 6] \cdot [4 .. 6] = [16 .. 36]$$

$$\text{true range} = [24 .. 25]$$

Affine arithmetic

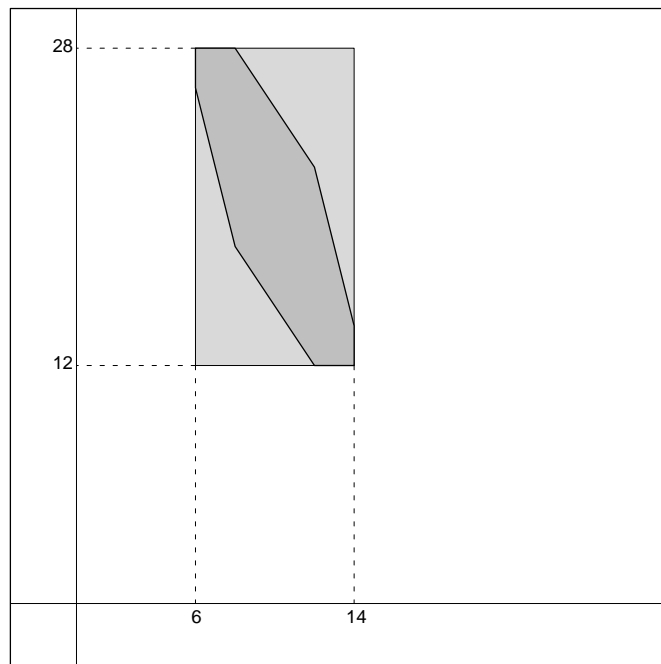
- quantities represented by affine forms

$$\hat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \cdots + x_n\varepsilon_n$$

- simple conversion IA \leftrightarrow AA
- more complex and expensive than IA
- higher accuracy worth extra cost
- keeps track of *correlations* between quantities

$$\hat{x} = 10 + 2\varepsilon_1 + 1\varepsilon_2 - 1\varepsilon_4$$

$$\hat{y} = 20 - 3\varepsilon_1 + 1\varepsilon_3 + 4\varepsilon_4$$



Computing with affine arithmetic

$$z \leftarrow f(x, y) \quad \hat{z} \leftarrow \hat{f}(\hat{x}, \hat{y})$$

- simple for affine functions

$$\hat{x} \pm \hat{y} = (x_0 \pm y_0) + (x_1 \pm y_1)\varepsilon_1 + \cdots + (x_n \pm y_n)\varepsilon_n$$

$$\alpha \hat{x} = (\alpha x_0) + (\alpha x_1)\varepsilon_1 + \cdots + (\alpha x_n)\varepsilon_n$$

$$\hat{x} \pm \alpha = (x_0 \pm \alpha) + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n$$

- in general, use best (Chebyshev) affine approximation; append extra term to represent error

$$\hat{z} = z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n + z_k\varepsilon_k$$

- example revisited

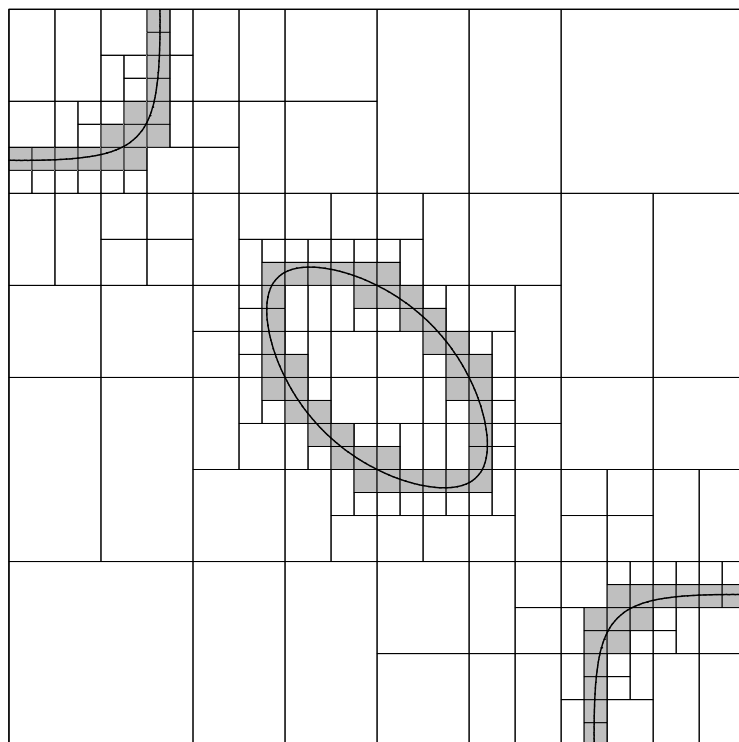
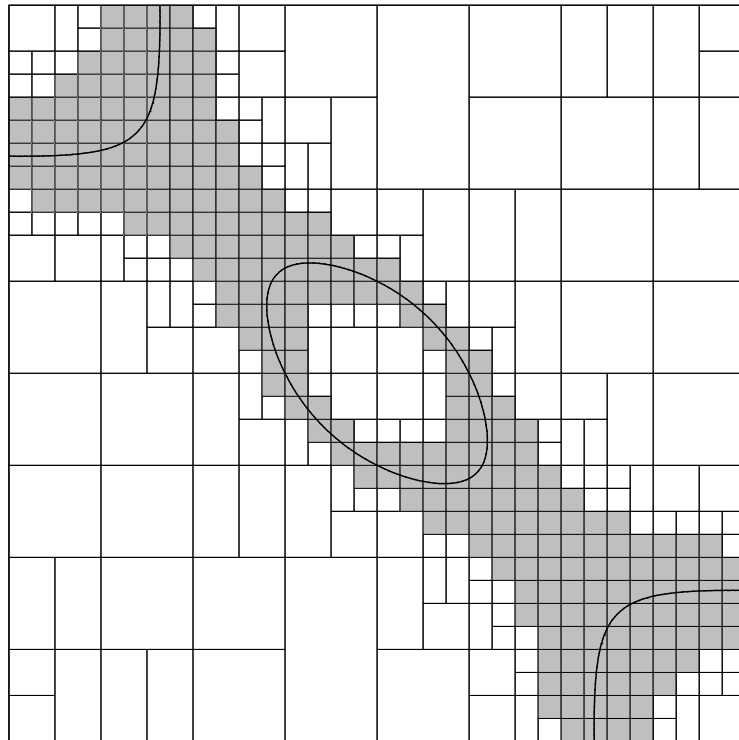
$$\hat{x} = 5 + 1\varepsilon_1$$

$$10 - \hat{x} = 5 - 1\varepsilon_1$$

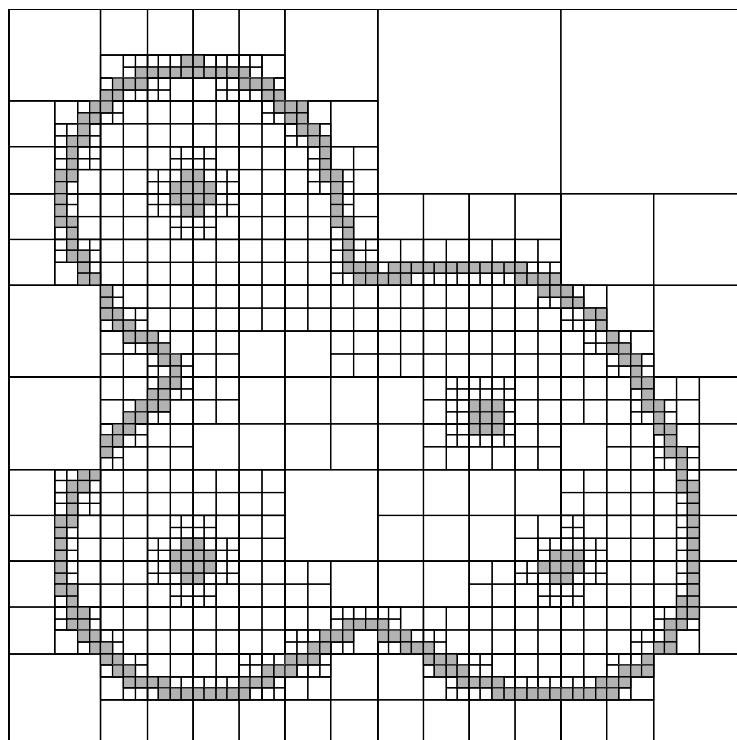
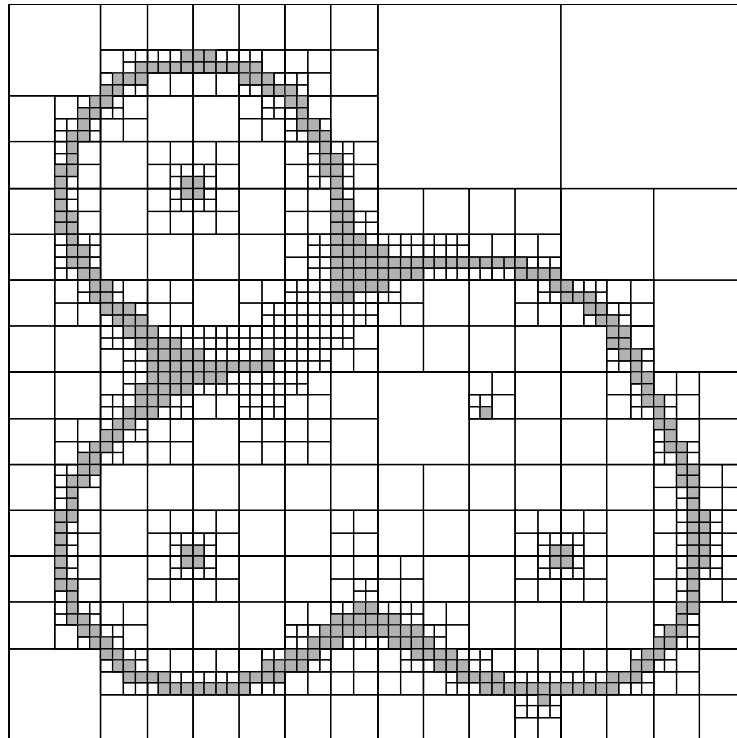
$$\hat{z} = \hat{x}(10 - \hat{x}) = 25 + 0\varepsilon_1 - 1\varepsilon_2.$$

$$[\hat{z}] = [25 - 1 \text{ .. } 25 + 1] = [24 \text{ .. } 26]$$

Example: $h(x, y) = x^2 + y^2 + xy - (xy)^2/2 - 1/4$



Example: quadratic electrostatic potential



Example: quartic in 3d

	level	evaluations	leaves	time
IA	1	9	8	0.0
	2	73	62	0.0
	3	569	334	0.0
	4	3241	1276	0.1
	5	13449	5122	0.2
	6	54425	20580	0.4
	7	219065	82502	1.5
	8	879081	329746	5.7
	9	3517049	1318058	22.5
	10	14061513	5275400	89.5

	level	evaluations	leaves	time
AA	1	9	8	0.0
	2	73	56	0.0
	3	521	190	0.1
	4	2041	710	0.2
	5	7721	2664	0.5
	6	29033	10104	1.7
	7	109865	39960	6.4
	8	429545	158282	24.8
	9	1695801	630380	97.8
	10	6738841	2516356	388.5

Rendering

- can render directly
 - ▷ no need to build octree, just traverse it
- simplified setting
 - ▷ Ω aligned rectangular box
 - ▷ orthogonal projection onto horizontal plane
 - ▷ image resolution $2^L \times 2^L$
- enumeration proceeds down octree to level L
 - ▷ leaves correspond to pixels
 - ▷ need normals for shading
- painter's algorithm to remove hidden parts
 - ▷ traverse octree from bottom to top (in z)

Rendering

- need to compute mean normal to assign color to pixels
- estimates of gradient for full enumeration
 - ▷ central differences
- estimates of gradient with interval arithmetic
 - ▷ formal differentiation
 - ▷ Bauer–Strassen differentiation
- estimates of gradient with affine arithmetic

$$\hat{x} = x_0 + x_1\varepsilon_1$$

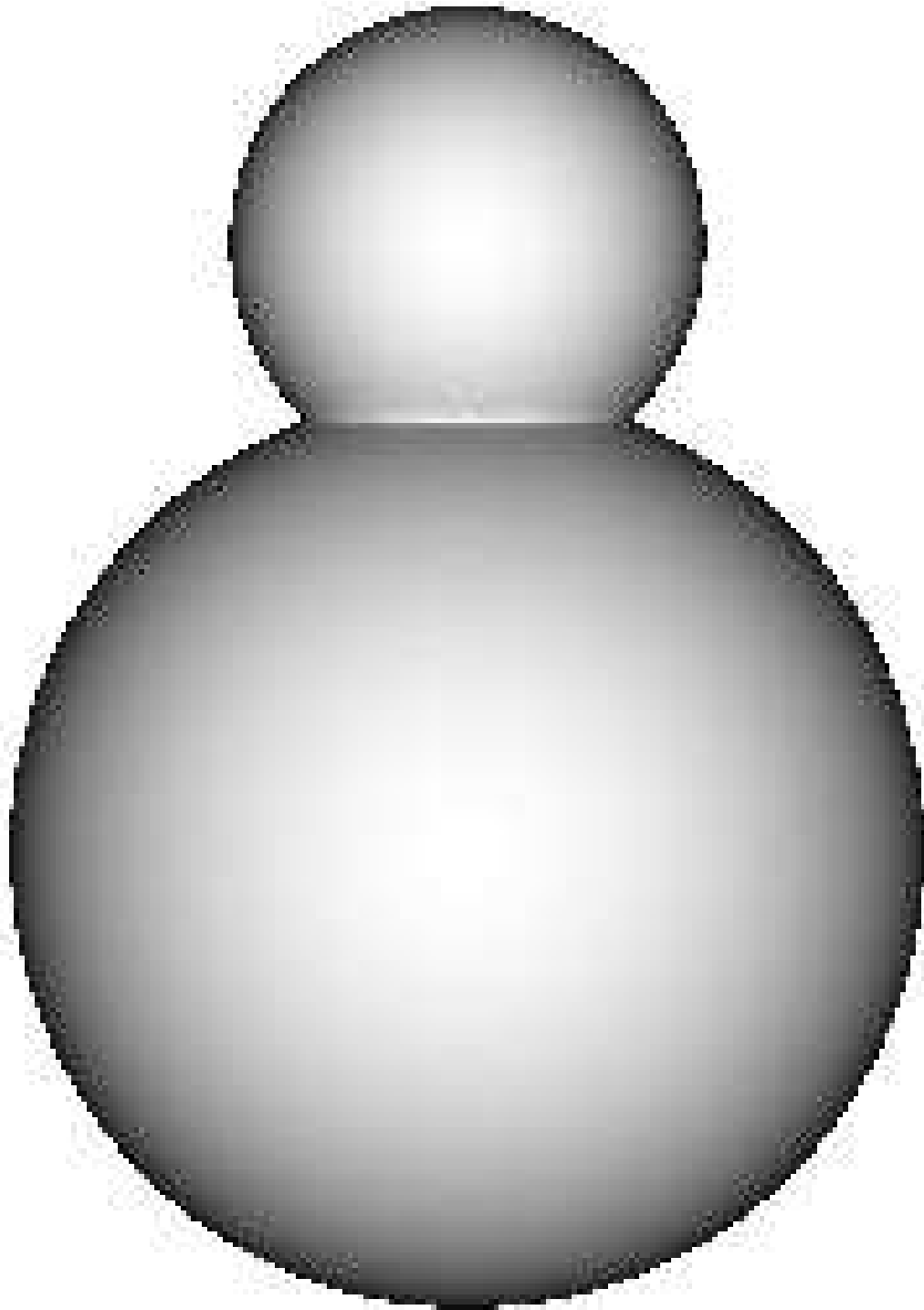
$$\hat{y} = y_0 + y_2\varepsilon_2$$

$$\hat{z} = z_0 + z_3\varepsilon_3$$

$$\hat{h} = h_0 + h_1\varepsilon_1 + h_2\varepsilon_2 + h_3\varepsilon_3 + \dots$$

$$\nabla h \approx (h_1/x_1, h_2/y_2, h_3/z_3)$$

Rendering example: two spheres



Other applications

- intersection of implicit surfaces
 - ▷ $h = f^2 + g^2$
 - ▷ not good for sampling
 - ▷ ok for range analysis
 - ▷ simultaneous enumeration

- ray tracing
 - ▷ binary search finds first intersection
 - ▷ IA evaluates h on bounding box of ray
 - ▷ AA exploits correlations

Conclusion

- AA more accurate than IA
- AA more complex and expensive than IA
- higher accuracy worth extra cost
 - ▷ smaller models
- relative speeds depend on expression
 - ▷ AA now typically 4–5 times slower than IA
- future work
 - ▷ improve performance of AA
 - ▷ use AA to estimate surface curvature