Interactive Implicit Modeling with Hierarchical Spatial Caching

Ryan Schmidt[1], Brian Wyvill[1], Eric Galin[2]

[1]University of Calgary, Canada

[2]LIRIS-CNRS, Université Claude Bernard Lyon 1, France

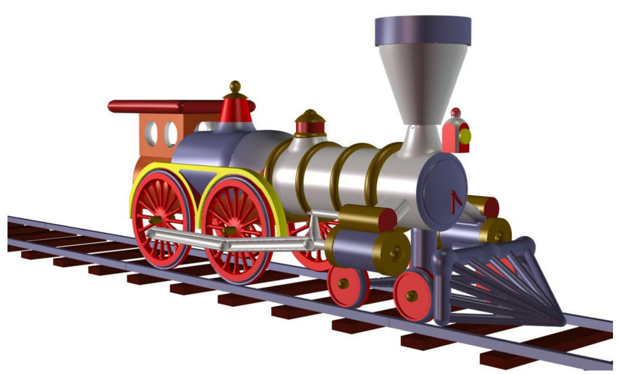
Outline

- We present a hierarchical spatial caching technique that:
  - Enables interactive visualization of BlobTree implicit volume models
  - Is suitable for use in an interactive BlobTree modeling system
- We demonstrate an order-of-magnitude improvement in polygonization time
- We explore several applications

Background

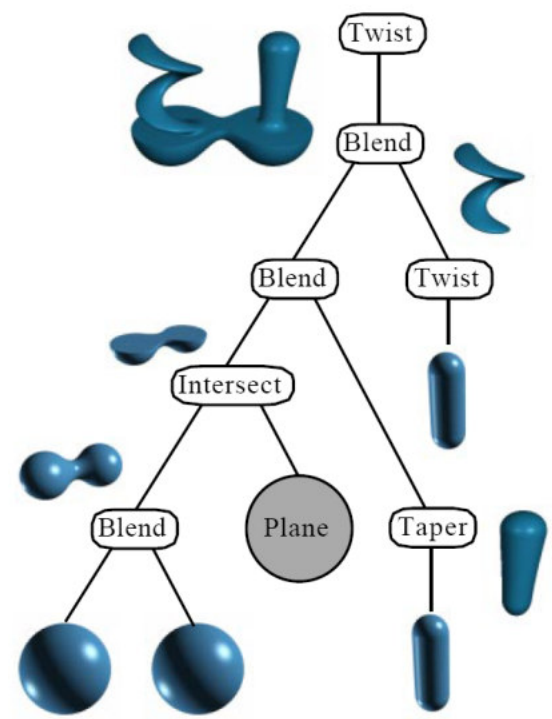
Which Implicits?

- Many, many alternatives
- We choose Hierarchical Implicit Volume Modeling (BlobTrees [Wyvill et al 99])
  - Complex user-created models have been demonstrated



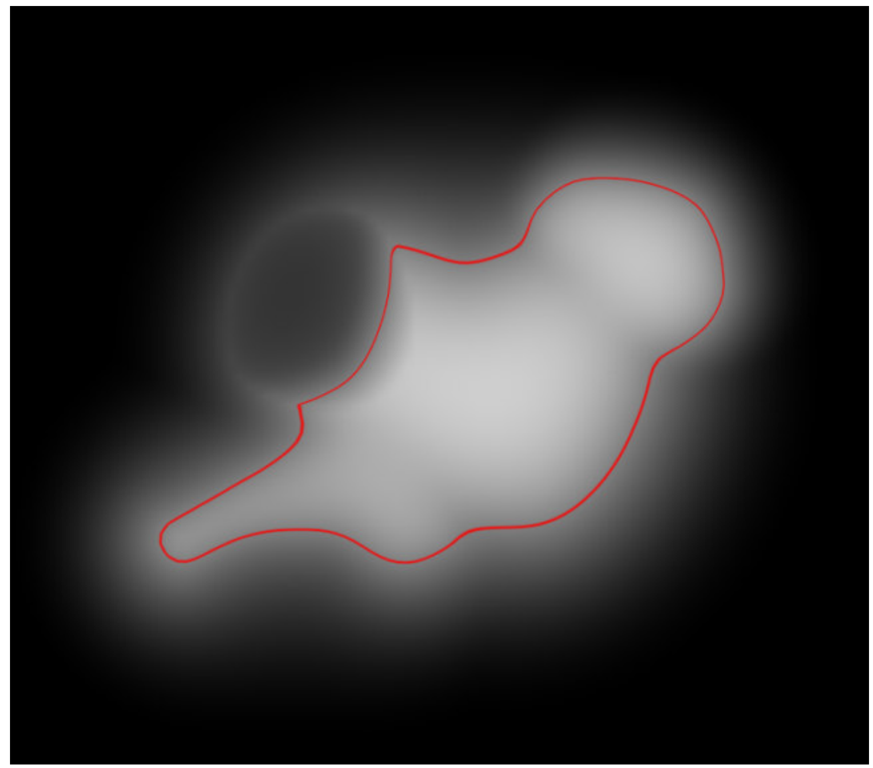

BlobTrees

- Hierarchical implicit model data structure
- Leaves are implicit volumes
  - Must have *bounded* scalar fields
- Internal nodes are c*omposition operators*
  - CSG, many types of blending, PCM, warping, texturing, etc…
- Implicit volume is procedurally defined at the root of the tree



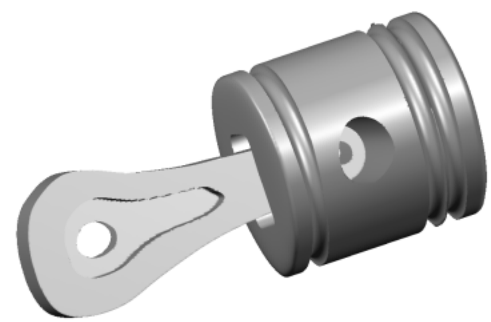
BlobTree Scalar Fields

- Volume defined as $f(\mathbf{p}) \geq v_{iso}$
- $f(\mathbf{p})$ is *bounded* at all nodes of tree
  - non-zero values are contained inside a finite bounding box
  - local influence guaranteed



Shape Modeling with BlobTrees

- Volume Modeling
  - Blending, Warping and CSG → free-form and CAD
- Functional representation
  - Sharp edges, smooth surfaces
- Scene Graph
  - Animation
- Construction History
  - Non-linear editing



Interactive Modeling with BlobTrees

- Too Slow

- Visualization is the bottleneck
  - Designers need interactive feedback (surface)
  - Visualization algorithms do not scale interactively
    - Require many potential field evaluations

- Local updates? [Jevans88]
  - Do not help for large-scale assembly, expensive primitives, high-frequency details

Fundamental Problem

- BlobTree evaluations are too expensive
  - Tree traversal cost increases with each new primitive / operator
  - Tree optimization schemes are insufficient
- Our Solution: Hierarchical Spatial Caching
  - Discretely approximate tree branches
  - Reduces the cost of **all** field evaluations
  - Reduces tree depth

Hierarchical Spatial Caching

Related Work

- Akleman & Chen 99
  - Similar approach for ray-linear & ray-quadric implicits (only requires 2D cache)
  - No BlobTree hierarchy
- Barthe et al 02
  - Incremental modeling w/ volume data sets
  - Ray-traced triquadratic reconstruction
  - Scalability limits

Observations

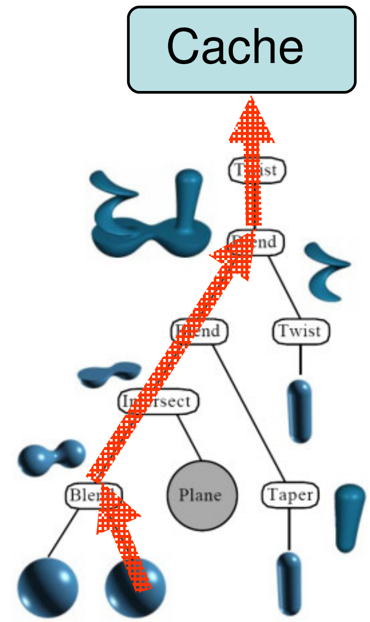
- Polygonization algorithms run interactively on volume data sets [Ferley et al 2000]

- For interactive modeling, most of the BlobTree structure is not changing frame-to-frame

- Memory is cheap
  - Less than $200 USD for 1GB of RAM
  - Easy to use

Global Spatial Caching

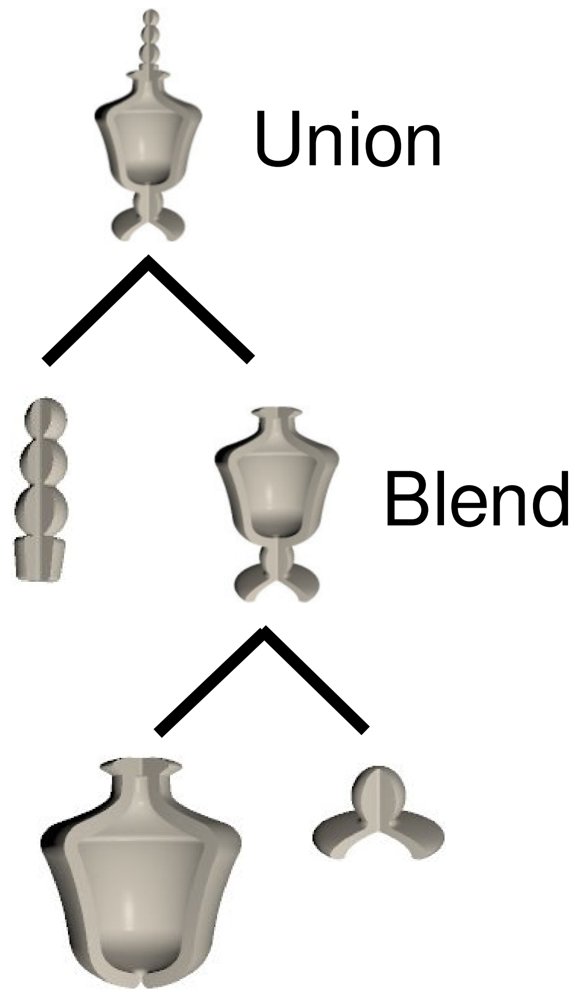
- Cache $f(\mathbf{p})$ in a volume data set:
  - O(N) → O(1) after lazy evaluation
  - Invalidate cache using field bounds of modified region
- Resolution is limited
- Does not fundamentally reduce evaluation cost
  - Large / expensive updates are still very slow



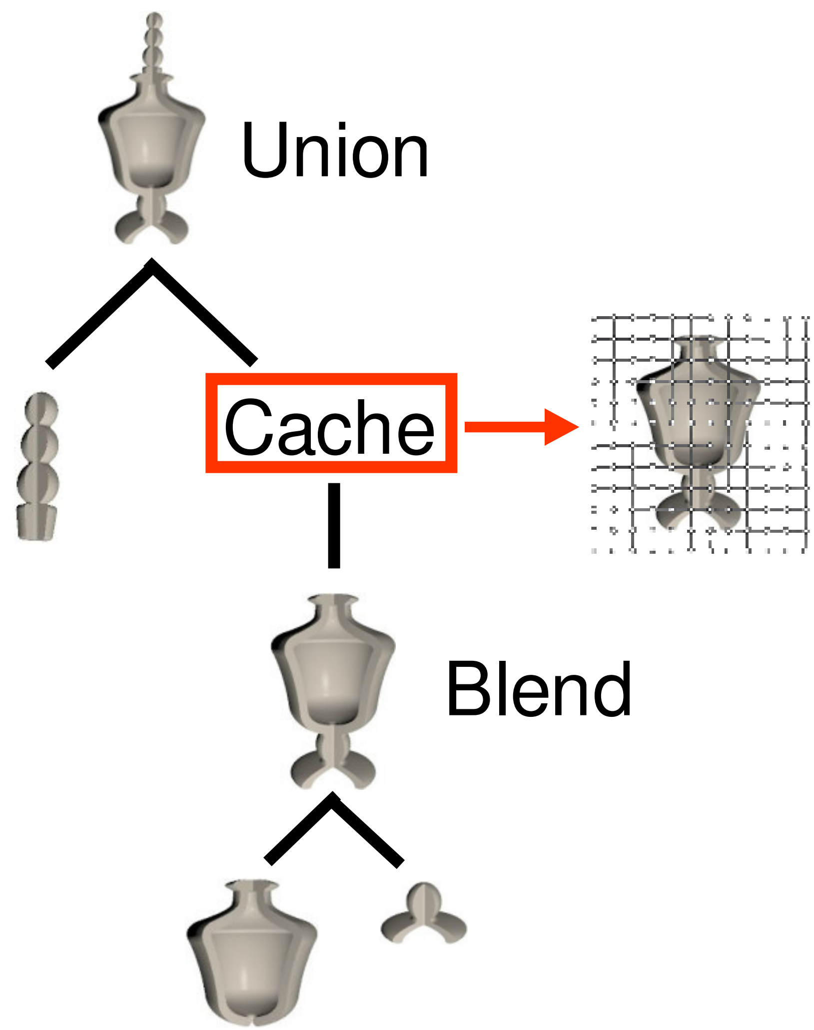
Hierarchical Spatial Caching

- Use multiple Spatial Caches
- Insert into the BlobTree as *Cache Nodes*
  - O(m) → O(1) for cache node subtree
- Faster updates than global spatial cache
  - Interactive modification only affects parent cache node(s)
  - Large / complex updates are possible

Standard BlobTree



Blob Tree With Cache Node



Cache Implementation

Spatial Cache Implementation

- Uniform Grids
- Trilinear and Triquadratic sampling
- Lazy evaluation
  - minimizes start-up overhead

Reconstruction Filters

- Trilinear Filter
  - 8 samples
  - C^0 Continuous [Marschner & Lobb 94]
  - Interpolating
- Triquadratic Filter [Barthe et al 02]
  - 27 samples required
  - C^1 Continuous
  - Approximating

Gradient Continuity

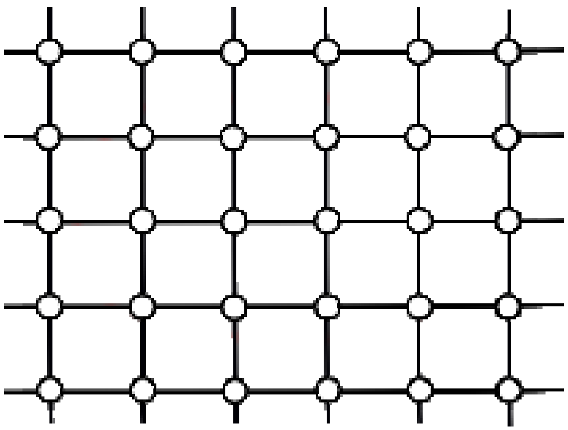
Trilinear

Triquadratic

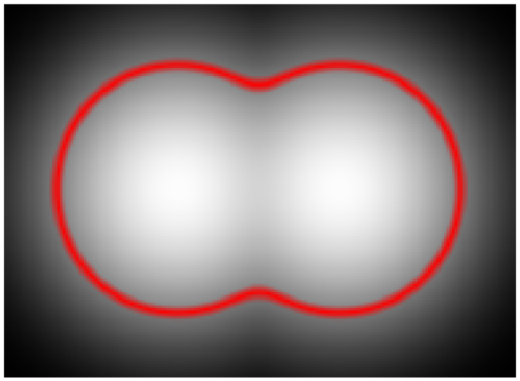
Interactive Visualization

- Trilinear filter is twice as fast
  - But gradient is unacceptable
- Gradient evaluation is only 10% of the polygonization cost
- Hybrid Solution:
  - Trilinear reconstruction for field value
  - Triquadratic reconstruction for gradient
- *Perceptually* smooth surface

Cache Sampling w/ Lazy Evalution

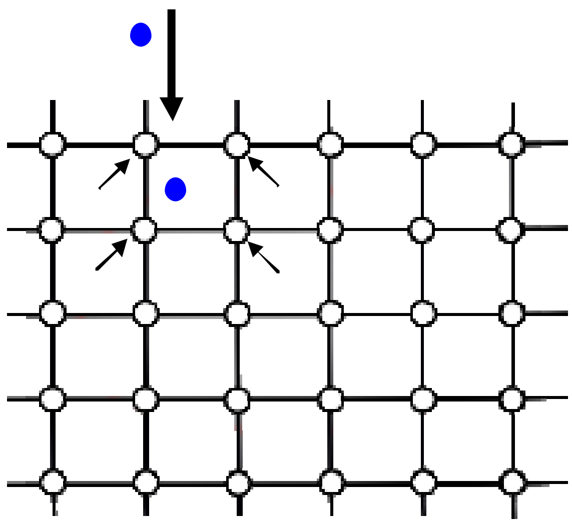


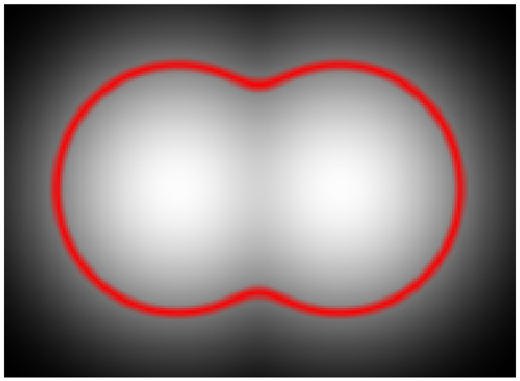
← Cache (Initially empty)



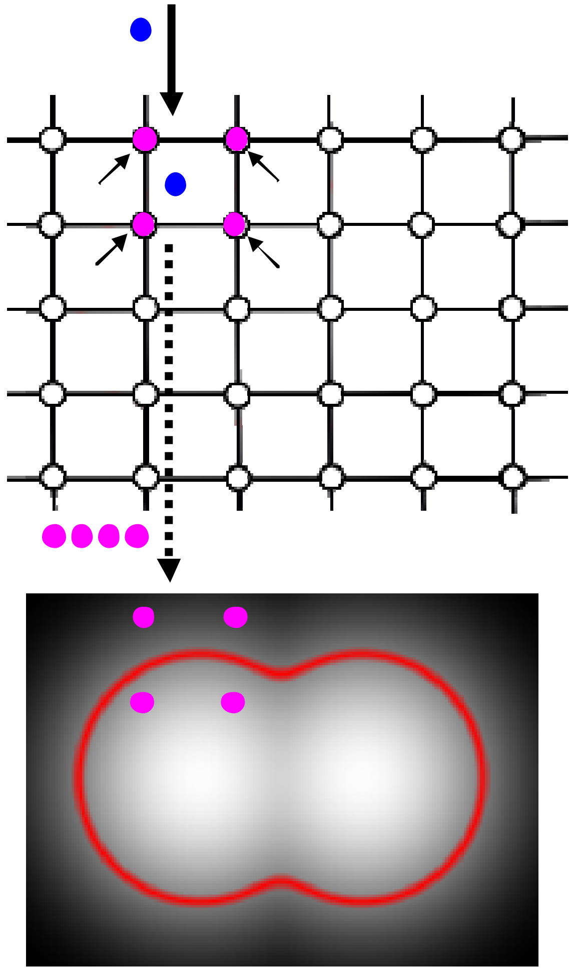
← Cached Subtree

Step 1: Incoming Point Query

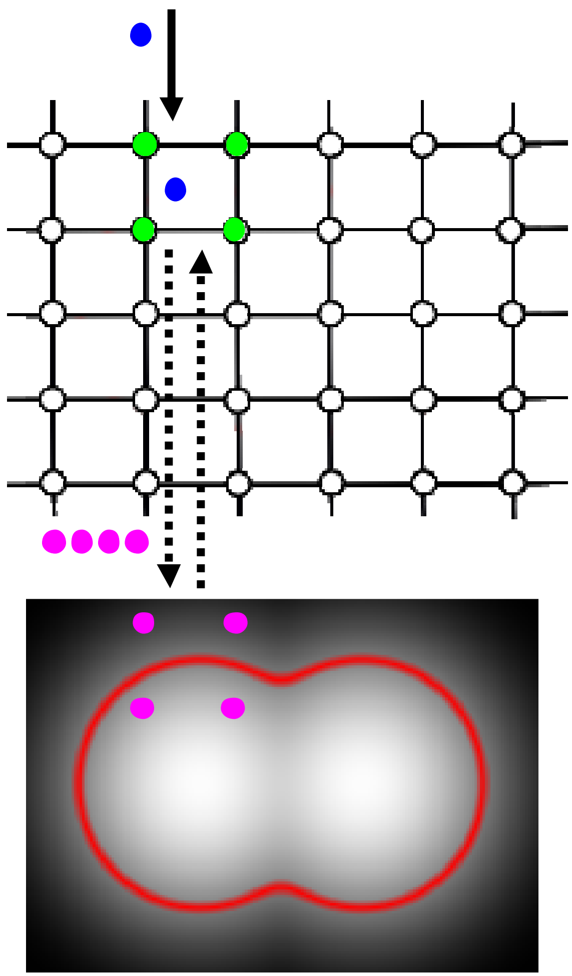




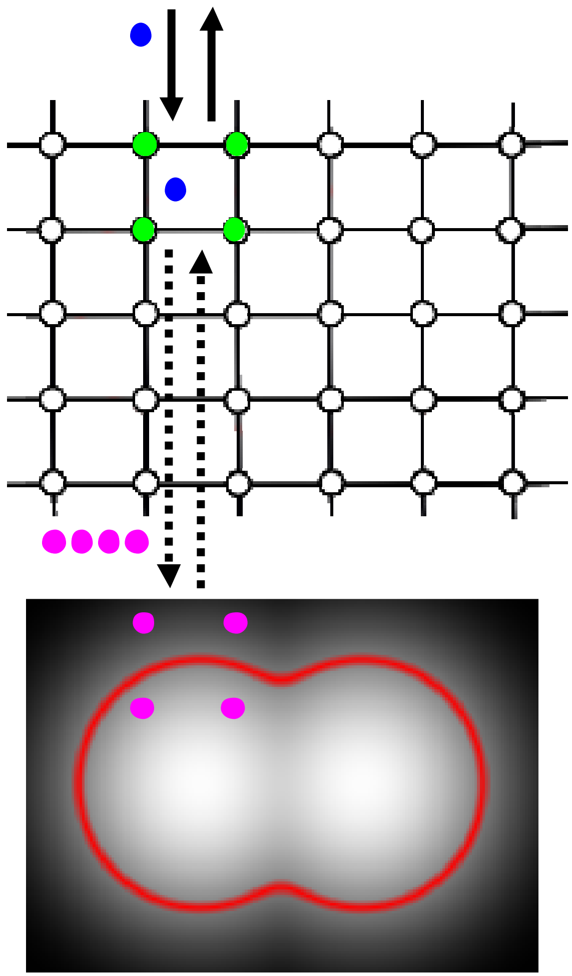
Step 2: Cached Subtree Evaluation



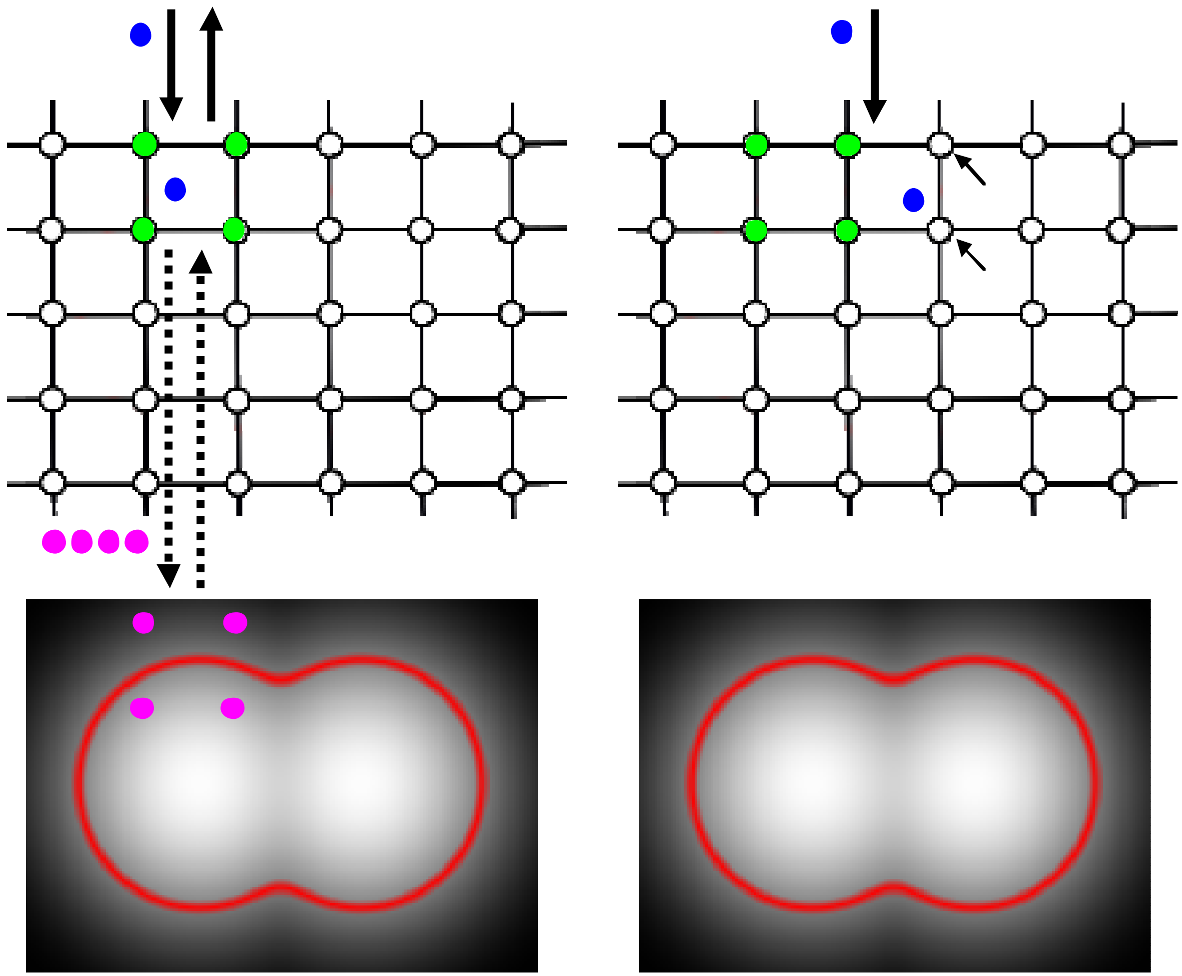
Step 3: Cache Initialization



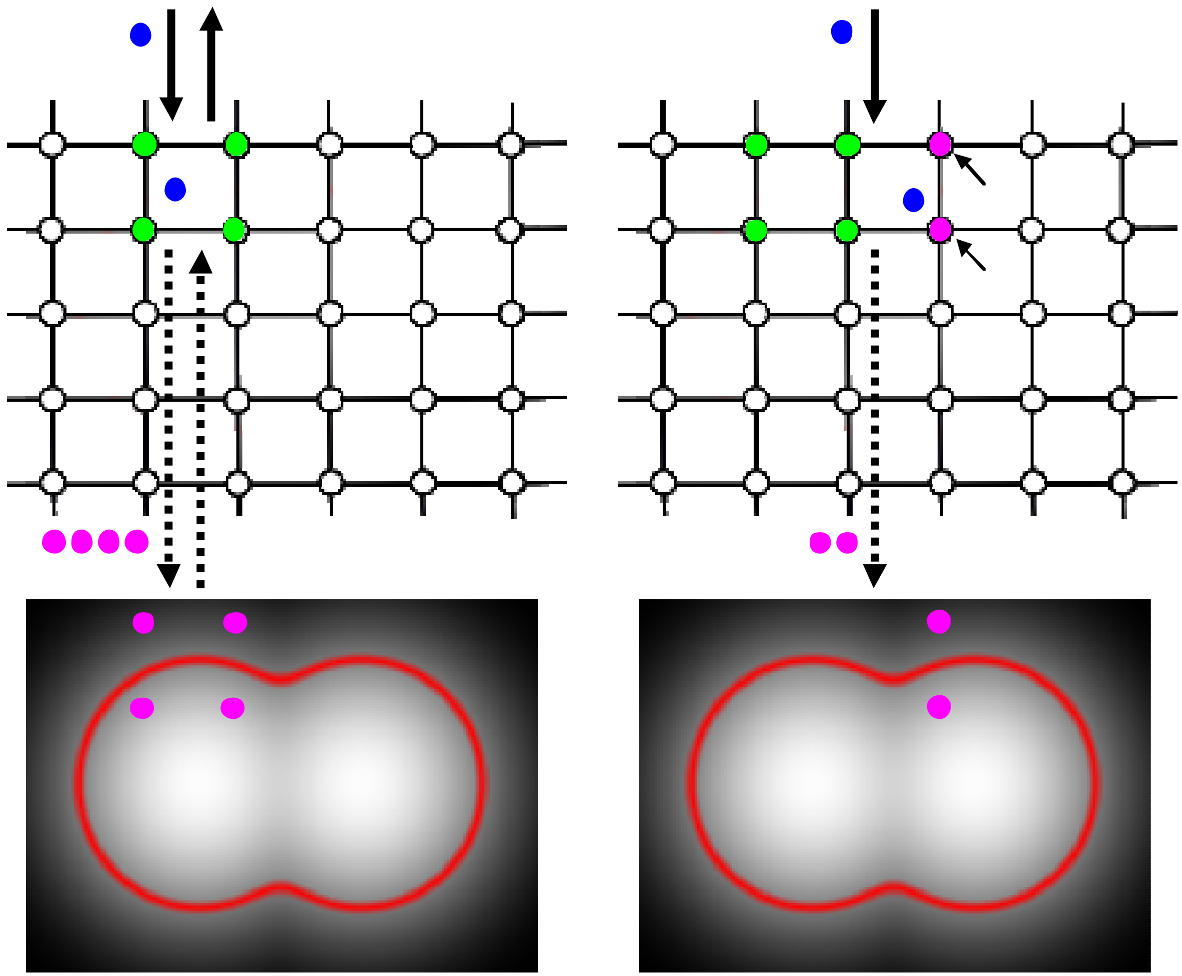
Step 4: Field Approximation



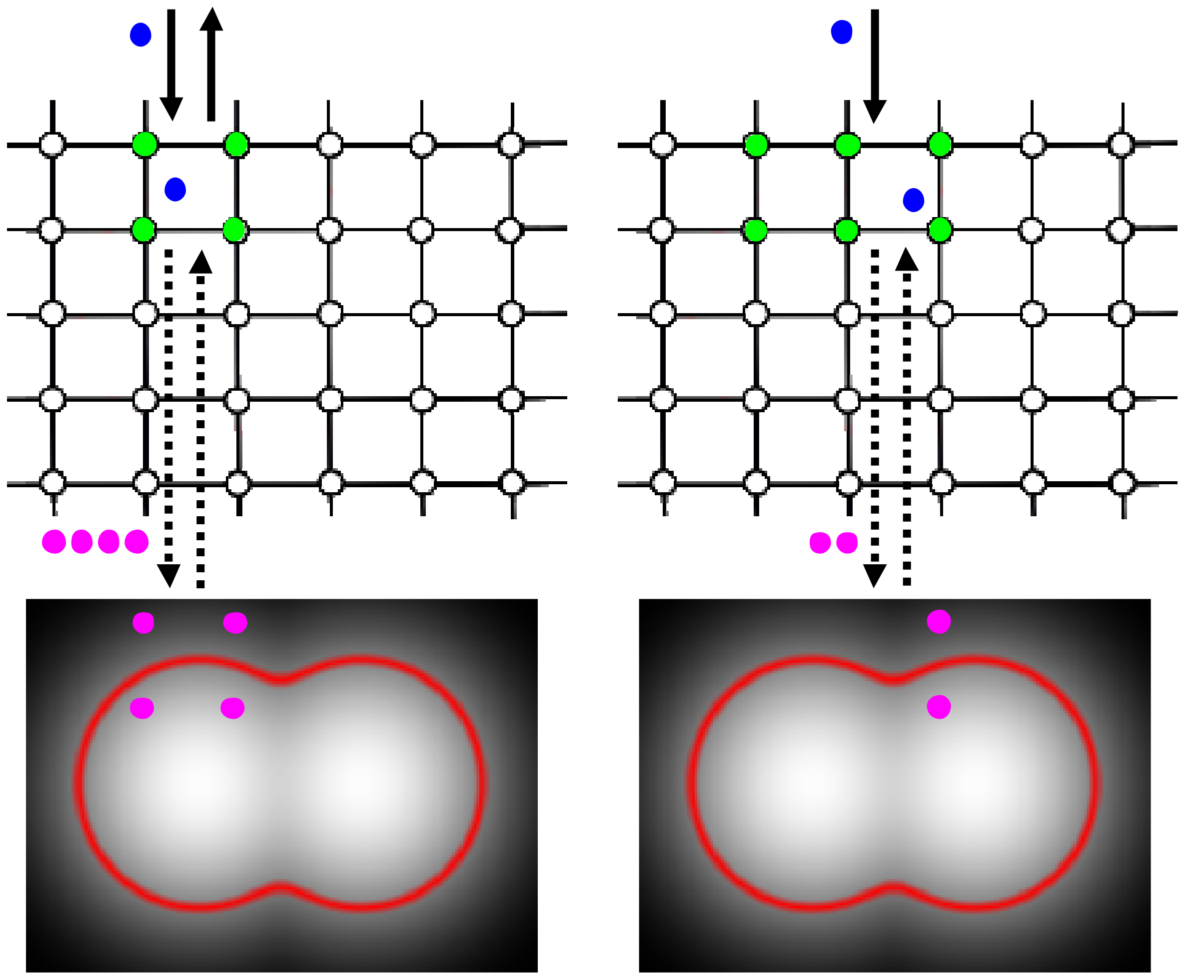
Step 1: Incoming Point Query



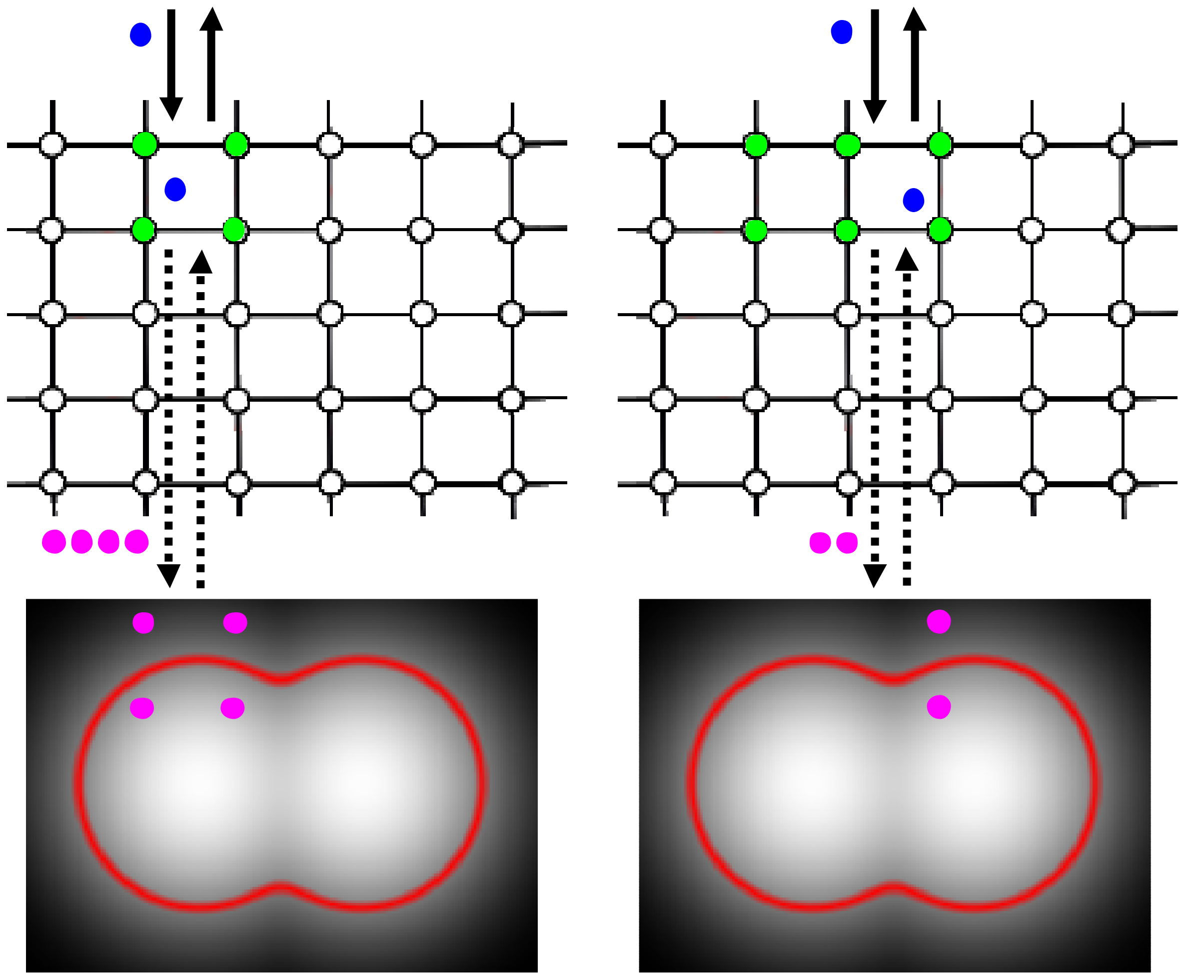
Step 2: Subtree Evaluation



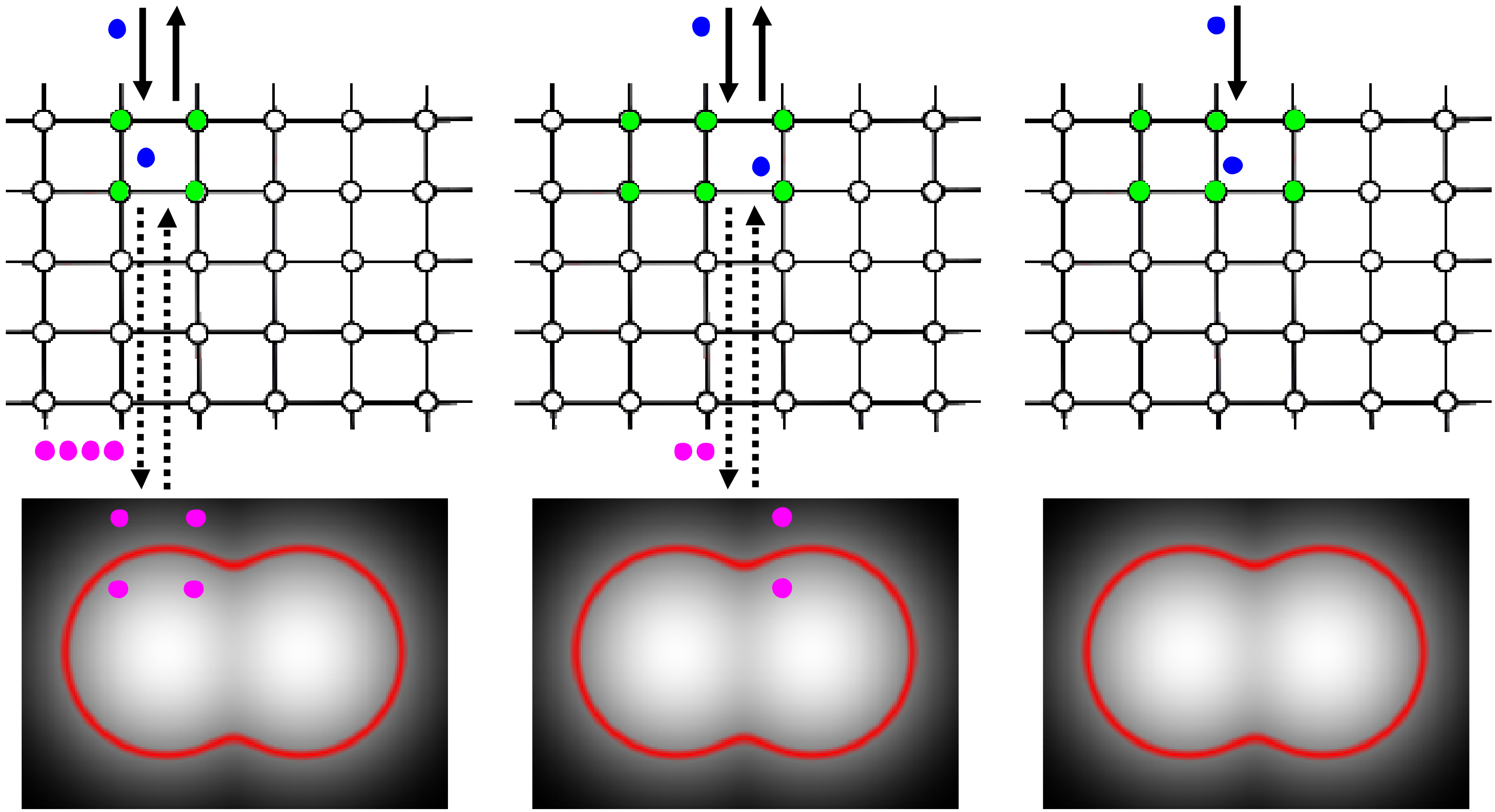
Step 3: Cache Initialization



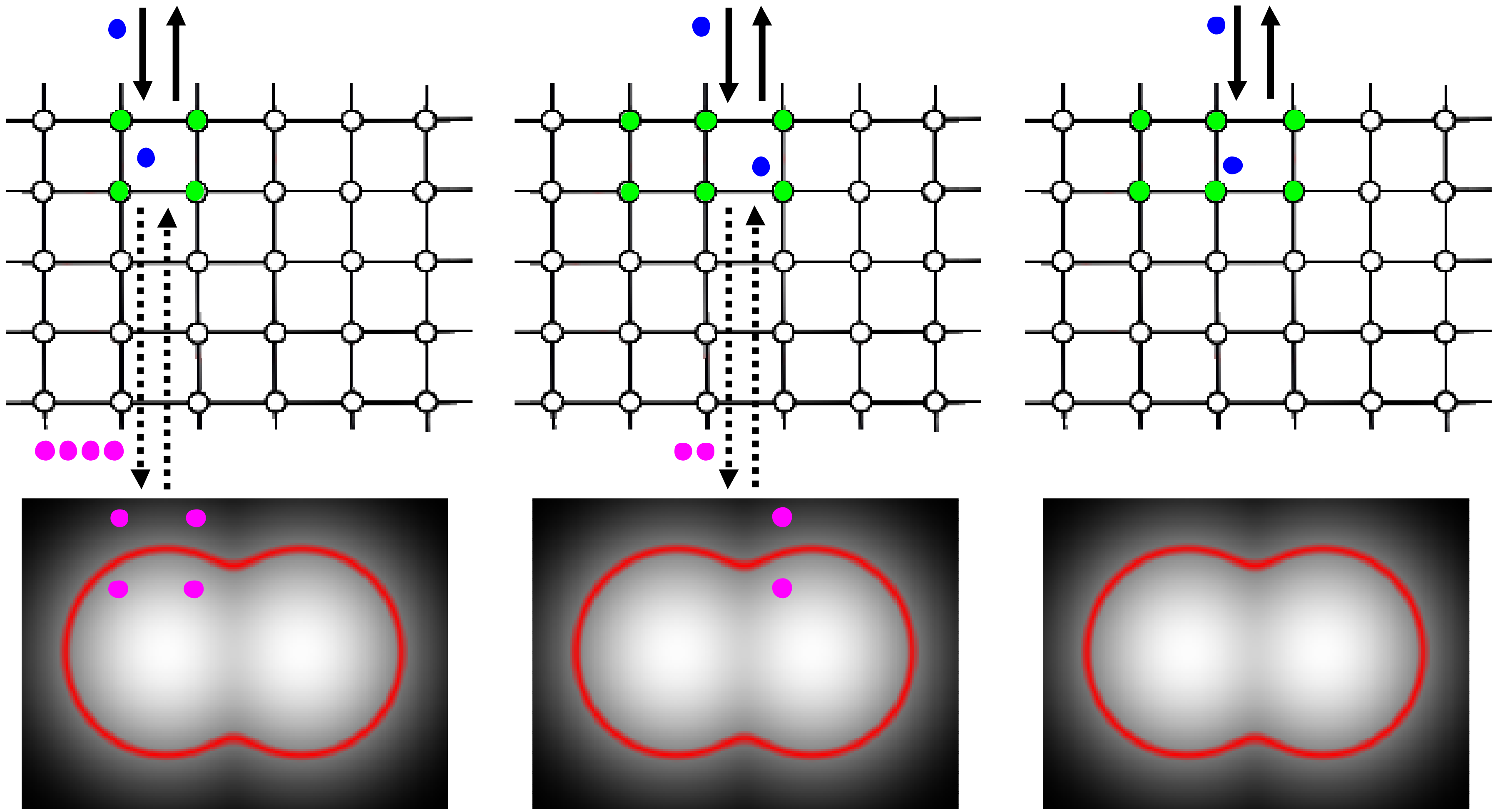
Step 4: Field Approximation



Step 1: Incoming Point Query

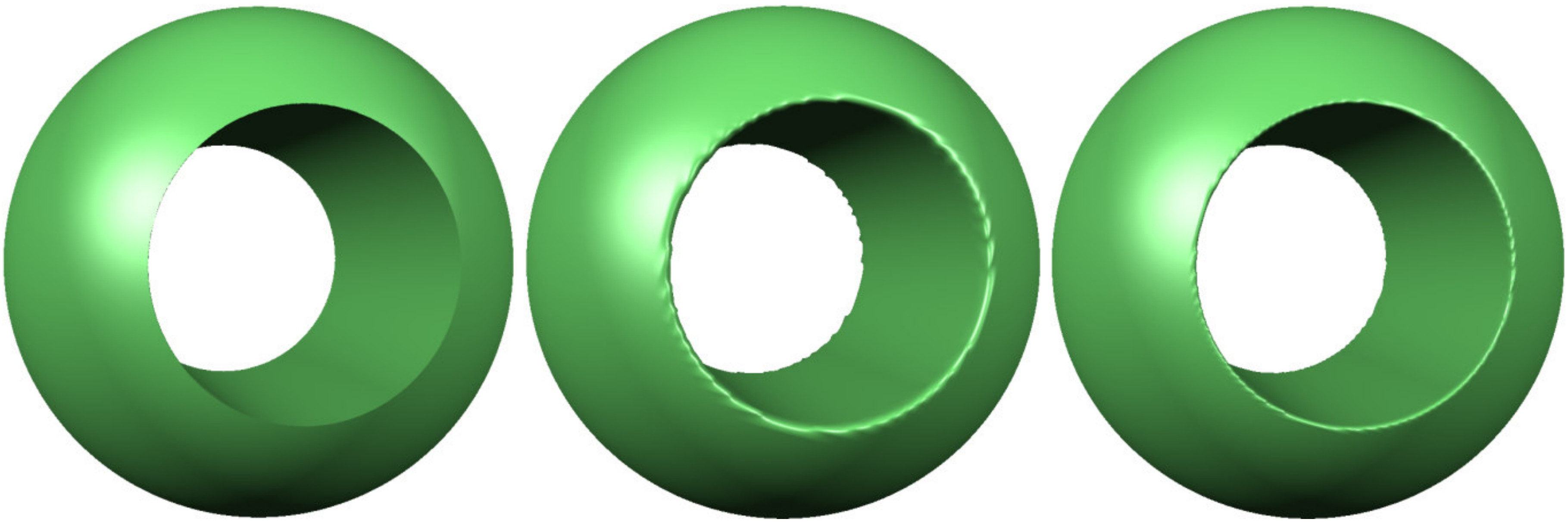


Step 4: Field Approximation



Sharp Feature Reconstruction

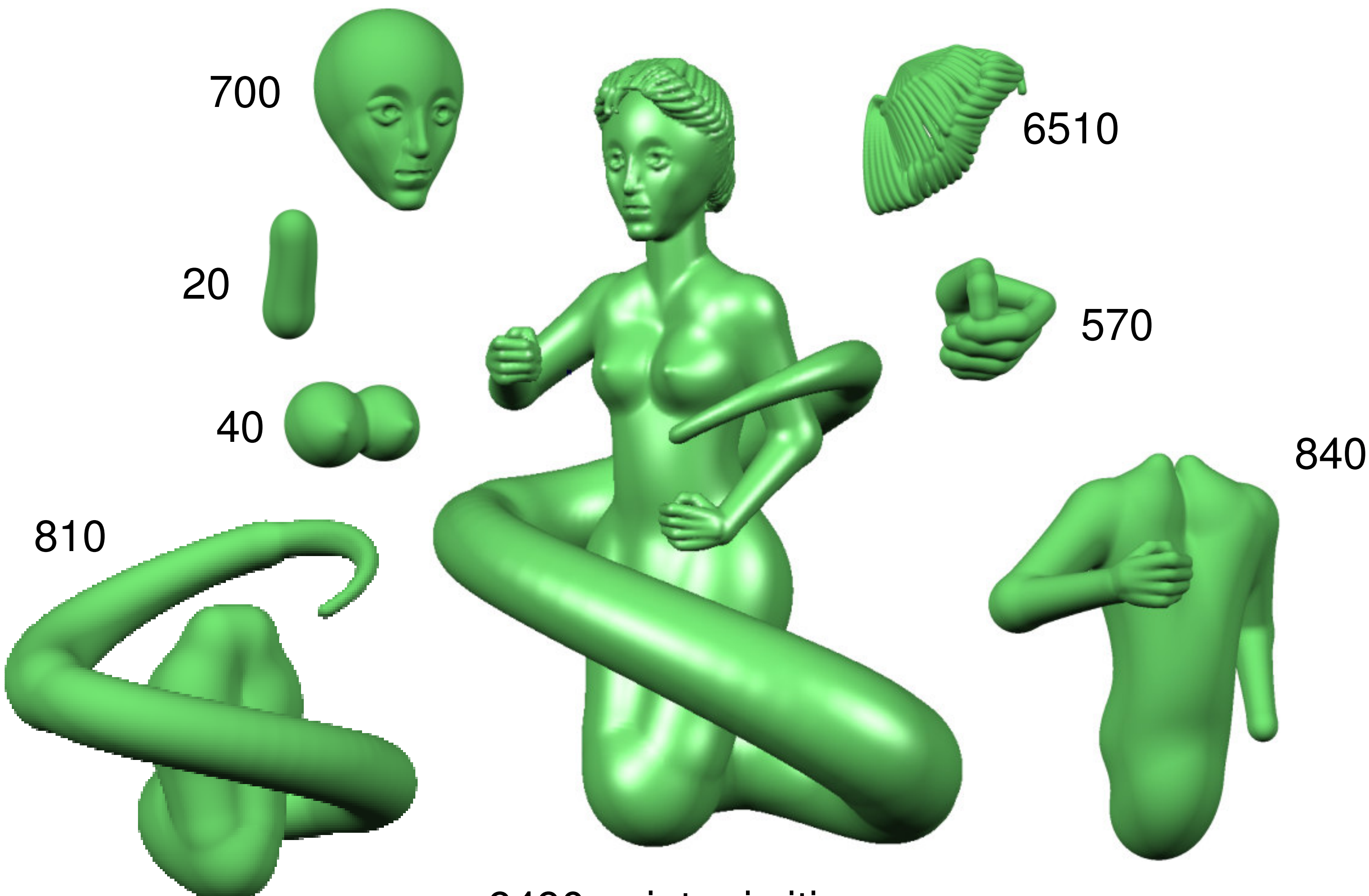
No Cache | 128^3 Cache | 256^3 Cache



(polygonizer resolution is much higher than cache resolution)

Profiling Results

Test Model



9490 point primitives

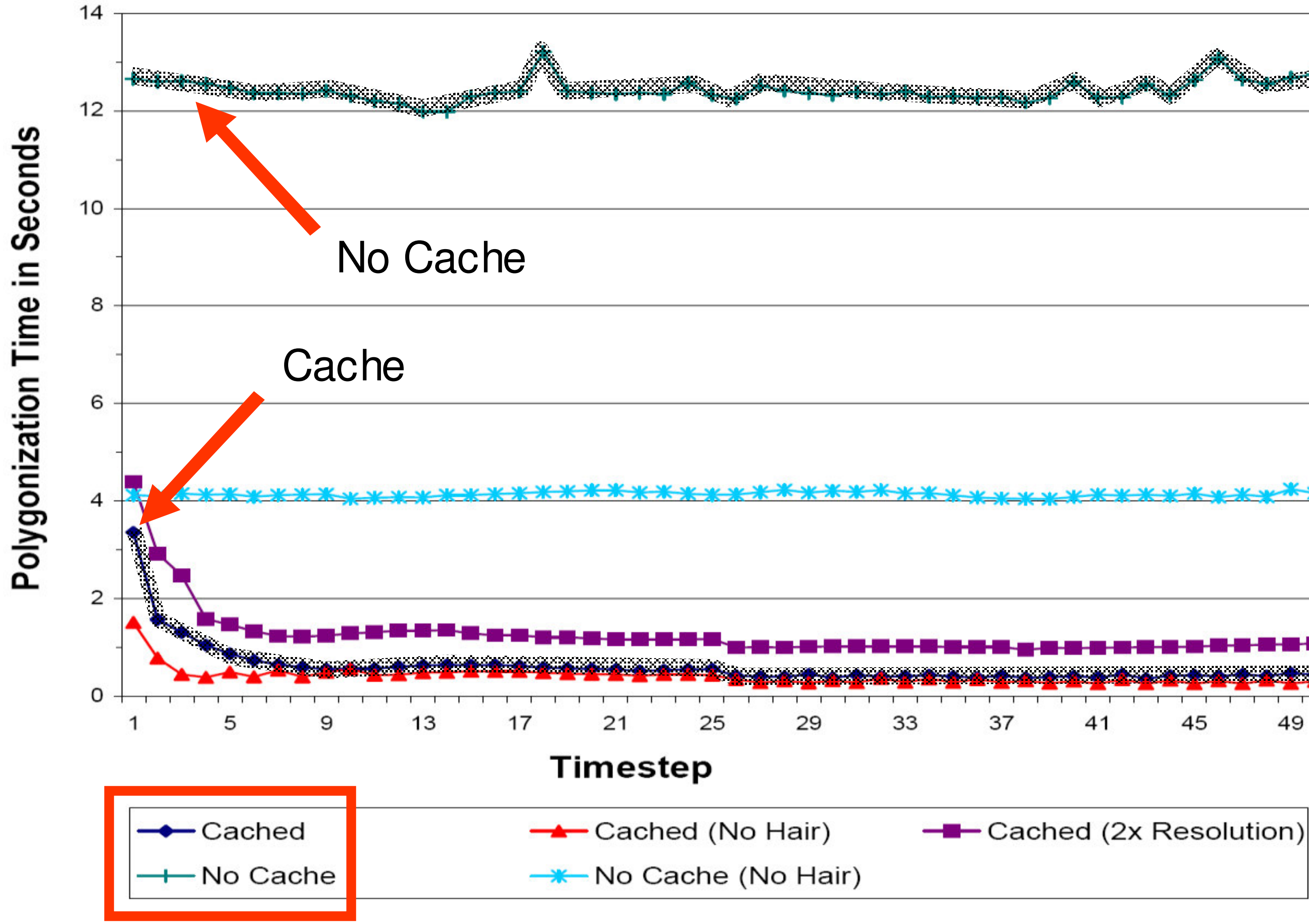
Static Polygonization

- Caches cleared before each test
- Time increase when resolution doubles:

  Cache → 1.7x   No Cache → ~4x

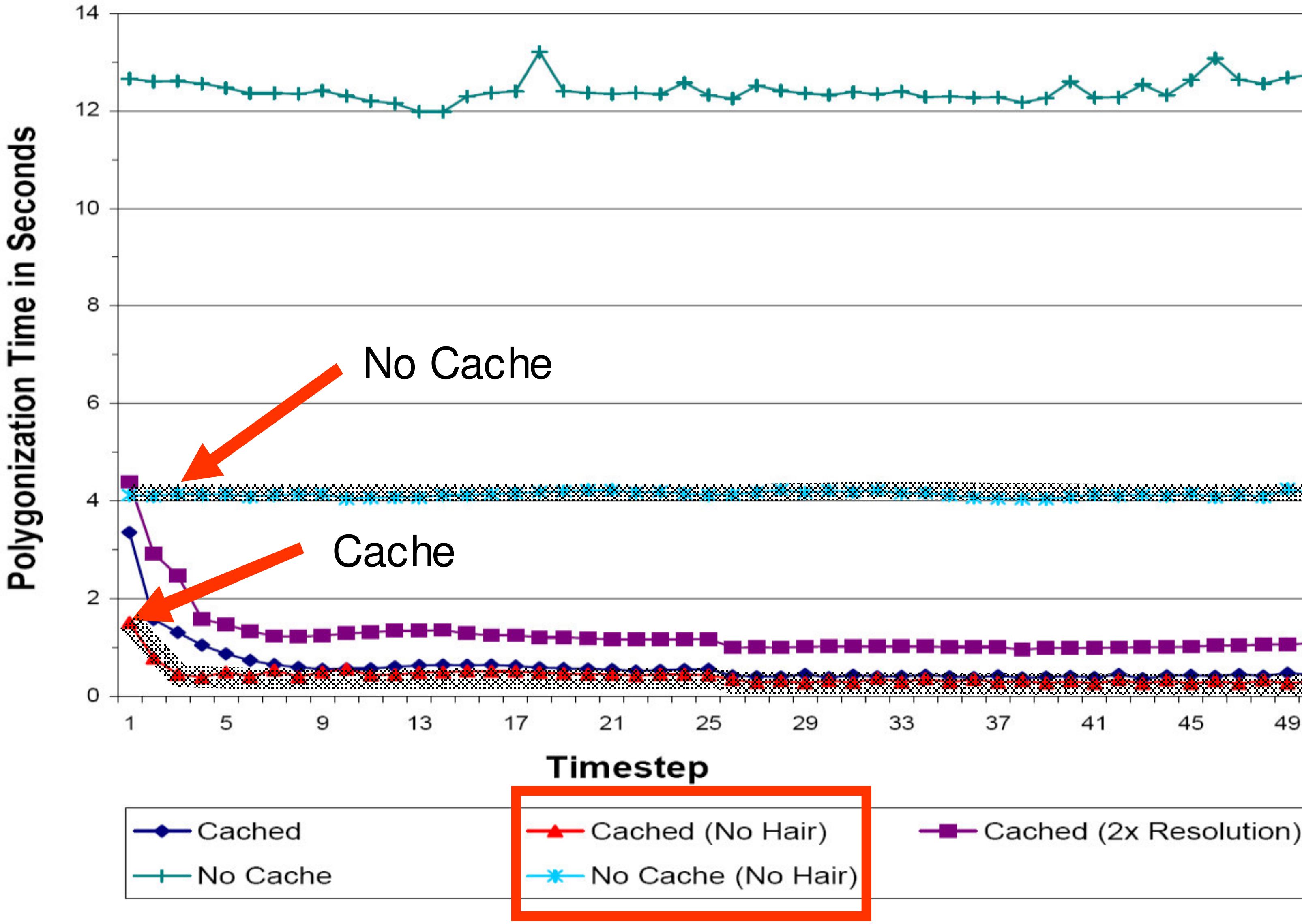
Cubes	Cache	No Cache	Ratio	Triangles (Approx.)
32^3	5.77	4.90	0.8×	4,000
64^3	10.34	14.36	1.4×	16,000
128^3	17.40	51.97	3×	61,000
256^3	29.23	199.37	6.5×	239,000
512^3	49.83	809.66	16×	955,000

Head Translation Test



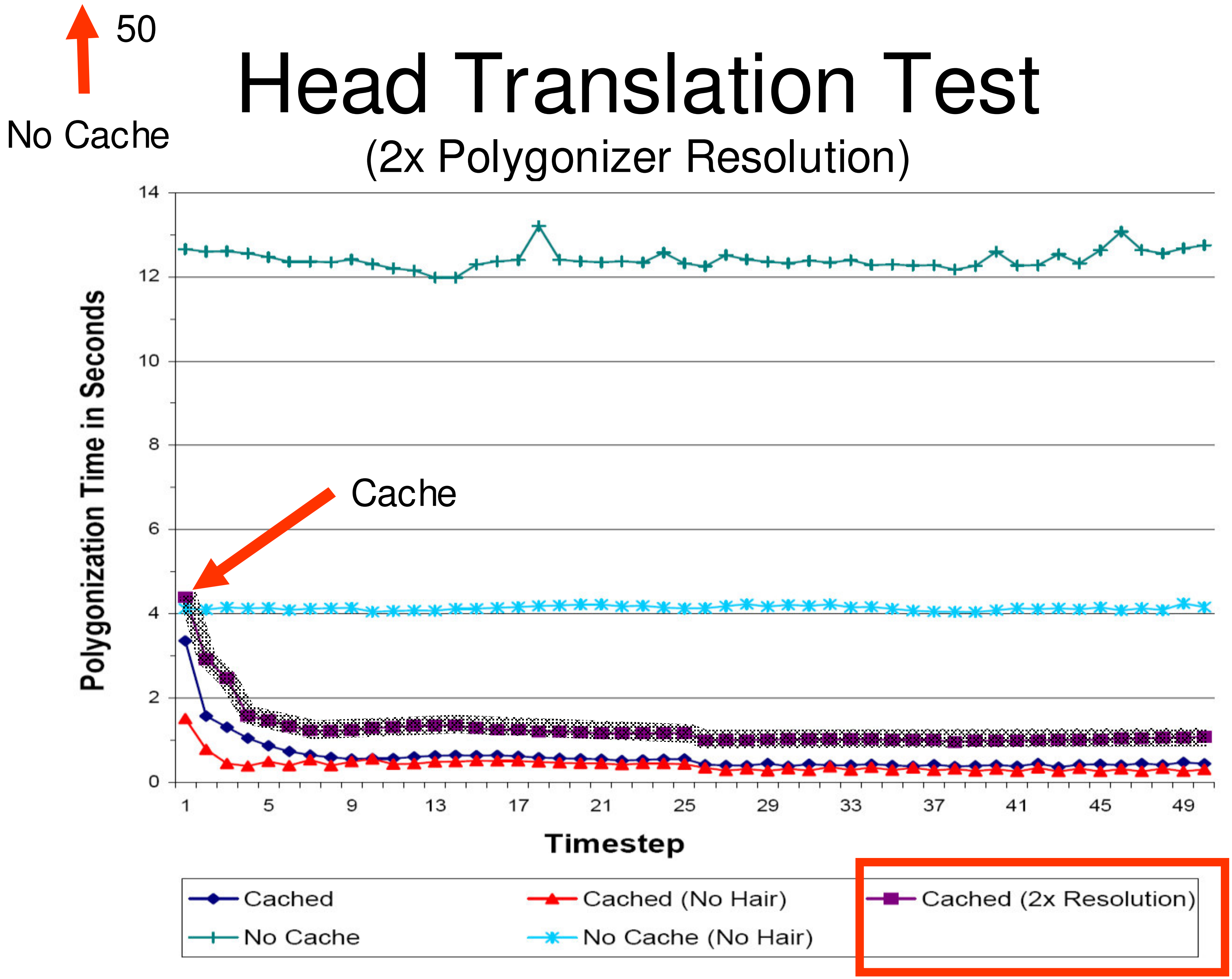
Head Translation Test

(No Hair)



Head Translation Test

(2x Polygonizer Resolution)



Local Update Polygonization

- Recompute polygonization only in modified region
- Point / Hair Test simulates detail work on hair component (6500 point primitives)

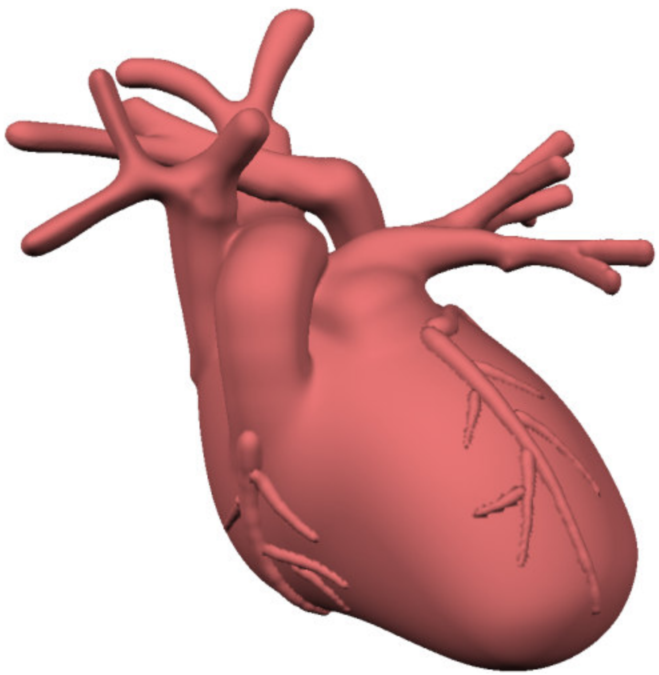
Test	Cubes	Improvement
Head Translation	60^3	$7\times$
Head Translation	120^3	$12\times$
Point / Hair Test	60^3	$30\times$
Point / Hair Test	120^3	$47\times$

Applications

Interactive Model Assembly

Demo

Proof-of-concept:
Interactive Sketch-Based BlobTree Modeling



30 minutes

1 hour

15 minutes

(Come see our SIGGRAPH 05 sketch)

Future Work

- Improving Accuracy
  - Alternate spatial caching schemes?
- Automatic Cache Management Algorithms
  - Take advantage of interactive context
- Minimize invalidation regions
  - Better bounds
- Interactive implicit modeling interfaces
  - Largely unexplored territory…

Summary

- We described hierarchical spatial caching for the BlobTree implicit modeling system
- We compared polygonization times with traditional approaches
  - An order-of-magnitude improvement was found
- We described several interactive modeling applications enabled by this technique

Questions?