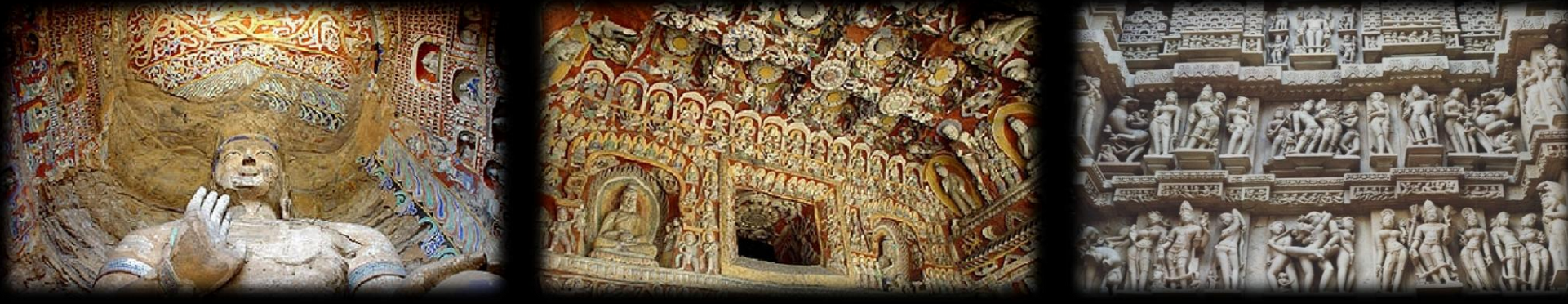


GPU based Global Illumination of Point Models using FMM

Rhushabh Goradia, Prekshu Ajmera, Sharat Chandran | ViGIL IIT Bombay
{rhushabh, prekshu, sharat}@cse.iitb.ac.in

Motivation

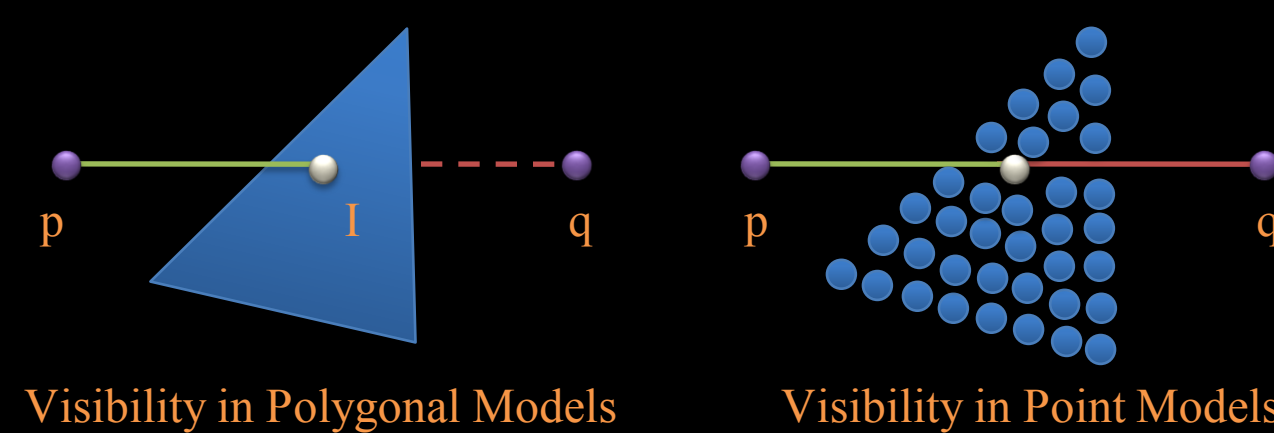


- Scans of cultural heritage structures involve millions of points
- Seamless, geometrically consistent mesh creation is virtually not possible as it is difficult to automatically segment different objects
- Wish to preserve in a virtual museum and view them under novel, complex lighting conditions

Problem Statement

- Capturing radiosity based inter-reflection effects in a scene when the inputs are point samples of hard to segment entities
- Exploit the inherent parallelism in the proposed solution using CUDA based GPU techniques to achieve fast solutions

Global Illumination and Visibility



- Computing visibility between a point-pair is difficult due to lack of intermediate surface information

Contributions

- **View-Independent visibility** solution between point-pairs required for computing *correct* global illumination
- **Radiosity** solution for complex point models using the **Fast Multipole Method**
- Exploit **parallelism** in the proposed algorithm to achieve upto **20x speed-up** using GPUs

Fast Multipole Method

- FMM is concerned with evaluating the effect of a “set of source points” Y , on a set of “evaluation points” X

$$X = \{x_1, x_2, \dots, x_n\}, x_i \in \mathbb{R}^3, i = 1 \dots N$$

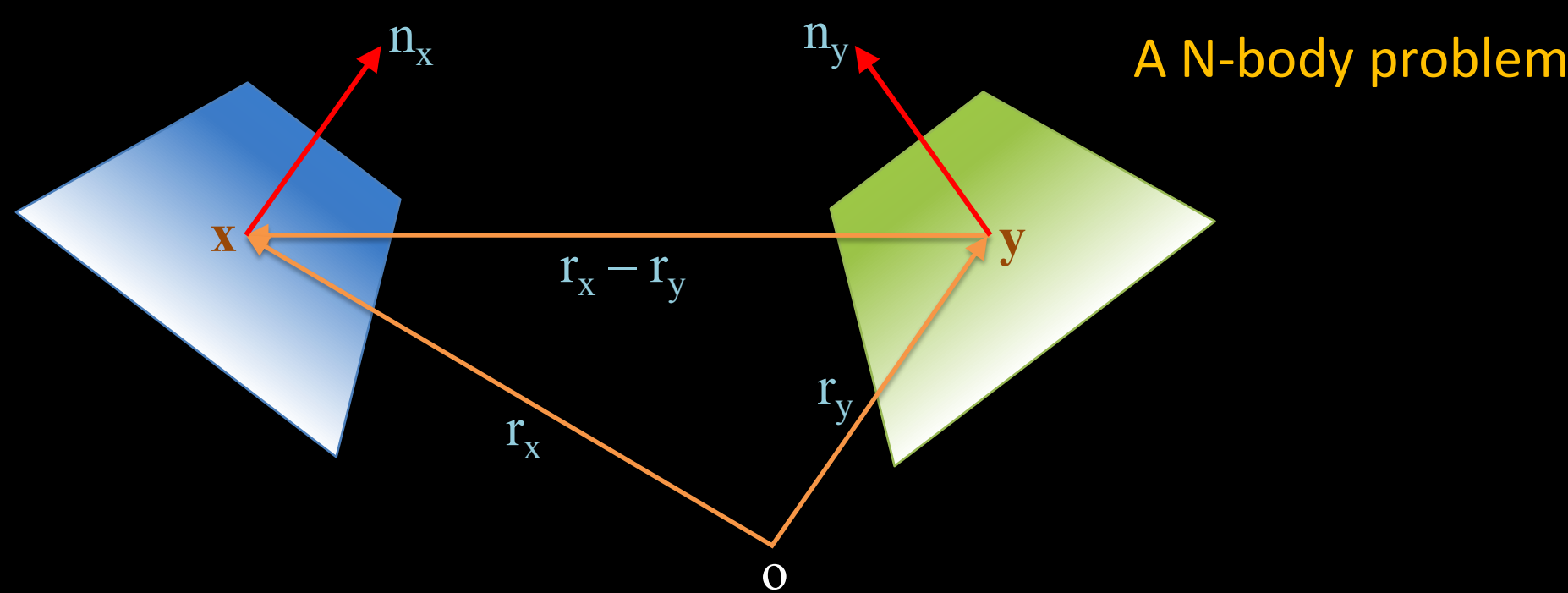
$$Y = \{y_1, y_2, \dots, y_m\}, y_j \in \mathbb{R}^3, j = 1 \dots M$$

$$f(x_i) = \sum_{j=1}^M \phi(x_i, y_j), i = 1 \dots N$$

Total complexity : $O(NM)$

- FMM attempts to reduce this irreducible complexity to $O(N+M)$
- The three main insights that make this possible are
 1. **Factorization** of the kernel into source and receiver terms
 2. Many application domains can tolerate **approximations** upto a certain level
 3. Clever use of hierarchical structure (**Octree**)
- Each node has an associated Interaction List

Radiosity-based Global Illumination

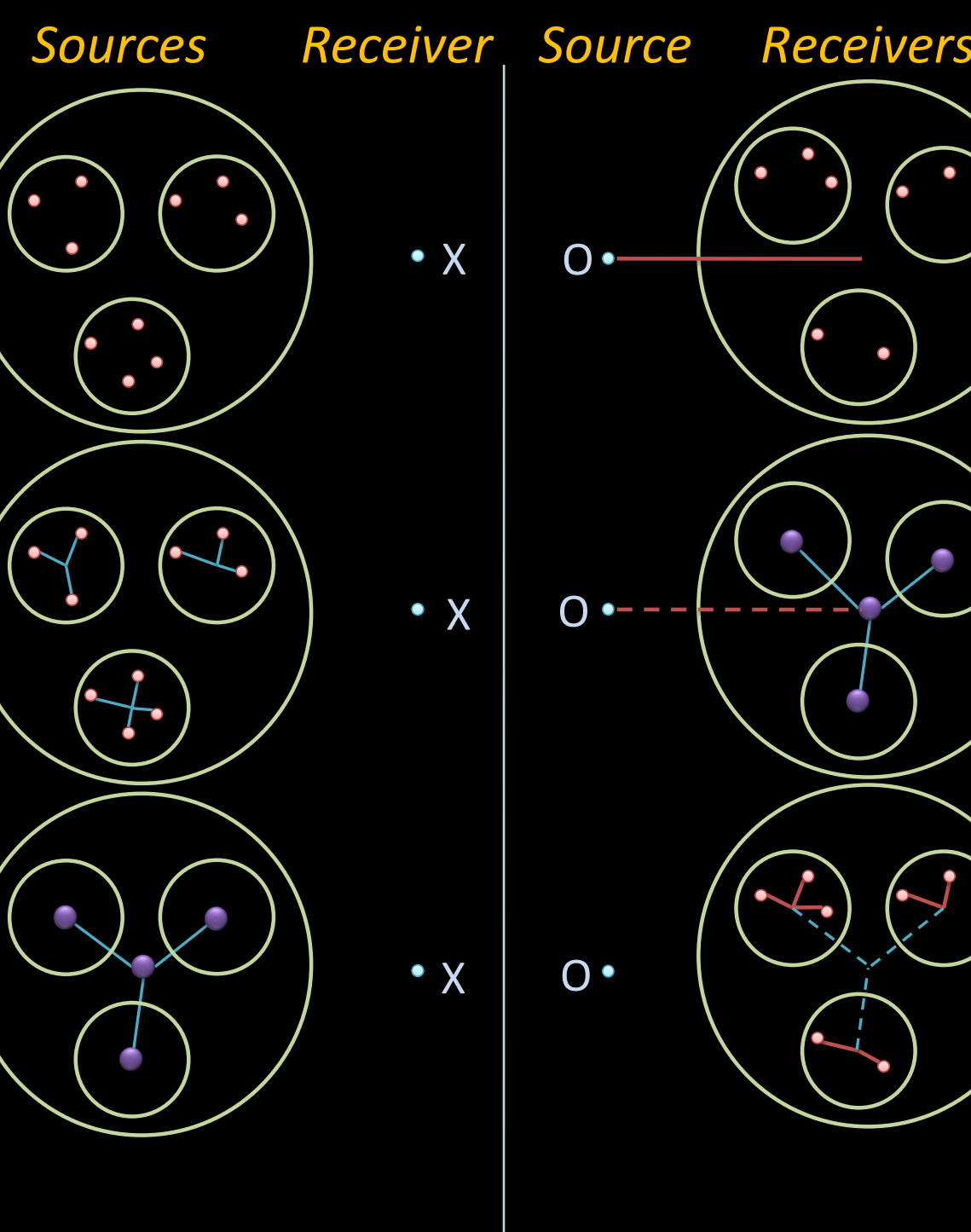
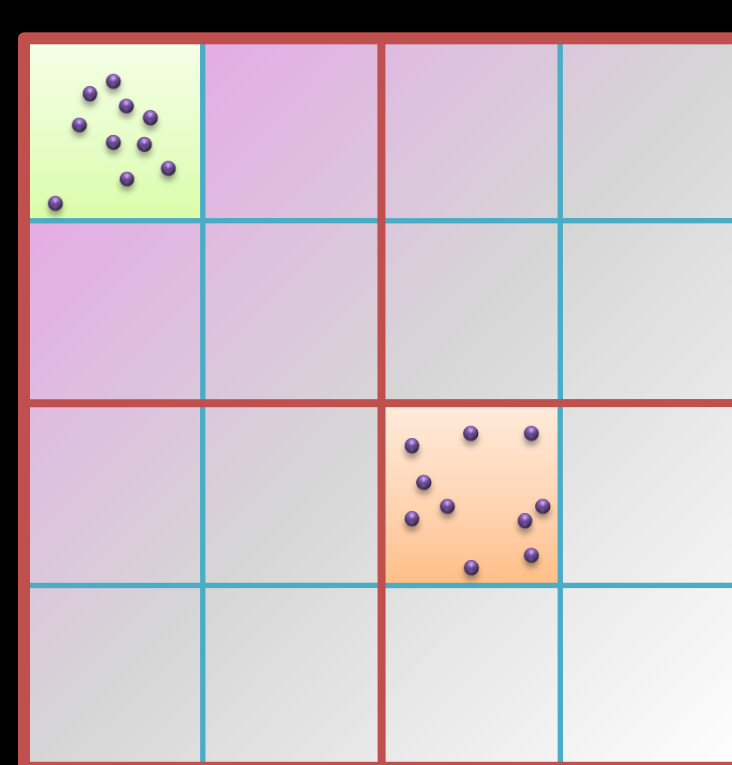


$$B(x) = E(x) + \rho(x) \int_{y \in S} \frac{[\vec{n}_y \cdot (\vec{r}_x - \vec{r}_y)] [\vec{n}_x \cdot (\vec{r}_y - \vec{r}_x)]}{\pi |\vec{r}_y - \vec{r}_x|^4} B(y) dy$$

FMM: The Technique

Phases of FMM

1. Octree Construction
2. Interaction List Construction
3. Point-Pair Visibility
4. Upward Pass
5. Downward Pass
6. Final Summation Stage



$$\frac{1}{|\vec{r}_y - \vec{r}_x|^4} = \sum_{n=0}^{\infty} \sum_{j=0}^{\lfloor n/2 \rfloor} \sum_{m=-n+2j}^{n-2j} \pi_n^j \left\{ \frac{1}{r_x^{n+4}} Y_{n-2j}^m(\theta_x, \phi_x) \right\} \left\{ \overline{r_y^n Y_{n-2j}^m(\theta_y, \phi_y)} \right\}$$

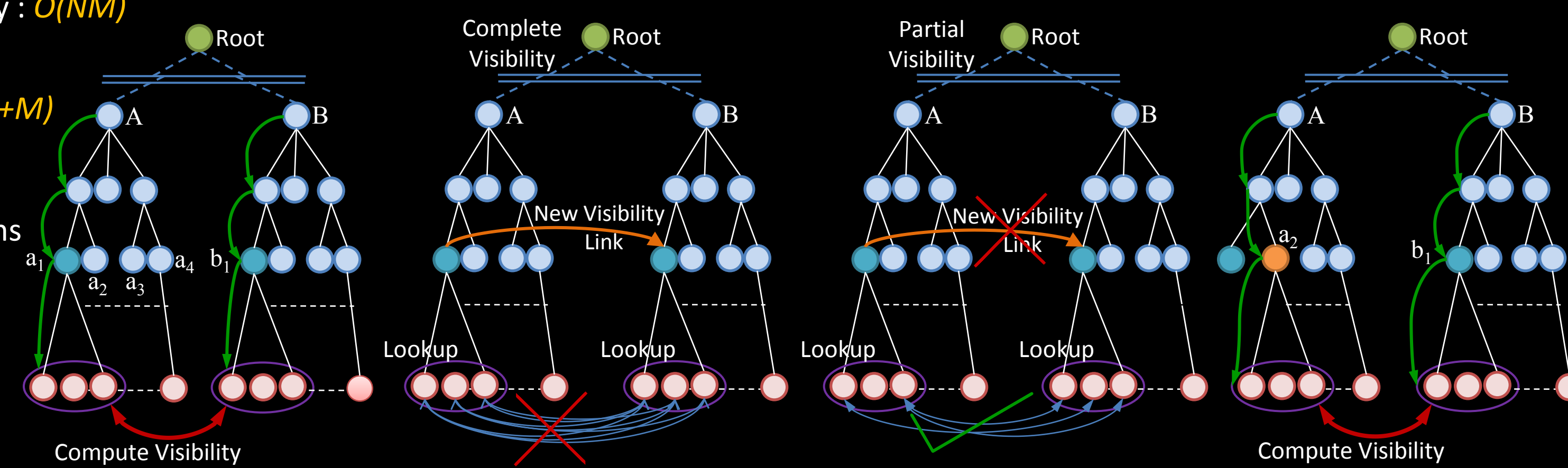
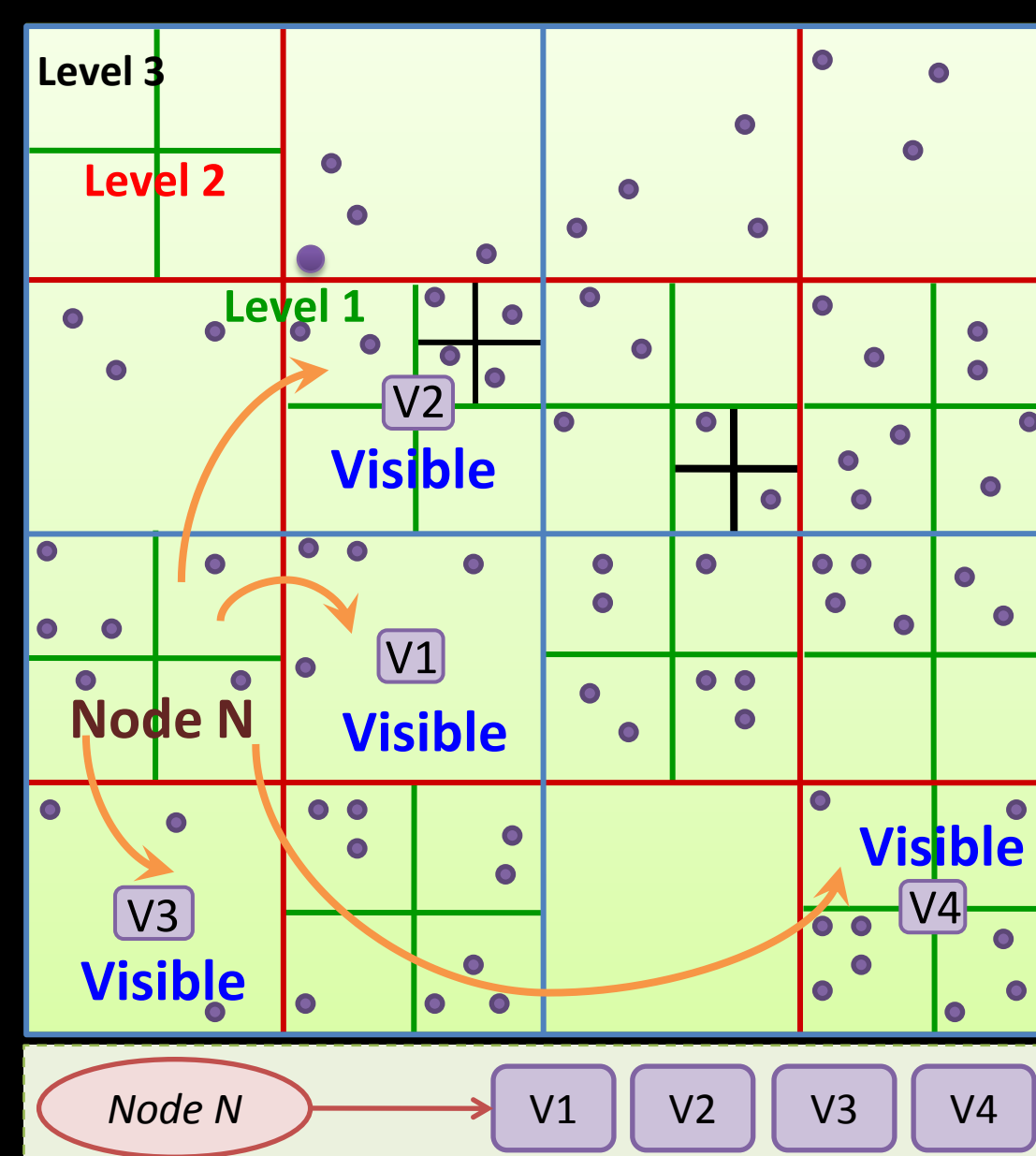
[Karapurkar A., Chandran S., ICGIP 2004]

Assigning Weights to Points

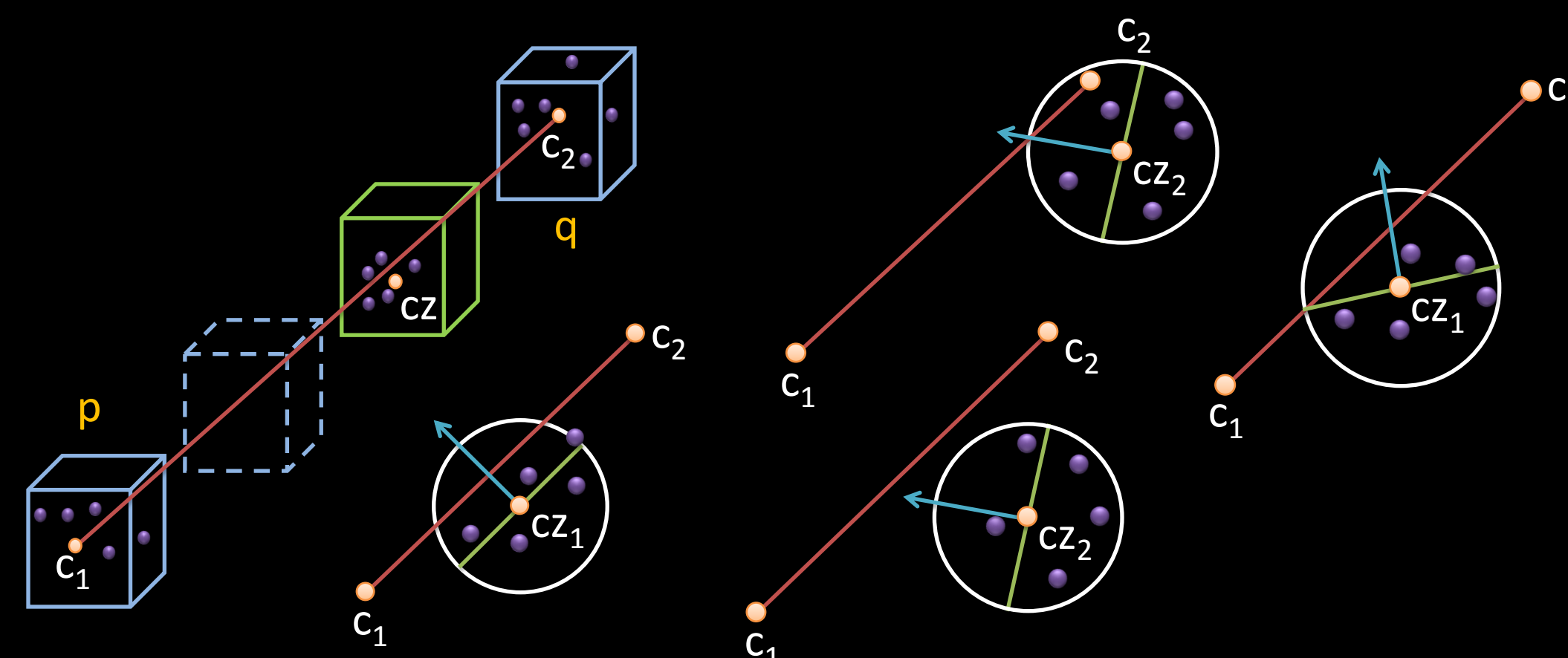
- Requires integration over a surface area
- For Point Based Models, we do not have any surface information; we therefore approximate this integration
- Weights are assigned to each point and signify the contribution of the point to the reconstruction of the surface
- Local property based on the normal available at points. As the number of points increase, the integration is computed more accurately

Visibility Map Construction

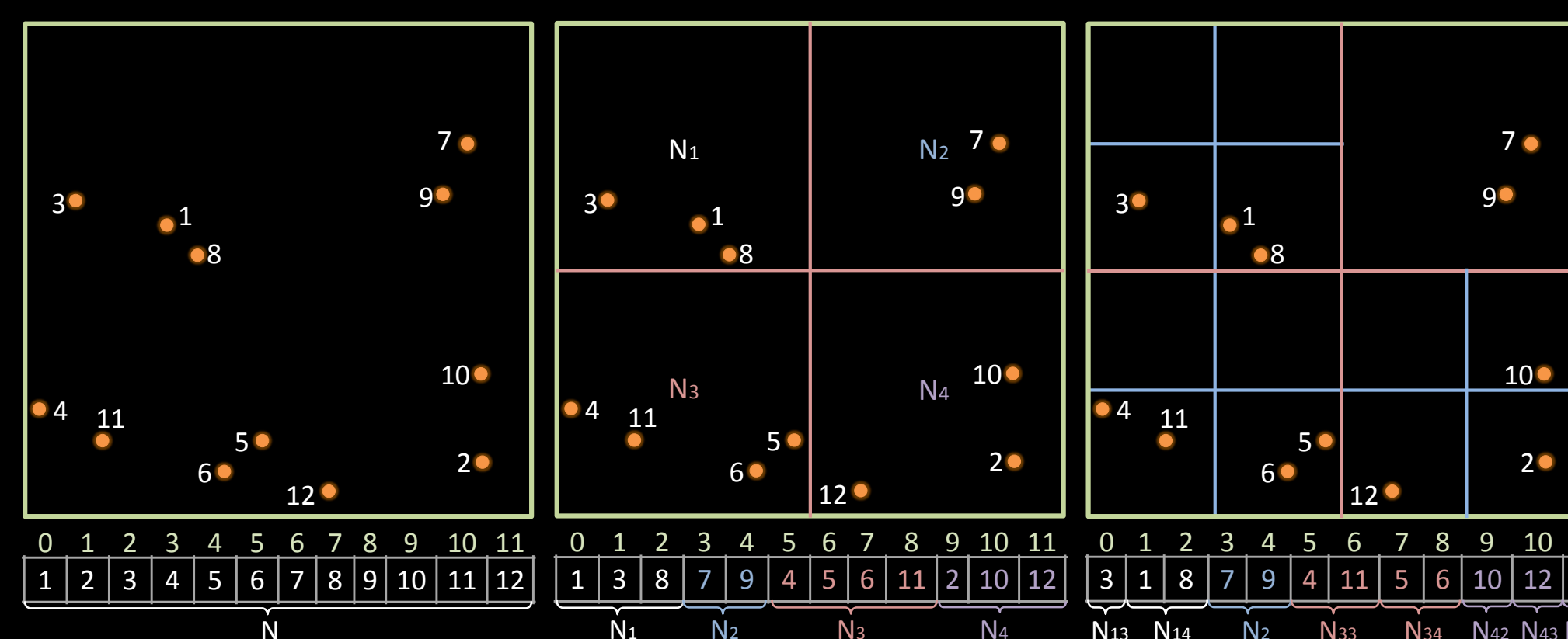
- The visibility map (or V-map) for a tree is a collection of visibility links for every node in the tree. The **visibility link** for any node p is a list L of nodes at the same level; every point in any node in L is guaranteed to be visible from every point in p
- The visibility map is constructed recursively by a variation of **DFS**
- In general, it is advantageous to have links at high level in the tree



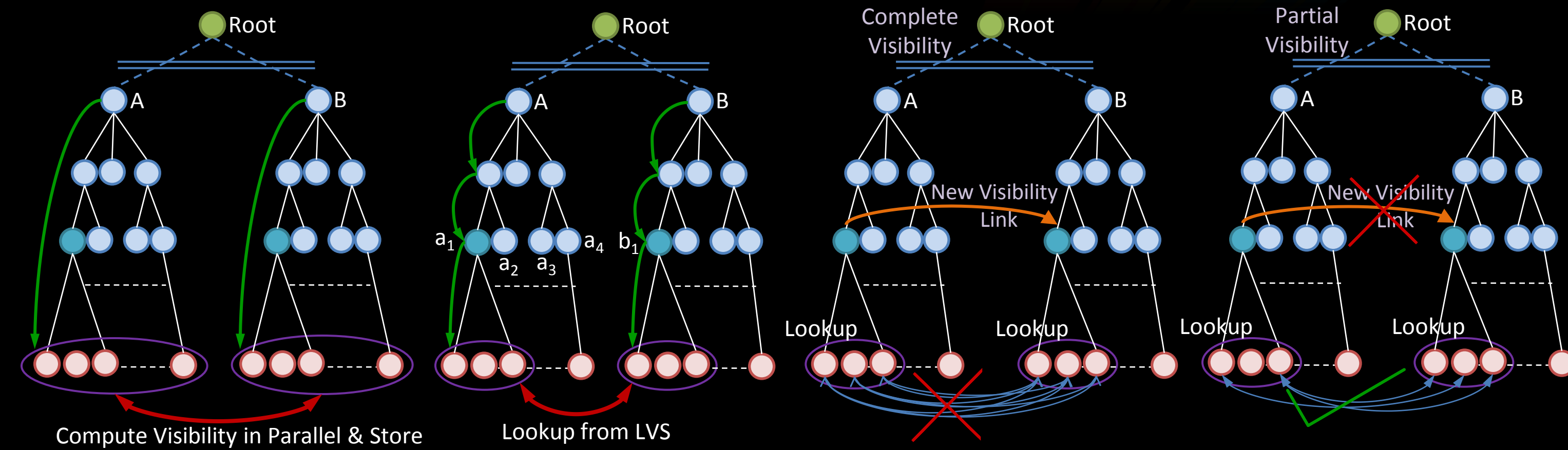
Leaf-Leaf Visibility



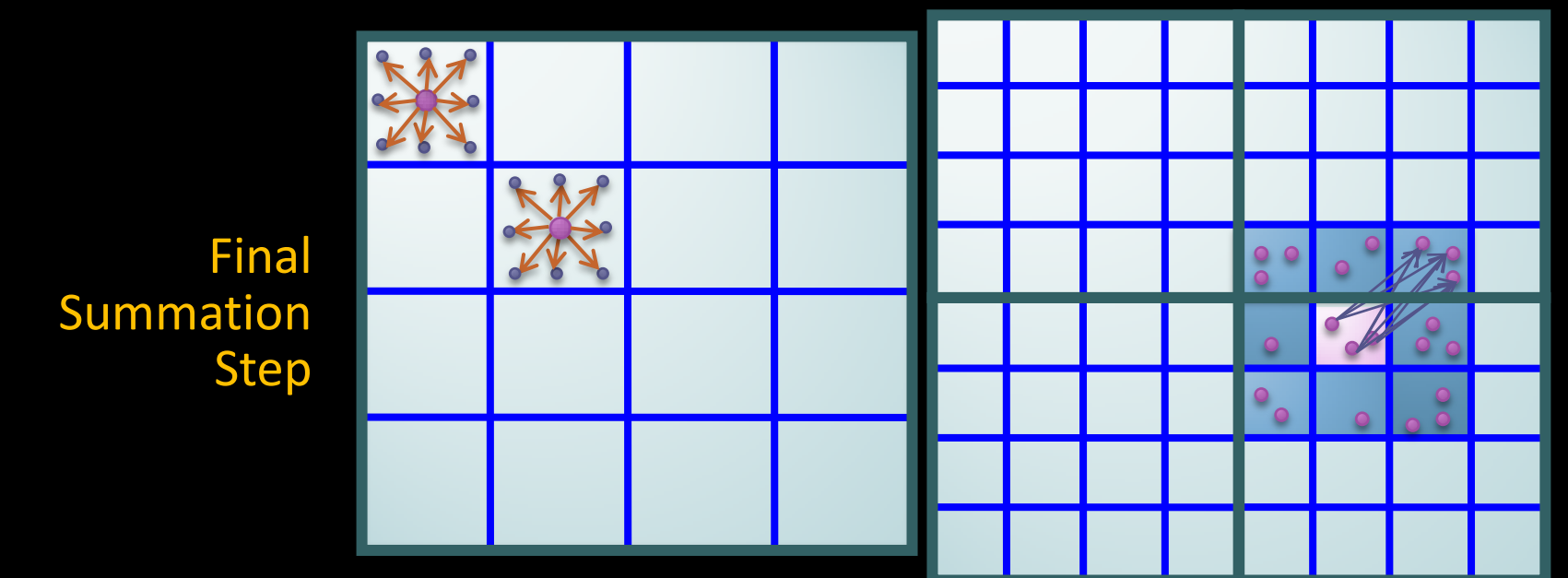
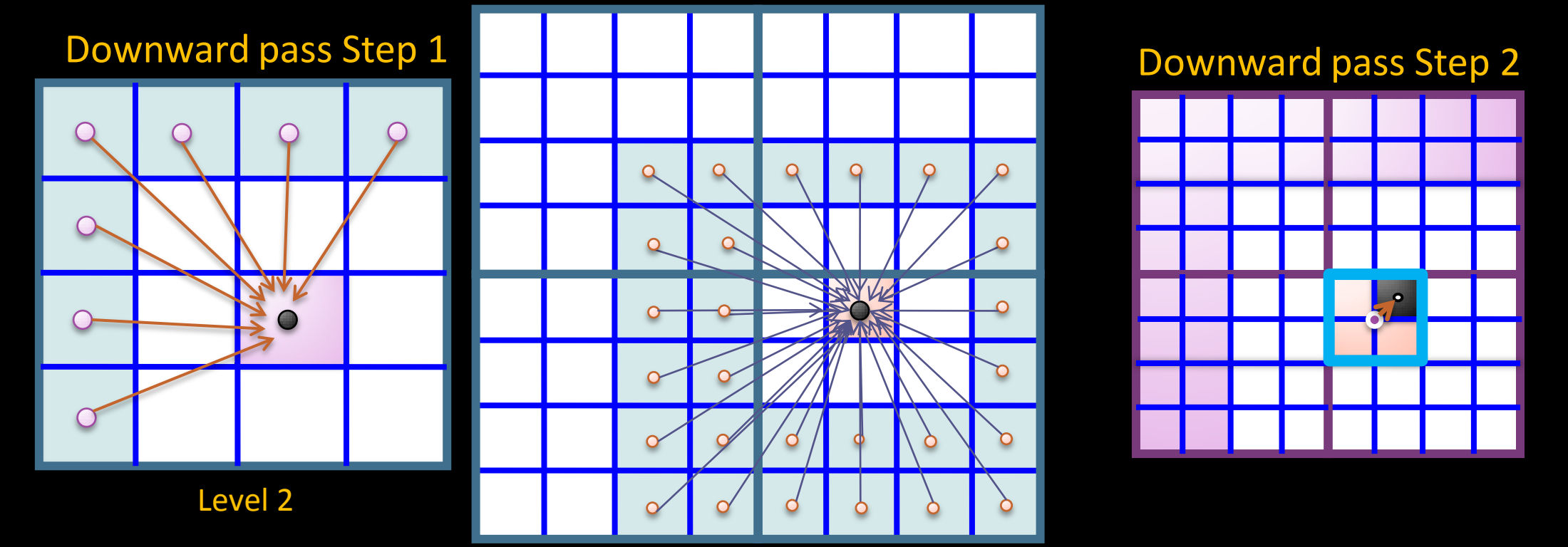
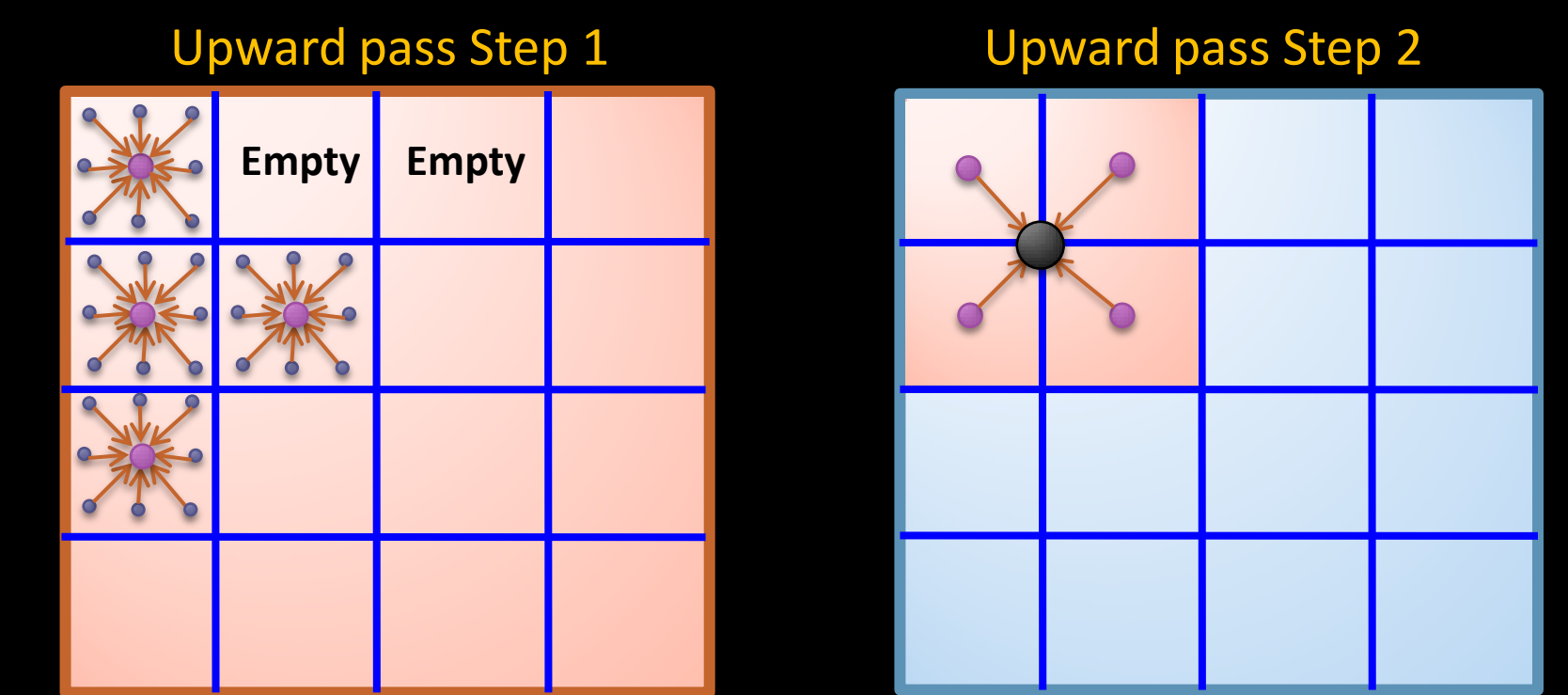
Parallel Octree Construction on GPU



Parallel V-map Construction on GPU



Parallel FMM on GPU



Qualitative Results

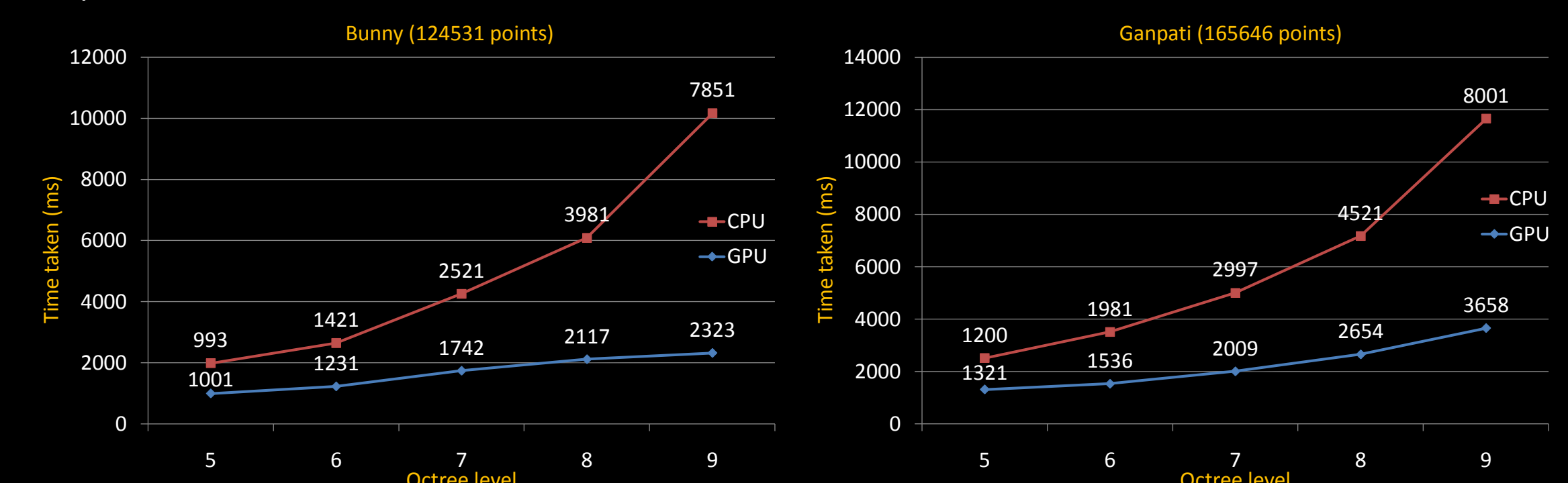


CPU GPU

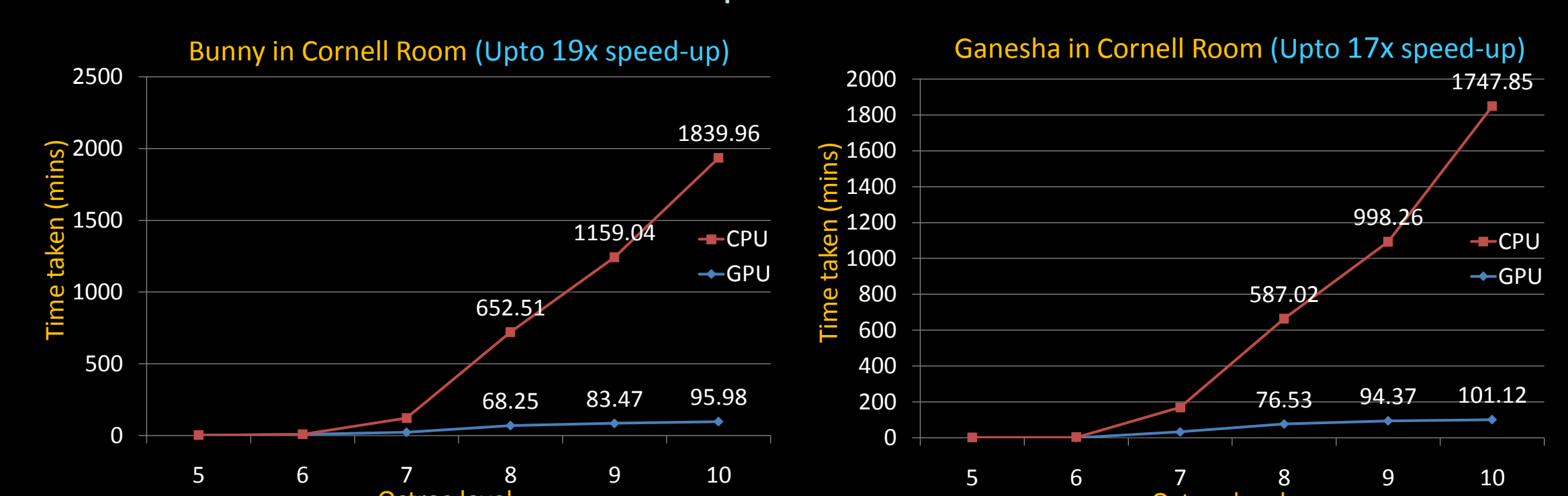


Quantitative Results

Octree Construction



V-map Construction



Phases of FMM

