Noise Inheritance

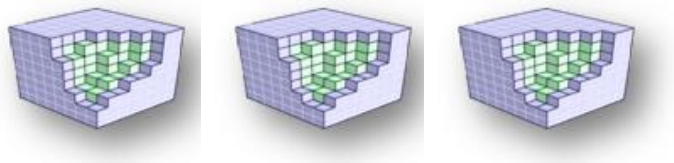
IDEA

- Noise is a sum octaves : $F(x) = Sum_i(A . noise(B . x))$
- A (amplitude = $1/2^i$) , B : frequency (2^i)

- Replace expensive sum by adding one octave at each level of resolution.
- Each level computes only one octave and asks its parent node (at corser resolution) its previously computed noise value.

Noise Inheritance

Voxel channels : < uchar4, half4, float >
=> [color, normal, noise]

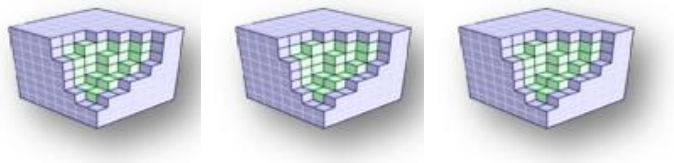


1 HOST producer → 2 produce Nodes → do nothing…

3 DEVICE producer → 4 produce Nodes → oracle says => « data everywhere »

Noise Inheritance

Voxel channels : < uchar4, half4, float >
=> [color, normal, noise]



1 HOST producer → 2 produce Bricks → do nothing…

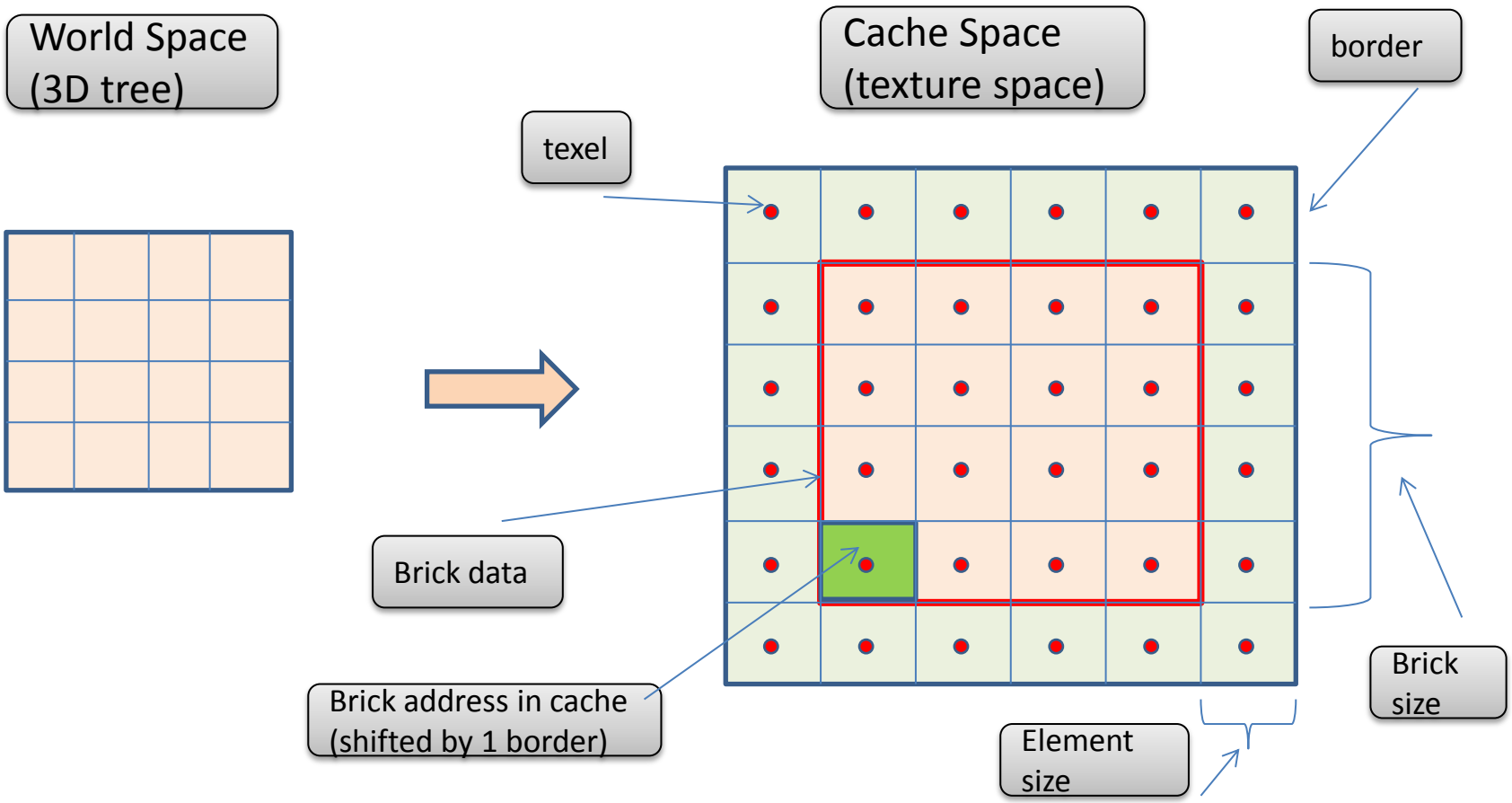
3 DEVICE producer → 4 produce Bricks →

- Compute octave at given depth
- Retrieve previous noise from parent node
- F(x) = noise + previousSumNoise
- STORE previousSumNoise in 3rd channel (#2)

Noise Inheritance

Equivalence : brick space / cache space

Brick
- 4x4 voxels
- 1 border

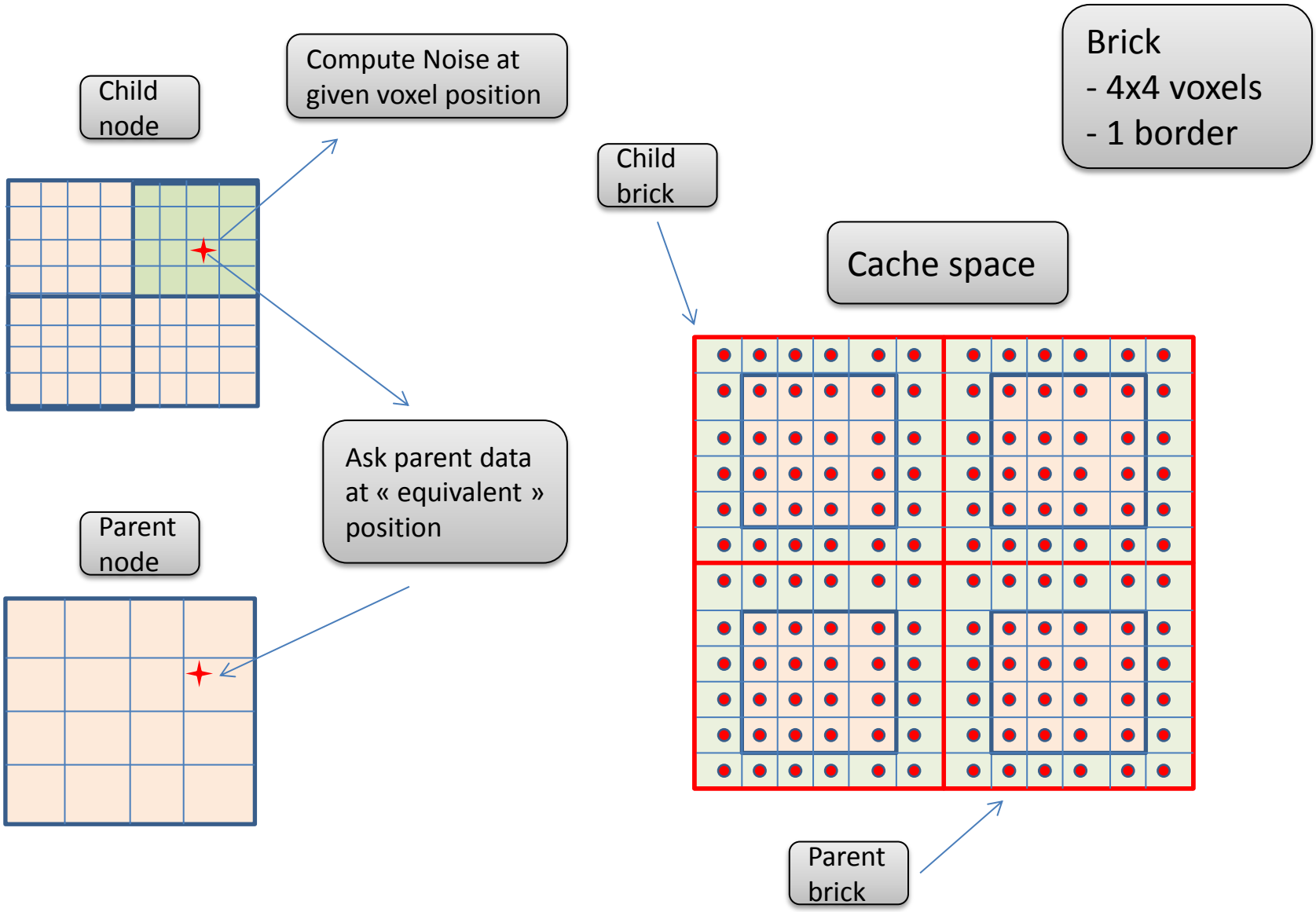


Noise Inheritance

Retrieve noise from parent node

- Use NodeVisitor to « query » parent node
- Parent node gives its brick address in cache to be able to sample data inside
- User needs to compute the « equivalent sample position » in cache (given cache parameters : element size, etc…)

Noise Inheritance



Noise Inheritance

Noise

- Sum of octaves

- $F(x) = Sum_i(A . noise(B . x))$
- A : amplitude = $1/2^i$
- B : frequency = 2^i

- $F_n(x) = F_{n-1}(x) + 1/2^n . noise(2^n . x)$
- $F_0(x) = noise(x)$

Noise Inheritance

Problems :

- Border computations ? => may require a lot of parent nodes

- Sample positions ?

- $F_n(x) = F_{n-1}(x) + 1/2^n \cdot noise(2^n \cdot x)$

- $F_{n-1}(x)$: coarser level : « x » is different => associated voxel position in parent node, because coarser value has been written at parent texel position which is different than child texel position.