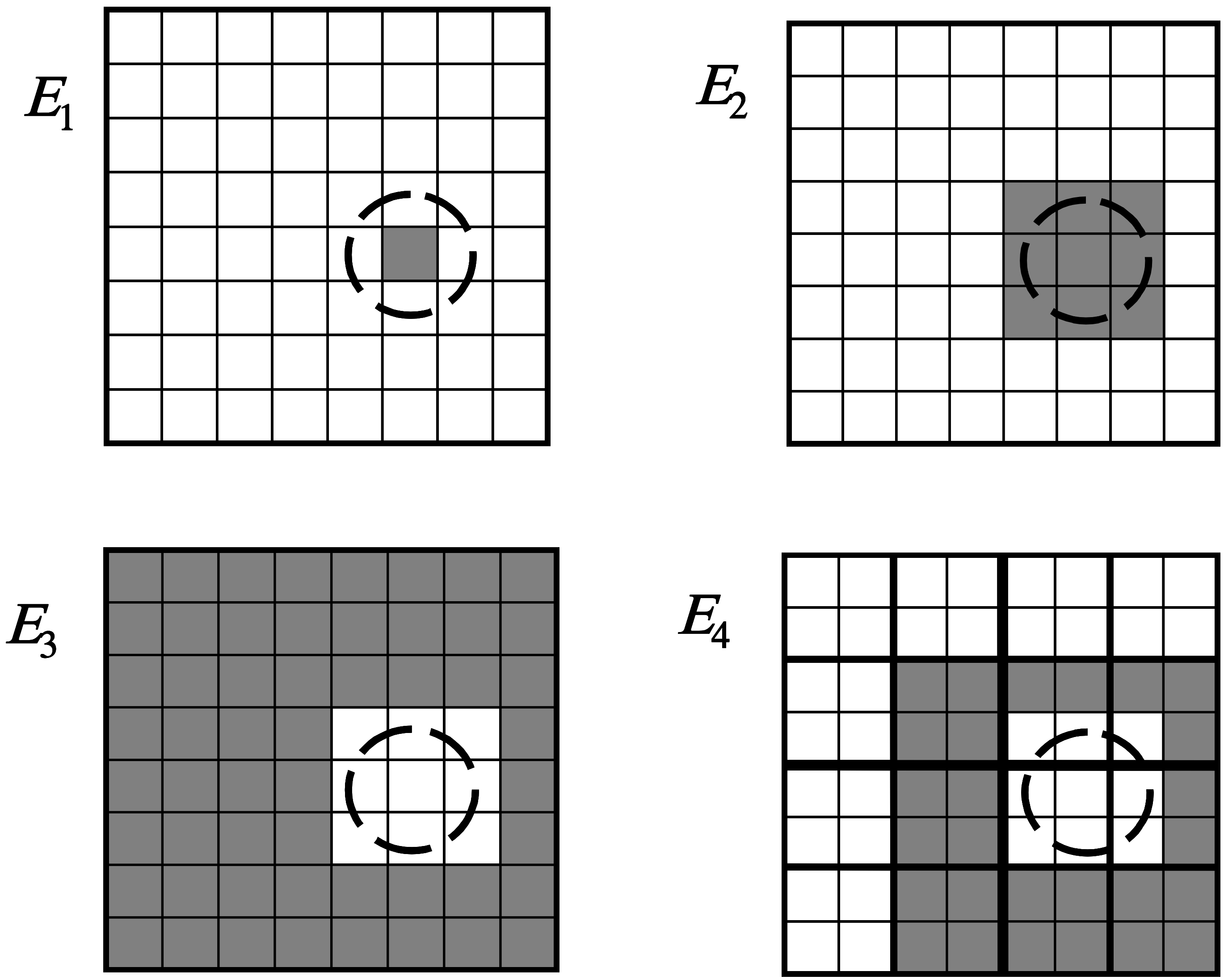
CMSC 878R/AMSC 698R
Lecture 15

Prepare Data Structures

- Convert data set into integers given some maximum number of bits allowed/dimensionality of space
- Interleave
- Sort
- Go through the list and check at what bit position two strings differ
  - For a given *s* determine the number of levels of subdivision needed
- Digression (what is the algorithm to determine the number of levels)?

Hierarchical Spatial Domains



UPWARD PASS

- Partition sources into a source hierarchy.
- Stop hierarchy so that boxes at the finest level contain *s* sources
- Let the number of levels be *L*
- Consider the finest level
- For non-empty boxes we create S expansion about center of the box $\Phi(x_i,y)=\sum^P u_i B(x_*,x_i)\, S(x_*,y)$

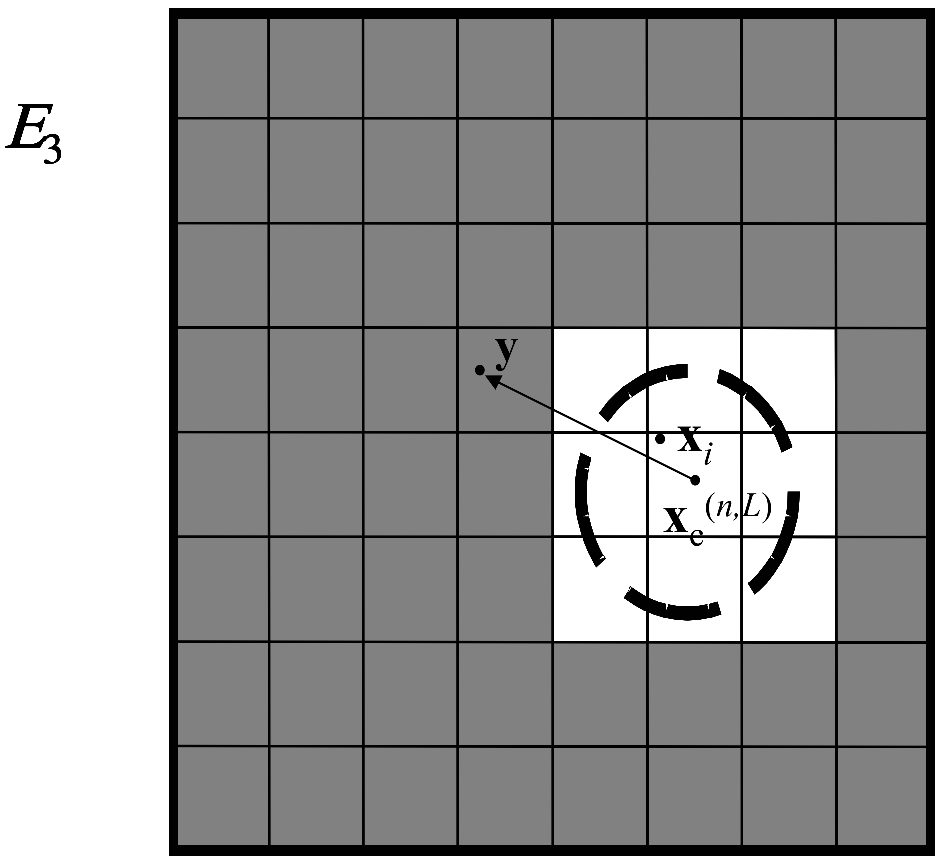
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}(\mathbf{y}-\mathbf{x}_c^{(n,L)}),$$
$$\mathbf{C}^{(n,L)} = \sum_{\mathbf{x}_i \in E_1(n,L)} u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right).$$

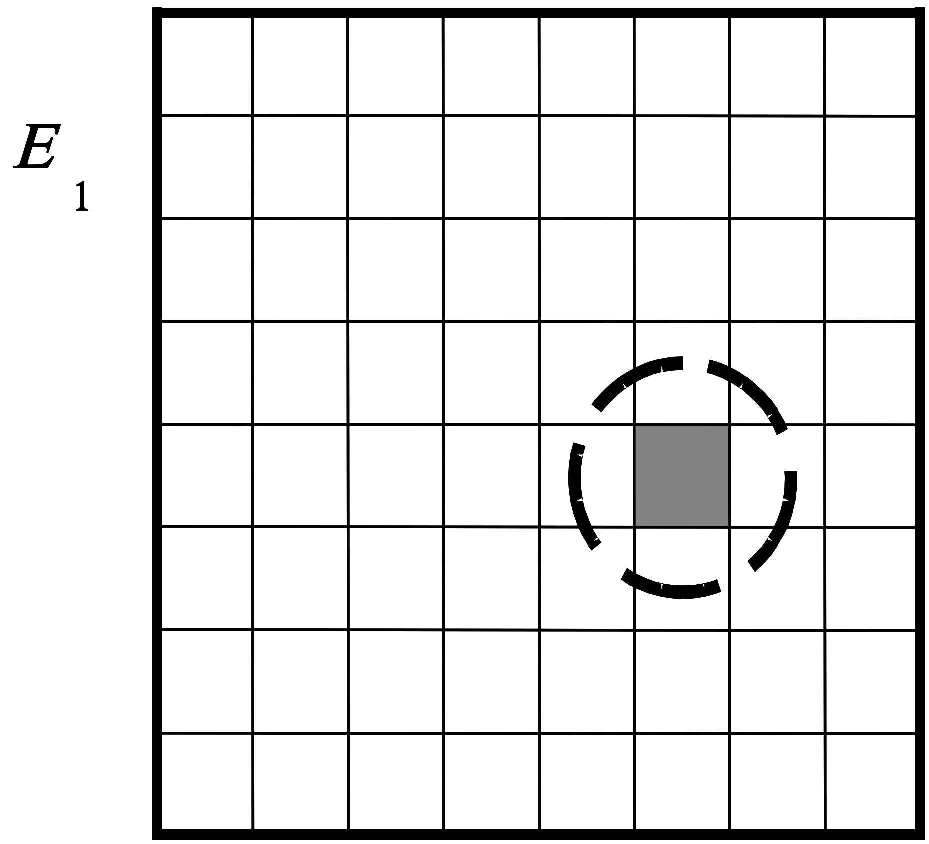
- We need to keep these coefficients. $\boldsymbol{C}^{(n,l)}$ for each level as we will need it in the downward pass
- Then use S/S translations to go up level by level up to level 2.
- Cannot go to level 1 (Why?)

UPWARD PASS

- At the end of the upward pass we have a set of S expansions (i.e. we have coefficients for them)
- we have a set of coefficients $\boldsymbol{C}^{(n,l)}$ for $n=1,\ldots,2^{ld}\quad l=L,\ldots,2$
- Each of these expansions is about a center, and is valid in some domain
- We would like to use the coarsest expansions in the downward pass (have to deal with fewest numbers of coefficients)
- But may not be able to --- because of domain of validity

- *S* expansion is valid in the domain *E_3* outside domain *E_1* (provided *d<9*)





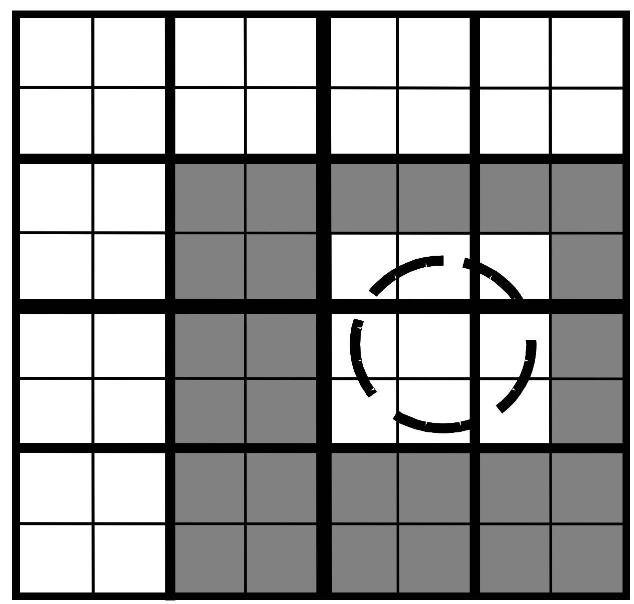
DOWNWARD PASS

- Starting from level 2, build an *R* expansion in boxes where *R* expansion is valid

$$\Phi_4^{(n,l)}(\mathbf{y}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$
$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

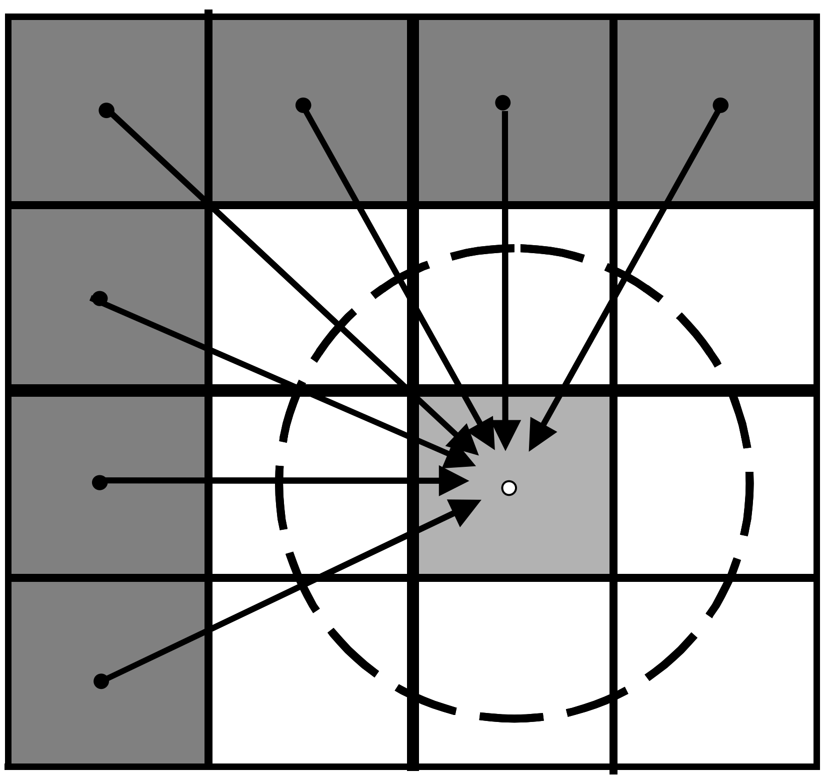
- Must to do $S|R$ translation
- The *S* expansion is not valid in boxes immediately surrounding the current box
- So we must exclude boxes in the E_4 neighborhood

E_4

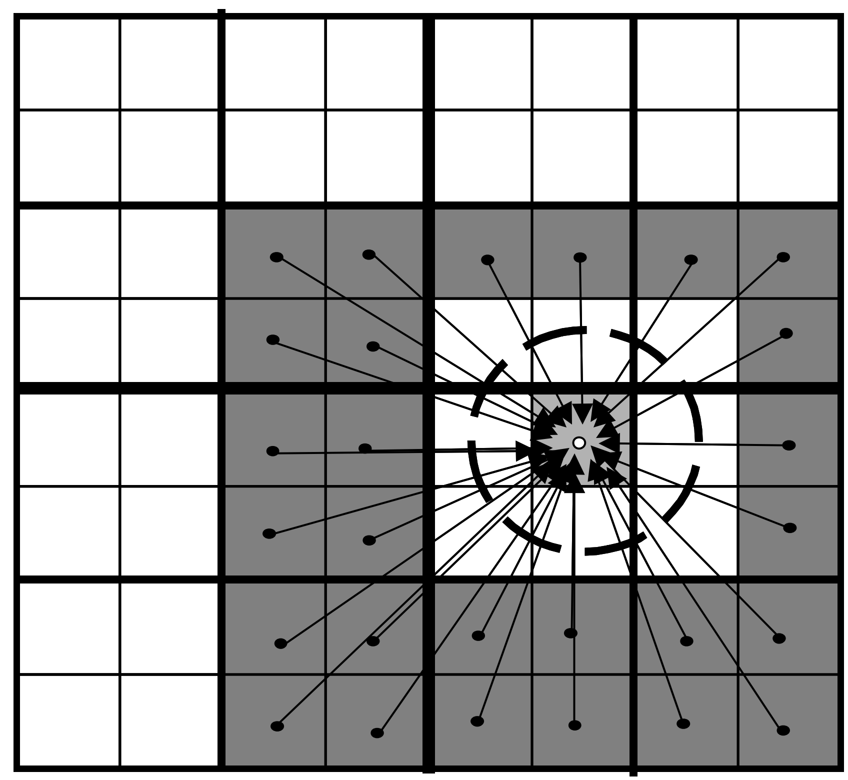


Downward Pass. Step 1.

Level 2:

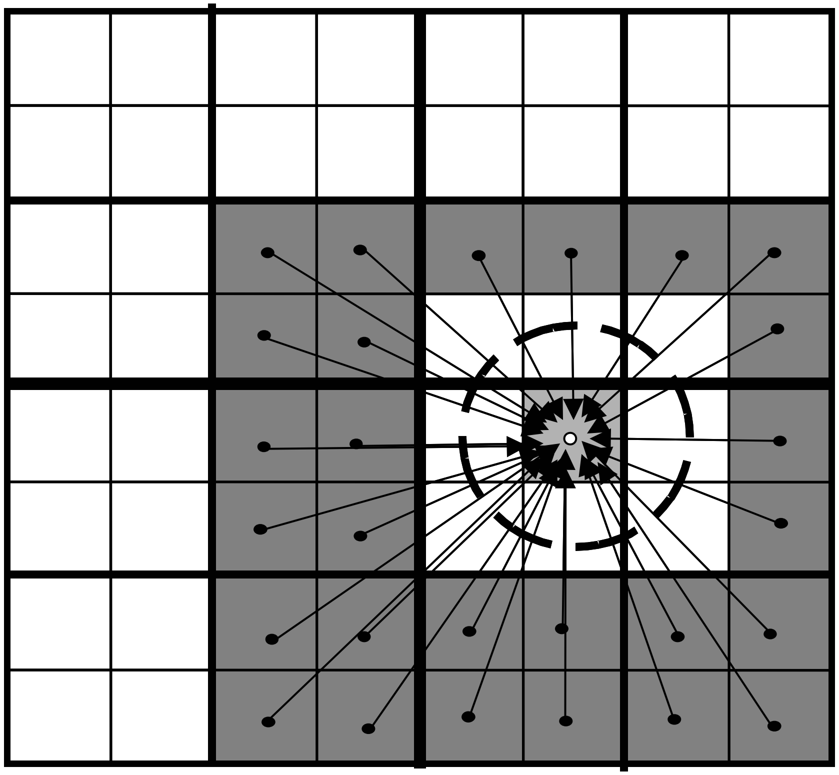


Level 3:



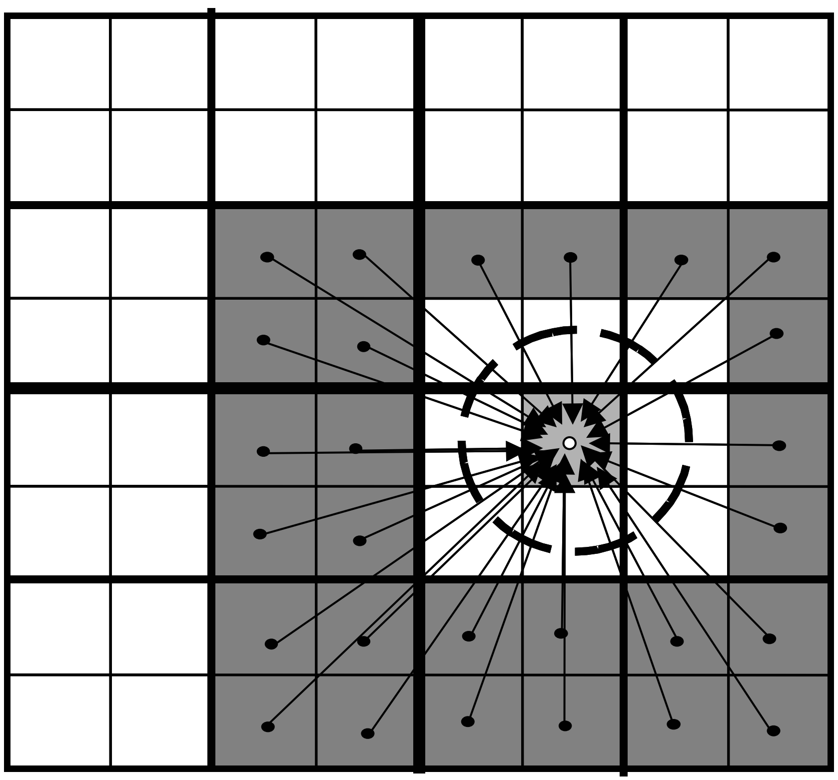
Downward Pass. Step 1.

THIS MIGHT BE
THE MOST EXPENSIVE
STEP OF THE ALGORITHM



Downward Pass. Step 1.

$$P_4 = PowerOfE_4Neighborhood = 3^d 2^d - 3^d = 3^d\left(2^d - 1\right)$$



$d = 1: \quad P_4 = 3,$

$d = 2: \quad P_4 = 27,$

$d = 3: \quad P_4 = 189,$

$d = 4: \quad P_4 = 1215,$

$\ldots$

Exponential Growth

Total number of S|R-translations per 1 box in *d*-dimensional space

(far from the domain boundaries)

Downward Pass Step 2

- Now consider we already have done the S|R translation at some level at the center of a box.
- So we have a R expansion that includes contribution of most of the points, but not of points in the E_4 neighborhood
- We can go to a finer level to include these missed points
- But we will now have to translate the already built R expansion to a box center of a child
  - (Makes no sense to do S|R again, since many S|R are consolidated in this R expansion)
- Add to this translated one, the S|R of the E_4 of the finer level

- Formally

Step 2. At $l = 2$ we have

$$\Phi_3^{(n,2)}(\mathbf{y}) = \Phi_4^{(n,2)}(\mathbf{y}), \quad \mathbf{D}^{(n,2)} = \widetilde{\mathbf{D}}^{(n,2)},$$

Form $\Phi_3^{(n,l)}(\mathbf{y})$ (or expansion coefficients of this function) by adding $\Phi_4^{(Parent(n),l-1)}(\mathbf{y})$ to (R|R)- translated coefficients of the parent box to the child center:

$$\Phi_3^{(n,l)}(\mathbf{y}) = \mathbf{D}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

$$\mathbf{D}^{(n,l)} = \widetilde{\mathbf{D}}^{(n,l)} + (\mathbf{R}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l-1)}\right)\mathbf{D}^{(m,l-1)}, \quad m = Parent(n).$$

$$\Phi_4^{(n,l)}(\mathbf{y}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

Downward Pass. Step 2.

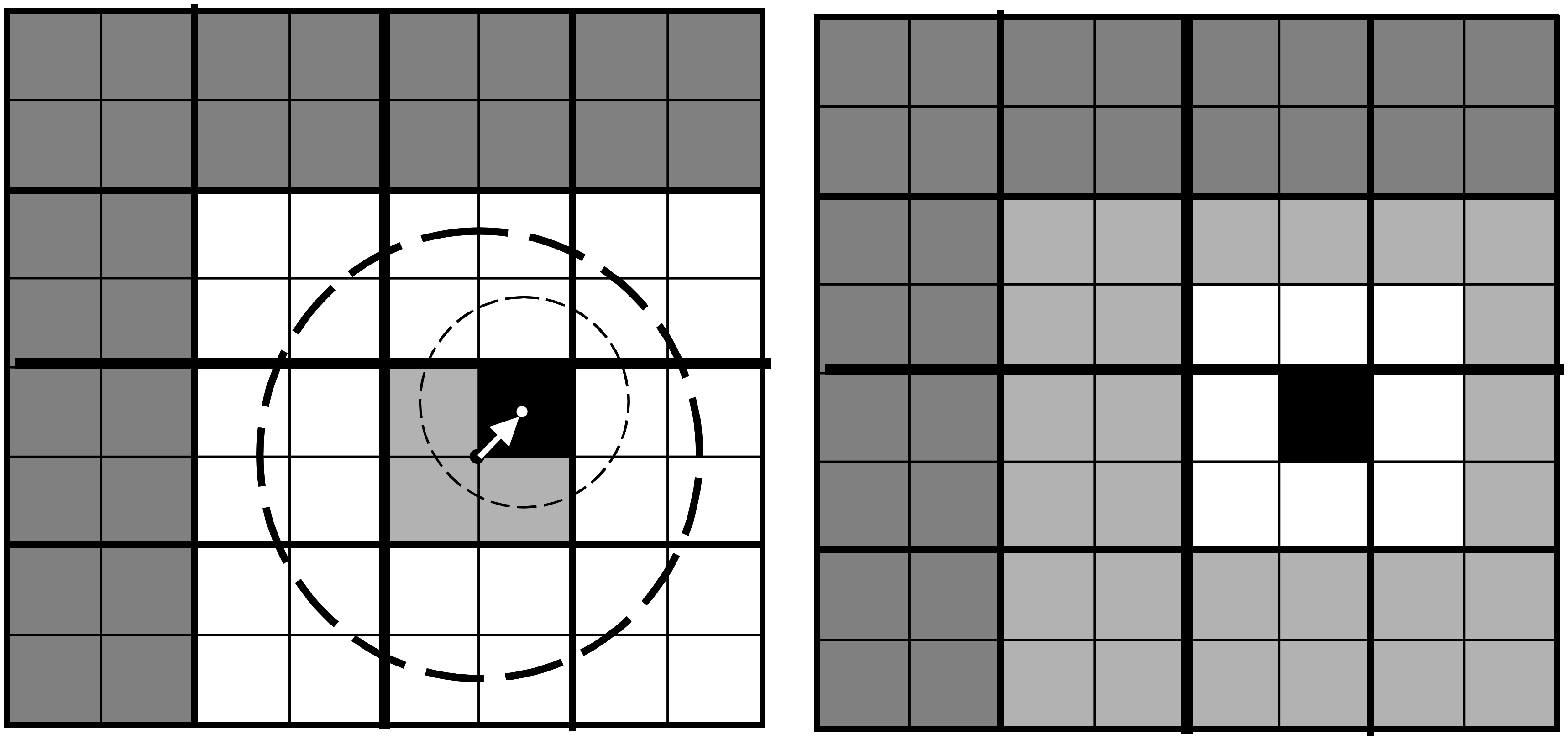


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

$$E_3(n,l+1) = E_3(n,l) \cup E_4(n,l+1).$$

Final Summation

- At this point we are at the finest level.
- We cannot do any S|R translation for x_i ‘s that are in the E_3 neighborhood of our y_j’s
- Must evaluate these directly

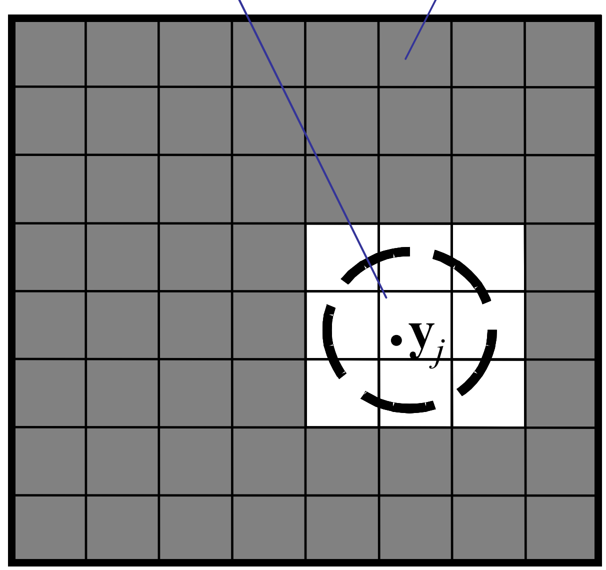
Final Summation

As soon as coefficients $\mathbf{D}^{(n,L)}$ are determined total potential can be computed for any point $\mathbf{y}_j \in E_1(0,0)$, where $\Phi_2^{(n,l)}(\mathbf{y})$ can be computed straightforward. So:

$$v_j = \Phi(\mathbf{y}_j) = \sum_{\mathbf{x}_i \in E_2(n,L)} u_i \Phi(\mathbf{y}_j, \mathbf{x}_i) + \mathbf{D}^{(n,L)} \circ \mathbf{R}(\mathbf{y}_j - \mathbf{x}_c^{(n,L)}), \quad \mathbf{y}_j \in E_1(n,L).$$

Contribution of E_2

Contribution of E_3



Cost of FMM --- Upward Pass

- Upward Step1. Cost of creating an S expansion for each source point. *O(NP)*
- Upward Step2. Cost of performing an S|S translation
  - If we use expensive (matrix vector) method cost is $O(P^2)$ for one translation.
- Step 2 is repeated from level *L-1* to level *2*

$$CostUpward_2 = 2^d\left(2^{(L-1)d} + 2^{(L-2)d} + \ldots + 2^{2d}\right)CostSS(P)$$
$$< \frac{2^d}{2^d - 1}\left(2^{Ld} - 1\right)CostSS(P) \sim \frac{N}{s}CostSS(P)$$

- Total Cost of Upward Pass $\sim NP + (N/s)\,(P^2)$

COST of MLFMM

- Cost of downward pass, step 1 is the cost of performing S|R translations at each level

$$CostDownward_1 \lesssim P_4(d)\left(2^{2d} + \ldots + 2^{Ld}\right)CostSR(P) \sim P_4(d)\frac{N}{s}CostSR(P),$$

- At the downward pass, 2nd step we have the cost of the R|R translation, and S|R translation from the E_4 neighbourhood (already accounted for above)

$$CostDownward_2 = 2^d\left(2^{2d} + \ldots + 2^{(L-1)d}\right)CostRR(P) \sim \frac{N}{s}CostRR(P),$$

- Final summation cost is $CostEvaluation = M(P_2(d)sCostFunc + P).$

- Total

$$CostMLFMM = (M+N)P + (P_4(d)+2)\frac{N}{s}CostTranslation(P) + P_2(d)sMCostFunc$$