The Multilevel Fast Multipole Method

Ramani Duraiswami

Nail Gumerov

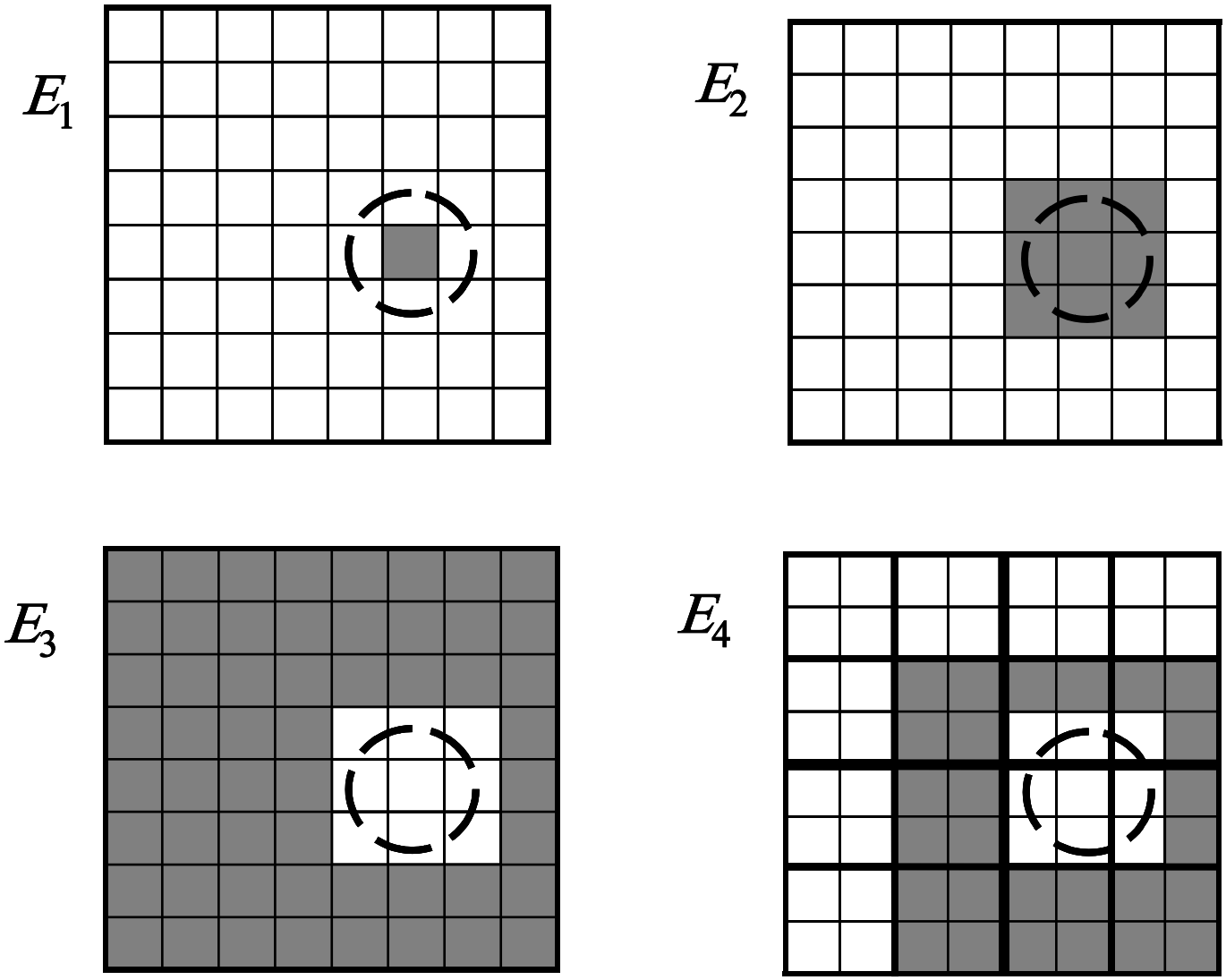
Hierarchical Spatial Domains

E_1: box

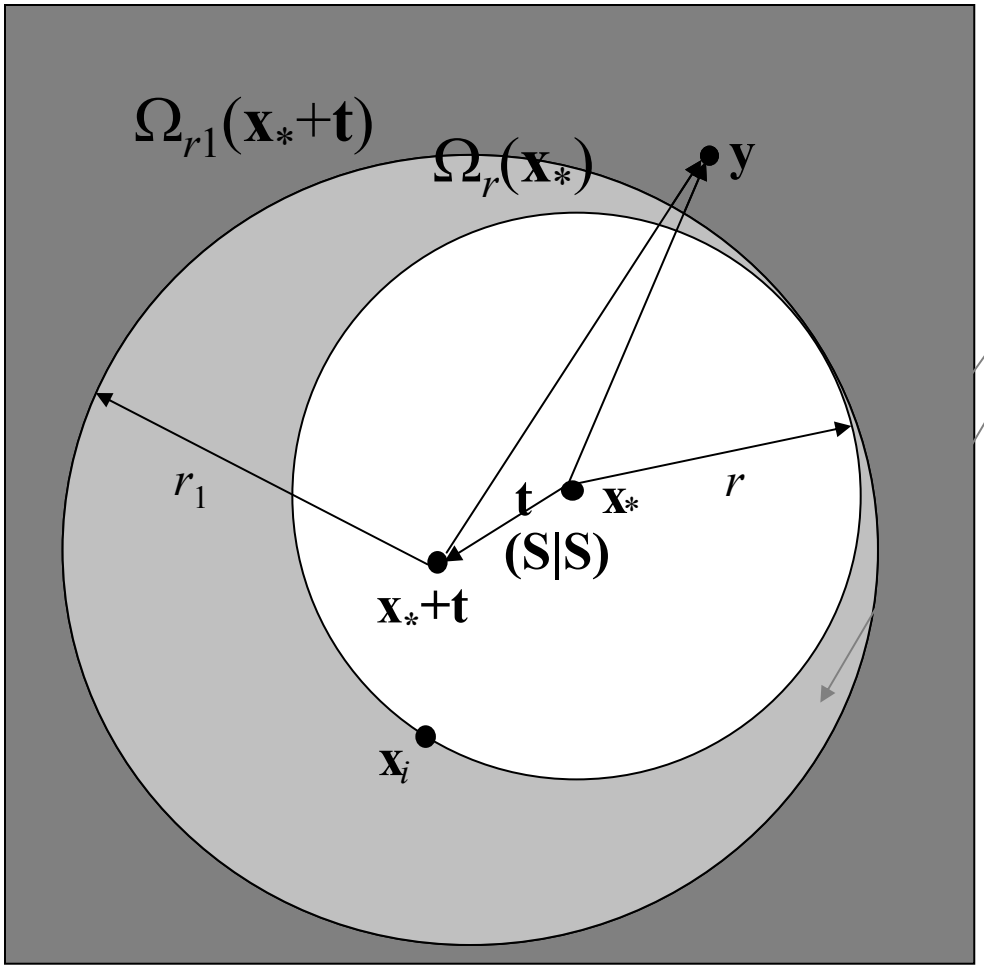
E_2: points in the box and in neighboring boxes

E_3: points in boxes outside neighborhood

E_4: points belonging to neighbors of parent box, but which do not belong to E_2



S|S-reexpansion (Far to Far, or Multipole to Multipole, or M2M)



Original expansion
Is valid only here!

$$|\mathbf{y} - \mathbf{x}_* - \mathbf{t}| > r_1 = r + |\mathbf{t}|$$

Since
$\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

Also

$$|\mathbf{x}_i - \mathbf{x}_*| < r$$

singular point !

UPWARD PASS

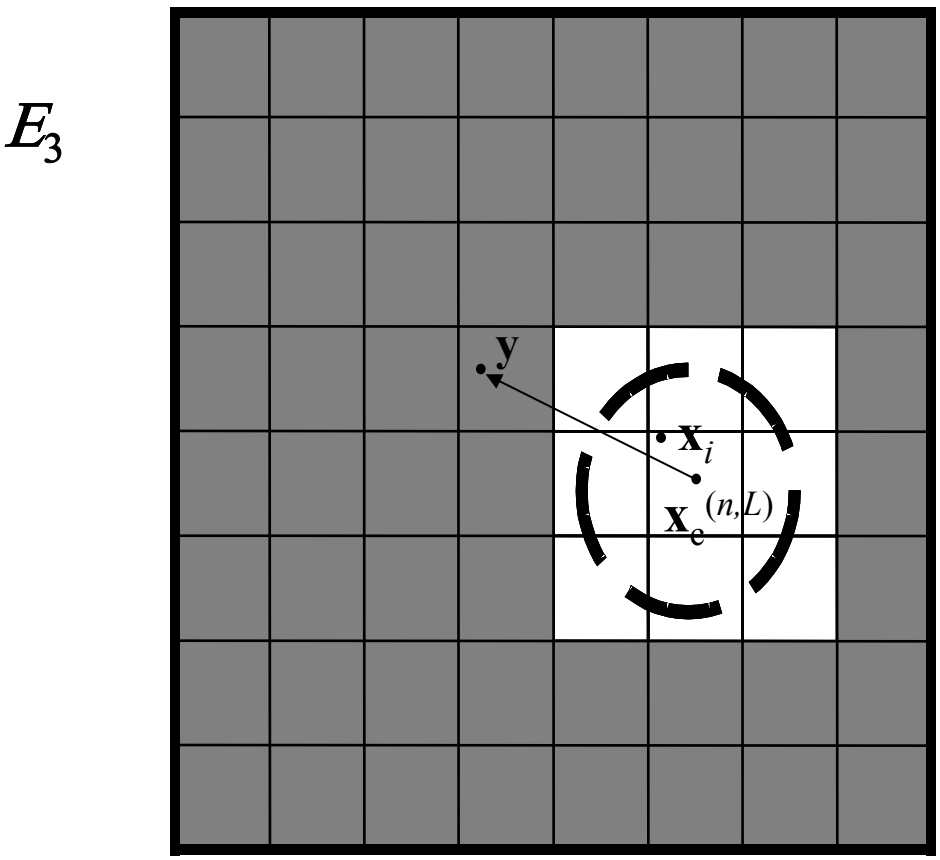
- Partition sources into a source hierarchy.
- Stop hierarchy so that boxes at the finest level contain at most *s* sources
- Let the number of levels be *L*
- Consider the finest level
- For non-empty boxes we create *S* expansion about center of the box $\Phi(x_i,y)=\sum^P u_i B(x_*,x_i)\, S(x_*,y)$

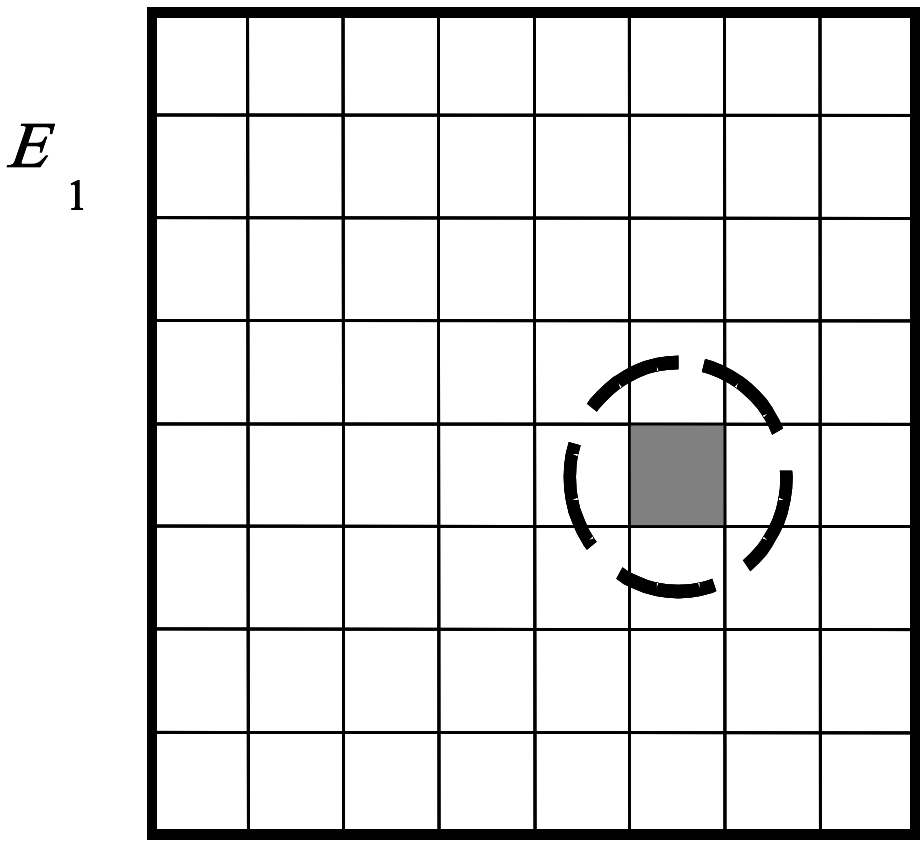
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,L)}),$$

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

- 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?)

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





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
- Upward pass works on source points and builds representations to be used in the downward pass, where the actual product will be evaluated

DOWNWARD PASS

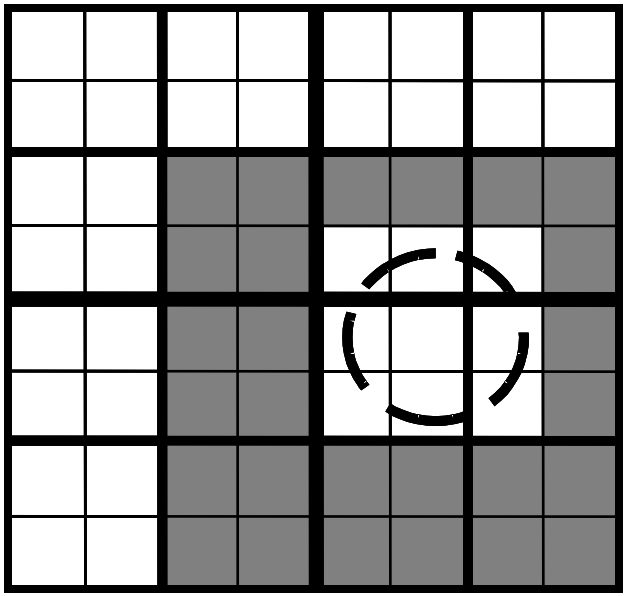
- Starting from level 2, build an R expansion in boxes where R expansion is valid

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

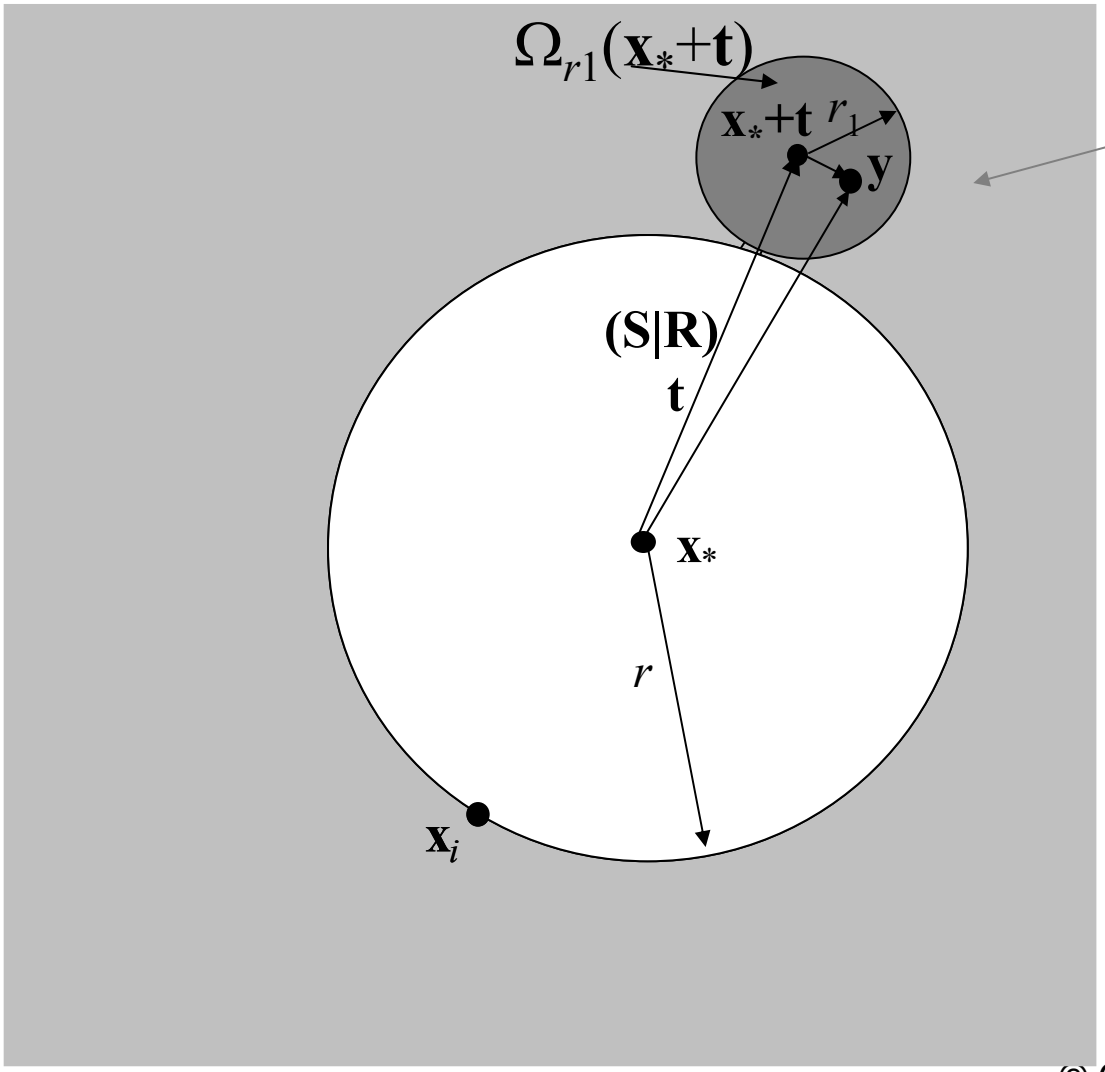
$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

- 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



S|R-reexpansion (Far to Local, or Multipole to Local, or M2L)



Original expansion
Is valid only here!

$$|\mathbf{y} - \mathbf{x}_* - \mathbf{t}| < r_1 = |\mathbf{t}| - r$$

Since
$\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

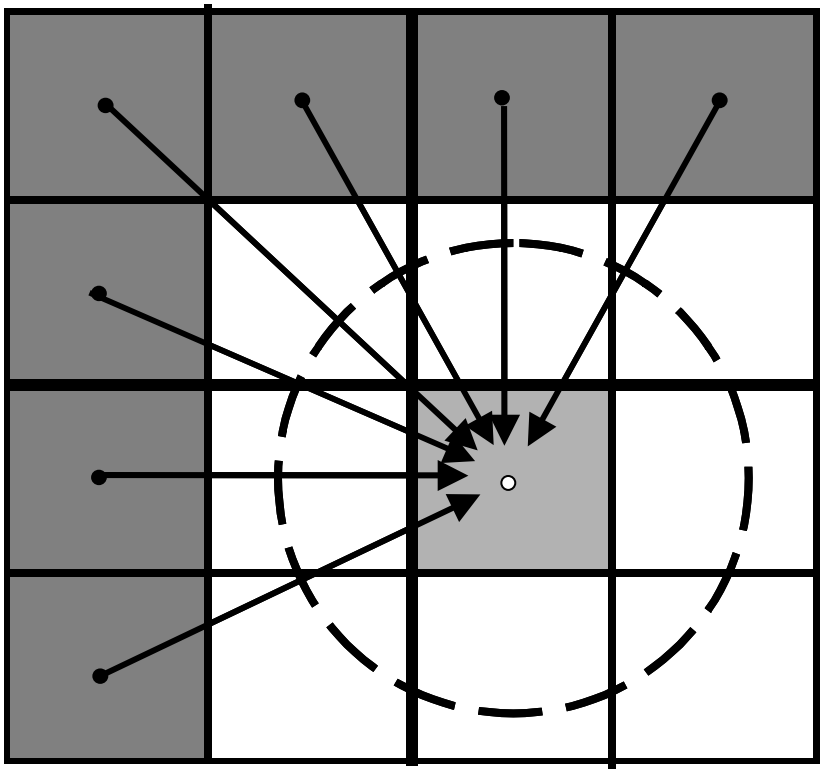
Also

$$|\mathbf{x}_i - \mathbf{x}_*| < r$$

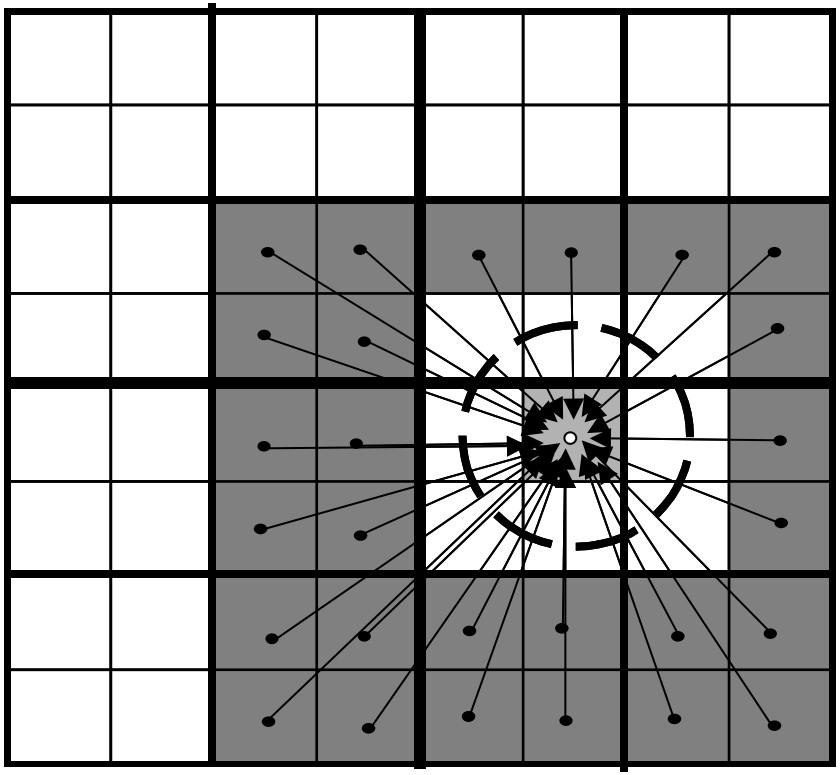
singular point !

Downward Pass. Step 1.

Level 2:

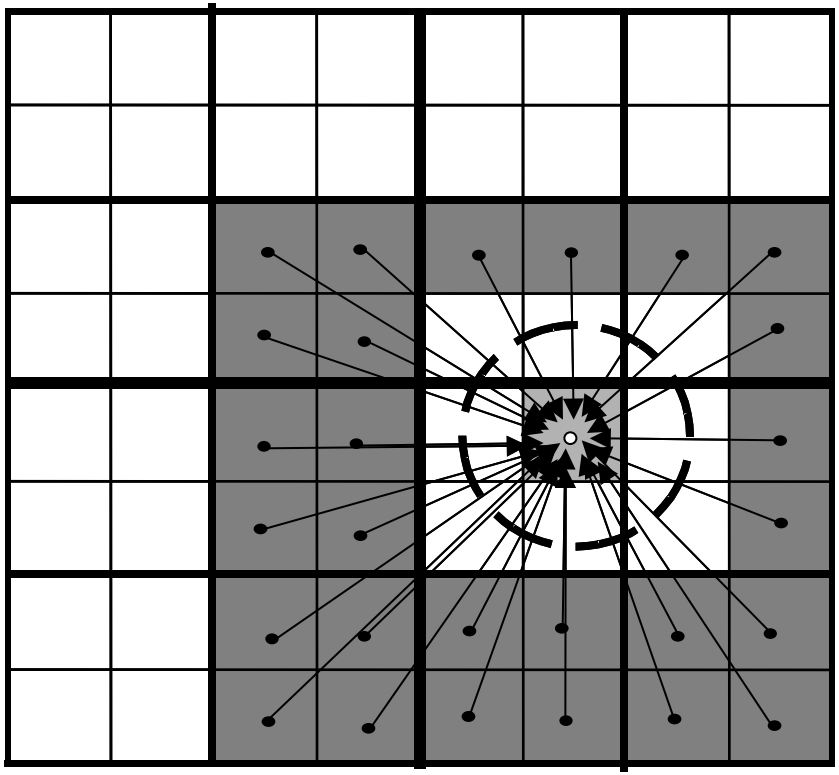


Level 3:



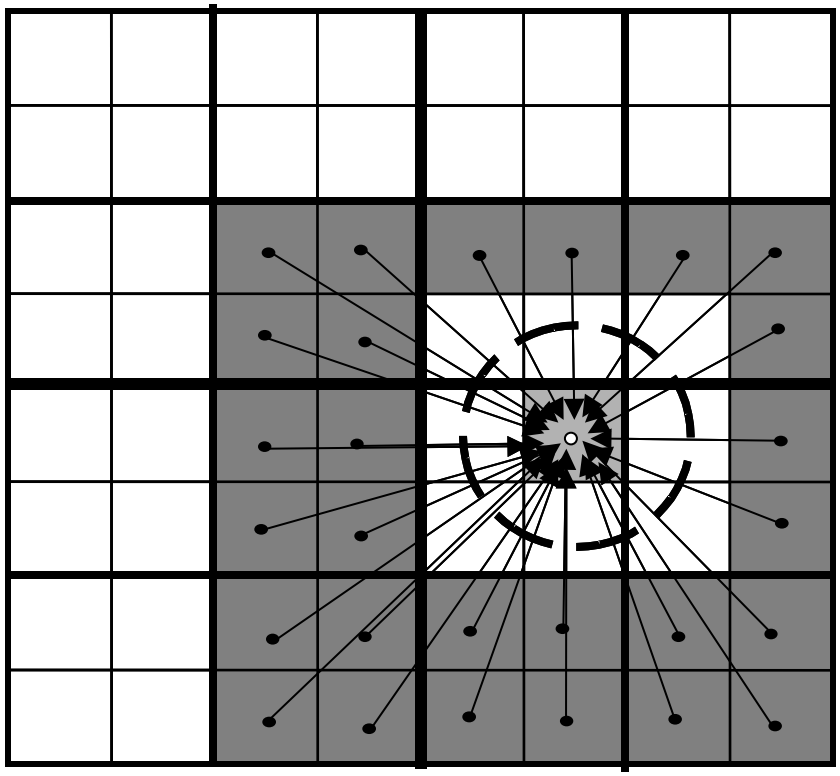
Downward Pass. Step 1.

THIS MIGHT BE
THE MOST EXPENSIVE
STEP OF THE ALGORITHM



Downward Pass. Step 1.

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



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

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

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

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

...

Exponential Growth

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

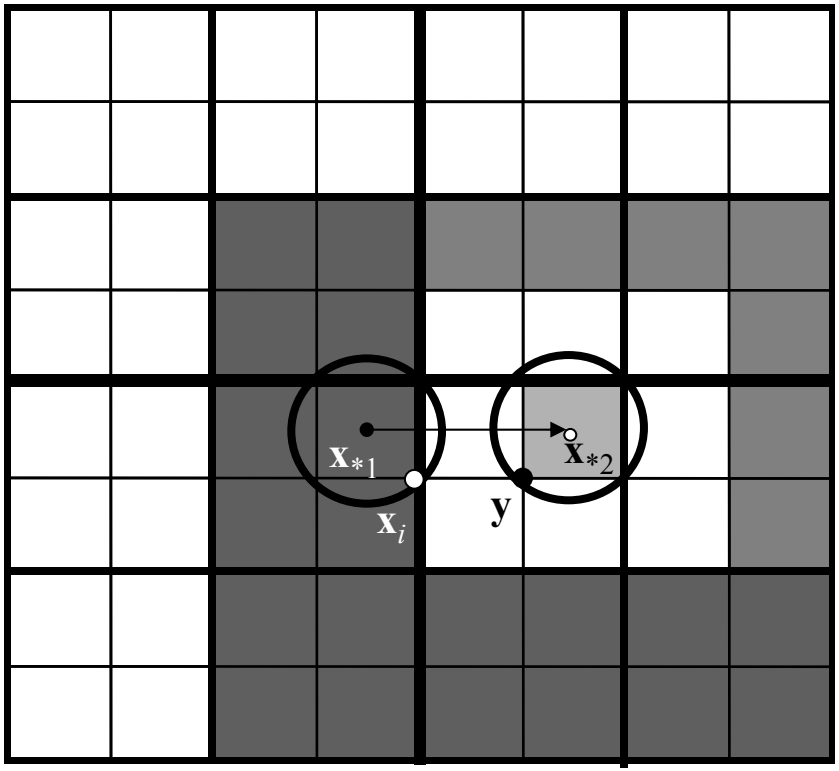
(far from the domain boundaries)

Domains of Expansion Validity (6). S|R-translation.

- S-expansion coefficients can be $S|R$-translated (converted to R-expansion coefficients)

$$|\mathbf{y} - \mathbf{x}_{*2}| < |\mathbf{x}_{*1} - \mathbf{x}_{*2}| - |\mathbf{x}_i - \mathbf{x}_{*1}|,$$

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{S}|\mathbf{R})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*1})$$



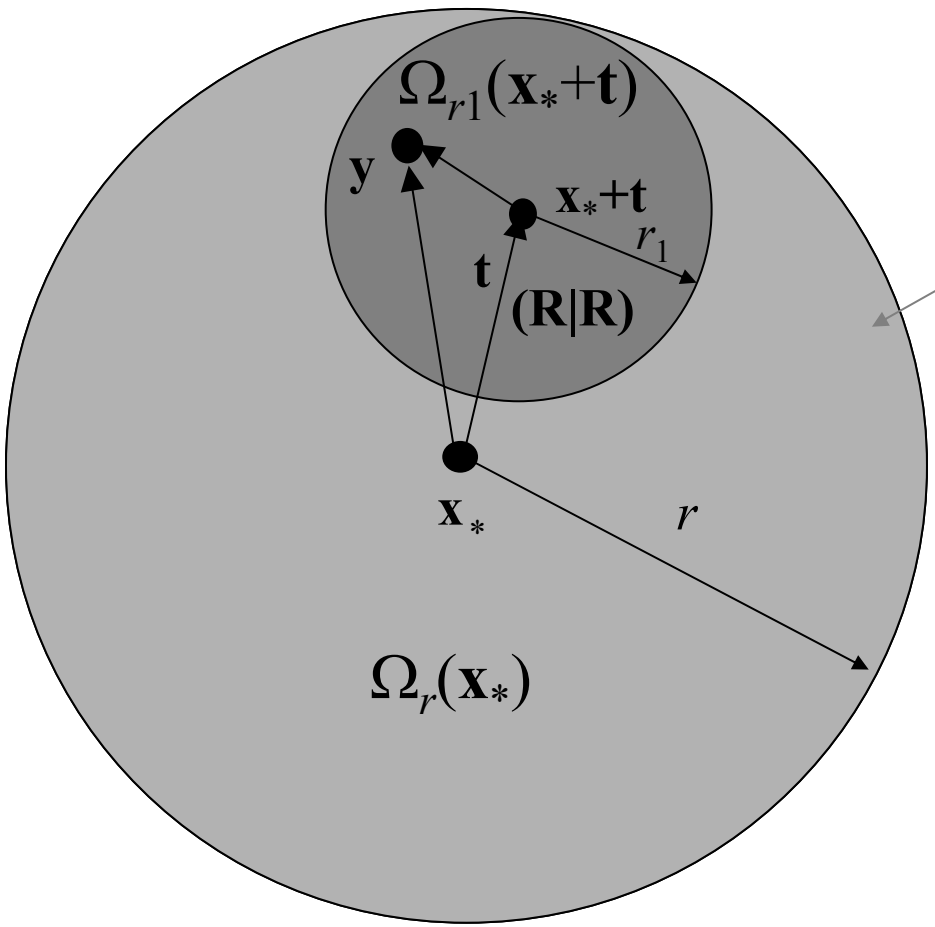
$$|\mathbf{y} - \mathbf{x}_*| < r_{\min}(l), \quad |\mathbf{x}_i - \mathbf{x}_*| < r_{\min}(l),$$

$$\min|\mathbf{x}_{*1} - \mathbf{x}_{*2}| = 2size(l).$$

$$2^{-l-1}\sqrt{d} + 2^{-l-1}\sqrt{d} < 2^{-l+1}, \quad d < 4.$$

$$d < 4$$

R|R-reexpansion (Local to Local, or L2L)



Original expansion
Is valid only here!

$$|\mathbf{y} - \mathbf{x}_* - \mathbf{t}| < r_1 = r - |\mathbf{t}|$$

Since $\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

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 $(\mathbf{R}|\mathbf{R})$- 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.

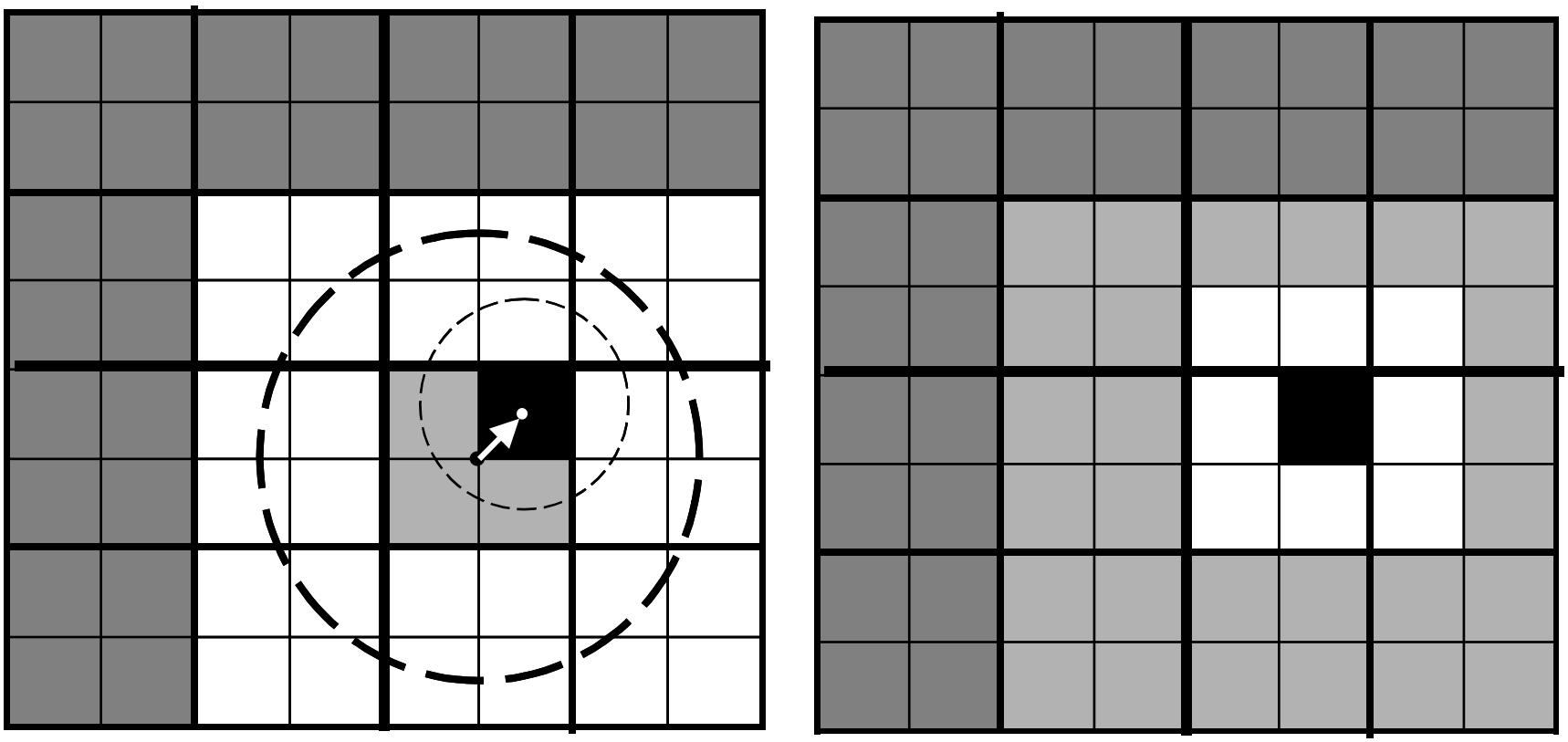


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

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

Domains of Expansion Validity (5). R|R and S|S-translations.

- R-expansion coefficients can be $R|R$-translated:

$$|\mathbf{y} - \mathbf{x}_{*2}| < |\mathbf{x}_i - \mathbf{x}_{*1}| - |\mathbf{x}_{*1} - \mathbf{x}_{*2}| :$$

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{R}|\mathbf{R})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*1})$$

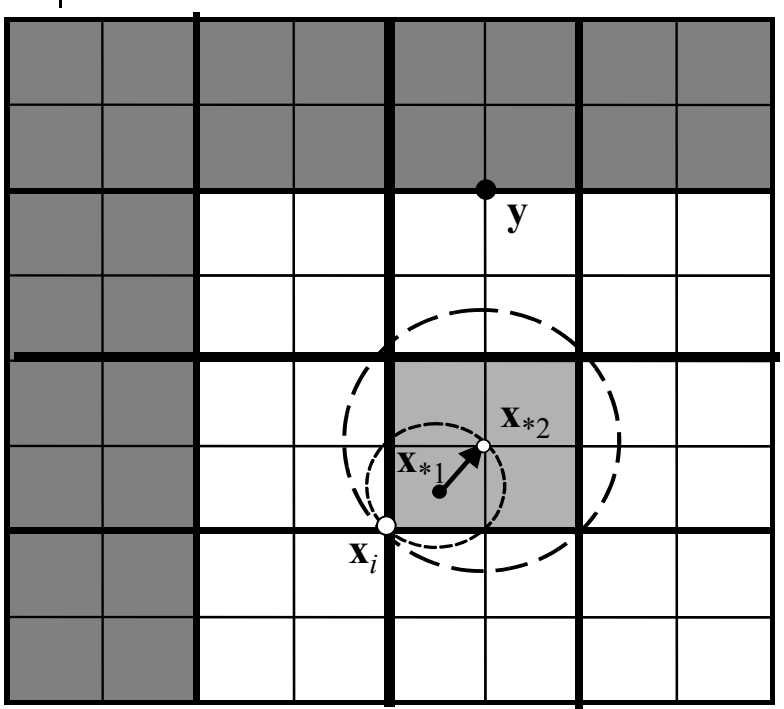
- S-expansion coefficients can be $S|S$-translated:

$$|\mathbf{y} - \mathbf{x}_{*2}| > |\mathbf{x}_{*1} - \mathbf{x}_{*2}| + |\mathbf{x}_i - \mathbf{x}_{*1}|,$$

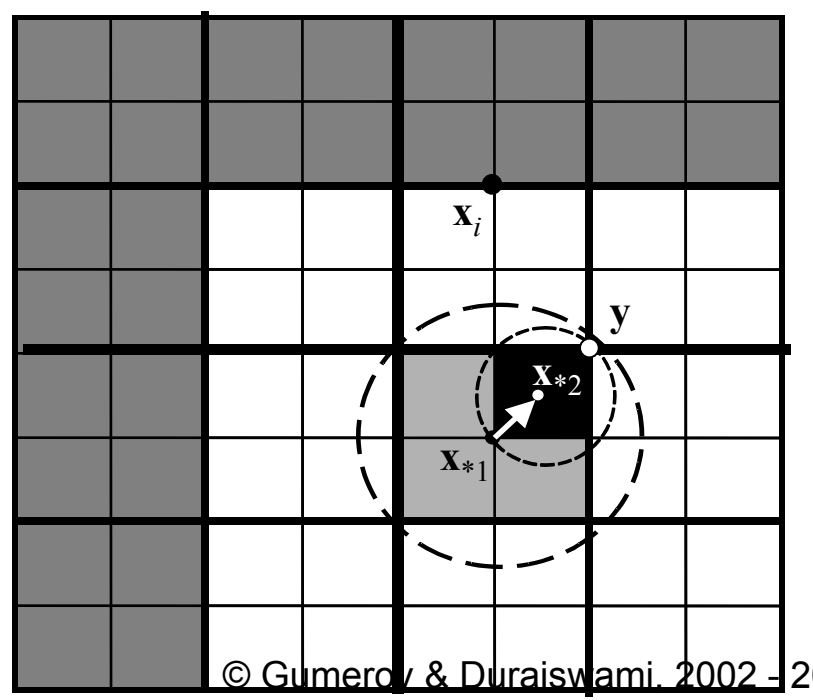
$$\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{S}|\mathbf{S})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*1})$$

Not as restrictive as S|R

S|S



R|R



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

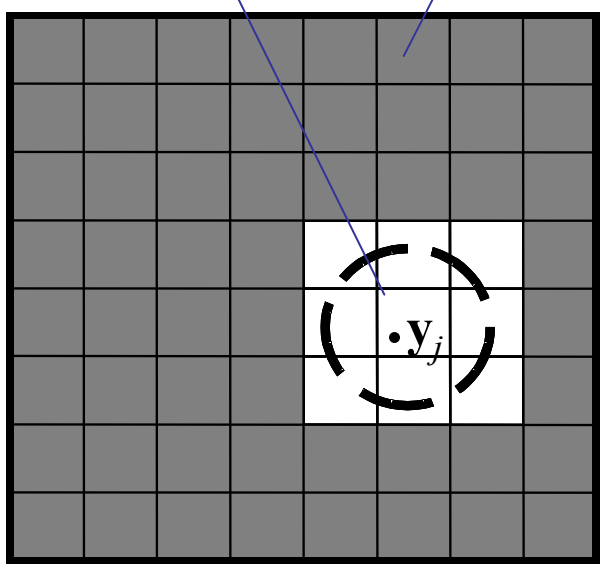
Final Summation

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

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

Itemized Cost of MLFMM

Regular mesh:

$$N = 2^{L_* d}, \quad s = 2^{L_s d}, \quad L = L_{\max} = L_* - L_s.$$

Assume that all translation costs are the same, $CostTranslation(P)$

$$CostUpward_1 = NCostExpansion(P) = O(NP).$$

$$CostUpward_2 = 2^d \left(2^{(L-1)d} + 2^{(L-2)d} + \ldots + 2^{2d}\right) CostSS(P)$$

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

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

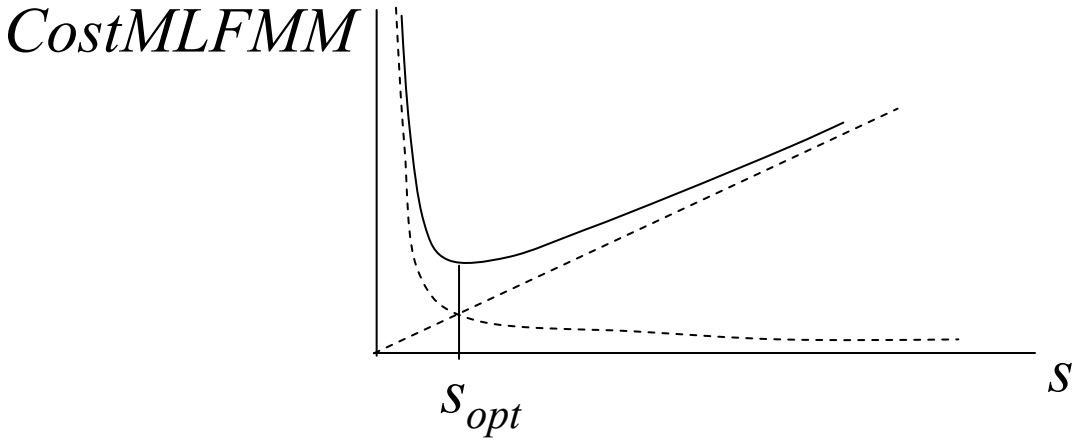
$$CostDownward_2 = 2^d \left(2^{2d} + \ldots + 2^{(L-1)d}\right) CostRR(P) \sim \frac{N}{s} CostRR(P),$$

$$CostEvaluation = M(P_2(d) s CostFunc + P).$$

Powers of E_4 and E_2 neighborhoods

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

Optimization of the Grouping Parameter



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

$$\frac{\partial CostMLFMM}{\partial s} = -(P_4(d)+2)\frac{N}{s^2}CostTranslation(P) + P_2(d)MCostFunc = 0$$

$$s_{opt} = \left[\frac{N(P_4(d)+2)CostTranslation(P)}{MP_2(d)CostFunc}\right]^{1/2}.$$

$$CostMLFMM_{opt} = (M+N)P + 2[MN(P_4(d)+2)P_2(d)CostTranslation(P)CostFunc]^{1/2}.$$

Optimization of the Grouping Parameter (Example)

$$s_{opt} = \left[\frac{N(P_4(d) + 2)CostTranslation(P)}{MP_2(d)CostFunc} \right]^{1/2}.$$

$$CostMLFMM_{opt} = (M+N)P + 2[MN(P_4(d) + 2)P_2(d)CostTranslation(P)CostFunc]^{1/2}.$$

Example:

$$N = M, \quad P_4(d) = 3^d\left(2^d - 1\right), \quad P_2(d) = 3^d,$$

$$CostTranslation(P) = P^2, \quad CostFunc = 1$$

$$s_{opt} \sim 2^{d/2}P; \quad CostMLFMM_{opt} \sim 2NP\left(1 + 3^d 2^{d/2}\right)$$

$$For\ d = 2, \quad P = 10, \quad s_{opt} \sim 38, \quad CostMLFMM_{opt} \sim 38NP = 380N.$$

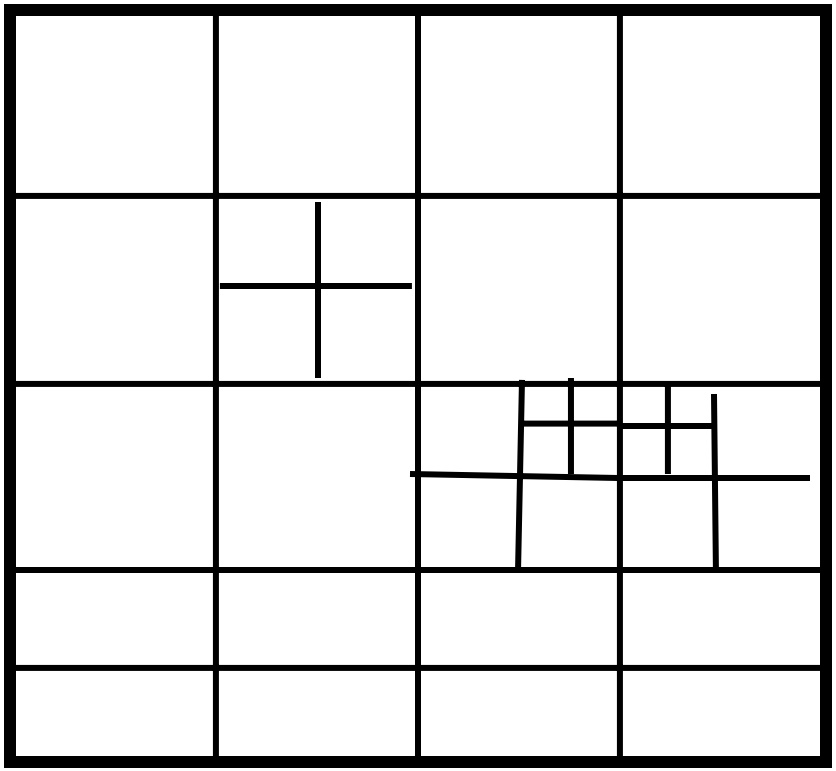
If non-optimized,

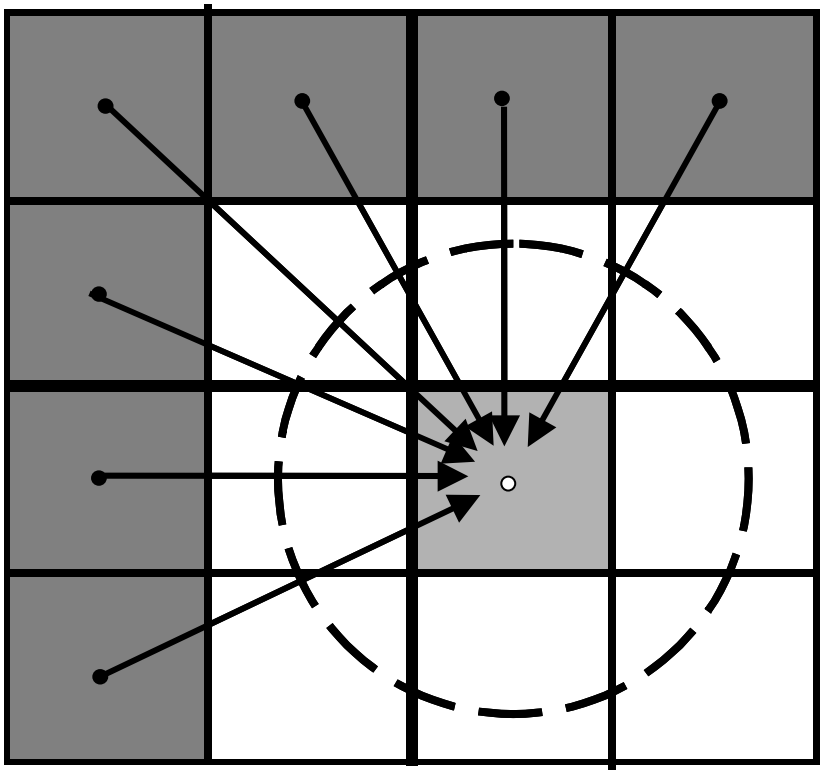
$$s = 1; \quad CostMLFMM_{opt} \sim NP\left(2 + 3^d 2^d P\right)$$

$$For\ d = 2, \quad P = 10, \quad s = 1, \quad CostMLFMM_{opt} \sim 360NP = 3600N.$$

In this example optimization results in about 10 times savings!

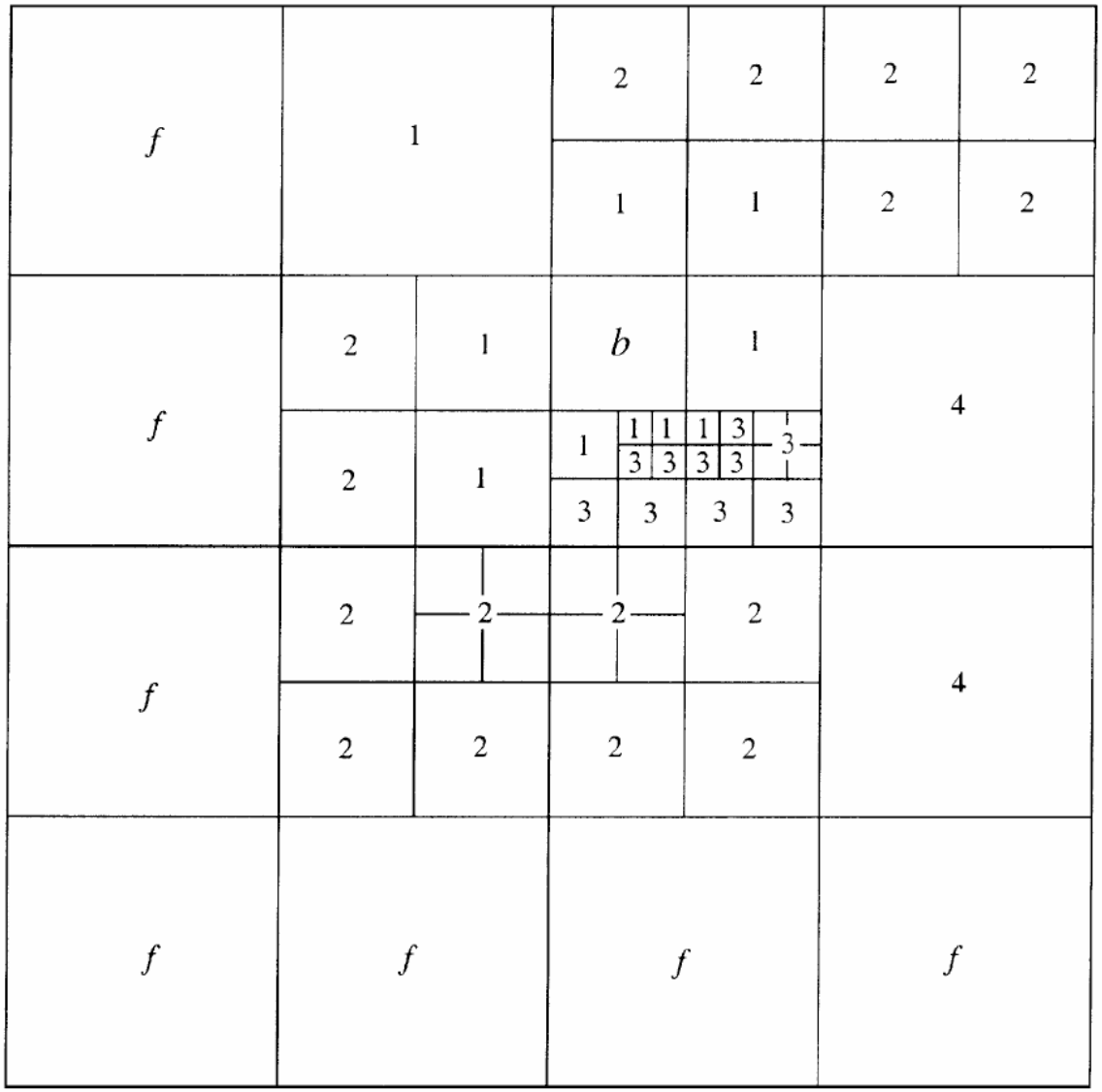
Adaptivity

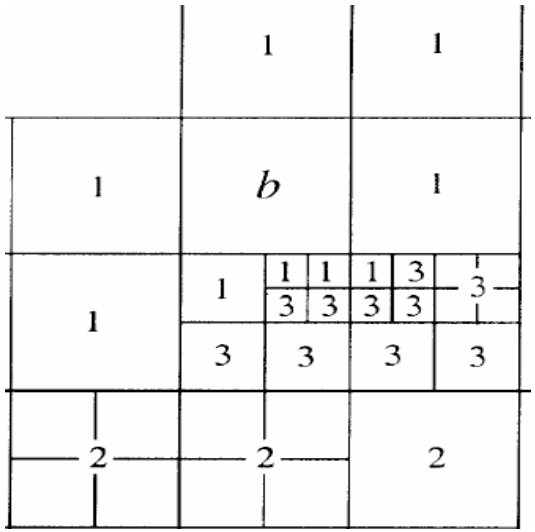


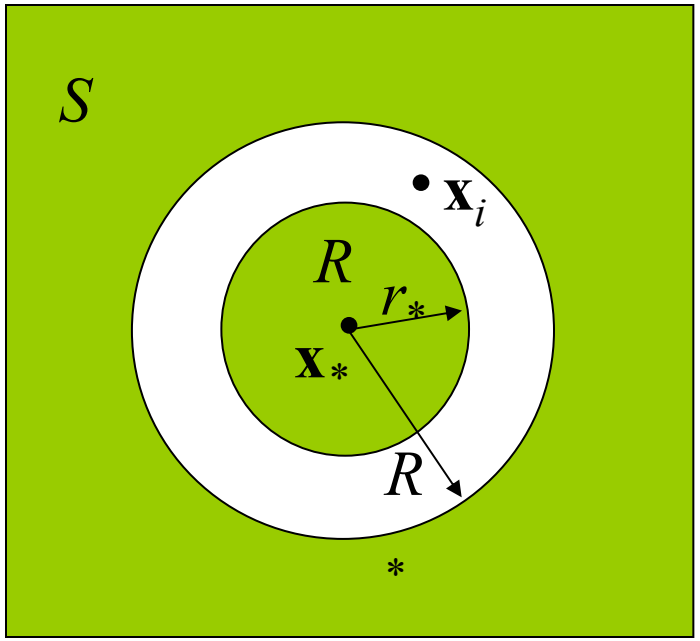


- Parameter s is determined by the box with most particles.
  - Most boxes will be empty (ok)
  - or have much fewer than s particles
- Reducing number of levels is good
- Alternate strategy
  - Continue subdivision so that each parent box has $> \ell$ points and each leaf-box has $\leq \ell$ points
- Consider an algorithm due to Cheng et al (J. Comput. Phys. 1999).

- At the finest level consider evaluation at box *b*.
- It is surrounded by boxes
  - Coarser, Same or Finer level
  - Already considered, in a coarser S|R translation
- Divide surrounding boxes not already considered in to L_1, L_2, L_3, L_4 domains
  - Boxes L_1 share a boundary with box *b*, or they lie within the sphere of box *b* and need to be evaluated directly.
  - Boxes L_2 are separated by at least one box of size of *b* S|R translate these
  - Boxes in L_3 contain sources that lie outside the sphere of b and are closer than boxes L_2 R-preFMM these
  - Box *b* lies outside the sphere of boxes in L_4. but is too close to S|R. S-preFMM these







For x_i in L_3 we can build R expansions in *b*

$r^* < R^*$

S-expansion: $|\mathbf{y} - \mathbf{x}_*| > R_* > |\mathbf{x}_i - \mathbf{x}_*|$

R-expansion: $|\mathbf{y} - \mathbf{x}_*| < r_* < |\mathbf{x}_i - \mathbf{x}_*|$

Algorithm parameters & optimization

- Suggest that P and l should be balanced but do not provide an optimal choice
- A good project
  - to implement this and find out optimal choices.
  - compare this scheme with our scheme and check efficiency
  - Perhaps develop a combined scheme.

What Does Mean ``Adaptive”?

- The number of operations depends on data sets.
- Skipping boxes that do not contain sources and evaluation points in FMM is the first step of adaptation (one should not go through all boxes).
- Fully adaptive method should be able to design the computational process depending on 2 data sets: Source and Evaluation Points.
- The regular multilevel FMM uses pyramid data structure, while for the adaptive method we use tree (and/or forest) data structures.

General Idea

1). Potential at *any* level of hierarchical space subdivision can be computed as

Contribution of neighborhood

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

Contribution of all other sources

2). Therefore for each point $\mathbf{y}_j$ we can determine a level l_j of the space subdivision at which the number of sources in the neighborhood does not exceed some prescribed number q and the cost of direct computation of the field of sources in the neighborhood is $O(q)$.

To determine the second term in the right hand side we need to compute coefficients $\mathbf{D}^{(n,l)}$ only for level $l = l_j \leqslant L$ that is specific for each point $\mathbf{y}_j$ and is determined by the number of points of the set $\mathbb{X}$ in the neighborhood of that point. The set of box numbers and levels (n_j, l_j), $j = 1, \ldots, M$, for which we need to evaluate $\mathbf{y}_j$ will provide a set of *target* boxes for which we need to obtain $\mathbf{D}^{(n_j,l_j)}$.

General Idea (2)

3). To compute coefficients $\mathbf{D}^{(n_j,l_j)}$ we will construct a tree which will show what other coefficients $\mathbf{D}^{(n,l)}$, $l < l_j$ are needed to obtain the value of $\mathbf{D}^{(n_j,l_j)}$. We will call such tree as *D-tree*. Since $\mathbf{D}^{(n_j,l_j)}$ depends not only on $\mathbf{D}^{(n,l)}$, but also on $\widetilde{\mathbf{D}}^{(n,l)}$, we will also construct a $\widetilde{D}$-*tree*, which shows the entire set of coefficients $\widetilde{\mathbf{D}}^{(n,l)}$ that is needed for computation of $\mathbf{D}^{(n_j,l_j)}$.

4). More detailed look at the formulae for computation of coefficients show that the $\widetilde{D}$-*tree* is the same as the *D-tree*.

5). Further, we will build a *C-forest* (which may be a single tree or a union of several trees), that will contain the numbers and levels of boxes, (n, l), for which we need to have $\mathbf{C}^{(n,l)}$ coefficients to obtain $\widetilde{\mathbf{D}}^{(n,l)}$ and further $\mathbf{D}^{(n,l)}$ belonging to the **D**-*tree* and $\widetilde{\mathbf{D}}$-*tree*.

General Idea (3)

The described scheme is:

- Adaptive, since it provides the level of the box specific for each point $\mathbf{y}_j$, which depends on the both sets $\mathbb{X}$ and $\mathbb{Y}$.
- Presumably more economic than the regular FMM, since it is seeking for computation of the coefficients that are necessary to achieve the objective and does not need computation of unnecessary coefficients.
- Presumably more accurate than the regular FMM, since it avoids unnecessary translations.

Setting Data Structure. Step 0.

Generate the same data structure as for regular MLFMM.

Setting Data Structure. Step 1.

Determine the Set of Target Boxes.

The first step of the present adaptive algorithm includes determination of all target boxes and an appropriate number of levels of the hierarchical space subdivision. This depends on the prescribed number q that is the maximum allowed number of sources which can be summed directly to evaluate the potential due to these sources. All these tasks can be performed within one step procedure that algorithmically can be described as follows. In our notation we denote the set of target boxes as $\mathbb{T}$ and the number of levels in the hierarchical space subdivision as L. The set of target boxes at level l we will denote as $\mathbb{T}_l$. So $\mathbb{T} = \mathbb{T}_2 \cup \ldots \cup \mathbb{T}_L$.

1. Set the level of consideration $l = 2$ and the remaining set of the evaluation points $\mathbb{Y}_{rem} = \mathbb{Y}$.

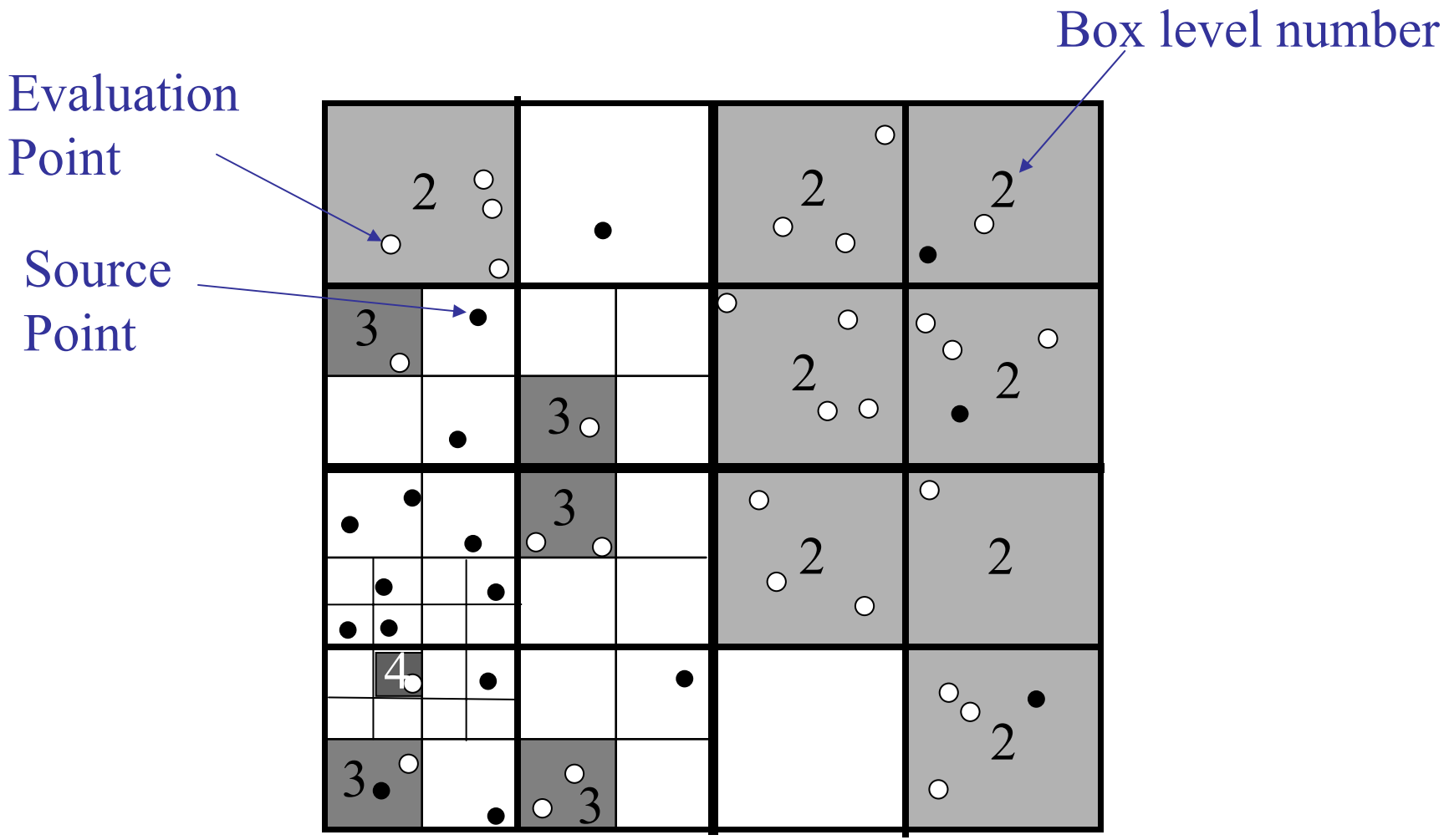
2. For each box (n, l), that contains a point $\mathbf{y}_j \in \mathbb{Y}_{rem}$ find the number N_n of points $\mathbf{x}_i \in E_2(n, l)$.

3. If $N_n \leqslant q$ refer the box (n, l) to the set of target boxes $\mathbb{T}_l$ and exclude all $\mathbf{y}_j \in E_1(n, l)$ from the set $\mathbb{Y}_{rem}$.

4. If $\mathbb{Y}_{rem} = \emptyset$, set the maximum number of levels $L = l$ and stop the procedure, else increase the level, $l = l + 1$, and repeat steps 2-4 of this algorithm.

Example of Determination of Target Boxes

In this example target boxes are shaded. Each of them contains not more than 3 source points in the neighborhood.



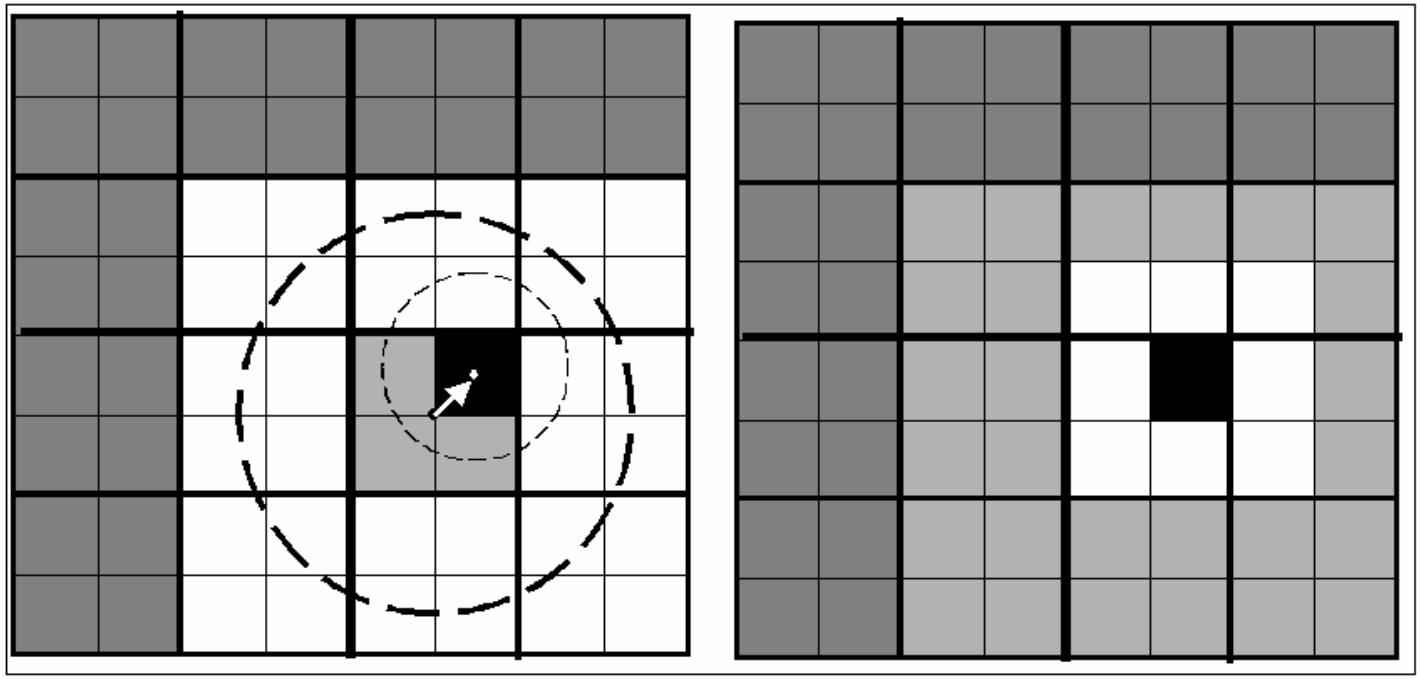
Setting Data Structure. Step 2.

Build D-tree.

Consider Step 2 of the MLFMM Downward Pass.

$$\Phi_{n,l}^{(3)}(\mathbf{x}) = \mathbf{D}^{(n,l)} \circ \mathbf{R}(\mathbf{x} - \mathbf{x}_c^{(n,l)}),$$

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



To compute *D* for a box we need *D* for the *Parent* of this box

Setting Data Structure. Step 2 (2).

Build D-tree.

Algorithm

Let us denote the set of the boxes in the D-tree as $\mathbb{D}$. We also denote through $\mathbb{D}_l$ sets of boxes in the D-tree at level $l \geqslant 2$, so $\mathbb{D} = \mathbb{D}_2 \cup \ldots \cup \mathbb{D}_L$. The tree can be generated from the target box set by applying function *Parent* until the level $l = 2$ reached. In other words if box $(n, l) \in \mathbb{D}$, $l > 2$, then box $(Parent(n), l-1) \in \mathbb{D}$. The algorithm here is straightforward:

1. Refer all target boxes at the finest level of space subdivision L to set $\mathbb{D}_L$ and set $l = L$.
2. If $l = 2$ stop the procedure, else set $l = l - 1$ and perform steps 2-4 in a loop.
3. Determine $(Parent(n), l)$ for all boxes from $\mathbb{D}_{l+1}$ and refer them to set $\mathbb{D}_l$.
4. Make a union of $\mathbb{D}_l$ and target boxes $\mathbb{T}_l$ and return the result back to $\mathbb{D}_l$.

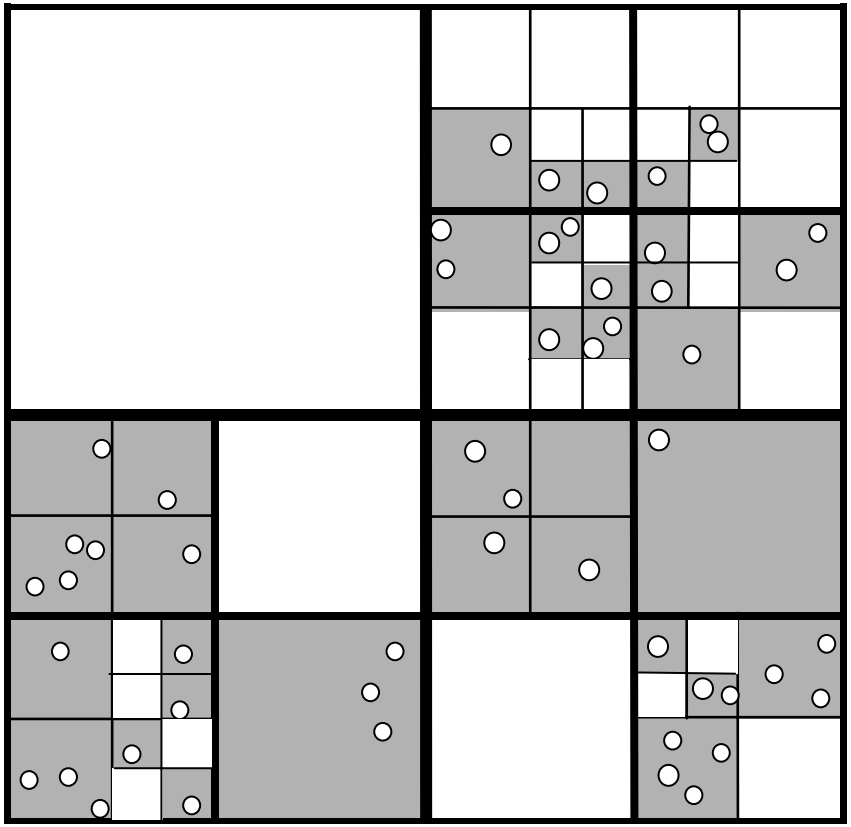
Note that in the D-tree all target boxes represent *leaves* of the tree (in other words end points for each branch of the tree). So $\mathbb{T}$ is the set of all leaves in the D-tree and $\mathbb{T}_l$ is the set of leaves of this tree at level $l = 2, \ldots, L$.

Setting Data Structure. Step 2 (3).

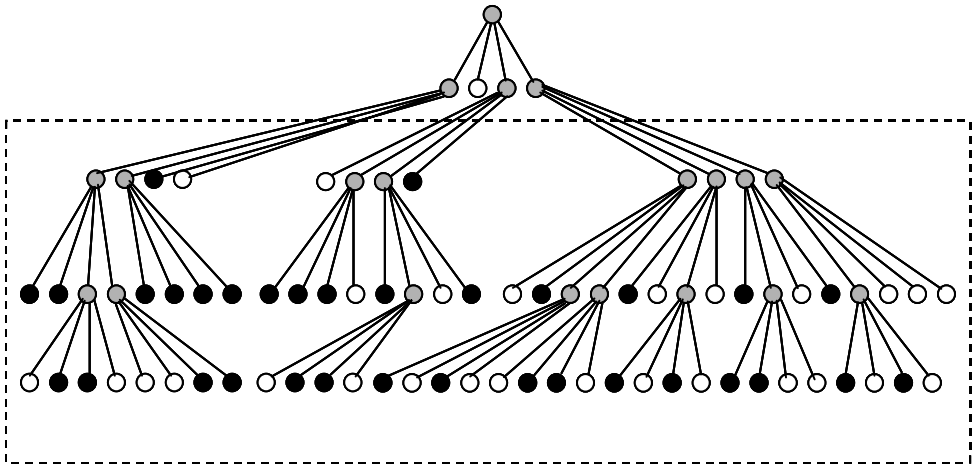
Build D-tree.

Example

Target Boxes



D-tree



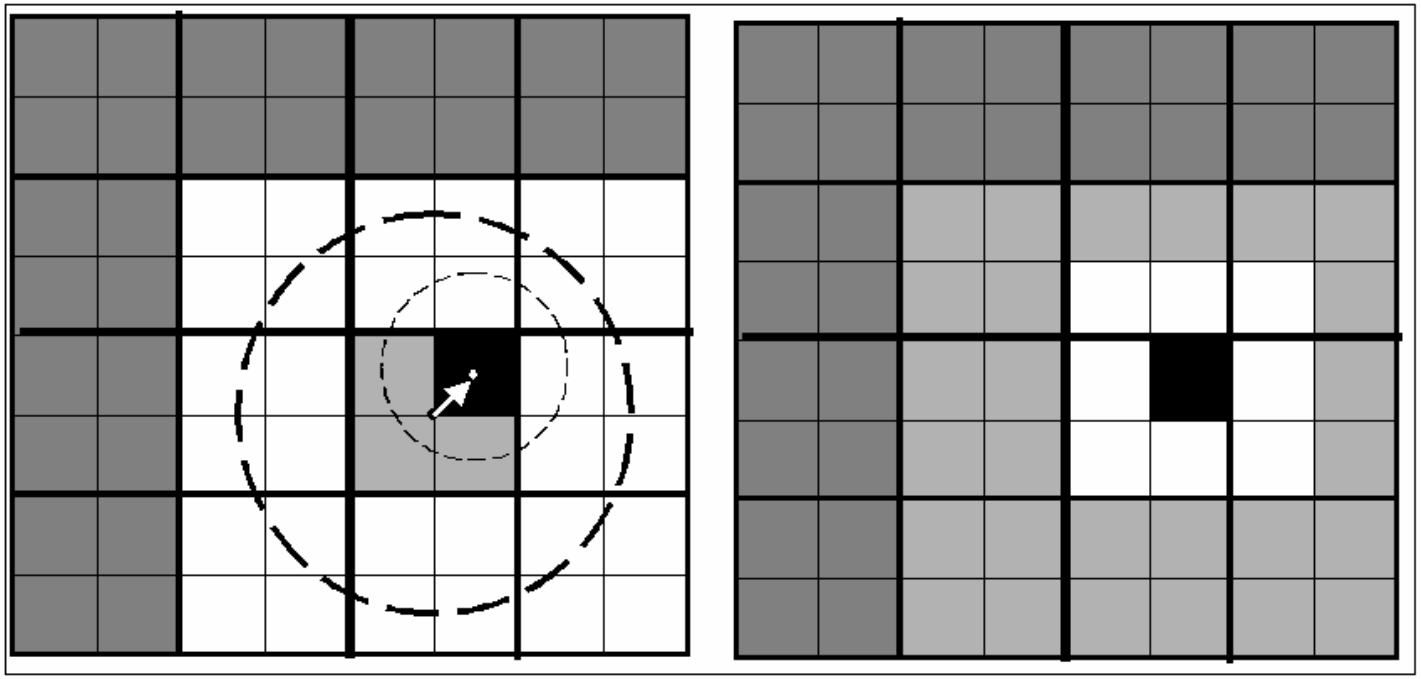
Setting Data Structure. Step 2 (4).

Build $\widetilde{\mathrm{D}}$-tree=D-tree.

Consider again Step 2 of the MLFMM Downward Pass.

$$\Phi_{n,l}^{(3)}(\mathbf{x}) = \mathbf{D}^{(n,l)} \circ \mathbf{R}(\mathbf{x} - \mathbf{x}_c^{(n,l)}),$$

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



To compute D for a box we need $\widetilde{D}$ for the *Same* box

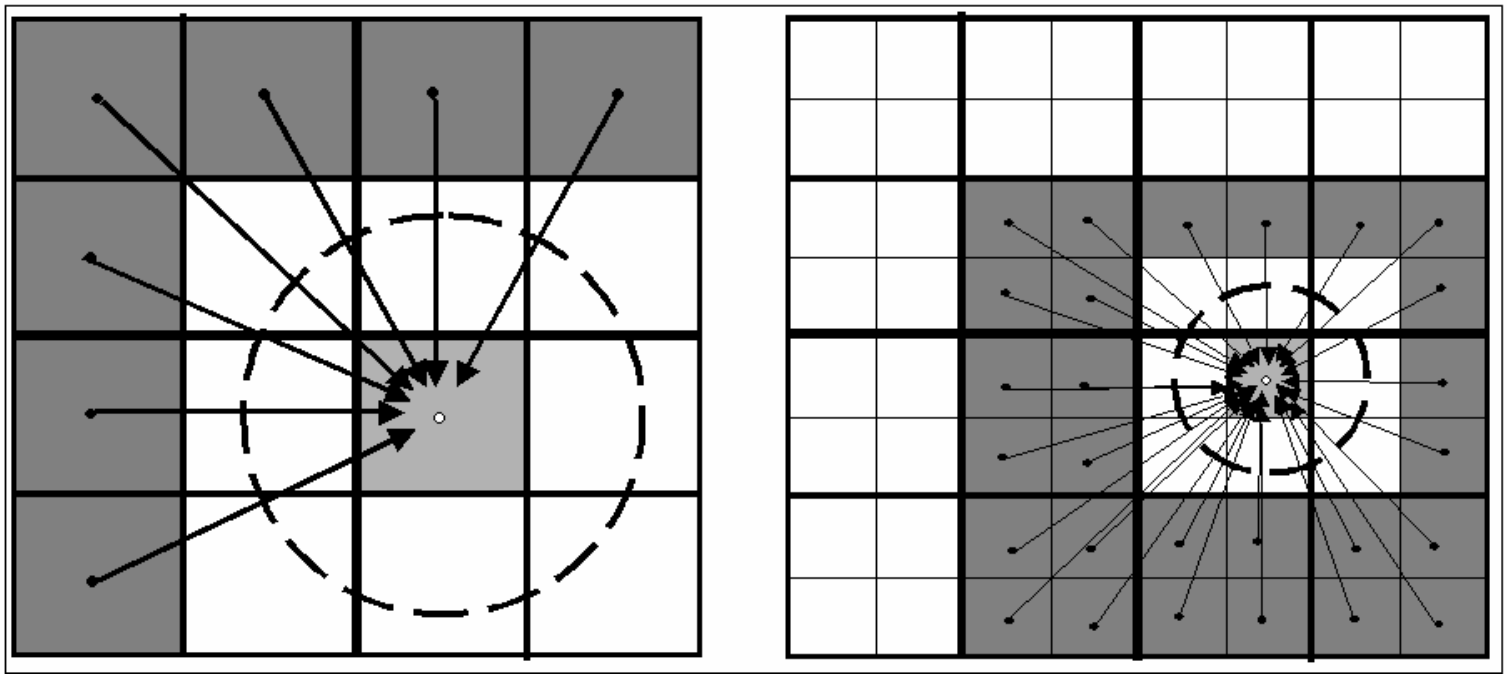
Setting Data Structure. Step 3 (1).

Build C-set.

Consider Step 1 of the MLFMM Downward Pass.

$$\Phi_{n,l}^{(4)}(\mathbf{x}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{x} - \mathbf{x}_c^{(n,l)}),$$

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



To compute $\widetilde{D}$ for a box we need C for the *I4 neighborhood*.

Setting Data Structure. Step 3 (2).

Build C-set.

$\mathbb{C}$ is the set of boxes for which we need to compute $\mathbf{C}^{(n,l)}$.
$\mathbb{C} = \mathbb{C}_2 \cup \ldots \cup \mathbb{C}_{L-1}$ (for each level)

1. For levels $l = L$ to $l = 2$ perform the following operation.
2. For each $(n,l) \in \mathbb{D}_l$ refer boxes $\in I_4(n,l)$ and contain points of $\mathbb{X}$ to $\mathbb{C}_l$.

Setting Data Structure. Step 4 (1).

Build C-forest.

The C-forest as it is clear from this name has a different structure than D-tree. We consider the structure called 'forest' as a union of K trees. The number K of these trees depends on the D-tree and the source data set $\mathbb{X}$, and we will determine it within the algorithm that constructs the C-forest. We denote these trees as $\mathbb{C}^1, \ldots, \mathbb{C}^K$. These trees are independent, so

$$\mathbb{C}^k \cap \mathbb{C}^j = \emptyset, \quad k \neq j, \quad \mathbb{C} = \mathbb{C}^1 \cup \ldots \cup \mathbb{C}^K.$$

Each tree will have its own coarsest and finest level, which we denote as l_k and L_k, respectively, $k = 1, \ldots, K$. All these levels are inside the range $2 \leqslant l_k \leqslant L_k \leqslant L$, and will be determined within the algorithm provided below. We also denote as $\mathbb{C}_l$ the sets of boxes in the C-forest at level $l \geqslant 2$, so $\mathbb{C} = \mathbb{C}_2 \cup \ldots \cup \mathbb{C}_{L-1}$. The boxes of level l can belong to different trees. We denote the set of boxes at level l that belongs to the k-th tree as $\mathbb{C}_l^k$. It follows from the definition that $\mathbb{C}_l^k = \mathbb{C}^k \cap \mathbb{C}_l$.

Setting Data Structure. Step 4 (2).

Build C-forest.

Algorithm

Determine K, l_k, L_k and sets $\mathbb{C}_l^k$, $l = 2,...,L$, $k = 1,...,K$.

1. Find the number of elements P_2 in the set $\mathbb{C}_2$. Set $K = P_2$, and $\mathbb{K} = [1,...,K]$. For $k = 1,...,K$ set $l_k = 2$, and refer each element to $\mathbb{C}_2^k$.

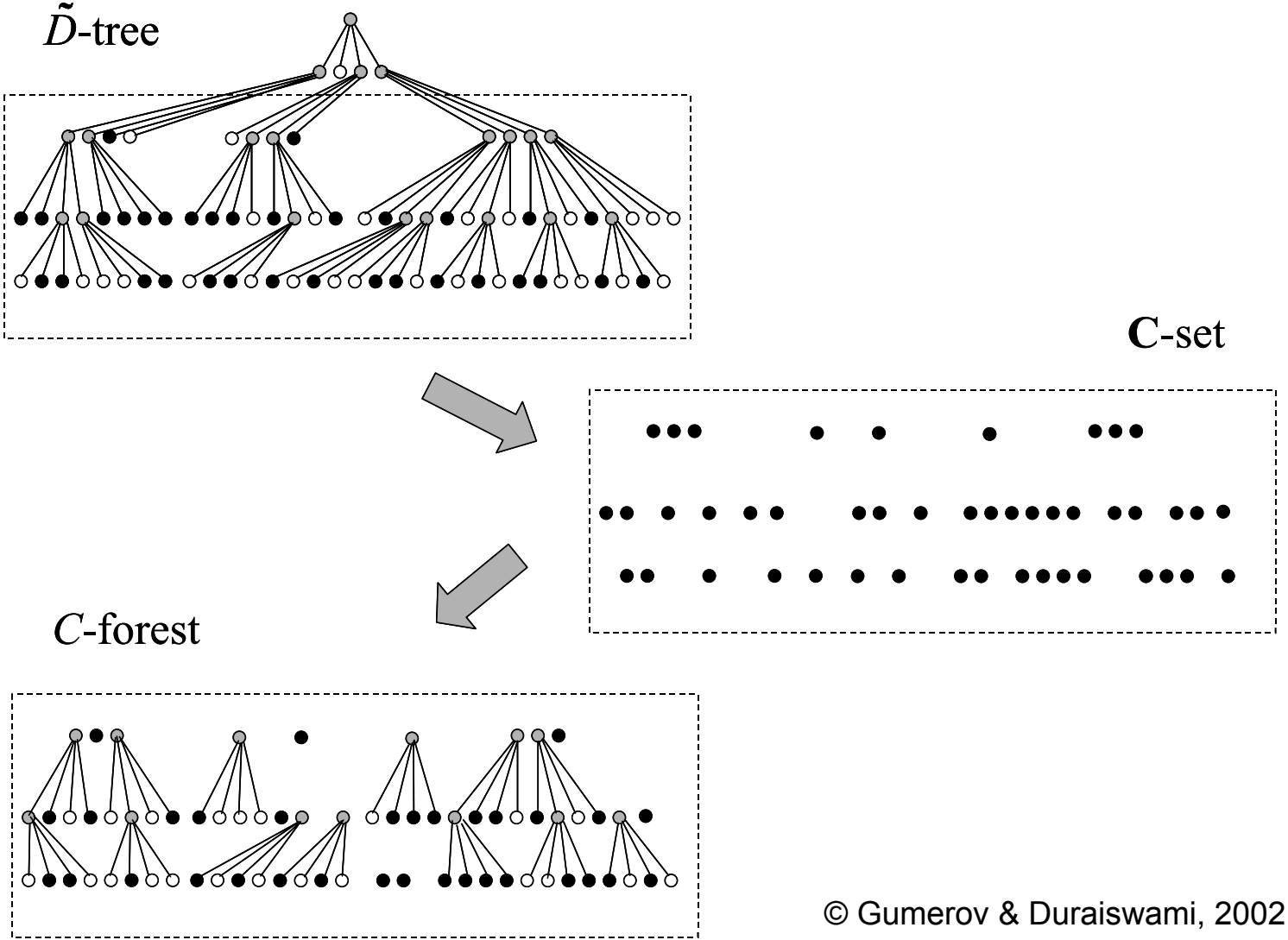
2. For $l = 3,...,L$ perform the following loop.

3. For $k \in \mathbb{K}$ find sets $\mathbb{G}_l^k = \mathit{Children}(\mathbb{C}_{l-1}^k) \cap \mathbb{C}_l$. If $\mathbb{G}_l^k \neq 0$, refer all elements of $\mathbb{G}_l^k$ to $\mathbb{C}_l^k$, else exclude k from the set $\mathbb{K}$ and set $L_k = l - 1$.

4. Find the number of elements P_l in the set $\mathbb{C}_l \backslash \left(\bigcup_{k \in \mathbb{K}} \mathbb{G}_l^k \right)$. Increase $K = K + P_l$, $\mathbb{K} = \mathbb{K} \cup [K+1,...,K+P_l]$. For $k = K+1,...,K+P_l$ set $l_k = l$, and refer each element of $\mathbb{C}_l \backslash \left(\bigcup_{k \in \mathbb{K}} \mathbb{G}_l^k \right)$ to $\mathbb{C}_l^k$.

Setting Data Structure. Step 4 (3).

Build C-forest.



Upward Pass (1)

- For each Tree in the forest can be computed independently;
- Skipping unnecessary translations, computing all leaves directly;
- Skipping unnecessary translations when going Upward.

Upward Pass (2)

1. For $k = 1, \ldots, K$ (for each tree) perform the following steps.

2. Generate $\mathbf{C}^{(n,l)}$ for the leave boxes, $l = L_k$, $\mathbf{C}^{(n,l)} \in \mathbb{C}^k_{L_k}$:

$$\forall (n, L_k) \in \mathbb{C}^k_{L_k}, \quad \Phi^{(1)}_{n,L_k}(\mathbf{y}) = \mathbf{C}^{(n,L_k)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,L_k)}),$$

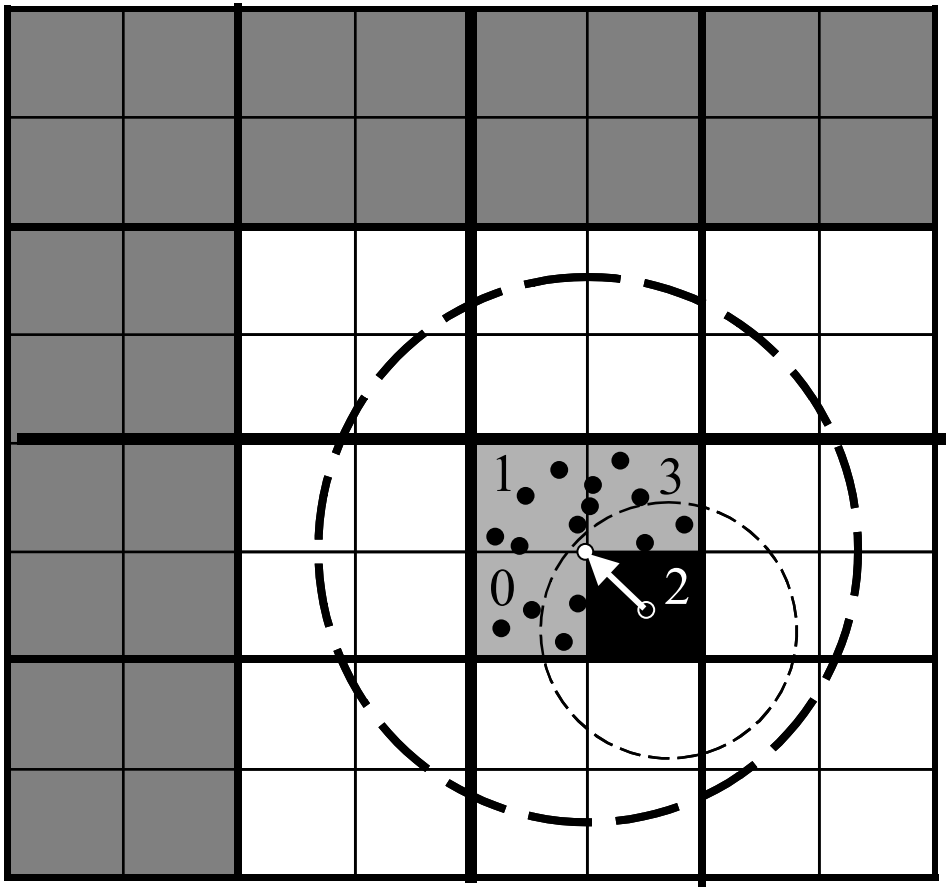
$$\mathbf{C}^{(n,L_k)} = \sum_{\mathbf{x}_i \in E_1(n,L_k)} u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L_k)}\right).$$

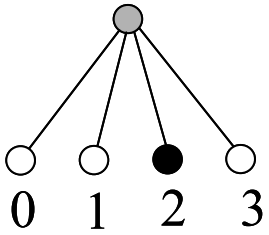
3. For $l = L_k - 1, \ldots, l_k$ recursively generate $\mathbf{C}^{(n,l)}$ for all other boxes in tree $\mathbb{C}^k$:

$$\forall (n, l) \in \mathbb{C}^k_l, \quad \Phi^{(1)}_{n,l}(\mathbf{y}) = \mathbf{C}^{(n,l)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

$$\mathbf{C}^{(n,l)} = \sum_{\substack{n' \in Children(n) \\ \& (n', l+1) \in \mathbb{C}^k_{l+1}}} (\mathbf{S}|\mathbf{S})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(n',l+1)}\right)\mathbf{C}^{(n',l+1)} + \sum_{\substack{\mathbf{x}_i \in E_1(n',l+1) \\ n' \in Children(n) \\ \& (n', l+1) \notin \mathbb{C}^k_{l+1}}} u_i \mathbf{B}\left(\mathbf{x}_i - \mathbf{x}_c^{(n,l)}\right).$$

Upward Pass (3)





Downward Pass

1. For $l = 2, \ldots, L$ perform the following steps

2. For $(n, l) \in \mathbb{D}_l$ find $\widetilde{\mathbf{D}}^{(n,l)}$:

$$\Phi_{n,l}^{(4)}(\mathbf{y}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{n' \in I_4^1(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(n',l)}\right)\mathbf{C}^{(n',l)} + \sum_{n' \in Parent\left(I_4^2(n,l)\right)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(n',l-1)}\right)\mathbf{C}^{(n',l-1)}.$$

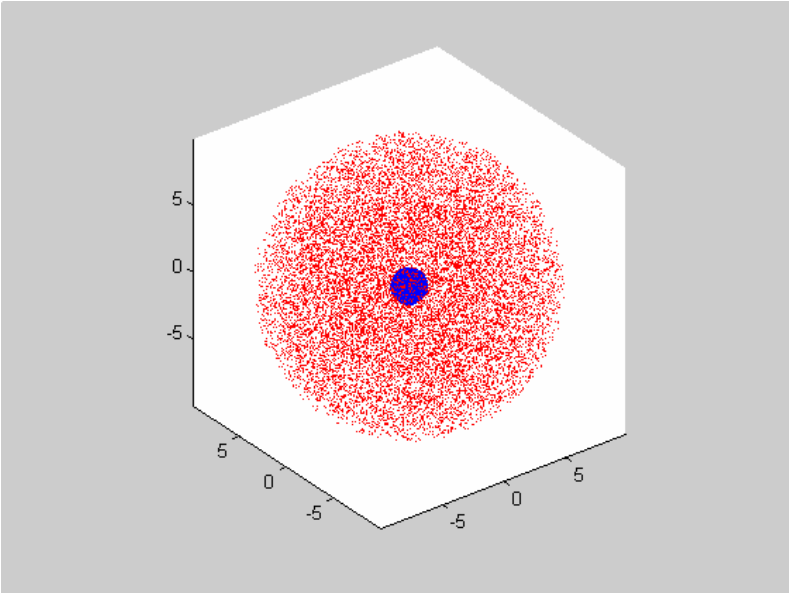
3. For $(n, l) \in \mathbb{D}_l$ find $\mathbf{D}^{(n,l)}$ (set $\mathbf{D}^{(n,2)} = \widetilde{\mathbf{D}}^{(n,2)}$):

$$\Phi_{n,l}^{(3)}(\mathbf{y}) = \mathbf{D}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

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

Final Summation

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



Comparison of the Regular and Adaptive FMM. Sources (blue) distributed uniformly over the sphere. Targets (red) distributed uniformly inside a larger sphere.