Introduction to the FMM

Nail A. Gumerov

(Lecture for CMSC 828E)

Outline

- About the FMM
- Fast Matrix-Vector Multiplication
- Key Ideas
  - Factorization/Expansions
  - Sparse+Dense Low Rank Decomposition (Space partitioning)
  - Translations
- Single level FMM
- Multilevel FMM
- On Data Structures

About the FMM

- Introduced by Rokhlin & Greengard (1987,1988) for computation of 2D and 3D fields for Laplace Equation;
- Reduces complexity of matrix-vector product from $O(N2)$ to $O(N)$ or $O(N\log N)$ (depends on data structure);
- Hundreds of publications for various 1D, 2D, and 3D problems (Laplace, Helmholtz, Maxwell, Yukawa Potentials, etc.);
- We taught the first in the country course on FMM fundamentals & application at the University of Maryland (2002);
- Our reports on fundamentals of the FMM and lectures are available online (visit our web pages).

Large problems. Example Stellar dynamics.

Problem:

Compute dynamics of star cluster (Solve large system of ODE's).

Info: A galaxy like Milky Way has 100 millions stars and evolves for billions years.

Example: N-body problem

N stars ($i,j=1,\ldots,N$)

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{v}_j,$$

$$m_j \frac{d\mathbf{v}_j}{dt} = \mathbf{F}_j,$$

$$F_j = -G \sum_{i \neq j} m_i m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{\left|\mathbf{x}_j - \mathbf{x}_i\right|^3} = G m_j \nabla\phi\big|_{\mathbf{x}=\mathbf{x}_j},$$

$$\phi(\mathbf{x}) = \sum_i \frac{m_i}{\left|\mathbf{x} - \mathbf{x}_i\right|}.$$

About the FMM

Problem:

Compute matrix-vector product

$$\mathbf{v} = \mathbf{\Phi u}$$

$$v_j = \sum_{i=1}^{N} \Phi_{ji} u_i, \quad j = 1, \ldots, M,$$

$$\Phi_{ji} = \Phi(\mathbf{y}_j, \mathbf{x}_i), \quad j = 1, \ldots, M, \quad i = 1, \ldots, N,$$

Some kernels

Laplace 3D:

$$\Phi(\mathbf{y}, \mathbf{x}_i) = \frac{1}{|\mathbf{y} - \mathbf{x}_i|}$$

Helmholtz 3D:

$$\Phi(\mathbf{y}, \mathbf{x}_i) = \frac{\exp\{ik|\mathbf{y} - \mathbf{x}_i|\}}{|\mathbf{y} - \mathbf{x}_i|}$$

Gaussian nD:

$$\Phi(\mathbf{y}, \mathbf{x}_i) = \exp\{-|\mathbf{y} - \mathbf{x}_i|^2/\sigma\}$$

Cases of Fast Matrix-Vector Multiplication

- Low rank matrices (can be dense)
- Sparse matrices (can be full rank)
- Structured matrices
  - Loosely speaking: Information on $O(N^2)$ matrix elements is provided by $O(N)$ numbers
  - Examples:
    - Discrete Fourier transform matrix
    - Toeplitz matrix
    - Cauchy matrix

Matrices treatable with the FMM are structured matrices

- Depend on O(N) coordinates of sources/receivers;
- Can be decomposed to

  Sparse Matrix+

  Low Rank Dense Matrix+

  Error
- Generally speaking the FMM is the algorithm which trades exactness for efficiency

Key Ideas: Factorization

Global Factorization

Expansion center

Truncation number

$$\forall \mathbf{x}_i, \mathbf{y}_j \in \Omega \subset \mathbb{R}^d :$$

$$\Phi\left(\mathbf{y}_j, \mathbf{x}_i\right) = \sum_{m=0}^{\infty} a_m(\mathbf{x}_i - \mathbf{x}_*) f_m\left(\mathbf{y}_j - \mathbf{x}_*\right) = \sum_{m=0}^{p-1} a_m(\mathbf{x}_i - \mathbf{x}_*) f_m\left(\mathbf{y}_j - \mathbf{x}_*\right) + Error(p, \mathbf{x}_i, \mathbf{y}_j)$$

Expansion coefficients

Basis functions

Factorization Trick

$$
\begin{aligned}
v_j &= \sum_{i=1}^{N} \Phi\left(\mathbf{y}_j, \mathbf{x}_i\right) u_i \\
&= \sum_{i=1}^{N} \left[\sum_{m=0}^{p-1} a_m(\mathbf{x}_i - \mathbf{x}_*) f_m\left(\mathbf{y}_j - \mathbf{x}_*\right) + Error(p; \mathbf{x}_i, \mathbf{y}_j) \right] u_i \\
&= \sum_{m=0}^{p-1} f_m\left(\mathbf{y}_j - \mathbf{x}_*\right) \sum_{i=1}^{N} a_m(\mathbf{x}_i - \mathbf{x}_*) u_i + \sum_{i=1}^{N} Error(p; \mathbf{x}_i, \mathbf{y}_j) u_i \\
&= \sum_{m=0}^{p-1} c_m f_m\left(\mathbf{y}_j - \mathbf{x}_*\right) + Error(N, p),
\end{aligned}
$$

where

$$
c_m = \sum_{i=1}^{N} a_m(\mathbf{x}_i - \mathbf{x}_*) u_i .
$$

Reduction of Complexity

Straightforward (nested loops):

$$
\begin{aligned}
&\text{for } j = 1,\ldots,M \\
&\quad v_j = 0; \\
&\quad \text{for } i = 1,\ldots,N \\
&\quad\quad v_j = v_j + \Phi\left(\mathbf{y}_j, \mathbf{x}_i\right) u_i; \\
&\quad \text{end;} \\
&\text{end;}
\end{aligned}
$$

Complexity: $O(MN)$

Factroized:

$$
\begin{aligned}
&\text{for } m = 0,\ldots,p-1 \\
&\quad c_m = 0; \\
&\quad \text{for } i = 1,\ldots,N \\
&\quad\quad c_m = c_m + a_m(\mathbf{x}_i - \mathbf{x}_*) u_i; \\
&\quad \text{end;} \\
&\text{end;}
\end{aligned}
$$

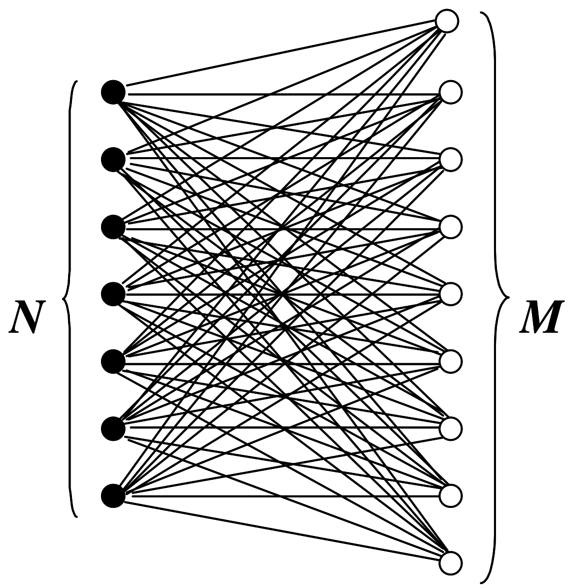
$$
\begin{aligned}
&\text{for } j = 1,\ldots,M \\
&\quad v_j = 0; \\
&\quad \text{for } m = 0,\ldots,p-1 \\
&\quad\quad v_j = v_j + c_m f_m\left(\mathbf{y}_j - \mathbf{x}_*\right); \\
&\quad \text{end;} \\
&\text{end;}
\end{aligned}
$$

Complexity: $O(pN+pM)$

If $p \ll \min(M,N)$ then complexity reduces!

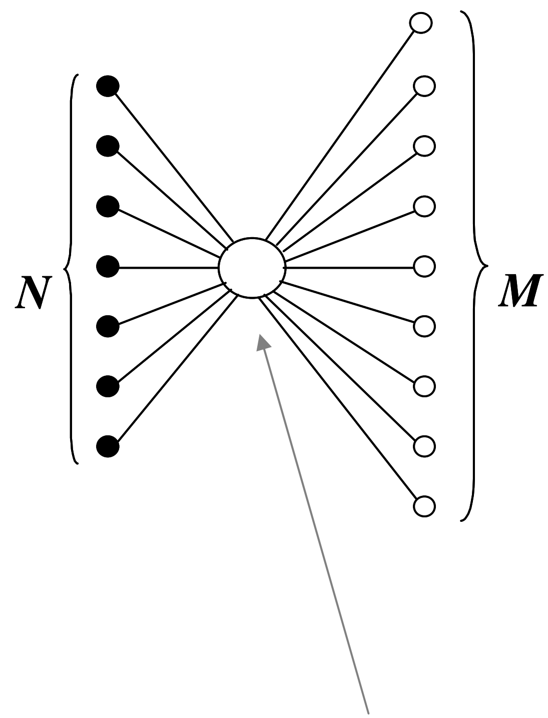
Middleman Scheme

Straightforward



Complexity: $O(pN+pM)$

Middleman



Set of coefficients $\{c_m\}$

Far Field and Near Field

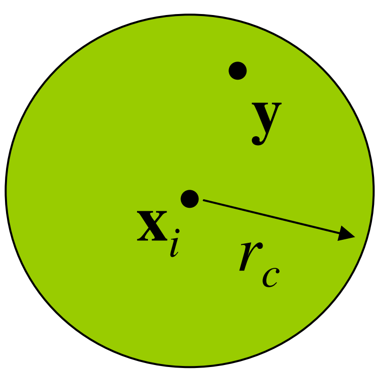
- Near Field of the *i*th source:

$$|\mathbf{y} - \mathbf{x}_i| < r_c.$$

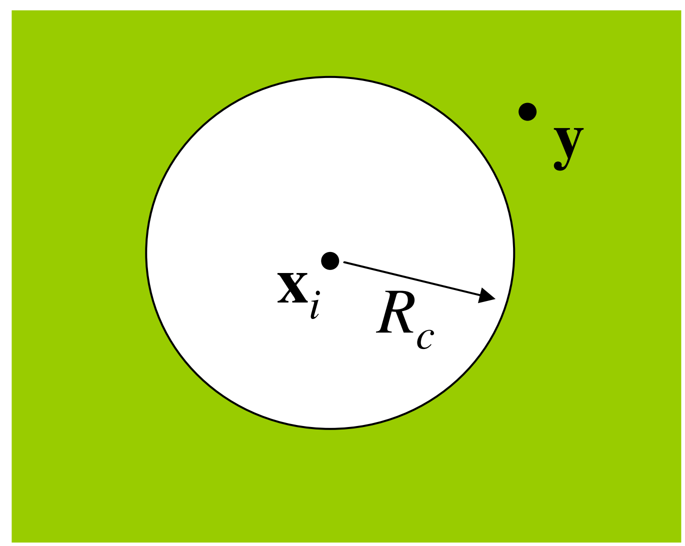
- Far Field of the *i*th source:

$$|\mathbf{y} - \mathbf{x}_i| > R_c.$$

Near Field



Far Field



What are these r_c and R_c ?
depends on the potential + some conventions for the terminology

Local (Regular) Expansion

Do not confuse with the Near Field!

Let

$$\mathbf{x}_* \in \mathbb{R}^d.$$

We call expansion

$$\Phi(\mathbf{y}, \mathbf{x}_i) = \sum_{m=0}^{\infty} a_m(\mathbf{x}_i, \mathbf{x}_*) R_m(\mathbf{y} - \mathbf{x}_*)$$

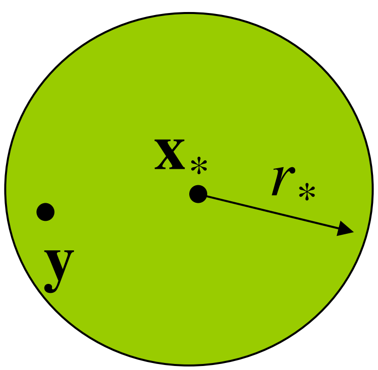
Basis Functions

Expansion Coefficients

local (regular) inside a sphere

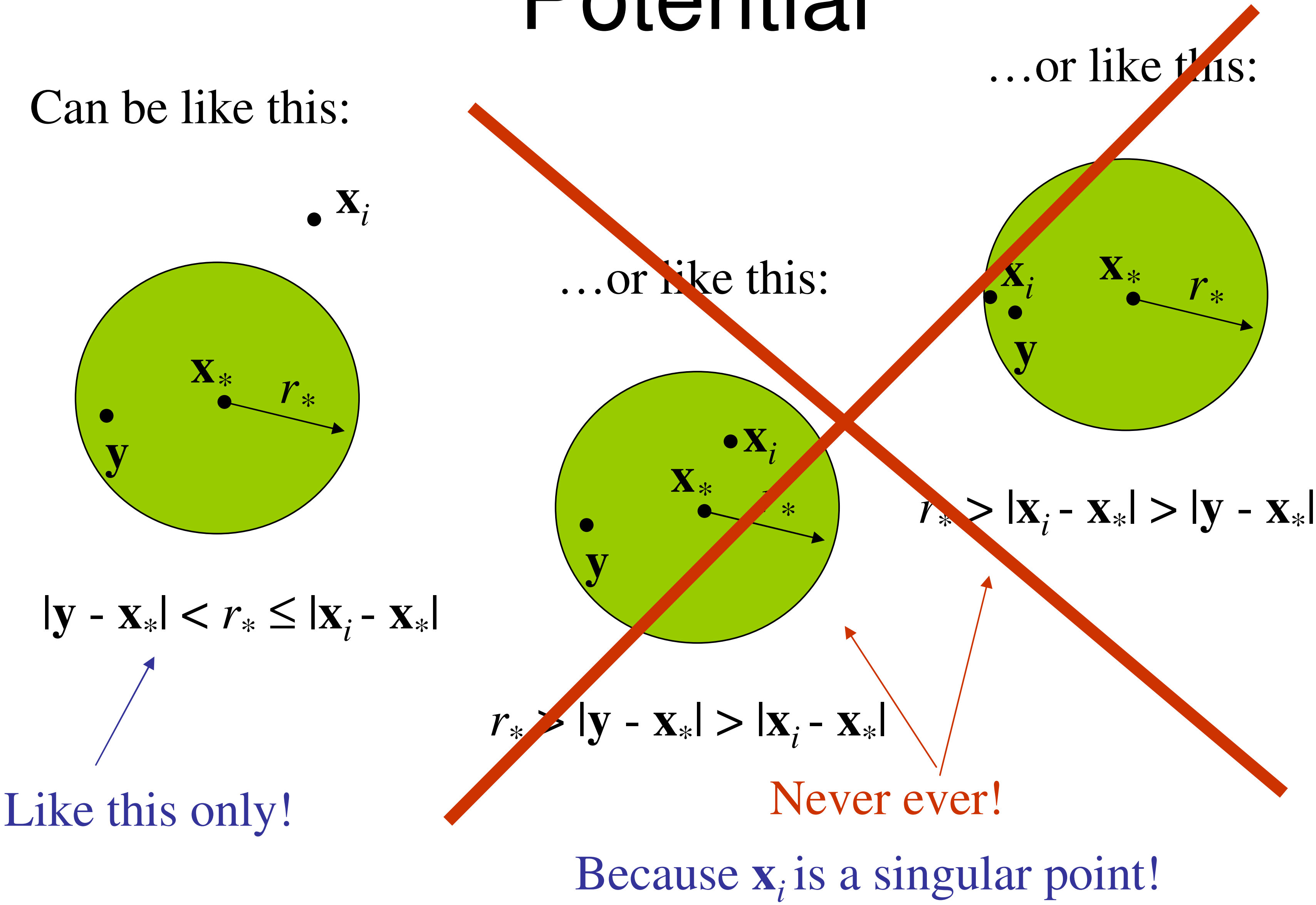
$$|\mathbf{y} - \mathbf{x}_*| < r_*,$$

if the series converges for $\forall \mathbf{y},\ |\mathbf{y} - \mathbf{x}_*| < r_*$.



We also call this R-expansion, since basis functions R_m should be *regular*

Local Expansion of a Singular Potential



Far Field Expansions (S-expansions)

Let $\mathbf{x}_* \in \mathrm{R}^d$.

Might be Singular (at $\mathbf{y} = \mathbf{x}_*$) Basis Functions

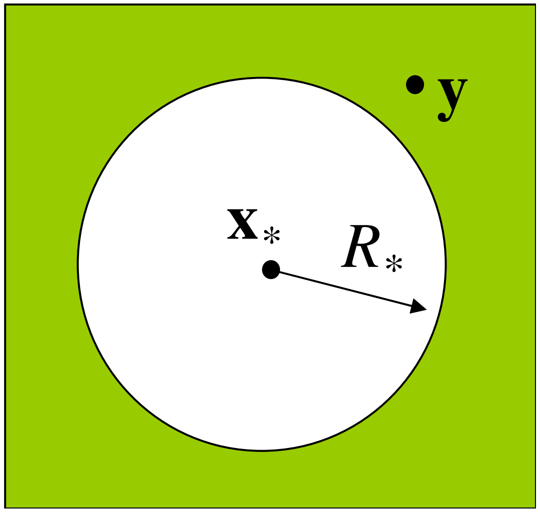
We call expansion

$$\Phi(\mathbf{y}, \mathbf{x}_i) = \sum_{m=0}^{\infty} b_m(\mathbf{x}_i, \mathbf{x}_*) S_m(\mathbf{y} - \mathbf{x}_*)$$

far field expansion (or S-expansion) outside a sphere

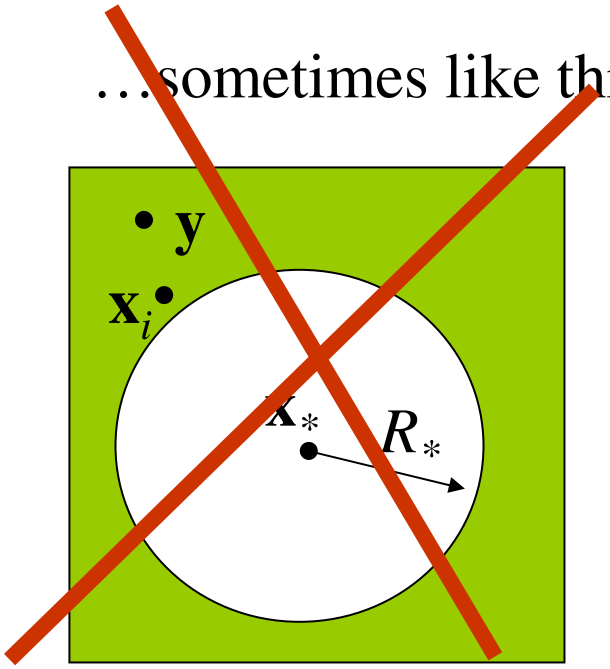
$$|\mathbf{y} - \mathbf{x}_*| > R_*,$$

if the series converges for $\forall \mathbf{y},\ |\mathbf{y} - \mathbf{x}_*| > R_*$.



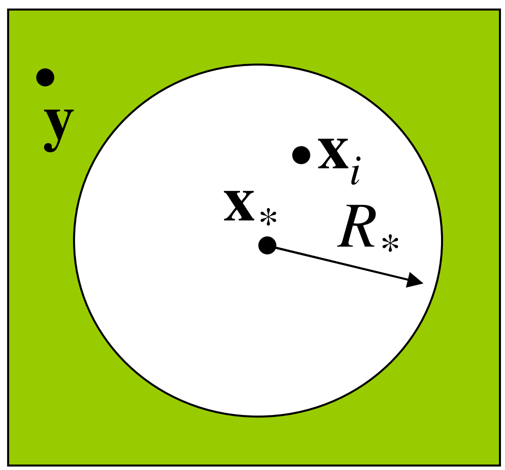
Far Field Expansion of a Singular Potential

…sometimes like this:



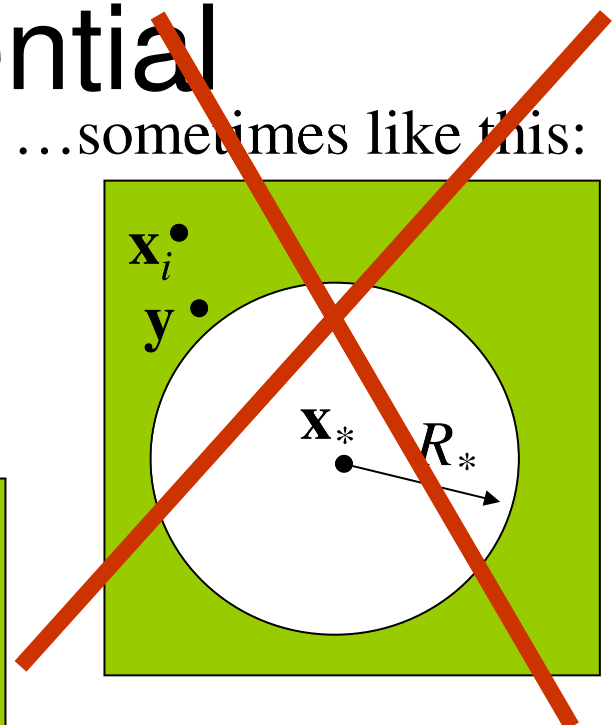
$|\mathbf{y} - \mathbf{x}_*| > R_* > |\mathbf{x}_i - \mathbf{x}_*|$

Can be like this:



$|\mathbf{y} - \mathbf{x}_*| > R_* \geq |\mathbf{x}_i - \mathbf{x}_*|$

…sometimes like this:



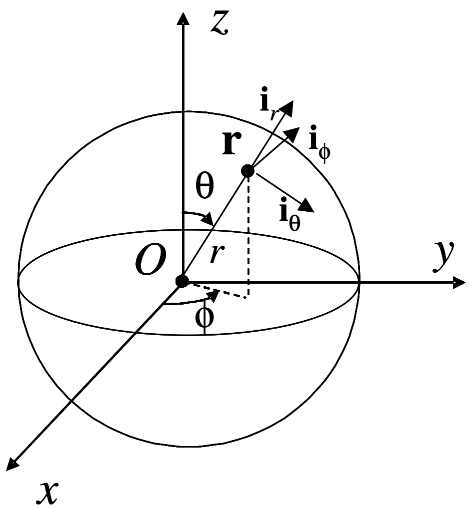
$|\mathbf{x}_i - \mathbf{x}_*| > |\mathbf{y} - \mathbf{x}_*| > R_*$

This case only!

Example: S- and R- expansions of Fundamental Solution of 3D Laplace Equation

$$\frac{1}{|\mathbf{r}-\mathbf{r}_0|} = \frac{4\pi}{r_0}\sum_{n=0}^{\infty}\frac{1}{2n+1}\left(\frac{r}{r_0}\right)^n\sum_{m=-n}^{n}Y_n^{-m}(\theta',\varphi')Y_n^m\left(\widehat{\theta},\widehat{\varphi}\right), \quad r < r_0,$$

$$\frac{1}{|\mathbf{r}-\mathbf{r}_0|} = \frac{4\pi}{r_0}\sum_{n=0}^{\infty}\frac{1}{2n+1}\left(\frac{r_0}{r}\right)^{n+1}\sum_{m=-n}^{n}Y_n^{-m}(\theta',\varphi')Y_n^m\left(\widehat{\theta},\widehat{\varphi}\right), \quad r > r_0.$$



$$\mathbf{r} = r(\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta),$$

$$R_n^m(\mathbf{r}) = r^n Y_n^m(\theta,\varphi),$$
$$S_n^m(\mathbf{r}) = r^{-n-1} Y_n^m(\theta,\varphi),$$

Multipole (!)

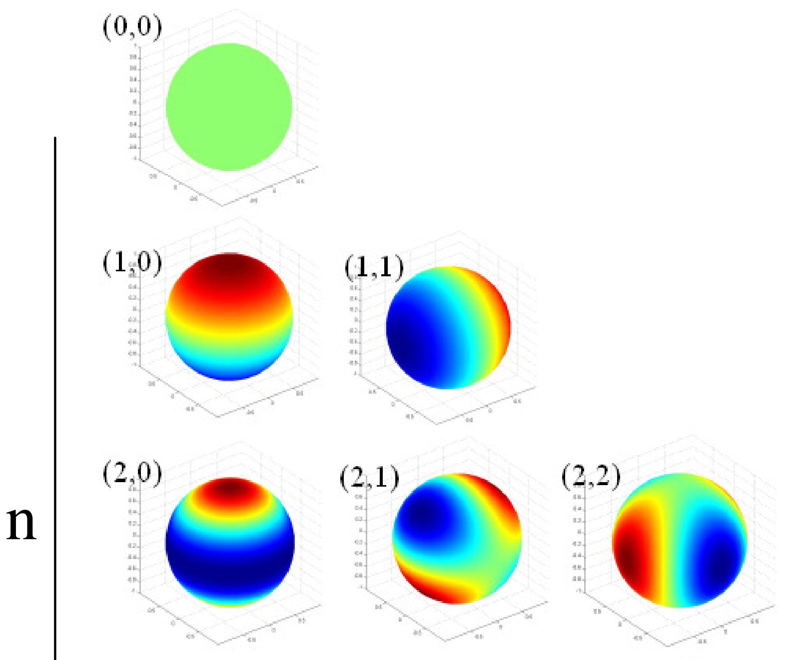
$$\frac{1}{4\pi|\mathbf{r}-\mathbf{r}_0|} = \sum_{n=0}^{\infty}\sum_{m=-n}^{n}\frac{1}{2n+1}S_n^{-m}(\mathbf{r}_0)R_n^m(\mathbf{r}), \quad r < r_0,$$

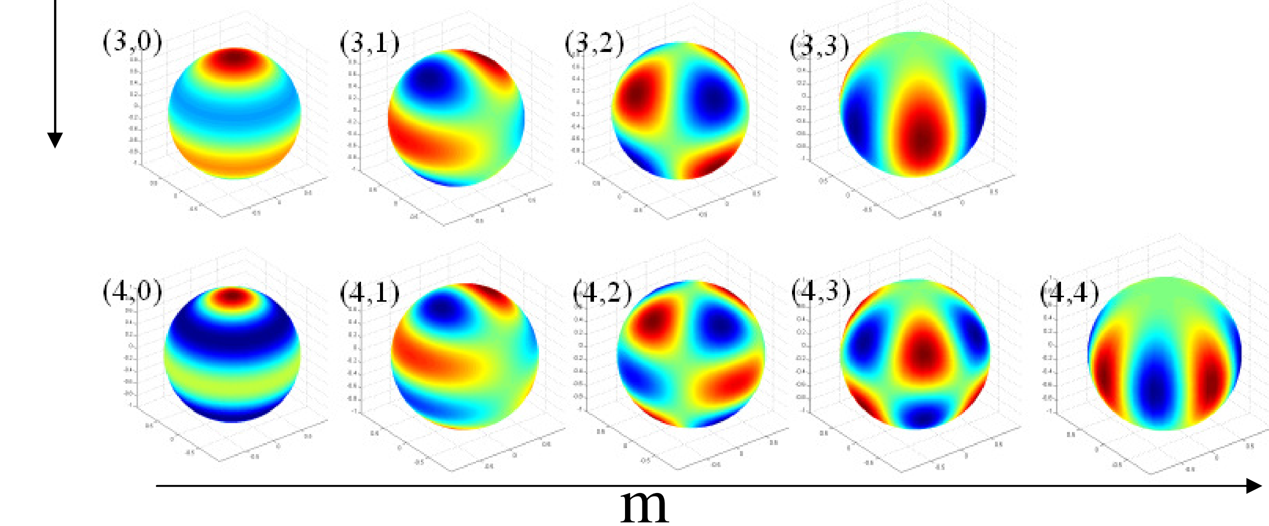
$$\frac{1}{4\pi|\mathbf{r}-\mathbf{r}_0|} = \sum_{n=0}^{\infty}\sum_{m=-n}^{n}\frac{1}{2n+1}R_n^{-m}(\mathbf{r}_0)S_n^m(\mathbf{r}), \quad r > r_0.$$

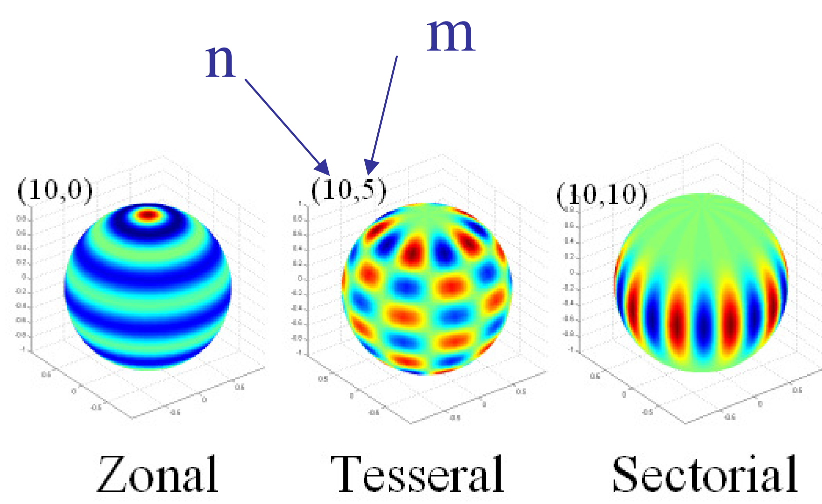
Spherical Harmonics

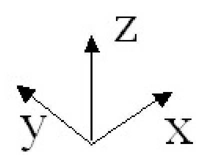
$$Y_n^m(\theta,\varphi) = (-1)^m \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}}\,P_n^{|m|}(\cos\theta)e^{im\varphi},$$

$$n = 0,1,2,\dots; \qquad m = -n,\dots,n.$$









$$Y_n^{-m}(\theta,\varphi) = \overline{Y_n^m(\theta,\varphi)}.$$

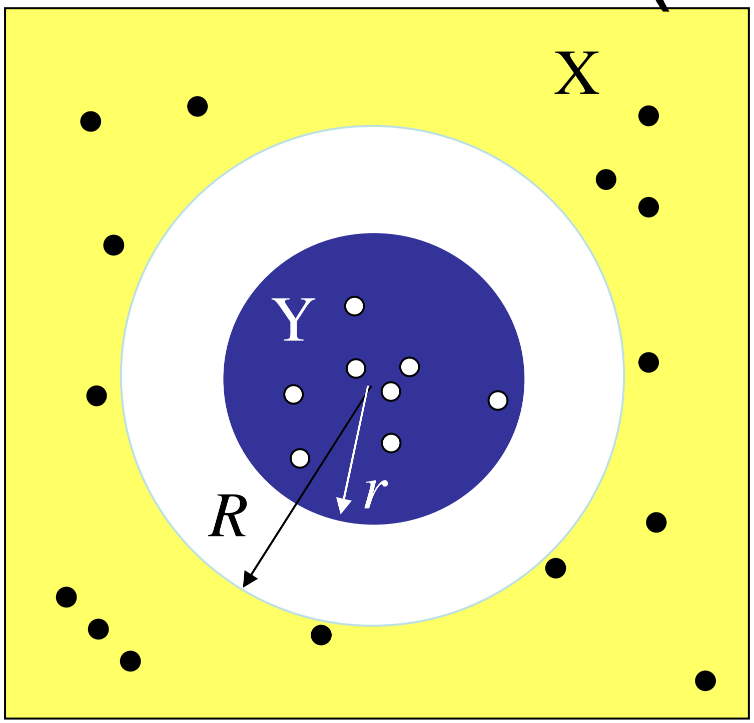
$$Y_0^0(\theta,\varphi) = \text{const} = \sqrt{\frac{1}{4\pi}}.$$

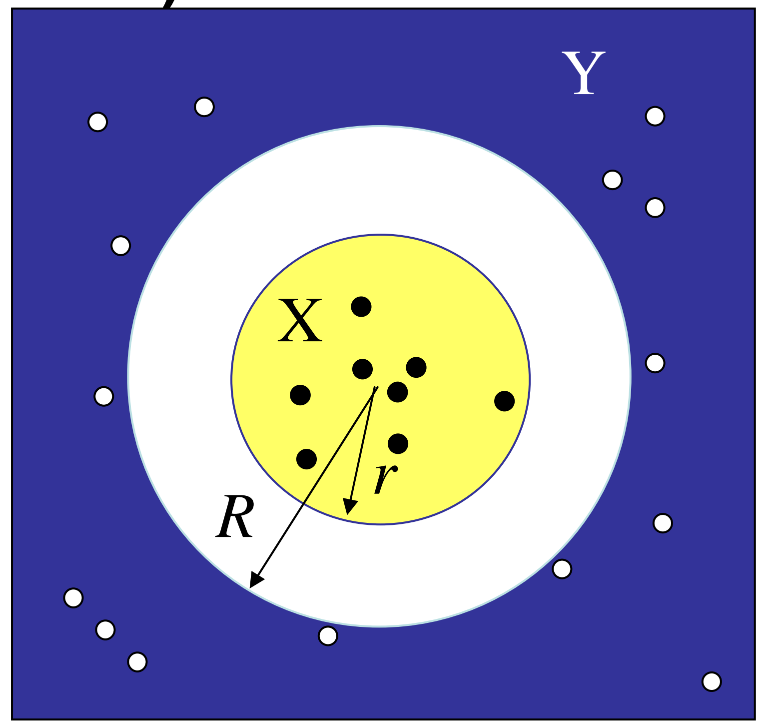
Key Ideas: Sparse+Low Rank Dense Matrix Decomposition

Well separated sets

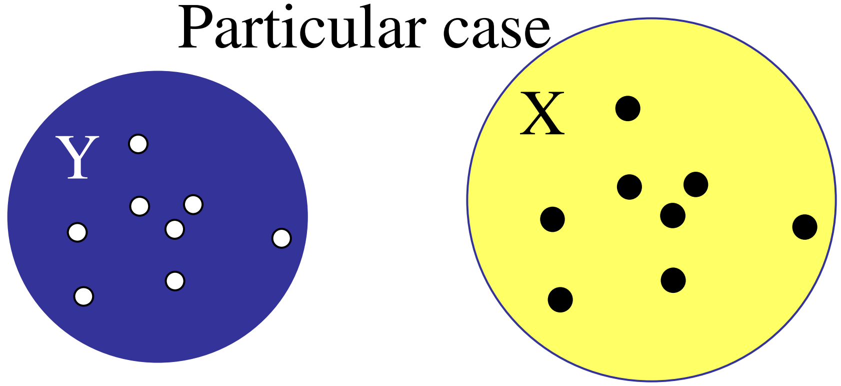
Definition: Two sets of points in R^d, X and Y, are called well separated, if there exist two co-centric spheres of radii r and R, $r < R$, such that all points of Y are located inside the smaller sphere, and there are no points of X located inside the larger sphere. (In this definition sets X and Y can be exchanged).

Well separated sets (examples)

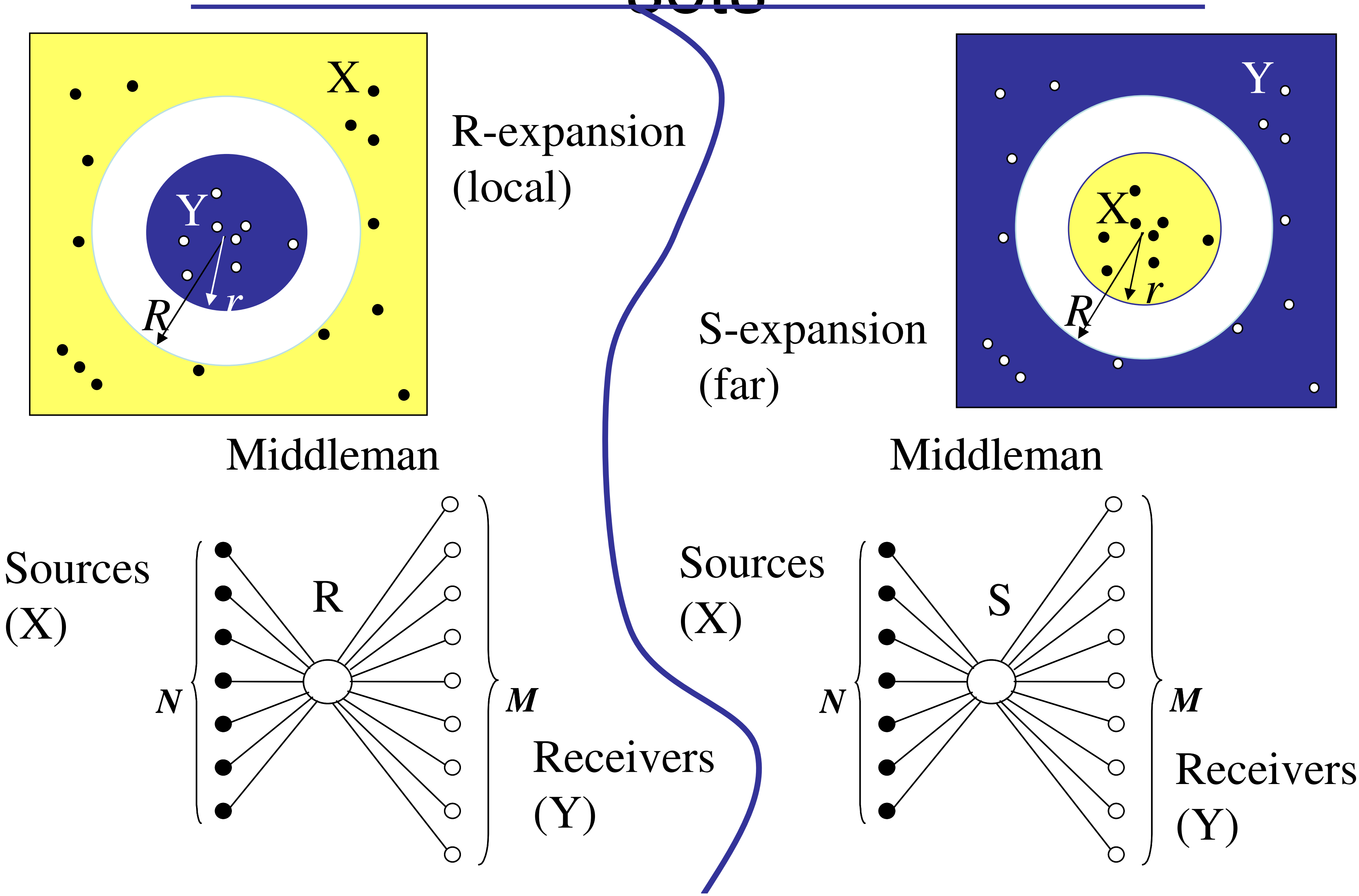




Particular case



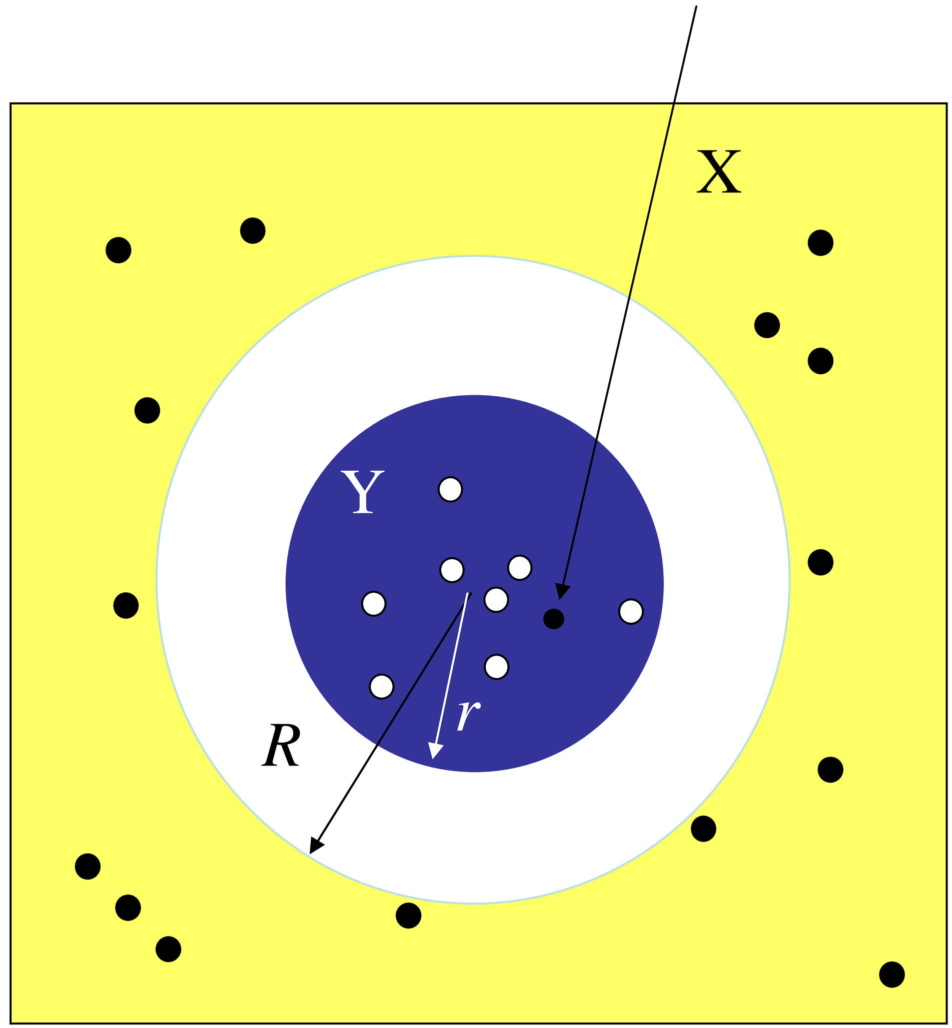
Middleman for well separated sets



Pecularities of ``Middleman” for singular kernels

- Separation of sets is crucial;
- Type of factorization (S or R) depends on the type of source/receiver distribution;
- Separation parameter, *r/R* controls the convergence of the series and for given accuracy the truncation number substantially depends on this parameter (so the efficiency of the fast summation method).

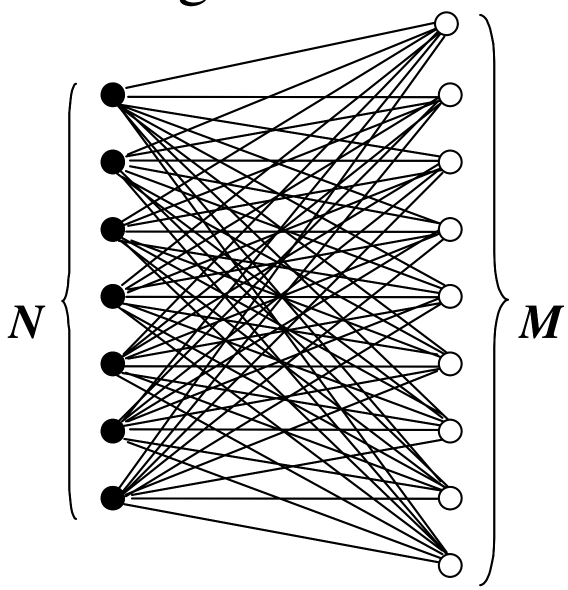
One point that spoils algorithm…



“bad point”,
“outlier”

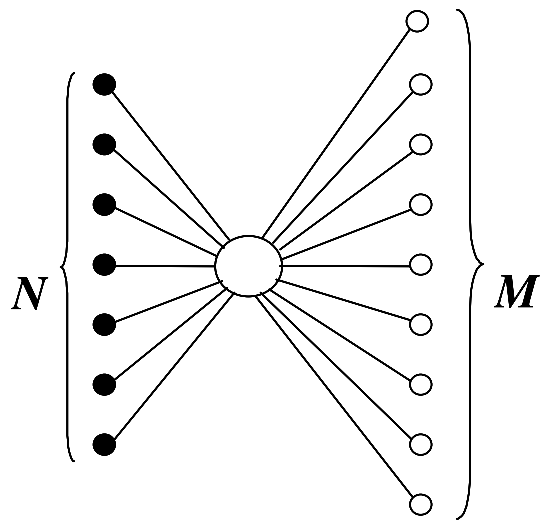
Modification of the “Middleman” for outliers

Straightforward



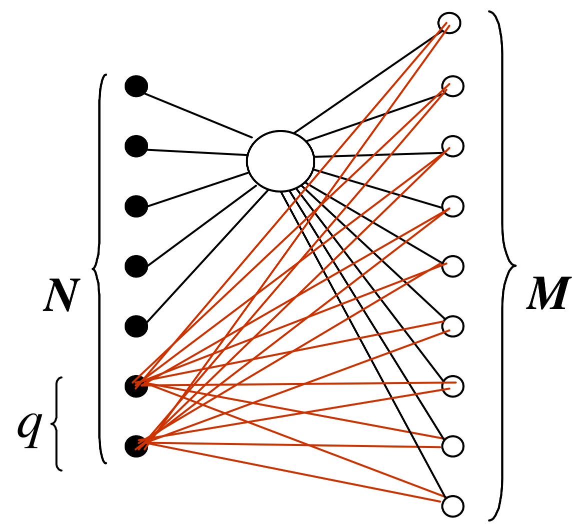
Complexity: $O(NM)$

Middleman



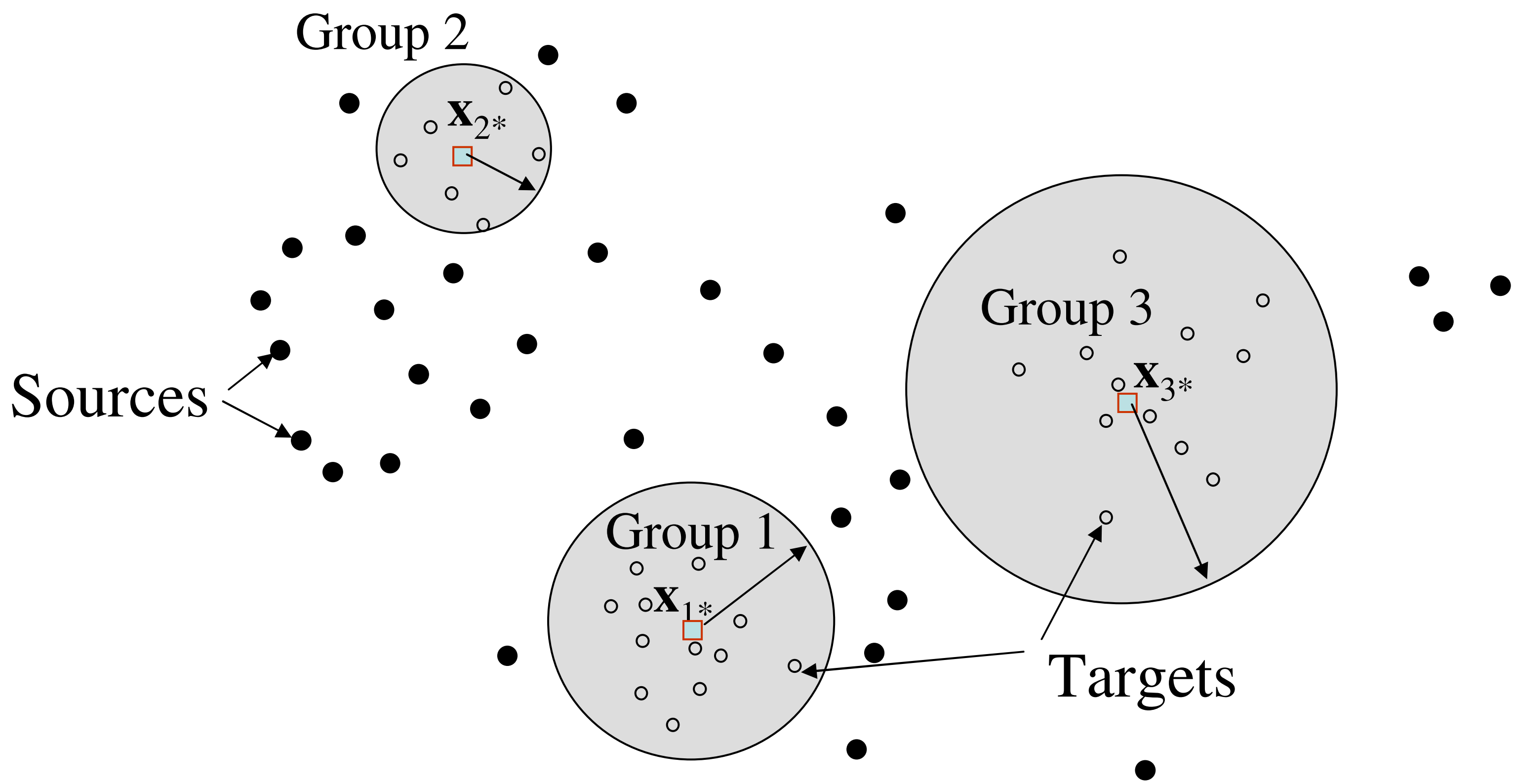
Complexity: $O(pN+pM)$

Middleman with outliers (sources)

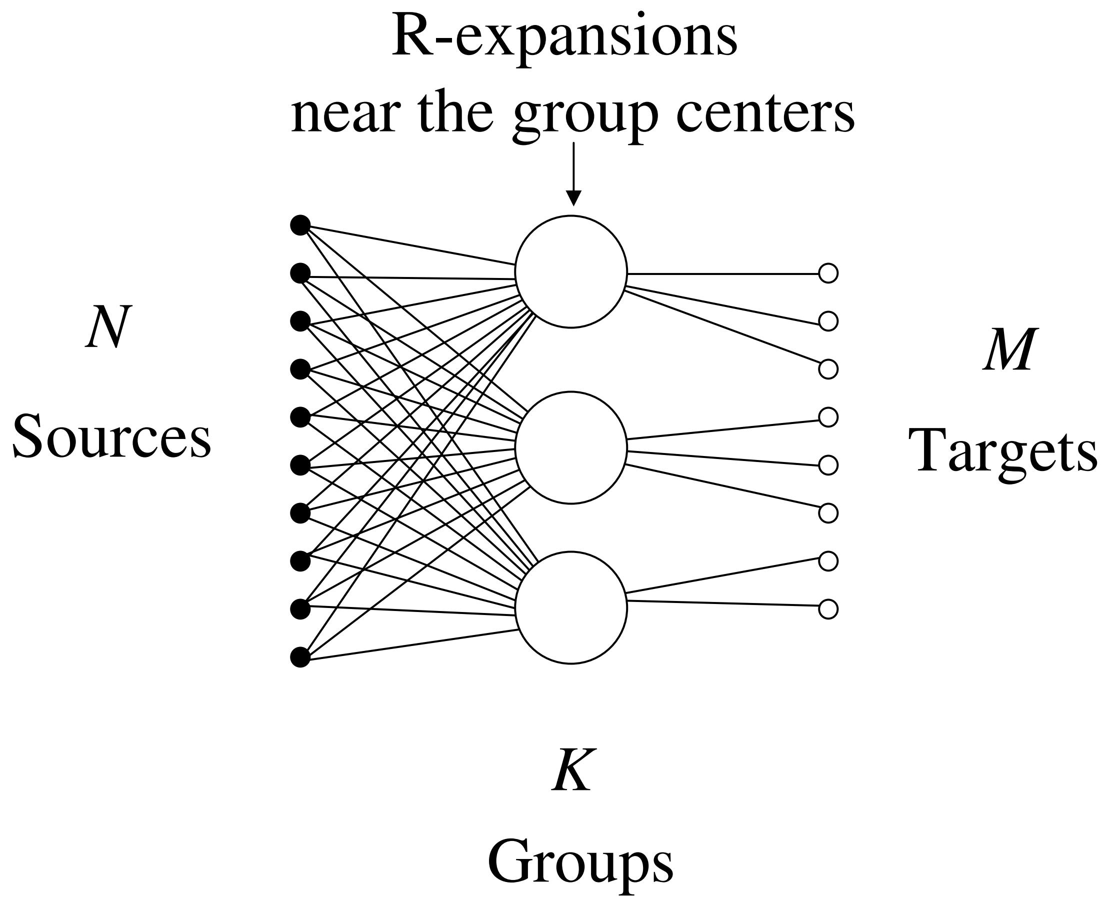


Complexity: $O(p(N-q)+pM+qM)$

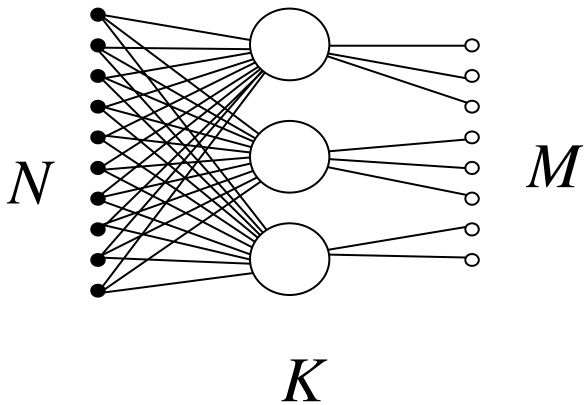
Natural spatial grouping (grouping with respect to the target set)



Natural spatial grouping (continuation)



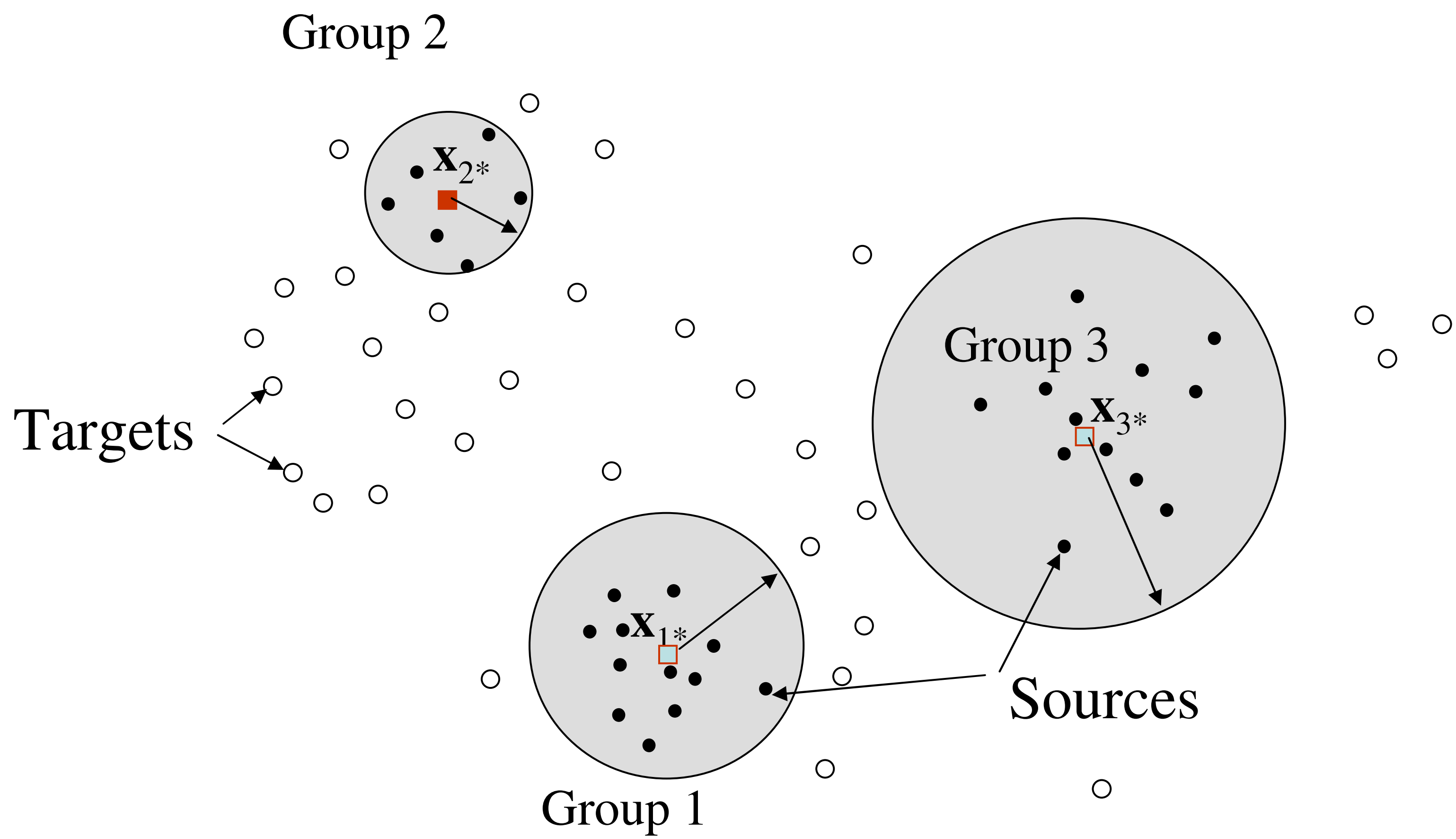
Natural spatial grouping (continuation)



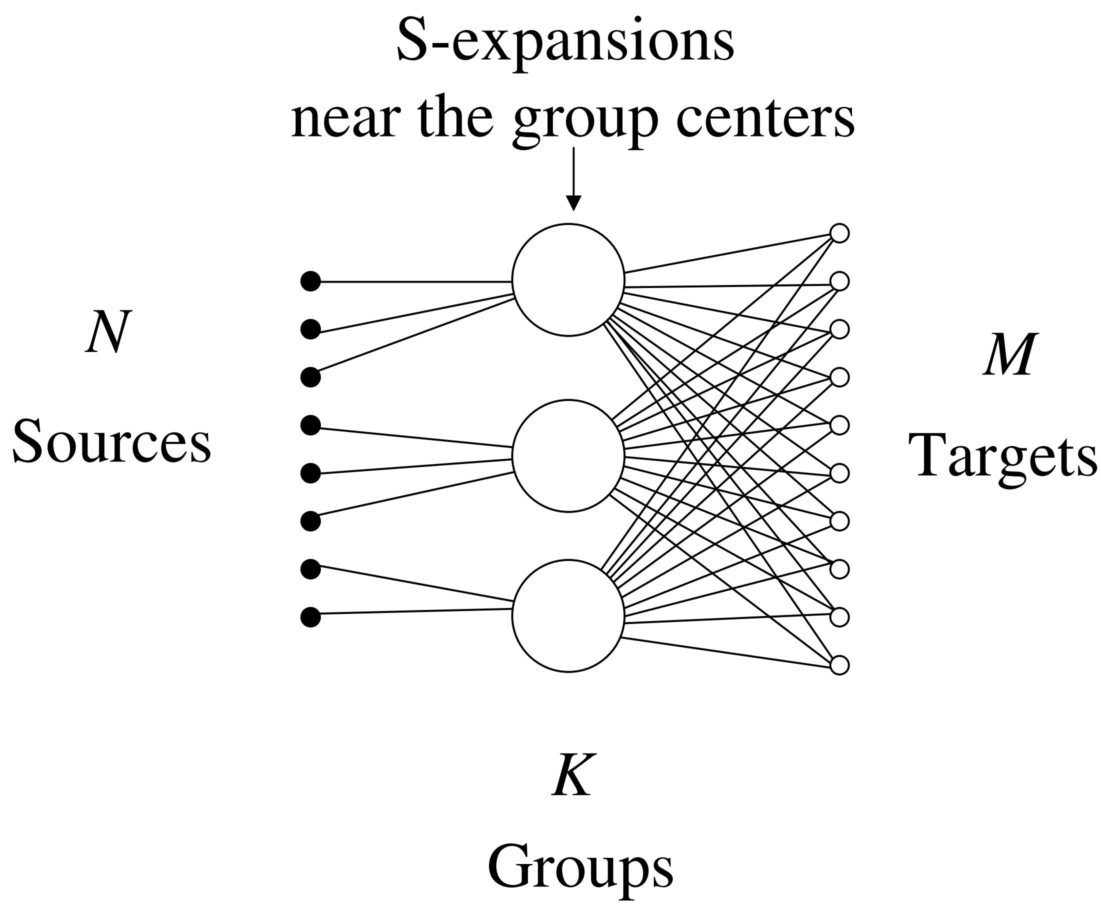
Asymptotic Complexity:

1) Let the R-expansion has p-terms;
2) To build them for K groups we need $O(pNK)$ operations.
3) To evaluate them we need $O(pM)$ operations.
4) Total complexity: $O(p(NK+M))$.
5) Better then the Straightforward method, if $pK<<M$. In this case $p(NK+M)<<NM$

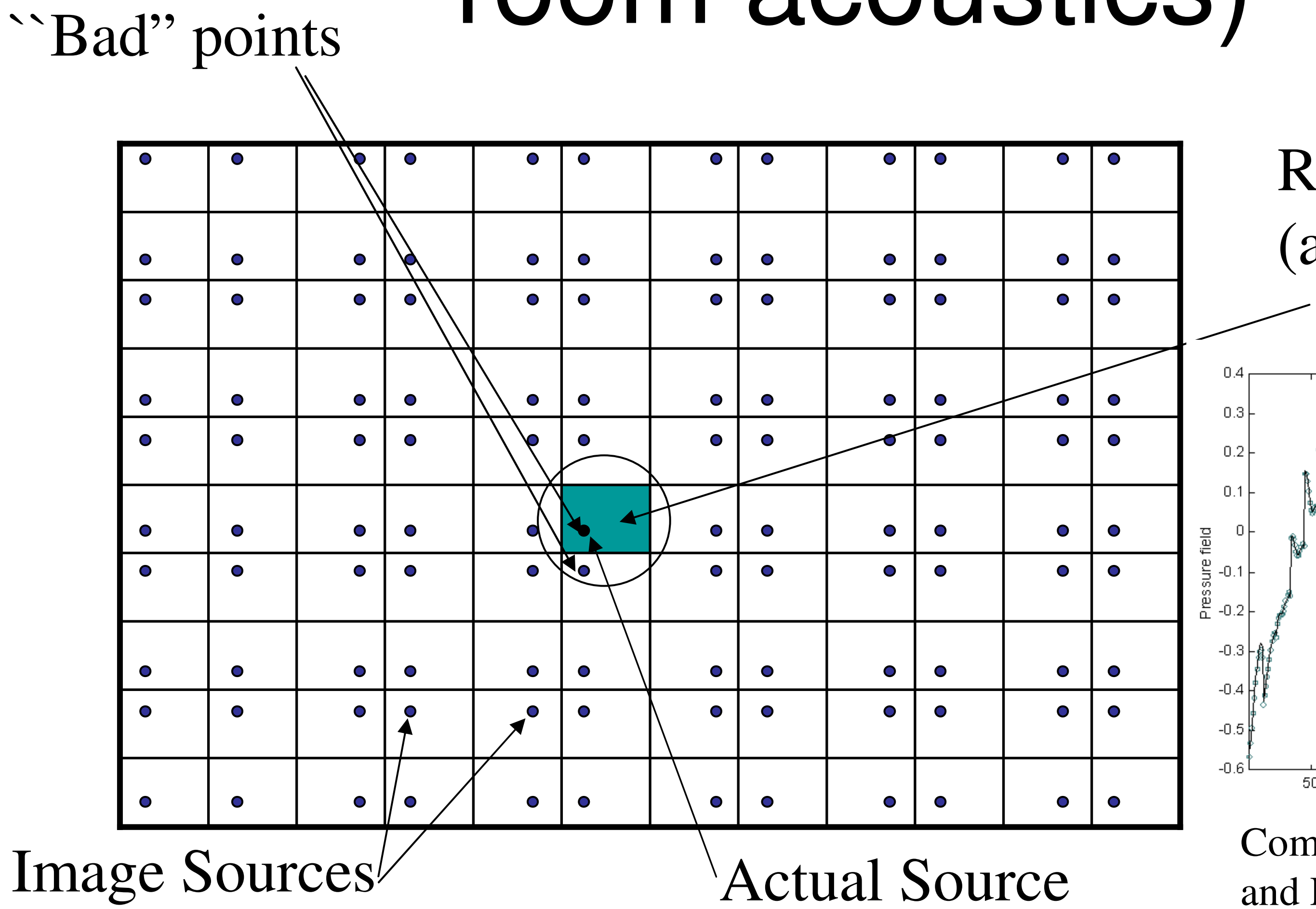
Natural spatial grouping for (Grouping with respect to the source set)

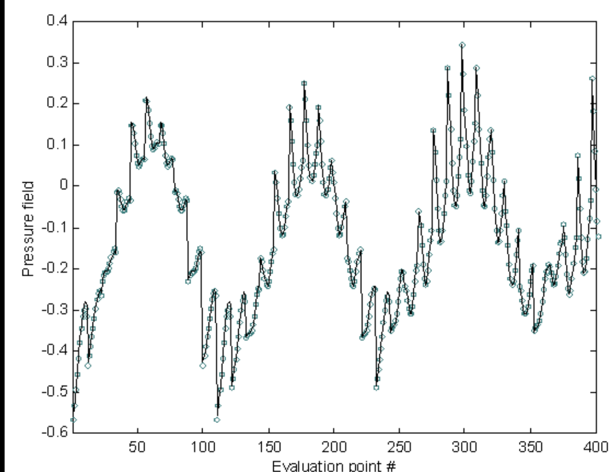


Natural spatial grouping (continuation)



Outliers (an example from room acoustics)





Comparison of Straightforward and Fast Solutions

(R. Duraiswami, N.A. Gumerov, D.N. Zotkin & L.S. Davis, Efficient Evaluation Of Reverberant Sound Fields, 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001).

Outliers (continued)

Universal Recipe: If the number of the outliers is small, then compute their contribution directly.

E.g. if this number is smaller than *p*,
then the outliers do not change the algorithm complexity.

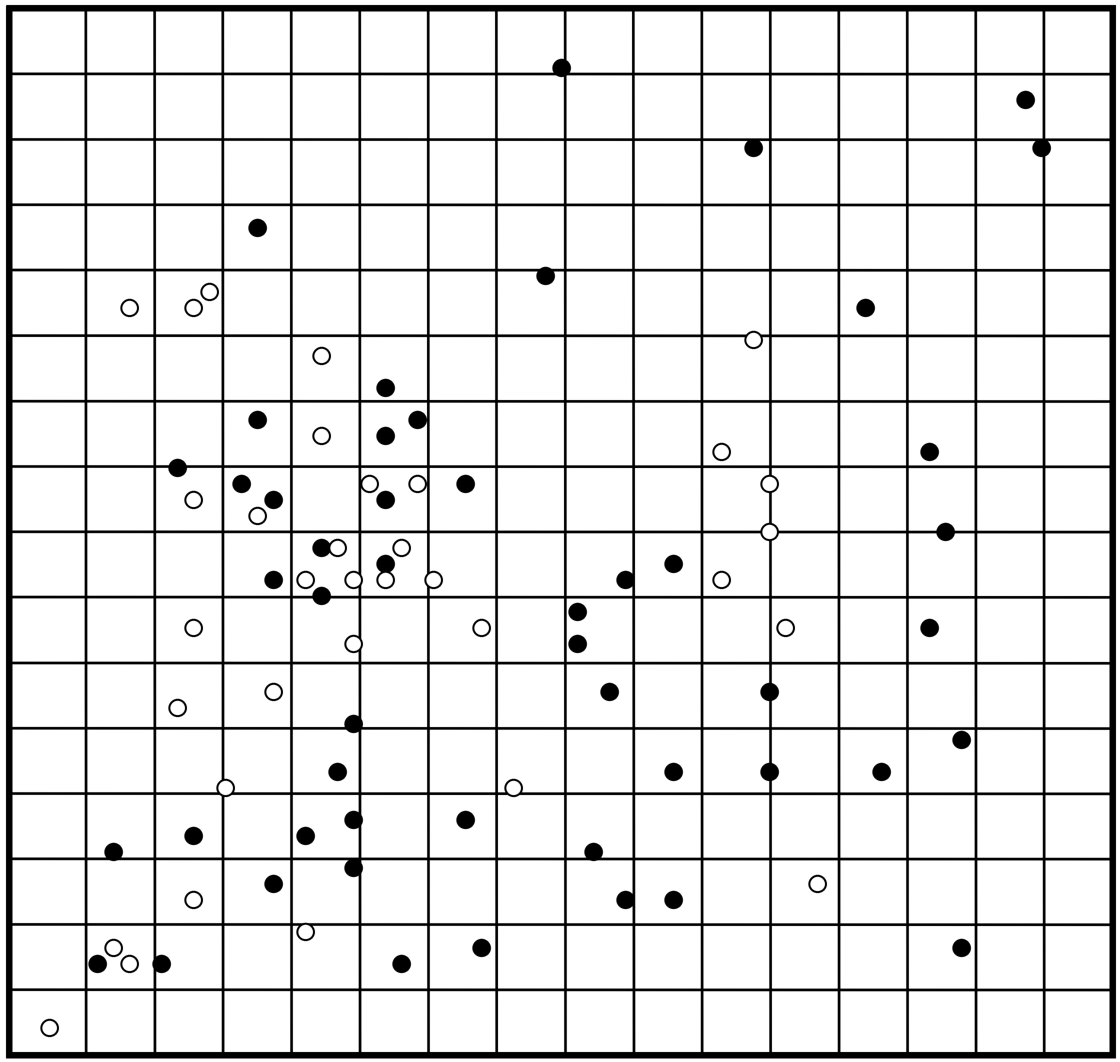
Examples of natural spatial grouping

- Stars (form galaxies, gravity);
- Flow past a body (vortices are grouped in a wake);
- Statistics (clusters of statistical data points);
- People (Organized in groups, cities, etc.);
- Create your own example !

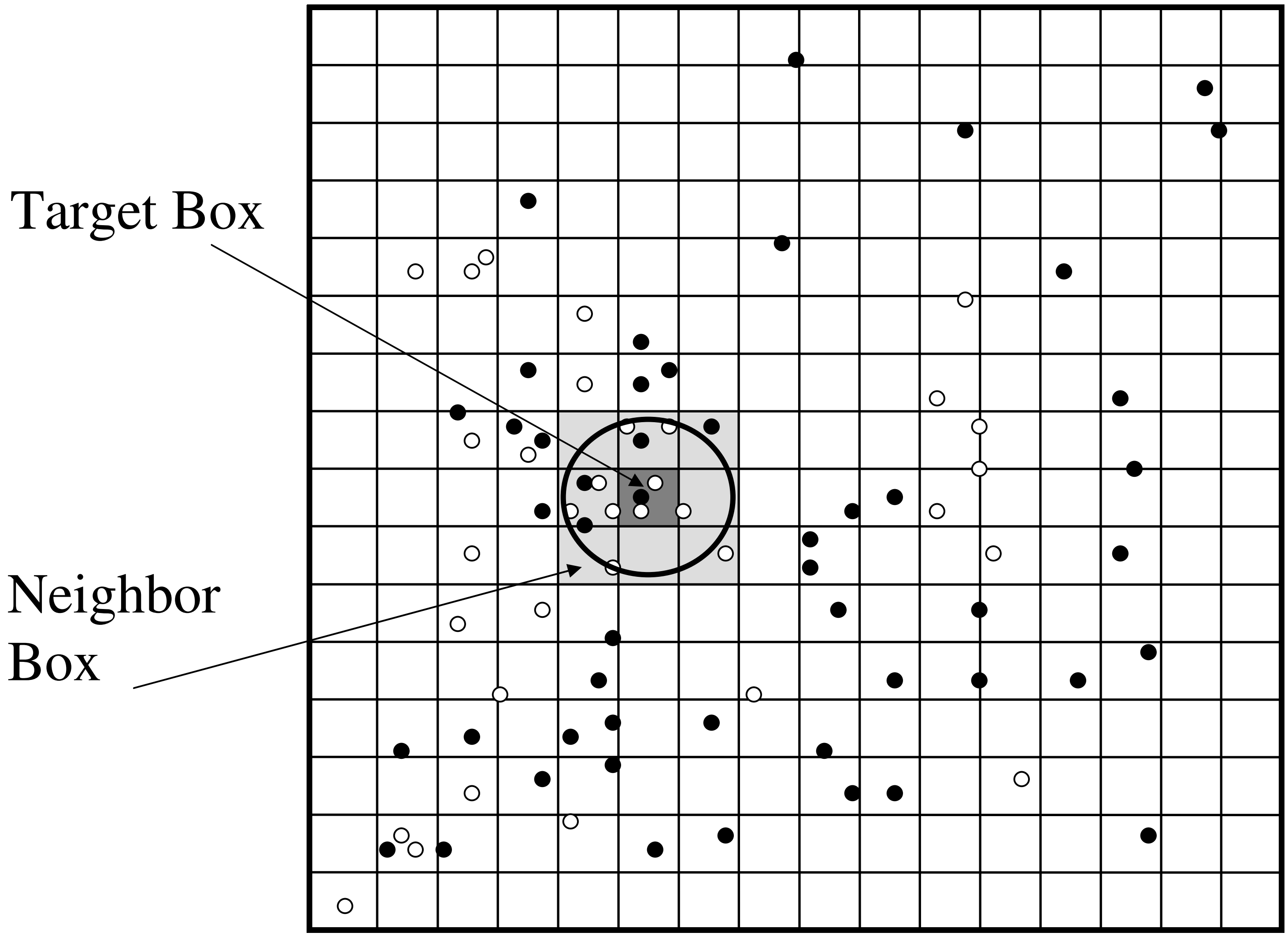
Deficiencies

- Data points may be not naturally grouped;
- Need intelligence to identify the groups: Problem with the algorithms (Artificial Intelligence?)
- Problem dependent.

The Answer is: Space Partitioning



Space partitioning with respect to the target set



Matrix Decomposition

$$\Phi = \Phi_{sparse} + \Phi_{dense}$$

Includes interaction only in the neighborhood of the receiver boxes.

Sparse m-v product computed directly.

Includes interactions only outside the neighborhood of the receiver boxes.

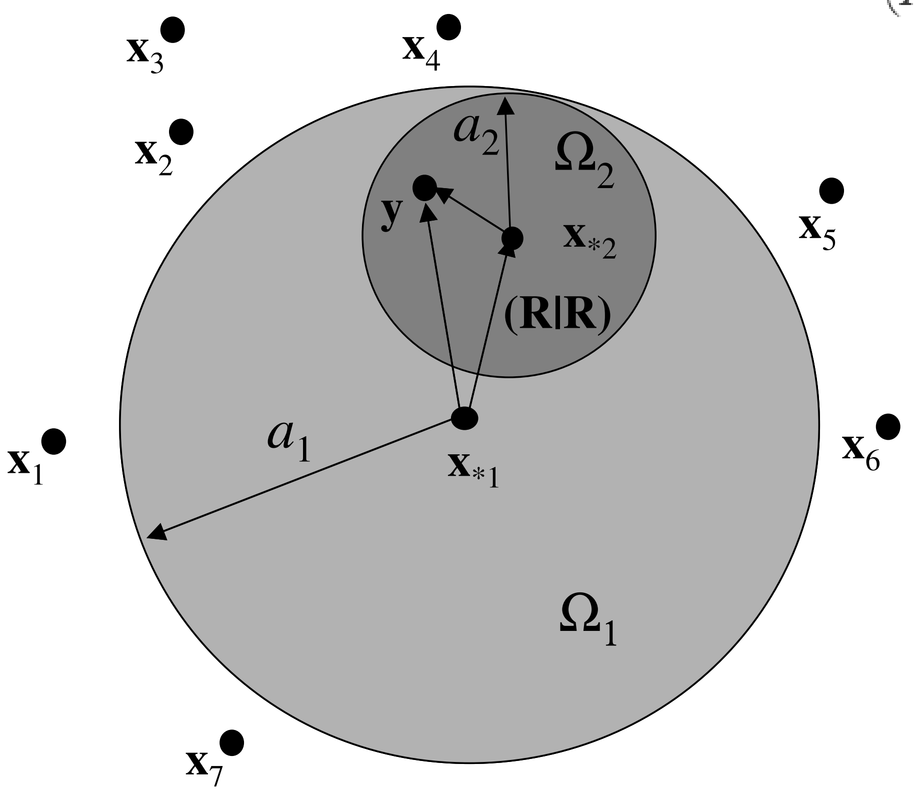
Computed via Factorization tricks (requires space partitioning and function expansions with evaluation of the error bounds)

Key Ideas: Translations

Given coefficients of expansion of some function over some basis about center x_{*1}, translation operators allow us to obtain coefficients of expansions of the same function over a different basis about center x_{*2}.

Translation operators are linear;
Can be represented via matrices;

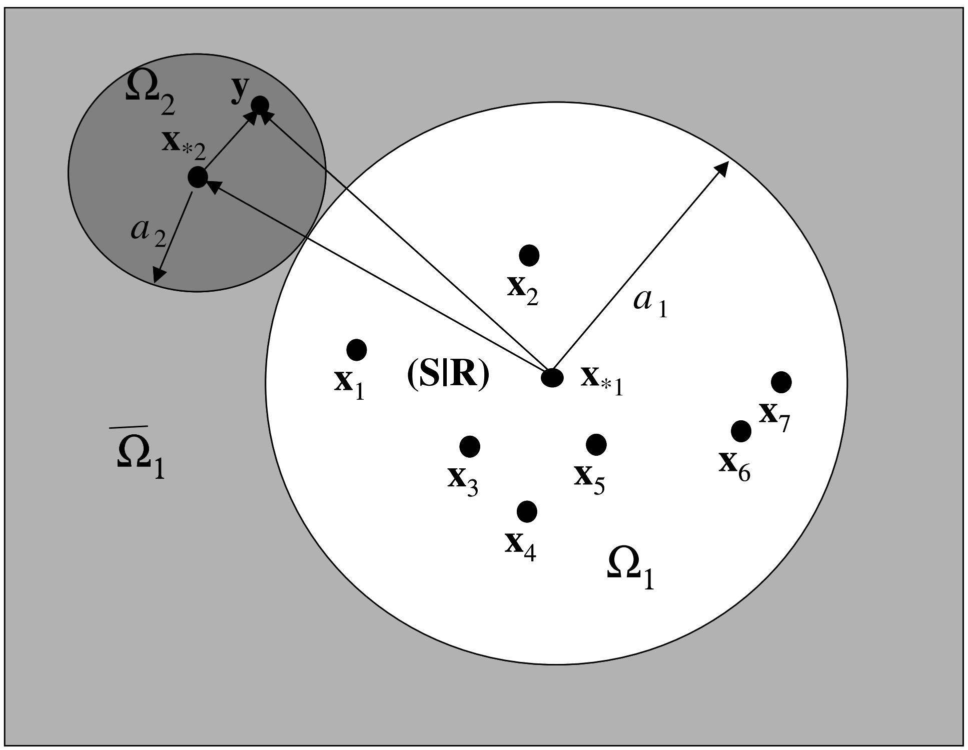
Translations: Local-to-local



$$(R|R)(\mathbf{x}_{*2} - \mathbf{x}_{*1})[\mathbf{C}_1(\mathbf{x}_{*1})] = \mathbf{C}_2(\mathbf{x}_{*2}).$$

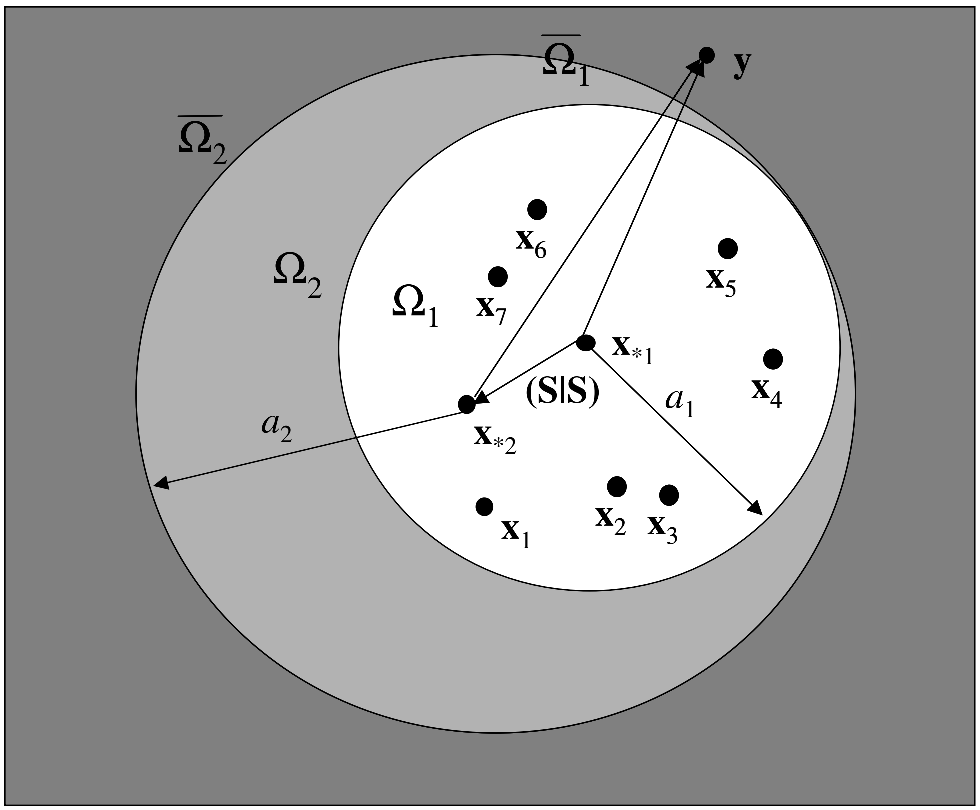
Translations: Multipole-to-local

$$(S|R)(\mathbf{x}_{*2} - \mathbf{x}_{*1})[\mathbf{C}_1(\mathbf{x}_{*1})] = \mathbf{C}_2(\mathbf{x}_{*2}).$$



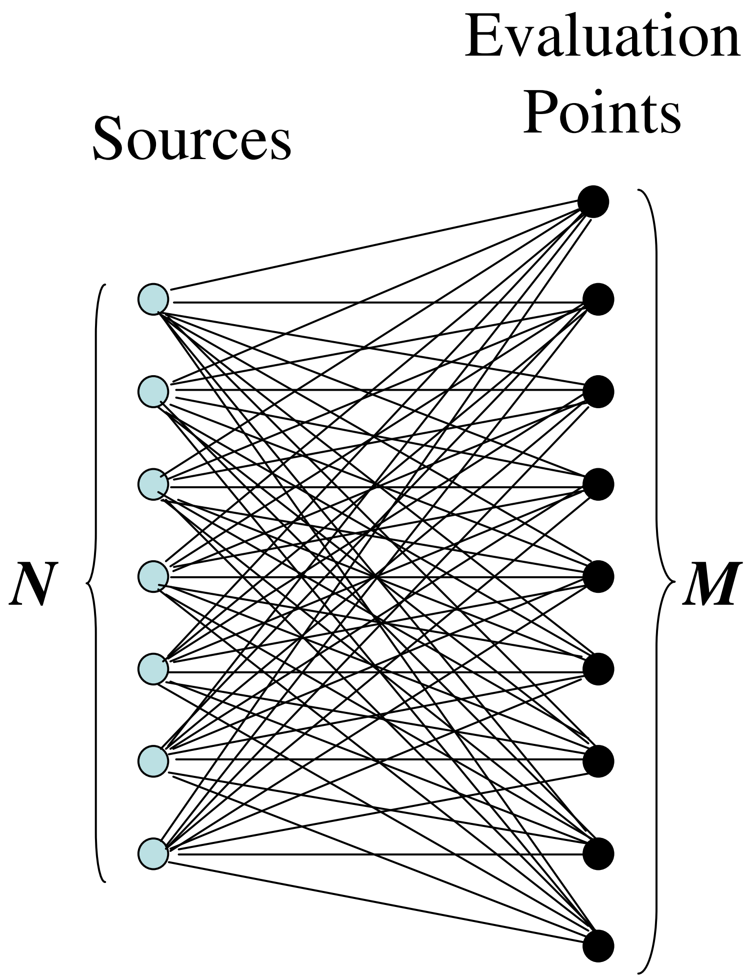
Translations: Multipole-to-multipole

$$(S|S)(\mathbf{x}_{*2} - \mathbf{x}_{*1})[\mathbf{C}_1(\mathbf{x}_{*1})] = \mathbf{C}_2(\mathbf{x}_{*2}).$$



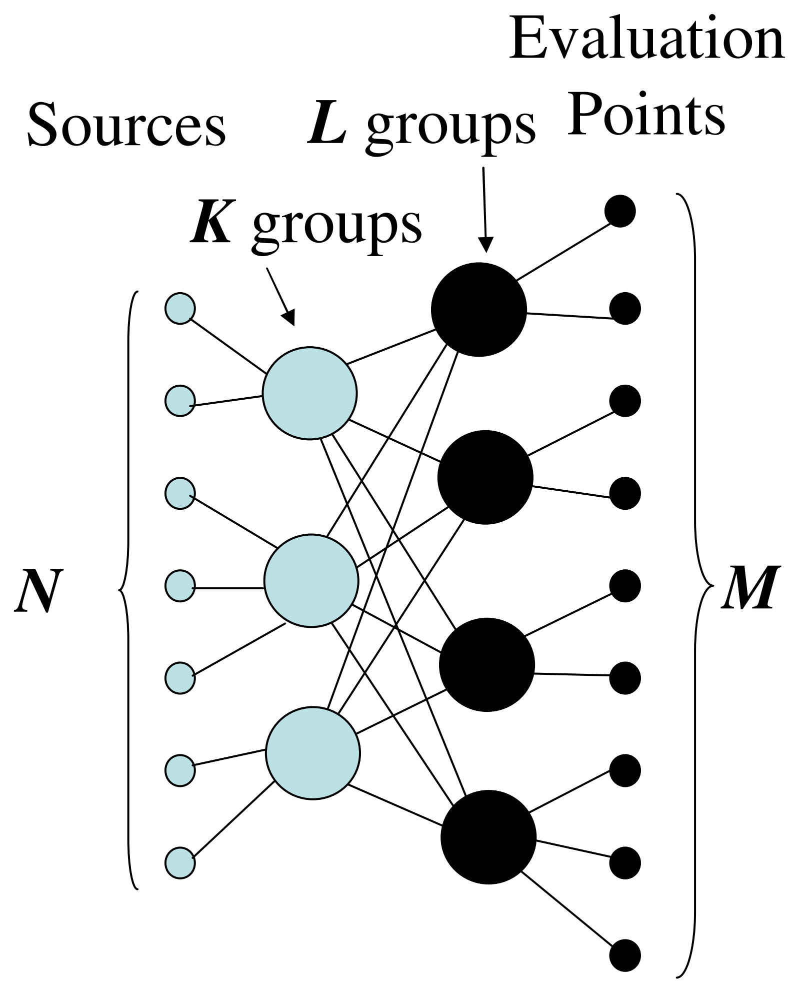
Idea of a Single Level FMM

Standard algorithm



Total number of operations: ***O(NM)***

SLFMM

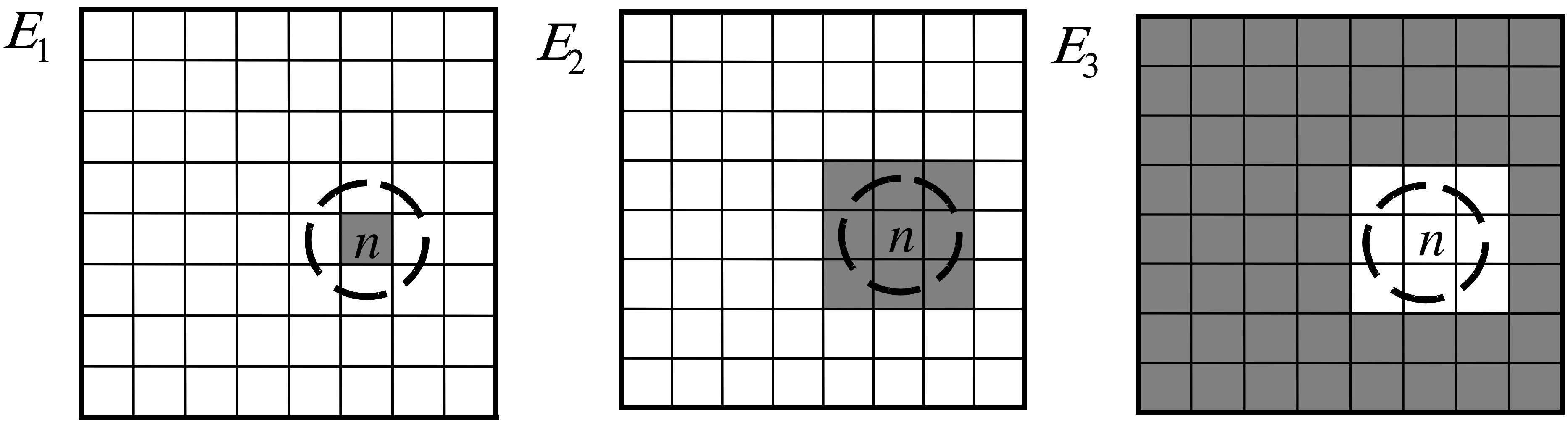


Total number of operations: ***O(N+M+KL)***

Spatial Domains

Potentials due to sources in these spatial domains

$\Phi_1^{(n)}(\mathbf{y})$ $\Phi_2^{(n)}(\mathbf{y})$ $\Phi_3^{(n)}(\mathbf{y})$



$I_1(n) = n$

$I_2(n) = \{Neighbors(n)\} \cup n$

$I_3(n) = \{All\ boxes\} \setminus I_2(n)$

Boxes with these numbers belong to these spatial domains

Definition of potentials

$$\Phi_1^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_1(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_2^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_2(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_3^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_3(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

Since domains $E_2(n)$ and $E_3(n)$ are complimentary:

$$\Phi(\mathbf{y}) = \sum_{i=1}^{N} u_i \Phi(\mathbf{y}, \mathbf{x}_i) = \sum_{\mathbf{x}_i \in E_2(n) \cup E_3(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i) = \Phi_2^{(n)}(\mathbf{y}) + \Phi_3^{(n)}(\mathbf{y})$$

for arbitrary n.

SLFMM Algorithm

Step 1. Generate S-expansion coefficients for each box

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

loop over all non-empty source boxes

For $n \in NonEmptySource$

Get $\mathbf{x}_c^{(n)}$, the center of the box;

$\mathbf{C}^{(n)} = \mathbf{0}$;

For $\mathbf{x}_i \in E_1(n)$ — loop over all sources in the box

Get $\mathbf{B}(\mathbf{x}_i, \mathbf{x}_c^{(n)})$, the S-expansion coefficients near the center of the box;

$\mathbf{C}^{(n)} = \mathbf{C}^{(n)} + u_i\mathbf{B}(\mathbf{x}_i, \mathbf{x}_c^{(n)})$;

End;

End;

Implementation can be different!
All we need is to get $\mathbf{C}^{(n)}$.

SLFMM Algorithm
Step 2. (S|R)-translate expansion coefficients

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

$$\mathbf{D}^{(n)} = \sum_{m \in I_3(n)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)}\right)\mathbf{C}^{(m)}.$$

loop over all non-empty evaluation boxes

For $n \in$ *NonEmptyEvaluation*

Get $\mathbf{x}_c^{(n)}$, the center of the box;

$\mathbf{D}^{(n)} = \mathbf{0}$;

loop over all non-empty source boxes outside the neighborhood of the *n*-th box

For $m \in I_3(n)$

Get $\mathbf{x}_c^{(m)}$, the center of the box;

$\mathbf{D}^{(n)} = \mathbf{D}^{(n)} + (\mathbf{S}|\mathbf{R})(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)})\ \mathbf{C}^{(m)}$;

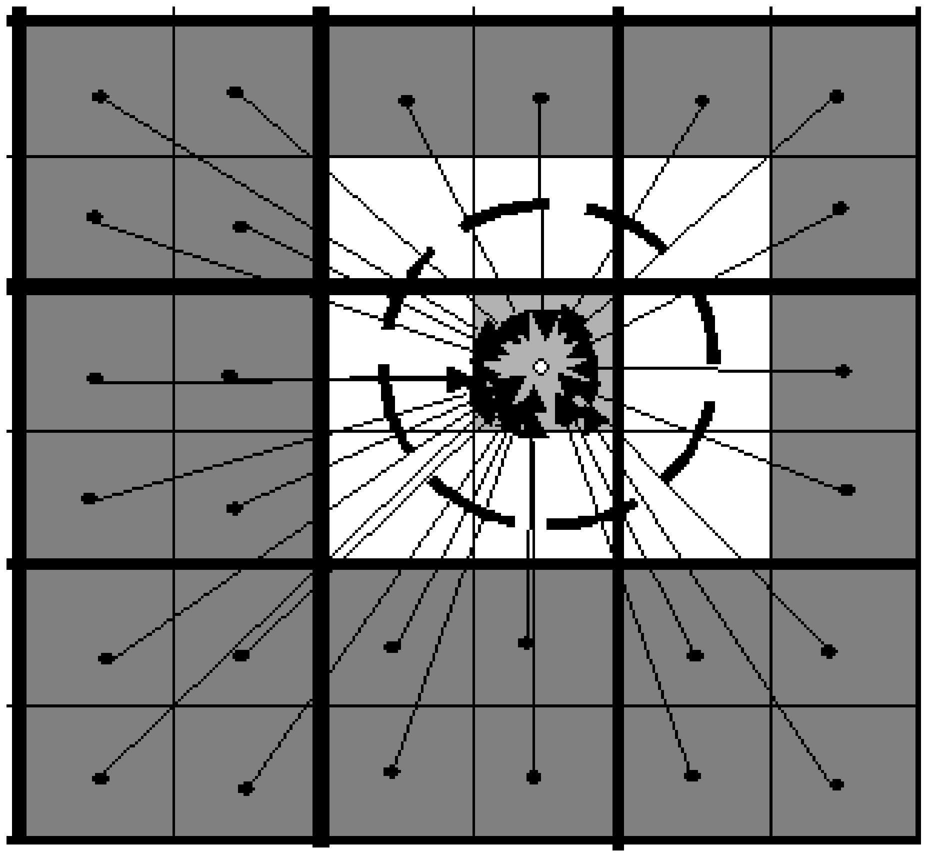
End;

End;

Implementation can be different!

All we need is to get $\mathbf{D}^{(n)}$.

S|R-translation



SLFMM Algorithm

Step 3. Final Summation

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

$For\ \ n \in NonEmptyEvaluation$ ← loop over all boxes containing evaluation points

Get $\mathbf{x}_c^{(n)}$, the center of the box;

$For\ \mathbf{y}_j \in E_1(n)$ ← loop over all evaluation points in the box

$v_j = \mathbf{D}^{(n)} \circ \mathbf{R}(\mathbf{y}_j - \mathbf{x}_c^{(n)})$;

$For\ \mathbf{x}_i \in E_2(n)$ ← loop over all sources in the neighborhood of the *n*-th box

$v_j = v_j + \Phi(\mathbf{y}_j , \mathbf{x}_i)$;

End;

End;

End;

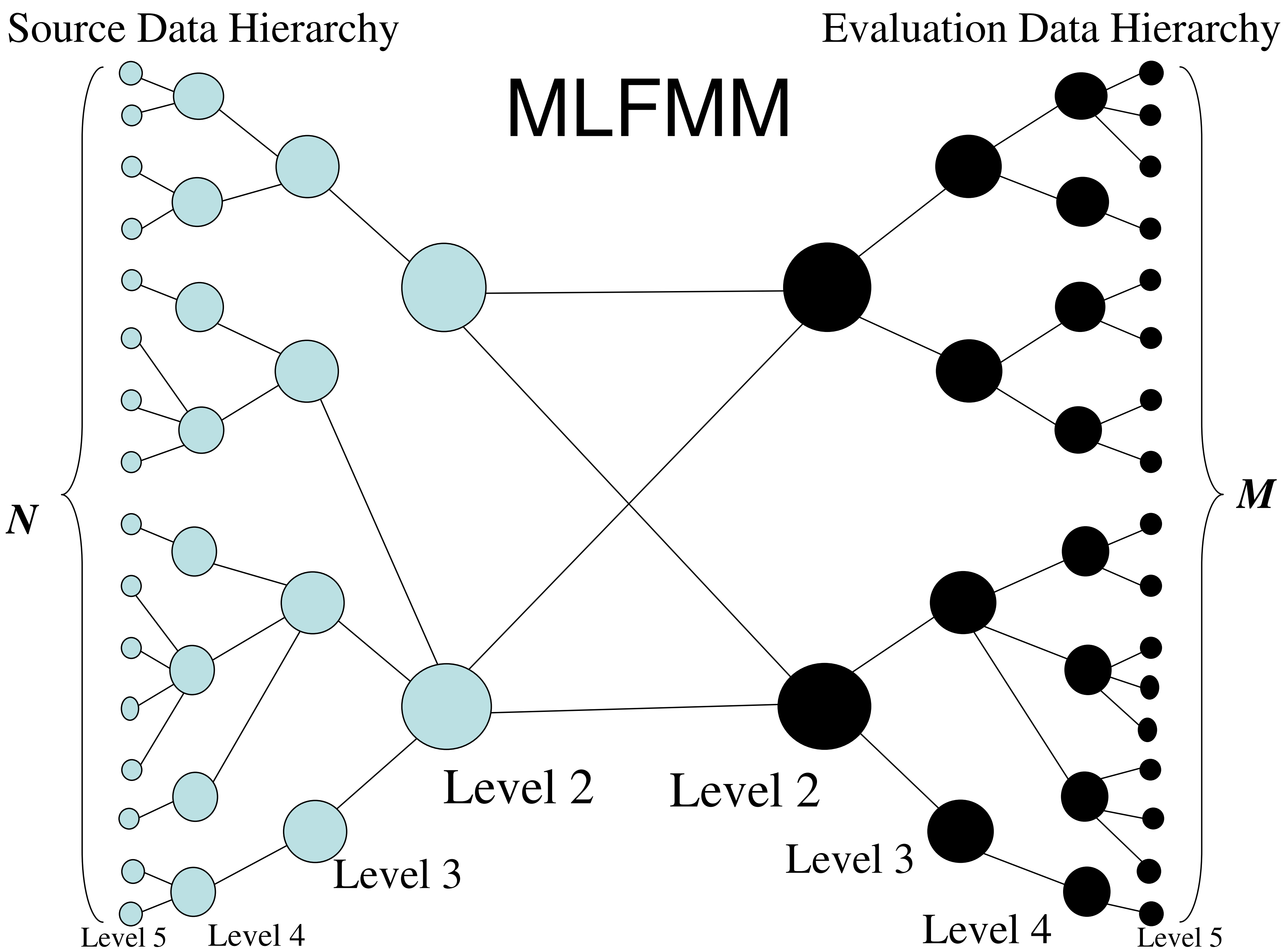
Implementation can be different!
All we need is to get v_j

SLFMM

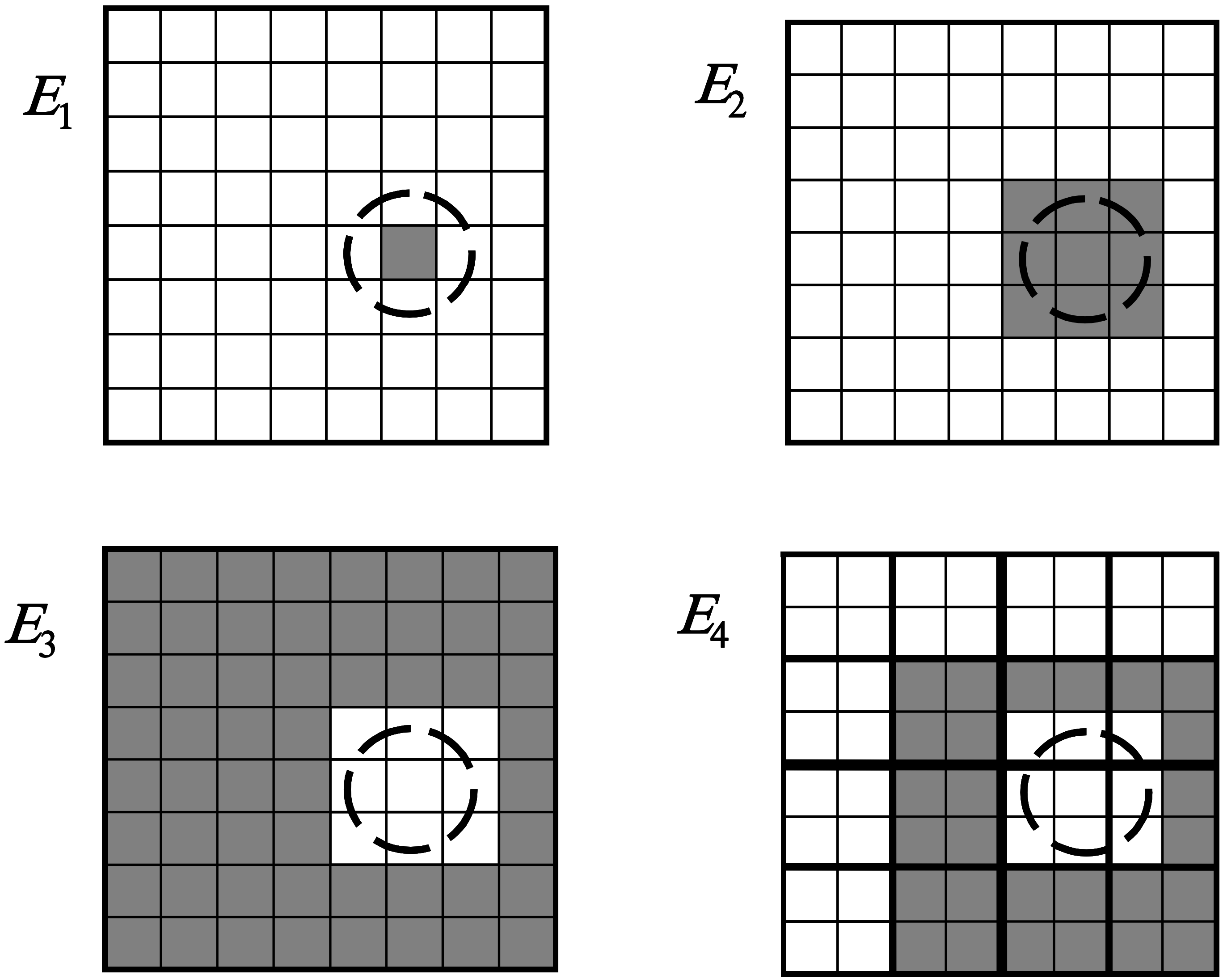
Unfortunately, this algorithm has complexity $O(N^{4/3})$

MLFMM

This algorithm has complexity O(N)

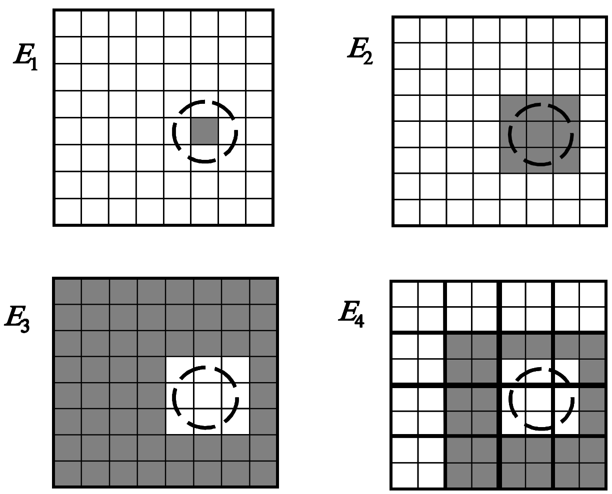

Source Data Hierarchy
Evaluation Data Hierarchy
MLFMM
N
M
Level 2
Level 2
Level 3
Level 3
Level 4
Level 4
Level 5
Level 5

Hierarchical Spatial Domains



Hierarchical Potentials (Functions)

Based on these domains for each box the following functions (potentials) are defined:



$$\Phi_1^{(n,l)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_1(n,l)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_2^{(n,l)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_2(n,l)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_3^{(n,l)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_3(n,l)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_4^{(n,l)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_4(n,l)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

Note that since domains $E_2(n,l)$ and $E_3(n,l)$ are complimentary, and

$$\Phi(\mathbf{y}) = \Phi_2^{(n,l)}(\mathbf{y}) + \Phi_3^{(n,l)}(\mathbf{y})$$

for arbitrary l and n.

The MLFMM Algorithm (Solver)

- ``Build Function” or ``Build Potential” means find its expansion coefficients over some basis;
- The MLFMM Algorithm (we also call it sometimes ``Regular FMM”) consists of
  - Upward Pass;
  - Downward Pass;
  - Final Summation;

Upward Pass. Step 1.

Step 1. At the finest level of space subdivision, build far-field expansion for sources inside each non-empty box of set $\mathbb{X}$ near the center of that box $\mathbf{x}_c^{(n,L)}$:

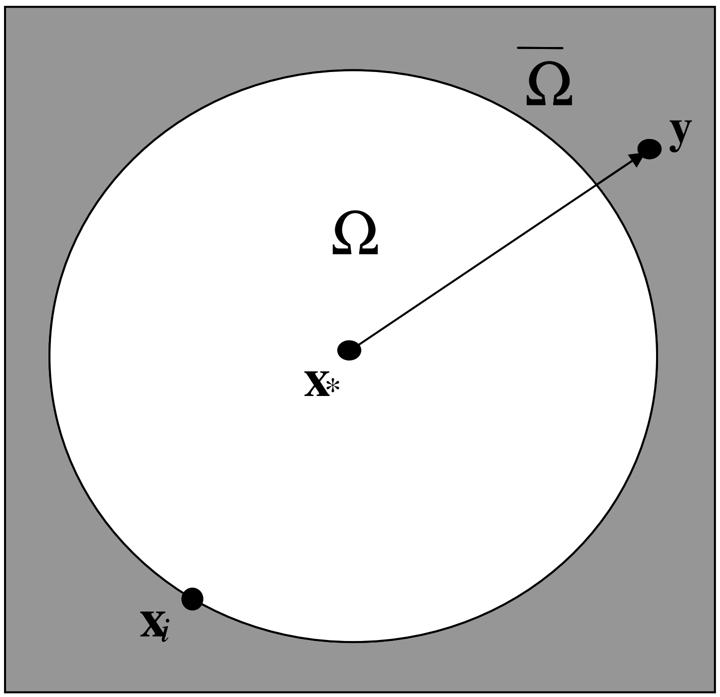
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,L)}),$$

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

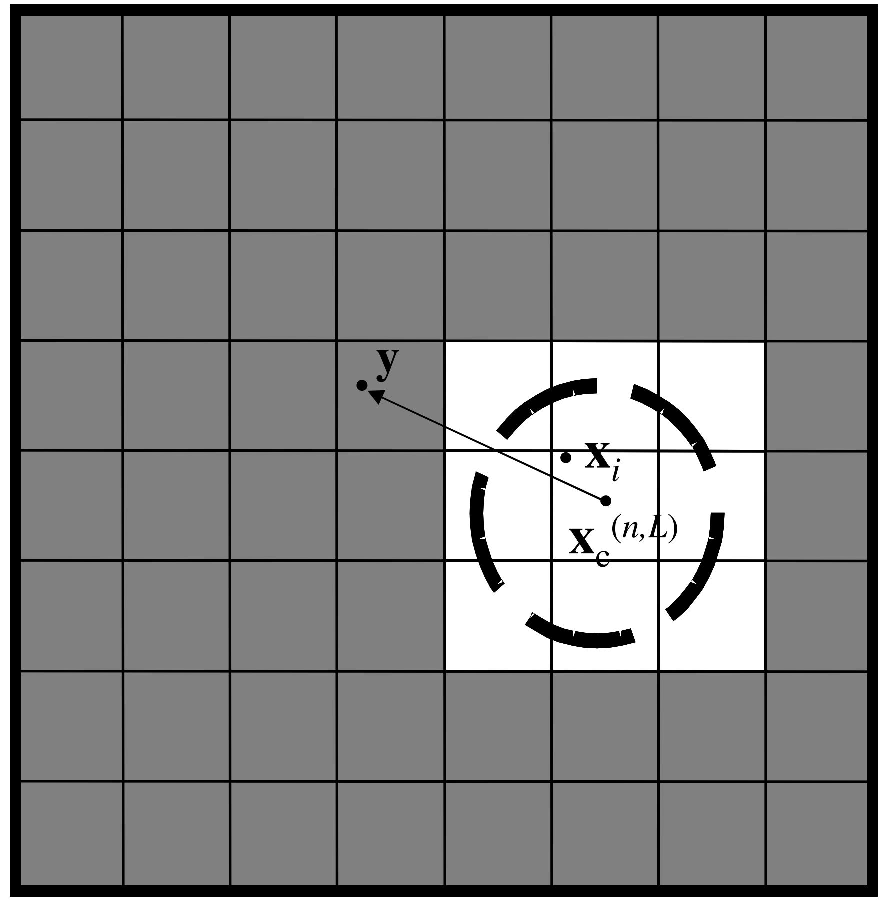
In the algorithm this means generation of the expansion coefficients $\mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$ and determination of $\mathbf{C}^{(n,L)}$ for each box. If at the finest level each non-empty box contains only one source $\mathbf{x}_i$, then for such box $\mathbf{C}^{(n,L)} = u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$. Note that this expansion for nth box is valid in domain $E_3(n,L)$. If the nth box is empty $\Phi_1^{(n,L)}(\mathbf{y}) = 0$ (or $\mathbf{C}^{(n,L)} = 0$) for such a box. There is no need to keep zero $\mathbf{C}^{(n,L)}$ in the memory, since the empty boxes can be skipped in the procedure.

Upward Pass. Step 1.

S-expansion valid in $\overline{\Omega}$



E_3



S-expansion valid in $E_3(n,L)$

Upward Pass. Step 2.

Step 2. For $l = L-1, ..., 2$ recursively form $\Phi_1^{(n,l)}(\mathbf{y})$ (in other words determine expansion coefficients of this function) by reexpansion of $\Phi_1^{(Children(n),l+1)}(\mathbf{y})$ near the center of the parent box and summing up of contribution of all children boxes:

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

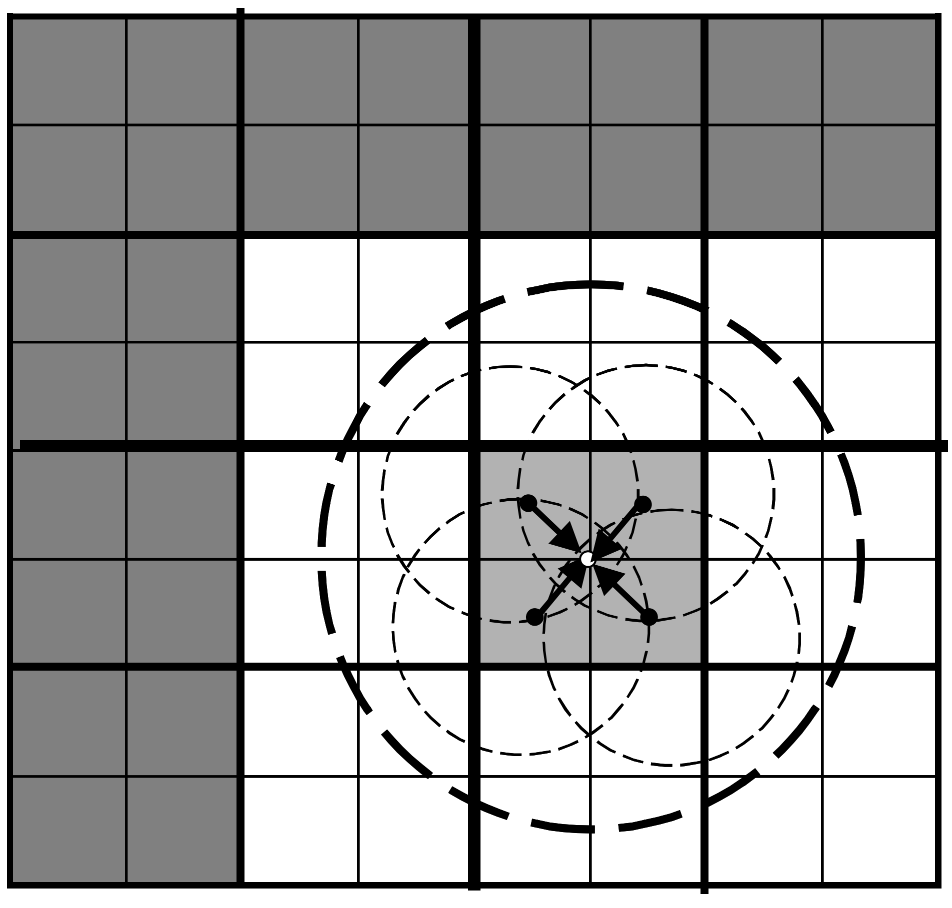
$$\mathbf{C}^{(n,l)} = \sum_{n' \in Children(n)} (\mathbf{S}|\mathbf{S})\left(\mathbf{x}_c^{(n',l+1)} - \mathbf{x}_c^{(n,l)}\right)\mathbf{C}^{(n',l+1)}.$$

For the nth box this expansion is valid in domain $E_3(n,l)$ which is a subdomain, where far-to-far translation is applicable. The set $Children(n)$ has 2^d entries, and summation over empty boxes of set $\mathbb{X}$ can be skipped (anyway for such boxes $\mathbf{C}^{(n',l+1)} = 0$).

Upward Pass. Step 2.

S|S-translation.

Build potential for the parent box (find its S-expansion).



Result of the Upward Pass

In the entire hierarchy of boxes containing *sources* S-expansion coefficients for potentials due to *sources* in each box (domains E_1) are found. Expansions are valid in E_3 domains.

Downward Pass. Step 1.

Step 1. Steps 1 and 2 should be performed recursively for levels $l = 2, \ldots L$ of space subdivision. At this step form coefficients of regular expansion for function $\Phi_4^{(n,l)}(\mathbf{y})$. To build local expansion near the center of each box at level l coefficients $\mathbf{C}^{(m,l)}, m \in I_4(n,l)$ should be $(\mathbf{S}|\mathbf{R})$- translated to the center of this box. So we have

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

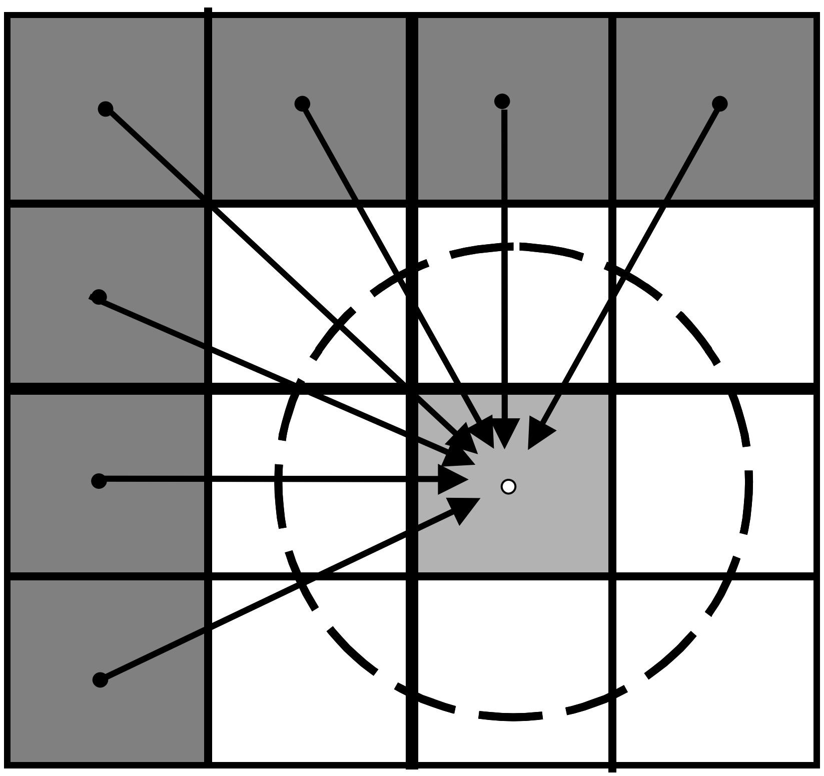
$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

Since each box of level l is separated from boxes of $I_4(n,l)$ by a sphere drown near its center, then the far-to-local translation is applicable. Note that summation over empty boxes $m \in I_4(n,l)$ of set X can be skipped.

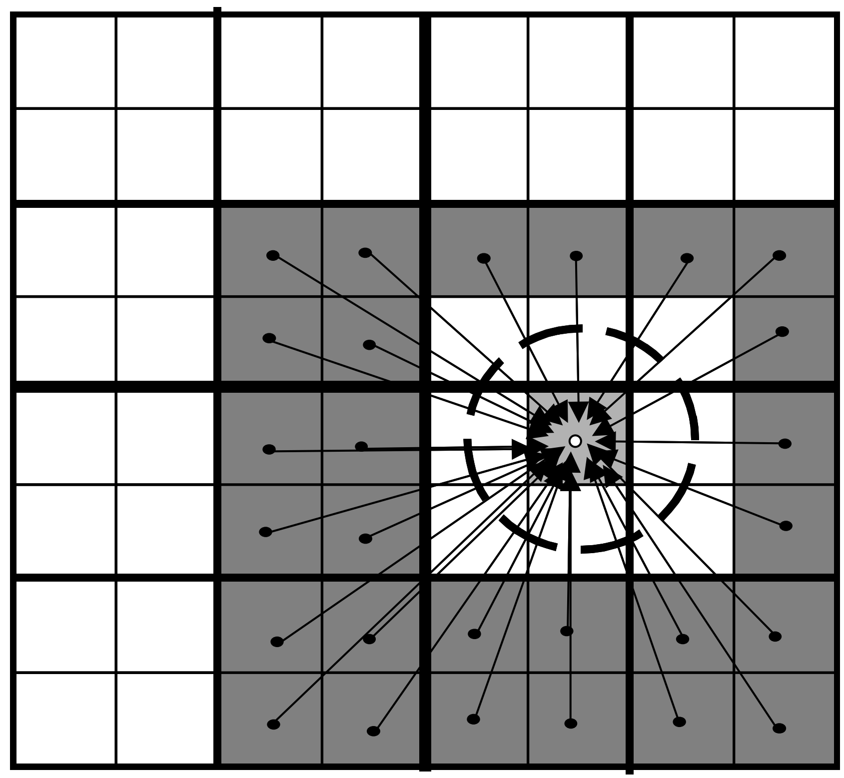
Note that this is conversion from the Source Hierarchy to Evaluation Hierarchy!

Downward Pass. Step 1.

Level 2:

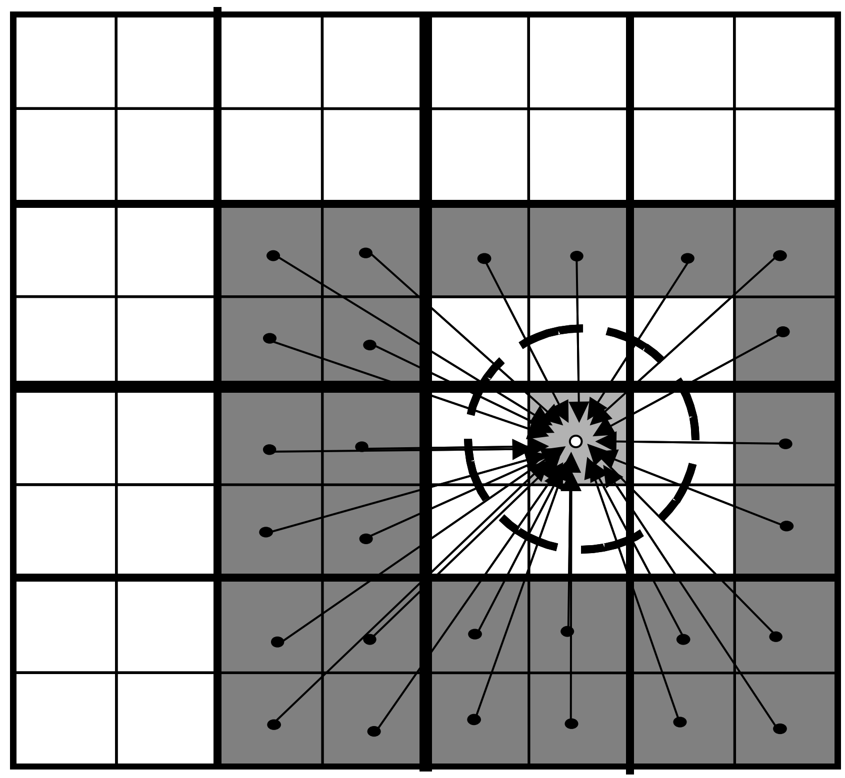


Level 3:



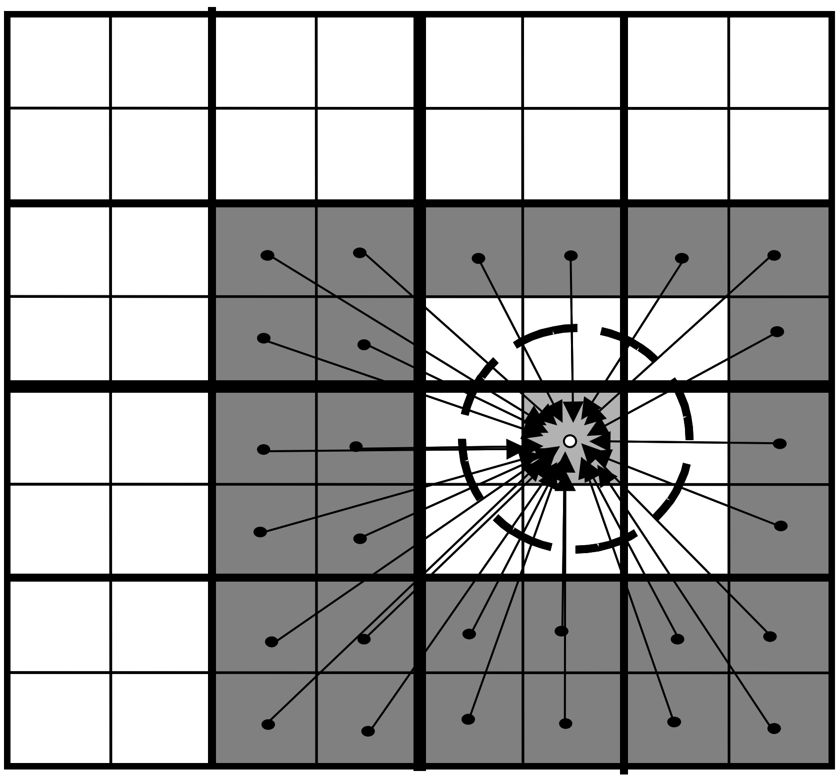
Downward Pass. Step 1.

THIS MIGHT BE
THE MOST EXPENSIVE
STEP OF THE ALGORITHM



Downward Pass. Step 1.

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



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

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

$d = 3: \quad P_4 = 189$

Exponential Growth

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

(far from the domain boundaries)

It is worth to think about optimizations

Downward Pass. Step 2.

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:

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

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

Downward Pass. Step 2.

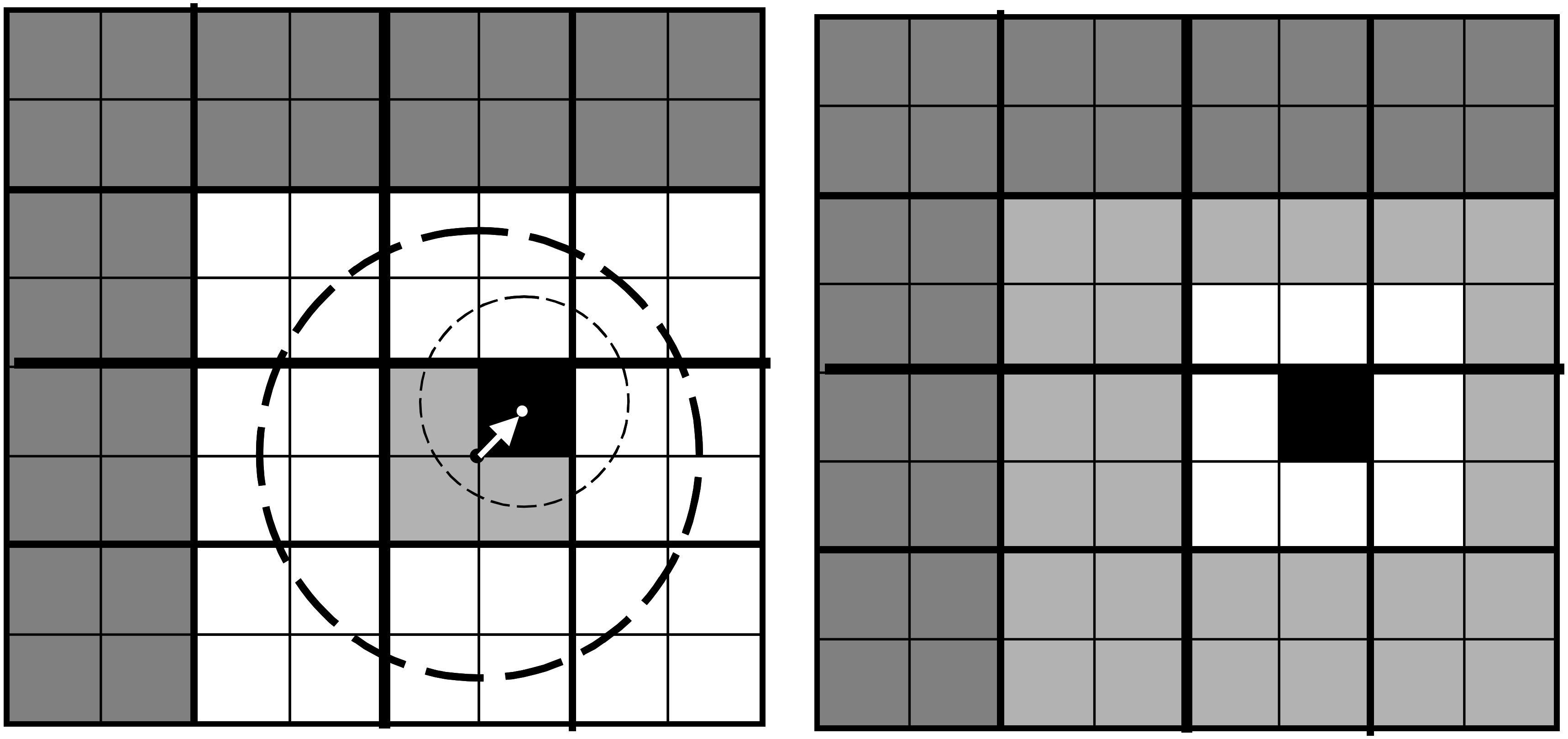


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

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

Result of the Downward Pass

In the entire hierarchy of boxes containing *evaluation points* R-expansion coefficients for potentials due to *sources* outside each *evaluation point* neighborhood (domains E_3) are found. Expansions are valid in E_1 domains.

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