Key Ideas of the FMM

Nail Gumerov &
Ramani Duraiswami
UMIACS
[gumerov][ramani]@umiacs.umd.edu

Content

- Summation Problems
- Factorization (Middleman Method)
- Space Partitioning (Modified Middleman Method)
- Translations (Single Level FMM)
- Hierarchical Space Partitioning (Multilevel FMM)

Summation Problems

Matrix-Vector Multiplication

Compute matrix vector product

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

or

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

where

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

or

$$\mathbf{\Phi} = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \ldots & \Phi_{1N} \\ \Phi_{21} & \Phi_{22} & \ldots & \Phi_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ \Phi_{M1} & \Phi_{M2} & \ldots & \Phi_{MN} \end{pmatrix} = \begin{pmatrix} \Phi(\mathbf{y}_1, \mathbf{x}_1) & \Phi(\mathbf{y}_1, \mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_1, \mathbf{x}_N) \\ \Phi(\mathbf{y}_2, \mathbf{x}_1) & \Phi(\mathbf{y}_2, \mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_2, \mathbf{x}_N) \\ \ldots & \ldots & \ldots & \ldots \\ \Phi(\mathbf{y}_M, \mathbf{x}_1) & \Phi(\mathbf{y}_M, \mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_M, \mathbf{x}_N) \end{pmatrix}.$$

Generally we have two sets of points in d-dimensions:

$$\text{Sources}: \mathbb{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathbb{R}^d, \quad i = 1, \ldots, N,$$
$$\text{Receivers}: \mathbb{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\}, \quad \mathbf{y}_j \in \mathbb{R}^d, \quad j = 1, \ldots, M,$$

The receivers also can be called "targets" or "evaluation points".

Why $\mathbf{R}^d$?

- d = 1
  - Scalar functions, interpolation, etc.
- d = 2,3
  - Physical problems in 2 and 3 dimensional space
- d = 4
  - 3D Space + time, 3D grayscale images
- d = 5
  - Color 2D images, Motion of 3D grayscale images
- d = 6
  - Color 3D images
- d = 7
  - Motion of 3D color images
- d = arbitrary
  - d-parametric spaces, statistics, database search procedures

Fields (Potentials)

Field (Potential) of a single (*i*th) unit source

$$v(\mathbf{y}) = \sum_{i=1}^{N} u_i \Phi(\mathbf{y}, \mathbf{x}_i), \quad \mathbf{y} \in \mathbb{R}^d,$$

$$v_j = v(\mathbf{y}_j), \quad j = 1, \ldots, M.$$

Field (Potential) of the set of sources of intensities $\{u_i\}$

Fields are continuous!
(Almost everywhere)

CSCAMM FAM04: 04/19/2004
© Duraiswami & Gumerov, 2003-2004

Examples of Fields

- There can be vector or scalar fields (we focus mostly on scalar fields)
- Fields can be *regular* or *singular*

Scalar Fields:

Gravity

(singular at $\mathbf{y} = \mathbf{x}_i$)

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

Monochromatic Wave (k is the wavenumber)

(singular at $\mathbf{y} = \mathbf{x}_i$)

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

Gaussian

(regular everywhere)

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

Vector Field:

3D Velocity field:

(singular at $\mathbf{y} = \mathbf{x}_i$)

$$\Phi(\mathbf{y},\mathbf{x}_i) = \nabla_y \frac{1}{|\mathbf{y}-\mathbf{x}_i|} = \mathbf{i}_1 \frac{\partial}{\partial y_1}\frac{1}{|\mathbf{y}-\mathbf{x}_i|} + \mathbf{i}_2 \frac{\partial}{\partial y_2}\frac{1}{|\mathbf{y}-\mathbf{x}_i|} + \mathbf{i}_3 \frac{\partial}{\partial y_3}\frac{1}{|\mathbf{y}-\mathbf{x}_i|},$$

$$\mathbf{y} = (y_1, y_2, y_3) \in \mathbb{R}^3.$$

Straightforward Computational Complexity:

$O(MN)$ Error: 0 (“machine” precision)

The Fast Multipole Methods look for computation of the same problem with complexity $o(MN)$ and error < prescribed error.

In the case when the error of the FMM does not exceed the machine precision error (for given number of bits) there is no difference between the “exact” and “approximate” solution.

Factorization
“Middleman Method”

Global Factorization

Expansion center

Truncation number

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

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

Expansion coefficients

Basis functions

CSCAMM FAM04: 04/19/2004
© Duraiswami & Gumerov, 2003-2004

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):

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

Complexity: *O*(*MN*)

Factroized:

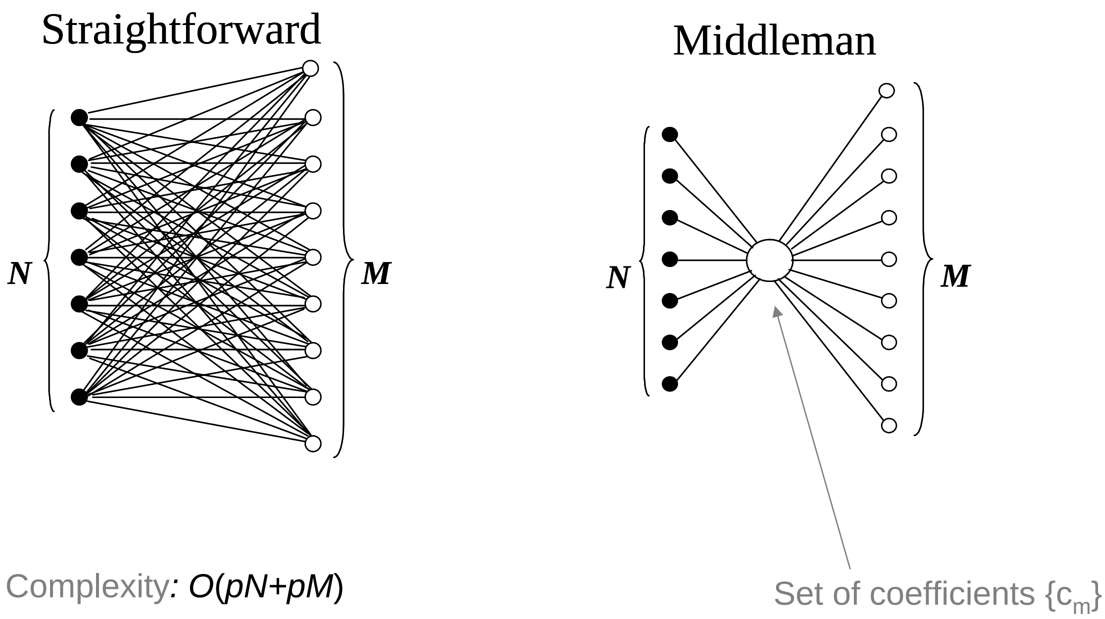
for $m = 0, \ldots, p-1$
    $c_m = 0;$
    for $i = 1, \ldots, N$
        $c_m = c_m + a_m(\mathbf{x}_i - \mathbf{x}_*) u_i;$
    end;
end;

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

Complexity: *O*(*pN*+*pM*)

If *p* << min*(M,N)* then complexity reduces!

Middleman Scheme



Example Problem (1D Gauss Transform)

Compute

$$v_j = \sum_{i=1}^{N} \Phi(y_j, x_i) u_i, \quad j = 1, \ldots, M, \quad \Phi(y, x_i) = e^{-(y-x_i)^2}$$

where x_i, y_j, and u_i are random numbers distributed on [0,1].

Solution:

We have

$$\Phi(y, x_i) = e^{-(y-x_i)^2} = e^{-[y-x_*-(x_i-x_*)]^2} = e^{-(y-x_*)^2} e^{-(x_i-x_*)^2} e^{2(x_i-x_*)(y-x_*)}$$

$$= e^{-(y-x_*)^2} e^{-(x_i-x_*)^2} \left[\sum_{m=0}^{p-1} \frac{2^m (x_i - x_*)^m (y - x_*)^m}{m!} + error_p \right],$$

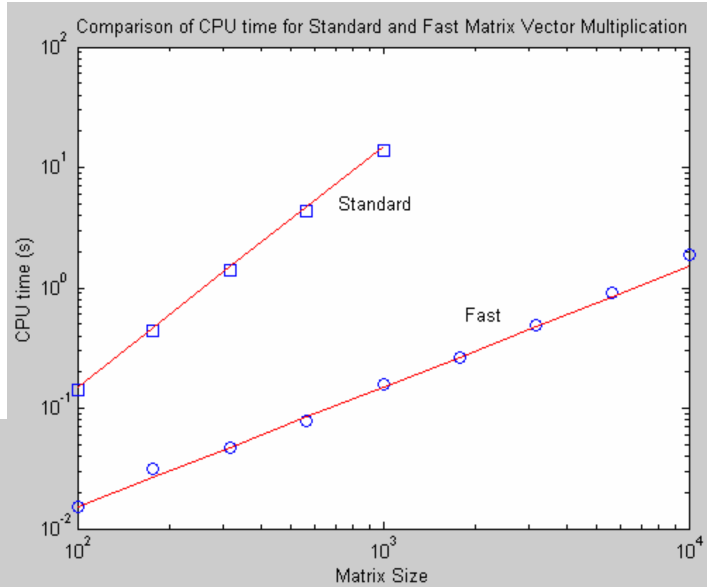
$$|error_p| \leqslant \frac{|y - x_*|^p}{p!} \sup_{0 \leqslant y \leqslant 1} \left| \frac{\partial^p e^{2(x_i-x_*)(y-x_*)}}{\partial y^p} \right| = \frac{2^p |y - x_*|^p |x_i - x_*|^p}{p!} \sup_{0 \leqslant y \leqslant 1} e^{2(x_i-x_*)(y-x_*)}.$$

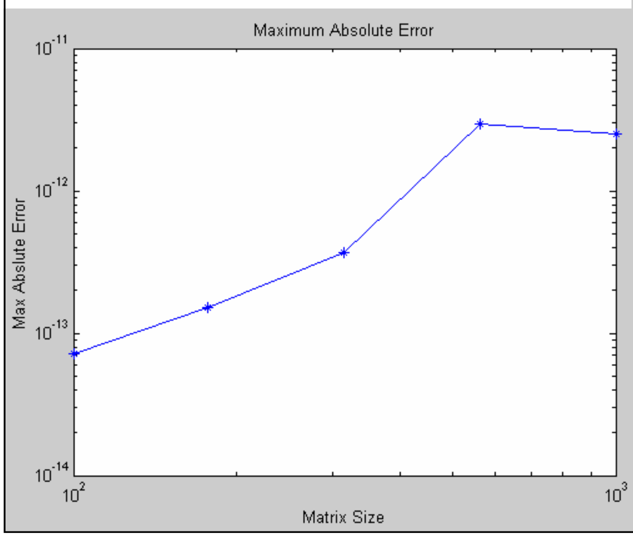
Let us select $x_* = 0.5$, then truncation number $p = 10$ is sufficient for computations with $\epsilon = 10^{-6}$ and $N \leqslant 10^4$. The formula for fast computations will be then

$$v_j = e^{-(y_j-x_*)^2} \sum_{m=0}^{p-1} c_m (y_j - x_*)^m, \quad j = 1, \ldots, M.$$

$$c_m = \frac{2^m}{m!} \sum_{i=1}^{N} e^{-(x_i-x_*)^2} (x_i - x_*)^m u_i.$$

Example Problem





Complexity of the Middleman Method

$$
\begin{aligned}
|error_p| &\leqslant \sigma^{-p}, \\
FMMerror_p &\leqslant \sigma^{-p} N, \\
p &\sim \log \frac{N}{\epsilon}, \\
ComplexityFMM &= O(pN) = O(N \log \frac{N}{\epsilon})
\end{aligned}
$$

© Duraiswami & Gumerov, 2003-2004

Local (Regular) Expansion

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

We call expansion

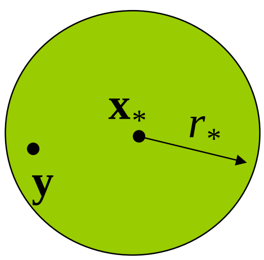
$$\Phi(\mathbf{y},\mathbf{x}_i) = \sum_{m=0}^{\infty} a_m(\mathbf{x}_i,\mathbf{x}_*) R_m(\mathbf{y}-\mathbf{x}_*)$$

local (regular) inside a sphere $|\mathbf{y}-\mathbf{x}_*| < r_*$,

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

Basis Functions

Expansion Coefficients



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

Local Expansion (Example)

$x, y \in \mathbb{R}^1$.

$\Phi(y, x_i) = \frac{1}{y - x_i}$.

Valid for any $|x_i - x_*| > |y - x_*|$

Looking for factorization:

$$\Phi(y,x_i) = \sum_{m=0}^{\infty} a_m(x_i - x_*) R_m(y - x_*).$$

We have

$$\frac{1}{y-x_i} = \frac{1}{y-x_*-(x_i-x_*)} = -\frac{1}{(x_i-x_*)[1-\frac{y-x_*}{x_i-x_*}]} = -\frac{1}{(x_i-x_*)}\left[1-\frac{y-x_*}{x_i-x_*}\right]^{-1}.$$

Geometric progression:

$$(1-\alpha)^{-1} = 1 + \alpha + \alpha^2 + \ldots = \sum_{m=0}^{\infty} \alpha^m, \quad |\alpha| < 1.$$

$$\left[1-\frac{y-x_*}{x_i-x_*}\right]^{-1} = \sum_{m=0}^{\infty} \frac{(y-x_*)^m}{(x_i-x_*)^m}, \quad |y-x_*| < |x_i-x_*|.$$

Choose

$$a_m(x_i - x_*) = -\frac{1}{(x_i-x_*)^{m+1}}, \quad m = 0, 1, \ldots,$$

$$R_m(y - x_*) = (y-x_*)^m, \quad m = 0, 1, \ldots$$

Example:

Let

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

We call expansion

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

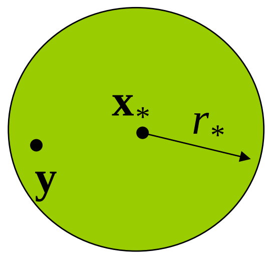
local (regular) inside a sphere

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

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

Basis Functions

Expansion Coefficients



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

Far Field (Singular) Expansions

Let

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

We call expansion

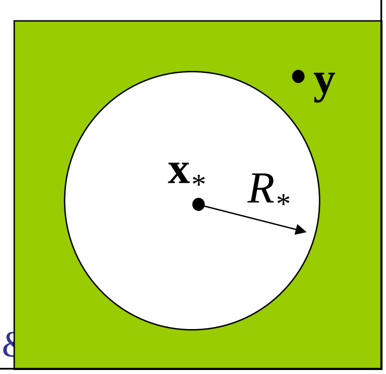
$$\Phi(\mathbf{y}, \mathbf{x}_i) = \sum_{m=0}^{\infty} b_m(\mathbf{x}_i, \mathbf{x}_*) S_m(\mathbf{y} - \mathbf{x}_*)$$

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

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

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

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



Example:

$$\Phi(y, x_i) = \frac{1}{y - x_i}.$$

$$\frac{1}{y - x_i} = \frac{1}{y - x_* - (x_i - x_*)} = \frac{1}{(y - x_*)\left[1 - \frac{x_i - x_*}{y - x_*}\right]} = \frac{1}{(y - x_*)}\left[1 - \frac{x_i - x_*}{y - x_*}\right]^{-1}.$$

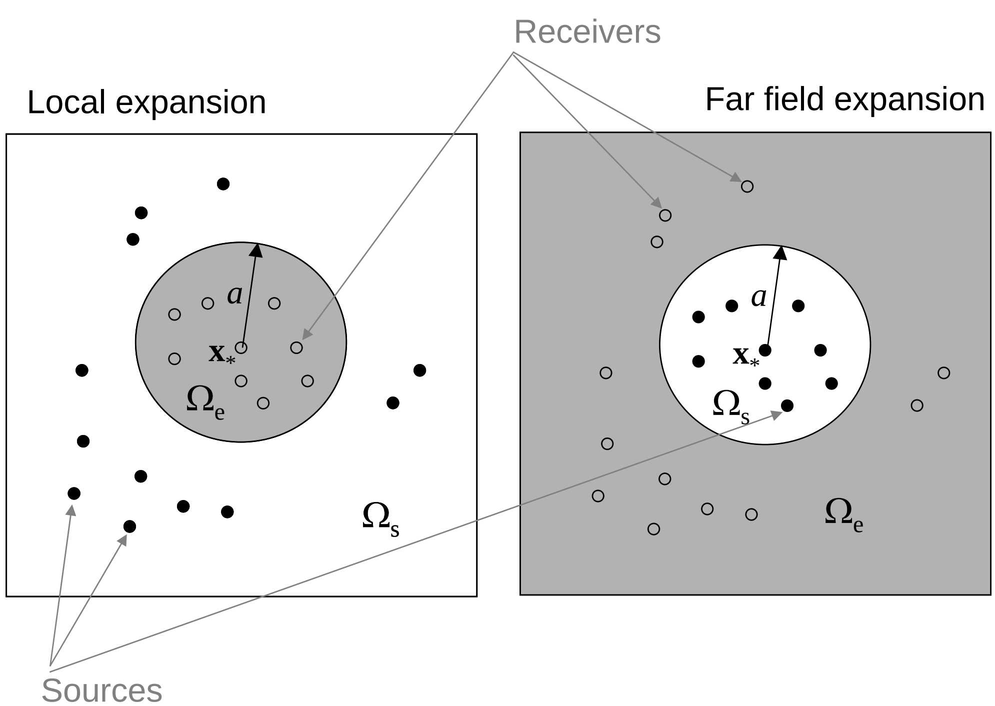
$$\left[1 - \frac{x_i - x_*}{y - x_*}\right]^{-1} = \sum_{m=0}^{\infty} \frac{(x_i - x_*)^m}{(y - x_*)^m}, \quad |y - x_*| > |x_i - x_*|.$$

$$\Phi(y, x_i) = \sum_{m=0}^{\infty} b_m(x_i, x_*) S_m(y - x_*),$$

$$b_m(x_i, x_*) = (x_i - x_*)^m, \quad m = 0, 1, \ldots,$$

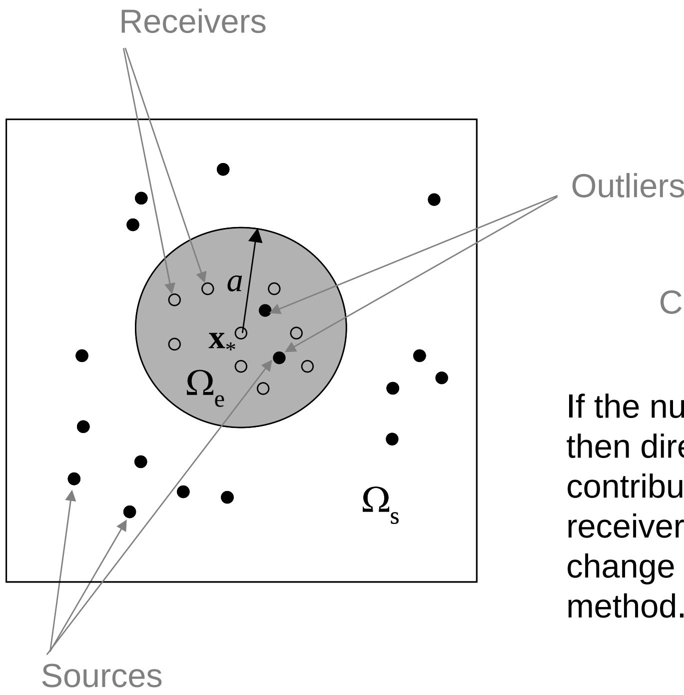
$$S_m(y - x_*) = (y - x_*)^{-m-1}, \quad m = 0, 1, \ldots$$

Middleman for Well Separated Domains:



Complexity: $O(pN+pM)$

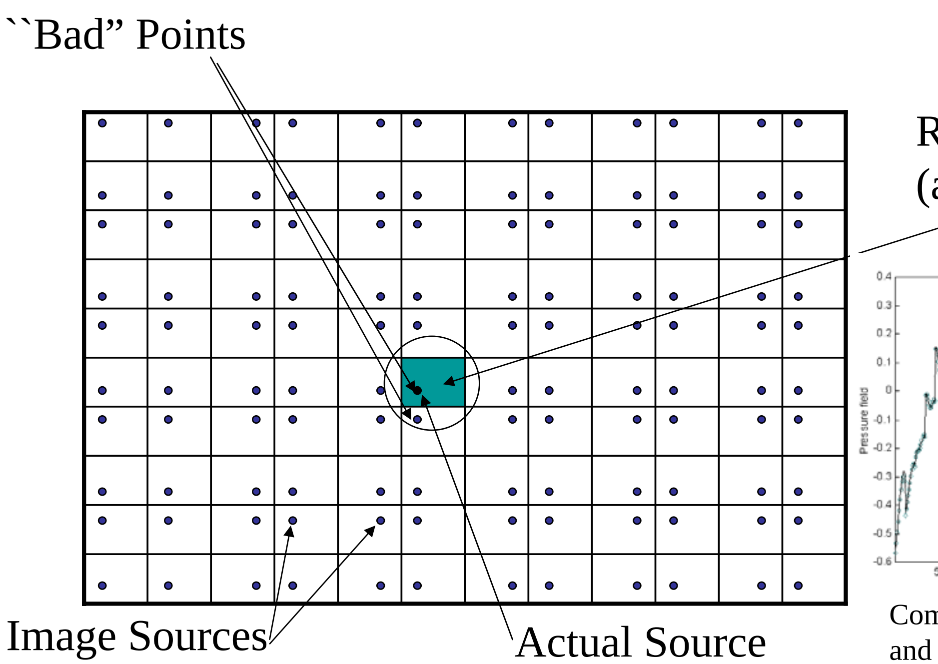
Problem with "Outliers", or "Bad" Points

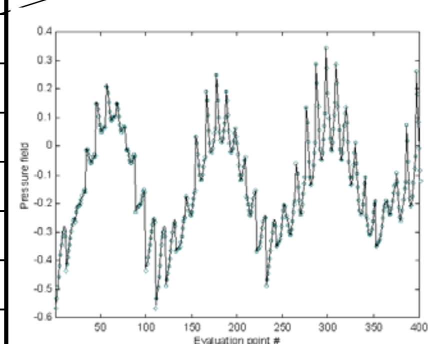


Complexity: $O(pN+pM)$

If the number of outliers is $O(p)$, then direct computation of their contribution to the field at M receivers is $O(pM)$, which does not change the complexity of the method.

Example from Room Acoustics

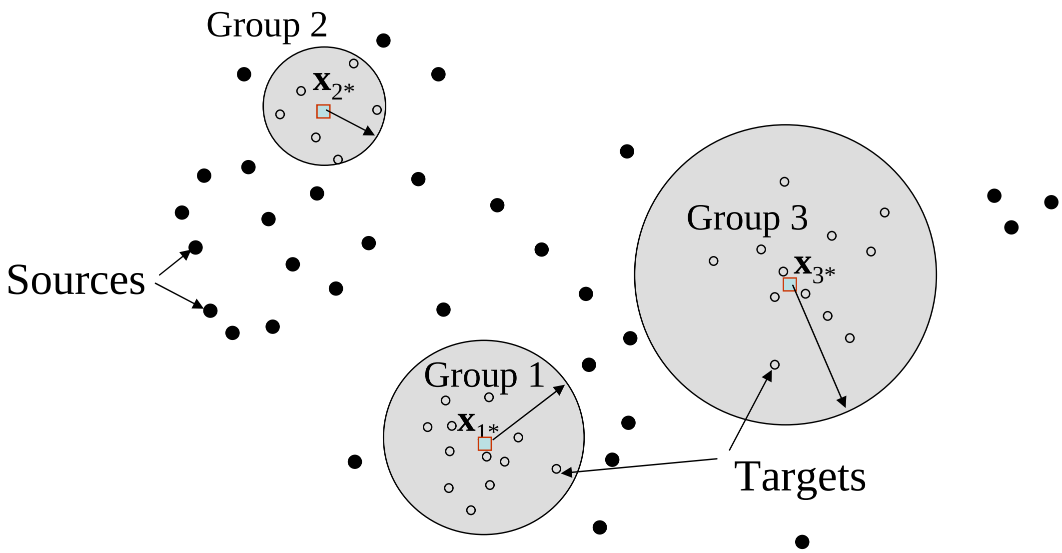




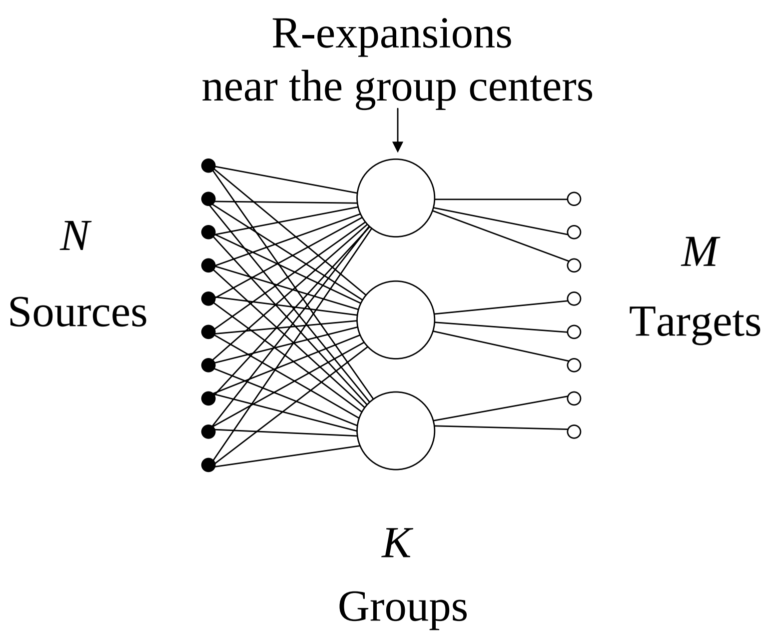
Comparison of Straightforward and Fast Solutions

(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).

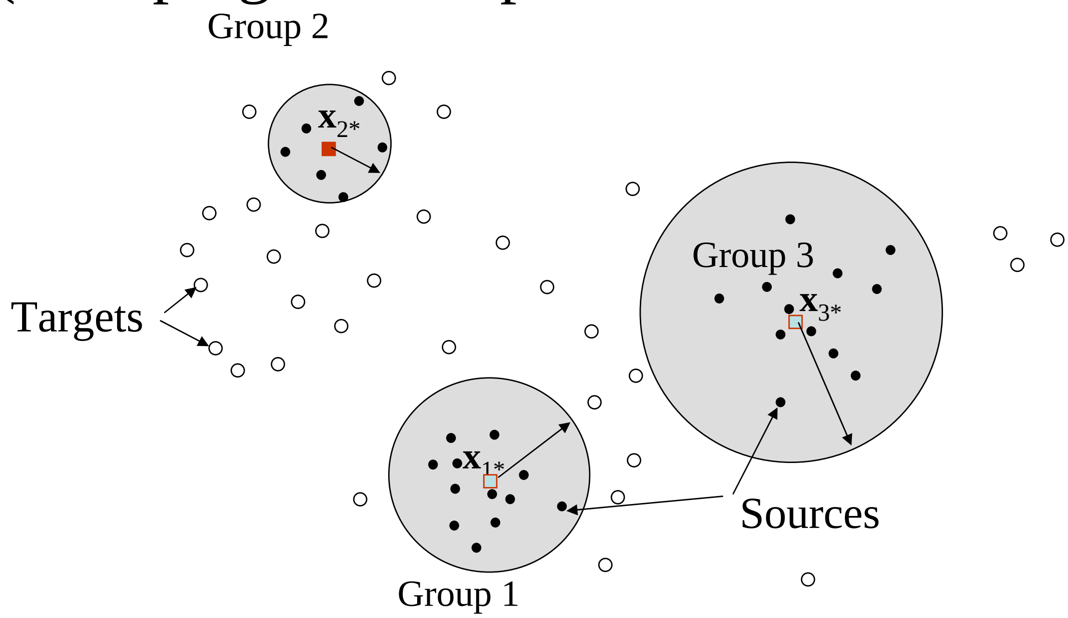
Natural Spatial Grouping for Well Separated Sets (Grouping with Respect to the Target Set)



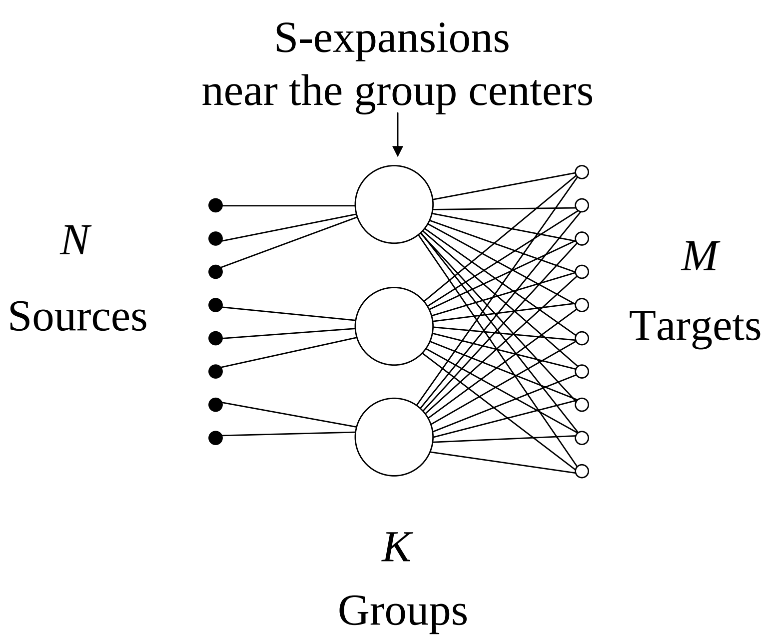
Natural Spatial Grouping for Well Separated Sets (continuation)



Natural Spatial Grouping for Well Separated Sets (Grouping with respect to the Source Set)



Natural Spatial Grouping for Well Separated Sets (continuation)



Examples of Natural Spatial Grouping

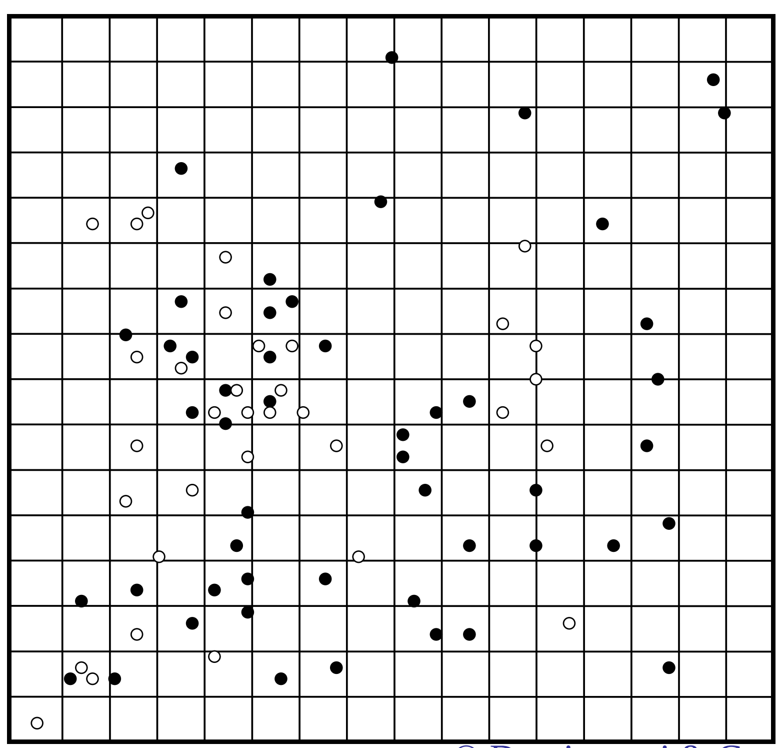
- Stars (Form Galaxies, Gravity);
- Flow Past a Body (Vortices are Grouped in a Wake);
- Statictics (Clusters of Statictical Data Points);
- People (Organized in Groups, Cities, etc.);
- Create your own example !

Space Partitioning
"Modified Middleman"

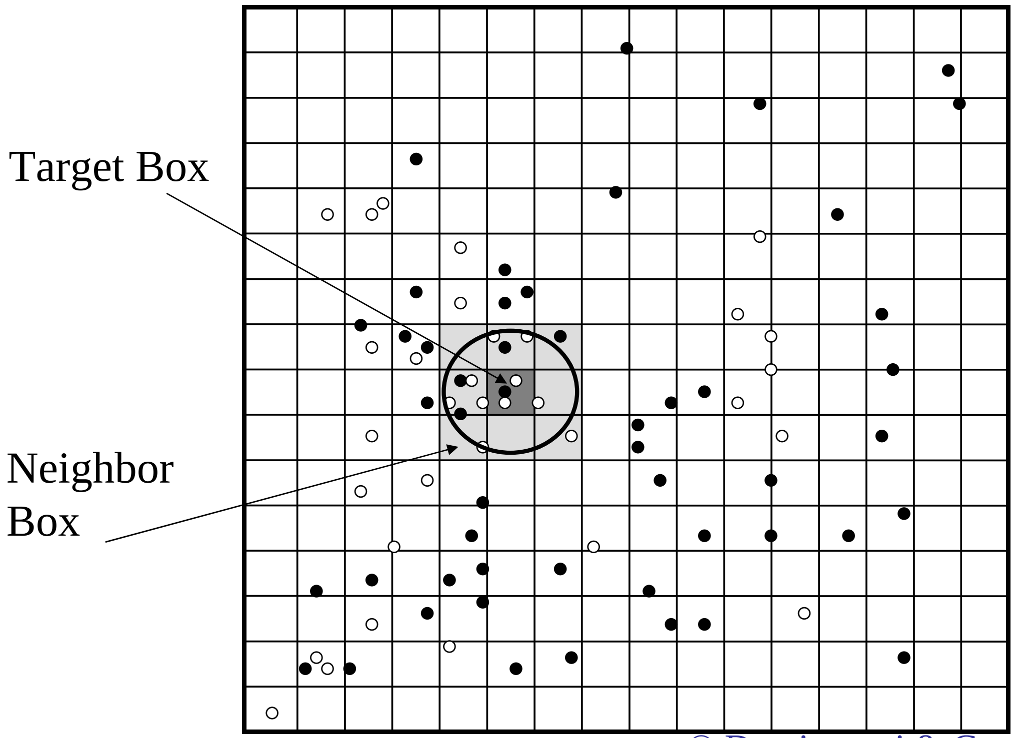
Deficiencies of "Natural Grouping"

- 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



A Modified Middleman Algorithm

- Decomposition of the sum:

Singular Part (sources in the neighborhood)

$$v(\mathbf{y}_j) = \sum_{\mathbf{x}_i \in R_n^+} u_i \Phi(\mathbf{y}_j - \mathbf{x}_i) + \sum_{\mathbf{x}_i \in R_n^-} u_i \Phi(\mathbf{y}_j - \mathbf{x}_i), \quad \mathbf{y}_j \in R_n.$$

Regular Part (sources outside the neighborhood)

- Factorization of the regular part

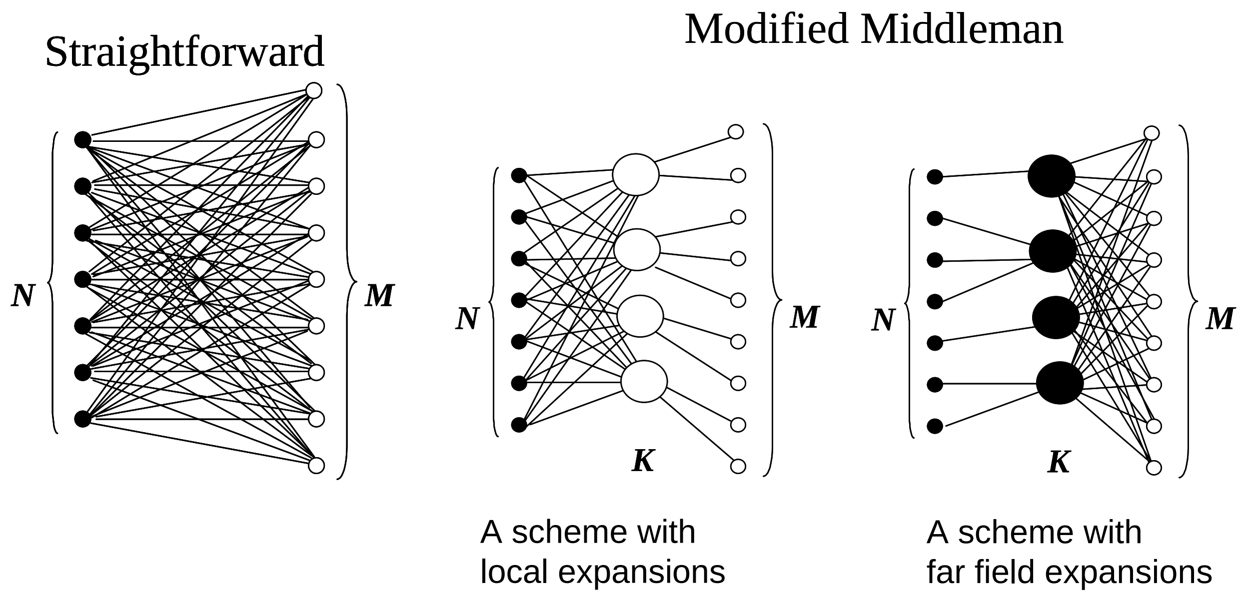
$$\Phi(\mathbf{y}_j - \mathbf{x}_i) = \sum_{m=0}^{p-1} a_m(\mathbf{x}_i, \mathbf{x}_{n*}) R_m(\mathbf{y}_j - \mathbf{x}_{n*}) + Error_p, \quad \mathbf{y}_j, \mathbf{x}_{n*} \in R_n, \quad \mathbf{x}_i \in R_n^-.$$

- Fast computation of the regular part

$$\sum_{\mathbf{x}_i \in R_n^-} u_i \Phi(\mathbf{y}_j - \mathbf{x}_i) = \sum_{m=0}^{p-1} \left[\sum_{\mathbf{x}_i \in R_n^-} u_i a_m(\mathbf{x}_i, \mathbf{x}_{n*}) \right] R_m(\mathbf{y}_j - \mathbf{x}_{n*}).$$

- Direct summation of the singular part, $\sum_{\mathbf{x}_i \in R_n^+} u_i \Phi(\mathbf{y}_j - \mathbf{x}_i)$

A Scheme of "Modified Middleman"



Asymptotic Complexity of the "Modified Middleman Method"

- Let N be the number of sources, M the number of targets, and K the number of target boxes.
- Each target box, R_n, contains M_n targets, $n = 1, \ldots, K$.
- The *neighborhood* of each target box contains N_n sources, $n = 1, \ldots, K$.
- Computation of the expansion coefficients for the regular part for the nth box requires $O((N - N_n)p)$ operations.
- Evaluation of the regular expansion for the nth box requires $O(M_n p)$ operations.
- Direct computation of the singular part requires $O(M_n N_n)$ operations.
- Total complexity is:

$$Complexity = O\left(\sum_{n=1}^{K}[(N - N_n)p + M_n p + M_n N_n]\right).$$

Asymptotic Complexity of the Modified Middleman (continued)

We have

$$\sum_{n=1}^{K} M_n = M$$

Consider a uniform distribution, then

Power of the neighborhood of dimensionality *d* (the number of boxes in the neighborhood)

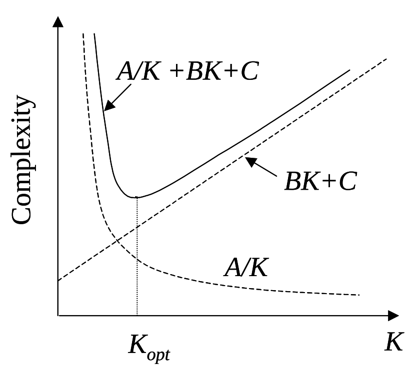
$$N_n \sim const \sim \frac{NPow(d)}{K},$$

$$F(K) = \sum_{n=1}^{K} [(N - N_n)p + M_n p + M_n N_n] = KNp - NpPow(d) + Mp + \frac{MNPow(d)}{K}$$

$$= \frac{MN}{K} Pow(d) + (K - Pow(d))Np + Mp$$

$$Complexity = O(F(K))$$

Optimization of the box number



Optimum complexity

$$F(K) = \frac{MN}{K} Pow(d) + (K - Pow(d))Np + Mp$$

$$K_{opt} = \left[\frac{MNPow(d)}{Np} \right]^{1/2} = \sqrt{\frac{MPow(d)}{p}}.$$

$$Complexity = O(F(K_{opt})) = O\left(Np\left(2\sqrt{\frac{MPow(d)}{p}} - Pow(d) \right) + Mp \right)$$

For $M \sim N,\ p \ll N$:

$$Complexity = O\left(N^{3/2} p^{1/2}\right)$$

Translations
Single Level FMM

Translations (Reexpansions)

Let $\{F_m(\mathbf{y}-\mathbf{x}_{*1})\}$ and $\{G_m(\mathbf{y}-\mathbf{x}_{*2})\}$ be two sets of basis functions centered at $\mathbf{x}_{*1}$ and $\mathbf{x}_{*2}$, such that $\Phi(\mathbf{y}_j, \mathbf{x}_i)$ can be represented by two absolutely and uniformly convergent series in domains Ω_1 and $\Omega_2 \subset \Omega_1$:

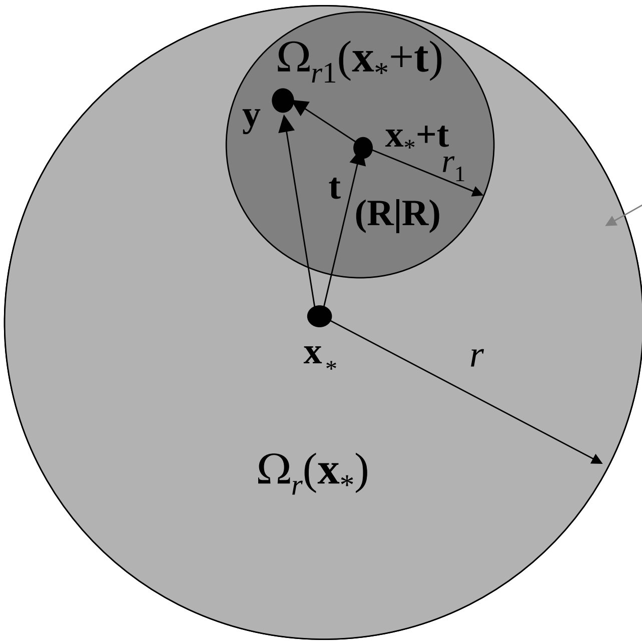
$$\Phi(\mathbf{y}_j, \mathbf{x}_i) = \sum_{n=0}^{\infty} a_n(\mathbf{x}_i - \mathbf{x}_{*1}) F_n(\mathbf{y}_j - \mathbf{x}_{*1}), \quad \mathbf{y}_j \in \Omega_1$$

$$\Phi(\mathbf{y}_j, \mathbf{x}_i) = \sum_{m=0}^{\infty} b_m(\mathbf{x}_i - \mathbf{x}_{*2}) G_m(\mathbf{y}_j - \mathbf{x}_{*2}), \quad \mathbf{y}_j \in \Omega_2 \subset \Omega_1.$$

Under "translation" or "reexpansion" we mean an operator which relates the two sets of expansion coefficients:

$$\{b_m(\mathbf{x}_i - \mathbf{x}_{*2})\} = (F|G)(\mathbf{t})\{a_n(\mathbf{x}_i - \mathbf{x}_{*1})\}, \quad \mathbf{t} = \mathbf{x}_{*2} - \mathbf{x}_{*1}.$$

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

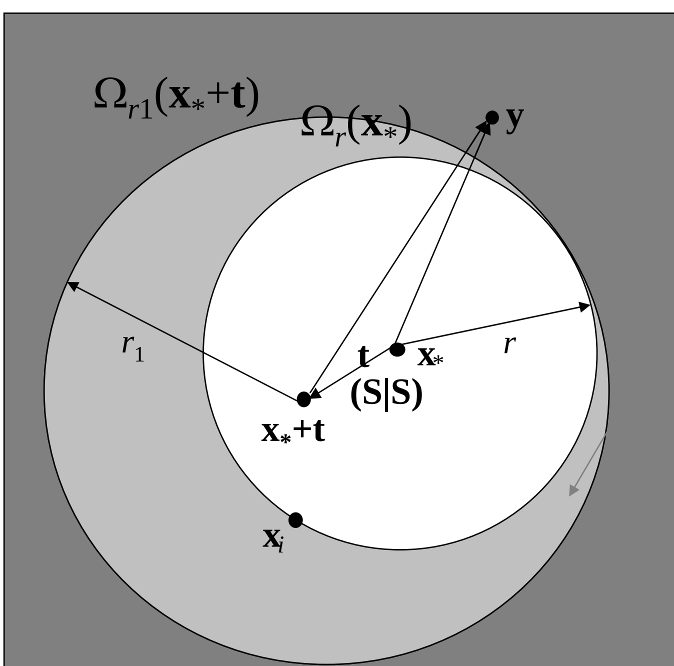


Original expansion
Is valid only here!

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

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

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



Original expansion
Is valid only here!

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

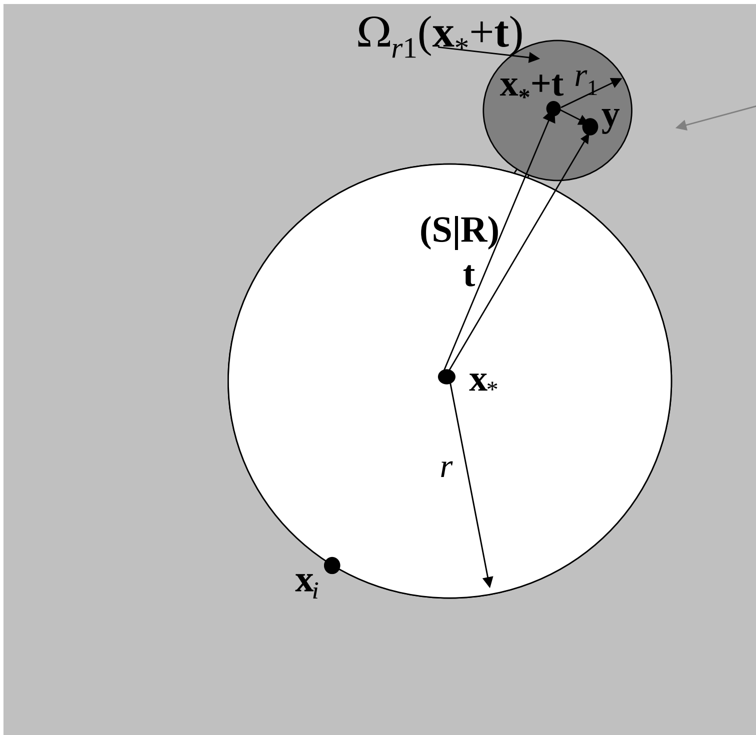
Since
$\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

Also

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

singular point !

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

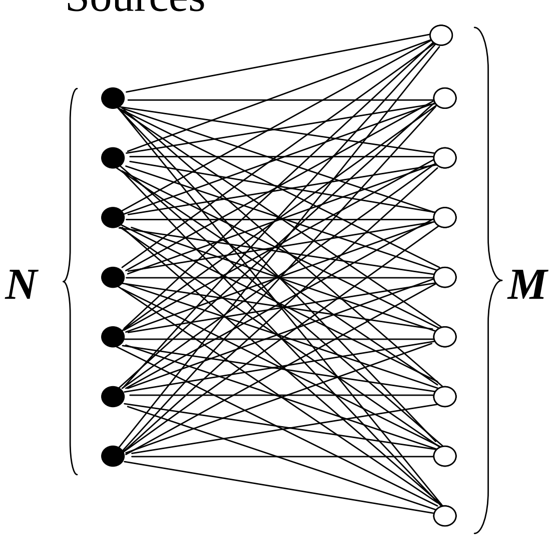
Also

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

singular point !

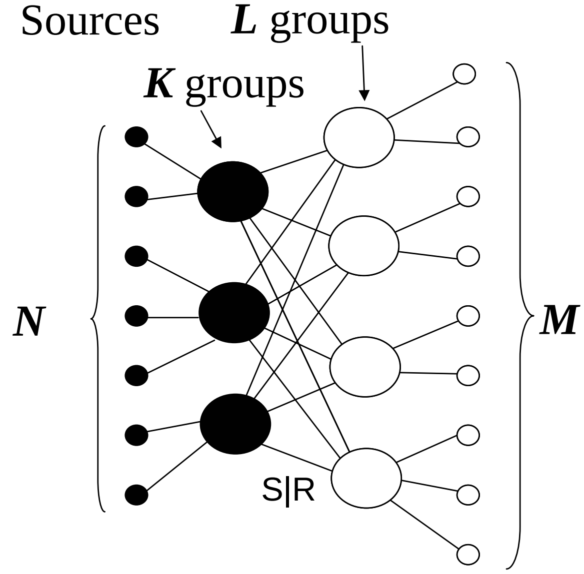
Single Level FMM

Standard algorithm



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

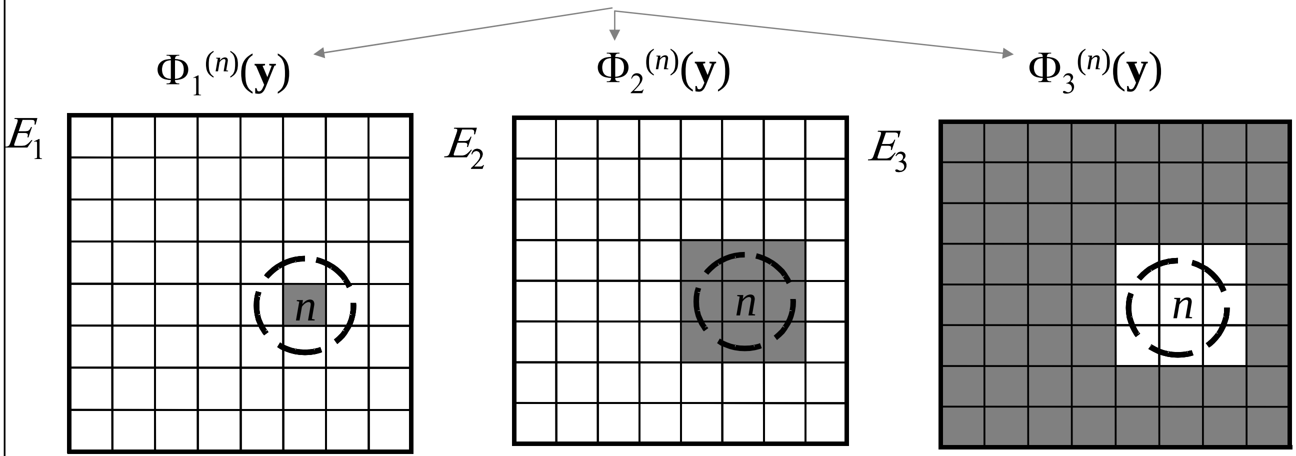
SLFMM



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

Spatial Domains

Potentials due to sources in these spatial domains



$I_1(n) = n$ $\quad$ $I_2(n) = \{Neighbors(n)\} \cup n$

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

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.

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;

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

loop over all sources in the box

For $\mathbf{x}_i \in E_1(n)$

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 $\mathbf{C}^{(n)}$

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;

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

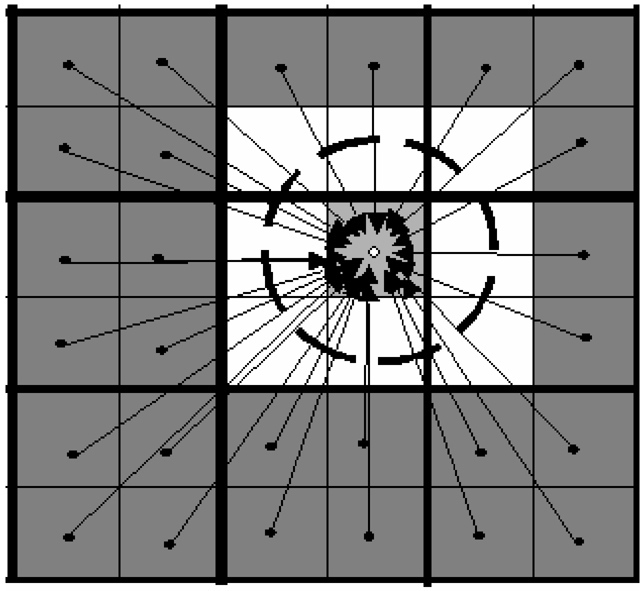
End;

End;

Implementation can be different!

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

S|R-translation



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

Asymptotic Complexity of SLFMM

Assume that:

- By some magic we can easily find neighbors, and lists of points in each box.
- Translation is performed by straightforward *P*x*P* matrix-vector multiplication, where $P(p)$ is the total length of the translation vector. So the complexity of a single translation is $O(P^2)$.
- The source and evaluation points are distributed uniformly, and there are K boxes, with s source points in each box ($s=N/K$). We call s the *grouping* (or *clustering*) parameter.
- The number of neighbors for each box is $O(1)$.

Then Complexity is:

- For Step 1: $O(PN)$
- For Step 2: $O(P^2K^2)$
- For Step 3: $O(PM+Ms)$
- Total: $O(PN+ P^2K^2 +PM+Ms) =$

$$O(PN+ P^2K^2 +PM+MN/K)$$

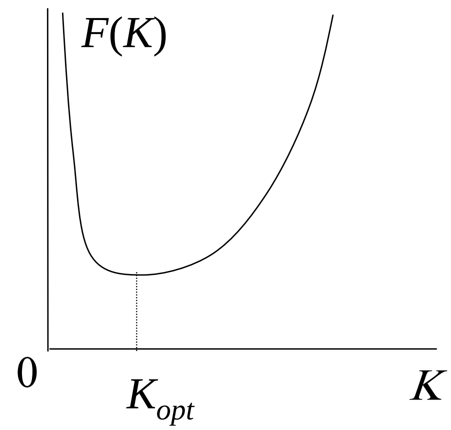
Selection of Optimal K (or s)

$$F(K) = PN + P^2K^2 + PM + PMN/K.$$

$$F'(K) = 2P^2K - PMN/K^2 = 0.$$

$$K_{opt} = \left(\frac{MN}{2P}\right)^{1/3} = O\left(\left(\frac{MN}{P}\right)^{1/3}\right).$$

$$s_{opt} = \frac{N}{K_{opt}} = \left(\frac{2PN^2}{M}\right)^{1/3} = O\left(\frac{PN^2}{M}\right)^{1/3}.$$



Complexity of Optimized SLFMM

$$F(K_{opt}) = PN + P^2\left(\frac{MN}{2P}\right)^{2/3} + PM + PMN\left(\frac{MN}{2P}\right)^{-1/3}$$
$$= P(M+N) + (MN)^{2/3}O(P^{4/3}).$$

At $K = K_{opt}$, and $M = O(N)$, the complexity of SLFMM is:

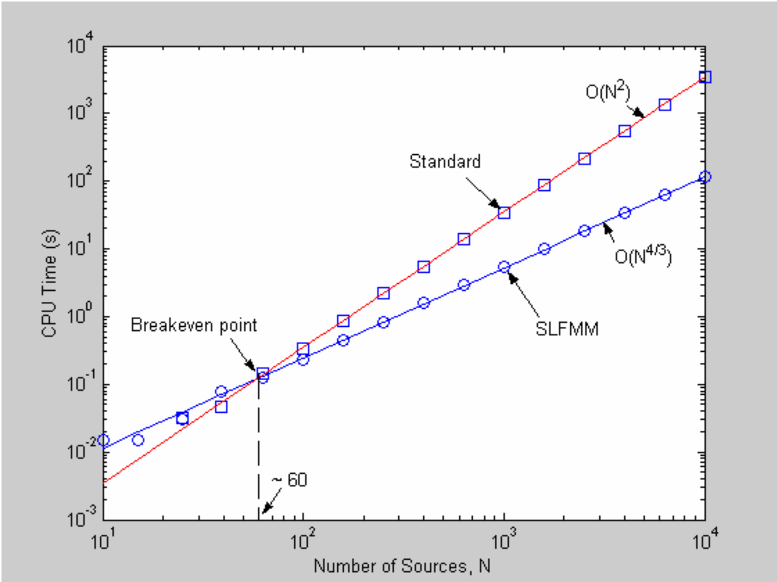
$$O(PN + P^{4/3}N^{4/3}) = O(P^{4/3}N^{4/3}).$$

Example of SLFMM

Compute matrix-vector product

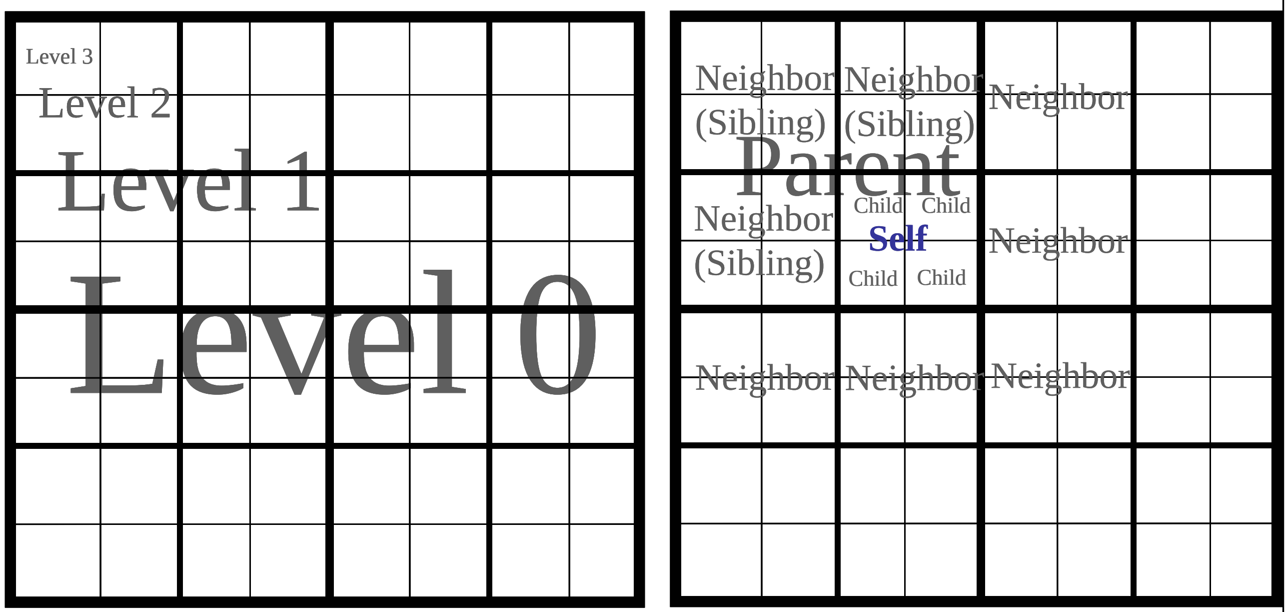
$$v_j = \sum_{i=1}^{N} \Phi_{ji} u_i, \quad j = 1, \ldots, M, \quad \Phi_{ji} = \frac{1}{y_j - x_i},$$

where and $x_1, \ldots, x_N$ are random points uniformly distributed on [0,10], $M = N - 1$, and each y_j is located between the closest x_i's on each side, $j = 1, \ldots, N-1$.

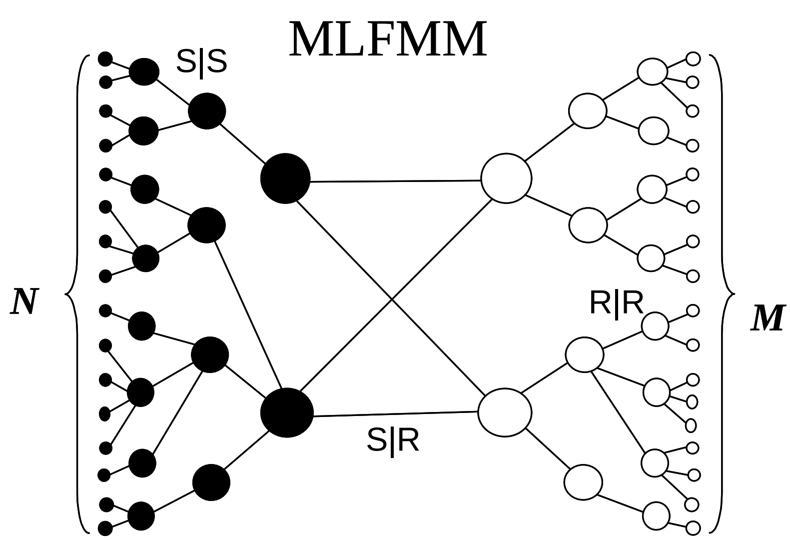


Hierarchical Space Partitioning (Multilevel FMM)

Hierarchy in 2^d-tree



A Scheme of MLFMM



Complexity = $O(pM+pN)$

Example of Multi Level Structure (Post Offices)

Source Hierarchy (Area) ↓

- People (sources, Level 5)
- Mail Box, Post Master (Level 4)
- Local Post Offices (Level 3)
- City Post Office (Level 2)

↓ Mail Transfer

AIRCRAFT

Receiver Hierarchy (Area) ↑

- City Post Office (Level 2)
- Local Post Offices (Level 3)
- Post Master (Level 4)
- People (receivers, Level 5)

↓ Mail Transfer

The MLFMM will be considered in more details in separate lectures