Lecture 25, CMSC 878R/AMSC 698R

Lectures 1 – 3 and Homework 1

- Introduction Applications: Physics, Computer Vision, etc. Simple example of factorization for degenerate kernel
- FMM in General Terms. Review of Publications. Asymptotic Complexity. Examples of complexity
- Simple factorization example. Intro to Matlab.
- Factorization. Local expansions. Far and local expansions. Taylor series. Power series. Taylor series. Error bounds.

A very simple algorithm

- Not FMM, but has some key ideas
- Consider

$$S(x_i)=\sum_{j=1}^{N} \alpha_j (x_i - y_j)^2 \qquad i=1, \ldots ,M$$

- Naïve way to evaluate the sum will require *MN* operations
- Instead can write the sum as

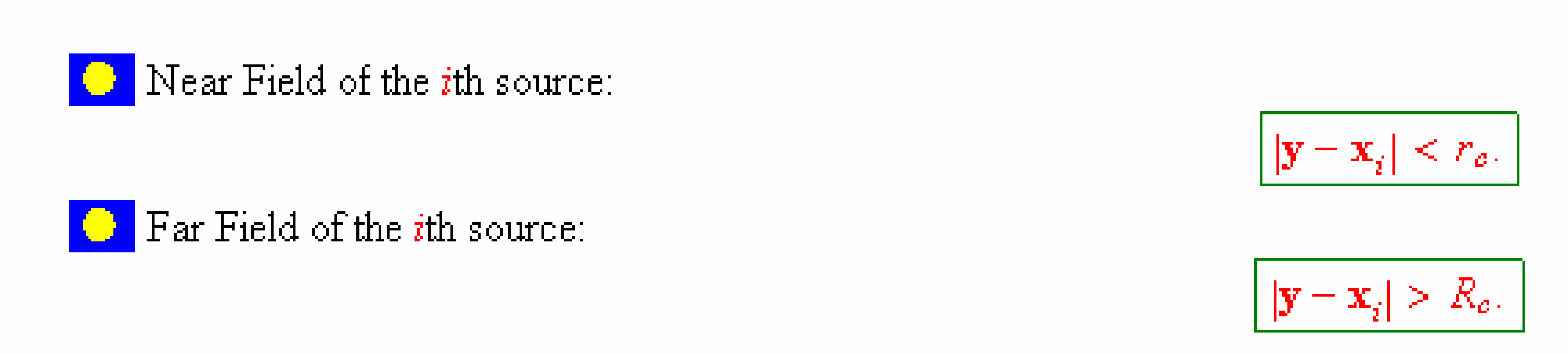
$$S(x_i)=(\sum_{j=1}^{N} \alpha_j)x_i^2 + (\sum_{j=1}^{N} \alpha_j y_j^2) -2x_i(\sum_{j=1}^{N} \alpha_j y_j)$$

  - Can evaluate each bracketed sum over *j* and evaluate an expression of the type

$$S(x_i)=\beta x_i^2 + \gamma -2x_i\delta$$

  - Requires O(M+N) operations
- Key idea – use of analytical manipulation of series to achieve faster summation

Far Field and Near Field



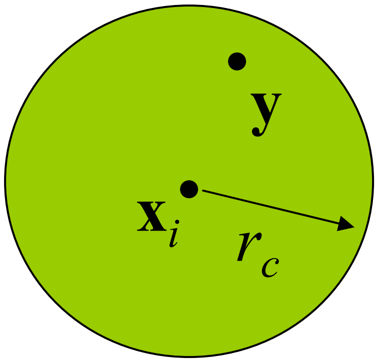
- Near Field of the *i*th source:

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

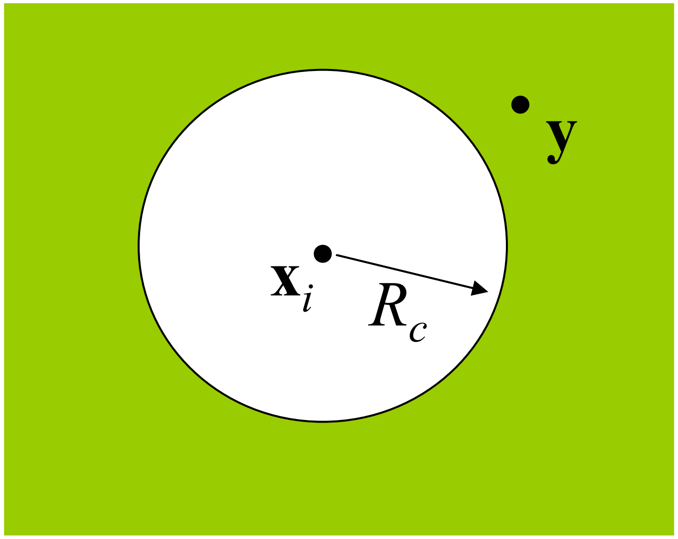
- Far Field of the *i*th source:

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

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

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

Basis Functions

We call expansion

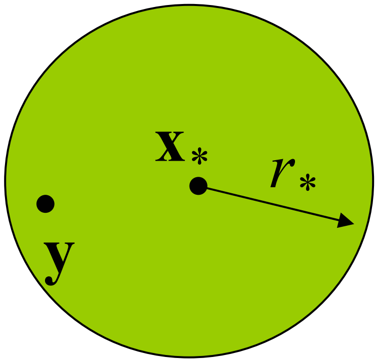
$$\Phi(\mathbf{y},\mathbf{x}_i) = \sum_{m=0}^{\infty} a_m(\mathbf{x}_i,\mathbf{x}_*) R_m(\mathbf{y}-\mathbf{x}_*)$$

local (regular) inside a sphere

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

Expansion Coefficients

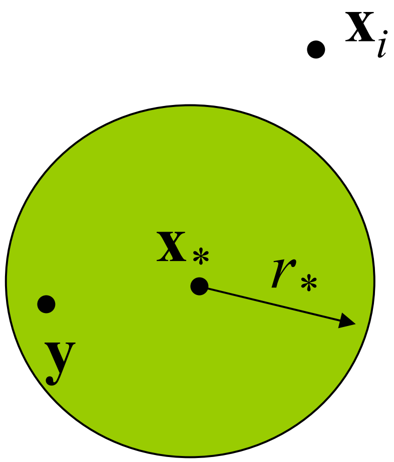
if the series converges for $\forall \mathbf{y},\ |\mathbf{y}-\mathbf{x}_*| < r_*.$



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

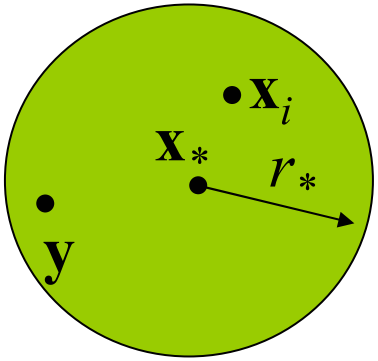
Local Expansion of a Regular Potential

Can be like this:



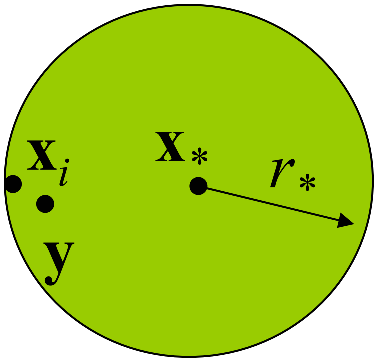
$|\mathbf{y} - \mathbf{x}_*| < r_* < |\mathbf{x}_i - \mathbf{x}_*|$

…or like this:



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

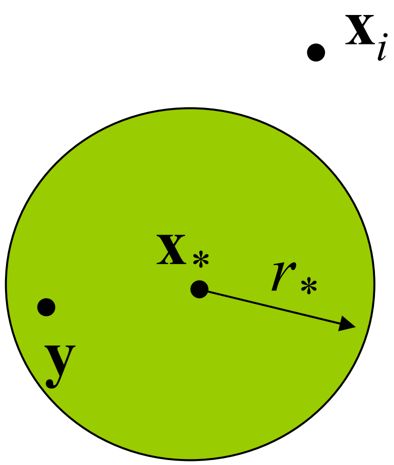
…or like this:



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

Local Expansion of a Singular Potential

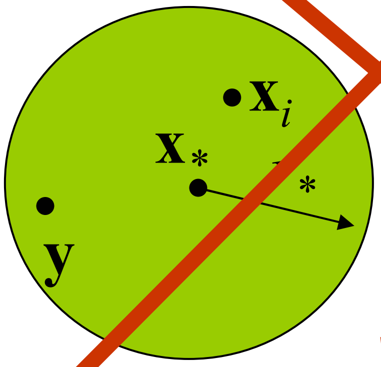
Can be like this:



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

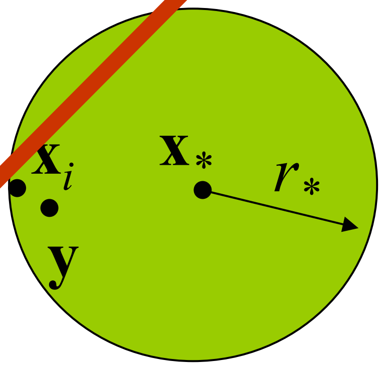
Like this only!

…or like this:



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

…or like this:



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

Never ever!

Because $\mathbf{x}_i$ is a singular point!

Lectures 4-5, Homework 2

- Multidimensional Taylor series.
- Kronecker product. Dot product.
- General form of factorization.
- Properties of Kronecker and dot product.
- Middleman factorization for sums of Gaussians using Taylor series.
- Compression of multidimensional series. Compression operator. Complexity in *d*-dimensions. Use of compression in multidimensional FMM.

Use Compression!

Compression operator:

$$\mathbf{A}^n = \mathbf{Compress}(\mathbf{a}^n)$$

Required Property:

$$\mathbf{a}^n \bullet \mathbf{b}^n = \mathbf{Compress}(\mathbf{a}^n) \bullet \mathbf{Compress}(\mathbf{b}^n).$$

Consider $\mathbf{R}^2$:

$$\mathbf{a}^n \bullet \mathbf{b}^n = (\mathbf{a} \bullet \mathbf{b})^n = (a_1 b_1 + a_2 b_2)^n$$

$$= a_1^n b_1^n + \binom{n}{1} a_1^{n-1} b_1^{n-1} a_2 b_2 + \binom{n}{2} a_1^{n-2} b_1^{n-2} a_2^2 b_2^2 + \ldots + a_2^n b_2^n$$

The length is only (n +1), not 2^n

Let us define:

$$\mathbf{A}^n = \mathbf{Compress}(\mathbf{a}^n) = \left(a_1^n, \sqrt{\binom{n}{1}} a_1^{n-1} a_2, \sqrt{\binom{n}{2}} a_1^{n-2} a_2^2, \ldots, a_2^n \right),$$

$$\mathbf{B}^n = \mathbf{Compress}(\mathbf{b}^n) = \left(b_1^n, \sqrt{\binom{n}{1}} b_1^{n-1} b_2, \sqrt{\binom{n}{2}} b_1^{n-2} b_2^2, \ldots, b_2^n \right)$$

Example of Fast Computation

$$v_j = \sum_{i=1}^{N} u_i \Phi\left(\mathbf{y}_j, \mathbf{x}_i\right) = \sum_{m=0}^{p-1} \mathbf{c}_m \bullet \left(\mathbf{y}_j - \mathbf{x}_*\right)^m + \text{Residual}, \qquad \mathbf{c}_m = \frac{1}{m!} \sum_{i=1}^{N} u_i e^{\mathbf{x}_* \bullet \mathbf{x}_i} \mathbf{x}_i^m.$$

Equivalent to:

$$v_j = \sum_{m=0}^{p-1} \mathbf{C}_m \bullet \mathbf{Compress}\left(\left(\mathbf{y}_j - \mathbf{x}_*\right)^m\right) + \text{Residual}, \qquad \mathbf{C}_m = \frac{1}{m!} \sum_{i=1}^{N} u_i e^{\mathbf{x}_* \bullet \mathbf{x}_i} \mathbf{Compress}(\mathbf{x}_i^m).$$

Number of multiplications (complexity) to obtain v_j:

$$Complexity = 1 + 2 + \ldots + p = \frac{p(p+1)}{2}.$$

Compression Can be Performed for any Dimensionality (Example for 3D):

$$\mathbf{a}^n \bullet \mathbf{b}^n = (\mathbf{a} \bullet \mathbf{b})^n = (a_1 b_1 + a_2 b_2 + a_3 b_3)^n$$

$$= [(a_1 b_1 + a_2 b_2) + a_3 b_3]^n = \sum_{m=0}^{n} \binom{n}{m} (a_1 b_1 + a_2 b_2)^{n-m} a_3^m b_3^m$$

$$= \sum_{m=0}^{n} \sum_{l=0}^{n-m} \binom{n}{m} \binom{n-m}{l} a_1^{n-m-l} b_1^{n-m-l} a_2^l b_2^l a_3^m b_3^m$$

$$= a_1^n b_1^n + \binom{n}{1} a_1^{n-1} b_1^{n-1} a_2 b_2 + \binom{n}{2} a_1^{n-2} b_1^{n-2} a_2^2 b_2^2 + \ldots + a_2^n b_2^n$$

$$+ \binom{n}{1} a_1^{n-1} b_1^{n-1} a_3 b_3 + \binom{n}{1} \binom{n-1}{1} a_1^{n-2} b_1^{n-2} a_2 b_2 a_3 b_3 + \ldots + a_3^n b_3^n,$$

$$\mathbf{Compress}(\mathbf{a}^n) = \left(a_1^n, \sqrt{\binom{n}{1}} a_1^{n-1} a_2, \sqrt{\binom{n}{2}} a_1^{n-2} a_2^2, \ldots, a_2^n, \sqrt{\binom{n}{1}} a_1^{n-1} a_3, \ldots, a_3^n \right)$$

The length of $\mathbf{a}^n$ is $(n+1)+n+\ldots+1 = (n+1)(n+2)/2$

Lectures 6 -8, Homework 3

- Far field expansions. Functional analysis review. Basis functions. Far field expansions. Examples of far field expansions (power, asymptotic)
- Multidimensional middleman for FGT and use of the "Compression" operator.
- Operators in space of coefficients. Truncation operators. Translation operators. Operator norms. Errors.
- $S|S$, $R|R$, $S|R$ operators
- Single level FMM
- Requirements for functions in FMM.
- Formalization of FMM.

Four Key Stones of FMM

- Factorization
- Error
- Translation
- Grouping

Summary of formal requirements for functions that can be used in FMM

- We have two sets of points:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathrm{R}^d, \quad i = 1, ..., N,$$
$$Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}, \quad \mathbf{y}_j \in \mathrm{R}^d, \quad j = 1, ..., M.$$

- We have functions (potentials):

$$\Phi(\mathbf{x}_i, \mathbf{y}) : \mathrm{R}^d \to \mathrm{R}, \quad \mathbf{y} \in \mathrm{R}^d, \quad i = 1, ..., N.$$

- These functions can be factorized as (local expansion):

$$\Phi(\mathbf{x}_i, \mathbf{y}) = \mathbf{A}(\mathbf{x}_i, \mathbf{x}_*) \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_*), \quad |\mathbf{y} - \mathbf{x}_*| < r < |\mathbf{x}_i - \mathbf{x}_*|, \quad i = 1, ..., N$$

- These functions can be factorized as (far field expansion):

$$\Phi(\mathbf{x}_i, \mathbf{y}) = \mathbf{B}(\mathbf{x}_i, \mathbf{x}_*) \circ \mathbf{S}(\mathbf{x} - \mathbf{x}_*), \quad |\mathbf{y} - \mathbf{x}_*| > R > |\mathbf{x}_i - \mathbf{x}_*|, \quad i = 1, ..., N$$

- The product is distributive operation with respect to addition

$$(u_1\mathbf{A}_1 + u_2\mathbf{A}_2) \circ \mathbf{F} = u_1\mathbf{A}_1 \circ \mathbf{F} + u_2\mathbf{A}_2 \circ \mathbf{F}, \quad \mathbf{F} = \mathbf{S}, \mathbf{R}$$

Summary of formal requirements for functions that can be used in FMM (2)

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

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

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

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

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

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

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

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

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

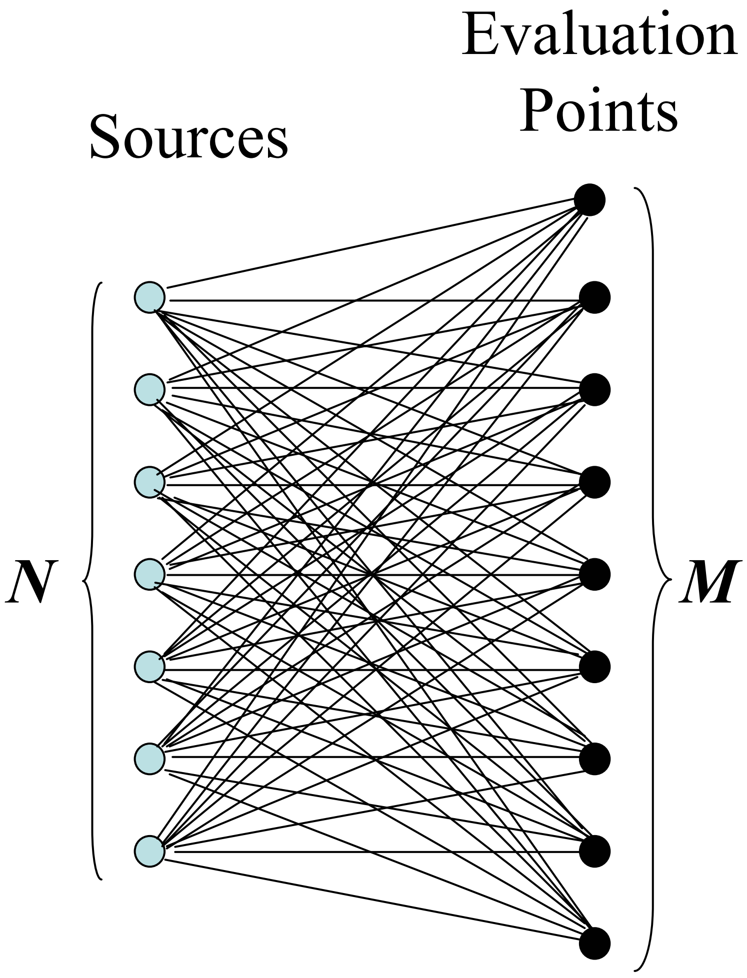
- And we are looking for sums:

$$v_j = \sum_{i=1}^{N} u_i \Phi(\mathbf{y}_j,\mathbf{x}_i), \quad j = 1,\ldots,M.$$

- Some generalization are possible, say instead of $\Phi(\mathbf{y}_j,\mathbf{x}_i)$ we can consider $\Phi_i(\mathbf{y}_j)$, etc.

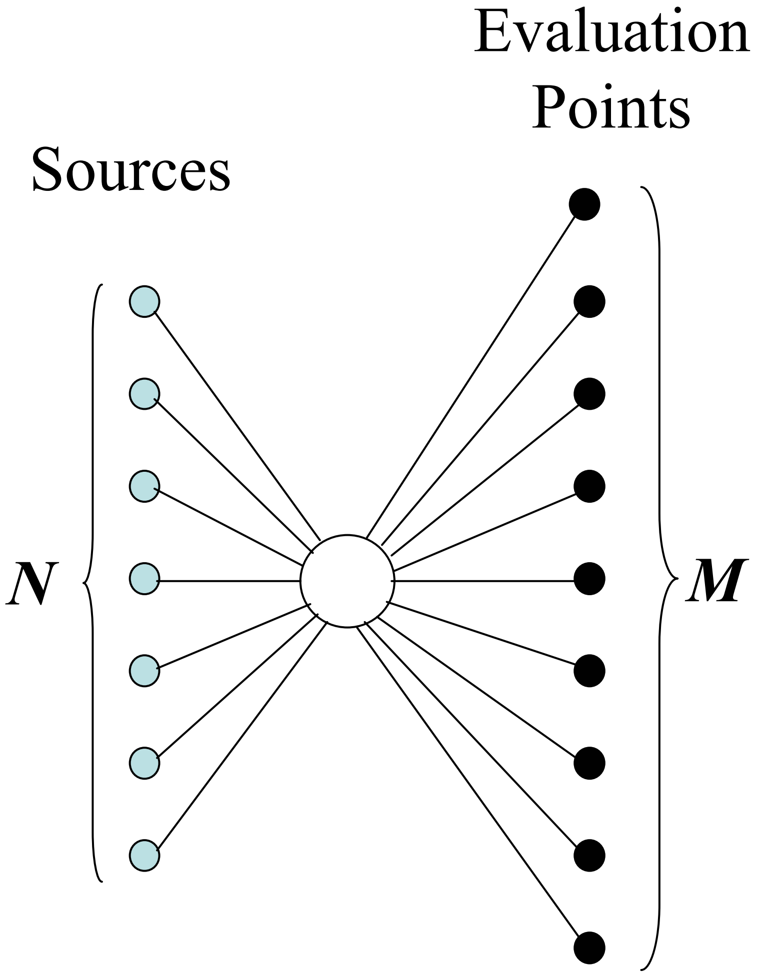
Middleman Algorithm

Standard algorithm



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

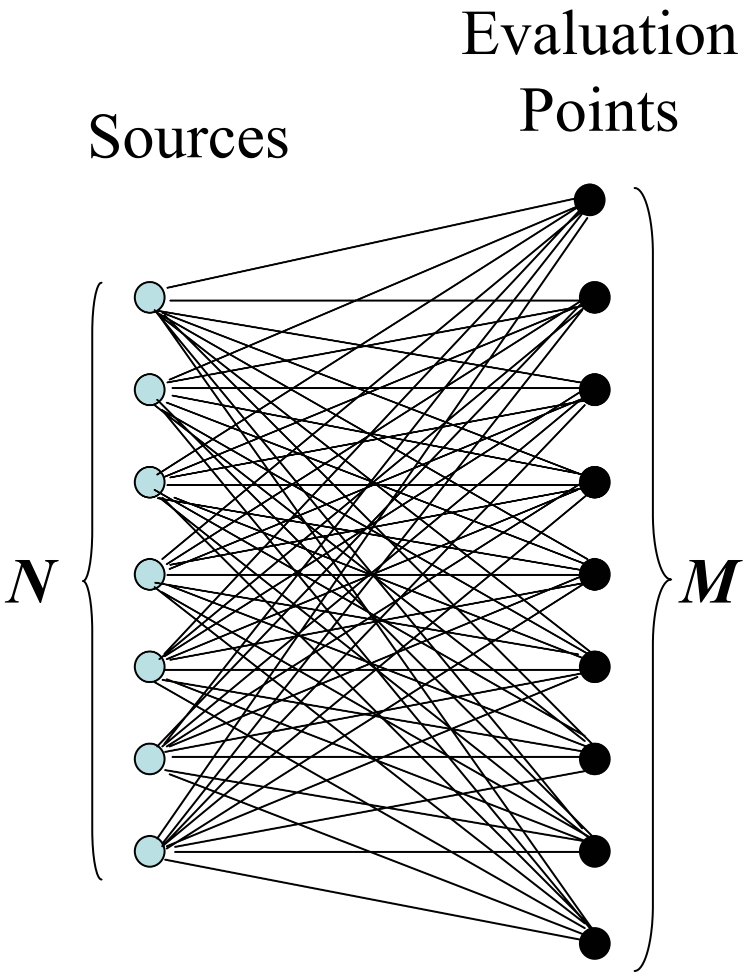
Middleman algorithm



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

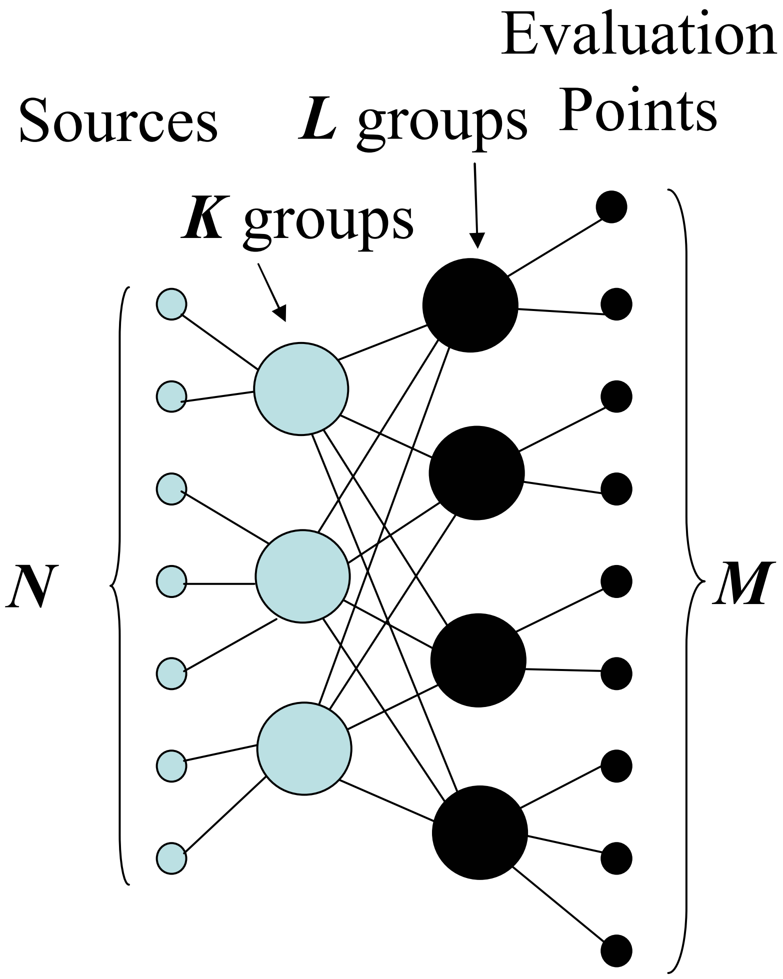
Idea of a Single Level FMM

Standard algorithm



Total number of operations: $O(NM)$

SLFMM

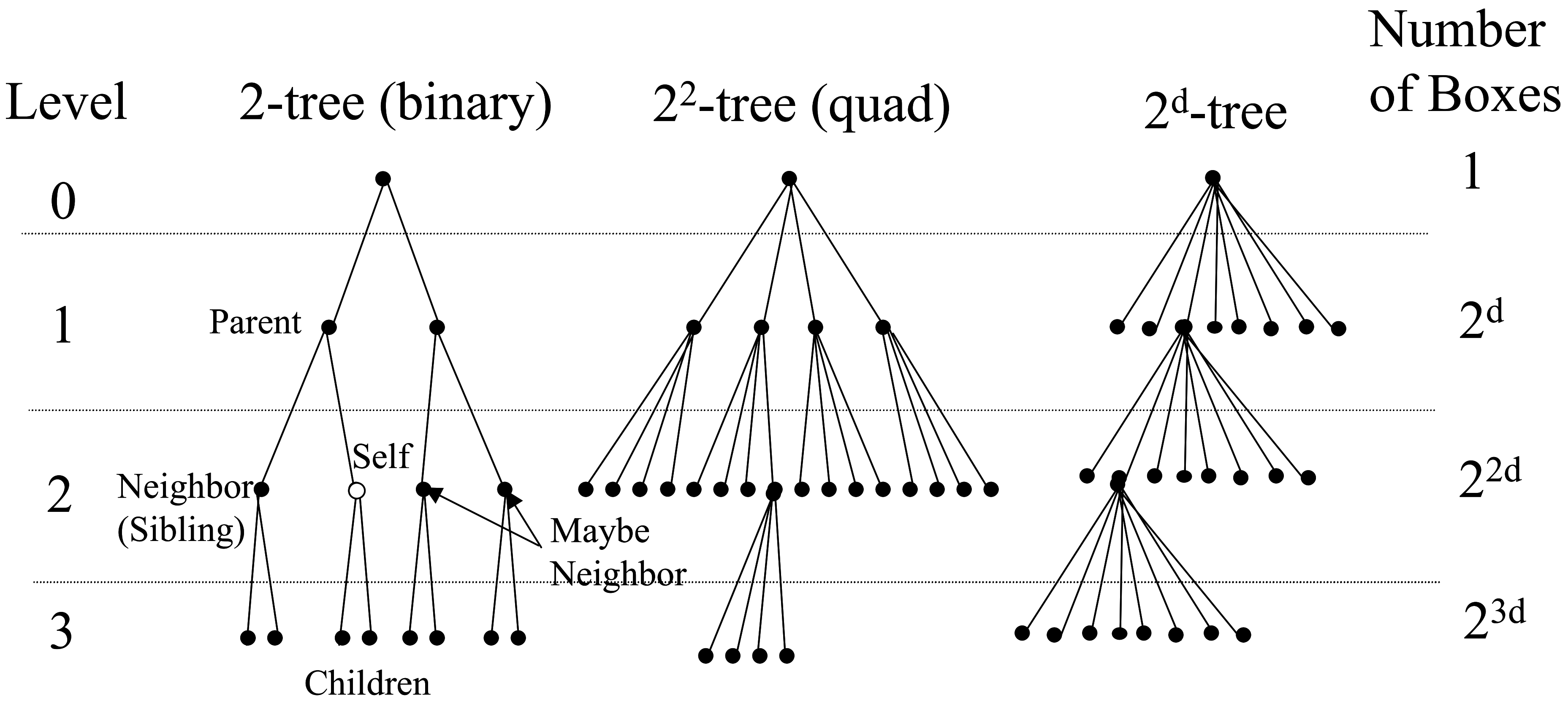


Total number of operations: $O(N+M+KL)$

Lecture 8-10, Homeworks 4,5

- SLFMM and optimization.
- S|R translation operator for $(x-y)^{-1}$.
- Error bounds. Data Structures. 2^d-trees.
- Need for data structures. 2^d trees.
- Parents, children, etc. Hierarchical space subdivision.
- Data Structures. Bit interleaving and spatial ordering. Neighbor search.
- Threshold level of space subdivision.
- Single Level FMM for $(x-y)^{-1}$. Error bounds

2^d-trees


Level
2-tree (binary)
2^2-tree (quad)
2^d-tree
Number of Boxes
0
1
1
2^d
2
2^{2d}
3
2^{3d}
Parent
Self
Neighbor (Sibling)
Maybe Neighbor
Children

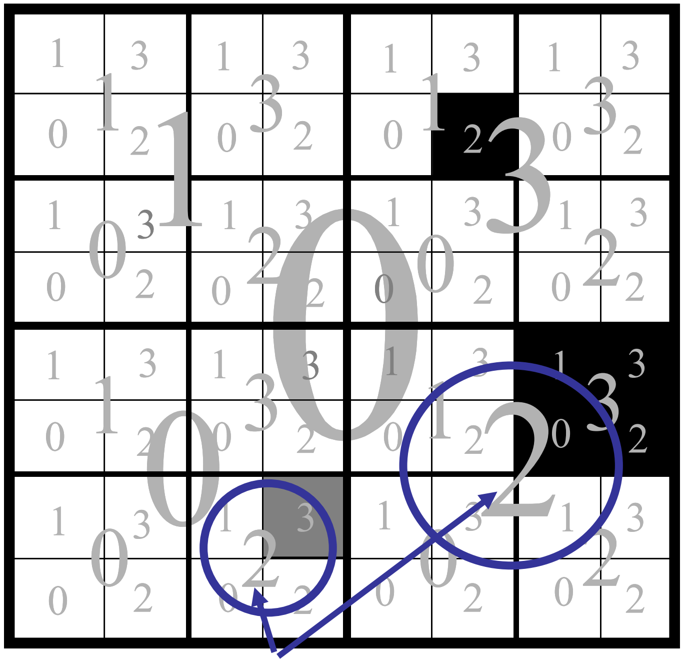
Parent Number

Parent numbering string:

$$Parent(N_1, N_2, \ldots, N_{l-1}, N_l) = (N_1, N_2, \ldots, N_{l-1}).$$

Parent number:

$$Parent(Number) = \left(2^d\right)^{l-2} \bullet N_1 + \left(2^d\right)^{l-3} \bullet N_2 + \ldots + N_{l-1}.$$



Parent number does not depend on the level of the box! E.g. in the quad-tree at any level

$$Parent(11_{10}) = Parent(23_4) = 2_4 = 2_{10}.$$

Parent's universal number:

$$Parent((Number, l)) = (Parent(Number), l-1).$$

Algorithm to find the parent number:

$$Parent(Number) = \left[Number/2^d\right]$$

For box #23_4 (gray or black) the parent box number is 2_4.

Children Numbers

Children numbering strings:

$$Children(N_1,N_2,...,N_{l-1},N_l) = \{(N_1,N_2,...,N_{l-1},N_l,N_{l+1})\}, \quad N_{l+1} = 0,...,2^d - 1.$$

Children numbers:

$$Children(Number) = \left\{ (2^d)^l \bullet N_1 + (2^d)^{l-1} \bullet N_2 + ... + (2^d) \bullet N_l + N_{l+1} \right\}, \quad N_{l+1} = 0,...,2^d - 1.$$

Children numbers do not depend on the level of the box! E.g. in the quad-tree at any level:

$$Children(11_{10}) = Children(23_4) = \{230_4, 231_4, 232_4, 233_4\} = \{44_{10}, 45_{10}, 46_{10}, 47_{10}\}$$

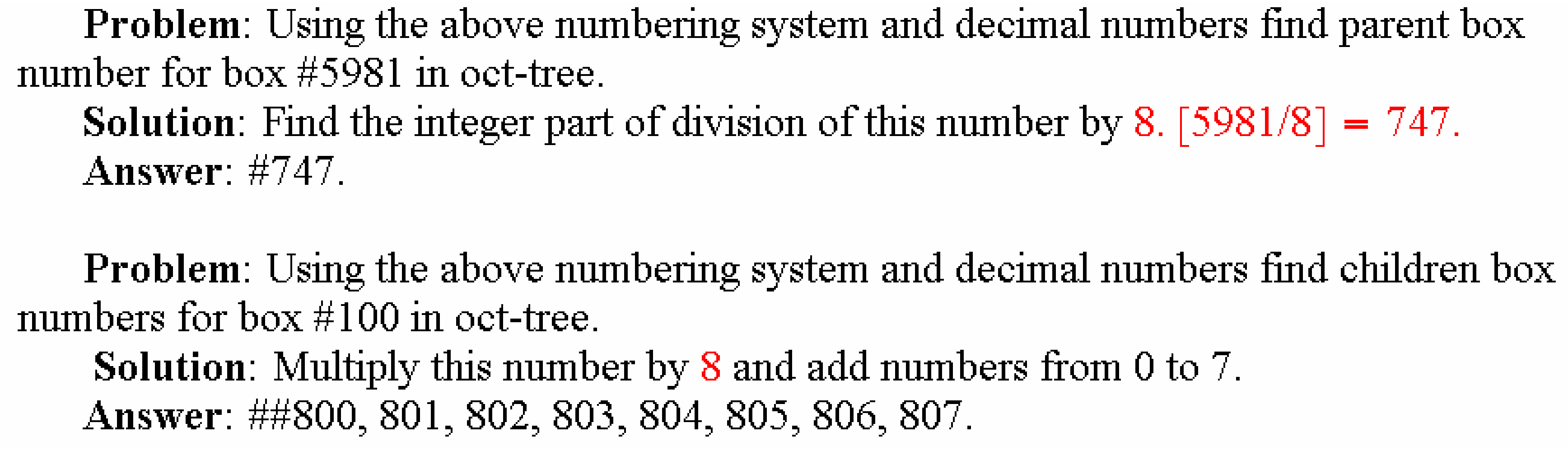
Children universal numbers:

$$Children((Number, l)) = (Children(Number), l+1).$$

Algorithm to find the children numbers:

$$Children(Number) = \left\{ 2^d \bullet Number + j \right\}, \quad j = 0,...,2^d - 1,$$

A couple of examples:



Problem: Using the above numbering system and decimal numbers find parent box number for box #5981 in oct-tree.

Solution: Find the integer part of division of this number by 8. [5981/8] = 747.

Answer: #747.

Problem: Using the above numbering system and decimal numbers find children box numbers for box #100 in oct-tree.

Solution: Multiply this number by 8 and add numbers from 0 to 7.

Answer: ##800, 801, 802, 803, 804, 805, 806, 807.

Lectures 11-13, Exam 1, Homework 6

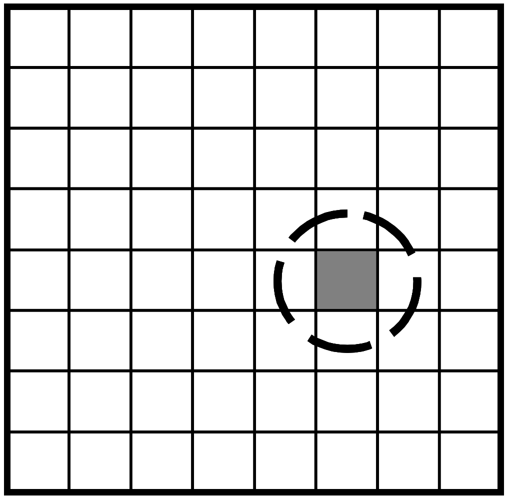
- Multilevel FMM. Structure of the algorithm.
- Setting data structure.
- Upward Pass. Hierarchical domains and potentials. Multilevel FMM. Downward Pass.
- Asymptotic Complexity of the MLFMM. Downward Pass. Complexity of each step.
- Bookkeeping routines necessary for MLFMM based on bit-interleaving
- Multilevel FMM. Optimization. Results of MLFMM tests. Dependence of FMM performance on parameters.
- Review of concepts.

Prepare Data Structures

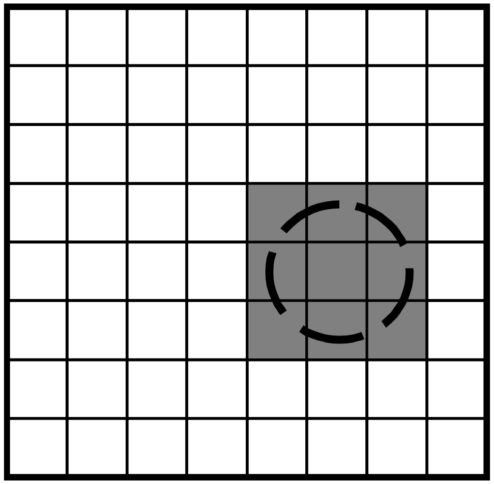
- Convert data set into integers given some maximum number of bits allowed/dimensionality of space
- Interleave
- Sort
- Go through the list and check at what bit position two strings differ
  - For a given *s* determine the number of levels of subdivision needed

Hierarchical Spatial Domains

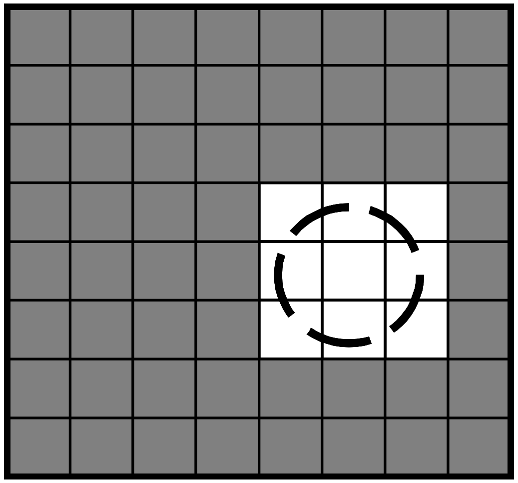
E_1



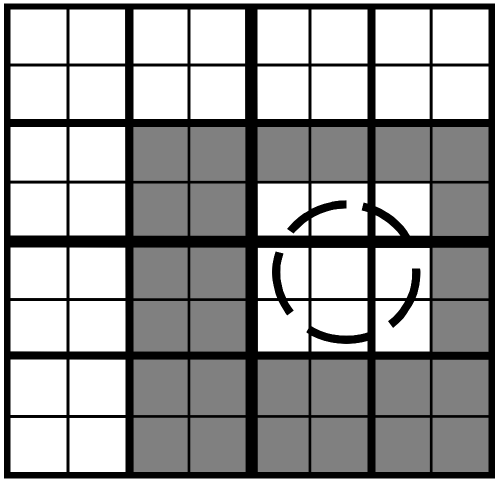
E_2



E_3



E_4



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 $\Phi(x_i,y)=\sum^P u_i B(x_*,x_i)\ S(x_*,y)$

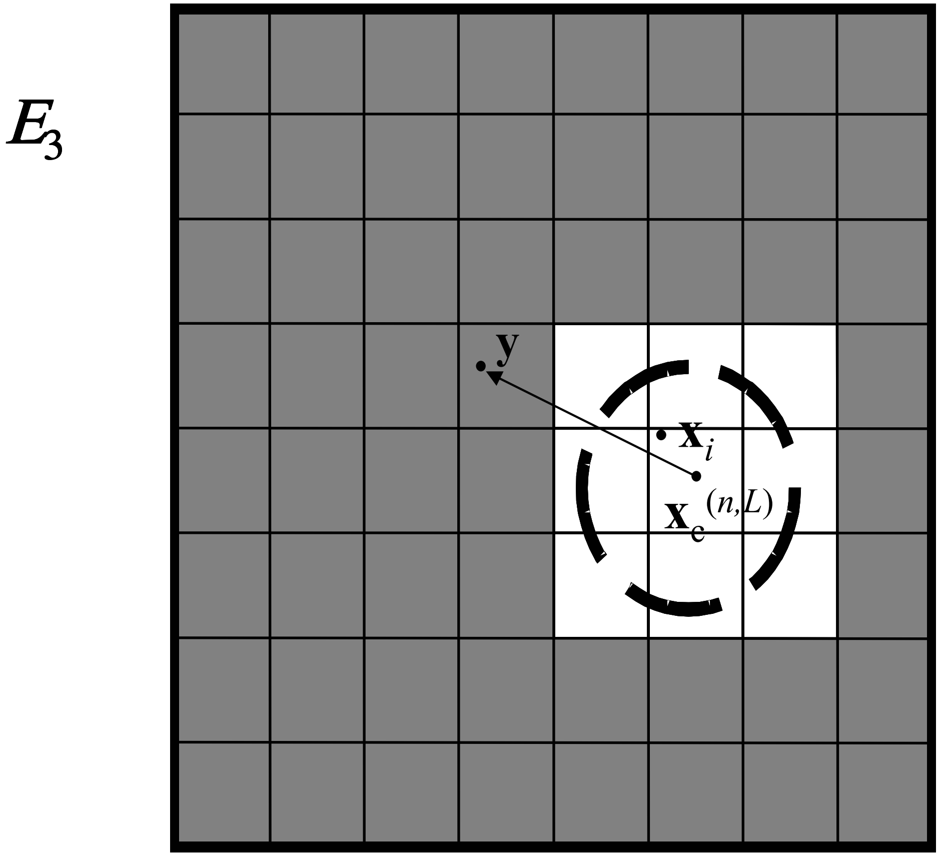
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}\left(\mathbf{y}-\mathbf{x}_c^{(n,L)}\right),$$
$$\mathbf{C}^{(n,L)} = \sum_{\mathbf{x}_i \in E_1(n,L)} u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right).$$

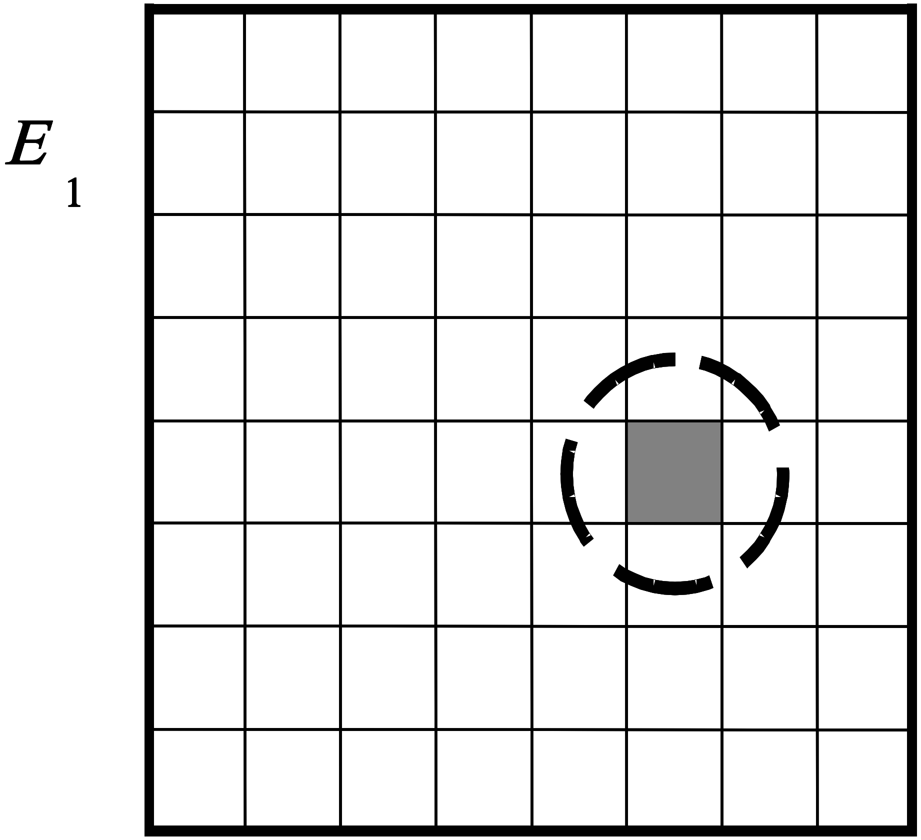
- We need to keep these coefficients. $\boldsymbol{C}^{(n,l)}$ for each level as we will need it in the downward pass
- Then use S/S translations to go up level by level up to level 2.
- Cannot go to level 1 (Why?)

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 $n=1,\ldots,2^{ld}$ $l=L,\ldots,2$
- 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*)





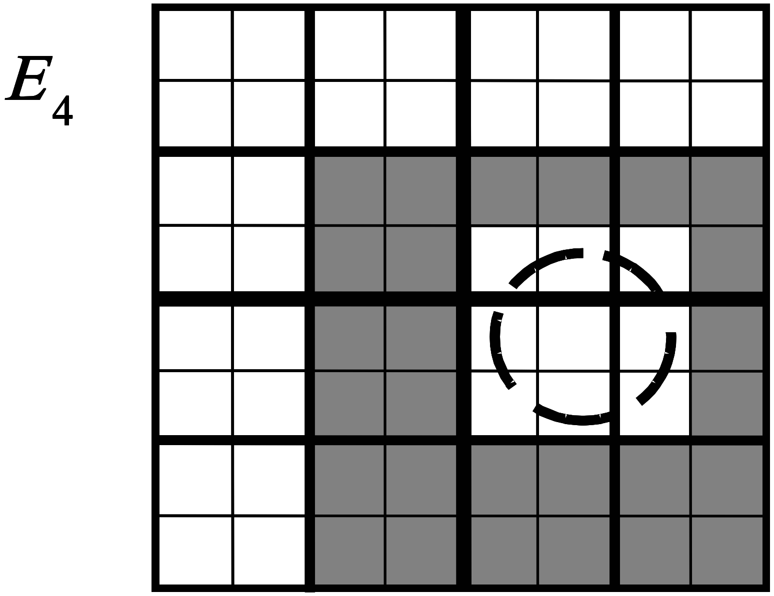
DOWNWARD PASS

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

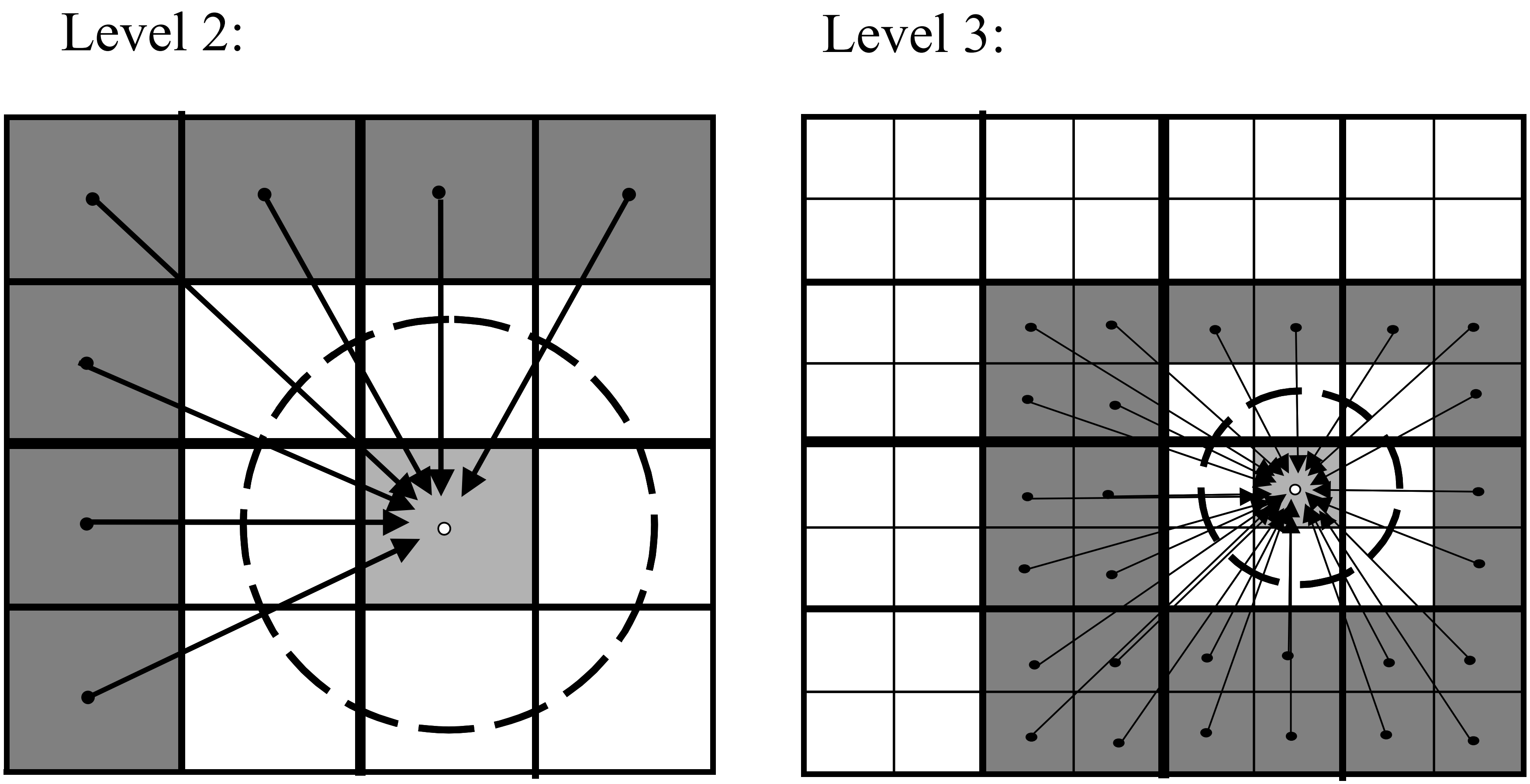
$$\Phi_4^{(n,l)}(\mathbf{y}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

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

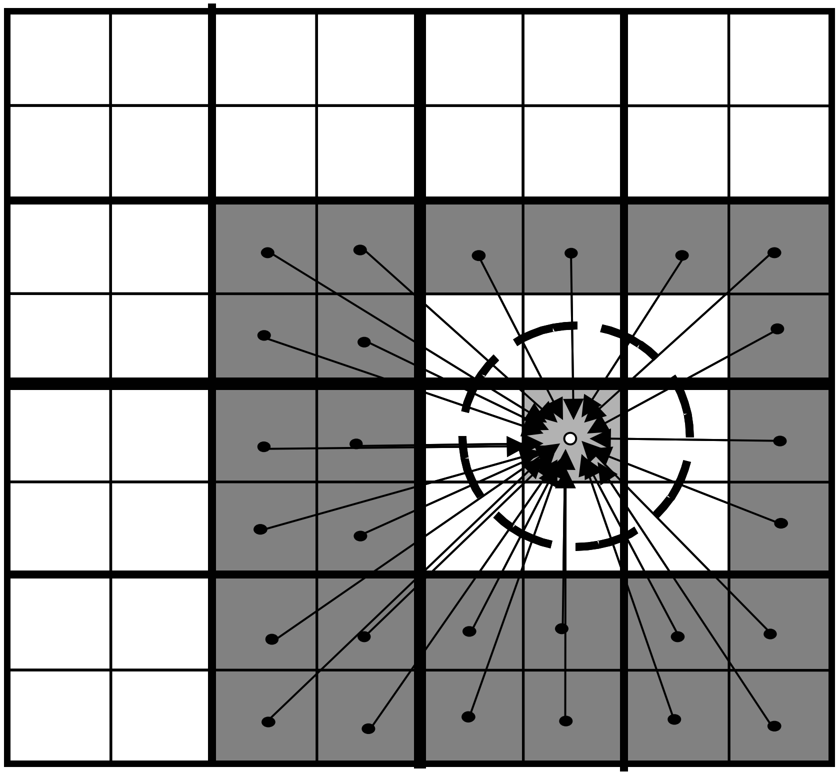
- Must to do *S*|*R* translation
- The *S* expansion is not valid in boxes immediately surrounding the current box
- So we must exclude boxes in the E_4 neighborhood



Downward Pass. Step 1.



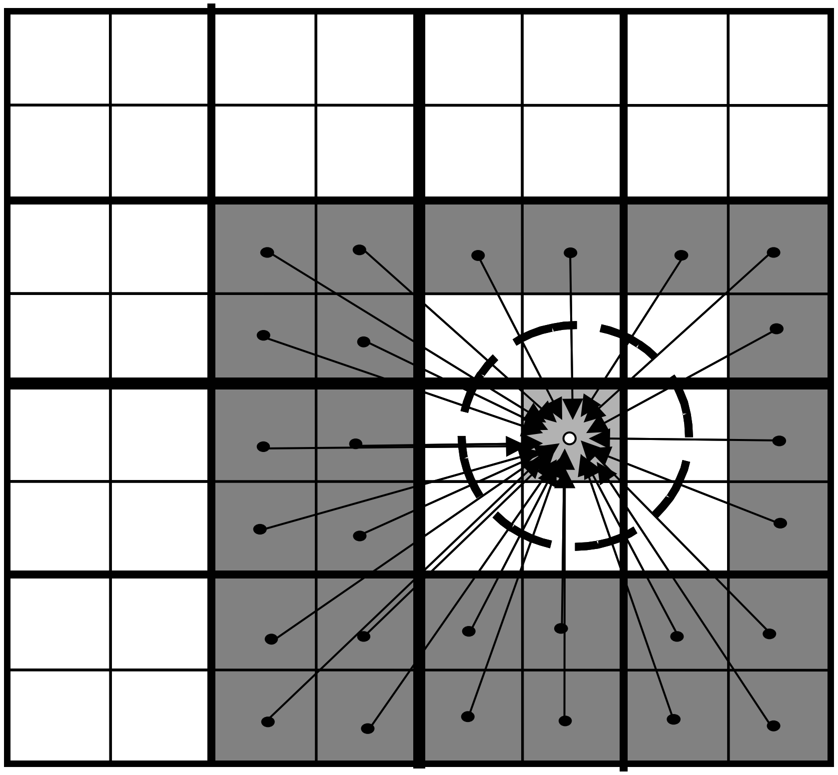
Downward Pass. Step 1.

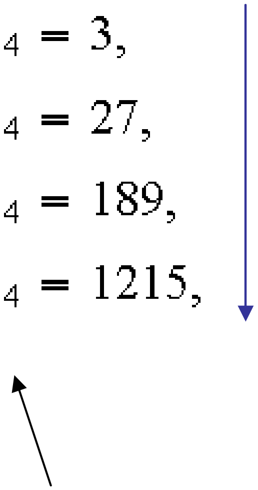


THIS MIGHT BE
THE MOST EXPENSIVE
STEP OF THE ALGORITHM

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)

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.

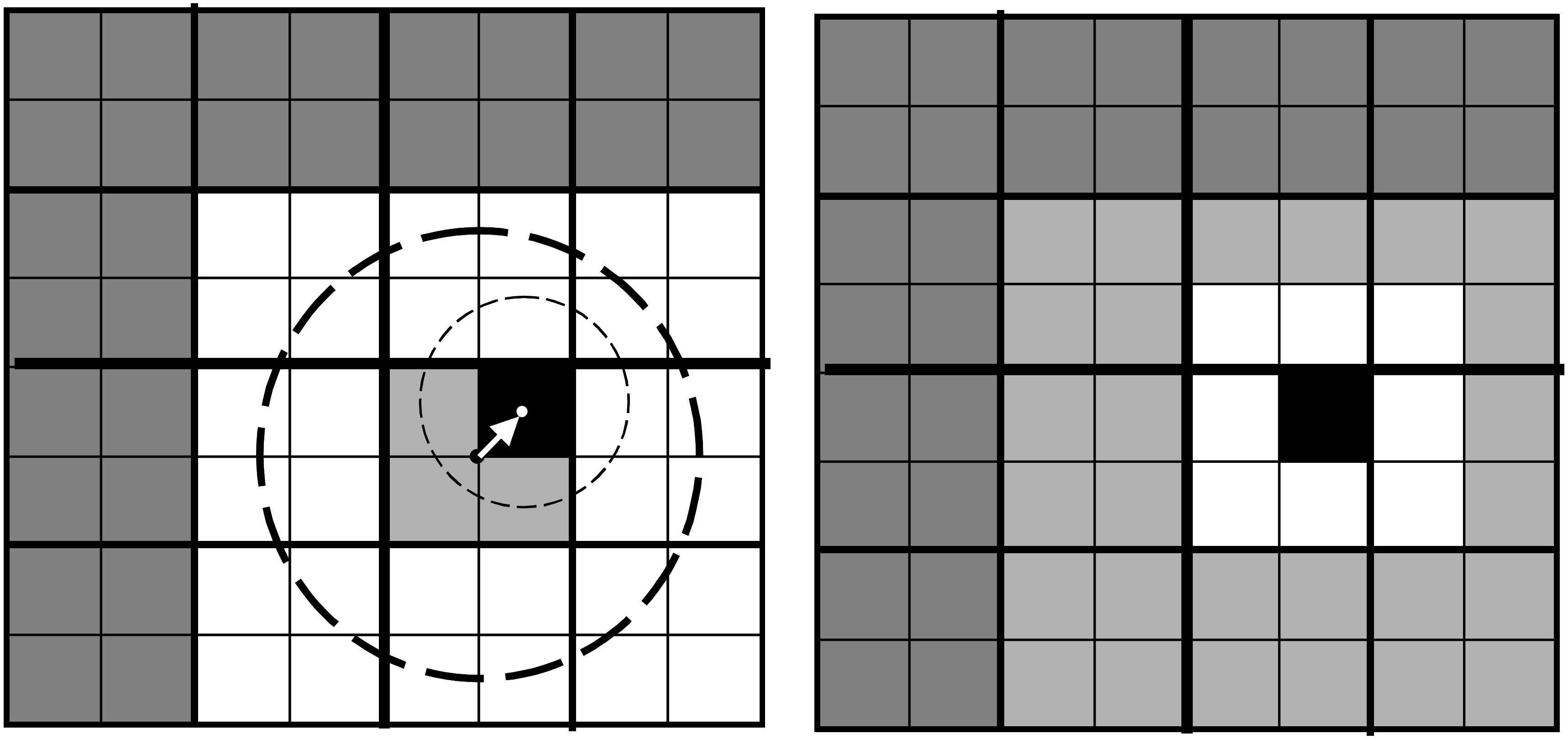


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

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

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

Lectures 14 – 16

- Adaptive multilevel FMM.
- Data structures. *D*-trees and *C*-forests.
- Adaptive MLFMM algorithm. Discussion of multilevel FMMs
- Data structures and elements of the algorithms.
- More insight into the MLFMM.
  - Neighborhoods and dimensionality.
- Methods for solution of linear systems
- Iterative methods (Conjugate gradient, Krylov, etc.)
- Use of the FMM in iterative solvers.

Lecture 17-20, Homework 7

- Error bounds for the MLFMM.
- A scheme to obtain the error bounds.
- Error for sequences of translations.
- MLFMM for $(x-y)^{-1}$ in 1-D
- Error bounds for the MLFMM
- Error bounds and optimization.
- $S|S$, $R|R$, and $S|R$-translation errors.
- 2D, 3D potential fields, particle simulations, RBF. Complex potentials. Example problems.
- 3D problems and vector analysis. Green functions. Green's identity.

Lecture 20 - 22

- Boundary element method.
- Reduction of 3D problems to forms for FMM use. Spherical Harmonics.
- (Guest Lecturer: Prof. D. Healy) Fast spherical transform and applications. Spherical harmonics. Properties.
- Algorithm. Introduction, examples, and computational results .
- 3D Laplace and Helmholtz equations. Multipoles.
- Translation theory. Recursive computations.
- Rotation-coaxial translation decomposition.
- Multipoles. Differentiation/recursion.

Lectures 23-25

- Research directions in the FMM.
  - Identification of problems and possible ways of their solution.
  - Scaling, adaptivity, etc.
- Integral transforms and fast translations. Integral transforms.
  - Signature function.
- Sparse matrix decomposition. Examples for the 3D Helmholtz equation.
- Review …