Parallel and Distributed Systems Lab.

Department of Computer Sciences
Purdue University.

Jie Chi, Ronaldo Ferreira, Ananth Grama, Tzvetan Horozov,
Ioannis Ioannidis, Mehmet Koyuturk, Shan Lei, Robert Light,
Ramakrishna Muralikrishna, Paul Ruth, Amit Shirsat

`http://www.cs.purdue.edu/homes/ayg/lab.html`
`ayg@cs.purdue.edu`

Acknowledgements: National Science Foundation, Department of Energy (Krell Fellowship), Department of Education (GAANN Fellowship).

Areas of Research:

- High Performance Computing Applications
- Large-Scale Data Handling, Compression, and Data Mining
- System Support for Parallel and Distributed Computing
- Parallel and Distributed Algorithms

High Performance Computing Applications:

- Fast Multipole Methods
  - Particle Dynamics (Molecular Dynamics, Materials Simulations)
  - Fast Solvers and Preconditioners for Integral Equation Formulations
  - Error Control
  - Preconditioning Sparse Linear Systems
- Discrete Optimization
- Visualization

Large-Scale Data Handling, Compression, and Mining:

- Bounded Distortion Compression of Particle Data
- Highly Asymmetric Compression of Multimedia Data
- Data Classification and Clustering Using Semi-Discrete Matrix Decompositions.

System Support for Parallel and Distributed Computing:

- MOBY: A Wireless Peer-to-peer Network
- Scalable Resource Location in Service Networks
- Scheduling in Clustered Environments

Parallel and Distributed Algorithms:

- Scalable Load Balancing Techniques
- Metrics and Analysis Frameworms (Isoefficiency, Architecture Abstractions for Portability)

Introduction to Multipole Methods.

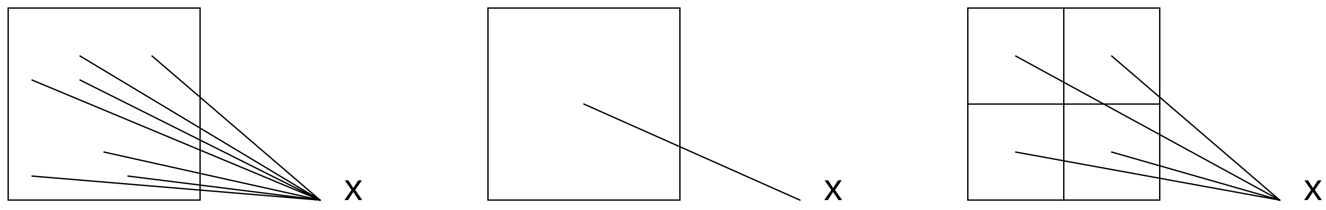
- Many systems can be modeled as a set of interacting entities such that each entity impacts every other entity.

  Examples: Bodies in space, Charged particles, Electrons and holes in semiconductors, Vortex blobs in fluids.

- Influence of an entity diminishes with distance (either in an oscillatory or non-oscillatory manner).
- Aggregate impact of several particles into a single expression (a multipole series).
- Hierarchical methods provide systematic methods of aggregation while controlling error.

Introduction to Multipole Methods.

- Represent the domain hierarchically using a spatial tree.



Set of particles impacting x.

Can they be approximated by their center of mass?

If not, divide domain and recursively apply the same criteria to sub-domains.

- Accurate formulation requires $O(n^2)$ force computations.
- Hierarchical methods reduce this complexity to $O(n)$ or $O(n \log n)$.
- Since particle distributions can be very irregular, the tree can be highly unbalanced.

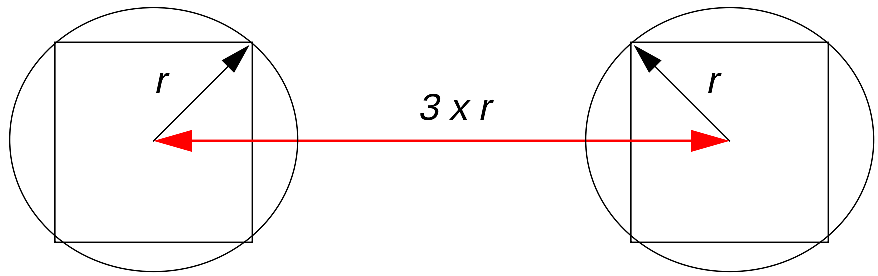
Introduction to Multipole Methods.

Fast Multipole Method (FMM)

```
construct hierarchical representation of domain
/* top down pass */
for each box in tree {
  construct well separated box list

  for each box in well separated list
    translate multipole expansion of box and add
}
for each leaf node
  translate series to particle position and apply
```

Well separatedness criteria:



Introduction to Multipole Methods.

- Each of the three phases, tree construction, series computation, and potential estimation are linear in number of particles n for uniform distributions.
- For non-uniform distributions, the complexity can be unbounded!
- Using box collapsing and fair-split trees, we can make the complexity distribution independent.

Introduction to Multipole Methods.

Solving Boundary Element Problems:

Boundary element methods result in a dense linear system:

$$E(\chi) = E_o(\chi) + \gamma \int_{\Phi} E_o(\chi) f(\chi, X) dX$$

- $E(X)$ is the known physical quantity (boundary value),
- $E_o(X)$ is the unknown (Both are defined over the domain ϕ).
- $f(a,b)$ is a function of points a and b and is a decaying function of the distance r between a and b.

Introduction to Multipole Methods.

Boundary Element Method for Integral Equations:

Solution of the integral form of Laplace equation:

- $E(X)$: specified boundary conditions,
- $E_o(X)$: vector of unknowns,
- The function f is the Green's function of Laplace's equation.

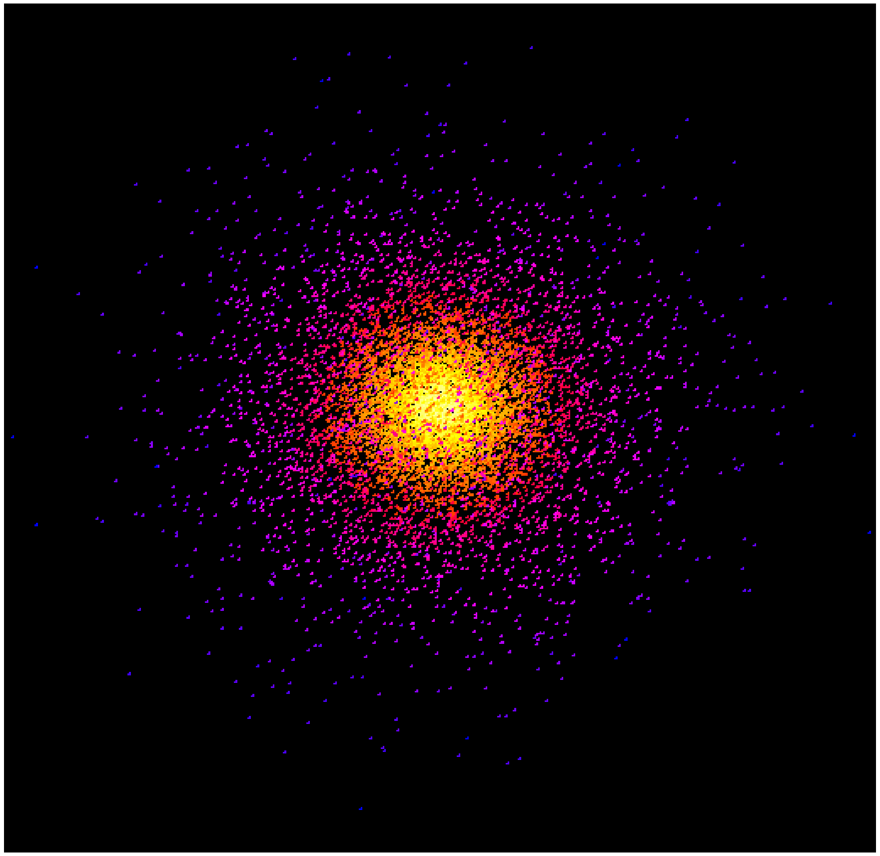
$f(r) = -log\ (r)$ (in two-dimensions)

$f(r) = 1/r$ (in three-dimensions)

Introduction to Multipole Methods.

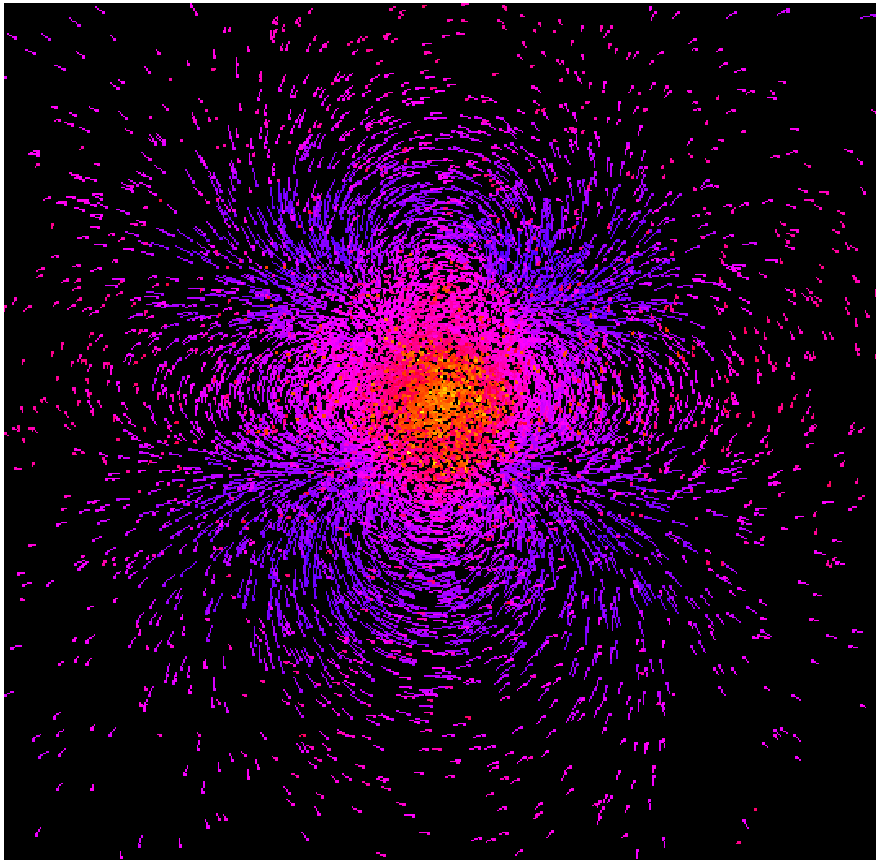
- Boundary integral forms are particularly suited to problems where boundary conditions cannot be easily specified.
- For example, while solving the field integral equations (EFIE/MFIE/CFIE), the associated Green's function (e^{ikr}/r) implicitly satisfies boundary conditions at infinity. This obviates the need for absorbing boundary conditions.
- Surface integral equations are, however, infeasible for non-homogeneous media, consequently, a mixed finite element / boundary element approach is often used.

Experimental Results:



Sample charge distribution.

Experimental Results:



Force vectors at charges.

Experimental Results.

Timing / Efficiency results of the force computation routine on a Cray T3D.

Problem	Alpha	Interactions	*P = 64* Time	*P = 64* Eff	*P = 256* Time	*P = 256* Eff
g_28060	0.7	9058880	1.81	0.744		
	1.0	6477568	1.18	0.823		
p_41776	0.7	11990208	2.27	0.787	0.61	0.729
	1.0	8223552	1.49	0.822	0.40	0.761
p_120062	0.7	36079624	6.53	0.827	1.75	0.766
	1.0	25670656	4.40	0.869	1.18	0.813
g_650691	0.7	105038751			4.94	0.793
	1.0	73010765			3.29	0.828

(all times in seconds)

Performance of the SGI Origin Implementation.

Problem	Serial		Parallel	
	Classical	Accelerated	Classical	Accelerated
uniform40K	195.46	155.41	6.68 (29.26)	5.07 (30.65)
ip46K	360.93	295.68	11.67 (30.92)	9.40 (31.46)

Parallel Performance: Serial (single thread) and parallel times of the algorithms along with their speedups (in parenthesis) on a 32 processor Origin 2000 (alpha = 0.91, beta = 2.0, degree = 6).

Multipole-based Boundary Element Solvers.
(3-D Laplace Equation)

Convergence and runtime of the GMRES solver.

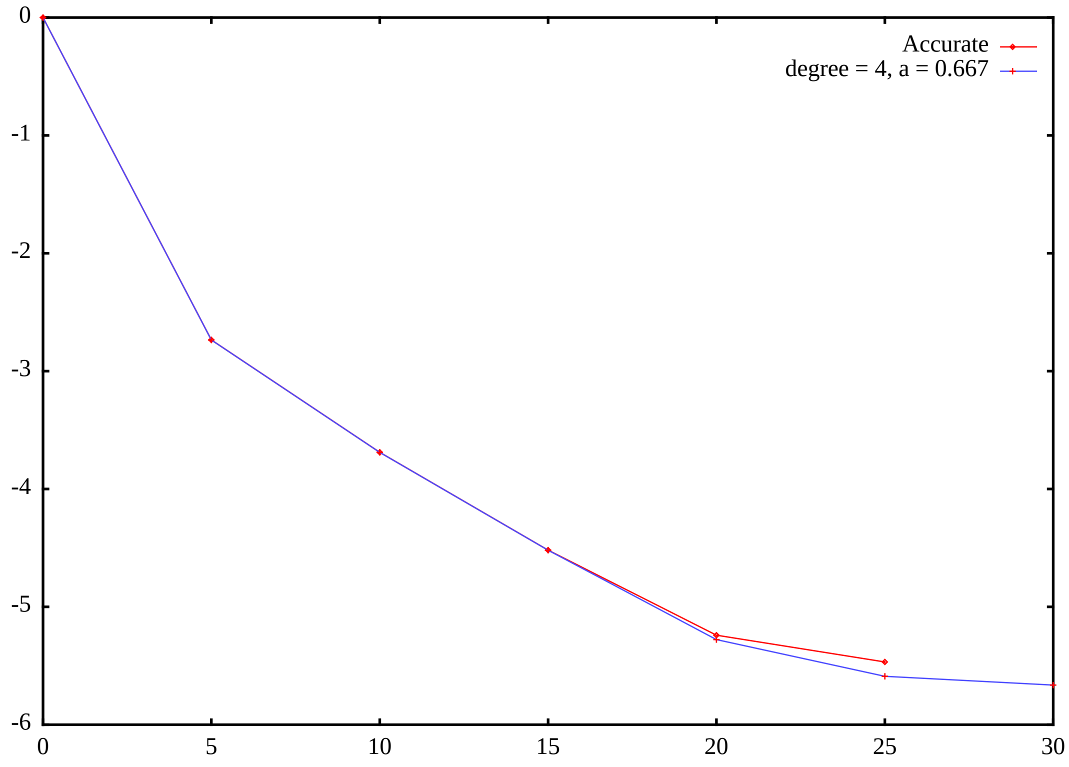
Log_{10} of Relative Error Norm.

Iteration	Accur.	a = 0.5		a = 0.667	
		deg. = 4	deg. = 7	deg. = 4	deg. = 7
0	0.000000	0.000000	0.000000	0.000000	0.000000
5	-2.735160	-2.735311	-2.735206	-2.735661	-2.735310
10	-3.688762	-3.688920	-3.688817	-3.689228	-3.689304
15	-4.518760	-4.518874	-4.518805	-4.519302	-4.518911
20	-5.240810	-5.260901	-5.260881	-5.278029	-5.261029
25	-5.467409	-5.521396	-5.510483	-5.589781	-5.531516
30	-5.627895	-5.626917	-5.663971	-5.627989	
Time		124.46	156.19	92.16	112.02

All times in seconds. Timings taken on a 64 processor Cray T3D (24192 unknowns).

Multipole-based Boundary Element Solver.
(3-D Laplace Equation)

Convergence of the GMRES solver (only accurate and fastest approximation with degree = 4 and *a* = 0.667 shown).



Multipole-based Boundary Element Solver.
(3-D Laplace Equation)

Preconditioning the GMRES solver

- ❏ Inner Outer Scheme:

  This scheme uses an inner iteration solve based on a low accuracy (lower degree, lower a) hierarchical method.

- ❏ Block Diagonal Preconditioner:

  Nodes are aggregated in groups of n nearest neighbors. The corresponding (truncated) system is factorized a-priori. This factorized matrix is used for approximate solves in the preconditioner.

Multipole-based Boundary Solvers.
(3-D Laplace Equation)

Preconditioning the GMRES solver

Log_{10} of Relative Error Norm.

Iteration	a = 0.5 deg. = 7	Inner-outer scheme	Block diagonal
0	0.000000	0.000000	0.000000
5	-2.735206	-3.109289	-2.833611
10	-3.688817	-5.750103	-4.593091
15	-4.518805		-5.441140
20	-5.260881		-5.703691
25	-5.510483		
30	-5.663971		
Time	156.19	125.40	103.61

All times in seconds. Timings taken on a 64 processor Cray T3D.

Using Multipole Methods for Preconditioning Sparse Iterative Solvers.

Problem Formulation

- ❑ Arises in simulation of time-dependent Navier-Stokes equations for incompressible fluid flow
- ❑ One of the most time consuming steps
- ❑ Large scale 3D domains
- ❑ Multiprocessing is indispensable
- ❑ Robust, parallel preconditioners required

Nature of the System and Preconditioning.

Linear System

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

where

$A = \alpha M + \nu T$, in which $\alpha \cong \frac{1}{\Delta t}$ and $\nu \cong \frac{1}{Re}$

M : Mass matrix; T : Laplace operator
B : Gradient operator (n x k)

Properties of the Linear System

Symmetric indefinite (n +ve and k -ve eigenvalues)

Typically $k = \frac{n}{8}$ (2D) or $k = \frac{n}{24}$ (3D)

Uzawa-type Methods

Solve $B^T A^{-1} B p = B^T A^{-1} f$ using iterative method

Accelerate by CG

Assumption : div-stability $\Rightarrow B^T A^{-1} B$ is well-conditioned[1] (steady state)

Single-level iterative schemes with suitable preconditioning

Key Issues

Two-level nested iteratitve schemes

Expensive iterations due to inner iterative solver

1. Condition number independent of mesh discretization

Adapting Dense Methods to the Preconditioning Problem

Use a dense solver to compute the preconditioner for the matrix A.

The dominant behavior of matrix A is $(\nabla\nabla - k^2)$.

The Green's function of this operator is $\frac{e^{-kr}}{r}$.

Issue: Implementing boundary conditions?

Implementing Boundary Conditions for Dense Preconditioner

Analogous problem in potential theory: Compute the potential over a domain resulting from a set of given charges provided the boundary potential is pre-specified.

Solution strategy:

Assume (unknown) charges residing on the boundary.

The result of the boundary and internal charges result in the boundary conditions. Use this to compute the values of the unknown boundary charges.

Finally, use these boundary charges along with given internal charges to compute the potential in the interior.

Computational steps: a dense boundary element solve of an $n \times n$ system (for n boundary nodes) followed by a dense mat-vec with an $n \times n$ system.

Preconditioning Effect of Dense Solvers:

Preconditioning of Hierarchical Approximate Techniques

n_i	Incomplete Factorization	Hierarchical Approximation
297	14	9
653	20	14
1213	25	14
2573	35	16
4953	45	18

Performance and Errors in Hierarchical Methods:

Theorem 1. In Barnes-Hut method, the ratio r_s/r for particle-cluster interactions is bounded as follows:

$$\alpha' \le r_s/r \le \alpha$$

where α and α' are constants.

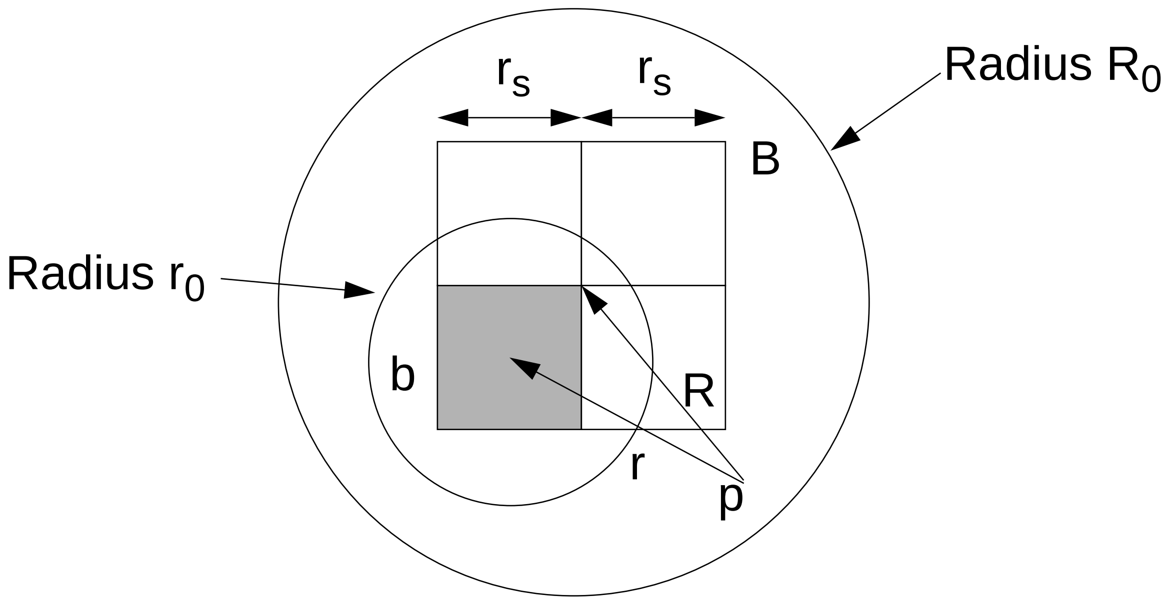
Proof: Since a particle *p* interacts with a box *b*, but not with its parent box *B*,

$$r \ge r_0 \qquad R \le R_0$$

where $r_0 = r_s/\alpha$ and $R_0 = 2r_s/\alpha$. Use triangle inequality $R + r_s/\sqrt{2} \ge r$ to show that

$$\left(\frac{2}{\alpha} + \frac{1}{\sqrt{2}}\right)^{-1} \le r_s/r \le \alpha$$

Interactions in Hierarchical Methods:



Theorem 2. In Barnes-Hut method, a particle interacts with a bounded number of boxes of fixed size.

Proof: Since $\alpha' \le r_s / r \le \alpha$, the centers of all boxes of size r_s lie within annulus defined by

$$\frac{r_s}{\alpha} \le r \le \frac{r_s}{\alpha'}$$

and the boxes lie completely within the annulus defined by

$$\frac{r_s}{\alpha} - \frac{r_s}{\sqrt{2}} \le r \le \frac{r_s}{\alpha'} + \frac{r_s}{\sqrt{2}}$$

The ratio of the volumes of the annulus and a box gives the following upper bound on the number of boxes of size r_s :

$$n_{max} \le \frac{4\pi}{3}\left[\left(\frac{1}{\alpha'} + \frac{1}{\sqrt{2}}\right)^3 - \left(\frac{1}{\alpha} - \frac{1}{\sqrt{2}}\right)^3\right]$$

Theorem 3. Suppose that k charges of strengths {q_j , j=1,...,k} are located within a sphere of radius r_s. Then, for Barnes-Hut method with α-criterion for well-separatedness, the error in potential outside the sphere at a distance r from the center of the sphere due to these charges is bounded by

$$\varepsilon \le \frac{A}{r - r_s} \cdot \left(\frac{r_s}{r}\right)^{p+1} \le \frac{A}{r_s} \cdot \frac{\alpha^{p+2}}{1-\alpha}$$

where p is the degree of the truncated multipole expansion such that $p > 1$, and

$$A = \sum_{j=1}^{k} |q_j|$$

Theorem 4. Controlling error in B-H

$$p_k = p + \frac{kd}{\log\alpha} + \frac{\log A - \log A_k}{\log\alpha}$$

The theorem defines a growth rate for the polynomial degree with the net charge inside a subdomain such that the total error associated with the subdomain remains constant.

For a uniform distribution, this growth rate can also be expressed in terms of the domain sizes.

Theorem 5. Error in the piecewise approximate B-H method $O(\alpha^{p+1} \log A)$

This follows naturally from the following results:

- The number of interactions with subdomains at any level are constant.

- The number of subdomain interactions is logarithmic in number of particles (independent of particle distribution -- sequence of theorems on fair-split trees omitted).

- The error associated with a single particle-subdomain interaction is constant.

Theorem 6. Computational complexity of the piecewise approximate B-H method

$$O(n(p+l)^3)$$

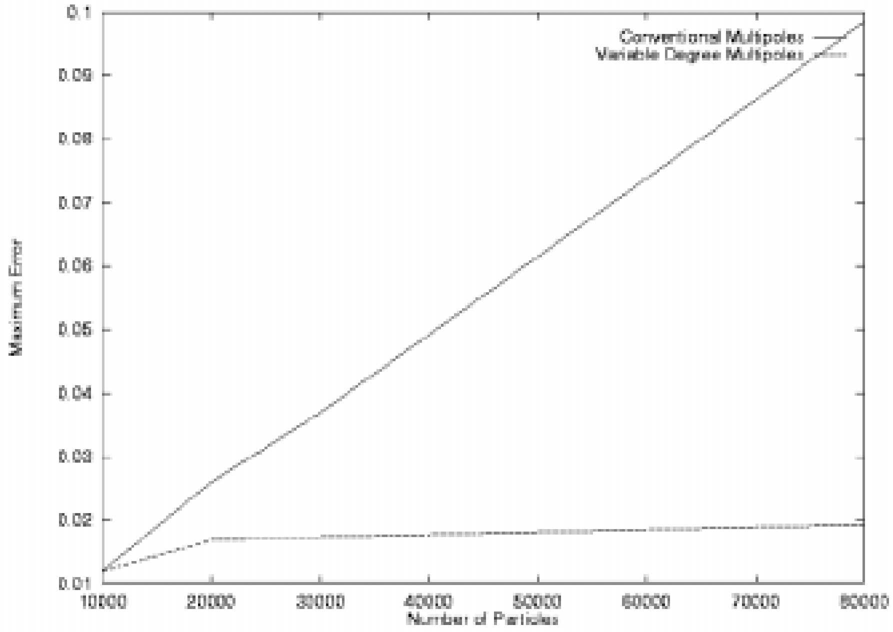
where l is the levels of the hierarchical decomposition.

For a uniform distribution, l grows as $\log_8 n$.

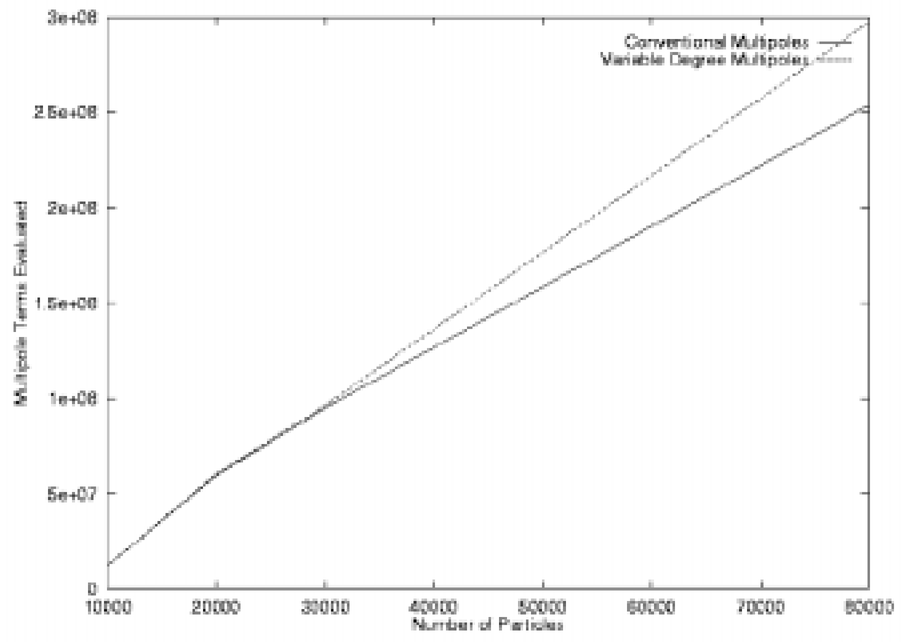
For l < p, we can show that the operation count of the new method is within a fraction 7/3 of the fixed-degree multipole method. For smaller values of l this difference is smaller.

For typical values of p (6 - 7 degree approximations), this corresponds to between 256K - 2M node points. Thus, even for very large scale simulations, the improved method is within a small constant off the fixed-degree method while yielding significant improvements in error.

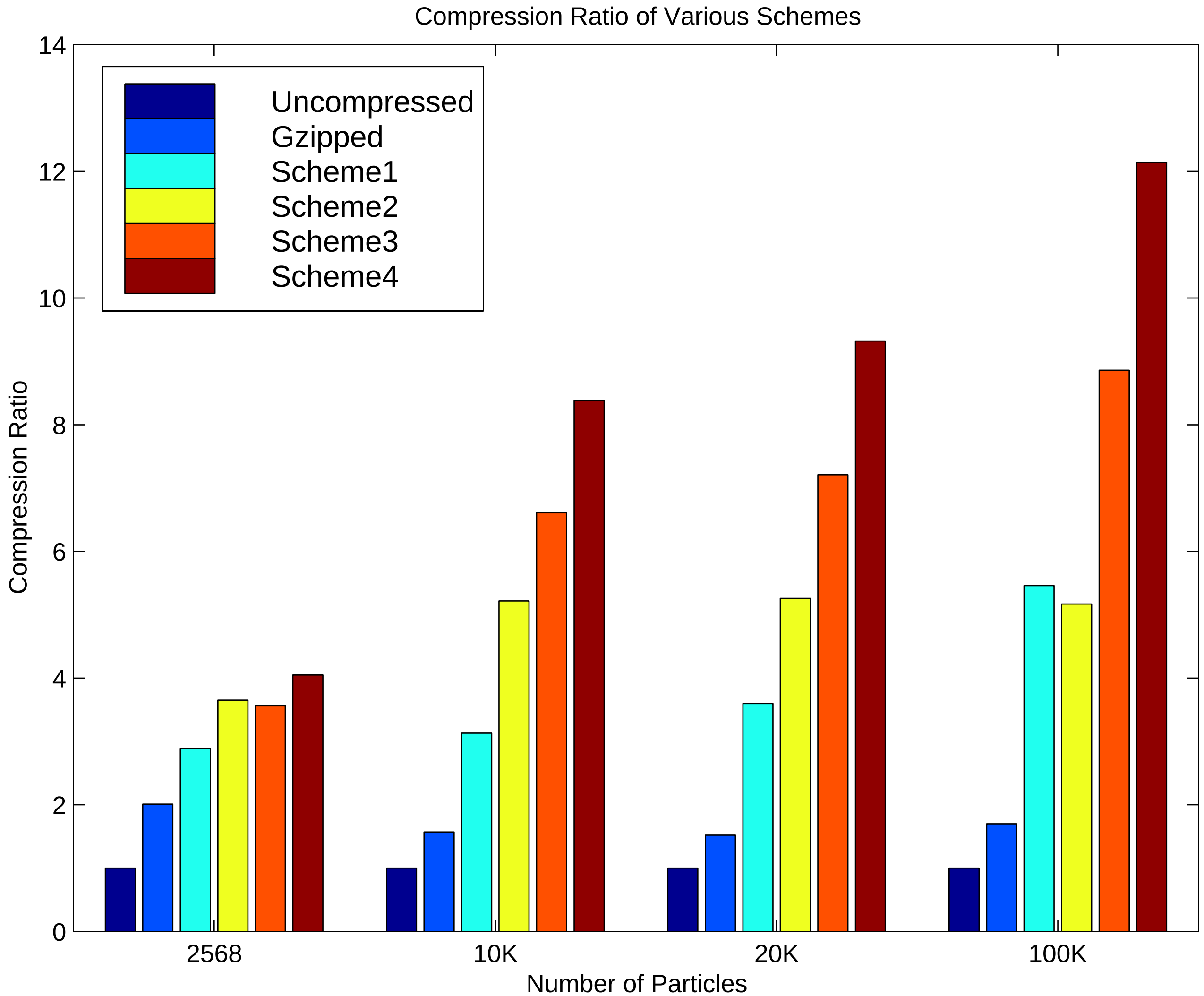
Comparison of Errors



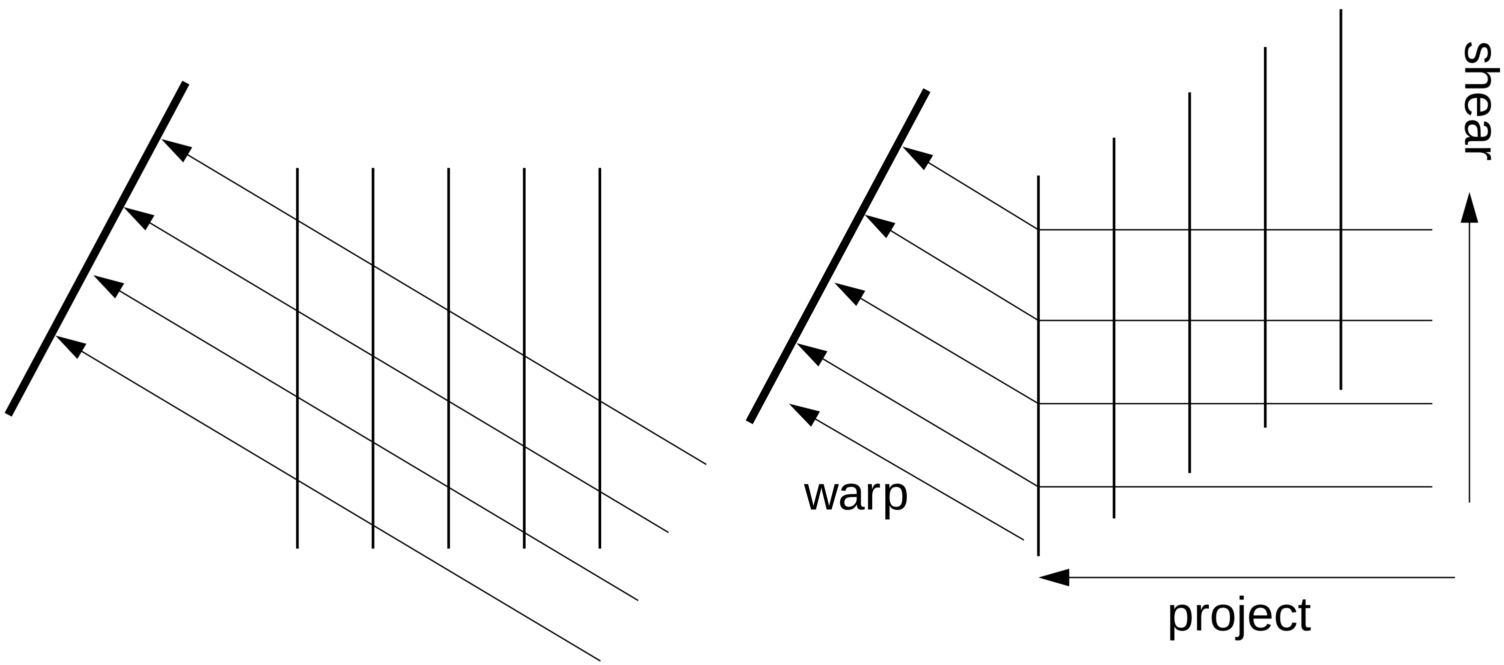
Comparison of Runtimes



Bounded Error Pointset Compression Results



Some Other Parallel Applications: Shear-Warp.



Optimizations for volume Rendering:

Early Termination:

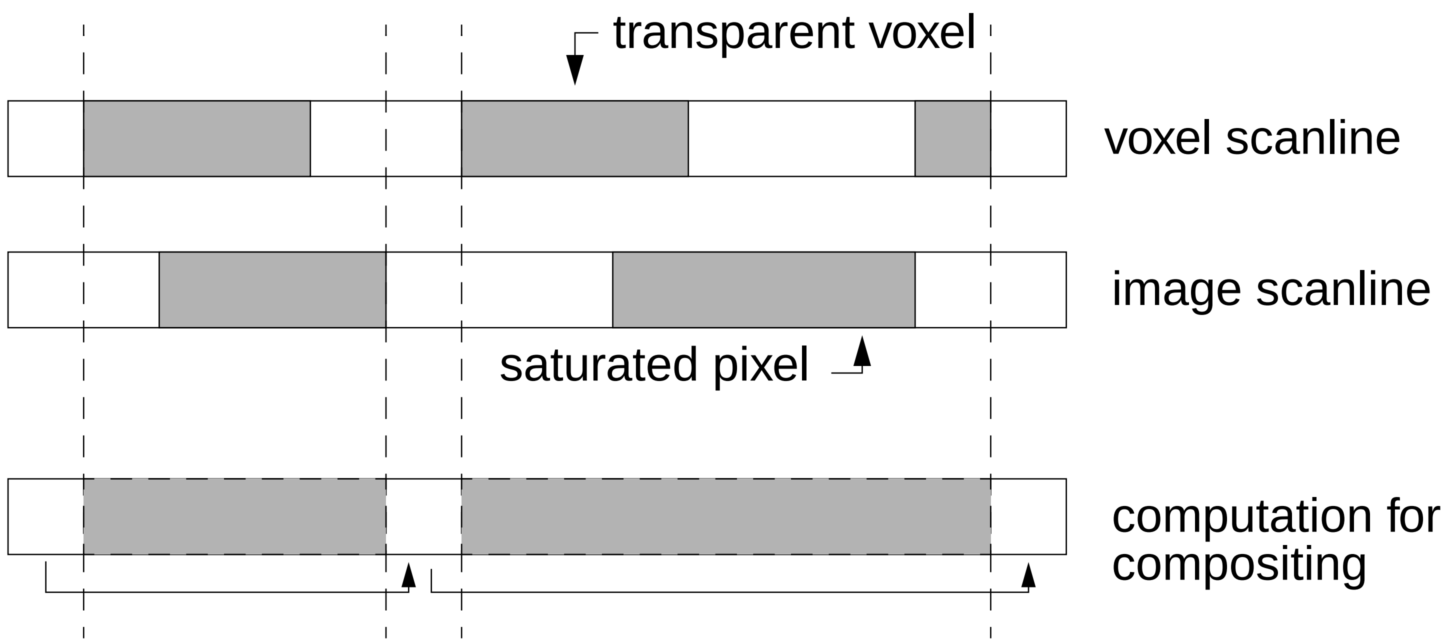
Instead of traveling back to front, it is possible to travel front to back. In this case, it is possible to stop when the accrued opacity is high enough that additional slices do not make any difference.

Skipping Empty Spaces:

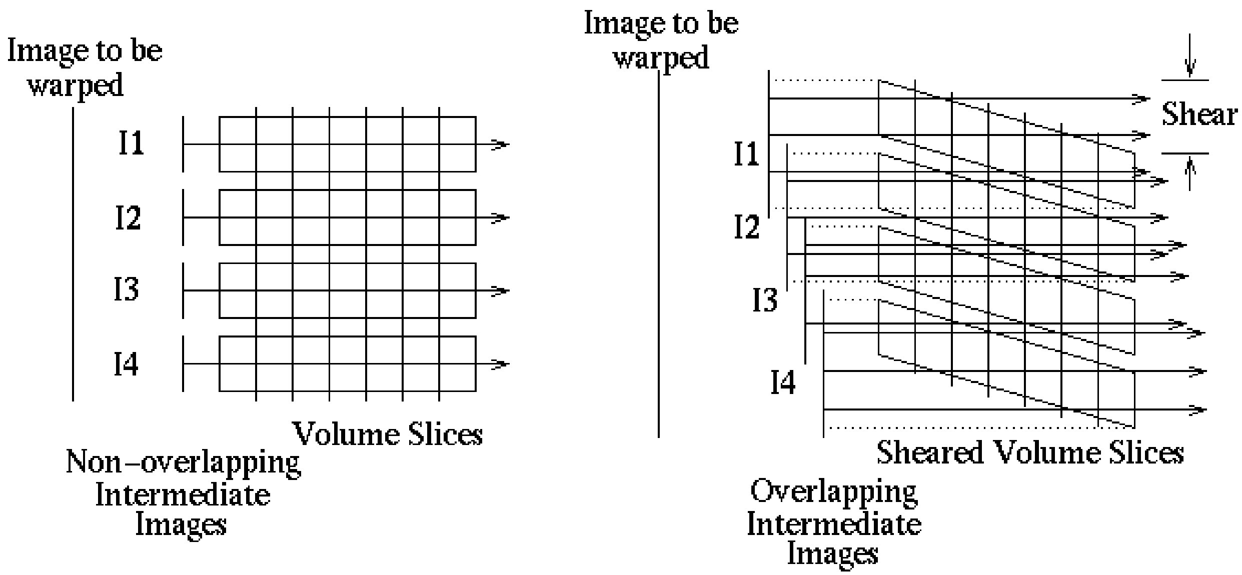
In typical datasets, a significant part of the volume is empty (the opacity is 0). These voxels need not be traversed. Using smart data-structures, it is possible to skip these all-together. Run-length encode the scanlines.

Optimizations for Volume Rendering

Compositing volume slices: skip transparent voxels and saturated intermediate image pixels.

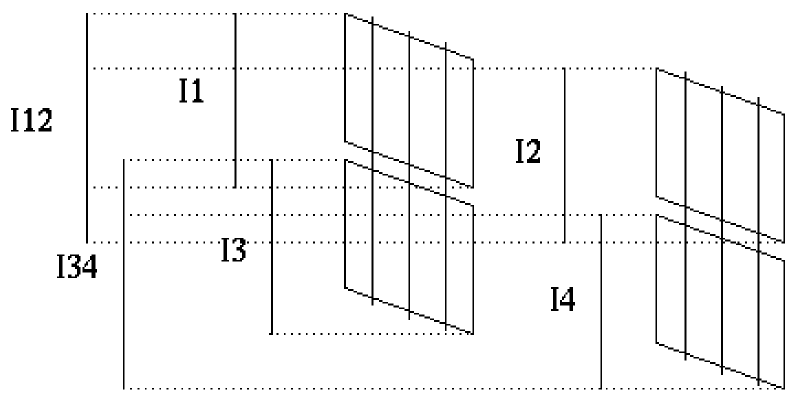


Parallel Shear-Warp: Volume Space Partitioning



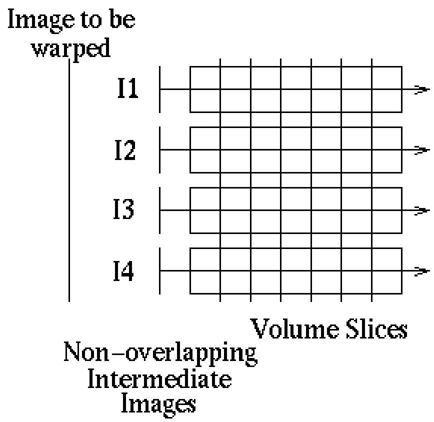
(a) Striped partitioning, no shear

(b) Striped partitioning, with shear

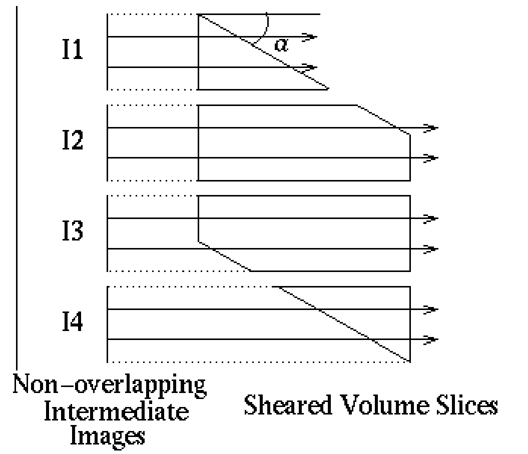


(c) Two−dimensional volume partitioning with shear

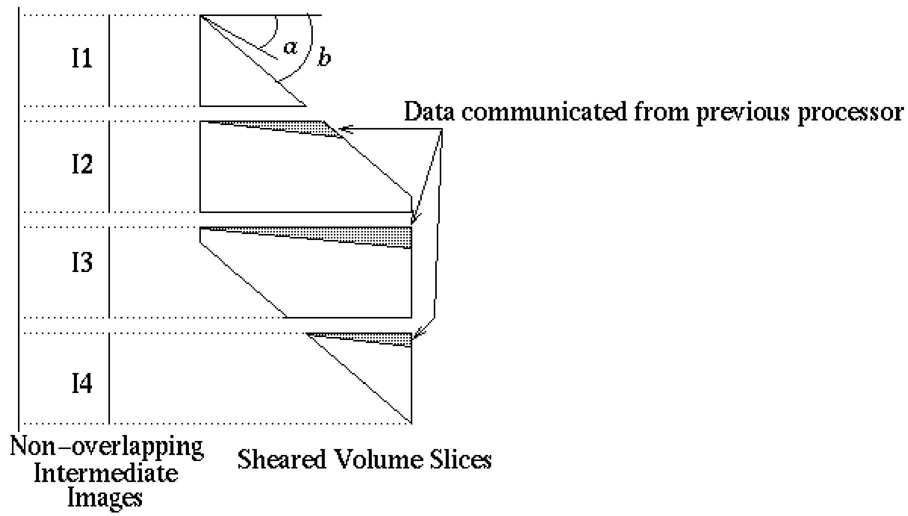
Parallel Shear-Warp: Sheared Volume Space Partitioning



(a) sheared volume space partitioning, no shear



(b) sheared volume space partitioning, shear:



(c) relative shear of $b-\alpha$ and resulting communication

Load Balancing

- ❑ Due to optimizations such as run-length encoding and early termination, different scanlines can have widely varying workloads.
- ❑ Naively partitioning the sheared volume among processors leads to significant load imbalance.
- ❑ It is impossible to determine the load associated with a scanline accurately a-priori.
- ❑ Since the viewpoint is not expected to shift very drastically between frames, we can use load information from one frame to balance load in the next.
- ❑ Each processor keeps track of load at each scanline. At the end of computation, processors exchange this information and rebalance load. The communication can be integrated with shear.

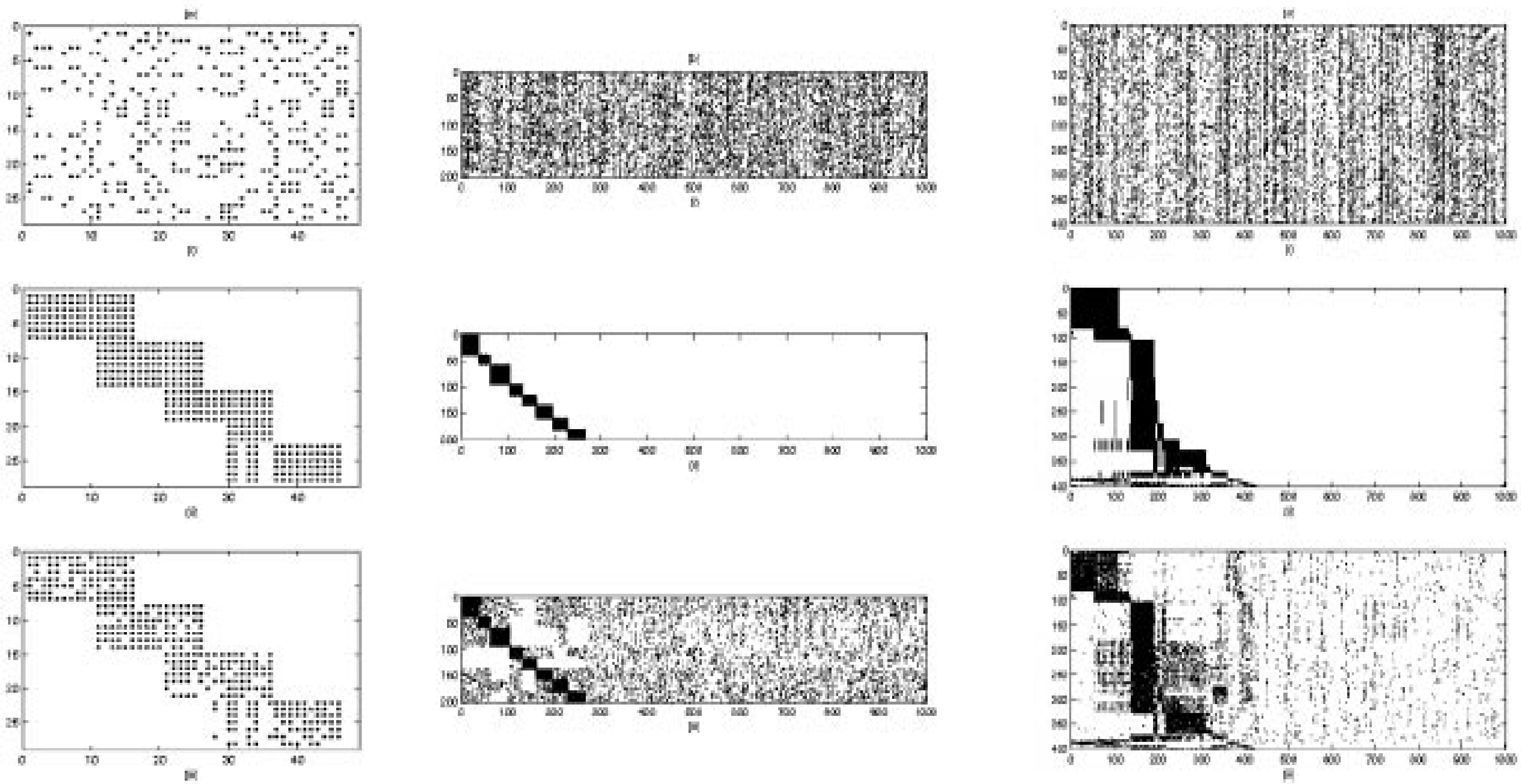
Experimental Results (the brain dataset)

Rendering Time (ms)

p	volume			
	No load balance		Load Balanced	
	Large	Small	Large	Small
1	3193	976	3193	976
2	1627	548	1625	551
4	892	309	910	310
8	620	197	593	196
16	345	127	327	121
32	216	86	204	81
64	142	74	118	70
128	103		85	

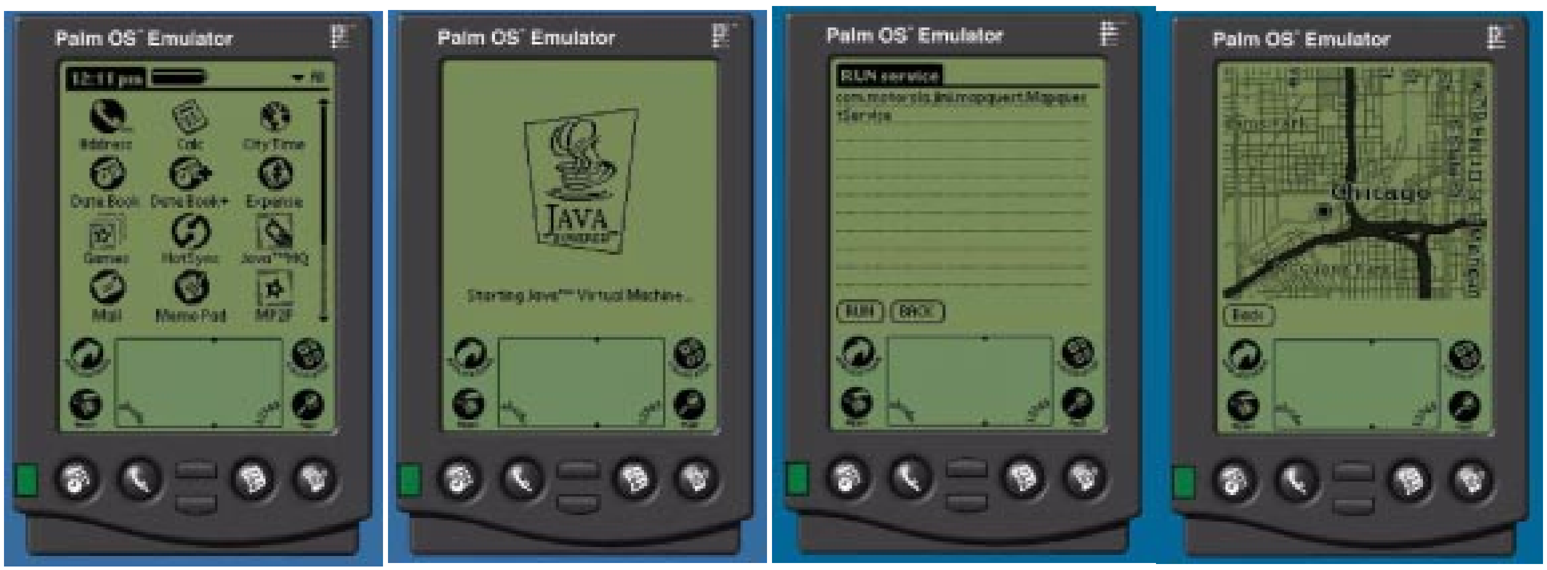
Large-Scale Data Handling, Compression, and Mining.

Proximus: a tool for bounded error compression of discrete attrubute sets.



MOBY: A Wireless P2P Network

Accessing services (software, hardware, data) from your wireless device, seamlessly!



Other Research on P2P Networks:

Evolving Topology Based on Access Patterns

Service Mobility

Dynamic Client Mapping.