Geometric Learning Algorithms for Vision, Robotics, and Graphics

Stephen M. Omohundro

International Computer Science Institute

Berkeley, California

- *Geometric* - Mathematics
- *Learning* - Statistics
- *Algorithms* - Computer Science

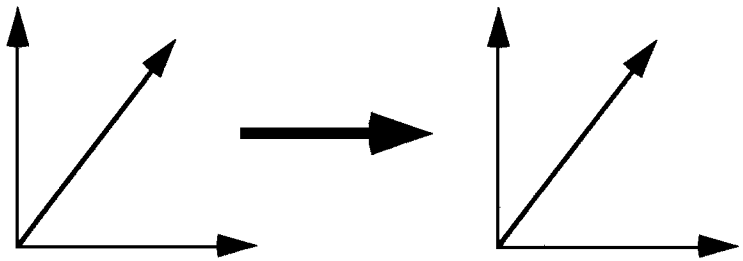
Central to the Technology of the next Century.

The Issues.

- Free labor through robotics.
- Robots need senses.
- "Knowledge is power."
- Learning.
- Connecting geometry to symbols.

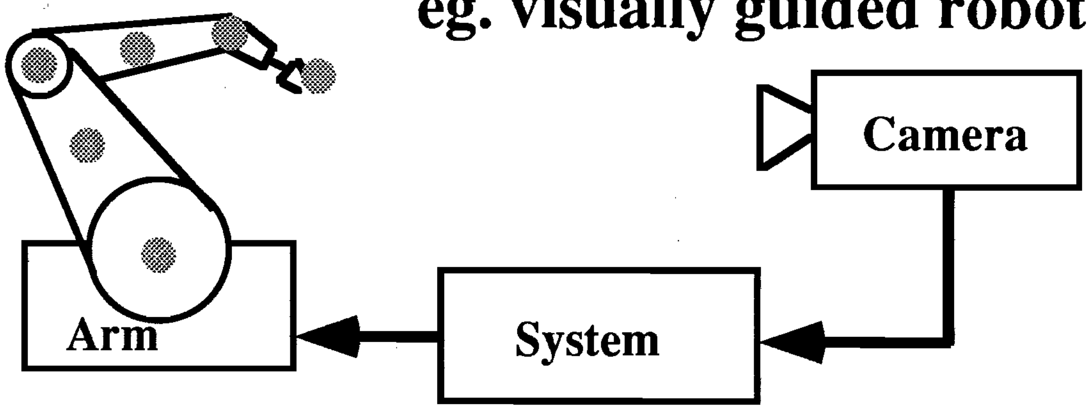
Smooth Nonlinear Mappings

Input Space



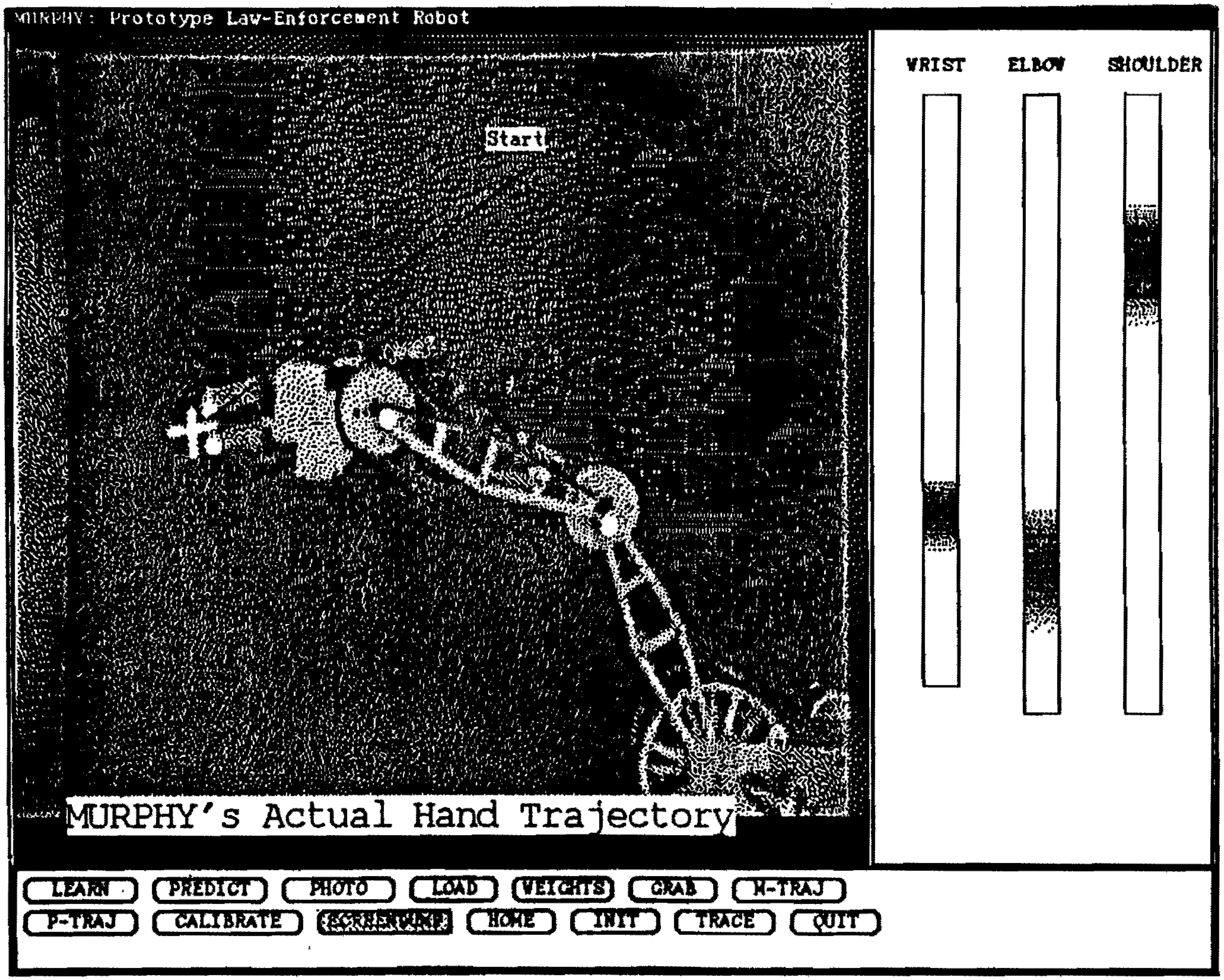
Output Space

eg. visually guided robot arm



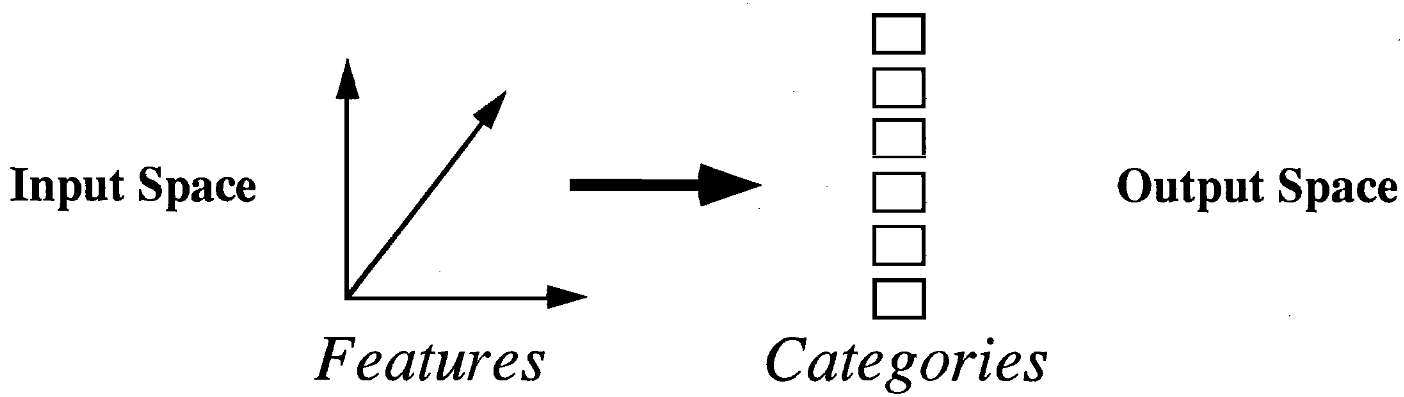
kinematic space $\longrightarrow$ visual space

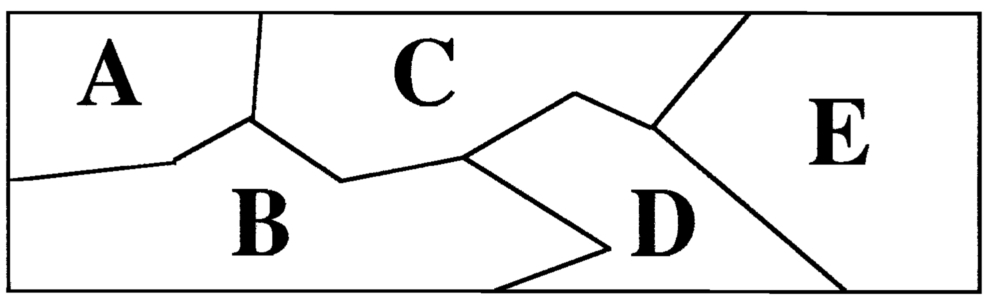
$$\mathbf{R}^3 \longrightarrow \mathbf{R}^{12}$$


MURPHY: Prototype Law-Enforcement Robot
Start
MURPHY's Actual Hand Trajectory
WRIST
ELBOW
SHOULDER
LEARN
PREDICT
PHOTO
LOAD
WEIGHTS
GRAB
P-TRAJ
CALIBRATE
HOME
INIT
TRACE
QUIT

Classification

<u>*Examples:*</u> *OCR, Speech recognition, face recognition*

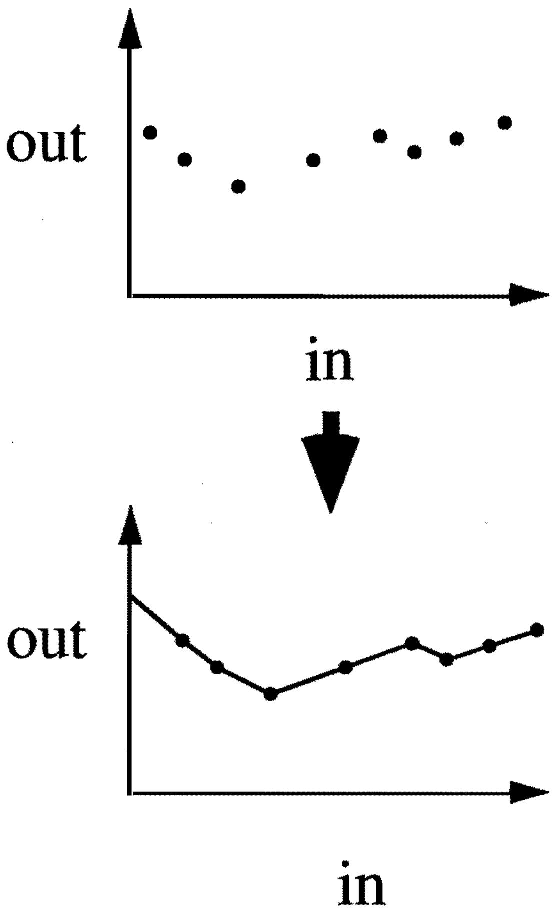




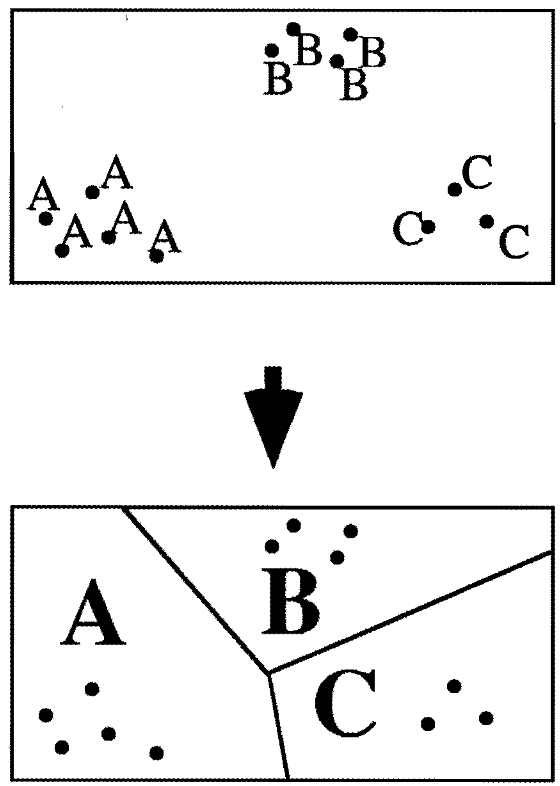
A classifier defines a partition of the feature space.

Learning From Examples

Mappings



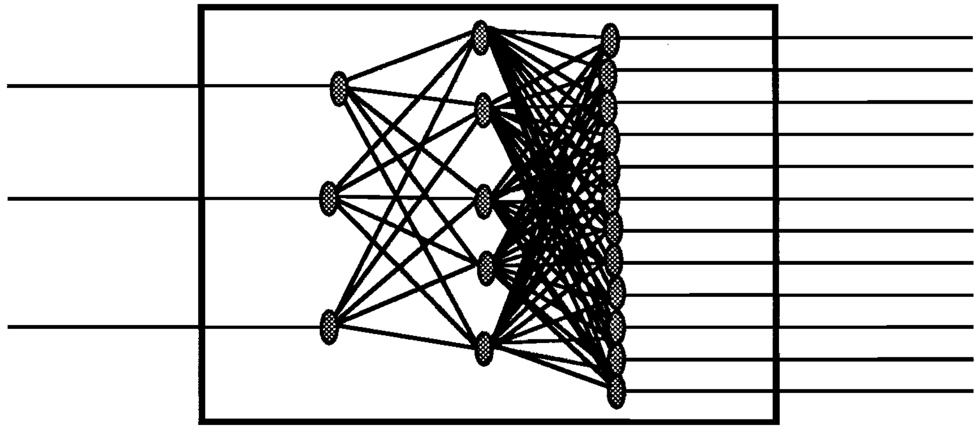
Classification



Popular Approach: Backpropagation

R^3

input space



R^{12}

output space

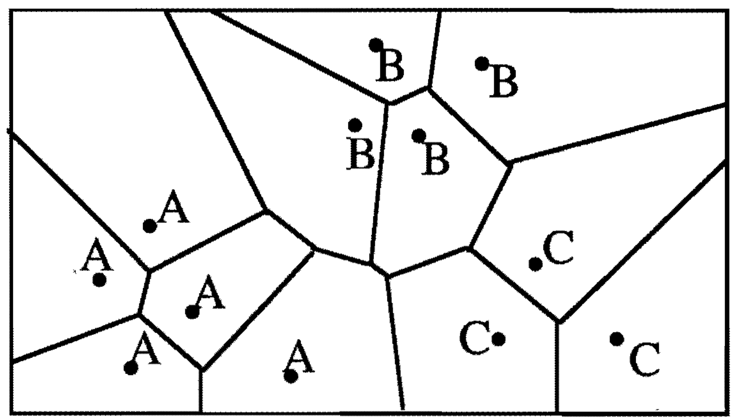
"Hill Climbing in Weight Space"

Disadvantages:

- no error bounds
- how many units, how many weights?
- training is slow and unpredictable
- gets stuck at local maxima
- poor scaling behavior
- units don't have "meaning"
- biologically implausible

Nearest Neighbor Classification

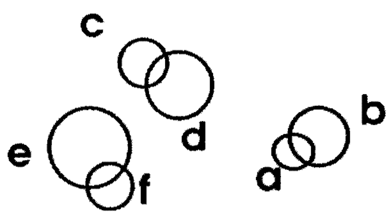
<u>Theorem</u> *(Cover and Hart)*: **Asymptotically the probability of error in using nearest neighbor classification is at most twice that of any other technique.**



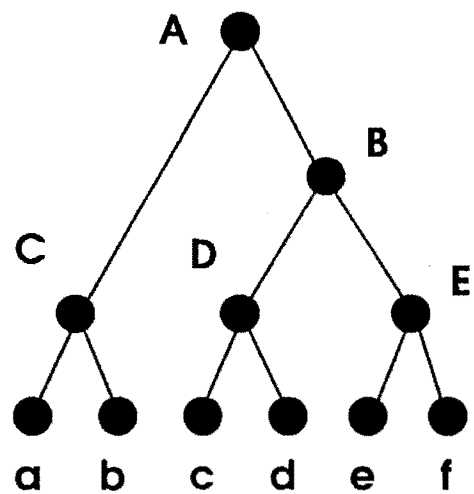
<u>Voronoi diagram</u>: **Partition of space induced by a set of points in which partition regions are all points closest to a given sample point.**

Balltrees.

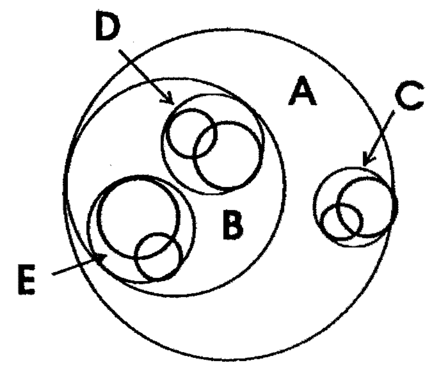
A *balltree* is a complete binary tree with a ball associated to each node such that an interior node's ball is the smallest which contains the balls of its children.



2-d leaves.

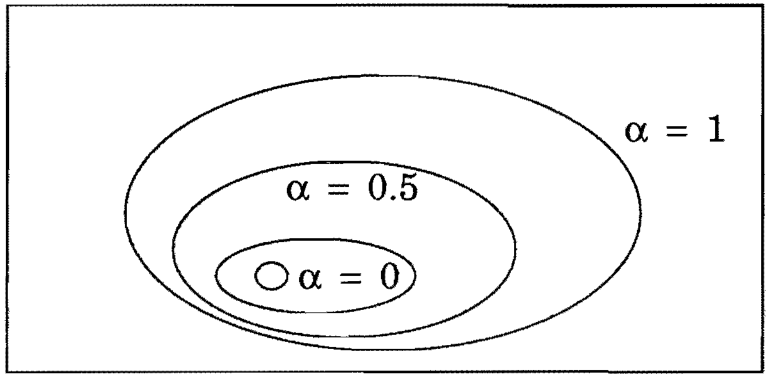


tree structure.



tree balls.

The beta distribution and non-parametric statistics.



Let S_α be a nested family of sets parameterized by α such that $p(S_\alpha) = \alpha$. If we draw N points and choose the smallest set S_α containing exactly n points, then the α's are distributed according to the Beta distribution:

$$p_n(\alpha) = \frac{N!}{(n-1)!(N-n)!}a^{n-1}(1-a)^{N-n}$$

$$E(\alpha) = \frac{n}{N+1} \rightarrow \frac{n}{N} \qquad \sigma^2(\alpha) \rightarrow \frac{n}{N^2}$$

Balltree queries.

Pruning:

- Return leaf balls containing a query point.

Branch and bound queries:

- Return nearest leaf to query point.

Distribution independent average performance:

If leaves and queries are drawn from an underlying distribution ρ then would like good performance on average with respect to ρ.

Five balltree construction algorithms.

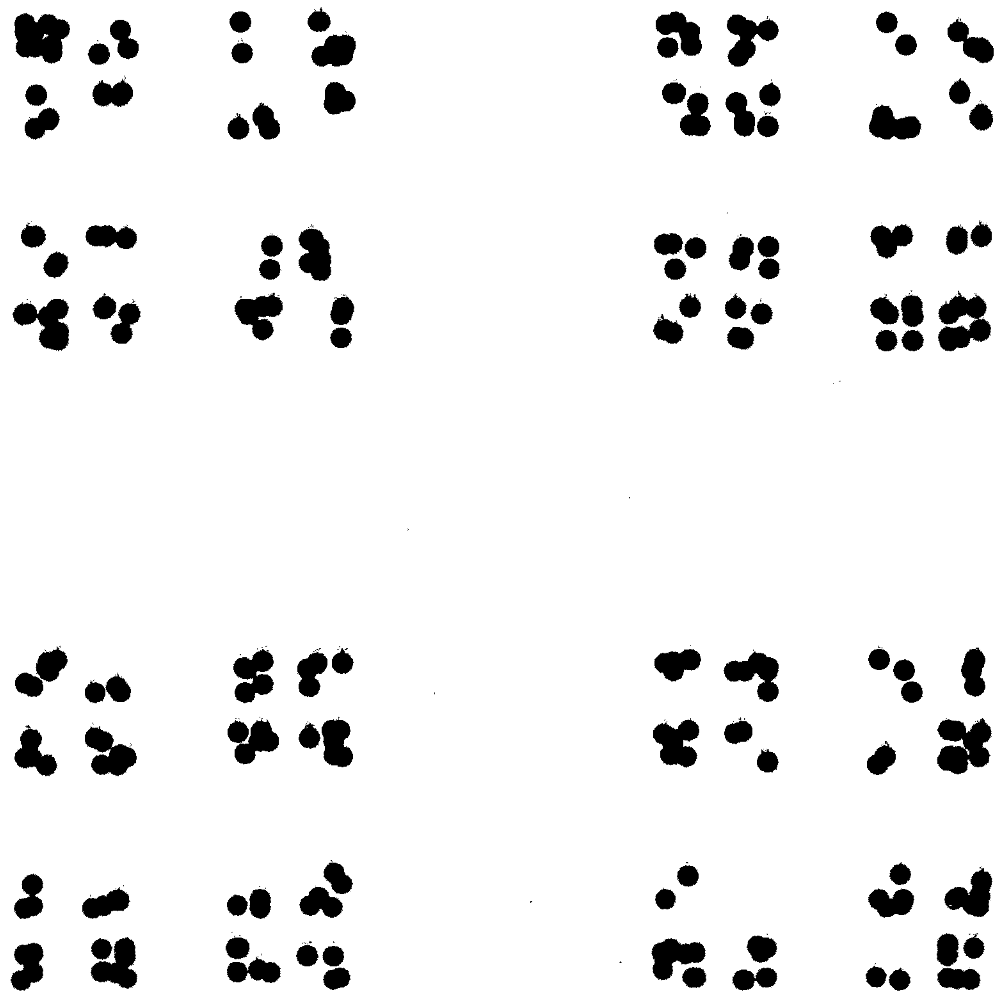
- **K-d algorithm**
  Split most spread dimension at median.
- **Top-down algorithm**
  Split best dimension at point to minimize new ball volume.
- **Insertion algorithm**
  Insert ball on-line at minimum volume insertion location.
- **Cheap insertion algorithm**
  Insert ball on-line at heuristically good location.
- **Bottom up algorithm**
  Repeatedly pair the best two balls.

Balltrees can find n nearest neighbors in log expected time.

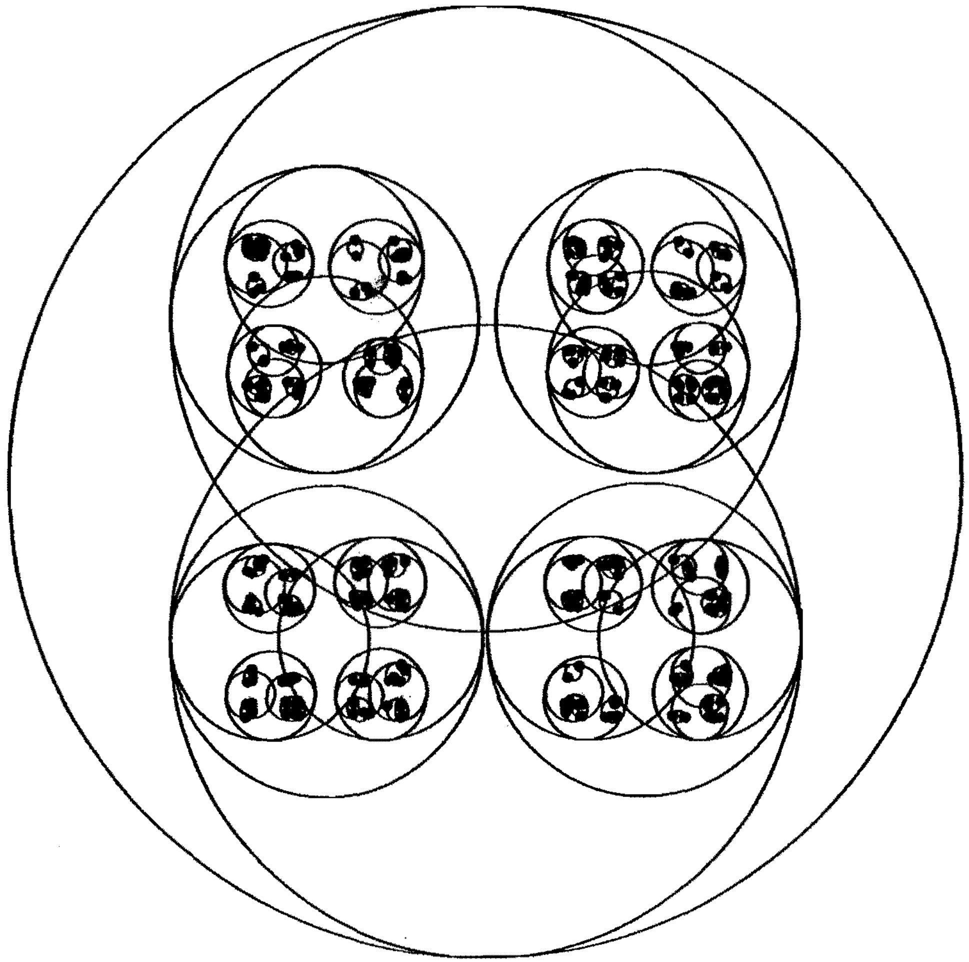
If N samples are drawn from a non-vanishing, smooth distribution on a compact region, then we can use a balltree to find the n nearest neighbors of a new sample in O(log N) expected time, asymptotically for large N.

Idea: Asymptotically, volume of n nearest neighbor ball is beta distributed. With k-d construction, balltree regions are also beta distributed. n nearest neighbor ball will overlap only a constant number of balltree balls on average. Using branch and bound, we do log time initial search and then constant extra work.

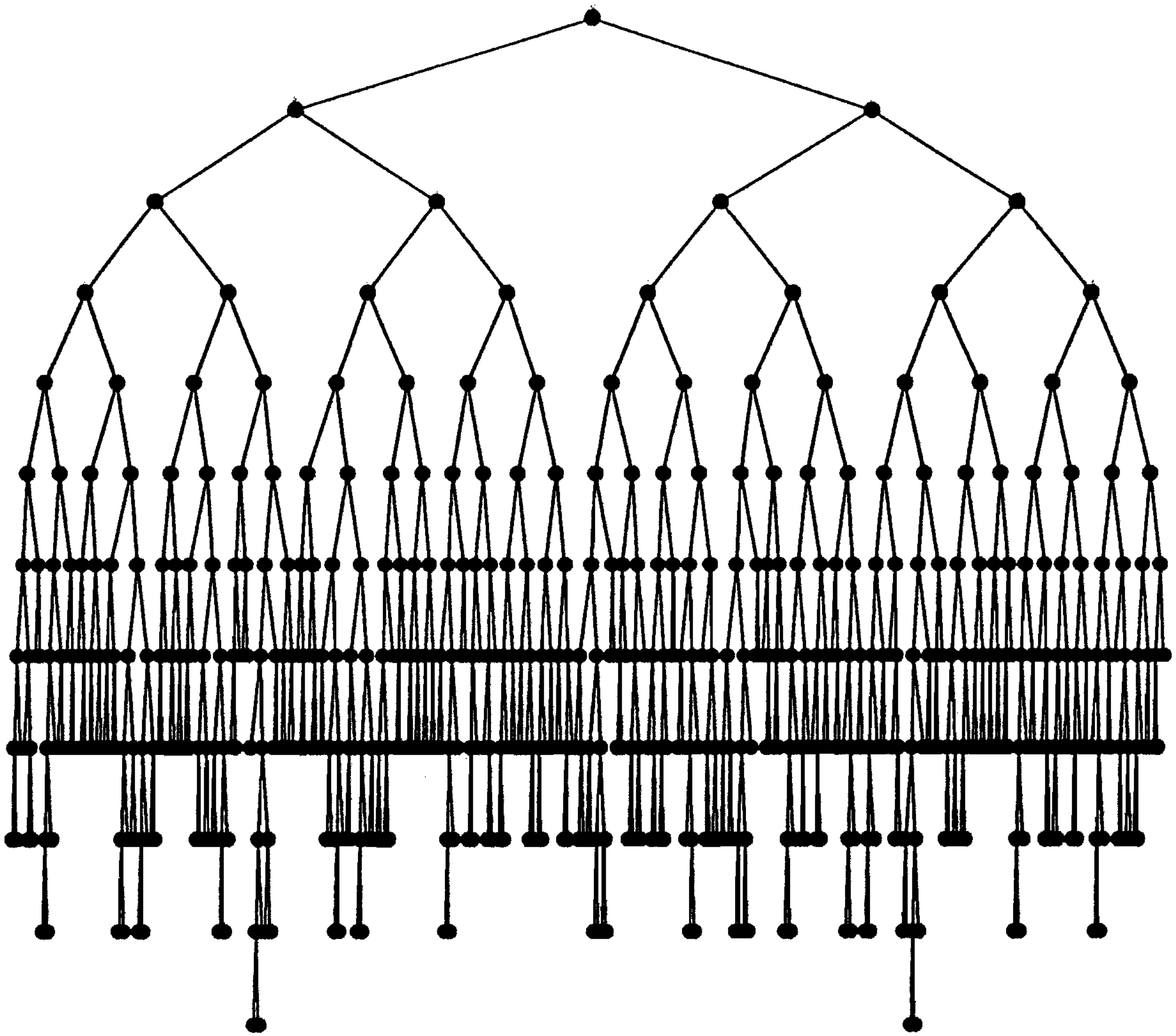
2-d Random Cantor points.



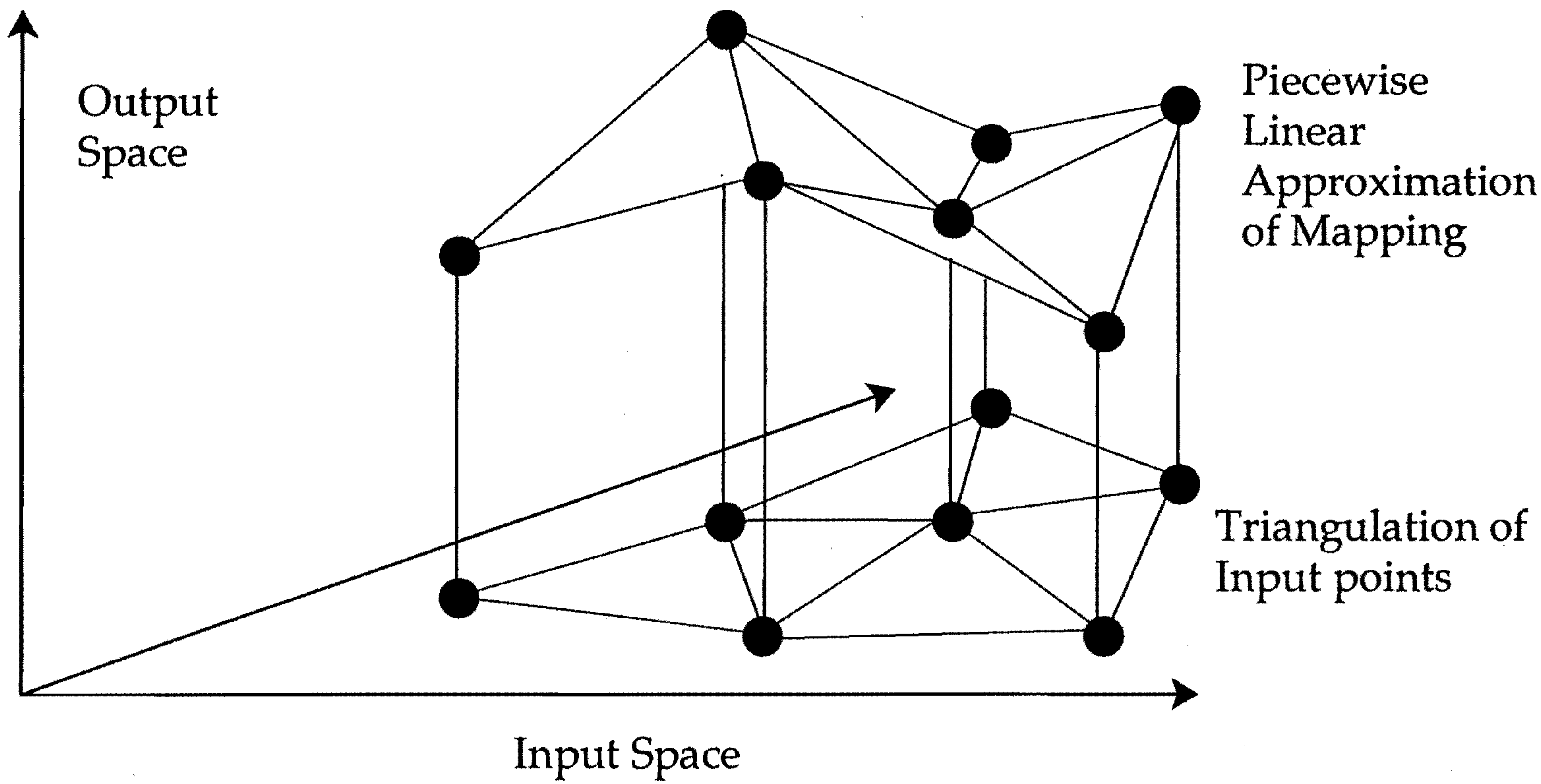
Balltree balls for bottom up construction from 2-d Cantor random points.



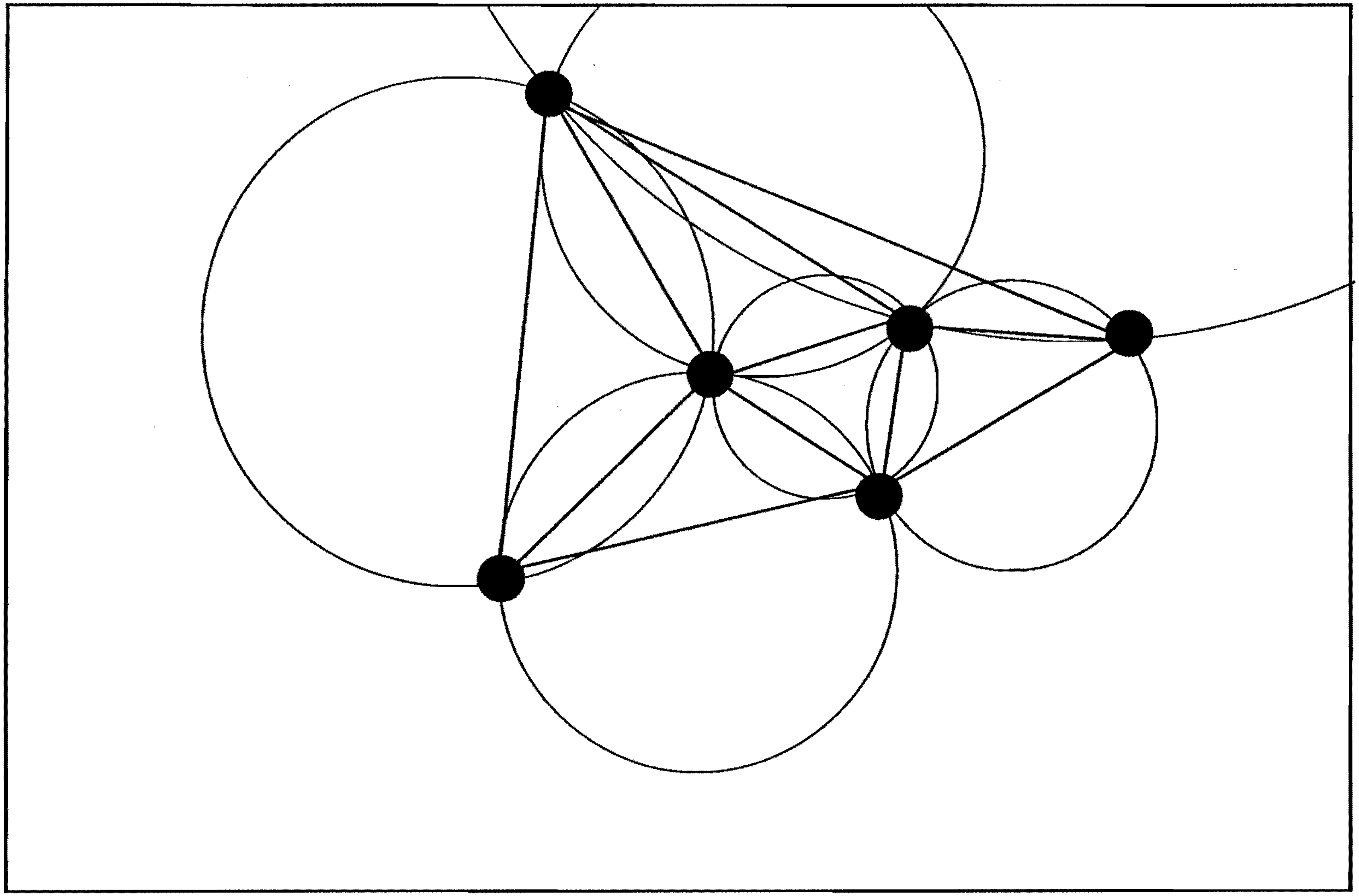
Tree structure of bottom up balltree over 2-d Cantor random points.



Triangulation for Piecewise Linear Approximation

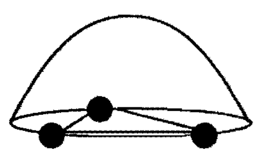


The Delaunay Triangulation



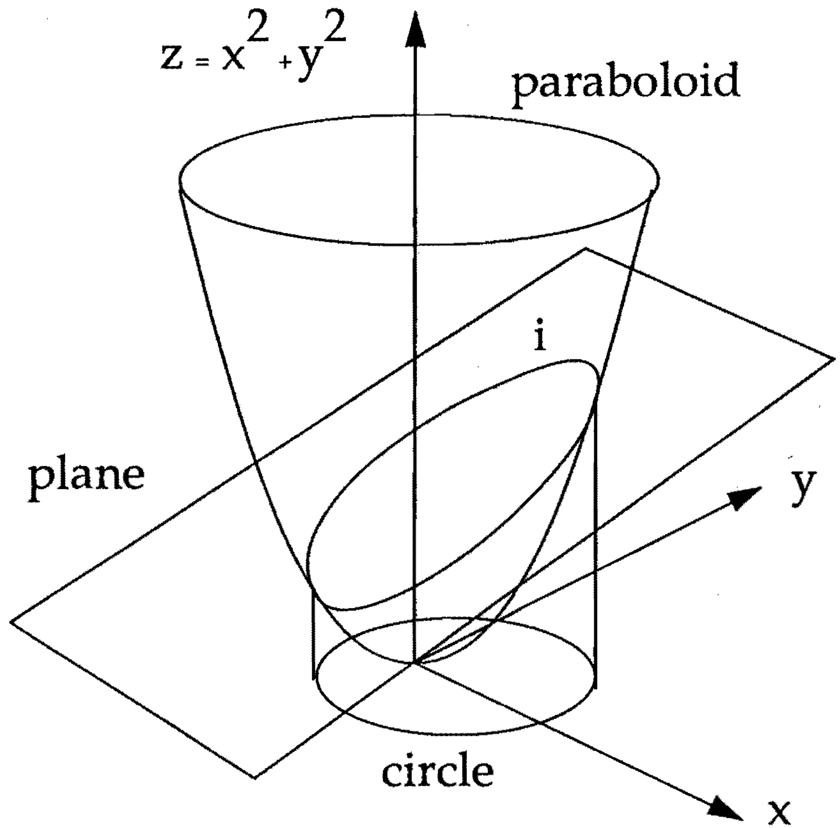
Delaunay is good for piecewise linear approximation.

Theorem: Among all triangulations of a given set of input points, the Delaunay triangulation gives the smallest worst case error at each point for piecewise linear approximation of mappings with a bounded second derivative in each direction.



Worst case error function in a simplex is quadratic, vanishing on vertices. Level sets are spheres, so error is monotonic with radius of sphere determined by sample points.

Spheres correspond to hyperplanes in the next higher dimension.



Can find Delaunay simplex in log expected time.

If N samples are drawn from a non-vanishing, smooth distribution on a compact region, then we can use a balltree to find the Delaunay simplex containing a new sample in O(log N) expected time, asymptotically for large N.

Idea: Delaunay vertices are found among n nearest neighbors with high probability where n is constant but depends on the probability bound. Nearest neighbors are described by beta distribution as are balltree balls in k-d algorithm. Expected overlap of n nearest neighbor ball with balltree balls is constant, so get logarithmic search time asymptotically.

Some Geometric Learning tasks.

- Learning smooth mappings.
- Learning discrete mapping.
- Probability density estimation.
- Learning submanifolds.
- Inverting mappings.
- Least squares inverse of a map.
- Nearest point in a parameterized family.
- Partial match queries.
- Discovering product structure.
- Constraint networks.
- Baysian networks.