Day 08

ICP

ICP Algorithm

1. Find an initial guess for the rotation R and translation d.
2. Apply R and d to P to obtain $Q_0 = \{q_1, ..., q_n\}$
3. For k = 1, 2, ...
   1. Find the points on the surface $Y_k = \{y_1, ..., y_n\}$ closest to Q_{k-1}.
   2. Use Horn's method to match P to Y_k; this yields the new estimate for R and d.
   3. Apply R and d to P to obtain $Q_k = \{q_1, ..., q_n\}$
   4. Compute the mean squared error

$$\delta_k = \frac{1}{n}\sum_{i=1}^{n}\left\|y_i - q_i\right\|^2$$

Until $\delta_{k+1} - \delta_k < \tau$ for some threshold value τ

Computational Complexity

- for each iteration of ICP
  - compute set of closest points $O(n_p n_x)$
  - compute registration using Horn's method $O(n_p)$
  - apply registration $O(n_p)$

- overall $O(n_p n_x)$

Accelerating Closest Point Search

- common method for accelerating closest point search is to use a kd-tree
  - k is the dimension of the data
  - Jon Bentley. Communications of the ACM, 18(9), 1975.

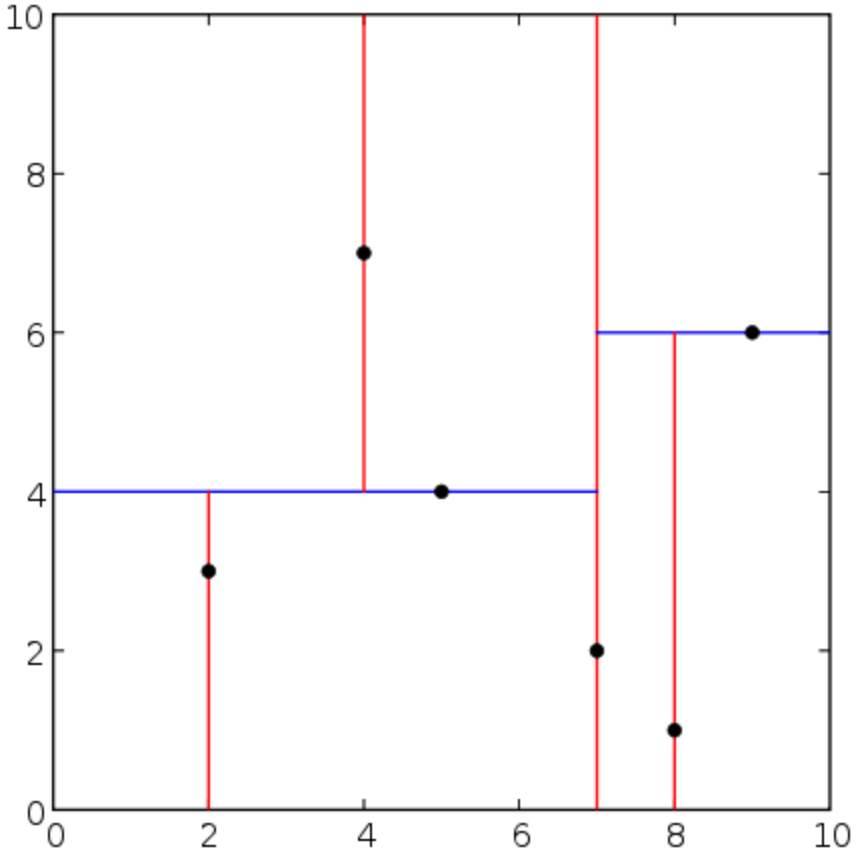
- supports $O(\log n)$ nearest neighbor queries if data is uniformly distributed

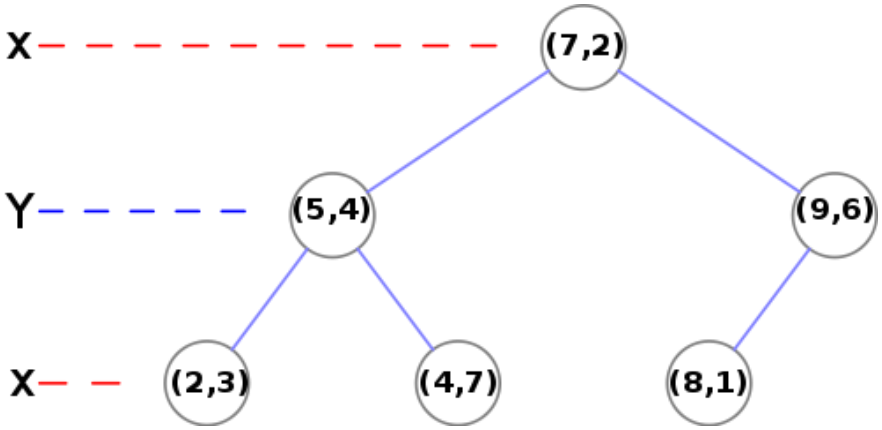
kd-tree

- binary tree where every non-leaf node can be thought of as being on an axis aligned hyperplane that splits the hypervolume of space into two parts (called subspaces or cells)
  - points on the "left" side of the hyperplane are held in the left subtree of the node, and points on the "right" side of the hyperplane are held in the right subtree of the node
- requires a splitting rule to determine where to put the hyperplane

kd-tree Example

- http://en.wikipedia.org/wiki/Kd-tree



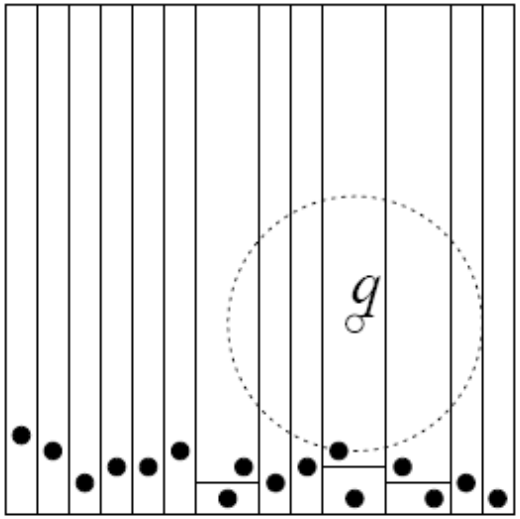


Splitting Rules

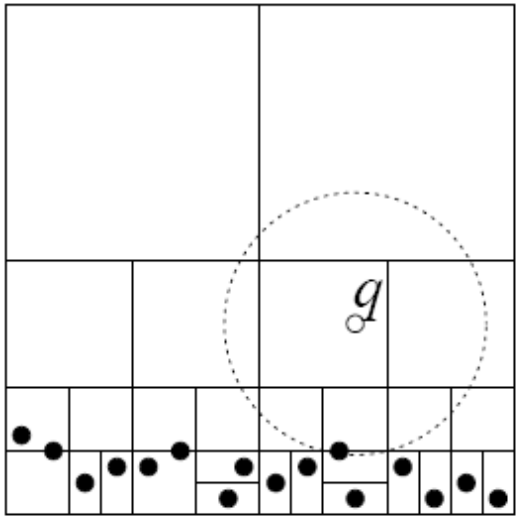
- standard split
  - splitting dimension is chosen to be the one for which the data points have the maximum spread
  - splitting value is the coordinate median
- midpoint split
  - splitting hyperplane passes through the center of the cell and bisecting the longest side of the cell
- sliding midpoint split
  - midpoint split attempted first; if there are no points on one side of the hyperplane then the hyperplane slides towards the data points until it reaches a data point

Splitting Rules

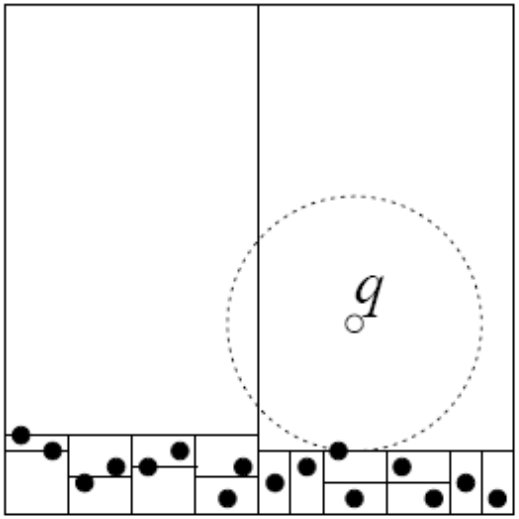
- S. Maneewongvatana and D. M. Mount,
  4th Annual CGC Workshop on Comptutational Geometry, 1999.



Standard split



Midpoint split



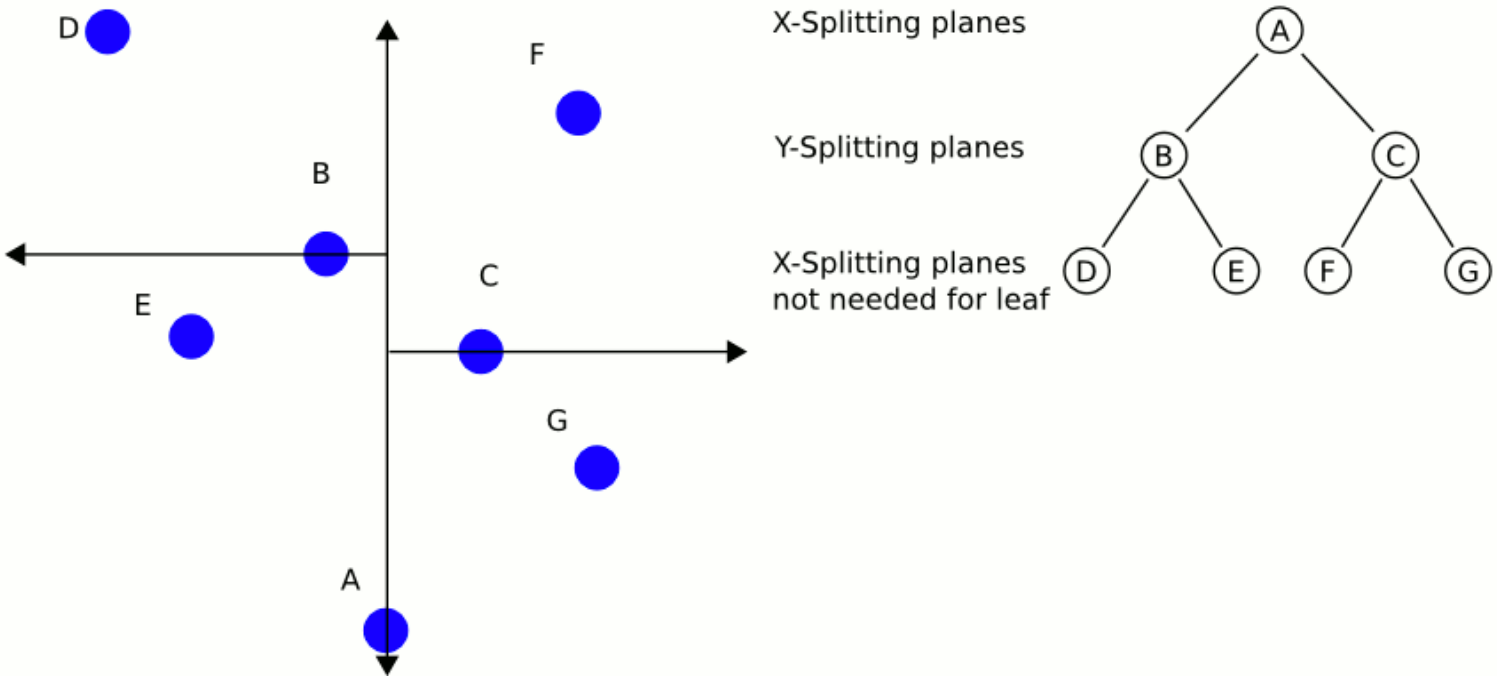
Sliding-midpoint split

Nearest Neighbor Search

1. start at root node and descend recursively
2. when a leaf node is reached, save the node as the current best nearest neighbor (with distance r)
3. unwind the recursion, performing the following steps
    1. if the current node is closer than the current best then it becomes the current best (update r)
    2. check if points on the other side of the hyperplane might be closer than the current best
        1. if the hypersphere of radius r intersects the hyperplane then the other branch of the tree must be searched
        2. if there is no intersection then continue up the tree

Search Animation

- http://en.wikipedia.org/wiki/Kd-tree



Problems with Local Minima

- ICP is guaranteed to converge to a minimum in the mean-squared error
  - there may be many local minima
  - the global minimum may not be the true registration