**Background**

- Uncertain Location Information
  - Sensor environments: GPS consumes batteries
  - Mobile robots: Localization may not be accurate
  - Location privacy: Exact locations are hidden

- Location-based Queries
  - Range queries, nearest neighbor queries
  - Spatial index-based processing
  - What’s happen for uncertain locations?

**Objectives**

- Query Processing Based on Uncertain Location Information
  - Location of a query object is specified as a Gaussian distribution
  - Target data: spatial points

- Probabilistic Nearest Neighbor Query (PNNQ)
  - Find objects such that the probabilities that they are the nearest neighbors of \( q \) are greater than \( \theta \)

**Naïve Approach**

- \( \Pr_{NN}(q, o) \): Probability that target object \( o \) is the nearest neighbor of query object \( q \)
  - Can be calculated by integrating \( p_q(x) \) over Voronoi region \( V_o \)
    \[
    \Pr_{NN}(q, o) = \int_{V_o} p_q(x) \, dx
    \]
  - If the result is greater than \( \theta \), object \( o \) satisfies the condition

- Compute \( \Pr_{NN}(q, o) \) for each object \( o \) using numerical integration: quite costly!

**Our Approach**

- Use of Filtering
  - Prune non-candidate objects using low-cost filtering conditions
  - Only the remaining candidate objects require numerical integration
  - Filtering should be conservative: no false negatives

  We propose two filtering strategies

**Strategy 1: \( \theta \)-Region-Based Approach**

- \( \theta \)-Region: Ellipsoidal region for which the integration of \( p_q(x) \) becomes \( 1 - 2\theta \):

  \[
  \int_{(x-q)'\Sigma^{-1}(x-q) \leq r_{\theta}^2} p_q(x) \, dx = 1 - 2\theta
  \]

  Ellipsoidal region
  \[
  (x-q)'\Sigma^{-1}(x-q) \leq r_{\theta}^2
  \]
  is the \( \theta \)-region

  - \( \theta \)-region can be derived using \( r_{\theta}-table \) and transformation

**Strategy 2: Use of SES and \( p_q^T(x) \)**

- Compute the smallest enclosing sphere (SES) for each Voronoi region beforehand
  - Idea: Calculate integration

  \[
  \int_{\text{SES}_o} p_q(x) \, dx > \Pr_{NN}(q, o),
  \]
  which overestimates \( \Pr_{NN}(q, o) \)

  - Integration over a spherical region is more easier to compute

  Additional approximation: Use of upper bounding function \( p_q^T(x) \)
  - It gives the upper bound for \( p_q(x) \), and has a spherical isosurface
  - Easy to compute integration using a pre-computed table

  In summary, we perform two-step approximations

  \[
  \int_{\text{SES}_o} p_q^T(x) \, dx \geq \int_{\text{SES}_o} p_q(x) \, dx > \Pr_{NN}(q, o)
  \]

**Experimental Results**

- Performance of two strategies depends on parameters and given queries
  - No apparent winner
  - The hybrid strategy shows the best performance

- Query Example (see figure)
  - Enclosing box: bounding box for the \( \theta \)-region
  - Red cells: candidate cells
  - Green cells: answer cells

**Our Related Work**

  - Consider the case for range queries