Localization in Wireless Sensor Network

AICTE Sponsored Staff Development Program on
Wireless Networks & Emerging Research Areas
7 - 11 June 2010
organized by Computer Engineering & Information Technology Department,
U V Patel College of Engineering, Mehasana

M. A. Zaveri
Computer Engineering Department
Sardar Vallabhbhai National Institute of Technology, Surat
mazaveri@coed.svnit.ac.in

Localization in WSN

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$
  - S is the spatial location of the measurement
  - T the time of the measurement
  - M the measurement itself

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$
  - S is the spatial location of the measurement
  - T the time of the measurement
  - M the measurement itself
- Why is the location of node (or tuple information) in the network so important?

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$
  - S is the spatial location of the measurement
  - T the time of the measurement
  - M the measurement itself
- Why is the location of node (or tuple information) in the network so important?
- How can the spatial location of nodes be determined?

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$
  - S is the spatial location of the measurement
  - T the time of the measurement
  - M the measurement itself
- Why is the location of node (or tuple information) in the network so important?
- How can the spatial location of nodes be determined?
- Localization is quite a broad problem domain

Localization in WSN

- WSN provides information about the spatio-temporal characteristics of the observed physical world
- sensor observation as a tuple of the form $< S, T, M >$
  - S is the spatial location of the measurement
  - T the time of the measurement
  - M the measurement itself
- Why is the location of node (or tuple information) in the network so important?
- How can the spatial location of nodes be determined?
- Localization is quite a broad problem domain
- Location information of nodes in the network is fundamental for a number of reasons:

Application of Localization Information

- to provide location stamps for individual sensor measurements that are being gathered

Application of Localization Information

- to provide location stamps for individual sensor measurements that are being gathered
- to locate and track point objects in the environment

Application of Localization Information

- to provide location stamps for individual sensor measurements that are being gathered
- to locate and track point objects in the environment
- to monitor the spatial evolution of a diffuse phenomenon over time, such as an expanding chemical plume

Application of Localization Information

- to provide location stamps for individual sensor measurements that are being gathered
- to locate and track point objects in the environment
- to monitor the spatial evolution of a diffuse phenomenon over time, such as an expanding chemical plume
- for instance, this information is necessary for in-network processing algorithms that determine and track the changing boundaries of such a phenomenon

Application of Localization Information

- to provide location stamps for individual sensor measurements that are being gathered
- to locate and track point objects in the environment
- to monitor the spatial evolution of a diffuse phenomenon over time, such as an expanding chemical plume
- for instance, this information is necessary for in-network processing algorithms that determine and track the changing boundaries of such a phenomenon
- to achieve load balancing in topology control mechanisms; if nodes are densely deployed, geographic information of nodes can be used to selectively shut down some percentage of nodes in each geographic area to conserve energy, and rotate these over time to achieve load balancing

Application of Localization Information

- to form clusters; it can be used to define a partition of the network into separate clusters for hierarchical routing and collaborative processing

Application of Localization Information

- to form clusters; it can be used to define a partition of the network into separate clusters for hierarchical routing and collaborative processing
- to facilitate routing of information through the network; geographic routing algorithms that utilize location information instead of node addresses to provide efficient routing

Application of Localization Information

- to form clusters; it can be used to define a partition of the network into separate clusters for hierarchical routing and collaborative processing
- to facilitate routing of information through the network; geographic routing algorithms that utilize location information instead of node addresses to provide efficient routing
- to perform efficient spatial querying; a sink or gateway node can issue queries for information about specific locations or geographic regions; it can be used to scope the query propagation instead of flooding the whole network, which would be wasteful of energy

Application of Localization Information

- to form clusters; it can be used to define a partition of the network into separate clusters for hierarchical routing and collaborative processing
- to facilitate routing of information through the network; geographic routing algorithms that utilize location information instead of node addresses to provide efficient routing
- to perform efficient spatial querying; a sink or gateway node can issue queries for information about specific locations or geographic regions; it can be used to scope the query propagation instead of flooding the whole network, which would be wasteful of energy
- to determine the quality of coverage; if node locations are known, the network can keep track of the extent of spatial coverage provided by active sensors at any time

Is it so easy ??????

Is it so easy ??????

- Ad hoc sensor networks

Is it so easy ??????

- Ad hoc sensor networks
- various applications: necessary to accurately orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful
- determine physical coordinates; these coordinates can be global, or relative, they are an arbitrary rigid transformation away from the global coordinate system

Is it so easy ??????

- Ad hoc sensor networks
- various applications: necessary to accurately orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful
- determine physical coordinates; these coordinates can be global, or relative, they are an arbitrary rigid transformation away from the global coordinate system
- Anchor/Beacon nodes that know their global coordinates a priori, may be hard coded, or acquired through additional hardware like GPS receiver

Is it so easy ??????

- Ad hoc sensor networks
- various applications: necessary to accurately orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful
- determine physical coordinates; these coordinates can be global, or relative, they are an arbitrary rigid transformation away from the global coordinate system
- Anchor/Beacon nodes that know their global coordinates a priori, may be hard coded, or acquired through additional hardware like GPS receiver
- At a minimum, three non-collinear beacon nodes are required to define a global coordinate system in two dimensions

Is it so easy ??????

- Ad hoc sensor networks
- various applications: necessary to accurately orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful
- determine physical coordinates; these coordinates can be global, or relative, they are an arbitrary rigid transformation away from the global coordinate system
- Anchor/Beacon nodes that know their global coordinates a priori, may be hard coded, or acquired through additional hardware like GPS receiver
- At a minimum, three non-collinear beacon nodes are required to define a global coordinate system in two dimensions
- If three dimensional coordinates are required, then at least four non-coplanar beacons must be present

Typical Application and Issues

Typical Application and Issues

- Passive habitat monitoring:

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system
- passive source localization system - localizing sound, camera system

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system
- passive source localization system - localizing sound, camera system
- nodes with microphones distributed; estimate location of source using times of arrival of signal

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system
- passive source localization system - localizing sound, camera system
- nodes with microphones distributed; estimate location of source using times of arrival of signal
- structured - location information can be obtained using existing infrastructure GPS or Cellular phone positioning techniques

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system
- passive source localization system - localizing sound, camera system
- nodes with microphones distributed; estimate location of source using times of arrival of signal
- structured - location information can be obtained using existing infrastructure GPS or Cellular phone positioning techniques
- GPS receiver - expensive, works if good satellite coverage

Typical Application and Issues

- Passive habitat monitoring: tracking movements and population counts of animals
- currently employed techniques do not scale; no automated system
- passive source localization system - localizing sound, camera system
- nodes with microphones distributed; estimate location of source using times of arrival of signal
- structured - location information can be obtained using existing infrastructure GPS or Cellular phone positioning techniques
- GPS receiver - expensive, works if good satellite coverage
- later provides poor location accuracy - subset of nodes have know location a priori and positions of other nodes must be determined using some localization technique

Basic Issues in Localization

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile;

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment,

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment, on-the-fly or refresh location information as objects or nodes move around

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment, on-the-fly or refresh location information as objects or nodes move around
- how well to localize? resolution of location information desired

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment, on-the-fly or refresh location information as objects or nodes move around
- how well to localize? resolution of location information desired
- where to localize? at central location or distributed iterative manner within reference nodes or unknown nodes

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment, on-the-fly or refresh location information as objects or nodes move around
- how well to localize? resolution of location information desired
- where to localize? at central location or distributed iterative manner within reference nodes or unknown nodes
- how to localize? different signal measurements: RF signal strength, packet-loss statistics, acoustic/ultrasound signals, infrared

Basic Issues in Localization

- what to localize? reference node: having a priori known locations, unknown nodes
- these nodes could be static or mobile; unknown nodes may be cooperative or non-cooperative (can not participate actively)
- when to localize? one time process - static environment, on-the-fly or refresh location information as objects or nodes move around
- how well to localize? resolution of location information desired
- where to localize? at central location or distributed iterative manner within reference nodes or unknown nodes
- how to localize? different signal measurements: RF signal strength, packet-loss statistics, acoustic/ultrasound signals, infrared
- the signals may be emitted and measured by the reference nodes, by the unknown nodes, or both

Issues in Localization

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment
- cooperative vs. non-cooperative target - animal

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment
- cooperative vs. non-cooperative target - animal
- accuracy, active vs. passive infrastructure, dynamic vs. static

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment
- cooperative vs. non-cooperative target - animal
- accuracy, active vs. passive infrastructure, dynamic vs. static
- granularity and scale of measurement, cost involved

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment
- cooperative vs. non-cooperative target - animal
- accuracy, active vs. passive infrastructure, dynamic vs. static
- granularity and scale of measurement, cost involved
- coordinate system: global, absolute, relative

Issues in Localization

- no A-priori knowledge of sensor locations in large scale and ad-hoc deployments
- must be able to operate outdoors in various weather conditions
- power efficiency - battery lifetime
- outdoor vs. indoor environment, smart environment
- cooperative vs. non-cooperative target - animal
- accuracy, active vs. passive infrastructure, dynamic vs. static
- granularity and scale of measurement, cost involved
- coordinate system: global, absolute, relative
- Form factor: size of node, multiple sensors, communications requirements, time synchronization

Issues in Localization Algorithm Design

Issues in Localization Algorithm Design

- Resource constraints

Issues in Localization Algorithm Design

- Resource constraints
- Node density

Issues in Localization Algorithm Design

- Resource constraints
- Node density
- Environmental obstacles and terrain irregularities

Issues in Localization Algorithm Design

- Resource constraints
- Node density
- Environmental obstacles and terrain irregularities
- Non-convex topologies

Issues in Localization Algorithm Design

- Resource constraints
- Node density
- Environmental obstacles and terrain irregularities
- Non-convex topologies
  - having trouble positioning nodes near the edges
  - fewer range measurements for border nodes
  - sensors outside the main convex body of the network can often prove unlocalizable

Categories of Localization System

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets
- active non-cooperative and cooperative target

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets
- active non-cooperative and cooperative target
- cooperative infrastructure

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets
- active non-cooperative and cooperative target
- cooperative infrastructure
- passive localization: discover ranges and locations by passively monitoring existing signals in a particular channel (ToF, TDoA)

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets
- active non-cooperative and cooperative target
- cooperative infrastructure
- passive localization: discover ranges and locations by passively monitoring existing signals in a particular channel (ToF, TDoA)
- blind Source Localization: a signal source is localized without any a priori knowledge of the type of signal emitted; done by blind beam-forming which effectively cross-correlates the signals from different receivers

Categories of Localization System

- active localization: emit signals into the environment that are used to measure range to the target; signals emitted by infrastructure components or by targets
- active non-cooperative and cooperative target
- cooperative infrastructure
- passive localization: discover ranges and locations by passively monitoring existing signals in a particular channel (ToF, TDoA)
- blind Source Localization: a signal source is localized without any a priori knowledge of the type of signal emitted; done by blind beam-forming which effectively cross-correlates the signals from different receivers
- these techniques generally only work so long as the signals being compared are coherent

Broad Category of Localization approaches

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling
- Coarse-grained localization: minimal information used to compute location

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling
- Coarse-grained localization: minimal information used to compute location
  - discrete measurements
  - binary proximity (nodes near to each other)
  - cardinal direction (node in the north, east etc.)

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling
- Coarse-grained localization: minimal information used to compute location
  - discrete measurements
  - binary proximity (nodes near to each other)
  - cardinal direction (node in the north, east etc.)
- Fine-grained localization: detailed information

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling
- Coarse-grained localization: minimal information used to compute location
  - discrete measurements
  - binary proximity (nodes near to each other)
  - cardinal direction (node in the north, east etc.)
- Fine-grained localization: detailed information
  - RF power, signal waveform, time stamps etc.
- tradeoff

Broad Category of Localization approaches

- the basic localization algorithm based on techniques: proximity, calculation of centroids, constraints, ranging, angulation, pattern recognition, multi-dimensional scaling
- Coarse-grained localization: minimal information used to compute location
  - discrete measurements
  - binary proximity (nodes near to each other)
  - cardinal direction (node in the north, east etc.)
- Fine-grained localization: detailed information
  - RF power, signal waveform, time stamps etc.
- tradeoff
  - lower resource consumption and equipment cost
  - provide lower accuracy than the detailed information techniques

Coarse-grained localization

Coarse-grained localization

- set of reference nodes, emit beacon include location IDs and unknown node determines which node it is closet to;

Coarse-grained localization

- set of reference nodes, emit beacon include location IDs and unknown node determines which node it is closet to;
- unknown node emits a beacon then reference node that hears the beacon uses its own location to determine the location of the unknown node

Coarse-grained localization

- set of reference nodes, emit beacon include location IDs and unknown node determines which node it is closet to;
- unknown node emits a beacon then reference node that hears the beacon uses its own location to determine the location of the unknown node
- proximity detection e.g. active badge in an indoor office environment, or passive RF identification detected by reader within a short range

Coarse-grained localization

- set of reference nodes, emit beacon include location IDs and unknown node determines which node it is closet to;
- unknown node emits a beacon then reference node that hears the beacon uses its own location to determine the location of the unknown node
- proximity detection e.g. active badge in an indoor office environment, or passive RF identification detected by reader within a short range
- inventory tracking

Coarse-grained localization

- set of reference nodes, emit beacon include location IDs and unknown node determines which node it is closet to;
- unknown node emits a beacon then reference node that hears the beacon uses its own location to determine the location of the unknown node
- proximity detection e.g. active badge in an indoor office environment, or passive RF identification detected by reader within a short range
- inventory tracking
- key difference in RFID proximity detection compared with active badges is that the unknown nodes are passive tags, being queried by the reference nodes

Coarse-grained localization

- Centroid calculation: used when density of reference nodes is high within the range of the unknown node

Coarse-grained localization

- Centroid calculation: used when density of reference nodes is high within the range of the unknown node
- Let there be n reference nodes detected within the proximity of the unknown node

Coarse-grained localization

- Centroid calculation: used when density of reference nodes is high within the range of the unknown node
- Let there be n reference nodes detected within the proximity of the unknown node
- the location of the ith such reference denoted by (x_i, y_i)

Coarse-grained localization

- Centroid calculation: used when density of reference nodes is high within the range of the unknown node
- Let there be n reference nodes detected within the proximity of the unknown node
- the location of the ith such reference denoted by (x_i, y_i)
- the location of the unknown node (x_u, y_u) is determined as

$$x_u = \frac{1}{n}\sum_{i=1}^{n} x_i \quad y_u = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Coarse-grained localization

- Centroid calculation: used when density of reference nodes is high within the range of the unknown node
- Let there be n reference nodes detected within the proximity of the unknown node
- the location of the ith such reference denoted by (x_i, y_i)
- the location of the unknown node (x_u, y_u) is determined as

$$x_u = \frac{1}{n}\sum_{i=1}^{n} x_i \quad y_u = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- each node having a simple circular range R in an infinite square mesh of reference nodes spaced a distance d apart
- shown through simulations that, as the overlap ratio R/d is increased from 1 to 4, the average RMS error in localization is reduced from $0.5d$ to $0.25d$

Coarse-grained localization

- Geometric constraints
- radio signal coverage described by a geometric shape

Coarse-grained localization

- Geometric constraints
- radio signal coverage described by a geometric shape
- location estimates by determining which geometric regions that node is constrained to be in, because of intersections between overlapping coverage regions

Coarse-grained localization

- Geometric constraints
- radio signal coverage described by a geometric shape
- location estimates by determining which geometric regions that node is constrained to be in, because of intersections between overlapping coverage regions
- region of radio coverage using a circle of radius $R_{\max}$
- if an unknown node hears from several reference nodes, it can determine that it must lie in the geometric region described by the intersection of circles of radius $R_{\max}$ centered on these nodes
- rectangular bounding boxes: node determines bounds $x_{\min}, y_{\min}, x_{\max}, y_{\max}$ on its position

Coarse-grained localization

- Geometric constraints
- radio signal coverage described by a geometric shape
- location estimates by determining which geometric regions that node is constrained to be in, because of intersections between overlapping coverage regions
- region of radio coverage using a circle of radius R_{max}
- if an unknown node hears from several reference nodes, it can determine that it must lie in the geometric region described by the intersection of circles of radius R_{max} centered on these nodes
- rectangular bounding boxes: node determines bounds $x_{min}, y_{min}, x_{max}, y_{max}$ on its position
- the unknown nodes use the centroid of the overlapping region as a specific location estimate if necessary, determine a bound on the location error using the size of this region

Coarse-grained localization

- Geometric constraints
- radio signal coverage described by a geometric shape
- location estimates by determining which geometric regions that node is constrained to be in, because of intersections between overlapping coverage regions
- region of radio coverage using a circle of radius $R_{\max}$
- if an unknown node hears from several reference nodes, it can determine that it must lie in the geometric region described by the intersection of circles of radius $R_{\max}$ centered on these nodes
- rectangular bounding boxes: node determines bounds $x_{\min}, y_{\min}, x_{\max}, y_{\max}$ on its position
- the unknown nodes use the centroid of the overlapping region as a specific location estimate if necessary, determine a bound on the location error using the size of this region
- arbitrary shapes can be potentially computed in this manner

Coarse-grained localization

- Approximate point in triangle (APIT)

Coarse-grained localization

- Approximate point in triangle (APIT)
- localization using geometric constraints is the APIT technique
- provides location estimates as the centroid of an intersection of regions

Coarse-grained localization

- Approximate point in triangle (APIT)
- localization using geometric constraints is the APIT technique
- provides location estimates as the centroid of an intersection of regions
- novelty lies in how the regions are defined as triangles between different sets of three reference nodes (rather than the coverage of a single node)

Coarse-grained localization

- Approximate point in triangle (APIT)
- localization using geometric constraints is the APIT technique
- provides location estimates as the centroid of an intersection of regions
- novelty lies in how the regions are defined as triangles between different sets of three reference nodes (rather than the coverage of a single node)
- Identifying codes

Coarse-grained localization

- Approximate point in triangle (APIT)
- localization using geometric constraints is the APIT technique
- provides location estimates as the centroid of an intersection of regions
- novelty lies in how the regions are defined as triangles between different sets of three reference nodes (rather than the coverage of a single node)
- Identifying codes
- utilizes overlapping coverage regions to provide localization
- referred to as the identifying code construction (ID-CODE) algorithm
- the sensor deployment is planned in such a way as to ensure that each resolvable location is covered by a unique set of sensors

Coarse-grained localization

- algorithm runs on a deployment region graph $G = (V, E)$ vertices V represent the different regions, and the edges E represent radio connectivity between regions
- goal is to construct an identifying code for any distinguishable graph, with each vertex in the code corresponding to a region where a reference node must be placed
- Once this is done, by the definition of the identifying code, each location region in the graph will be covered by a unique set of reference nodes
- obtaining a minimal cardinality identifying code is known to be NP-complete

Fine-grained localization

Fine-grained localization

- basic time-of-flight techniques using RF signals are not capable of providing precise distance estimates over short ranges typical of WSN because of synchronization limitations

Fine-grained localization

- basic time-of-flight techniques using RF signals are not capable of providing precise distance estimates over short ranges typical of WSN because of synchronization limitations
- other techniques such as radio signal strength (RSS) measurements, time difference of arrival (TDoA) and Angle of Arrival (AoA) are used for distance-estimation

Fine-grained localization

- basic time-of-flight techniques using RF signals are not capable of providing precise distance estimates over short ranges typical of WSN because of synchronization limitations
- other techniques such as radio signal strength (RSS) measurements, time difference of arrival (TDoA) and Angle of Arrival (AoA) are used for distance-estimation
- triangulation using distance estimates, pattern matching, and sequence decoding used in the large-scale GPS

Received Signal Strength Indication (RSSI)

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network
- two important techniques for using radio information to compute ranges
- hop count and Received Signal Strength Indication (RSSI)

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network
- two important techniques for using radio information to compute ranges
- hop count and Received Signal Strength Indication (RSSI)
- the energy of a radio signal diminishes with the square of the distance from the signal's source

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network
- two important techniques for using radio information to compute ranges
- hop count and Received Signal Strength Indication (RSSI)
- the energy of a radio signal diminishes with the square of the distance from the signal's source
- a node listening to a radio transmission should be able to use the strength of the received signal to calculate its distance from the transmitter

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network
- two important techniques for using radio information to compute ranges
- hop count and Received Signal Strength Indication (RSSI)
- the energy of a radio signal diminishes with the square of the distance from the signal's source
- a node listening to a radio transmission should be able to use the strength of the received signal to calculate its distance from the transmitter
- RSSI ranging measurements contain noise on the order of several meters

Received Signal Strength Indication (RSSI)

- hardware methods of computing distance measurements between nearby sensor nodes (i.e. ranging)
- every sensor has a radio and it helps localize the network
- two important techniques for using radio information to compute ranges
- hop count and Received Signal Strength Indication (RSSI)
- the energy of a radio signal diminishes with the square of the distance from the signal's source
- a node listening to a radio transmission should be able to use the strength of the received signal to calculate its distance from the transmitter
- RSSI ranging measurements contain noise on the order of several meters
- this noise occurs because radio propagation tends to be highly non-uniform in real environments

Received Signal Strength Indication (RSSI)

- radio propagates differently over asphalt than over grass, physical obstacles such as walls, furniture, etc. reflect and absorb radio waves

Received Signal Strength Indication (RSSI)

- radio propagates differently over asphalt than over grass, physical obstacles such as walls, furniture, etc. reflect and absorb radio waves
- first-order approximation, mean radio signal strengths diminish with distance according to a power law

Received Signal Strength Indication (RSSI)

- radio propagates differently over asphalt than over grass, physical obstacles such as walls, furniture, etc. reflect and absorb radio waves
- first-order approximation, mean radio signal strengths diminish with distance according to a power law
- model used for wireless radio propagation

$$P_{\mathrm{r,dB}}(d) = P_{\mathrm{r,dB}}(d_0) - \eta 10 \log\left(\frac{d}{d_0}\right) + X_{\sigma,\mathrm{dB}}$$

Received Signal Strength Indication (RSSI)

- radio propagates differently over asphalt than over grass, physical obstacles such as walls, furniture, etc. reflect and absorb radio waves
- first-order approximation, mean radio signal strengths diminish with distance according to a power law
- model used for wireless radio propagation

$$P_{\mathrm{r,dB}}(d) = P_{\mathrm{r,dB}}(d_0) - \eta 10 \log\left(\frac{d}{d_0}\right) + X_{\sigma,\mathrm{dB}}$$

- $P_{\mathrm{r,dB}}(d)$ is the received power at distance d
- $P(d_0)$ is the received power at some reference distance d_0
- η the path-loss exponent
- $X_{\sigma,\mathrm{dB}}$ a log-normal random variable with variance σ^2 that accounts for fading effects

Radio signal-based distance-estimation (RSS)

- if the path-loss exponent for a given environment is known the received signal strength can be used to estimate the distance
- the fading term often has a large variance, which can significantly impact the quality of the range estimates

Radio signal-based distance-estimation (RSS)

- if the path-loss exponent for a given environment is known the received signal strength can be used to estimate the distance
- the fading term often has a large variance, which can significantly impact the quality of the range estimates
- this is the reason RF-RSS-based ranging techniques may offer location accuracy only on the order of meters or more

Radio signal-based distance-estimation (RSS)

- if the path-loss exponent for a given environment is known the received signal strength can be used to estimate the distance
- the fading term often has a large variance, which can significantly impact the quality of the range estimates
- this is the reason RF-RSS-based ranging techniques may offer location accuracy only on the order of meters or more
- RSS-based ranging may perform much better in situations where the fading effects can be combated by diversity techniques that take advantage of separate spatio-temporally correlated signal samples

Radio Hop Count

Radio Hop Count

- even though RSSI is too inaccurate for many applications, the radio can still be used to assist localization

Radio Hop Count

- even though RSSI is too inaccurate for many applications, the radio can still be used to assist localization
- hop count to be a useful way to compute inter-node distances

Radio Hop Count

- even though RSSI is too inaccurate for many applications, the radio can still be used to assist localization
- hop count to be a useful way to compute inter-node distances
- the local connectivity information provided by the radio defines an unweighted graph, where the vertices are sensor nodes, and edges represent direct radio links between nodes

Radio Hop Count

- even though RSSI is too inaccurate for many applications, the radio can still be used to assist localization
- hop count to be a useful way to compute inter-node distances
- the local connectivity information provided by the radio defines an unweighted graph, where the vertices are sensor nodes, and edges represent direct radio links between nodes
- the hop count h_{ij} between sensor nodes s_i and s_j is then defined as the length of the shortest path in the graph between s_i and s_j

Radio Hop Count

- even though RSSI is too inaccurate for many applications, the radio can still be used to assist localization
- hop count to be a useful way to compute inter-node distances
- the local connectivity information provided by the radio defines an unweighted graph, where the vertices are sensor nodes, and edges represent direct radio links between nodes
- the hop count h_{ij} between sensor nodes s_i and s_j is then defined as the length of the shortest path in the graph between s_i and s_j
- if the hop count between s_i and s_j is h_{ij} then the distance between s_i and s_j, d_{ij}, is less than $R * h_{ij}$, where R is again the maximum radio range

Time Difference of Arrival (TDoA)

Time Difference of Arrival (TDoA)

- time-of-flight techniques show poor performance due to precision constraints
- RSS techniques, although somewhat better, are still limited by fading effects

Time Difference of Arrival (TDoA)

- time-of-flight techniques show poor performance due to precision constraints
- RSS techniques, although somewhat better, are still limited by fading effects
- A more promising technique is the combined use of ultrasound/ acoustic and radio signals to estimate distances by determining the TDoA of these signals

Time Difference of Arrival (TDoA)

- time-of-flight techniques show poor performance due to precision constraints
- RSS techniques, although somewhat better, are still limited by fading effects
- A more promising technique is the combined use of ultrasound/acoustic and radio signals to estimate distances by determining the TDoA of these signals
- in this technique: simultaneously transmit both the radio and acoustic signals (audible or ultrasound) and measure the times T_r and T_s of the arrival of these signals respectively at the receiver

Time Difference of Arrival (TDoA)

- time-of-flight techniques show poor performance due to precision constraints
- RSS techniques, although somewhat better, are still limited by fading effects
- A more promising technique is the combined use of ultrasound/ acoustic and radio signals to estimate distances by determining the TDoA of these signals
- in this technique: simultaneously transmit both the radio and acoustic signals (audible or ultrasound) and measure the times T_r and T_s of the arrival of these signals respectively at the receiver
- the speed of the radio signal is much larger than the speed of the acoustic signal, the distance is then simply estimated as $(T_s - T_r)$ V_s, where V_s is the speed of the acoustic signal

Time Difference of Arrival (TDoA): Issues

- limitation of acoustic ranging is that it generally requires the nodes to be in fairly close proximity to each other (within a few meters) and preferably in line of sight

Time Difference of Arrival (TDoA): Issues

- limitation of acoustic ranging is that it generally requires the nodes to be in fairly close proximity to each other (within a few meters) and preferably in line of sight
- there is also some uncertainty in the calculation because the speed of sound varies depending on many factors such as altitude, humidity, and air temperature

Time Difference of Arrival (TDoA): Issues

- limitation of acoustic ranging is that it generally requires the nodes to be in fairly close proximity to each other (within a few meters) and preferably in line of sight
- there is also some uncertainty in the calculation because the speed of sound varies depending on many factors such as altitude, humidity, and air temperature
- acoustic signals also show multi-path propagation effects that may impact the accuracy of signal detection

Time Difference of Arrival (TDoA): Issues

- limitation of acoustic ranging is that it generally requires the nodes to be in fairly close proximity to each other (within a few meters) and preferably in line of sight
- there is also some uncertainty in the calculation because the speed of sound varies depending on many factors such as altitude, humidity, and air temperature
- acoustic signals also show multi-path propagation effects that may impact the accuracy of signal detection
- the basic idea is to send a pseudo-random noise sequence as the acoustic signal and use a matched filter for detection
- acoustic TDoA ranging techniques can be very accurate in practical settings

Distance-estimation using TDoA

- each node is equipped with a speaker and a microphone
- some systems use ultrasound while others use audible frequencies
- the transmitter first sends a radio message, waits some fixed interval of time, t_{delay} (which might be zero), and then produces a fixed pattern of chirps on its speaker

Distance-estimation using TDoA

- each node is equipped with a speaker and a microphone
- some systems use ultrasound while others use audible frequencies
- the transmitter first sends a radio message, waits some fixed interval of time, t_{delay} (which might be zero), and then produces a fixed pattern of chirps on its speaker
- listening nodes hear the radio signal, they note the current time, t_{radio}, then turn on their microphones

Distance-estimation using TDoA

- each node is equipped with a speaker and a microphone
- some systems use ultrasound while others use audible frequencies
- the transmitter first sends a radio message, waits some fixed interval of time, t_{delay} (which might be zero), and then produces a fixed pattern of chirps on its speaker
- listening nodes hear the radio signal, they note the current time, t_{radio}, then turn on their microphones
- when their microphones detect the chirp pattern, they again note the current time, t_{sound}

Distance-estimation using TDoA

- each node is equipped with a speaker and a microphone
- some systems use ultrasound while others use audible frequencies
- the transmitter first sends a radio message, waits some fixed interval of time, t_{delay} (which might be zero), and then produces a fixed pattern of chirps on its speaker
- listening nodes hear the radio signal, they note the current time, t_{radio}, then turn on their microphones
- when their microphones detect the chirp pattern, they again note the current time, t_{sound}
- once they have t_{radio}, t_{sound}, and t_{delay}, the listeners can compute the distance d between themselves and the transmitter using the fact that radio waves travel substantially faster than sound in air

$$d = (s_{radio} - s_{sound}) * (t_{sound} - t_{radio} - t_{delay})$$

Triangulation using distance estimates

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$
- this can be formulated as a least squares minimization problem

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$
- this can be formulated as a least squares minimization problem
- let d_i be the correct Euclidean distance to the n reference nodes,

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$
- this can be formulated as a least squares minimization problem
- let d_i be the correct Euclidean distance to the n reference nodes,

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

- the difference between the measured and actual distances can be represented as

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$
- this can be formulated as a least squares minimization problem
- let d_i be the correct Euclidean distance to the n reference nodes,

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

- the difference between the measured and actual distances can be represented as

$$\rho_i = \hat{d}_i - d_i$$

- the least squares minimization problem is then to determine the (x_0, y_0) that minimizes $\sum_{i=1}^{n} (\rho_i)^2$

Triangulation using distance estimates

- the location of the unknown node (x_0, y_0) can be determined based on measured distance estimates $\hat{d}_i$ to n reference nodes $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$
- this can be formulated as a least squares minimization problem
- let d_i be the correct Euclidean distance to the n reference nodes,

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

- the difference between the measured and actual distances can be represented as

$$\rho_i = \hat{d}_i - d_i$$

- the least squares minimization problem is then to determine the (x_0, y_0) that minimizes $\sum_{i=1}^{n} (\rho_i)^2$
- it can be solved by the use of gradient descent techniques or by iterative successive approximation techniques
- alternative approach provides a numerical solution to an over-determined $(n \geq 3)$ linear system

Triangulation using distance estimates

- the over-determined linear system can be obtained as follows: rearranging and squaring terms -

$$x_i^2 + y_i^2 - d_i^2 = 2x_0x_i + 2y_0y_i - (x_0^2 + y_0^2)$$

Triangulation using distance estimates

- the over-determined linear system can be obtained as follows: rearranging and squaring terms -

$$x_i^2 + y_i^2 - d_i^2 = 2x_0x_i + 2y_0y_i - (x_0^2 + y_0^2)$$

by subtracting out the nth equation from the rest, $n-1$ equations of the following form:

$$x_i^2 + y_i^2 - x_n^2 - y_n^2 - d_i^2 + d_n^2 = x_0 2(x_i - x_n) + y_0 2(y_i - y_n)$$

Triangulation using distance estimates

- the over-determined linear system can be obtained as follows: rearranging and squaring terms -

$$x_i^2 + y_i^2 - d_i^2 = 2x_0x_i + 2y_0y_i - (x_0^2 + y_0^2)$$

by subtracting out the nth equation from the rest, $n-1$ equations of the following form:

$$x_i^2 + y_i^2 - x_n^2 - y_n^2 - d_i^2 + d_n^2 = x_0 2(x_i - x_n) + y_0 2(y_i - y_n)$$

which yields the linear relationship

$$\mathbf{A}\bar{x} = \mathbf{B}$$

- $\mathbf{A}$ is an $(n-1) \times 2$ matrix, such that ith row of $\mathbf{A}$ is $[2(X_i - x_n), 2(y_i - y_n)]$, $\bar{x}$ is the column vector representing the coordinates of the unknown location $[x_0, y_o]^T$

Triangulation using distance estimates

- the over-determined linear system can be obtained as follows: rearranging and squaring terms -

$$x_i^2 + y_i^2 - d_i^2 = 2x_0x_i + 2y_0y_i - (x_0^2 + y_0^2)$$

by subtracting out the nth equation from the rest, $n-1$ equations of the following form:

$$x_i^2 + y_i^2 - x_n^2 - y_n^2 - d_i^2 + d_n^2 = x_0 2(x_i - x_n) + y_0 2(y_i - y_n)$$

which yields the linear relationship

$$\mathbf{A}\bar{x} = \mathbf{B}$$

- $\mathbf{A}$ is an $(n-1) \times 2$ matrix, such that ith row of $\mathbf{A}$ is $[2(X_i - x_n), 2(y_i - y_n)]$, $\bar{x}$ is the column vector representing the coordinates of the unknown location $[x_0, y_o]^T$
- $\mathbf{B}$ is the $(n-1)$ element column vector whose ith term is the expression $x_i^2 + y_i^2 - x_n^2 - y_n^2 - d_i^2 + d_n^2$

Triangulation using distance estimates

- in practice **B** can not be determined, there is an access to the estimated distance only

Triangulation using distance estimates

- in practice $\mathbf{B}$ can not be determined, there is an access to the estimated distance only
- so calculate the elements of the related vector $\hat{\mathbf{B}}$, which is the same as $\mathbf{B}$ with $\hat{d}_i$ substituted for d_i
- least squares solution is to determine an estimate for $\bar{x}$ that minimizes $||\mathbf{A}\bar{x} - \hat{\mathbf{B}}||_2$

Triangulation using distance estimates

- in practice $\mathbf{B}$ can not be determined, there is an access to the estimated distance only
- so calculate the elements of the related vector $\hat{\mathbf{B}}$, which is the same as $\mathbf{B}$ with $\hat{d}_i$ substituted for d_i
- least squares solution is to determine an estimate for $\bar{x}$ that minimizes $||\mathbf{A}\bar{x} - \hat{\mathbf{B}}||_2$
- an estimate is provided by

$$\bar{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{B}}$$

Triangulation using distance estimates

- in practice $\mathbf{B}$ can not be determined, there is an access to the estimated distance only
- so calculate the elements of the related vector $\hat{\mathbf{B}}$, which is the same as $\mathbf{B}$ with $\hat{d}_i$ substituted for d_i
- least squares solution is to determine an estimate for $\bar{x}$ that minimizes $||\mathbf{A}\bar{x} - \hat{\mathbf{B}}||_2$
- an estimate is provided by

$$\bar{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{B}}$$

- solving for the above may not directly yield a numerical solution if the matrix $\mathbf{A}$ is ill-conditioned
- a recommended approach is to instead use the pseudo-inverse $\mathbf{A}^+$ of the matrix $\mathbf{A}$

$$\bar{x} = \mathbf{A}^+\hat{\mathbf{B}}$$

Angle of arrival (AoA)

Angle of arrival (AoA)

- use of angular estimates instead of distance estimates

Angle of arrival (AoA)

- use of angular estimates instead of distance estimates
- angles can potentially be estimated by using rotating directional beacons, or by using nodes equipped with a phased array of RF or ultrasonic receivers

Angle of arrival (AoA)

- use of angular estimates instead of distance estimates
- angles can potentially be estimated by using rotating directional beacons, or by using nodes equipped with a phased array of RF or ultrasonic receivers
- involving three rotating reference beacons at the boundary of a sensor network providing localization for all interior nodes
- if the angular information provided to a given reference node can be combined with a good distance estimate to that reference node, then localization can be performed with a single reference using polar coordinate transformation

General Framework for Localization

General Framework for Localization

- in all the localization approaches the network is modeled by a graph

General Framework for Localization

- in all the localization approaches the network is modeled by a graph
- $G = (V, E)$ V is the set of nodes some with known positions in the Euclidean space R^{dim} and E is the set of edges defined by the network topology (connectivity)

General Framework for Localization

- in all the localization approaches the network is modeled by a graph
- $G = (V, E)$ V is the set of nodes some with known positions in the Euclidean space R^{dim} and E is the set of edges defined by the network topology (connectivity)
- in one case a set of weights $\{d_{ij} : (i, j) \in E\}$ on the graph's edges is given, representing the (estimated) distances between the corresponding nodes

General Framework for Localization

- in all the localization approaches the network is modeled by a graph
- $G = (V, E)$ V is the set of nodes some with known positions in the Euclidean space R^{dim} and E is the set of edges defined by the network topology (connectivity)
- in one case a set of weights $\{d_{ij} : (i, j) \in E\}$ on the graph's edges is given, representing the (estimated) distances between the corresponding nodes
- the problem is then to place all nodes in such a way that the Euclidean distance between every pair of nodes v and w $(v, w) \in E$ equals d_{vw}

General Framework for Localization

- in all the localization approaches the network is modeled by a graph
- $G = (V, E)$ V is the set of nodes some with known positions in the Euclidean space R^{dim} and E is the set of edges defined by the network topology (connectivity)
- in one case a set of weights $\{d_{ij} : (i, j) \in E\}$ on the graph's edges is given, representing the (estimated) distances between the corresponding nodes
- the problem is then to place all nodes in such a way that the Euclidean distance between every pair of nodes v and w $(v, w) \in E$ equals d_{vw}
- in other case, d_{ij} are not given a special value; it is only assumed that $d_{ij} < R$ R is the range of the transmitter of a wireless sensor node

General Framework for Localization

- in all the localization approaches the network is modeled by a graph
- $G = (V, E)$ V is the set of nodes some with known positions in the Euclidean space R^{dim} and E is the set of edges defined by the network topology (connectivity)
- in one case a set of weights $\{d_{ij} : (i, j) \in E\}$ on the graph's edges is given, representing the (estimated) distances between the corresponding nodes
- the problem is then to place all nodes in such a way that the Euclidean distance between every pair of nodes v and w $(v, w) \in E$ equals d_{vw}
- in other case, d_{ij} are not given a special value; it is only assumed that $d_{ij} < R$ R is the range of the transmitter of a wireless sensor node
- to describe the positions of the nodes of the network, form a corresponding matrix and store available distance information in matrix $D = \{d_{ij} i, j = 1, 2, \ldots, n\}$

Category based on Computational Organization

Category based on Computational Organization

- Centralized algorithms:

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)
- all data is used to solve the problem at hand

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)
- all data is used to solve the problem at hand
- Distributed algorithms:

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)
- all data is used to solve the problem at hand
- Distributed algorithms: run in the network, using massive parallelism and inter-node communication, subset of the data used

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)
- all data is used to solve the problem at hand
- Distributed algorithms: run in the network, using massive parallelism and inter-node communication, subset of the data used
- two types of approaches to distributed localization

Category based on Computational Organization

- Centralized algorithms: sensor nodes gather data and pass it to base station (computationally powerful)
- communication cost of moving data to base station and computed information back into the network
- scalable by deploying multiple base stations (forming a multi-tier network)
- all data is used to solve the problem at hand
- Distributed algorithms: run in the network, using massive parallelism and inter-node communication, subset of the data used
- two types of approaches to distributed localization
- first, beacon-based distributed algorithm: starts with some beacon nodes, few nodes compute their own location using distance measurements and become beacons to help other nodes localize

Category based on Computational Organization

- second, optimize a global metric over the network

Category based on Computational Organization

- second, optimize a global metric over the network
- relaxation based distributed algorithm is to use a coarse algorithm to roughly localize nodes and followed by refinement step involving node position adjustment to optimize a local error metric

Category based on Computational Organization

- second, optimize a global metric over the network
- relaxation based distributed algorithm is to use a coarse algorithm to roughly localize nodes and followed by refinement step involving node position adjustment to optimize a local error metric
- algorithms hope to approximate the optimal solution to a network-wide metric that is the sum of the local error metric at each of the nodes

Category based on Computational Organization

- second, optimize a global metric over the network
- relaxation based distributed algorithm is to use a coarse algorithm to roughly localize nodes and followed by refinement step involving node position adjustment to optimize a local error metric
- algorithms hope to approximate the optimal solution to a network-wide metric that is the sum of the local error metric at each of the nodes
- coordinate system stitching approach optimizing a network wide metric in a distributed manner

Category based on Computational Organization

- second, optimize a global metric over the network
- relaxation based distributed algorithm is to use a coarse algorithm to roughly localize nodes and followed by refinement step involving node position adjustment to optimize a local error metric
- algorithms hope to approximate the optimal solution to a network-wide metric that is the sum of the local error metric at each of the nodes
- coordinate system stitching approach optimizing a network wide metric in a distributed manner
- the network is divided into small overlapping sub-regions, each of which creates an optimal local map

Category based on Computational Organization

- second, optimize a global metric over the network
- relaxation based distributed algorithm is to use a coarse algorithm to roughly localize nodes and followed by refinement step involving node position adjustment to optimize a local error metric
- algorithms hope to approximate the optimal solution to a network-wide metric that is the sum of the local error metric at each of the nodes
- coordinate system stitching approach optimizing a network wide metric in a distributed manner
- the network is divided into small overlapping sub-regions, each of which creates an optimal local map
- the sub-regions use a peer-to-peer process to merge their local maps into a single global map; this global map approximates the global optimum map

Centralized Algorithms

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)
- SDP approach to localization was pioneered by Doherty et al.

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)
- SDP approach to localization was pioneered by Doherty et al.
- geometric constraints between nodes are represented as linear matrix inequalities (LMIs)

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)
- SDP approach to localization was pioneered by Doherty et al.
- geometric constraints between nodes are represented as linear matrix inequalities (LMIs)
- once all the constraints in the network are expressed in this form, the LMIs can be combined to form a single semidefinite program

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)
- SDP approach to localization was pioneered by Doherty et al.
- geometric constraints between nodes are represented as linear matrix inequalities (LMIs)
- once all the constraints in the network are expressed in this form, the LMIs can be combined to form a single semidefinite program
- this is solved to produce a bounding region for each node, which Doherty et al simplify to be a bounding box
- unfortunately, not all geometric constraints can be expressed as LMIs

Centralized Algorithms

- Two types of processing: semidefinite programming (SDP) and multidimensional scaling (MDS)
- SDP approach to localization was pioneered by Doherty et al.
- geometric constraints between nodes are represented as linear matrix inequalities (LMIs)
- once all the constraints in the network are expressed in this form, the LMIs can be combined to form a single semidefinite program
- this is solved to produce a bounding region for each node, which Doherty et al simplify to be a bounding box
- unfortunately, not all geometric constraints can be expressed as LMIs
- only constraints that form convex regions are amenable to representation as an LMI

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints
- inability to accommodate precise range data may prove to be a significant drawback

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints
- inability to accommodate precise range data may prove to be a significant drawback
- solving the linear or SDP must be done centrally

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints
- inability to accommodate precise range data may prove to be a significant drawback
- solving the linear or SDP must be done centrally
- the relevant operation is $O(k^2)$ for angle of arrival data, and $O(k^3)$ when radial (e.g. hop count) data is included, where k is the number of convex constraints needed to describe the network

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints
- inability to accommodate precise range data may prove to be a significant drawback
- solving the linear or SDP must be done centrally
- the relevant operation is $O(k^2)$ for angle of arrival data, and $O(k^3)$ when radial (e.g. hop count) data is included, where k is the number of convex constraints needed to describe the network
- advantage of this algorithm is that given a set of convex constraints on a node's position, SDP simply finds the intersection of the constraints

Centralized Algorithms: Issues with SDP

- angle of arrival data can be represented as a triangle and hop count data can be represented as a circle, but precise range data cannot be conveniently represented, since rings cannot be expressed as convex constraints
- inability to accommodate precise range data may prove to be a significant drawback
- solving the linear or SDP must be done centrally
- the relevant operation is $O(k^2)$ for angle of arrival data, and $O(k^3)$ when radial (e.g. hop count) data is included, where k is the number of convex constraints needed to describe the network
- advantage of this algorithm is that given a set of convex constraints on a node's position, SDP simply finds the intersection of the constraints
- SDP's poor scaling and inability to effectively use range data will likely preclude the algorithm's use in practice

Centralized Algorithms

- MDS-MAP developed by Shang et al.

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known
- MDS is an $O(n^3)$ algorithm that uses the Law of Cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known
- MDS is an $O(n^3)$ algorithm that uses the Law of Cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances
- MDS-MAP performs well on RSSI data alone

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known
- MDS is an $O(n^3)$ algorithm that uses the Law of Cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances
- MDS-MAP performs well on RSSI data alone
- classical metric MDS has fours stages

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known
- MDS is an $O(n^3)$ algorithm that uses the Law of Cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances
- MDS-MAP performs well on RSSI data alone
- classical metric MDS has fours stages
- Step 1 Gather ranging data from the network, and form a sparse matrix R, where r_{ij} is the range between nodes i and j, or zero if no range was collected (for instance if i and j are physically too far apart)

Centralized Algorithms

- MDS-MAP developed by Shang et al.
- there are n points, suspended in a volume, the positions of the points are not known, but the distance between each pair of points is known
- MDS is an $O(n^3)$ algorithm that uses the Law of Cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances
- MDS-MAP performs well on RSSI data alone
- classical metric MDS has fours stages
- Step 1 Gather ranging data from the network, and form a sparse matrix R, where r_{ij} is the range between nodes i and j, or zero if no range was collected (for instance if i and j are physically too far apart)
- Step 2 Run a standard all pairs shortest path algorithm (Dijkstra's, Floyd's) on R to produce a complete matrix of inter-node distances D

Centralized Algorithms: Issues with MDS

- Step 3 Run classical metric MDS on D to find estimated node positions X

Centralized Algorithms: Issues with MDS

- Step 3 Run classical metric MDS on D to find estimated node positions X
- Step 4 Transform the solution X into global coordinates using some number of fixed anchor nodes using a coordinate system registration routine

Centralized Algorithms: Issues with MDS

- Step 3 Run classical metric MDS on D to find estimated node positions X
- Step 4 Transform the solution X into global coordinates using some number of fixed anchor nodes using a coordinate system registration routine
- MDS-MAP estimates improve as ranging improves, it does not use anchor nodes very well, since it effectively ignores their data until stage 4

Centralized Algorithms: Issues with MDS

- Step 3 Run classical metric MDS on D to find estimated node positions X
- Step 4 Transform the solution X into global coordinates using some number of fixed anchor nodes using a coordinate system registration routine
- MDS-MAP estimates improve as ranging improves, it does not use anchor nodes very well, since it effectively ignores their data until stage 4
- its performance lags behind other algorithms as anchor density increases

Centralized Algorithms: Issues with MDS

- Step 3 Run classical metric MDS on D to find estimated node positions X
- Step 4 Transform the solution X into global coordinates using some number of fixed anchor nodes using a coordinate system registration routine
- MDS-MAP estimates improve as ranging improves, it does not use anchor nodes very well, since it effectively ignores their data until stage 4
- its performance lags behind other algorithms as anchor density increases
- poor asymptotic performance, which is $O(n^3)$ on account of stages 2 and 3
- this problem can be partially ameliorated using coordinate system stitching

Distributed Algorithms

Distributed Algorithms

- relevant computation is done on the sensor nodes themselves

Distributed Algorithms

- relevant computation is done on the sensor nodes themselves
- localize nodes directly into the global coordinate space of the beacons

Distributed Algorithms

- relevant computation is done on the sensor nodes themselves
- localize nodes directly into the global coordinate space of the beacons
- extrapolate unknown node positions from beacon positions
- four beacon-based distributed algorithms: diffusion, bounding box, gradient multilateration, and APIT
- Diffusion arises from a very simple idea: the most likely position of a node is at the centroid of its neighbors positions
- Diffusion algorithms require only radio connectivity data

Distributed Algorithms

- relevant computation is done on the sensor nodes themselves
- localize nodes directly into the global coordinate space of the beacons
- extrapolate unknown node positions from beacon positions
- four beacon-based distributed algorithms: diffusion, bounding box, gradient multilateration, and APIT
- Diffusion arises from a very simple idea: the most likely position of a node is at the centroid of its neighbors positions
- Diffusion algorithms require only radio connectivity data
- accuracy is poor when node density is low

Distributed Algorithms

- relevant computation is done on the sensor nodes themselves
- localize nodes directly into the global coordinate space of the beacons
- extrapolate unknown node positions from beacon positions
- four beacon-based distributed algorithms: diffusion, bounding box, gradient multilateration, and APIT
- Diffusion arises from a very simple idea: the most likely position of a node is at the centroid of its neighbors positions
- Diffusion algorithms require only radio connectivity data
- accuracy is poor when node density is low
- Bounding Box algorithm is a computationally simple method of localizing nodes given their ranges to several beacons

Distributed Algorithms

- each node assumes that it lies within the intersection of its beacons' bounding boxes

Distributed Algorithms

- each node assumes that it lies within the intersection of its beacons' bounding boxes
- the position of a node is then the center of this final bounding box

Distributed Algorithms

- each node assumes that it lies within the intersection of its beacons' bounding boxes
- the position of a node is then the center of this final bounding box
- accuracy of the bounding box approach is best when nodes' actual positions are closer to the center of their beacons

Distributed Algorithms

- each node assumes that it lies within the intersection of its beacons' bounding boxes
- the position of a node is then the center of this final bounding box
- accuracy of the bounding box approach is best when nodes' actual positions are closer to the center of their beacons
- bounding box works best when sensor nodes have extreme computational limitations, since other algorithms may simply be infeasible

Distributed Algorithms

- each node assumes that it lies within the intersection of its beacons' bounding boxes
- the position of a node is then the center of this final bounding box
- accuracy of the bounding box approach is best when nodes' actual positions are closer to the center of their beacons
- bounding box works best when sensor nodes have extreme computational limitations, since other algorithms may simply be infeasible
- otherwise, more mathematically rigorous approaches such as gradient multilateration may be more appropriate

Distributed Algorithms: Gradient Method

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points
- these algorithms all assume that there are at least three beacon nodes somewhere in the network (though probably more)

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points
- these algorithms all assume that there are at least three beacon nodes somewhere in the network (though probably more)
- each of these beacon nodes propagates a gradient through the network, which is the distributed equivalent of computing the shortest path distance between all the beacons and all of the unlocalized nodes
- the gradient propagation is as follows:

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points
- these algorithms all assume that there are at least three beacon nodes somewhere in the network (though probably more)
- each of these beacon nodes propagates a gradient through the network, which is the distributed equivalent of computing the shortest path distance between all the beacons and all of the unlocalized nodes
- the gradient propagation is as follows:
- Step 1 For each node j and beacon k, let d_{jk} (the distance from j to k) be 0 if $j = k$ and ∞ otherwise

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points
- these algorithms all assume that there are at least three beacon nodes somewhere in the network (though probably more)
- each of these beacon nodes propagates a gradient through the network, which is the distributed equivalent of computing the shortest path distance between all the beacons and all of the unlocalized nodes
- the gradient propagation is as follows:
- Step 1 For each node j and beacon k, let d_{jk} (the distance from j to k) be 0 if $j = k$ and ∞ otherwise
- Step 2 On each node j, perform the following steps repeatedly:

Distributed Algorithms: Gradient Method

- the principal mathematical operation - called multilateration
- Multilateration is a great deal like triangulation, except that it can incorporate ranges from more than three reference points
- these algorithms all assume that there are at least three beacon nodes somewhere in the network (though probably more)
- each of these beacon nodes propagates a gradient through the network, which is the distributed equivalent of computing the shortest path distance between all the beacons and all of the unlocalized nodes
- the gradient propagation is as follows:
- Step 1 For each node j and beacon k, let d_{jk} (the distance from j to k) be 0 if $j = k$ and ∞ otherwise
- Step 2 On each node j, perform the following steps repeatedly:
- Step 2a For each beacon k and neighbor i, retrieve d_{ik} from i

Distributed Algorithms: Gradient Method

- Step 2b For each beacon k and neighbor i, apply the following update formula: $d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{jk})$

Distributed Algorithms: Gradient Method

- Step 2b For each beacon k and neighbor i, apply the following update formula: $d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{jk})$
- $\hat{r}_{ij}$ is the estimated distance between nodes i and j. These internode distance estimates can be either unweighted (one if there is connectivity, zero otherwise) or measured distances (e.g. using RSSI or TDoA)

Distributed Algorithms: Gradient Method

- Step 2b For each beacon k and neighbor i, apply the following update formula: $d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{jk})$
- $\hat{r}_{ij}$ is the estimated distance between nodes i and j. These internode distance estimates can be either unweighted (one if there is connectivity, zero otherwise) or measured distances (e.g. using RSSI or TDoA)
- after some amount of settling time, each value d_{jk} will be the length of the shortest path between node j and beacon k

Distributed Algorithms: Gradient Method

- Step 2b For each beacon k and neighbor i, apply the following update formula: $d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{jk})$
- $\hat{r}_{ij}$ is the estimated distance between nodes i and j. These internode distance estimates can be either unweighted (one if there is connectivity, zero otherwise) or measured distances (e.g. using RSSI or TDoA)
- after some amount of settling time, each value d_{jk} will be the length of the shortest path between node j and beacon k
- given perfect internode distance estimates, gradient distance estimates will always be longer than (or exactly equal) to corresponding straight line distances

Distributed Algorithms: Gradient Method

- Step 2b For each beacon k and neighbor i, apply the following update formula: $d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{jk})$
- $\hat{r}_{ij}$ is the estimated distance between nodes i and j. These internode distance estimates can be either unweighted (one if there is connectivity, zero otherwise) or measured distances (e.g. using RSSI or TDoA)
- after some amount of settling time, each value d_{jk} will be the length of the shortest path between node j and beacon k
- given perfect internode distance estimates, gradient distance estimates will always be longer than (or exactly equal) to corresponding straight line distances
- given imperfect internode distance estimates, gradient based distance estimate can actually be shorter than straight distances

Distributed Algorithms: APIT

Distributed Algorithms: APIT

- quite a bit different from the beacon-based distributed algorithms
- it uses a novel area-based approach, in which nodes are assumed to be able to hear a fairly large number of beacons

Distributed Algorithms: APIT

- quite a bit different from the beacon-based distributed algorithms
- it uses a novel area-based approach, in which nodes are assumed to be able to hear a fairly large number of beacons
- it does not assume that nodes can range to these beacons, instead, a node forms some number of beacon triangles, where a beacon triangle is the triangle formed by three arbitrary beacons

Distributed Algorithms: APIT

- quite a bit different from the beacon-based distributed algorithms
- it uses a novel area-based approach, in which nodes are assumed to be able to hear a fairly large number of beacons
- it does not assume that nodes can range to these beacons, instead, a node forms some number of beacon triangles, where a beacon triangle is the triangle formed by three arbitrary beacons
- the node then decides whether it is inside or outside a given triangle by comparing signal strength measurements with its nearby non-beacon neighbors

Distributed Algorithms: APIT

- quite a bit different from the beacon-based distributed algorithms
- it uses a novel area-based approach, in which nodes are assumed to be able to hear a fairly large number of beacons
- it does not assume that nodes can range to these beacons, instead, a node forms some number of beacon triangles, where a beacon triangle is the triangle formed by three arbitrary beacons
- the node then decides whether it is inside or outside a given triangle by comparing signal strength measurements with its nearby non-beacon neighbors
- once this process is complete, the node simply finds the intersection of the beacon triangles that contains it

Distributed Algorithms: APIT

- quite a bit different from the beacon-based distributed algorithms
- it uses a novel area-based approach, in which nodes are assumed to be able to hear a fairly large number of beacons
- it does not assume that nodes can range to these beacons, instead, a node forms some number of beacon triangles, where a beacon triangle is the triangle formed by three arbitrary beacons
- the node then decides whether it is inside or outside a given triangle by comparing signal strength measurements with its nearby non-beacon neighbors
- once this process is complete, the node simply finds the intersection of the beacon triangles that contains it
- the node chooses the centroid of this intersection region as its position estimate

Network-wide localization

Network-wide localization

- Node localization is determining the location of a single unknown node given a number of nearby references

Network-wide localization

- Node localization is determining the location of a single unknown node given a number of nearby references
- A broader problem in sensor systems is that of network localization, where several unknown nodes have to be localized in a network with a few reference nodes

Network-wide localization

- Node localization is determining the location of a single unknown node given a number of nearby references
- A broader problem in sensor systems is that of network localization, where several unknown nodes have to be localized in a network with a few reference nodes
- the network localization problem can rarely be neatly decomposed into a number of separate node localization problems (since there may be unknown nodes that have no reference nodes within range)

Network-wide localization

- Node localization is determining the location of a single unknown node given a number of nearby references
- A broader problem in sensor systems is that of network localization, where several unknown nodes have to be localized in a network with a few reference nodes
- the network localization problem can rarely be neatly decomposed into a number of separate node localization problems (since there may be unknown nodes that have no reference nodes within range)
- the node localization and ranging techniques described above do often form an integral component of solutions to network localization

Network-wide localization

- Node localization is determining the location of a single unknown node given a number of nearby references
- A broader problem in sensor systems is that of network localization, where several unknown nodes have to be localized in a network with a few reference nodes
- the network localization problem can rarely be neatly decomposed into a number of separate node localization problems (since there may be unknown nodes that have no reference nodes within range)
- the node localization and ranging techniques described above do often form an integral component of solutions to network localization
- the performance of network localization depends very much on the resources and information available within the network

Network-wide localization

- Several scenarios are possible:

Network-wide localization

- Several scenarios are possible:
- for instance there may be no reference nodes at all, so that perhaps only relative coordinates can be determined for the unknown nodes

Network-wide localization

- Several scenarios are possible:
- for instance there may be no reference nodes at all, so that perhaps only relative coordinates can be determined for the unknown nodes
- if present, the number/density of reference nodes may vary (more reference nodes, the lower the network localization error)
- there may be just a single mobile reference

Network-wide localization

- Several scenarios are possible:
- for instance there may be no reference nodes at all, so that perhaps only relative coordinates can be determined for the unknown nodes
- if present, the number/density of reference nodes may vary (more reference nodes, the lower the network localization error)
- there may be just a single mobile reference
- some network localization approaches are centralized
- the nodes in the network need to be localized only once, post-deployment

Network-wide localization

- Several scenarios are possible:
- for instance there may be no reference nodes at all, so that perhaps only relative coordinates can be determined for the unknown nodes
- if present, the number/density of reference nodes may vary (more reference nodes, the lower the network localization error)
- there may be just a single mobile reference
- some network localization approaches are centralized
- the nodes in the network need to be localized only once, post-deployment
- other network localization approaches are distributed, often involving the iterative communication of updated location information

RSS-based joint estimation

RSS-based joint estimation

- if radio signal strengths can be measured between all pairs of nodes in the network that are within detection range, then a joint maximum likelihood estimation (MLE) technique can be used to determine the location of unknown nodes in a network

RSS-based joint estimation

- if radio signal strengths can be measured between all pairs of nodes in the network that are within detection range, then a joint maximum likelihood estimation (MLE) technique can be used to determine the location of unknown nodes in a network
- in the joint MLE technique, first an expression is derived for the likelihood that the obtained matrix of power measurements would be received given a particular location set for all nodes;

RSS-based joint estimation

- if radio signal strengths can be measured between all pairs of nodes in the network that are within detection range, then a joint maximum likelihood estimation (MLE) technique can be used to determine the location of unknown nodes in a network
- in the joint MLE technique, first an expression is derived for the likelihood that the obtained matrix of power measurements would be received given a particular location set for all nodes;
- the objective is then to find the location set that maximizes this likelihood

RSS-based joint estimation

- if radio signal strengths can be measured between all pairs of nodes in the network that are within detection range, then a joint maximum likelihood estimation (MLE) technique can be used to determine the location of unknown nodes in a network
- in the joint MLE technique, first an expression is derived for the likelihood that the obtained matrix of power measurements would be received given a particular location set for all nodes;
- the objective is then to find the location set that maximizes this likelihood
- the performance of this joint MLE technique has been verified through simulations and experiments to show that localization of the order of 2 meters is possible when there is a high density of unknown nodes, even if there are only a few reference nodes sparsely placed

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)
- the algorithm is quite simple and applies the basic triangulation technique for node localization in an iterative manner to determine the locations of all nodes

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)
- the algorithm is quite simple and applies the basic triangulation technique for node localization in an iterative manner to determine the locations of all nodes
- one begins by determining the location of an unknown node that has the most reference nodes in its neighborhood

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)
- the algorithm is quite simple and applies the basic triangulation technique for node localization in an iterative manner to determine the locations of all nodes
- one begins by determining the location of an unknown node that has the most reference nodes in its neighborhood
- in a distributed version, the location of any node with sufficient references in its neighborhood may be calculated as the initial step

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)
- the algorithm is quite simple and applies the basic triangulation technique for node localization in an iterative manner to determine the locations of all nodes
- one begins by determining the location of an unknown node that has the most reference nodes in its neighborhood
- in a distributed version, the location of any node with sufficient references in its neighborhood may be calculated as the initial step
- this node is then added to the set of reference nodes and the process is repeated
- it may not be applicable if there is no node that has sufficient (≥ 3 for the 2D plane) reference nodes in its neighborhood

Iterative multilateration

- it is applicable whenever inter-node distance information is available between all neighboring nodes (regardless of whether it is obtained through RSS measurements or TDoA or any other approach)
- the algorithm is quite simple and applies the basic triangulation technique for node localization in an iterative manner to determine the locations of all nodes
- one begins by determining the location of an unknown node that has the most reference nodes in its neighborhood
- in a distributed version, the location of any node with sufficient references in its neighborhood may be calculated as the initial step
- this node is then added to the set of reference nodes and the process is repeated
- it may not be applicable if there is no node that has sufficient (≥ 3 for the 2D plane) reference nodes in its neighborhood
- the use of localized unknown nodes as reference nodes can introduce substantial cumulative error

Collaborative multilateration

- approach to tackle the deficiency of iterative multilateration with respect to nodes with insufficient reference neighbors

Collaborative multilateration

- approach to tackle the deficiency of iterative multilateration with respect to nodes with insufficient reference neighbors
- the key insight is to determine collaborative subgraphs within the network that contain reference and unknown nodes in a topology such that
- their positions and inter-node distances can be written as an over-constrained set of quadratic equations with a unique solution for the location of unknown nodes (which can be obtained through gradient descent or local search algorithms)
- used in conjunction with iterative multilateration
- useful in portions of the network where the reference node density is low

Multi-hop distance-estimation approaches

- alternative approach to network localization utilizes estimates of distances to reference nodes that may be several hops away

Multi-hop distance-estimation approaches

- alternative approach to network localization utilizes estimates of distances to reference nodes that may be several hops away
- these distances are propagated from reference nodes to unknown nodes using a basic distance-vector technique

Multi-hop distance-estimation approaches

- alternative approach to network localization utilizes estimates of distances to reference nodes that may be several hops away
- these distances are propagated from reference nodes to unknown nodes using a basic distance-vector technique
- once distance estimates are available from each unknown node to different reference nodes throughout the network, a triangulation technique can be employed to determine their locations

Multi-hop distance-estimation approaches

- alternative approach to network localization utilizes estimates of distances to reference nodes that may be several hops away
- these distances are propagated from reference nodes to unknown nodes using a basic distance-vector technique
- once distance estimates are available from each unknown node to different reference nodes throughout the network, a triangulation technique can be employed to determine their locations
- there are three variants of this approach: DV-hop, DV distance, Euclidean propagation

Multi-hop distance-estimation approaches

- alternative approach to network localization utilizes estimates of distances to reference nodes that may be several hops away
- these distances are propagated from reference nodes to unknown nodes using a basic distance-vector technique
- once distance estimates are available from each unknown node to different reference nodes throughout the network, a triangulation technique can be employed to determine their locations
- there are three variants of this approach: DV-hop, DV distance, Euclidean propagation
- the average distance per hop depends upon the network density, and is assumed to be known
- the relative performance of these three schemes depends on factors such as the radio range and accuracy of available distance estimates

Multi-dimensional scaling

- the node localization problem with defining the network as an undirected graph with vertices V and edges E

Multi-dimensional scaling

- the node localization problem with defining the network as an undirected graph with vertices V and edges E
- the vertices correspond to the nodes, of which zero or more may be special nodes, called anchors, whose positions are already known

Multi-dimensional scaling

- the node localization problem with defining the network as an undirected graph with vertices V and edges E
- the vertices correspond to the nodes, of which zero or more may be special nodes, called anchors, whose positions are already known
- assume that all the nodes being considered in the positioning problem form a connected graph, i.e., there is a path between every pair of nodes

Multi-dimensional scaling

- the node localization problem with defining the network as an undirected graph with vertices V and edges E
- the vertices correspond to the nodes, of which zero or more may be special nodes, called anchors, whose positions are already known
- assume that all the nodes being considered in the positioning problem form a connected graph, i.e., there is a path between every pair of nodes
- classical MDS is the simplest case of MDS: the proximities of objects are treated as distances in a Euclidean space

Multi-dimensional scaling

- the node localization problem with defining the network as an undirected graph with vertices V and edges E
- the vertices correspond to the nodes, of which zero or more may be special nodes, called anchors, whose positions are already known
- assume that all the nodes being considered in the positioning problem form a connected graph, i.e., there is a path between every pair of nodes
- classical MDS is the simplest case of MDS: the proximities of objects are treated as distances in a Euclidean space
- the goal of MDS is to find a configuration of points in a multidimensional space such that the inter-point distances are related to the provided proximities by some transformation (e.g., a linear transformation)

Multi-dimensional scaling

- let p_{ij} refer to the proximity measure between objects i and j

Multi-dimensional scaling

- let p_{ij} refer to the proximity measure between objects i and j
- the Euclidean distance between two points $X_i = (x_{i1}; x_{i2}; x_{i3} \ldots x_{im})$ and $X_j = (x_{j1}; x_{j2}; x_{j3} \ldots x_{jm})$ in an m dimensional space is

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

Multi-dimensional scaling

- let p_{ij} refer to the proximity measure between objects i and j
- the Euclidean distance between two points $X_i = (x_{i1}; x_{i2}; x_{i3} \dots x_{im})$ and $X_j = (x_{j1}; x_{j2}; x_{j3} \dots x_{jm})$ in an m dimensional space is

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

- the Euclidean distances are related to the proximities by a transformation $d_{ij} = f(p_{ij})$

Multi-dimensional scaling

- let p_{ij} refer to the proximity measure between objects i and j
- the Euclidean distance between two points $X_i = (x_{i1}; x_{i2}; x_{i3} \ldots x_{im})$ and $X_j = (x_{j1}; x_{j2}; x_{j3} \ldots x_{jm})$ in an m dimensional space is

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

- the Euclidean distances are related to the proximities by a transformation $d_{ij} = f(p_{ij})$
- in the classical MDS, a linear transformation model is assumed, $d_{ij} = a + bp_{ij}$

Multi-dimensional scaling

- let p_{ij} refer to the proximity measure between objects i and j
- the Euclidean distance between two points $X_i = (x_{i1}; x_{i2}; x_{i3} \ldots x_{im})$ and $X_j = (x_{j1}; x_{j2}; x_{j3} \ldots x_{jm})$ in an m dimensional space is

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

- the Euclidean distances are related to the proximities by a transformation $d_{ij} = f(p_{ij})$
- in the classical MDS, a linear transformation model is assumed, $d_{ij} = a + bp_{ij}$
- the distances D are determined so that they are as close to the proximities P as possible

Multi-dimensional scaling

- define $I(P) = D + E$ $I(P)$ is a linear transformation of proximities, E is a matrix of errors

Multi-dimensional scaling

- define $I(P) = D + E$ $I(P)$ is a linear transformation of proximities, E is a matrix of errors
- D is a function of coordinates X, the goal of classical MDS is to calculate X such that the sum of squares of E is minimized, subject to suitable normalization of X

Multi-dimensional scaling

- define $I(P) = D + E$ $I(P)$ is a linear transformation of proximities, E is a matrix of errors
- D is a function of coordinates X, the goal of classical MDS is to calculate X such that the sum of squares of E is minimized, subject to suitable normalization of X
- in classical MDS, P is shifted to the center and coordinates X can be computed from the double centered P through singular value decomposition

Multi-dimensional scaling

- define $I(P) = D + E$ $I(P)$ is a linear transformation of proximities, E is a matrix of errors
- D is a function of coordinates X, the goal of classical MDS is to calculate X such that the sum of squares of E is minimized, subject to suitable normalization of X
- in classical MDS, P is shifted to the center and coordinates X can be computed from the double centered P through singular value decomposition
- for an $n \times n$ P matrix for n points and m dimensions of each point, it can shown that

$$-\frac{1}{2}\left(p_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} p_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} p_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}^2\right) = \sum_{k=1}^{m} x_{ik}x_{jk}$$

Multi-dimensional scaling

- the double centered matrix on the left hand side (B) is symmetric
- calculated B and performing singular value decomposition on B gives $B = VAV$

Multi-dimensional scaling

- the double centered matrix on the left hand side (B) is symmetric
- calculated B and performing singular value decomposition on B gives $B = VAV$
- the coordinate matrix becomes $X = VA^{1/2}$

Multi-dimensional scaling

- the double centered matrix on the left hand side (B) is symmetric
- calculated B and performing singular value decomposition on B gives $B = VAV$
- the coordinate matrix becomes $X = VA^{1/2}$
- retaining the first r largest eigenvalues and eigenvectors $(r < m)$ leads to a solution in lower dimension

Multi-dimensional scaling

- the double centered matrix on the left hand side (B) is symmetric
- calculated B and performing singular value decomposition on B gives $B = VAV$
- the coordinate matrix becomes $X = VA^{1/2}$
- retaining the first r largest eigenvalues and eigenvectors $(r < m)$ leads to a solution in lower dimension
- this implies that the summation over k runs from 1 to r instead of m
- this is the best low rank approximation in the least-squares sense
- for example, for a 2D network, we take the first 2 largest eigenvalues and eigenvectors to construct the best 2D approximation

Multilateration

Multilateration

- simple technique, but the specific mathematics of its implementation vary widely, as do its application in sensor networks

Multilateration

- simple technique, but the specific mathematics of its implementation vary widely, as do its application in sensor networks
- given m nodes with known Cartesian positions $b_i, i = 1 \ldots, m$

Multilateration

- simple technique, but the specific mathematics of its implementation vary widely, as do its application in sensor networks
- given m nodes with known Cartesian positions $b_i, i = 1 \ldots, m$
- possibly noisy range measurements r_i from the known nodes to an unknown node s
- multilateration finds the most likely position of s

Multilateration

- simple technique, but the specific mathematics of its implementation vary widely, as do its application in sensor networks
- given m nodes with known Cartesian positions $b_i, i = 1 \ldots, m$
- possibly noisy range measurements r_i from the known nodes to an unknown node s
- multilateration finds the most likely position of s
- multilateration is typically done by minimizing the squared error between the observed ranges r_i and the predicted distance $||s - b_i||$

$$s = \arg\min_s E(s)$$

$$E(s) = \sum_{i=1}^{m} (||s - b_i|| - r_i)^2$$

Multilateration

- simple technique, but the specific mathematics of its implementation vary widely, as do its application in sensor networks
- given m nodes with known Cartesian positions $b_i, i = 1 \ldots, m$
- possibly noisy range measurements r_i from the known nodes to an unknown node s
- multilateration finds the most likely position of s
- multilateration is typically done by minimizing the squared error between the observed ranges r_i and the predicted distance $||s - b_i||$

$$s = \arg\min_s E(s)$$

$$E(s) = \sum_{i=1}^{m} (||s - b_i|| - r_i)^2$$

- this minimization problem can be solved using Newton-Raphson/least squares as follows:

Multilateration

- first, approximate the error function $e(s, b_i) = ||s - b_i|| - r_i$ with a first order Taylor series about s_0

Multilateration

- first, approximate the error function $e(s, b_i) = ||s - b_i|| - r_i$ with a first order Taylor series about s_0

$$\begin{aligned} e(s, b_i) &\approx e(s_0, b_i) + \bigtriangledown e(s_0, b_i)(s - s_0) \\ &= \bigtriangledown e(s_0, b_i)s - (-e(s_0, b_i) + \bigtriangledown e(s_0, b_i)s_0) \end{aligned}$$

Multilateration

- first, approximate the error function $e(s, b_i) = ||s - b_i|| - r_i$ with a first order Taylor series about s_0

$$\begin{aligned} e(s, b_i) &\approx e(s_0, b_i) + \bigtriangledown e(s_0, b_i)(s - s_0) \\ &= \bigtriangledown e(s_0, b_i)s - (-e(s_0, b_i) + \bigtriangledown e(s_0, b_i)s_0) \end{aligned}$$

$$\bigtriangledown e(s, b_i) = \frac{s - b_i}{||s - b_i||}$$

substituting this approximation back

$$s \approx \underset{s}{\arg\min} \sum_{i=1}^{m} (\bigtriangledown e(s_0, b_i)s - (-e(s_0, b_i) + \bigtriangledown e(s_0, b_i)s_0))^2$$

Multilateration

- first, approximate the error function $e(s, b_i) = ||s - b_i|| - r_i$ with a first order Taylor series about s_0

$$\begin{aligned} e(s, b_i) &\approx e(s_0, b_i) + \bigtriangledown e(s_0, b_i)(s - s_0) \\ &= \bigtriangledown e(s_0, b_i)s - (-e(s_0, b_i) + \bigtriangledown e(s_0, b_i)s_0) \end{aligned}$$

$$\bigtriangledown e(s, b_i) = \frac{s - b_i}{||s - b_i||}$$

substituting this approximation back

$$s \approx \underset{s}{\arg\min} \sum_{i=1}^{m} (\bigtriangledown e(s_0, b_i)s - (-e(s_0, b_i) + \bigtriangledown e(s_0, b_i)s_0))^2$$

$$s \approx \underset{s}{\arg\min} \; ||As - b||^2$$

Multilateration

$$A = \begin{bmatrix} \nabla e(s_0, b_1) \\ \nabla e(s_0, b_2) \\ \vdots \\ \nabla e(s_0, b_m) \end{bmatrix}$$

$$b = \begin{bmatrix} -e(s_0, b_1) + \nabla e(s_0, b_1) s_0 \\ -e(s_0, b_2) + \nabla e(s_0, b_2) s_0 \\ \vdots \\ -e(s_0, b_m) + \nabla e(s_0, b_m) s_0 \end{bmatrix}$$

- the right side of equation is in exactly the right form to be solved by an off-the-shelf iterative least squares solver
- the resulting s is a good estimate of the unknown sensor's position, provided b_i and r_i are accurate

Multilateration method: Summary

1. Choose s_0 to be a starting point for the optimization. The choice is somewhat arbitrary, but the centroid $\bar{b}$ is a good one:

$$\bar{b} = \frac{1}{m}\sum_{i=1}^{m} b_i$$

Multilateration method: Summary

1. Choose s_0 to be a starting point for the optimization. The choice is somewhat arbitrary, but the centroid $\bar{b}$ is a good one:

$$\bar{b} = \frac{1}{m}\sum_{i=1}^{m} b_i$$

2. Computer A and B using s_0
3. Compute $s_0' = \underset{x}{\arg\min}\ ||Ax - b||^2$ using a least squares solver

Multilateration method: Summary

1. Choose s_0 to be a starting point for the optimization. The choice is somewhat arbitrary, but the centroid $\bar{b}$ is a good one:

$$\bar{b} = \frac{1}{m} \sum_{i=1}^{m} b_i$$

2. Computer A and B using s_0
3. Compute $s_0' = \underset{x}{\arg\min} \; ||Ax - b||^2$ using a least squares solver
4. If $E(s_0) - E(s_0') < \epsilon$ then s_0' is the solution, otherwise set $s_0 = s_0'$ and return to Step 2

Multilateration method: Summary

1. Choose s_0 to be a starting point for the optimization. The choice is somewhat arbitrary, but the centroid $\bar{b}$ is a good one:

$$\bar{b} = \frac{1}{m} \sum_{i=1}^{m} b_i$$

2. Computer A and B using s_0
3. Compute $s_0' = \underset{x}{\arg\min} \; ||Ax - b||^2$ using a least squares solver
4. If $E(s_0) - E(s_0') < \epsilon$ then s_0' is the solution, otherwise set $s_0 = s_0'$ and return to Step 2

- many ways to solve the multilateration problem
- the one presented above is equivalent to Newton-Raphson descent on the error function E
- most alternate methods also attempt to minimize squared error using some form of iterative optimization

Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS)

- MDS comes in many variations, share a common goal
- given a set of points whose position is unknown and measured distances between each pair of points

Multidimensional Scaling (MDS)

- MDS comes in many variations, share a common goal
- given a set of points whose position is unknown and measured distances between each pair of points
- MDS determines the underlying dimensionality of the points, and finds an embedding of the points in that space that honors the pairwise distances between them

Multidimensional Scaling (MDS)

- MDS comes in many variations, share a common goal
- given a set of points whose position is unknown and measured distances between each pair of points
- MDS determines the underlying dimensionality of the points, and finds an embedding of the points in that space that honors the pairwise distances between them
- using only ranging data, without anchors or GPS, it can solve for the relative coordinates of a group of sensor nodes

Multidimensional Scaling (MDS)

- MDS comes in many variations, share a common goal
- given a set of points whose position is unknown and measured distances between each pair of points
- MDS determines the underlying dimensionality of the points, and finds an embedding of the points in that space that honors the pairwise distances between them
- using only ranging data, without anchors or GPS, it can solve for the relative coordinates of a group of sensor nodes
- classical metric MDS: uses only one matrix of dissimilarity or distance information, and metric because the dissimilarity information is quantitative (e.g. distance measurements), as opposed to ordinal

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \ldots n$

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \dots n$
- let $X = [X_1, X_2, \dots, X_n]^T$ X is $n \times m$ where m is the dimensionality of X

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \ldots n$
- let $X = [X_1, X_2, \ldots, X_n]^T$ X is $n \times m$ where m is the dimensionality of X
- m to be an unknown; let $D = [d_{ij}]$ be the $n \times n$ matrix of pairwise distance measurements

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \ldots n$
- let $X = [X_1, X_2, \ldots, X_n]^T$ X is $n \times m$ where m is the dimensionality of X
- m to be an unknown; let $D = [d_{ij}]$ be the $n \times n$ matrix of pairwise distance measurements
- d_{ij} is the measured distance between X_i and X_j for $i \neq j$, and $d_{ij} = 0$ for all i

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \dots n$
- let $X = [X_1, X_2, \dots, X_n]^T$ X is $n \times m$ where m is the dimensionality of X
- m to be an unknown; let $D = [d_{ij}]$ be the $n \times n$ matrix of pairwise distance measurements
- d_{ij} is the measured distance between X_i and X_j for $i \neq j$, and $d_{ij} = 0$ for all i
- the distance measurements d_{ij} must obey the triangular inequality $d_{ij} + d_{ik} \geq d_{jk}$ for all (i, j, k)

Multidimensional Scaling (MDS)

- let there be n sensors in a network, with positions $X_i, i = 1 \dots n$
- let $X = [X_1, X_2, \dots, X_n]^T$ X is $n \times m$ where m is the dimensionality of X
- m to be an unknown; let $D = [d_{ij}]$ be the $n \times n$ matrix of pairwise distance measurements
- d_{ij} is the measured distance between X_i and X_j for $i \neq j$, and $d_{ij} = 0$ for all i
- the distance measurements d_{ij} must obey the triangular inequality $d_{ij} + d_{ik} \geq d_{jk}$ for all (i, j, k)
- the goal of MDS is to find an assignment of X in low-dimensional space that minimizes a "Stress function" defined as

$$X = \underset{X}{\arg\min} \ \text{Stress}(X)$$

$$\text{Stress}(X) = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1}(d_{ij} - \delta_{ij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{i-1}\delta_{ij}^2}}$$

Multidimensional Scaling (MDS)

- δ_{ij} is the distance between X_i and X_j

Multidimensional Scaling (MDS)

- δ_{ij} is the distance between X_i and X_j
- the metric MDS stress function is closely related to the squared error function as seen in other techniques such as multilateration

Multidimensional Scaling (MDS)

- δ_{ij} is the distance between X_i and X_j
- the metric MDS stress function is closely related to the squared error function as seen in other techniques such as multilateration
- classical metric MDS is derived from the Law of Cosines, which states that given two sides of a triangle d_{ij}, d_{ik} and the angle between them θ_{jik} the side can be computed using the formula:

Multidimensional Scaling (MDS)

- δ_{ij} is the distance between X_i and X_j
- the metric MDS stress function is closely related to the squared error function as seen in other techniques such as multilateration
- classical metric MDS is derived from the Law of Cosines, which states that given two sides of a triangle d_{ij}, d_{ik} and the angle between them θ_{jik} the side can be computed using the formula:

$$d_{jk}^2 = d_{ij}^2 + d_{jk}^2 - 2d_{ij}d_{jk}\cos\theta_{jik}$$

$$d_{ij}d_{ik}\cos\theta_{jik} = \frac{1}{2}(d_{ij}^2 + d_{ik}^2 - d_{jk}^2)$$

Multidimensional Scaling (MDS)

- δ_{ij} is the distance between X_i and X_j
- the metric MDS stress function is closely related to the squared error function as seen in other techniques such as multilateration
- classical metric MDS is derived from the Law of Cosines, which states that given two sides of a triangle d_{ij}, d_{ik} and the angle between them θ_{jik} the side can be computed using the formula:

$$d_{jk}^2 = d_{ij}^2 + d_{jk}^2 - 2d_{ij}d_{jk}\cos\theta_{jik}$$

$$d_{ij}d_{ik}\cos\theta_{jik} = \frac{1}{2}(d_{ij}^2 + d_{ik}^2 - d_{jk}^2)$$

- left side of equation can be rewritten as a dot product

$$(X_j - X_i)\cdot(X_k - X_i) = \frac{1}{2}(d_{ij}^2 + d_{ik}^2 - d_{jk}^2)$$

Multidimensional Scaling (MDS)

- if all measurements are perfect, then a good zero-stress way to solve for the positions X is to choose some X_0 from X to be the origin of a coordinate system and construct a matrix $B_{(n-1)x(n-1)}$ as follows:

$$b_{ij} = \frac{1}{2}(d_{0i}^2 + d_{0j}^2 - d_{ij}^2)$$

Multidimensional Scaling (MDS)

- if all measurements are perfect, then a good zero-stress way to solve for the positions X is to choose some X_0 from X to be the origin of a coordinate system and construct a matrix $B_{(n-1)x(n-1)}$ as follows:

$$b_{ij} = \frac{1}{2}(d_{0i}^2 + d_{0j}^2 - d_{ij}^2)$$

- B is known as the matrix of scalar products; B can be written in terms of X

Multidimensional Scaling (MDS)

- if all measurements are perfect, then a good zero-stress way to solve for the positions X is to choose some X_0 from X to be the origin of a coordinate system and construct a matrix $B_{(n-1)x(n-1)}$ as follows:

$$b_{ij} = \frac{1}{2}(d_{0i}^2 + d_{0j}^2 - d_{ij}^2)$$

- B is known as the matrix of scalar products; B can be written in terms of X
- $X'_{(n-1)xm}$ is the matrix X where each of the X_i's is shifted to have its origin at X_0: $X'_i = X_i - X_0$; then, $X'X'^T = B$
- X' can be solved by taking an eigen decomposition B into an orthonormal matrix of eigenvectors and a diagonal matrix of matching eigen values:

$$B = X'X'^T = UVU^T \quad X' = UV^{1/2}$$

Multidimensional Scaling (MDS)

- the problem is that X' has too many columns: need to find X in 2-D or 3-D space
- to do this, throw away all but the two or three largest eigenvalues from V, leaving 2×2 or 3×3 diagonal matrix,

Multidimensional Scaling (MDS)

- the problem is that X' has too many columns: need to find X in 2-D or 3-D space
- to do this, throw away all but the two or three largest eigenvalues from V, leaving 2×2 or 3×3 diagonal matrix,
- throw away the matching eigenvectors (columns) of U, leaving $U_{(n-1)x2}$ or $U_{(n-1)x3}$; then X' has the proper dimensionality
- this method produces a coordinate system that is a linear transformation from the coordinate system of the true X_i's

Multidimensional Scaling (MDS)

- the problem is that X' has too many columns: need to find X in 2-D or 3-D space
- to do this, throw away all but the two or three largest eigenvalues from V, leaving 2×2 or 3×3 diagonal matrix,
- throw away the matching eigenvectors (columns) of U, leaving $U_{(n-1)x2}$ or $U_{(n-1)x3}$; then X' has the proper dimensionality
- this method produces a coordinate system that is a linear transformation from the coordinate system of the true X_i's
- reconciling the two requires a registration procedure like that of Coordinate System Registration

Multidimensional Scaling (MDS)

- the problem is that X' has too many columns: need to find X in 2-D or 3-D space
- to do this, throw away all but the two or three largest eigenvalues from V, leaving 2×2 or 3×3 diagonal matrix,
- throw away the matching eigenvectors (columns) of U, leaving $U_{(n-1)x2}$ or $U_{(n-1)x3}$; then X' has the proper dimensionality
- this method produces a coordinate system that is a linear transformation from the coordinate system of the true X_i's
- reconciling the two requires a registration procedure like that of Coordinate System Registration
- this method only works when the data d_{ij} is perfect, which is an unrealistic assumption

Multidimensional Scaling (MDS)

- the problem is that X' has too many columns: need to find X in 2-D or 3-D space
- to do this, throw away all but the two or three largest eigenvalues from V, leaving 2×2 or 3×3 diagonal matrix,
- throw away the matching eigenvectors (columns) of U, leaving $U_{(n-1)x2}$ or $U_{(n-1)x3}$; then X' has the proper dimensionality
- this method produces a coordinate system that is a linear transformation from the coordinate system of the true X_i's
- reconciling the two requires a registration procedure like that of Coordinate System Registration
- this method only works when the data d_{ij} is perfect, which is an unrealistic assumption
- above, a single point from our data is chosen to be the origin
- the choice gives X_0 an undue influence on the error of X

Multidimensional Scaling (MDS)

- real MDS doesn't use a point from the data; rather, it uses a special point in the center of the X_i's

Multidimensional Scaling (MDS)

- real MDS doesn't use a point from the data; rather, it uses a special point in the center of the X_i's
- this point is found by "double centering" the squared distance matrix
- the squared matrix $D^2 = [d_{ij}^2]$

Multidimensional Scaling (MDS)

- real MDS doesn't use a point from the data; rather, it uses a special point in the center of the X_i's
- this point is found by "double centering" the squared distance matrix
- the squared matrix $D^2 = [d_{ij}^2]$
- to double center a matrix, subtract the row and column means from each element, then, add the grand mean to each element and finally, multiply by $-1/2$
- the element-wise formula for double centering is below:

$$\begin{aligned} b_{ij} &= -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{kj}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{ik}^2 + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} d_{kl}^2\right) \\ &= \sum_{a=1}^{m} x_{ia}x_{ja} \end{aligned}$$

Multidimensional Scaling (MDS)

$$B_{nxn} = -\frac{1}{2} J D^2 J = X X^T$$

$$J_{nxn} = I_{nxn} - \frac{1}{n} e^T e$$

$$e_{1xn} = [1, 1, 1, \ldots, 1]$$

- it is an expression for X in terms of D, in m-dimensional space
- if $m = n - 1$, then there is a trivial assignment of $X_1 \ldots X_n$ that makes Stress$(X) = 0$
- as m decreases, it turns out that Stress(X) must increase or stay the same; it can not decreases
- the measurements D originates from a 2 or 3 dimensional space
- if the measurements from D are perfect, then there is a zero stress assignment of X when $m = 2$ or 3

Multidimensional Scaling (MDS)

- measurement error makes it unlikely that such an assignment really exists
- some stress is inevitable as the dimensionality is reduced from n to 2 or 3
- this dimensionality reduction is done by taking an eigen-decomposition of B, then removing eigenvalues and eigenvectors

Multidimensional Scaling (MDS)

- measurement error makes it unlikely that such an assignment really exists
- some stress is inevitable as the dimensionality is reduced from n to 2 or 3
- this dimensionality reduction is done by taking an eigen-decomposition of B, then removing eigenvalues and eigenvectors
- this is a safe operation because B is symmetric positive definite, and therefore has n positive eigenvalues

$$B = XX^T = UVU^T \quad X = UV^{1/2}$$

Multidimensional Scaling (MDS)

- measurement error makes it unlikely that such an assignment really exists
- some stress is inevitable as the dimensionality is reduced from n to 2 or 3
- this dimensionality reduction is done by taking an eigen-decomposition of B, then removing eigenvalues and eigenvectors
- this is a safe operation because B is symmetric positive definite, and therefore has n positive eigenvalues

$$B = XX^T = UVU^T \quad X = UV^{1/2}$$

- MDS provides a method of converting a complete matrix of distance measurements to a matching topology in 2-space or 3-space
- this conversion is quite resilient to measurement error, since increased measurement error simply becomes an increase in the stress function

Multidimensional Scaling (MDS)

- MDS has some disadvantages

Multidimensional Scaling (MDS)

- MDS has some disadvantages
- the main computation of MDS, the eigen-decomposition of B requires $O(n^3)$ time

Multidimensional Scaling (MDS)

- MDS has some disadvantages
- the main computation of MDS, the eigen-decomposition of B requires $O(n^3)$ time
- as a result, a single pass of MDS cannot operate on a large topology, particularly in the constrained computational environment of sensor networks

Multidimensional Scaling (MDS)

- MDS has some disadvantages
- the main computation of MDS, the eigen-decomposition of B requires $O(n^3)$ time
- as a result, a single pass of MDS cannot operate on a large topology, particularly in the constrained computational environment of sensor networks
- classical MDS requires that D contain a distance measurement for all pairs of nodes

Multidimensional Scaling (MDS)

- MDS has some disadvantages
- the main computation of MDS, the eigen-decomposition of B requires $O(n^3)$ time
- as a result, a single pass of MDS cannot operate on a large topology, particularly in the constrained computational environment of sensor networks
- classical MDS requires that D contain a distance measurement for all pairs of nodes
- this requirement is impossible to meet with ranging hardware alone in large networks
- implementations of MDS in sensor networks must do pre-processing on measured data to generate D or use coordinate system stitching to distribute the computation

Multidimensional Scaling (MDS): Summary

1. Create the symmetric matrix $D = [d_{ij}]$ with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$

Multidimensional Scaling (MDS): Summary

1. Create the symmetric matrix $D = [d_{ij}]$ with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$
2. Create the symmetric matrix J
3. Compute B using $D^2 = [d_{ij}^2]$ and J

Multidimensional Scaling (MDS): Summary

1. Create the symmetric matrix $D = [d_{ij}]$ with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$
2. Create the symmetric matrix J
3. Compute B using $D^2 = [d_{ij}^2]$ and J
4. Take an eigen-decomposition UVU^T of B
5. Let V_d be the diagonal matrix of the d largest eigenvalues in V , where d is the desired dimensionality of the solution
6. Let U_d be the d eigenvectors from U that match the eigenvalues in V_d

Multidimensional Scaling (MDS): Summary

1. Create the symmetric matrix $D = [d_{ij}]$ with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$
2. Create the symmetric matrix J
3. Compute B using $D^2 = [d_{ij}^2]$ and J
4. Take an eigen-decomposition UVU^T of B
5. Let V_d be the diagonal matrix of the d largest eigenvalues in V , where d is the desired dimensionality of the solution
6. Let U_d be the d eigenvectors from U that match the eigenvalues in V_d
7. Compute $X_d = [X_1, X_2, \ldots, X_n]^T$ using $X_d = U_d V_d^{1/2}$. $V_d^{1/2}$ can be computed by taking the square root of each of V_d's diagonal elements

Multidimensional Scaling (MDS): Summary

1. Create the symmetric matrix $D = [d_{ij}]$ with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$
2. Create the symmetric matrix J
3. Compute B using $D^2 = [d_{ij}^2]$ and J
4. Take an eigen-decomposition UVU^T of B
5. Let V_d be the diagonal matrix of the d largest eigenvalues in V , where d is the desired dimensionality of the solution
6. Let U_d be the d eigenvectors from U that match the eigenvalues in V_d
7. Compute $X_d = [X_1, X_2, \ldots, X_n]^T$ using $X_d = U_d V_d^{1/2}$. $V_d^{1/2}$ can be computed by taking the square root of each of V_d's diagonal elements
8. (Optional) Transform the X_i's from X_d into the desired global coordinate space using some coordinate system registration algorithm, these transformed X_i's are the solution

Coordinate System Registration

Coordinate System Registration

- many localization algorithms compute a relative coordinate assignment for a group of sensors and later transform this local coordinate assignment into a different coordinate system

Coordinate System Registration

- many localization algorithms compute a relative coordinate assignment for a group of sensors and later transform this local coordinate assignment into a different coordinate system
- to do this, the algorithm must compute a translation vector, a scale factor, and an orthonormal rotation matrix that define the transformation from one coordinate system to the other

Coordinate System Registration

- many localization algorithms compute a relative coordinate assignment for a group of sensors and later transform this local coordinate assignment into a different coordinate system
- to do this, the algorithm must compute a translation vector, a scale factor, and an orthonormal rotation matrix that define the transformation from one coordinate system to the other
- the process of finding these quantities is known as Coordinate System Registration

Coordinate System Registration

- many localization algorithms compute a relative coordinate assignment for a group of sensors and later transform this local coordinate assignment into a different coordinate system
- to do this, the algorithm must compute a translation vector, a scale factor, and an orthonormal rotation matrix that define the transformation from one coordinate system to the other
- the process of finding these quantities is known as Coordinate System Registration
- registration can be performed for two dimensions as long as three points have known coordinates in both systems

Coordinate System Registration

- many localization algorithms compute a relative coordinate assignment for a group of sensors and later transform this local coordinate assignment into a different coordinate system
- to do this, the algorithm must compute a translation vector, a scale factor, and an orthonormal rotation matrix that define the transformation from one coordinate system to the other
- the process of finding these quantities is known as Coordinate System Registration
- registration can be performed for two dimensions as long as three points have known coordinates in both systems
- three dimensional version naturally requires four points

Coordinate System Registration

- Horn et al's method - many advantages over commonly used registration methods:
- it has provable optimality over the canonical least squares error metric

Coordinate System Registration

- Horn et al's method - many advantages over commonly used registration methods:
- it has provable optimality over the canonical least squares error metric
- it uses all the data available, though it can compute a correct result with as few as three (or four) points
- it can be computed quickly, since its running time is proportional to the number of common points n

Coordinate System Registration

- Horn et al's method - many advantages over commonly used registration methods:
- it has provable optimality over the canonical least squares error metric
- it uses all the data available, though it can compute a correct result with as few as three (or four) points
- it can be computed quickly, since its running time is proportional to the number of common points n
- one caveat: even after a rigid transformation, it is unlikely that the known points will precisely align, since the measurements used to localize the points are likely to have errors

Coordinate System Registration

- Horn et al's method - many advantages over commonly used registration methods:
- it has provable optimality over the canonical least squares error metric
- it uses all the data available, though it can compute a correct result with as few as three (or four) points
- it can be computed quickly, since its running time is proportional to the number of common points n
- one caveat: even after a rigid transformation, it is unlikely that the known points will precisely align, since the measurements used to localize the points are likely to have errors
- the best that can be done is a minimization of the misalignment between the two coordinate systems

Coordinate System Registration

- Let $x_{l,i}$ and $x_{r,i}$ be the known positions of node $i = 1 \dots n$ in the left hand and right hand coordinate systems respectively

Coordinate System Registration

- Let $x_{l,i}$ and $x_{r,i}$ be the known positions of node $i = 1 \ldots n$ in the left hand and right hand coordinate systems respectively
- goal of registration is to find a translation t, scale s, and rotation R that transform a point x in the left hand coordinate system to the equivalent point x' in the right hand coordinate system using formula:

$$x' = sRx + t$$

Coordinate System Registration

- Let $x_{l,i}$ and $x_{r,i}$ be the known positions of node $i = 1 \dots n$ in the left hand and right hand coordinate systems respectively
- goal of registration is to find a translation t, scale s, and rotation R that transform a point x in the left hand coordinate system to the equivalent point x' in the right hand coordinate system using formula:

$$x' = sRx + t$$

- Horn et al approach this problem using squared error; they look for a t, s, and R that meet the following condition

$$(t, s, R) = \arg\min_{t,s,R} \sum_{i=1}^{n} ||e_i||^2$$

$$e_i = x_{r,i} - sRx_{l,i} - t$$

Coordinate System Registration

- Horn et al derive a closed form for the above equation which can be computed in $O(n)$ time

Coordinate System Registration

- Horn et al derive a closed form for the above equation which can be computed in $O(n)$ time
- Step 1 - Compute the centroids of x_l and x_r

$$\bar{x}_l = \frac{1}{n}\sum_{i=1}^{n} x_{l,i} \quad \bar{x}_r = \frac{1}{n}\sum_{i=1}^{n} x_{r,i}$$

Coordinate System Registration

- Horn et al derive a closed form for the above equation which can be computed in $O(n)$ time
- Step 1 - Compute the centroids of x_l and x_r

$$\bar{x}_l = \frac{1}{n}\sum_{i=1}^{n} x_{l,i} \quad \bar{x}_r = \frac{1}{n}\sum_{i=1}^{n} x_{r,i}$$

- Step 2 - Shift the points so that they are defined with respect to the centroids:

$$x'_{l,i} = x_{l,i} - \bar{x}_l \quad x'_{r,i} - \bar{x}_r$$

the error term can be rewritten as

Coordinate System Registration

- Horn et al derive a closed form for the above equation which can be computed in $O(n)$ time
- Step 1 - Compute the centroids of x_l and x_r

$$\bar{x}_l = \frac{1}{n}\sum_{i=1}^{n} x_{l,i} \quad \bar{x}_r = \frac{1}{n}\sum_{i=1}^{n} x_{r,i}$$

- Step 2 - Shift the points so that they are defined with respect to the centroids:

$$x'_{l,i} = x_{l,i} - \bar{x}_l \quad x'_{r,i} - \bar{x}_r$$

the error term can be rewritten as

$$e_i = x'_{r,i} - sRx'_{l,i} - t' \qquad t' = t - \bar{x}_r + sR\bar{x}_l$$

the squared error is minimized when $t' = \mathbf{0}$ independent of s and R

$$t = \bar{x}_r - sR\bar{x}_l$$

$$e_i = \frac{1}{\sqrt{s}}x'_{r,i} - \sqrt{s}Rx'_{l,i}$$

Coordinate System Registration

$$
\begin{aligned}
(s, R) &= \underset{s,\,R}{\arg\min} \sum_{i=1}^{n} ||e_i||^2 \\
&= \underset{s,\,R}{\arg\min} \frac{1}{s} \sum_{i=1}^{n} ||x'_{r,i}||^2 + s \sum_{i=1}^{n} ||r_{l,i}||^2 - 2 \sum_{i=1}^{n} x'_{r,i} \cdot (R x'_{l,i})
\end{aligned}
$$

By completing the square in s it can be shown that the above equation is minimized when:

$$
s = \sqrt{\sum_{i=1}^{n} ||x'_{r,i}||^2 / \sum_{i=1}^{n} ||x'_{l,i}||^2}
$$

Coordinate System Registration

$$
\begin{aligned}
(s, R) &= \underset{s,R}{\arg\min} \sum_{i=1}^{n} ||e_i||^2 \\
&= \underset{s,R}{\arg\min} \frac{1}{s} \sum_{i=1}^{n} ||x'_{r,i}||^2 + s \sum_{i=1}^{n} ||r_{l,i}||^2 - 2 \sum_{i=1}^{n} x'_{r,i} \cdot (R x'_{l,i})
\end{aligned}
$$

By completing the square in s it can be shown that the above equation is minimized when:

$$
s = \sqrt{\sum_{i=1}^{n} ||x'_{r,i}||^2 / \sum_{i=1}^{n} ||x'_{l,i}||^2}
$$

- Step 3 Use this to compute the optimal scale factor s

$$
R = \underset{R}{\arg\min}\ 2 \left(\sqrt{\left(\sum_{i=1}^{n} ||x'_{r,i}||^2 \right) \left(\sum_{i=1}^{n} ||x'_{l,i}||^2 \right)} - \sum_{i=1}^{n} x'_{r,i} \cdot (R x'_{l,i}) \right)
$$

Coordinate System Registration

- it can be minimized when the following is true:

$$R = \underset{R}{\arg\max} \sum_{i=1}^{n} x'_{r,i} \cdot (R x'_{l,i})$$

this is the same as:

$$R = \underset{R}{\arg\max} \ \mathrm{Trace}(R^T M)$$

$$M = \sum_{i=1}^{n} x'_{r,i} (x'_{l,i})^T$$

M is a 2×2 or 3×3 matrix, depending on whether the points $x_{l,i}$ and $x_{r,i}$ are two or three dimensional. (Assume M 3×3, the results are similar for the two dimensional case.)

- Step 4 Compute M

Coordinate System Registration

- Step 5 Compute the eigen decomposition of $M^T M$. Find eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\hat{u}_1, \hat{u}_2, \hat{u}_3$ such that

$$M^T M = \lambda_1 \hat{u}_1 \hat{u}_1^T + \lambda_2 \hat{u}_2 \hat{u}_2^T + \lambda_3 \hat{u}_3 \hat{u}_3^T$$

Coordinate System Registration

- Step 5 Compute the eigen decomposition of $M^T M$. Find eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\hat{u}_1, \hat{u}_2, \hat{u}_3$ such that

$$M^T M = \lambda_1 \hat{u}_1 \hat{u}_1^T + \lambda_2 \hat{u}_2 \hat{u}_2^T + \lambda_3 \hat{u}_3 \hat{u}_3^T$$

- Step 6 Compute $S = (M^T M)^{1/2}$ and $U = MS^{-1}$

$$S = \sqrt{\lambda_1} \hat{u}_1 \hat{u}_1^T + \sqrt{\lambda_2} \hat{u}_2 \hat{u}_2^T + \sqrt{\lambda_3} \hat{u}_3 \hat{u}_3^T$$

$$U = MS^{-1} = M \left(\frac{1}{\sqrt{\lambda_1}} \hat{u}_1 \hat{u}_1^T + \frac{1}{\sqrt{\lambda_2}} \hat{u}_2 \hat{u}_2^T + \frac{1}{\sqrt{\lambda_3}} \hat{u}_3 \hat{u}_3^T \right)$$

$M = US$ and U is orthonormal; $U^T U = I$

- it can be written Trace$(R^T M)$

$$\begin{aligned} \text{Trace}(R^T US) &= \sqrt{\lambda_1}\text{Trace}(R^T U \hat{u}_1 \hat{u}_1^T) + \sqrt{\lambda_2}\text{Trace}(R^T U \hat{u}_2 \hat{u}_2^T) \\ &\quad + \sqrt{\lambda_3}\text{Trace}(R^T U \hat{u}_3 \hat{u}_3^T) \end{aligned}$$

Coordinate System Registration

- $\text{Trace}(R^T U\hat{u}_i\hat{u}_i^T)$ can be rewritten as $(R\hat{u}_i \cdot U\hat{u}_i)$

Coordinate System Registration

- $\text{Trace}(R^T U \hat{u}_i \hat{u}_i^T)$ can be rewritten as $(R\hat{u}_i \cdot U\hat{u}_i)$
- $\hat{u}_i$ is a unit vector and U, R are orthonormal transformations, $(R\hat{u}_i \cdot U\hat{u}_i \leq 1$ with equality only when $R\hat{u}_i = U\hat{u}_i$

$$\text{Trace}(R^T US) \leq \sqrt{\lambda_1} + \sqrt{\lambda_2} + \sqrt{\lambda_3} = \text{Trace}(S)$$

- the maximum value of $\text{Trace}(R^T US)$ occurs when $R^T U = I$ i.e. when $R = U$
- the rotation R necessary to minimize the error in equation of R

$$R = U = M\left(\frac{1}{\sqrt{\lambda_1}}\hat{u}_1\hat{u}_1^T + \frac{1}{\sqrt{\lambda_2}}\hat{u}_2\hat{u}_2^T + \frac{1}{\sqrt{\lambda_3}}\hat{u}_3\hat{u}_3^T\right)$$

Coordinate System Registration

- Trace$(R^T U \hat{u}_i \hat{u}_i^T)$ can be rewritten as $(R\hat{u}_i \cdot U\hat{u}_i)$
- $\hat{u}_i$ is a unit vector and U, R are orthonormal transformations, $(R\hat{u}_i \cdot U\hat{u}_i \leq 1$ with equality only when $R\hat{u}_i = U\hat{u}_i$

$$\text{Trace}(R^T US) \leq \sqrt{\lambda_1} + \sqrt{\lambda_2} + \sqrt{\lambda_3} = \text{Trace}(S)$$

- the maximum value of Trace$(R^T US)$ occurs when $R^T U = I$ i.e. when $R = U$
- the rotation R necessary to minimize the error in equation of R

$$R = U = M\left(\frac{1}{\sqrt{\lambda_1}}\hat{u}_1\hat{u}_1^T + \frac{1}{\sqrt{\lambda_2}}\hat{u}_2\hat{u}_2^T + \frac{1}{\sqrt{\lambda_3}}\hat{u}_3\hat{u}_3^T\right)$$

- Step 7 Compute R, R is an orthonormal matrix that encapsulates the rotation and possible reflection necessary to transform $x_{l,i}$ into $x_{r,i}$.

Coordinate System Registration

- Step 8 Having R and s, compute t (seen earlier equations at beginning) form a complete linear transformation between the two coordinate systems that minimizes equation (t, R, s)

Coordinate System Registration

- Step 8 Having R and s, compute t (seen earlier equations at beginning) form a complete linear transformation between the two coordinate systems that minimizes equation (t, R, s)
- Step 9 For each point x in the left hand coordinate system, compute the corresponding position x' in the right hand coordinate system using

$$x' = t + sRx$$

Coordinate System Registration

- Step 8 Having R and s, compute t (seen earlier equations at beginning) form a complete linear transformation between the two coordinate systems that minimizes equation (t, R, s)
- Step 9 For each point x in the left hand coordinate system, compute the corresponding position x' in the right hand coordinate system using

$$x' = t + sRx$$

- It is straightforward to implement, and gives provably optimal results
- many algorithms depend on coordinate system registration, either to shift a completely localized relative topology into global coordinates, or to stitch together small local topologies into a single consistent coordinate assignment

Semi-Definite Programming (SDP)

Semi-Definite Programming (SDP)

- optimization algorithm that uses techniques of linear programing
- it is a relaxation based method

Semi-Definite Programming (SDP)

- optimization algorithm that uses techniques of linear programing
- it is a relaxation based method
- trace of a given matrix A, $t_r(A)$ is the sum of the entries on the main diagonal of A

Semi-Definite Programming (SDP)

- optimization algorithm that uses techniques of linear programing
- it is a relaxation based method
- trace of a given matrix A, $t_r(A)$ is the sum of the entries on the main diagonal of A
- A symmetrical matrix is called semidefinite if all its eigenvalues are nonnegative and is presented by $A \succeq 0$

Semi-Definite Programming (SDP)

- optimization algorithm that uses techniques of linear programing
- it is a relaxation based method
- trace of a given matrix A, $t_r(A)$ is the sum of the entries on the main diagonal of A
- A symmetrical matrix is called semidefinite if all its eigenvalues are nonnegative and is presented by $A \succeq 0$
- suppose two nodes x_1 and x_2 are within radio range R of each other
- the proximity constraint can be represented as a convex second order cone constraint of the form $\| x_1 - x_2 \| \leq R$ and this can be formulated as a matrix linear inequality

Semi-Definite Programming (SDP)

- optimization algorithm that uses techniques of linear programing
- it is a relaxation based method
- trace of a given matrix A, $t_r(A)$ is the sum of the entries on the main diagonal of A
- A symmetrical matrix is called semidefinite if all its eigenvalues are nonnegative and is presented by $A \succeq 0$
- suppose two nodes x_1 and x_2 are within radio range R of each other
- the proximity constraint can be represented as a convex second order cone constraint of the form $\| x_1 - x_2 \| \leq R$ and this can be formulated as a matrix linear inequality

$$\begin{pmatrix} I_2 R & x_1 - x_2 \\ (x_1 - x_2)^T & R \end{pmatrix} \succeq 0$$

Semi-Definite Programming (SDP)

- the mathematical model:
- there are n distinct sensor points in R^{dim} whose locations are to be determined and other m fixed points (called the anchor points) whose locations are known

Semi-Definite Programming (SDP)

- the mathematical model:
- there are n distinct sensor points in R^{dim} whose locations are to be determined and other m fixed points (called the anchor points) whose locations are known
- known nodes are indicated by a and the unknown nodes by x, so that $X = [x_1, \ldots, x_n, a_1, \ldots, a_m]$

Semi-Definite Programming (SDP)

- the mathematical model:
- there are n distinct sensor points in R^{dim} whose locations are to be determined and other m fixed points (called the anchor points) whose locations are known
- known nodes are indicated by a and the unknown nodes by x, so that $X = [x_1, \ldots, x_n, a_1, \ldots, a_m]$
- all $(i, j) \in E$ where $i < j$ and if j is an anchor are denoted by N_a and unknown is denoted by N_x

Semi-Definite Programming (SDP)

- the mathematical model:
- there are n distinct sensor points in R^{dim} whose locations are to be determined and other m fixed points (called the anchor points) whose locations are known
- known nodes are indicated by a and the unknown nodes by x, so that $X = [x_1, \ldots, x_n, a_1, \ldots, a_m]$
- all $(i, j) \in E$ where $i < j$ and if j is an anchor are denoted by N_a and unknown is denoted by N_x
- the following constraints must be satisfied:

$$\| a_k - x_j \|^2 = d_{kj}^2 \quad \forall (k, j) \in N_a$$

$$\| x_i - x_j \|^2 = d_{ij}^2 \quad \forall (i, j) \in N_x$$

Semi-Definite Programming (SDP)

- when the node distances are known, let $X = [x_1; \ldots; x_n]^T$ be the matrix in $R^{\dim \times (n)}$ that needs to be determined

Semi-Definite Programming (SDP)

- when the node distances are known, let $X = [x_1; \ldots ; x_n]^T$ be the matrix in $R^{\mathrm{dim} \times (n)}$ that needs to be determined
- define $e_{ij} \in R^n$ with 1 on i th position and with -1 on j th position and everywhere else zeros; I_{dim} is the identity matrix, the constraints can be written:

$$(a_k; e_j)^T [I_{\mathrm{dim}}; X^T][I_{\mathrm{dim}}; X](a_k; e_j) = d_{kj}^2 \; \forall (k,j) \in N_a$$

$$e_{ij}^T X^T X e_{ij} = d_{ij}^2 \; \forall (i,j) \in N_x$$

Semi-Definite Programming (SDP)

- when the node distances are known, let $X = [x_1; \dots; x_n]^T$ be the matrix in $R^{\dim\times(n)}$ that needs to be determined
- define $e_{ij} \in R^n$ with 1 on i th position and with -1 on j th position and everywhere else zeros; $I_{\dim}$ is the identity matrix, the constraints can be written:

$$(a_k; e_j)^T [I_{\dim}; X^T][I_{\dim}; X](a_k; e_j) = d_{kj}^2 \ \forall (k,j) \in N_a$$

$$e_{ij}^T X^T X e_{ij} = d_{ij}^2 \ \forall (i,j) \in N_x$$

- need to find a symmetric matrix $Y \in R^{\dim\times\dim}$ and X that satisfy the following constraints:

$$(a_k; e_j)^T \begin{pmatrix} I_{\dim} & X \\ X^T & Y \end{pmatrix} (a_k; e_j) = d_{kj}^2 \ \forall (k,j) \in N_a$$

Semi-Definite Programming (SDP)

$$e_{ij}^T Y e_{ij} = d_{ij}^2 \;\; \forall (i,j) \in N_x$$

$$Y = X^T X$$

Semi-Definite Programming (SDP)

$$e_{ij}^T Y e_{ij} = d_{ij}^2 \;\; \forall (i,j) \in N_x$$
$$Y = X^T X$$

- this is the SDP formulation of the problem of WSN localization
- the constraint
$$Y = X^T X$$
is relaxed with $Y \succeq X^T X$
- this condition can be written as
$$Z = \begin{pmatrix} I_{\mathrm{dim}} & X \\ X^T & Y \end{pmatrix} \succeq 0$$

Semi-Definite Programming (SDP)

Semi-Definite Programming (SDP)

- m known points (anchors) $a_k \in \mathcal{R}^2, k = 1, \ldots, m$ and n unknown points (sensors) $x_j \in \mathcal{R}^2, j = 1, \ldots, n$

Semi-Definite Programming (SDP)

- m known points (anchors) $a_k \in \mathcal{R}^2, k = 1, \ldots, m$ and n unknown points (sensors) $x_j \in \mathcal{R}^2, j = 1, \ldots, n$
- for a pair of two points in N_e, Euclidean distance measurement $\hat{d}_{kj}$ between a_k and x_j or $\hat{d}_{ij}$ between x_i and x_j

Semi-Definite Programming (SDP)

- m known points (anchors) $a_k \in \mathcal{R}^2, k = 1, \ldots, m$ and n unknown points (sensors) $x_j \in \mathcal{R}^2, j = 1, \ldots, n$
- for a pair of two points in N_e, Euclidean distance measurement $\hat{d}_{kj}$ between a_k and x_j or $\hat{d}_{ij}$ between x_i and x_j
- for a pair of two points in N_l, a distance lower bound $\underline{r}_{kj}$ between a_k and x_j or $\underline{r}_{ij}$ between x_i and x_j

Semi-Definite Programming (SDP)

- m known points (anchors) $a_k \in \mathcal{R}^2, k = 1, \ldots, m$ and n unknown points (sensors) $x_j \in \mathcal{R}^2, j = 1, \ldots, n$
- for a pair of two points in N_e, Euclidean distance measurement $\hat{d}_{kj}$ between a_k and x_j or $\hat{d}_{ij}$ between x_i and x_j
- for a pair of two points in N_l, a distance lower bound $\underline{r}_{kj}$ between a_k and x_j or $\underline{r}_{ij}$ between x_i and x_j
- for a pair of two points in N_u, a distance upper bound $\bar{r}_{kj}$ between a_k and x_j or $\bar{r}_{ij}$ between x_i and x_j

Semi-Definite Programming (SDP)

- m known points (anchors) $a_k \in \mathcal{R}^2, k = 1, \ldots, m$ and n unknown points (sensors) $x_j \in \mathcal{R}^2, j = 1, \ldots, n$
- for a pair of two points in N_e, Euclidean distance measurement $\hat{d}_{kj}$ between a_k and x_j or $\hat{d}_{ij}$ between x_i and x_j
- for a pair of two points in N_l, a distance lower bound $\underline{r}_{kj}$ between a_k and x_j or $\underline{r}_{ij}$ between x_i and x_j
- for a pair of two points in N_u, a distance upper bound $\bar{r}_{kj}$ between a_k and x_j or $\bar{r}_{ij}$ between x_i and x_j
- the localization problem is to find x_js such that

$$||x_i - x_j||^2 = (\hat{d}_{ij})^2, ||a_k - x_j||^2 = (\hat{d}_{kj})^2, \forall (i,j), (k,j) \in N_e$$
$$||x_i - x_j||^2 \geq (\underline{r}_{ij})^2, ||a_k - x_j||^2 \geq (\underline{r}_{kj})^2, \forall (i,j), (k,j) \in N_l$$
$$||x_i - x_j||^2 \leq (\bar{r}_{ij})^2, ||a_k - x_j||^2 \leq (\bar{r}_{kj})^2, \forall (i,j), (k,j) \in N_u$$

Semi-Definite Programming (SDP)

- due to measurement error x_js can be chosen to minimize the sum of errors:

Semi-Definite Programming (SDP)

- due to measurement error x_js can be chosen to minimize the sum of errors:

$$\min \sum_{i,j\in N_e, i<j} | \| x_i - x_j \|^2 - (\hat{d}_{ij})^2 | + \sum_{k,j\in N_e} | \| a_k - x_j \|^2 - (\hat{d}_{kj})^2 |$$

$$+ \sum_{i,j\in N_l, i<j} \left(\| x_i - x_j \|^2 - (\underline{r}_{ij})^2 \right)_- + \sum_{k,j\in N_l} \left(\| a_k - x_j \|^2 - (\underline{r}_{kj})^2 \right)_-$$

$$+ \sum_{i,j\in N_u, i<j} \left(\| x_i - x_j \|^2 - (\bar{r}_{ij})^2 \right)_+ + \sum_{k,j\in N_u} \left(\| a_k - x_j \|^2 - (\bar{r}_{kj})^2 \right)_+$$

- $(u)_-$ and $(u)_+$ defined as $(u)_- = \max\{0, -u\}$ and $(u)_+ = \max\{0, u\}$
- by introducing slack variables and relaxing $Y = X^T X$ to $Y \succeq X^T X$, the problem can be rewritten as a standard SDP problem

Thank You