Algorithms for Radio Networks

Localization
Trilateration

- Assuming the distance to three points is given
- System of equations
  - \((x_i, y_i)\): coordinates of an anchor point \(i\),
  - \(r\) distance from the anchor point \(i\)
  - \((x_u, y_u)\): unknown coordinates of a node

\[
(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \quad \text{for} \ i = 1, 2, 3
\]

- Problem: Quadratic equations
  - Transformations lead to a linear system of equations
Trilateration

- System of equations
  
  \[(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \text{ for } i = 1, \ldots, 3\]

- Transformation

  \[
  \begin{align*}
  (x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 &= r_1^2 - r_3^2 \\
  (x_2 - x_u)^2 - (x_3 - x_u)^2 + (y_2 - y_u)^2 - (y_3 - y_u)^2 &= r_2^2 - r_3^2
  \end{align*}
  \]

  - results in:

  \[
  \begin{align*}
  2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u &= (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\
  2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u &= (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)
  \end{align*}
  \]
Trilateration as a Linear System of Equations

- Forming a system of equations

\[
2 \begin{bmatrix}
  x_3 - x_1 & y_3 - y_1 \\
  x_3 - x_2 & y_3 - y_2 
\end{bmatrix}
\begin{bmatrix}
  x_u \\
  y_u 
\end{bmatrix}
= \begin{bmatrix}
  (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\
  (r_2^2 - r_2^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)
\end{bmatrix}
\]

- Example:
  - \((x_1, y_1) = (2,1), (x_2, y_2) = (5,4), (x_3, y_3) = (8,2),\)
  - \(r_1 = 10^{1/2}, r_2 = 2, r_3 = 3\)

\[
2 \begin{bmatrix}
  6 & 1 \\
  3 & -2 
\end{bmatrix}
\begin{bmatrix}
  x_u \\
  y_u 
\end{bmatrix}
= \begin{bmatrix}
  64 \\
  22 
\end{bmatrix}
\]

\(\Rightarrow (x_u, y_u) = (5,2)\)
Trilateration as a Linear System of Equations

- In three dimensions
  - Intersection of four spheres

\[
\begin{bmatrix}
(d_1^2 - d_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2) \\
(d_2^2 - d_4^2) - (x_2^2 - x_4^2) - (y_2^2 - y_4^2) - (z_2^2 - z_4^2) \\
(d_3^2 - d_4^2) - (x_3^2 - x_4^2) - (y_3^2 - y_4^2) - (z_3^2 - z_4^2)
\end{bmatrix}
\begin{bmatrix}
\vec{b}
\end{bmatrix}
= 2
\begin{bmatrix}
(x_4 - x_1)(y_4 - y_1)(z_4 - z_1) \\
(x_4 - x_2)(y_4 - y_2)(z_4 - z_2) \\
(x_4 - x_3)(y_4 - y_3)(z_4 - z_3)
\end{bmatrix}
\begin{bmatrix}
x_{P1} \\
y_{P1} \\
z_{P1}
\end{bmatrix}
\]

\[A = \begin{bmatrix}
(d_1^2 - d_4^2) & (x_1^2 - x_4^2) & (y_1^2 - y_4^2) & (z_1^2 - z_4^2) \\
(d_2^2 - d_4^2) & (x_2^2 - x_4^2) & (y_2^2 - y_4^2) & (z_2^2 - z_4^2) \\
(d_3^2 - d_4^2) & (x_3^2 - x_4^2) & (y_3^2 - y_4^2) & (z_3^2 - z_4^2)
\end{bmatrix}
\]

- Solve \(Ax = b\) \(\Rightarrow x = A^{-1}b\)
Trilateration

- In case of measurement errors

- Averaging: e.g. centroid of triangle

[F. Höflinger, 2013]
Trilateration

› Measurement errors
  • Small distance errors can lead to large position errors

› flip ambiguity from noise
Multilateration with \textit{absolute} distances

\begin{itemize}
  \item Multilateration (absolute distances): Calculate the intersection of \textit{at least four} distance measurements
  \begin{itemize}
    \item May be over-determined equation system: More equations than variables
    \item “No solution” in case of measurement errors
  \end{itemize}
  \item Minimize sum of quadratic residuals: \textit{Least squares}
  \item Vector notation
    \begin{itemize}
      \item Solve $(A^T A)x = A^T b \implies x = (A^T A)^{-1} A^T b$
      \item Matrix inverse by Gauss-Jordan elimination
    \end{itemize}
\end{itemize}
Multilateration with relative distances

- Multilateration (relative): Calculate the intersection of relative distance measurements
  - Emission time $e$ unknown!
  - Measure only reception times $T_i$, $i = 1, ..., n$
  - System of equations $T_i = e + \| r_i - s \| / c$
  - ...for a signal traveling from $s$ to receivers $r_i$

- Subtract two absolute times $T_i$ and $T_j$:
  - $T_i - T_j = \| r_i - s \| / c - \| r_j - s \| / c =: \Delta t$  $(i, j = 1, ..., n)$
  - System of hyperbolic equations
  - Time Difference of Arrival $\Delta t$, relative distance $\Delta d = c \Delta t$
Multilateration with *relative* distances

- Advantages
  - No cooperation of signal emitter
  - Hardware delays cancel out (both emitter and receiver)
  - Passive localization / natural signal sources
- Disadvantages
  - Larger number of unknown values: Position and time
  - Synchronization still (usually) required
Anchor-free localization

“Anchor-free localization”:
- Hyperbolic multilateration
- Unknown emitters $s_j$, and unknown receivers $r_i$

Advantages:
- No need to measure receiver positions
- Self-positioning by passive information from the surroundings

Disadvantages:
- Even larger number of unknown variables
Anchor-free localization

- For absolute distances $d_{ik}$:
  - **Solve** $\|r_i - s_k\| = d_{ik}$ $(i, j = 1, \ldots, n; \ k = 1, \ldots, m)$
  - Problem of intersecting circles / spheres
  - Bipartite distance graph: $G = (\{r_i\}, \{s_k\}, \{d(i, k)\})$
  - **Minimum case closed-from solutions known** [Kuang, et al., ICASSP 2013]
Anchor-free localization

- For relative distances $\Delta d_{ijk} = d_{ik} - d_{jk}$:
  - Solve $\| r_i - s_k \| - \| r_j - s_k \| = \Delta d_{ijk}$
  - Problem of intersecting hyperbolas / hyperboloids
  - Closed-form solutions only for larger problem sets
    - [Pollefeys and Nister, ICASSP 2008], [Kuang and Åström, EUSIPCO 2013]
  - Minimum problem set: Iterative/recursive approximations
    - [Wendeberg and Schindelhauer, Algosensors 2012]
Anchor-free localization

 Degrees of freedom

\[ T_{ik} = e_{ik} + \| r_i - s_k \| / c \]
\[ (e_{ik}, r_i, s_k \text{ unknown}) \]

\[ G_{2D} = 2n + 3m - nm - 3 \]
\[ G_{3D} = 3n + 4m - nm - 6 \]
Anchor-free localization

- Minimum cases

<table>
<thead>
<tr>
<th></th>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>general setting</td>
<td>4 / 6</td>
<td>5 / 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 / 7</td>
</tr>
<tr>
<td>far-field setting</td>
<td>3 / 3 (sync.)</td>
<td>4 / 6 (sync.)</td>
</tr>
<tr>
<td></td>
<td>3 / 5 (unsync.)</td>
<td>4 / 9 (unsync.)</td>
</tr>
</tbody>
</table>

Minimum number of required receivers / emitters
Anchor-free localization

- Strategies:
  1. Estimate receiver topology from known information
  2. Assume large number of emitters and receivers
  3. Assume specific distribution of emitters and receivers
  4. Heat the CPU: Optimization, branch-and-bound search, ...
(1.) Topology: Hull element

• “The receiver which receives the last timestamp is an element of the convex hull”

\[
n_0 = \frac{n}{\|n\|}
\]

If exists \( i \) such that for all \( k \): \( T_i \geq T_k \), then holds:

\[
(m_i - s)^T n_0 = \|m_i - s\| \geq \|m_k - s\| \geq (m_i - s)^T n_0
\]
Anchor-free localization

- (2.) Large number of signals: Statistical assumptions
  [Schindelhauer, et al., SIROCCO 2011]
  - Lemma: Many signals occur from the long side of any two receivers.
  - Estimate the distance: \( d \sim \frac{c}{2} \left( \Delta t_{\text{max}} - \Delta t_{\text{min}} \right) \)
Anchor-free localization

- (3.) Assume that signals occur from far away:
  - “far-field assumption”, linear frontier of signal propagation
- The Ellipsoid TDoA Method [Wendeberg, et al., TCS, 2012]
  - Time differences of three receivers form an ellipse
Algorithms for Radio Networks

Localization

University of Freiburg
Technical Faculty
Computer Networks and Telematics
Prof. Christian Schindelhauer
Pythagoras

A_1

A_2

A_3

(x, y)

x(A_1), y(A_2)

d_1

d_2

d_3

1
general position
- no 3 points on a line
- no 4 points on a circle

# 2

\[
\frac{0}{>0} = 0
\]
\[ \vec{I} - \vec{Y} = A \]
\[ \vec{Y} - \vec{W} = B \]
\[ \vec{I} - \vec{W} = C \]

\[ C = A + B \]

\[ d^2 = x^2 + z^2 \]
\[ d^2 + y^2 = v^2 \]
\[ x^2 + y^2 + z^2 = v^2 \]

You \[ \Delta \] blue ... \[ k_u^2 \ldots y_v^2 \ldots z_w^2 \ldots \]

\[ k_u^2 \ldots y_v^2 \ldots z_w^2 \ldots \]
\[
\frac{d (x - r_1^1)^2}{dx} = 2x - 2r_1^1
\]
\[
A^T \begin{bmatrix}
1 & 2 \\
3 & 4 \\
5 & 6 \\
7 & 8 \\
9 & 10 \\
\end{bmatrix} \cdot \begin{bmatrix}
x \\
y \\
\end{bmatrix} = A^T \begin{bmatrix}
v_1 \\
v_2 \\
v_3 \\
v_4 \\
v_5 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
21 & 1003 \\
32 & 802 \\
\end{bmatrix} \begin{bmatrix}
x \\
y \\
\end{bmatrix} = \begin{bmatrix}
192 \\
181 \\
\end{bmatrix}
\]
\[ r_i = \text{dis} \cdot c = (\overrightarrow{r_2} - \overrightarrow{r_1}) \cdot c \]

\[ c = 3 \times 10^8 \, \frac{m}{s} = \frac{d}{t} \]
\[ v_2 = v_1 + \Delta \]
\[ \Delta = v_2 - v_1 \]
\[ \Delta = \sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x-x_2)^2 + (y-y_2)^2} \]
\[ y^2 - x^2 = a \]
\[ m : \# \text{rain} \]

\[ 2 \cdot n + 2 \cdot m + n \]

\[ x/x \times x/y \quad \text{time} \]

\[ -2 - 1 \]

\[ \text{\#equations} : m \text{ or m} \]