Abstract

• This paper proposes a robust tracking mobile targets framework using unreliable node sequences

• Without assumption of movement pattern or noise model

• Without accurate range-based localization

• Tracking is modeled as an optimal path matching problem in a graph

Outline

• Abstract
• System Overview
• Basic System Design
• Multi-dimensional Smoothing
• Discussion
• Simulation Evaluation
• System Evaluation
• Conclusion

Abstract

• Specific format of the physical sensing modality (e.g. RF radiation, acoustic, ...) is irrelevant to the tracking algorithm

• Design is evaluated with
  —Simulation
  —A system implementation using Pioneer III Robot and MICAz sensor nodes
Outline

• Abstract
• System Overview
• Basic System Design
• Multi-dimensional Smoothing
• Discussion
• Simulation Evaluation
• System Evaluation
• Conclusion

Basic System Design

• Division of the Map
• Unreliable Detection Node Sequence
• Sequence Distance
• Neighborhood Graph
• Tracking as Optimal Path Matching
Division of the Map

(a) Map Dividing Example 1

\[ f_i : S_{f_i} = (2, 1) \]

\[ f_i : S_{f_i} = (1, 2) \]

Sensor Node 1

(b) Map Dividing Example 2

With \( n \) sensor nodes, there are

\[ C_n^2 = \frac{n \cdot (n-1)}{2} \]

Perpendicular bisector lines, which divide the whole map into \( O(n^4) \) faces

Unreliable Detection Node Sequence

- In ideal case, a detection sequences \( S_d \) should be identical with one of the face signatures

- However, in a real system, sensing at each sensor node could be irregular and affected by many factors
  - including environment noise, obstacles and etc

- \( S_d \) is unreliable, which could be either
  - a full detection sequence including all the related sensor nodes
  - or a partial detection sequence, in which some of the nodes supposed to appear are missing

- In addition, nodes in \( S_d \) could get flipped due to noisy sensing

Detection Sequences v.s. Face Sequences

Detection Sequence vs. Face Sequence

Kendall Tau Distance (KT distance)

<table>
<thead>
<tr>
<th>Pair</th>
<th>Detection Sequence</th>
<th>Signature Sequence</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A,B)</td>
<td>1 -&gt; 2</td>
<td>1 -&gt; 2</td>
<td>X</td>
</tr>
<tr>
<td>(A,C)</td>
<td>1 -&gt; 3</td>
<td>3 -&gt; 1</td>
<td>X</td>
</tr>
<tr>
<td>(A,D)</td>
<td>1 -&gt; 5</td>
<td>1 -&gt; 5</td>
<td>X</td>
</tr>
<tr>
<td>(B,C)</td>
<td>2 -&gt; 3</td>
<td>3 -&gt; 2</td>
<td>X</td>
</tr>
<tr>
<td>(B,D)</td>
<td>2 -&gt; 4</td>
<td>4 -&gt; 2</td>
<td>X</td>
</tr>
<tr>
<td>(B,E)</td>
<td>2 -&gt; 5</td>
<td>5 -&gt; 2</td>
<td>X</td>
</tr>
<tr>
<td>(C,D)</td>
<td>3 -&gt; 4</td>
<td>4 -&gt; 3</td>
<td>X</td>
</tr>
<tr>
<td>(C,E)</td>
<td>3 -&gt; 5</td>
<td>5 -&gt; 3</td>
<td>X</td>
</tr>
<tr>
<td>(D,E)</td>
<td>4 -&gt; 5</td>
<td>5 -&gt; 4</td>
<td>X</td>
</tr>
</tbody>
</table>

The Kendall tau distance is 4.

\[ X \]

Sequence Distance

Sequence Distance Example

\[ S_{d1} = (1, 2), S_{d2} = (2, 1), \]

\[ S_{d3} = (2, 3), S_{d4} = (3, 1) \]

\[ SD(S_{d1}, S_{d2}) = 2 \]
The Insight of Sequence Distance

Sequence Distance vs. Geographic Distance

Extended KT Distance Algorithm

EKT Distance Example

Neighborhood Graph

\[ \Delta X_{\text{max}} = V_{\text{max}} \cdot \Delta T \]

Neighborhood Graph Building

Neighborhood Graphs with Randomly Deployed 4, 8, 12 and 16 Sensor Nodes

Tracking as Optimal Path Matching

Given a series of detection sequences \( S_d(k), k = 0, 1, \ldots, M \), a path composed of faces \( f(k) \) with minimal accumulated EKT distance to \( S_d(k) \) owns maximal overall likelihood. The tracking problem turns into an optimal path matching issue:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=0}^{M} \text{EKT}(S_d(k), S_f(k)) \\
\text{subject to} & \quad f(k) \in V(G) \\
& \quad \forall k, \text{edge}(f(k), f(k+1)) \in E(G)
\end{align*}
\]

Tracking as Optimal Path Matching

From Optimal Path Matching to Shortest Path Searching
Algorithm

Algorithm 1: Optimal Path Masking

Input: Detection sequences $S_k [y]$, $k = 1, \ldots , M$
Neighborhood graph $G$
Output: Optimal path $P$

1. $P(1) = $ Initialization ($S_1 [y]$)
2. repeat
3. $P(k), f a c e = $ Neighbor ($P(k-1), f a c e, E K T)$
4. end repeat
5. $P = \text{Unprocessed} (H(b), f a c e)$
6. $d(i, j) = E K T(S_i [y], S_j [y])$
7. $P = \text{Min}(H(k-1), f a c e, s o u r c e)$
8. $T o o k = P, \text{Face}, s o u r c e = \text{Face}$
9. until all faces in $H(b)$ are processed
10. until $k = M$
11. $P = \text{TrackBack} (\text{Minimum}(H(M), \text{Face}, \text{source}))$

Time complexity: $O(M \cdot n^2 \cdot \log(n))$
Storage Complexity: $O(M \cdot n^2)$

Outline

- Abstract
- System Overview
- Basic System Design
  - Multi-dimensional Smoothing
  - Discussion
  - Simulation Evaluation
  - System Evaluation
  - Conclusion

Multi-dimensional Smoothing

- Modality Domain Smoothing
- Time Domain Smoothing
- Space Domain Smoothing

Modality Domain Smoothing

Integrate sensing results from diverse modalities

Multi-Modality Integration

Time Domain Smoothing

- Time domain smoothing over continuous detection results is commonly used for filtering out random noise in many systems

- Average the EKT distance along the timeline over a smoothing window with odd length $L$

$$\overline{D_f}(k) = \frac{\sum_{i=-\frac{(L-1)}{2}}^{\frac{(L-1)}{2}} D_f(k+i)}{L}$$

Space Domain Smoothing

- Maps the position of the mobile target at each time instance to the center of gravity point of a face in the map

- Using a smoothing window with odd length $L'$

$$\overline{x}_k = \frac{\sum_{i=-\frac{(L'-1)}{2}}^{\frac{(L'-1)}{2}} x_{k+i}}{L'} \quad \overline{y}_k = \frac{\sum_{i=-\frac{(L'-1)}{2}}^{\frac{(L'-1)}{2}} y_{k+i}}{L'}$$

$(\overline{x}_k, \overline{y}_k)$ are the coordinates of the center of gravity of a face
$(\overline{x}_{k-1}, \overline{y}_{k-1})$ are the final estimated position after space domain smoothing
Discussion

- System Scalability and Multiple Targets
- Time Synchronization and Energy Efficiency

System Scalability and Multiple Targets

![Diagram of System Scalability and Multiple Targets]

Reduced Candidate Path Graph H

![Graph H with annotations]

Time Synchronization and Energy Efficiency

- Current time synchronization techniques can achieve microsecond level accuracy
  - Flooding Time Synchronization Protocol
  - The time interval between two samples varies in microsecond unit
- Most of the time, sensor nodes keep a low duty cycle until some event or target appears in the monitored area
  - Nodes near the target can adjust sampling rate according to real-time tracking results
  - Other nodes remain in sleep

Outline

- Abstract
- System Overview
- Basic System Design
- Multi-dimensional Smoothing
- Discussion
- Simulation Evaluation
- System Evaluation
- Conclusion
Simulation Setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Area</td>
<td>100 meters x 100 meters</td>
</tr>
<tr>
<td>Noise Model</td>
<td>Logarithmic (3 = 4, log ... 0)</td>
</tr>
<tr>
<td>Number of Sensor Nodes</td>
<td>10, randomly deployed with uniform distribution</td>
</tr>
<tr>
<td>Sensing Sampling Rate</td>
<td>100 Hz</td>
</tr>
<tr>
<td>Impact Velocity</td>
<td>Random between 1 ~ 5 (metres/s)</td>
</tr>
<tr>
<td>Averaging Window</td>
<td>2.5s (Time Domain), 99 points (Space Domain)</td>
</tr>
</tbody>
</table>

Noise Models

- linear delay noise model
  \[ S_i(k) \propto \frac{1}{(1 + \alpha) \cdot d_i(k)} , \quad \alpha \sim N(0, \sigma^2_\alpha) \]

- logarithmic attenuation noise model
  \[ S_i(k) \propto -10 \beta \log \left( \frac{d_i(k)}{d_0} \right) + X_i(k) \]
  \[ d_0 = 1 \quad \text{and} \quad X_i(k) \sim N(0, \sigma^2_X) \]

An Example by Figures

Smoothed Result

Linear Noise Model

Logarithmic Noise Model: \( \sigma_X \)
Logarithmic Noise Model: $\beta$

Number of Sensor Nodes

Number of Starting Faces

Effectiveness of Smoothing

Outline

- Abstract
- System Overview
- Basic System Design
- Multi-dimensional Smoothing
- Discussion
- Simulation Evaluation
  - System Evaluation
- Conclusion

System Evaluation
Robot Tracking Results

Outline

• Abstract
• System Overview
• Basic System Design
• Multi-dimensional Smoothing
• Discussion
• Simulation Evaluation
• System Evaluation
• Conclusion

Conclusion

• This paper presented the first work for mobile target tracking using unreliable node sequences in wireless sensor networks

• Tracking is modeled as an optimal path matching problem in a graph

• Beside the basic design, multi-dimensional smoothing is proposed for further enhancing system accuracy

Thank you very much for your attention!