Introduction & Concepts of Ubiquitous Tracking

By Troels Frimor

3rd Joint Advanced Student School
JASS 2005

Feel free to ask questions
and interrupt if something is not clear.

Roadmap

- Very short reminders on concepts
  - Augmented Reality
  - Tracking Devices
  - Ubiquitous Computing
- Ubiquitous Tracking
  - Motivation
  - Theoretical Framework
    - Relationship graphs
    - Data flow graphs
    - Super nodes
    - Example

Augmented Reality

- Characteristics of Augmented Reality
  - Combines real and virtual
  - Interactive in real time
  - Registered in 3D

Augmented Reality

Courtesy: TUM
http://wwwbruegge.in.tum.de/DWARF/ProjectSheep

Courtesy: MEDARPA http://www.medarpa.de/

Augmented Reality

- Characteristics of Augmented Reality
  - Combines real and virtual
  - Interactive in real time
  - Registered in 3D
- Many technical issues
  - Domain specific
    - Medical, maintenance, production, design, …
  - Visualization
  - User interfaces
  - Tracking
  - …

Tracking Devices

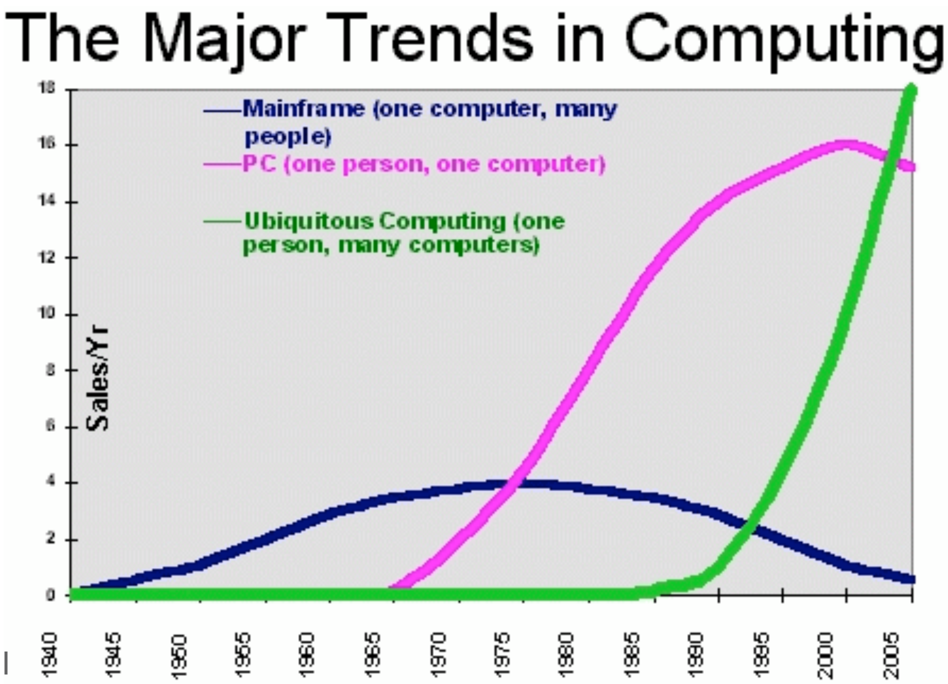
- Many different ways of tracking exists
  - All have different characteristics
    - Physical medium – mechanical, inertia, field sensing, …
    - Measurement – 2D pixel value, time of flight, …
    - Accuracy – level of noise
    - Update rate – time between each measurement
    - Mobile – is it user attached, weight, fashion, …
    - …

Tracking Devices

- Hybrid systems
  - Combine trackers to take advantages of their individual characteristics
  - Optic – inertial
  - Inside-out – outside-in
  - …
- Domain issues have to be considered
  - Magnetism in factories (pipes of iron/steel)
  - Is it possible to make sterile for a medical environment
  - Light pollution in infrared sensitive environments

Ubiquitous Computing

- Ubiquitous Computing
  - It is invisible, everywhere computing that does not live on a personal device of any sort, but is in the woodwork everywhere. - Mark Weiser



Courtesy: Mark Weiser
http://www.ubiq.com/hypertext/weiser/UbiHome.html

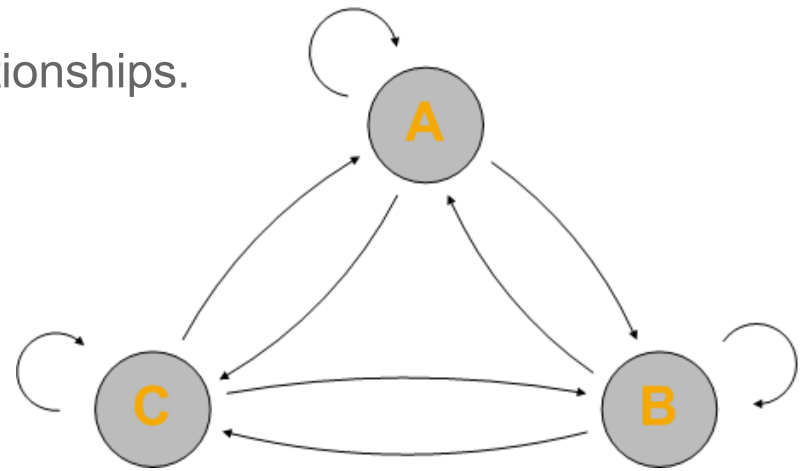
Ubiquitous Tracking

- Motivation:
  - All tracking devices have
    - limitations in range
    - very different attributes
  - Assuming an ubiquitous computing environment
    - Many trackers distributed out to help providing a wide-area augmented reality support
  - Each setup has to reinvent the wheel by creating a system for combining all the sensor data

- Goal:
  - To provide, at any point in time, an optimal estimate of the spatial relationship between any arbitrary objects
  - Make a separating layer between application and sensors

Fundamentals of Ubiquitous Tracking

- Ubiquitous Tracking framework.
  - Representing Spatial Relationships
    - What information is measured
  - Graph based model of spatial relationships.
  - Optimal path
  - Attributes – tracking quality
  - Error functions
- Optimizations
  - Data flow graphs
  - Super nodes



Representing Spatial Relationships

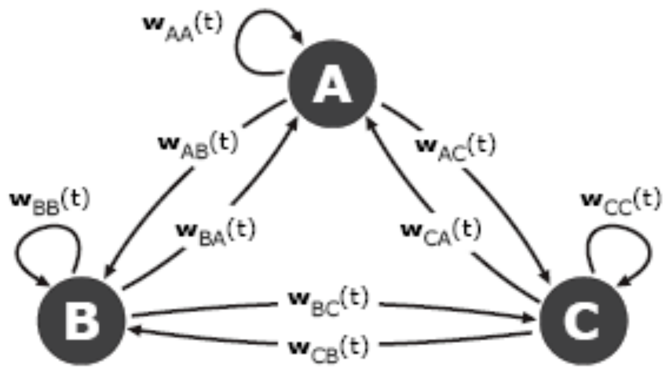
- Different sensors often measure properties other than pose
  - position or orientation only, 2D pixel values, acceleration
  - These can be related to a pose indirectly
- For the purpose of this presentation an 4x4 matrix is assumed for relations

Spatial Relationships

- Real
  - The actual relationships. Continuous over time.
- Measured
  - Every time a sensor provides a measurement, a new relation is added.
- Inferred
  - Relations are added by combining knowledge from measurements or using external information

Real Relationships

- All information from the view of an all-knowing observer
- Representation



$$N = \{A, B, C, \ldots\}, \text{objects}$$

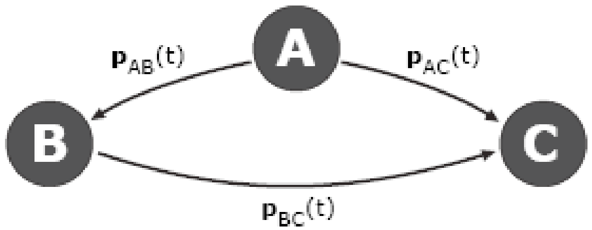
$$W : (\Omega = N \times N) \rightarrow w, \text{ where } w : D_t \rightarrow R^{4\times 4}$$

  - *w* is defined at all points in time

Measured Relationships

- Measurements made
  - Noise
  - Not all relations are known

- Representation



$N = \{A, B, C, \ldots\}$, objects

A, set of attributes

$P : (\Phi \subseteq N \times N) \rightarrow p$, where $p : D_t \rightarrow R^{4\times 4} \times A$

  - *p* is a function on a discrete set of points in time

Inferred Relationships

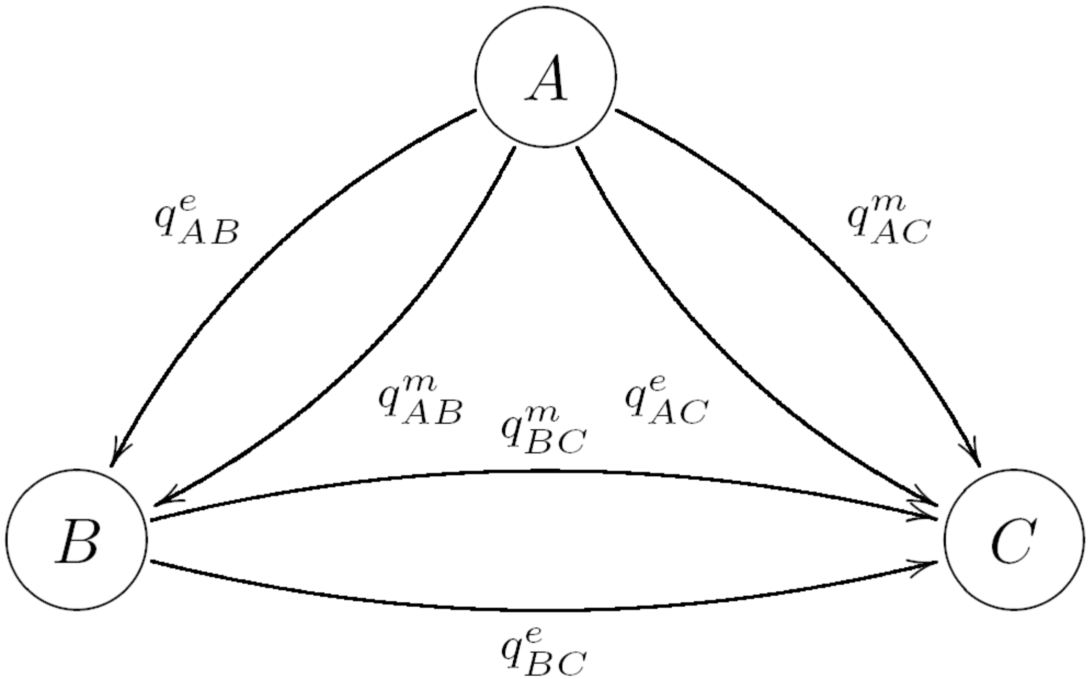
- Try to approximate Ω (real world), by adding *inferred* relationships

  $$Q : (\Psi \subseteq N \times N) \rightarrow q, \text{ where } q : D_t \rightarrow R^{4\times4} \times A$$

  - q_{XY}, is a union of basis relationships functions between X and Y and contains
    - The measurement relationships $q^m_{XY}(t) = p_{XY}(t)$
    - All inferred relationships $q^e_{XY}(t),\quad q^i_{XY}(t),\quad q^k_{XY}(t), \ldots$

Inferred Relationships – graph

- $G(\Psi)$, is the graph for the inferred relationships
  - Drawn with each element of *q* as an edge



Inferred Relationships – examples

- Measurements $$q^m_{XY}(t) = p_{XY}(t) = \begin{cases} (H_1, A_1) & \text{if } t = t_1 \\ (H_2, A_2) & \text{if } t = t_2 \\ undefined & \text{otherwise} \end{cases}$$

- Extrapolation $$q^e_{XY}(t) = \begin{cases} (H_1, A_1) & \text{if } |t - t_1| < |t - t_2| \\ (H_2, A_2) & \text{otherwise} \end{cases}$$

- Interpolation $$q^i_{XY}(t) = \begin{cases} (H_1, A_1) & \text{if } t_1 \leq t \leq t_1 + \frac{t_2 - t_1}{2} \\ (H_2, A_2) & \text{if } t_1 + \frac{t_2 - t_1}{2} < t \leq t_2 \\ undefined & \text{otherwise} \end{cases}$$

Inferred Relationships – examples

- General inference $q^{f}_{XY}(t) = f((H_1, A_1), (H_2, A_2), \ldots, t)$

- This is also used to model different filters like the Kalman filter. More on this in a later presentation.

Quick sum up on SR Graphs

- Real relationships
  - Defined at all points in time

$$\mathrm{W} : (\Omega = N \times N) \rightarrow \mathrm{w}, \quad \text{where} \quad \mathrm{w} : D_t \rightarrow \mathrm{R}^{4\times4}$$

- Measured relationships
  - Each measurement at a discrete point in time

$$\mathrm{P} : (\Phi \subseteq N \times N) \rightarrow \mathrm{p}, \quad \text{where} \quad \mathrm{p} : D_t \rightarrow \mathrm{R}^{4\times4} \times \mathrm{A}$$

- Inferred relationships
  - Deduced knowledge gives an approximation to real world

$$\mathrm{Q} : (\Psi \subseteq N \times N) \rightarrow \mathrm{q}, \quad \text{where} \quad \mathrm{q} : D_t \rightarrow \mathrm{R}^{4\times4} \times \mathrm{A}$$

Querying for Optimal Paths

- An error function is used to estimate the quality of a path

$$e : A \rightarrow \mathbb{R}$$

  - The error function can be designed to value, for example, latency over frequency or spatial accuracy.

    (more on attributes later)

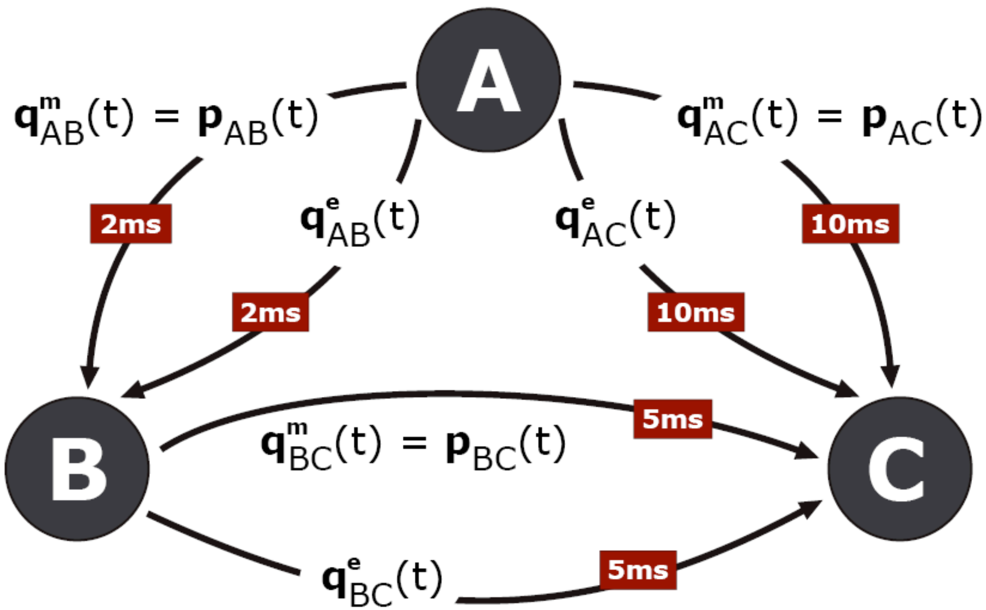
Querying for Optimal Paths

- Query for optimal path from *A* to *B*
  - Extend error function to take a ordered set of attributes

$$e : A^* \rightarrow \mathbb{R}$$

  - Algorithm – given graph $G(\Psi)$, objects *A* and *B,* error function *e* and time *t*
    - Find all paths *R* from *A* to *B* that are defined at given time
    - Evaluate given error function on collected attributes for every path $r \in R$
    - Calculate the spatial relationship between *A* and *B* by multiplying all matrices along the path *r*
  - If the result is a new inference then add it to Q

Querying example



$$\mathbf{P1}: A \xrightarrow{q^e_{AB}} B \xrightarrow{q^e_{BC}} C, \qquad \mathbf{P2}: A \xrightarrow{q^m_{AC}} C, \qquad \mathbf{P3}: A \xrightarrow{q^e_{AC}} C$$

Attributes for Relation Quality

- Many different attributes can be useful, examples:
  - Latency
    - s, time from actual measurement to it is available
  - Update frequency
    - Hz, rate of measurements
  - Confidence value
    - [0;1], probability that it is the correct feature that is detected
  - Pose accuracy
    - Gaussian noise?
  - Time to live
    - s, time a relationship is likely to be valid

Attributes for Relation Quality

- Other attributes like the costs of the sensors can also be useful.
  - Before building the system a model can be created with all available trackers from the marked. An error function, including cost, can be used for figuring out optimal combinations with certain cost restraints.

- Open question: can a general attribute set for all applications be found?

Error functions

- Can be an arbitrary complex function
  - speedup can be gained if an edge by edge evaluation is possible
    - → Weight in Dijkstra's shortest path algorithm

Simple example – weighting between lag and update rate:

$$e^t := \sum_{q \in path} \mathrm{lag}(q) + \frac{\lambda}{\mathrm{rate}(q)}$$

Optimizations

- The formalism described so far:

$$Q : (\Psi \subseteq N \times N) \rightarrow \text{q}, \quad \text{where} \quad \text{q} : D_t \rightarrow \text{R}^{4\times4} \times \text{A}$$

$$q^f_{XY}(t) = f((H_1, A_1), (H_2, A_2), \ldots, t)$$

$$e : A^* \longrightarrow \mathbb{R}$$

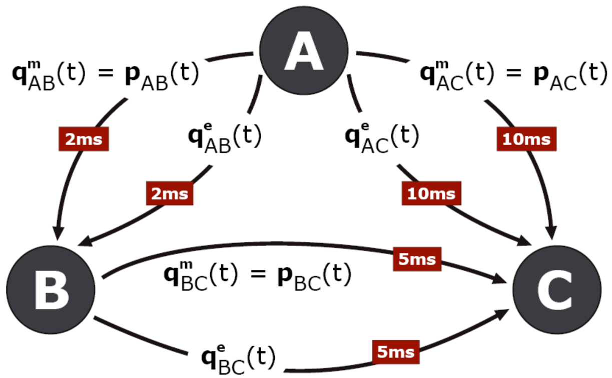
- Performance issues:
  - Data flow graphs – save optimal paths found
  - Super nodes – group nodes to reduce complexity

Optimizations – Data Flow Graphs

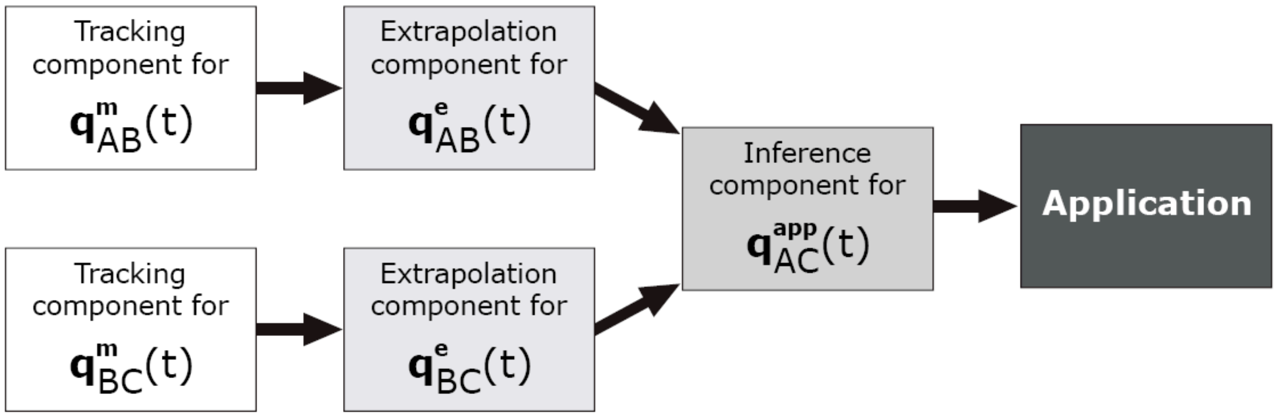
- Build data flow graphs from the found optimal paths in a component based manor so
  - An edge from a relationship graph is considered one node
  - Constructed as an a three structure with
    - Data flow from the leaves to the root
    - Measurements and information of static setups as leaves
    - Fusion, extra-, interpolation, filter and other inferred components as the internal nodes
    - The root is the final relation

Optimizations – Data Flow Graphs

- Example graph:



- Found optimal path *P1* from *A* to *C*: $\mathbf{P1} : A \xrightarrow{q^e_{AB}} B \xrightarrow{q^e_{BC}} C$
- Constructed data flow graph for *P1*:



Optimizations – Data Flow Graphs

- This approach depends on:
  - Rare structural changes to the relationship graph
    - Tracked person goes to another room
  - Rare changes of attributes for the components in the created data flow graph
    - The error function depends only on the attributes so if the attributes rarely change then an update to the data flow graph is rarely necessary

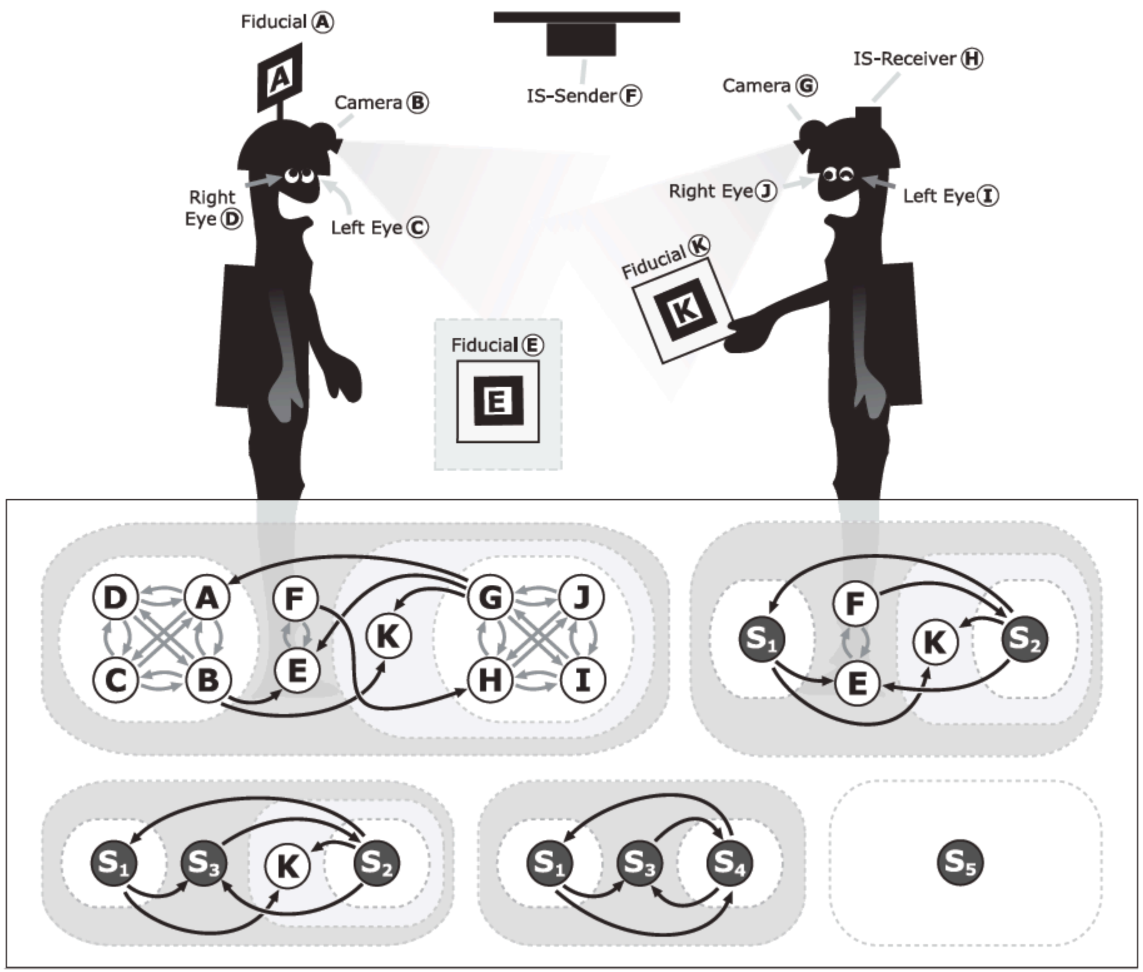
- Thus once setup, graphs searches may be performed in the background leaving computation of the pose estimation to be performed in real time.

Optimizations – Super Nodes

- Grouping of nodes to optimize performance can be done if
  - the nodes are statically related
  - the graph searches are concentrated to clusters of nodes – (near) cliques

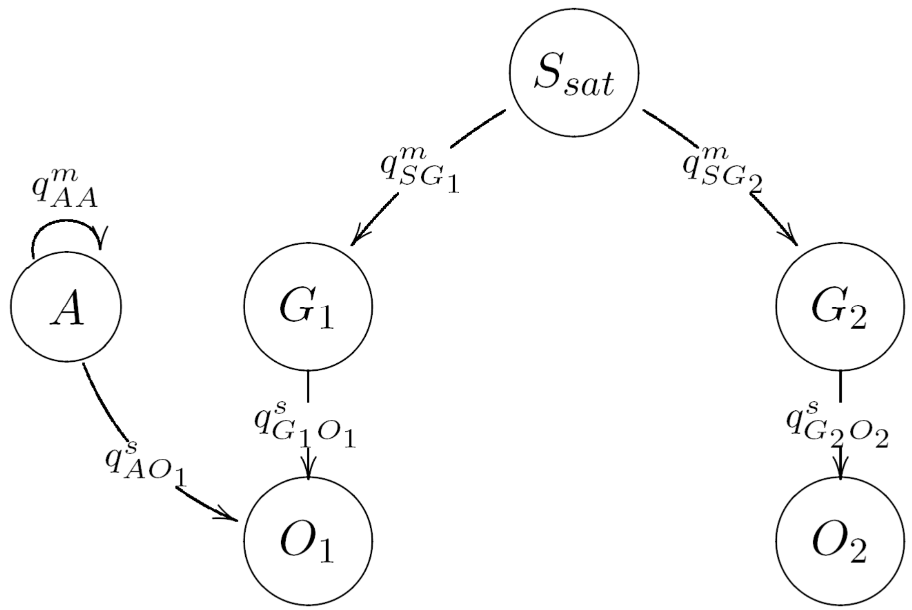
Optimizations – Super Nodes

- Example:



Example of Framework use
- combining GPS with an accelerometer

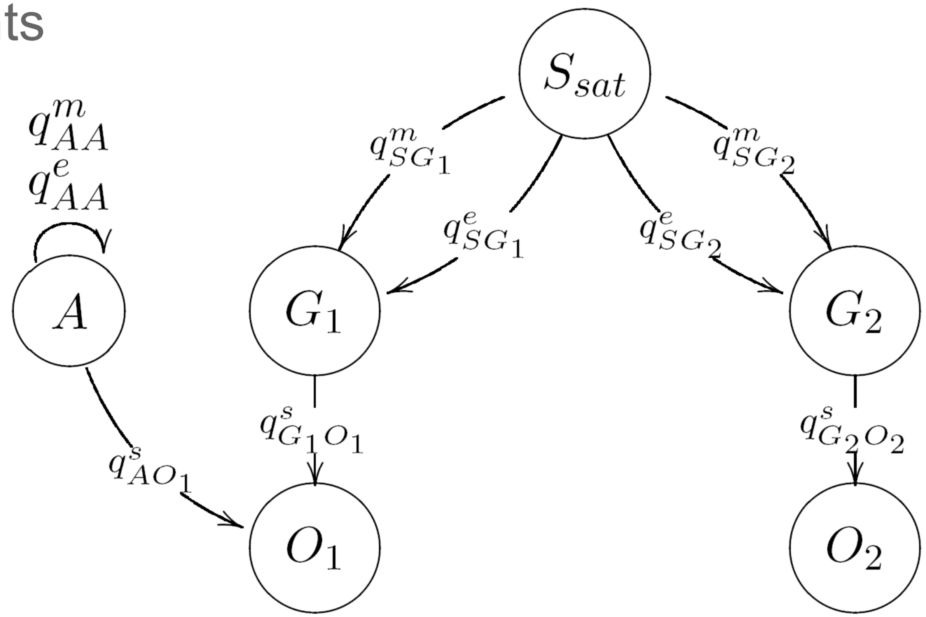
- Measurements:
  - O_1 is a moving car
  - O_2 is a stationary object



- Querying:
  - O_1 spatial relation to O_2 at time T+Δt (via Kalman filter)

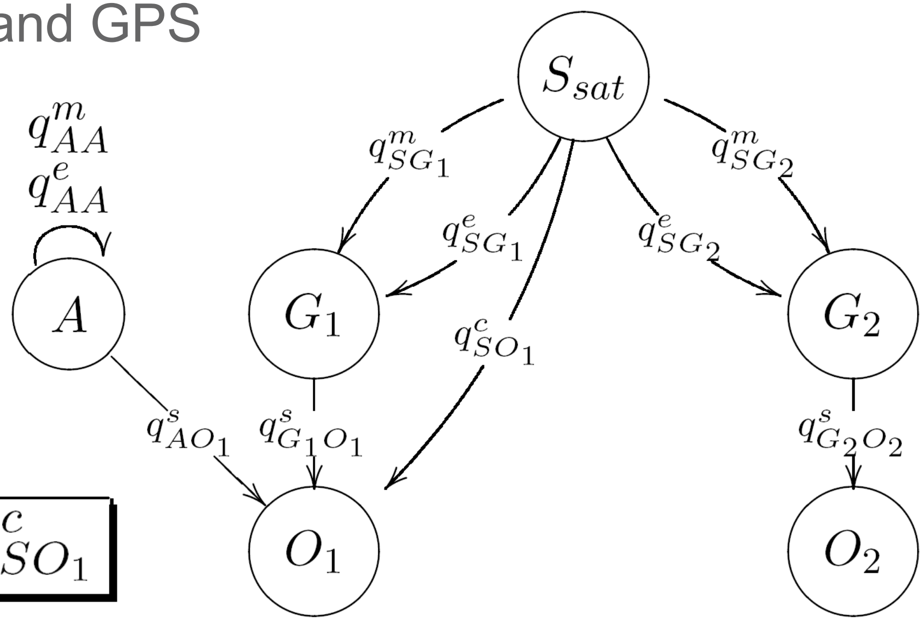
Example of Framework use
- combining GPS with an accelerometer

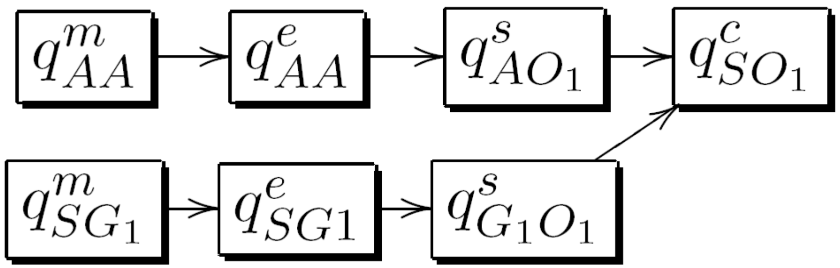
- O_1 spatial relation to O_2 at time T+Δt (via Kalman filter)
  - Extrapolate measurements



Example of Framework use
- combining GPS with an accelerometer

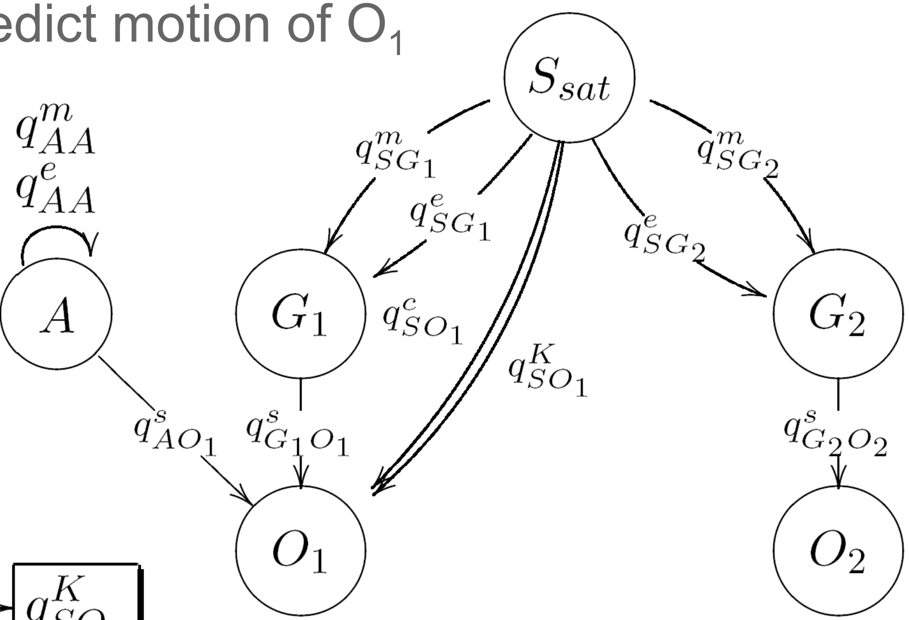
- O_1 spatial relation to O_2 at time T+Δt (via Kalman filter)
  - Combine accelerometer and GPS

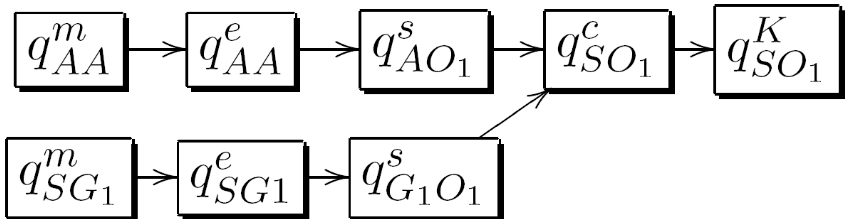




Example of Framework use
- combining GPS with an accelerometer

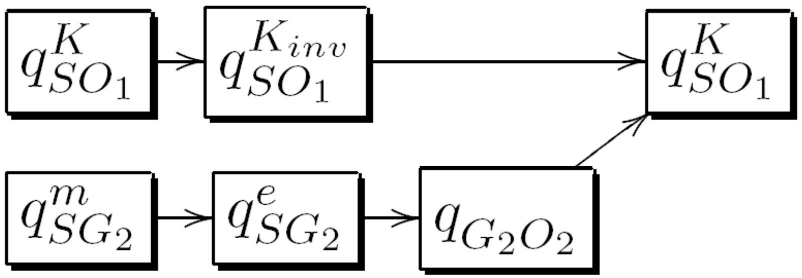
- O_1 spatial relation to O_2 at time T+Δt (via Kalman filter)
  - Use Kalman filtering to predict motion of O_1

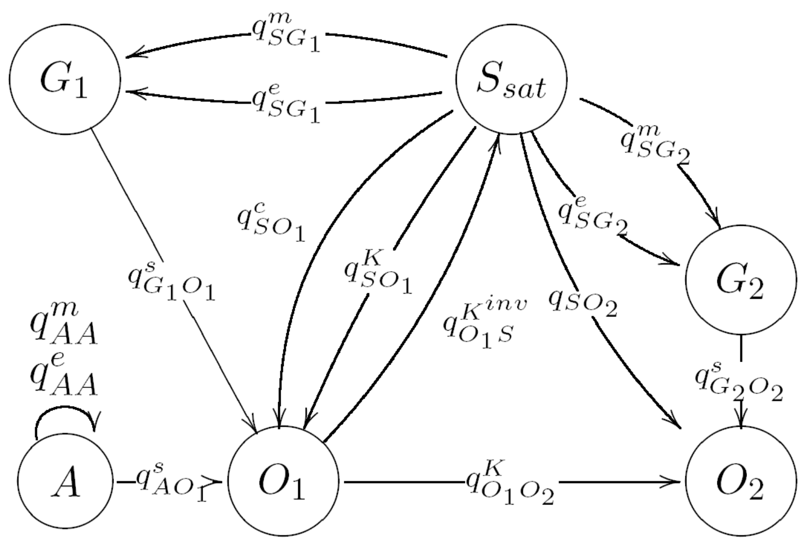




Example of Framework use
- combining GPS with an accelerometer

- O_1 spatial relation to O_2 at time T+Δt (via Kalman filter)
  - Infer relation between O_1 and O_2





Questions

Introduction & Concepts of Ubiquitous Tracking - References

- Fundamentals of Ubiquitous Tracking for Augmented Reality
  J. Newman, M. Wagner, T. Pintaric, A. MacWilliams, M. Bauer, G. Klinker, D. Schmalstieg

- An Architecture for Distributed Spatial Configuration of Context Aware Applications
  M. Wagner, G. Klinker

- Distributed User Tracking Concepts for Augmented Reality Applications
  G. Klinker, T. Reicher, B. Bruegge