Particle Filter Speed Up Using a GPU

15 September 2010

High Performance Embedded Computing Workshop

MIT Lincoln Labs

By

John Sacha & Andrew Shaffer

Applied Research Laboratory
The Pennsylvania State University
P. O. Box 30
State College, PA 16804

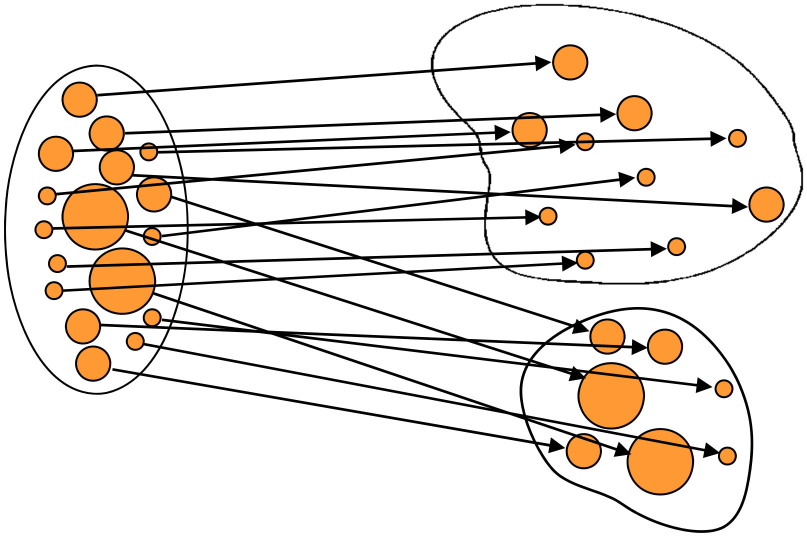
This work was sponsored by the Office of Naval Research, Code 333, under contract number N00014-05-G-0106/0006.
(Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of ONR.)

Target Tracking

- Things people want to track
  - Physical objects, using radar and sonar (airplanes, ships, fish schools, …)
  - Time series (financial information, weather statistics, …)
  - Classification features (spectral lines, …)
- Tracking is commonly formulated as a probabilistic state-space problem
  - Chapman-Kolmogorov/Fokker-Planck equations for density evolution
  - Bayesian approach to density updating
- There are few closed-form solutions
  - Gaussianity and linearity assumptions lead to the Kalman Filter
- Simple special-case extensions can be made (e.g., Extended Kalman Filter, mixture models)
- General solutions require use of numeric methods
  - Mesh representations
  - Monte Carlo integration (particle filters)

Particle Filters for Tracking in Complex Environments

- Particle Filters are a sequential Monte Carlo methodology for state estimation in which collections of weighted point particles are used to model state probability density functions
- Able to handle complicated non-Gaussian and non-linear problems not easily solvable by Kalman Filters
- Approximate solution to the exact problem, rather than an exact solution to an approximate problem
- Other names
  - Condensation Trackers
  - Bootstrap Filters
  - Survival of the Fittest
- Appropriate for:
  - Applying non-linear constraints
    - Boundaries
    - Kinematic limitations
  - Handling sensor blind spots (exploiting absence of measurements)
  - Tracking “features”
  - Groups of objects



Particle Filters and GPUs: A Marriage Made in Heaven?

- Particle Filters are flexible and powerful, but …
  - Accuracy increases with the number of particles
  - So does the computation cost
- Graphic Processing Units
  - Commodity items specialized for gaming/imaging applications
  - Highly parallel co-processors
  - Lots of bang for the buck (and watt)
- Particle Filter paradigm appears well-suited to parallel processing
  - Particles can be propagated independently (mostly!)

Particle Filter Algorithm

State PDF Representation

Collection of particles { (x,w) }

Arbitrary Models

$x' = F(x,\mathcal{U})$ (propagation)

$z = G(x,\mathcal{V})$ (measurement)

Prediction Step:

For each particle:

$x'' = F(x,u)$

Correction Step:

For each particle:

$w'' = P(x'|z)w$

Embarrassingly Parallel Steps

Periodic resampling of population:

Cumulative sum of weights

Sorted RNG

Binary search

Possibly Complicated

Loop