Adaptive Real-Time Computation for Prompt Localization of Transients

Marion Sudvarg (msudvarg@wustl.edu, www.sudvarg.com)
with Ye Hiit, Jeremy Buhler, Roger Chamberlain, Chris Gill, Jim Bulkeley, and Wentie Chen for the APT collaboration

The Astro2020 decadal survey identified "time-domain and multi-messenger" programs as the highest-priority sustaining activity for space-based missions.

Key Motivation
- Need to locate precocious bursts, e.g., promptly follow up observations.
- Can we perform localization on board space-based instruments?
- Limited computational capacity due to radiation hardening, size, weight, and power constraints, etc.
- But if we can, we are able to immediately communicate to space-based and ground-based follow-up instruments!

What do we want?
-ability to localize transients in real-time across space-based hardware.
- Make hard guarantees about latency using approaches from real-time, cyber-physical, and safety-critical computing.
- Adapt localization latency guarantees in the face of dynamic workloads and deadlines.

Dynamic Workloads:
- Amount of data to process may depend on transient’s flux, duration, etc.
- How long do we have access to communication network?
- Algorithms may change depending on quality of data, other characteristics.

Dynamic Deadlines:
- How long do we have to communicate to network?
- How far away are they (latency)?
- How exciting is this transient?
- How much time do we have for meaningful observations?

Case Study: Real-Time GRB Localization Aboard APT

The Advanced Particle-astrophysics Telescope (APT) is a future space-based observatory that will detect and localize GRBs in real time to enable concurrent, multi-messenger observations from any direction with minimal delay. For these soft transients, Compton regime gammas should dominate the emission spectrum. We have therefore designed a parallel computational pipeline for real-time multi-Compton GRB detection and GRB localization. To keep latency low, this will execute fully onboard the instrument, which imposes significant size, weight, and power constraints.

How can we adapt and compress the computational pipeline to maximize localization accuracy even for bright transients while guaranteeing short deadlines?

### 1. Identify Parameters
Identify the parameterized degrees of freedom over which computational workload may be compressed (i.e., reduced in a way that minimizes loss)

### 2. Characterize Loss
Identify the impact of reducing the computational workload over its multiple dimensions to an objective function for constrained optimization.

Through extensive simulation with synthetic bursts, we characterize loss as two monotonically-decreasing hulls of hypothesis in the 5 input dimensions.

For APT, loss is 68% containment (1o) localization error (degrees) and

### Approximation Techniques

- **Approx Circles**: Randomly select 20 rings from i, and uniformly distribute 720 points around each. Find the point on each ring with the greatest joint log likelihood with respect to all i, j, rings. Weighted mean over those points approximates the source.
- **Fibonacci Spiral**: A fast but less accurate technique. Distribute 100 points uniformly over the surface of the unit sphere. Find the joint log likelihood of each point with respect to all i, j, points. Weighted mean over the top 10 approximates the source.

### Approximation Techniques

- **Platform**: Raspberry Pi 3+ (RP3) Core Cortex A53 700MHz**
- **Wintel** (RC5) Core Intel Atom E840 1.2GHz
- **Wintel** (RC7) Core Intel Atom E840 1.2GHz

Let’s characterize the shape of the computational offline so that we can adapt online to achieve expected Pareto-optimal results within the imposed deadline.

Computational requirements and timing constraints may not be known a priori!

Let’s talk about how these ideas can extend to your application!