Numerical Optimization for The Artificial Retina Algorithm

preliminary study

Maxim Borisyak [1,2] Mikhail Belous [2]
Andrey Ustyuzhanin [1,2] Denis Derkach [1,2]

May 13, 2016

[1]National Research University Higher School of Economics (HSE)

[2]Yandex School of Data Analysis

Overview

Artificial Retina

Given set of hits $\{\mathbf{x}_i\}_{i=1}^{N}$ and track model parameterized by θ:

$$R(\theta) = \sum_{i=1}^{N} \exp\left(-\frac{\rho^2(\theta, \mathbf{x}_i)}{\sigma^2}\right)$$

where $\rho_i(\theta) = \rho(\theta, \mathbf{x}_i)$ --- distance from $\mathbf{x}_i$ to track with parameters θ.

Usage

For sufficiently small σ^2 local maxima[1]of R correspond to track parameters.

[1]For sufficiently large $R \gg 1$. A noisy hit still produces maximum however with $R \approx 1$.

Example

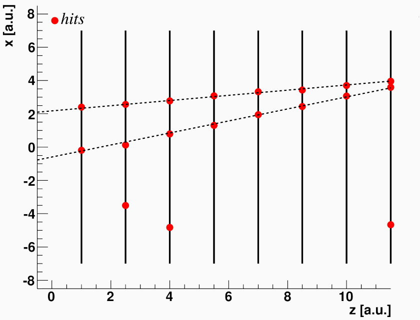
Consider VELO with tracks as straight lines coming from one point.
Possible track parameterization:

- pseudo-rapidity η and angle in the traverse plane ϕ;
- track direction $\mathbf{n} = (n_x, n_y, n_z)$

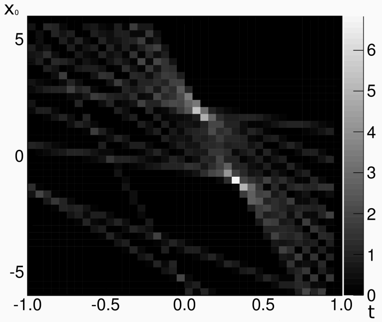
Possible distance functions:

- projection error: $\rho(\mathbf{n}, \mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{n}(\mathbf{n} \cdot \mathbf{x}_i)\|$
- projection error in the corresponding VELO plane $(z = \mathrm{const})$.

Example I



(a) An event example.



(b) Retina response.

Figure 1: An example of an event with two tracks (dashed lines) and some noisy hits (1a) and response of the Artificial Retina in parameter space (1b). Tracks parametrized by $\theta = (x_0, t)$: $x = x_0 + tz$

[1]Figures from [Abba et al., 2015].

Example II

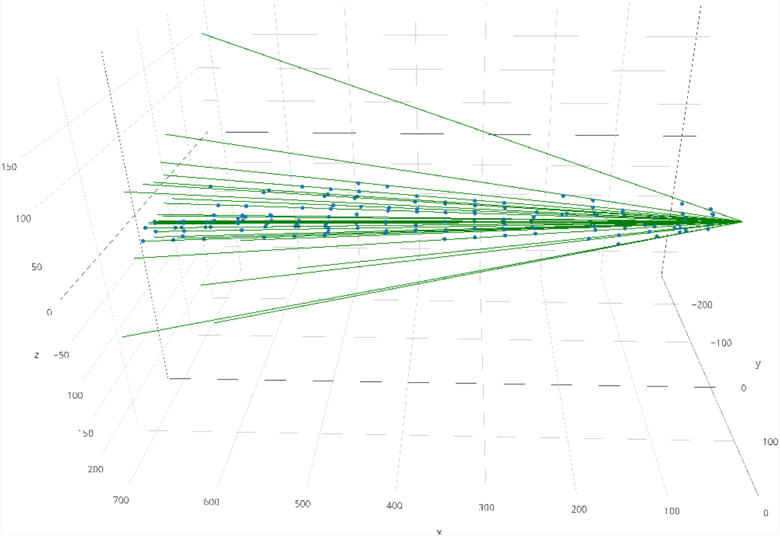


Figure 2: Another event example (simplified VELO model, see below).

Example II

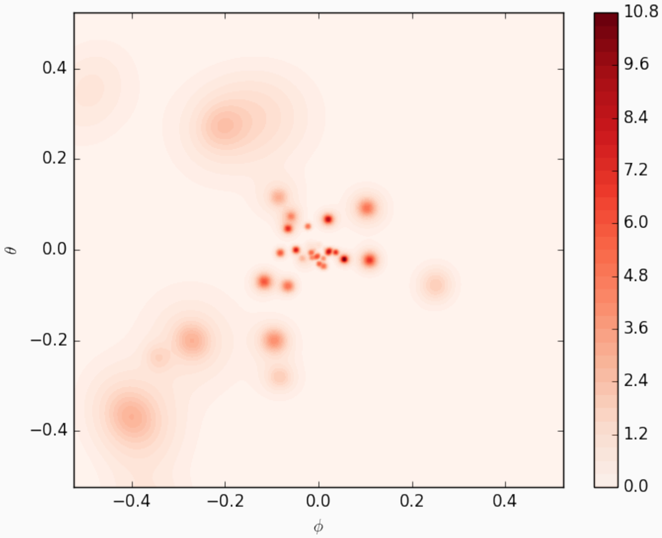


Figure 3: Retina response in for the example event on fig. 2.

Interpretation

- approximation of the number of hits that lie on the track;
- conceptually similar to Hough transform.

Features

- the algorithm is defined by the track model and the distance function ρ;
- the objective function is smooth;
- robust to noisy hits.

Application

Specialized Artificial Retina processor for LHCb

[Abba et al., 2014] proposes specialized Artificial Retina processor for *real-time* track reconstruction.

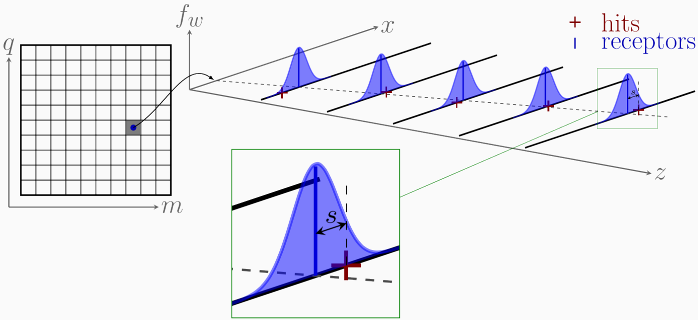


Figure 4: Retina schematic (from [Abba et al., 2015]).

Specialized Artificial Retina processor for LHCb

The processor performs grid-search[2] over the parameter space with two 'major' parameters and three 'minor' ones. However, the computational complexity is dramatically reduced due to intelligent distribution of hits over grid-cells.

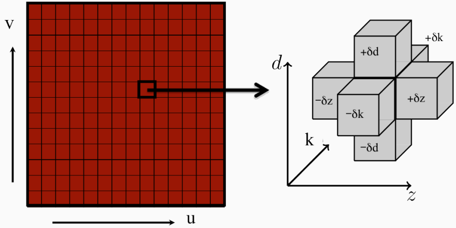


Figure 5: The Artificial Retina processor grid (from [Abba et al., 2014]).

[2] The actual algorithm is more advanced and more efficient, see [Abba et al., 2014] for details.

Numerical optimization study

Artificial Retina challenges

- computing Retina response in a point is a easy task for SIMD processors;
- still **the whole parameter space** needs to be explored.

Our aim

- reduction in **total** computational complexity;
- increase in precision;
- general-purpose Artificial Retina Algorithm for pattern search;
- study alternative approaches for searching maxima of Retina response.

Main idea

Exploit gradient information to reduce total computational time.

Elaboration

Intermediate results of $\mathrm{Hessian}(R)$ computation can be reused to compute R and ∇R.

Hence ∇R and $\mathrm{Hessian}(R)$ can be computed in $\approx 1.5\times$, $\approx 2\times$ time of R computation time. E.g. consider ∇R:

$$\begin{aligned}\nabla R(\theta) &= -\frac{1}{\sigma^2}\sum_i \exp\frac{-\rho^2(\theta, h_i)}{\sigma^2}\rho(\theta, h_i)\nabla\rho(\theta, h_i)\\ &= -\frac{1}{\sigma^2}\sum_i R_i(\theta)\rho(\theta, h_i)\nabla\rho(\theta, h_i)\end{aligned}$$

Multi-start algorithm

1. generate *n* initial guesses, set initial σ^2;
2. perform one step of hill climbing for each of *n* points;
3. decrease σ^2;
4. repeat *m* times from step 2.

Analysis

+ complexity is proportional to the number of initial guesses;
+ much less affected by dimensionality curse,
- stochastic nature;
- less parallelization capacity, hence increase in latency.

Experiment

Implementation details

Details

- `Python`;
- `theano` on GPU for computing R, ∇R and $\mathrm{Hessian}(R)$;
- truncated Newton–Raphson method.

Parallelization

- optimization processes are independent;
- R, ∇R and $\mathrm{Hessian}(R)$ can be efficiently implemented on SIMD processors (e.g. GPU);
- latency increases in n times, where n - number of steps for each initial guess.

Simplified model of VELO

Simplified model of VELO was simulated:

- tracks - straight lines;
- simplified 'VELO' parameters[3]:
  - 20 disc layers with inner $r = 8$ mm, outer - $R = 42$ mm;
  - length: $L = 700$ mm;
  - probability of a particle interacting with a layer: $P_{\mathrm{int}} = \frac{1}{2}$;
  - hit error: $\epsilon \sim \mathcal{N}(0, 10^{-3})$ mm;
  - number of noisy hits: $N' \sim \mathrm{Poisson}(250)$;
- number of secondary particles: $N \in [50, 350]$
- $\eta \sim \mathrm{Uniform}[1, 5]$;
- $\phi \sim \mathrm{Uniform}[0, 2\pi]$;
- primary vertex: $z_0 \sim \mathcal{N}(0, 5) mm$.

[3] Parameters are motivated by upgrade VELO TDR.

Experiment

Evaluation

- track parametrized by spherical angles (θ, ϕ):

$$\begin{aligned} n_x &= \sin\theta; \\ n_y &= \cos\theta \sin\phi; \\ n_z &= \cos\theta \cos\phi; \end{aligned}$$

- a track is considered detected if the method reports local maximum within $\epsilon = 5 \times 10^{-3}$ rad. from the track's parameters;
- computational time is relative to the amount required by grid-search to provide ϵ resolution.
- number of steps for each initial guess $n = 5$.

Multi-start

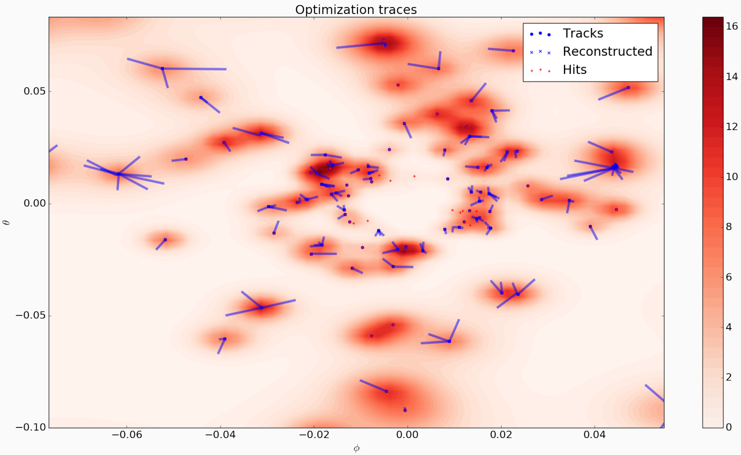


Figure 6: Optimization traces (blue lines) for an event in the simplified VELO. Heat map corresponds to the Retina response for the event.

Results

Computational limit is **1 / 3** of grid-search time.

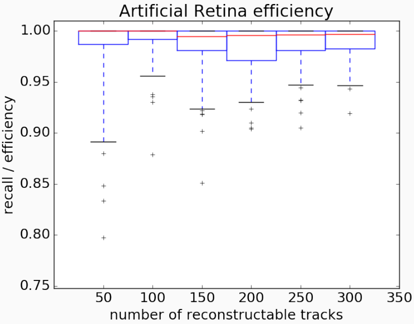


Figure 7: Box plot of method's efficiency (recall) depending on the number of reconstructable tracks. Red line and blue box represent median, lower and upper quartiles. Black lines correspond to 5 % and 95 % quantiles. Ghost rate for the method is strictly zero for all events.

Results

Computational limit is **1/10** of grid-search time.

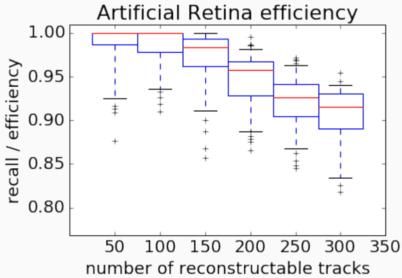


Figure 8: Box plot of method's efficiency (recall) depending on the number of reconstructable tracks. Note how efficiency decreases as the number of initial guesses (in this case ≈ 400) approaches to the number of tracks. Ghost rate for the method is strictly zero for all events.

Summary

Future work

General

- helix curve fitting;
- **hybrid method**: grid-search like [Abba et al., 2014] with local refinement.

Method improvements

- custom heuristic optimization procedure;
- memetic-like algorithms:
    - **global method + local search**;
    - presented: random guessing + Newton-Raphson method;
    - possible enhancement: simulated annealing + local search;
- σ^2 optimal regime;

Summary

Artificial Retina algorithm

- efficient for high-luminosities [Abba et al., 2015];
- high parallelization capacity;

Numerical optimization for Retina

- gradient and Hessian can be efficiently computed;
- numerical optimization for local track search;

Results

- reduction in *total* computation time, **but**
- probabilistic results and increase in latency;

References I

Abba, A., Bedeschi, F., Citterio, M., Caponio, F., Cusimano, A., Geraci, A., Marino, P., Morello, M., Neri, N., Punzi, G., et al. (2015).
Simulation and performance of an artificial retina for 40 mhz track reconstruction.
Journal of Instrumentation, 10(03):C03008.

Abba, A., Punzi, G., Spinella, F., Marino, P., Tonelli, D., Stracka, S., Lionetto, F., Ninci, D., Petruzzo, M., Cusimano, A., et al. (2014).
A specialized track processor for the lhcb upgrade.
Technical report.