Track and vertex reconstruction

From classical to adaptive methods

R. Frühwirth

Institute of High Energy Physics
Austrian Academy of Sciences, Vienna

ACAT 2010, Jaipur, India
February 23, 2010

Outline

Introduction

What is it all about?

- Track and vertex reconstruction are essential steps in the data analysis chain
- Crucial factor in quality of physics analysis
- Growing importance for high-level trigger
- T **Track reconstruction:** determine location, direction and (inverse) momentum of charged tracks
- V **Vertex reconstruction:** determine location of interaction point and momenta of participating tracks
- T V Many basic features in common

Introduction

Ingredients

Track reconstruction

- ☺ Track model known (analytical or numerical)
- ☺ Observation errors known
- ☺ Process noise known (approximately)
- ☹ Assignment of observations to tracks unknown
- ☹ Hits from background tracks, electronic noise
- ☹ Mass not always known

Vertex reconstruction

- ☺ Track model known (analytical or numerical)
- ☺ Track errors known
- ☺ No process noise
- ☹ Assignment of tracks to vertices unknown
- ☹ Background tracks

Introduction

Three aspects

- **Pattern recognition**
  - T Find out which detector hits belong to the same track — highly detector dependent
  - V Find out which tracks are produced at the same interaction point — nearly detector independent
- **Estimation**
  - T Estimate track parameters at one or several points along the track
  - V Estimate vertex location and momenta of attached tracks
  - T V Can be formulated as extended Kalman filter in both cases
- **Test**
  - T Test track hypothesis and reject outlying detector hits
  - V Test vertex hypothesis and reject outlying tracks

Introduction

Classical vs. Adaptive

- **Classical approach**:
  - Do pattern recognition (track/vertex finding)
  - Submit track/vertex candidate to least-squares fit
  - Inspect test quantities, identify outlying hits/tracks
  - Remove outliers, repeat fit
  - ...
- **Adaptive approach**:
  - Do preliminary pattern recognition or none at all
  - Submit hit/track collection to adaptive fit
  - Inspect posterior weights of hits/tracks and remove outliers

Track finding

Global vs. local

- Rough distinction: **local/sequential** and **global/parallel** methods
- **Local** method: generate seeds and complete them to track candidates
- **Global** method: simultaneous clustering of detector hits into track candidates

Track finding: Local methods

Some local methods

- Track road
- Track following
- Progressive track finding

Track finding: Local methods

Track road

- Select two or three hits
- Compute approximative track plus tolerance
- Pick up hits inside the tolerance

Track following

- Construct initial track segment (seed)
- Extrapolate seed
- Pick up matching hits
- Repeat until last detector layer

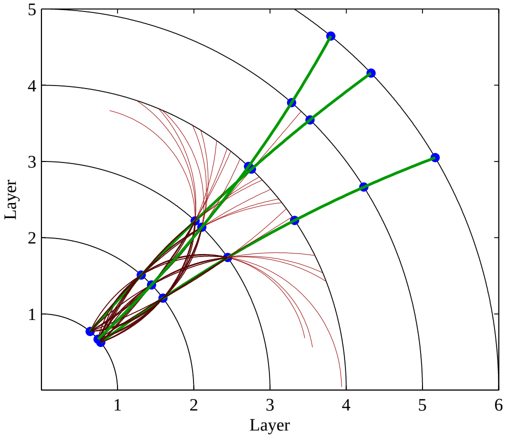
Track finding: Local methods

Progressive track finding

- Construct initial track segment (seed)
- Extrapolate seed
- Select best matching hit inside tolerance
- Update track parameters (weighted mean)
- Repeat until last detector layer
- Billoir and Qian (1990a), Billoir and Qian (1990b)

Track finding: Local methods

Example: progressive track finding



Track finding: Global methods

Some global methods

- Hough transform
- Legendre transform
- Hopfield network
- Elastic net
- Cellular automaton

Track finding: Global methods

Finding lines with the Hough transform

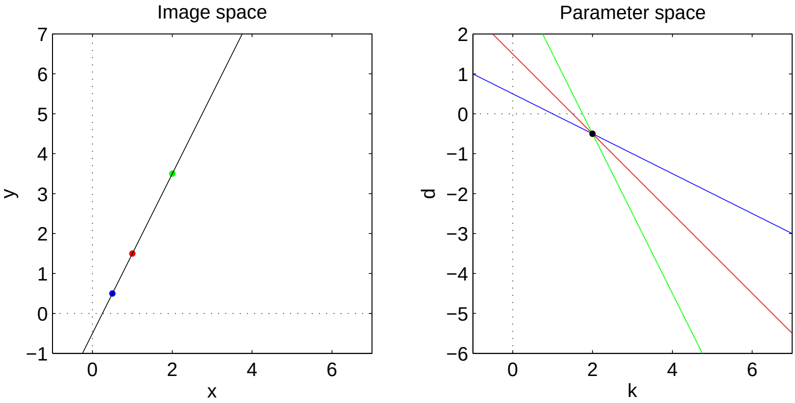
- Circles through the origin can be transformed to lines by a conformal transformation
- A point (x_0, y_0) in the image space (x/y) is transformed into a straight line

$$d = y_0 - kx_0$$

  in parameter space (k/d).
- Points on the straight line $y = k_0 x + d_0$ in image space are transformed into lines intersecting in the point (k_0/d_0) in parameter space
- If parameter space is discretized, intersections of lines can be found by histogramming

Track finding: Global methods

Finding lines with the Hough transform



Track finding: Global methods

Finding circles with the Hough transform

- A point (x_0, y_0) on a circle through the origin with center u/v is transformed into a straight line

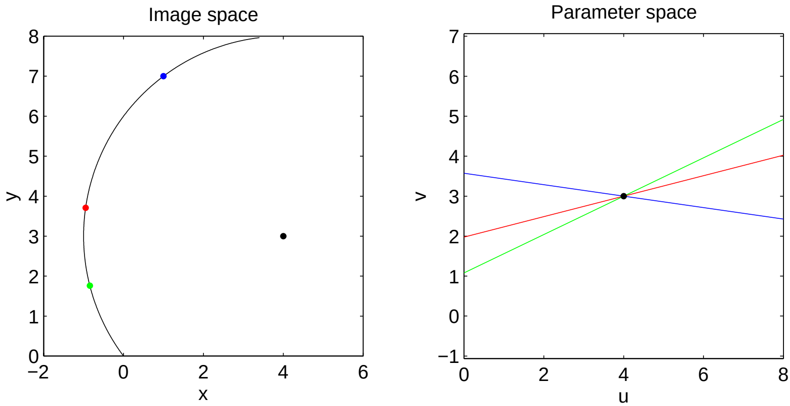
$$v = -\frac{x_0}{y_0}u + \frac{x_0^2 + y_0^2}{2y_0}$$

in parameter space (u/v)

- Points on the circle with center u_0/v_0 in image space are transformed into lines intersecting in the point (u_0/v_0) in parameter space
- If parameter space is discretized, intersections of lines can be found by histogramming

Track finding: Global methods

Finding circles with the Hough transform



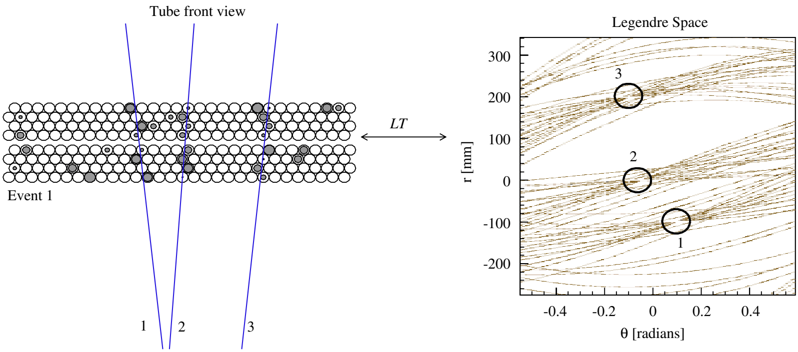
Track finding: Global methods

Finding tangents with the Legendre transform

- Track finding in drift tubes
- The drift circles are transformed by a Legendre transform to sine curves in polar coordinates
- Peaks at the intersections of several sine curves in this coordinate system correspond to common tangents to the drift circles
- See Alexopoulos (2008)

Track finding: Global methods

Example: Legendre transform



From: Alexopoulos *et al.*, NIM A 592 (2008) 456

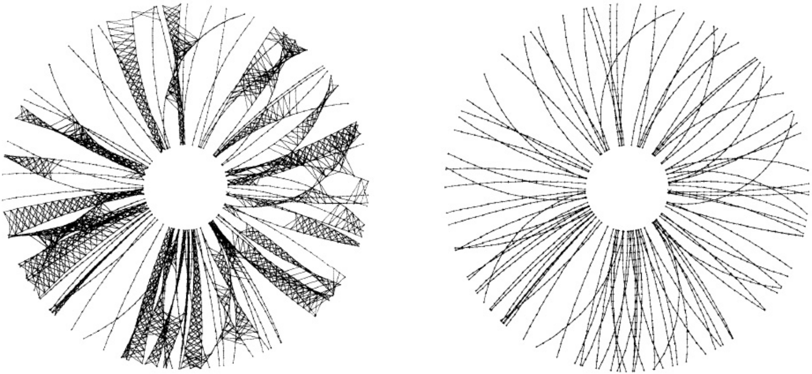
Track finding: Global methods

Track finding with a Hopfield network

- First adaptive approach to track reconstruction (Denby, 1988; Peterson, 1989)
- Track segments are neurons of a recursive ANN
- Network weights favor smooth tracks without bifurcations
- Energy function is minimized by gradient descent
- ☺ Deterministic annealing helps to find global optimum
- ☹ No physical track model

Track finding: Global methods

Example: Hopfield network



From: Stimpfl and Garrido, CPC 64 (1991) 46

Track finding: Global methods

Track finding with an elastic net

- Elastic net is a special type of neural network, related to Traveling Salesman Problem
- Neurons are attracted to the detector hits and to each other
- No physical track model
- See Kisel and Kovalenko (1996)
- Also used for ring finding in RICH detectors (see talk by S. Lebedev at this conference)

Track finding: Global methods

Example: elastic net

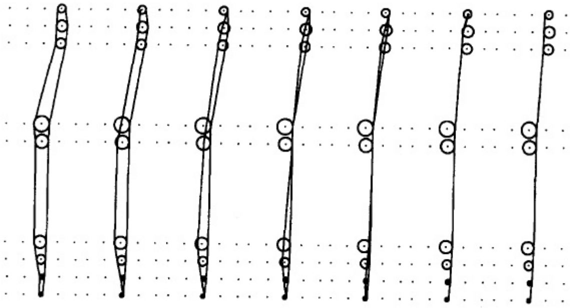


Fig. 7. Example of the elastic net evolution for a track with multiple scattering.

From: Kisel and Kovalenko, CPC 98 (1996) 45

Track finding: Global methods

Track finding with a cellular automaton

- Cells of the automaton are space points or track segments
- Find a sequence of cells that maximizes track length and track smoothness
- Optimization by appropriate update rules of the cells
- No physical track model
- See Kisel (1997), Abt (2002)

Track fitting: Traditional approach

Least-squares estimation

- Take track candidate and pass it to a least-squares estimator
- Three types: global, recursive, breakpoints
- **Global**: Set up regression model
- **Breakpoints**: Estimate track segments and multiple scattering angles
- **Recursive**: Interpret track as dynamic system and estimate with extended Kalman filter
- All three are **optimal** in the linear model with normal noise

Track fitting: Traditional approach

Regression

- In general non-linear model:

$$\boldsymbol{m} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \quad \mathsf{Cov}[\boldsymbol{\epsilon}] = \boldsymbol{V} = \boldsymbol{G}^{-1}$$

$\boldsymbol{m}$ ... Measurements $\boldsymbol{x}$ ... Initial track parameters
$\boldsymbol{h}$ ... Regression model $\boldsymbol{\epsilon}$ ... Measurement errors plus MS

- Minimize objective function:

$$M(\boldsymbol{x}) = (\boldsymbol{m} - \boldsymbol{h}(\boldsymbol{x}))^{\mathrm{T}} \boldsymbol{G} (\boldsymbol{m} - \boldsymbol{h}(\boldsymbol{x}))$$

Track fitting: Traditional approach

- Minimization methods: Gauss-Newton, Newton-Raphson, conjugate gradients, ...

$$\tilde{\boldsymbol{x}} = \arg\min M(\boldsymbol{x})$$

Test statistics

- Total χ^2

$$\chi^2 = M(\tilde{\boldsymbol{x}})$$

- Standardized residuals (pulls)

Track fitting: Traditional approach

Breakpoints

- Explicit estimation of multiple scattering angles
- Prior information about multiple scattering angles is used:

$$\mathsf{E}[\vartheta_{\mathrm{p}}] = 0, \quad \mathrm{var}[\vartheta_{\mathrm{p}}] = \sigma^2(m, p, d)$$

ϑ_{p} . . . Projected scattering angle

m . . . Mass of the particle

p . . . Momentum of the particle

d . . . Thickness of the material

Track fitting: Traditional approach

Kalman filter

- **Recursive**, no large matrices need to be inverted
- Estimated state vectors stay **close** to the actual track
- Track is described as **discrete dynamic system** (Frühwirth, 1987)
- **System equation**:

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) + \boldsymbol{\delta}_k, \quad \mathsf{Cov}[\boldsymbol{\delta}_k] = \boldsymbol{Q}_k$$

$\boldsymbol{x}_k \ldots$ State vector in layer k (local track parameters)

$\boldsymbol{f}_k \ldots$ Local track model

$\boldsymbol{\delta}_k \ldots$ Local process noise (multiple scattering)

Track fitting: Traditional approach

- **Measurement equation**:

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{\epsilon}_k, \quad \mathsf{Cov}[\boldsymbol{\epsilon}_k] = \boldsymbol{V}_k$$

$\boldsymbol{m}_k$ ... Measurement in layer k

$\boldsymbol{h}_k$ ... Measurement model

$\boldsymbol{\epsilon}_k$ ... Measurement error

- Kalman filter proceeds **recursively** by alternating two steps
  1. Prediction
  2. Update

Track fitting: Traditional approach

Prediction

- **Propagate** state vector and covariance matrix to the next layer, increment covariance matrix by contributions from multiple scattering and energy loss

Update

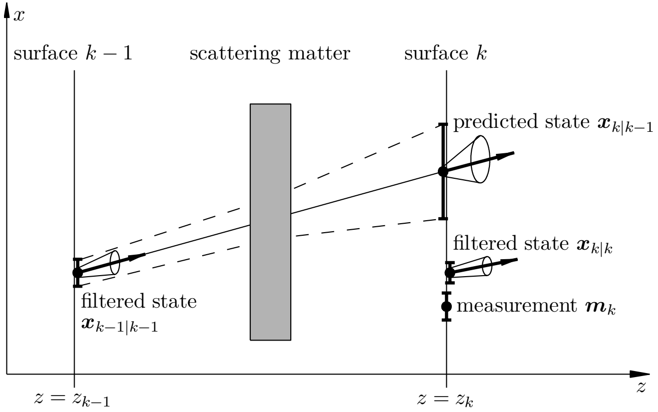
- Compute a **weighted mean** of the extrapolation and the observation

Test

- Local χ^2 statistic, ndf equals dimension of $\boldsymbol{m}_k$
- Total χ^2, sum of local χ^2 statistics

Track fitting: Traditional approach

Prediction and filter step



Track fitting: Traditional approach

Smoothing

- Optimal estimation of state vectors in **each layer**
- Standard algorithm, numerically unstable
- Combination of two filters (forward+backward) by a weighted mean, numerically stable

Test statistics

- Local χ^2s of the filter
- Total χ^2 of the filter (sum of local χ^2s)
- Local χ^2s of the smoother (correlated)

Track fitting: Adaptive approach

Problems of LS estimators

- ☹ The Kalman filter is LS-estimator, **not robust**
- ☹ Difficult to identify multiple outliers (bias)

Advantage of adaptive estimators

- ☺ **Concurrent** pattern recognition and estimation
  - Defer final decision to fitting stage
  - Complete information available
- ☺ **Automatic** suppression of background
  - Reduction of bias
  - No need to remove/add hits during fit
  - “Soft” assignment during entire fit
  - Can be made “hard” after optimal solution has been found

Track fitting: Adaptive approach

Various implementations

- **Elastic arms**, deformable templates: Ohlsson and Peterson (1992)
  - Based on neural network paradigm
- **Elastic tracking**: Gyulassi and Harlander (1991)
  - Inspired by Radon transform
- **Combinatorial Kalman filter**: Mankel (1997)
  - Full discrete combinatorial exploration
- **Gaussian-sum filter**: Frühwirth (1997)
  - Based on mixture models of noise
- **Deterministic annealing filter**: Frühwirth and Strandlie (1999)
  - Inspired by EM algorithm

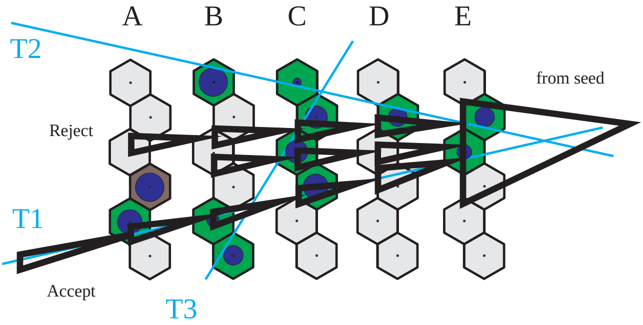
Track fitting: Adaptive approach

Combinatorial Kalman filter

- Extension of progressive track finding
- Full combinatorial exploration
- Several candidates are propagated in parallel
- Generate a branch for each compatible hit
- Generate a branch with a missing hit (optional)
- Limit growth by dropping branches
  - with bad total chi-square
  - with too many missing hits
  - which are subsets of other candidates
- Select "best" branch as final track

Track fitting: Adaptive approach

Example: Combinatorial Kalman filter



From: R. Mankel, NIM A 395 (1997) 169

Track fitting: Adaptive approach

Elastic arms, deformable templates

- First truly adaptive estimator
- Arms or templates are parameterized tracks
- Concurrent solution of two optimization problems
  - Continuous: minimize least-squares objective function
  - Discrete: decide which hit belongs to which template
- Discrete problem is transformed into a continuous one by **deterministic annealing**
- Minimization of the resulting non-quadratic energy function at each temperature

Track fitting: Adaptive approach

Deterministic annealing filter (DAF)

- Same principle as elastic arms
- Minimization by EM algorithm, implemented as **iterated re-weighted Kalman filter**
- Easy to deal with process noise
- Observations are assigned **weights**
- Iteration of two principal steps
  1. Full Kalman filter+smoother, using the current weights
  2. Calculation of weights, using current estimates
- The iteration ends when the weights are stable

Track fitting: Adaptive approach

Definition of the weights

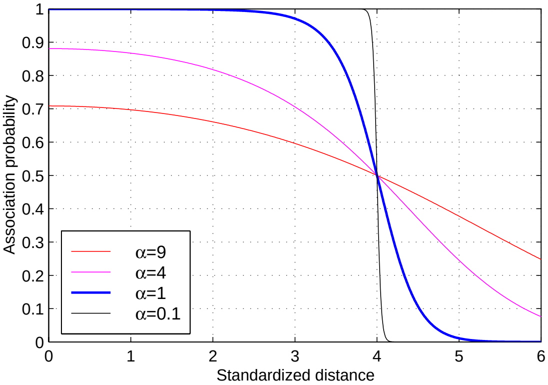
- Weight of observation i in layer k:

$$p_{ik} = \frac{\exp(-\chi^2_{ik}/2T)}{\exp(-\chi^2_{\mathrm{cut}}/2T) + \sum_j \exp(-\chi^2_{jk}/2T)}$$

- χ^2_{ik} measures the **distance** of observation i in layer k from the smoothed track state in layer k
- χ^2_{cut} is the **cut-off** parameter
- T is the **annealing** factor (temperature)
- For a single observation $p_{ik} = 0.5$ if $\chi^2_{ik} = \chi^2_{\mathrm{cut}}$

Track fitting: Adaptive approach

Weight function without competition

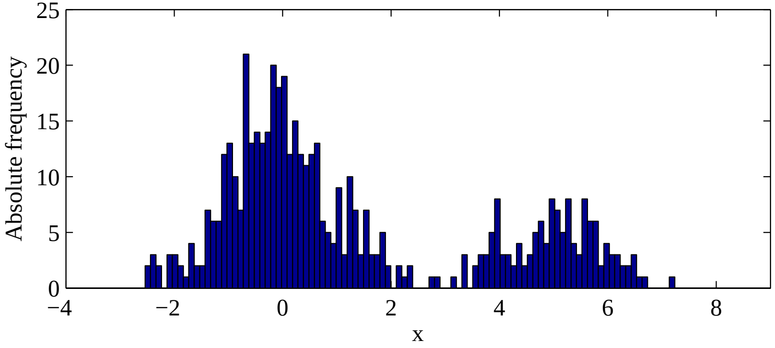


Weight function of an observation without competition

Track fitting: Adaptive approach

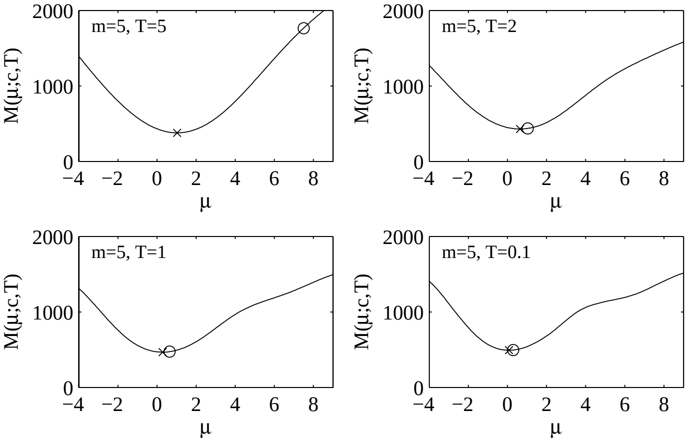
Example: 1D data with outliers

- Estimate location of the bulk of the data



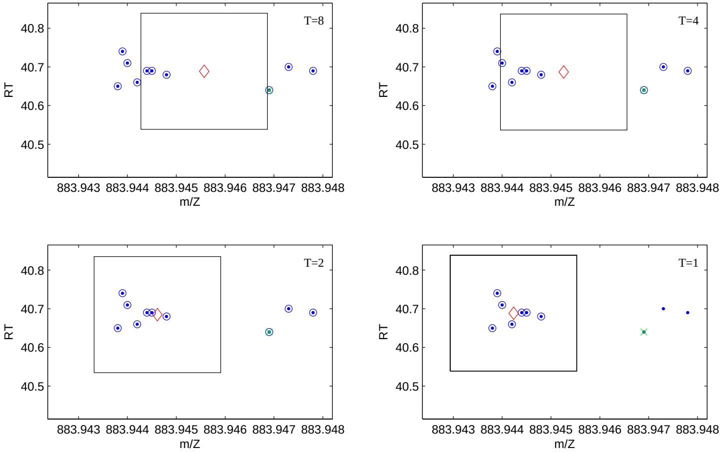
Track fitting: Adaptive approach

Evolution of the objective function



Track fitting: Adaptive approach

Example: 2D clustering of peptides



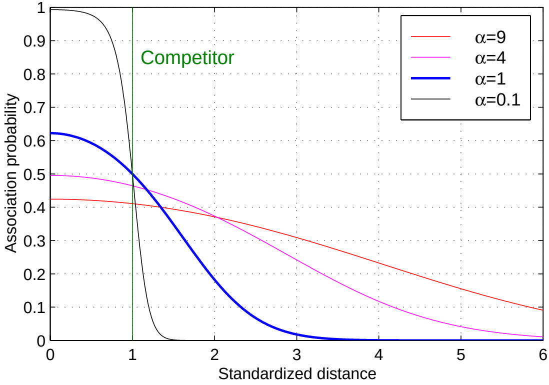
Track fitting: Adaptive approach

Weight function with competition

- If there are several observations in a detector layer, they may **compete** with each other
- A matching observations **suppresses** the other ones
- **Deterministic Annealing** helps to reach the optimal solution
  - At the start $T \gg 1$
  - During the iteration T is stepped down
  - The final value is $T = 1$
- Well-known technique of global optimization, e.g. with ANNs
- At $T > 0$ the association is "soft"
- "Cooling down" to $T = 0$ yields "hard" association. Not necessarily optimal!

Track fitting: Adaptive approach

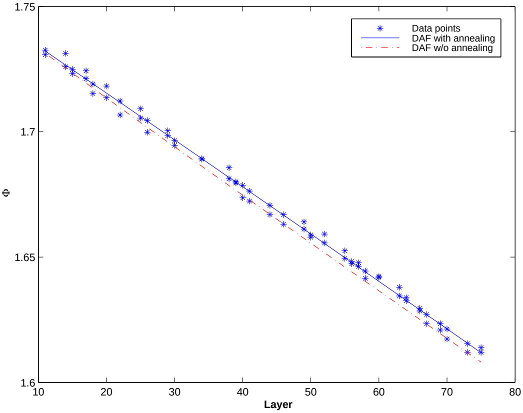
Weight function of the DAF



Weight function of an observation with competition

Track fitting: Adaptive approach

Example: DAF with and without annealing



From: Strandlie and Zerubia, CPC 123 (1999) 77

Track fitting: Adaptive approach

History of the DAF

- DAF was evaluated in CMS (Winkler, 2002)
- Studies with single tracks
- Studies in different physics contexts
- Implemented in CMS offline framework
- Implemented in ATLAS offline framework
- Studies of track finding with the DAF (Strandlie and Frühwirth, 2006)

Vertex reconstruction

Vertex finding

- Hierarchical clustering (agglomerative or divisive)
- Topological finding, Radon transform (Jackson, 1997)
- Minimum spanning tree (Hillert, 2008)
- Multi-layer perceptron (Lindsey and Denby, 1991)
- Adaptive vertex reconstructor (see below)

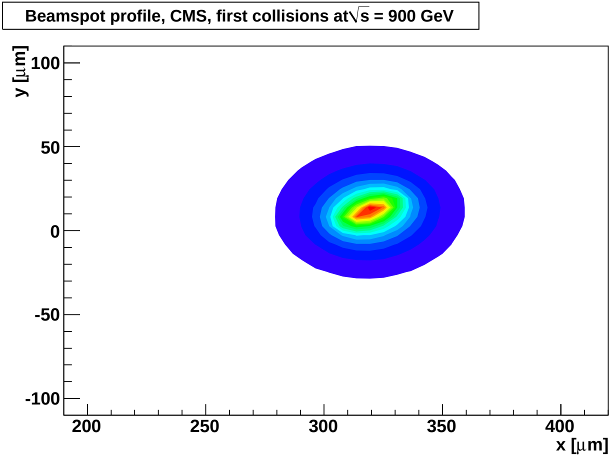
Vertex reconstruction

Adaptive vertex estimation

- Concepts of adaptive estimators can be transferred almost one-to-one from track to vertex fitting
- Algorithm is called **Adaptive Vertex Fitter (AVF)** (Waltenberger, 2004)
- Implemented as **iterated re-weighted Kalman filter**
- Outlying tracks are automatically down-weighted
- Resulting estimator is **highly robust**, but much easier to compute than other robust estimators such as LMS or LTS
- Extension to **Multi-Vertex Fitter (MVF)**: vertices compete for the tracks

Vertex reconstruction

Primary vertices estimated with AVF



Vertex reconstruction

Adaptive Vertex Reconstructor (AVR)

- Vertex finding by **iterated AVF** (Waltenberger, 2008)
  - Fit all tracks to a common vertex, using the AVF
  - Remove all tracks with weight above threshold
  - Fit all remaining tracks to a common vertex, using the AVF
  - Repeat until no valid vertex can be fitted
- Implemented and successfully validated in CMS offline software

Vertex reconstruction

RAVE Toolbox

- CMS algorithms (KF, AVF, AVR, ...) packed into detector-independent vertex reconstruction toolkit: **RAVE** (Waltenberger, 2007)
- Download from `http://projects.hepforge.org/rave`
- Used in new Belle2 framework
- Used for ILC studies (ILD, SiLC)

Conclusions and Outlook

- **Adaptive estimators** are useful tools for track and vertex reconstruction (Strandlie and Frühwirth, 2010)
- **Background** is automatically down-weighted, no need for iterative rejection of outliers
- **Competition** between hits or vertices is possible
- **Annealing** helps to reach the globally optimal solution
- **Implementations** are built on existing methods (Kalman filter)
- **Resistant** to high levels of noise, important for experiments at Super-LHC and the upgraded B-factory at KEK
- High-level trigger: Important work on **parallelization** going on (talks by M. McCool and I. Kisel at this conference)

Thanks to my colleagues:

Are Strandlie
Thomas Speer
Wolfgang Adam
Wolfgang Waltenberger
Matthias Winkler

Selected References (1)

- P. Billoir and S. Qian, NIM A 294 (1990) 219
- P. Billoir and S. Qian, NIM A 295 (1990) 492
- T. Alexopoulos *et al.*, NIM A 592 (2008) 456
- B. Denby, CPC 49 (1988) 429
- C. Peterson, NIM A 279 (1989) 537)
- G. Stimpfl-Abele and L. Garrido, CPC 64 (1991) 46
- I. Kisel and V. Kovalenko, CPC 98 (1996) 45
- I. Kisel *et al.*, NIM A 387 (1997) 433
- I. Abt *et al.*, NIM A 489 (2002) 389
- R. Frühwirth, NIM A 262 (1987) 444
- M. Ohlsson and C. Peterson, CPC 71 (1992) 77
- M. Gyulassi and M. Harlander, CPC 66 (1991) 31
- R. Mankel, NIM A 395 (1997) 169

Selected References (2)

- R. Frühwirth, CPC 100 (1997) 1
- R. Frühwirth and A. Strandlie, CPC 120 (1999) 197
- A. Strandlie and J. Zerubia, CPC 123 (1999) 77
- M. Winkler, PhD Thesis, Vienna Univ. of Technology (2002)
- A. Strandlie and R. Frühwirth, NIM A 566 (2006) 157
- D.J. Jackson, NIM A 388 (1997) 247
- S. Hillert, `http://arxiv.org/abs/0811.3904`
- C. Lindsey and B. Denby, NIM A 302 (1991) 217
- W. Waltenberger, PhD Thesis, Vienna Univ. of Technology (2004)
- W. Waltenberger *et al.*, NIM A 581 (2007) 549
- W. Waltenberger, CMS NOTE-2008/033, CERN, Geneva (2008)
- A. Strandlie and R. Frühwirth, Rev. Mod. Phys., to appear