# AIDA Alignment Package

*AIDA 3rd annual meeting, Vienna*
*27/03/2014*

Silvia Borghi, Christoph Hombach, Chris Parkes

**Outline:**

- Intro: Alignment problem and strategies
- Testbeam & VELO: AIDA TimePix telescope, LHCb VELO
- BACH: Standalone reconstruction & alignment software

# Deliverables:

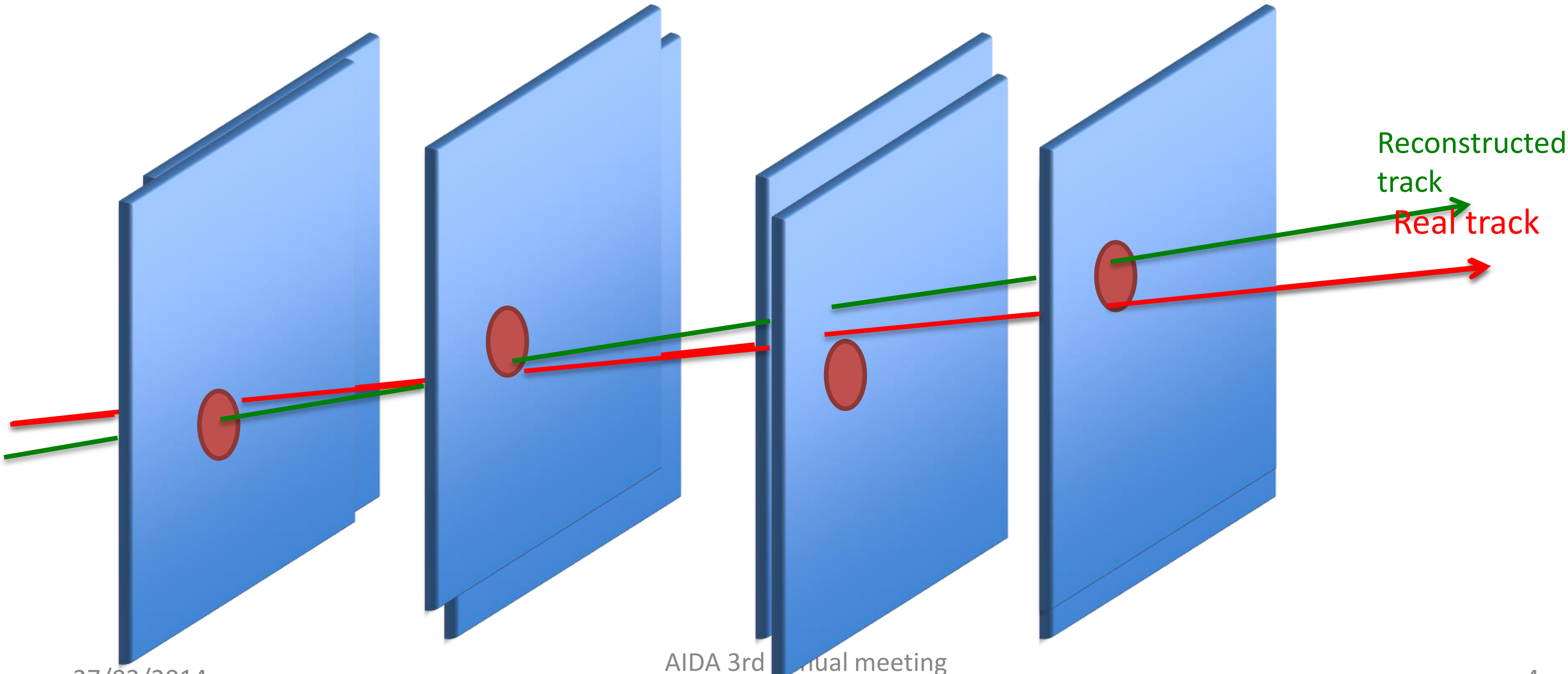
A new **software-package for telescope detectors** is in development, aiming to provide a standalone code that can be easily modified and used for various detector designs.

All necessary steps to do an analysis of testbeam data (e.g. clustering, pattern recognition, tracking, alignment, monitoring) will be provided to give a complete example. The WP2 Alignment Tools and Geometry Package (DD4HEP) will be included in this software. A user-friendly interface to implement customized algorithms will be provided.

- Introduction
  - Alignment Problem & Strategies

# Alignment Introduction

- Track leaves hits on sensors
- **Alignment problem:** Detector positions used in offline reconstruction do not correspond to the actual relative positions of the installed detector
  - Hit positions are misplaced
- Reconstructed tracks are biased
  - Can lead to inefficient/wrong physical conclusions



# Solutions to the alignment problems

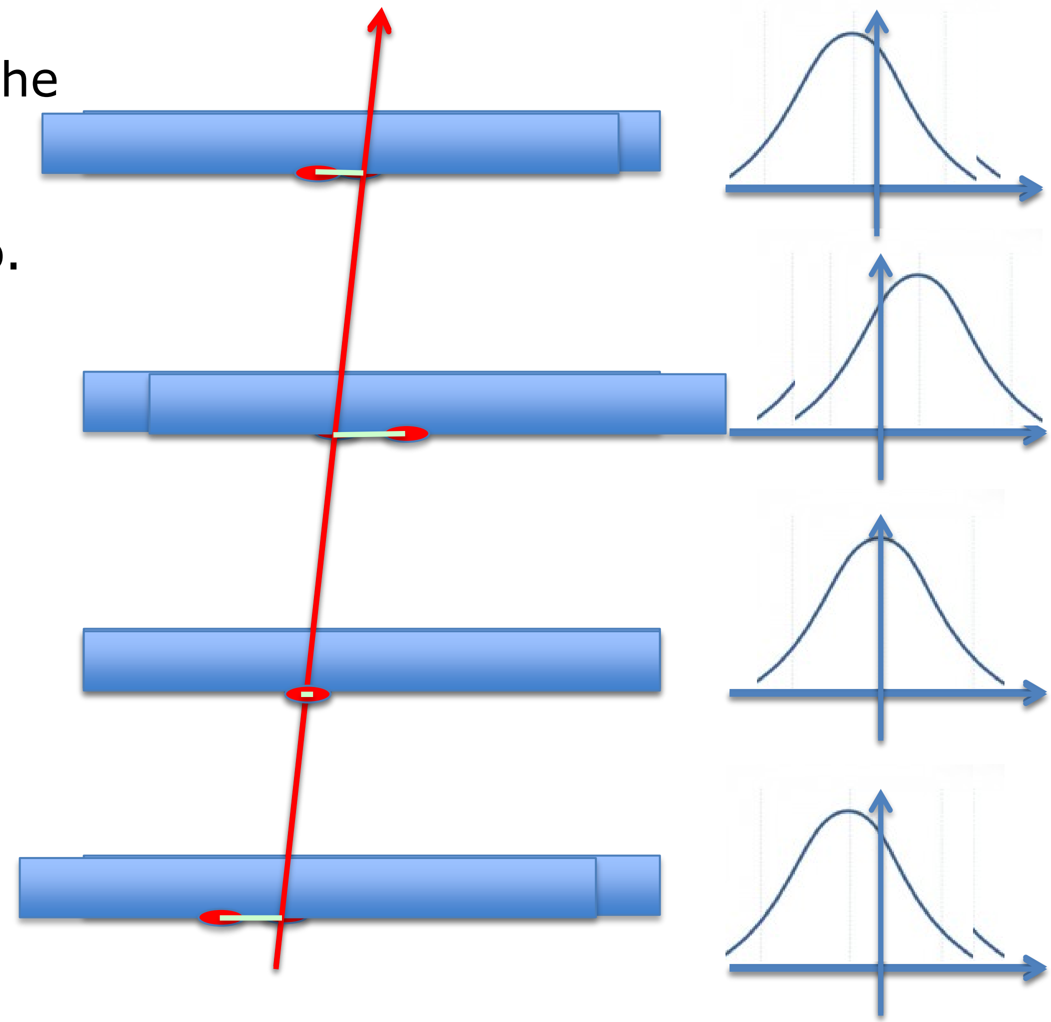
- **Assembly / survey measurements**
- External measurements of mounting positions
- Measurements during / after installation

- **Offline track based alignment algorithm**
- Use local track parameter to determine global alignment parameter
- Should be robust, stable and not too time consuming
- Precision of alignment parameter should be known to an order of detector resolution

- **Will concentrate on track based alignment**

# Track based alignment

- In a perfect aligned detector the residual-distribution (distance between measured hit and track) is centered around zero.
- Distribution gets shifted / spread, when modules are misaligned.
- Residual depends on
  - *local* (track) parameter
  - *global* (alignment) parameter.



# MILLEPEDE approach

Define:

Measured value — Reconstructed hit position

$$\chi^2 = \sum_{k=0}^{n} \frac{\left(z_k - \mathbf{a}^T \mathbf{d}_k\right)^2}{\sigma_k^2}$$

$\chi^2$ Is an explicit function of the **alignment parameters α.**

- It has a minimum at the true values of the alignment parameters
- Minimisation can be written in matrix-form.
- Alignment problem gets solved by **inverting a large matrix.**
- MILLEPEDE (by V. Blobel) is an algorithm, that can invert large matrices fast.
- Simultaneous fit of global and local parameters.

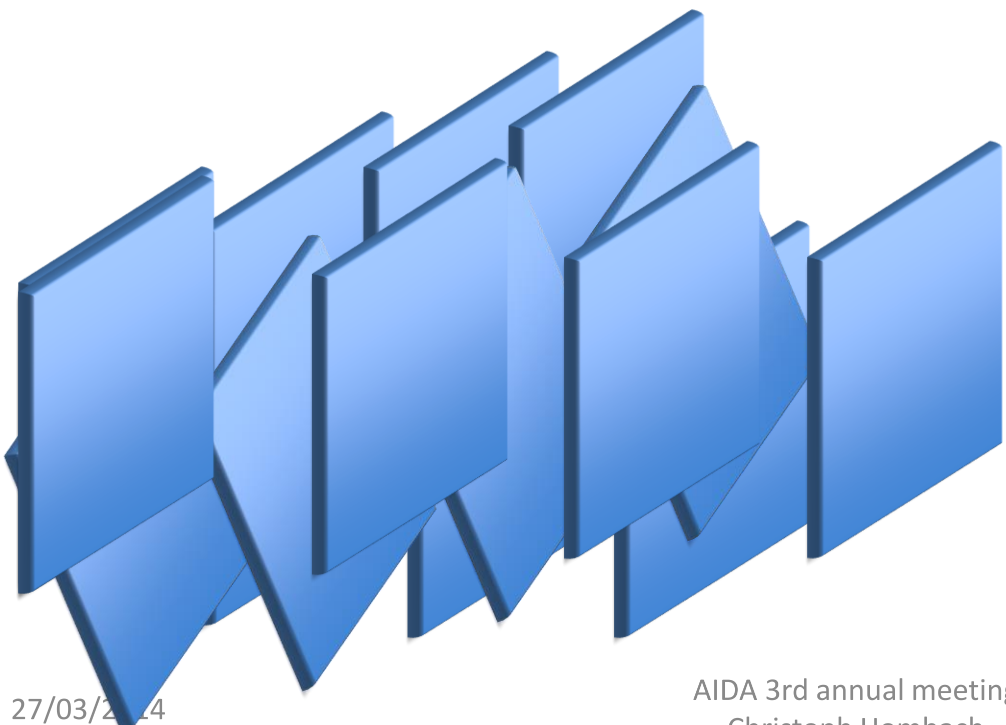
**A New Method for the High-Precision Alignment of Track Detectors,** Volker Blobel and Claus Kleinwort, Report DESY 02-077 (June 2002)

# Weak modes

- Detector deformations, that have **no impact on** $\chi^2$
- Solution is blind to multiple minima

For parallel tracks:

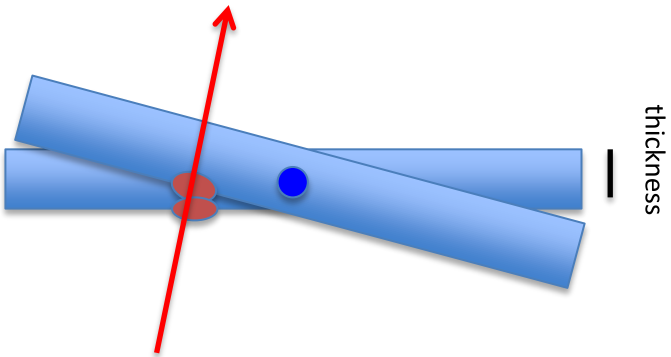
Linear transformations:
- Translation
- Rotation
- Shearing
- Scaling



# Dealing with weak mode

- Constraints, like fixing module position
- Constraint-equations
  - additional terms to $\chi^2$ that depend on alignment parameter
- Use set of tracks with different characteristics

- Some DOF more sensitive than others !
- One has to study impact on resolution.
- Could gain more sensitivity with carefully chosen set of tracks.



# AIDA Alignment Overview

The University of Manchester

- **AIDA Alignment web page**

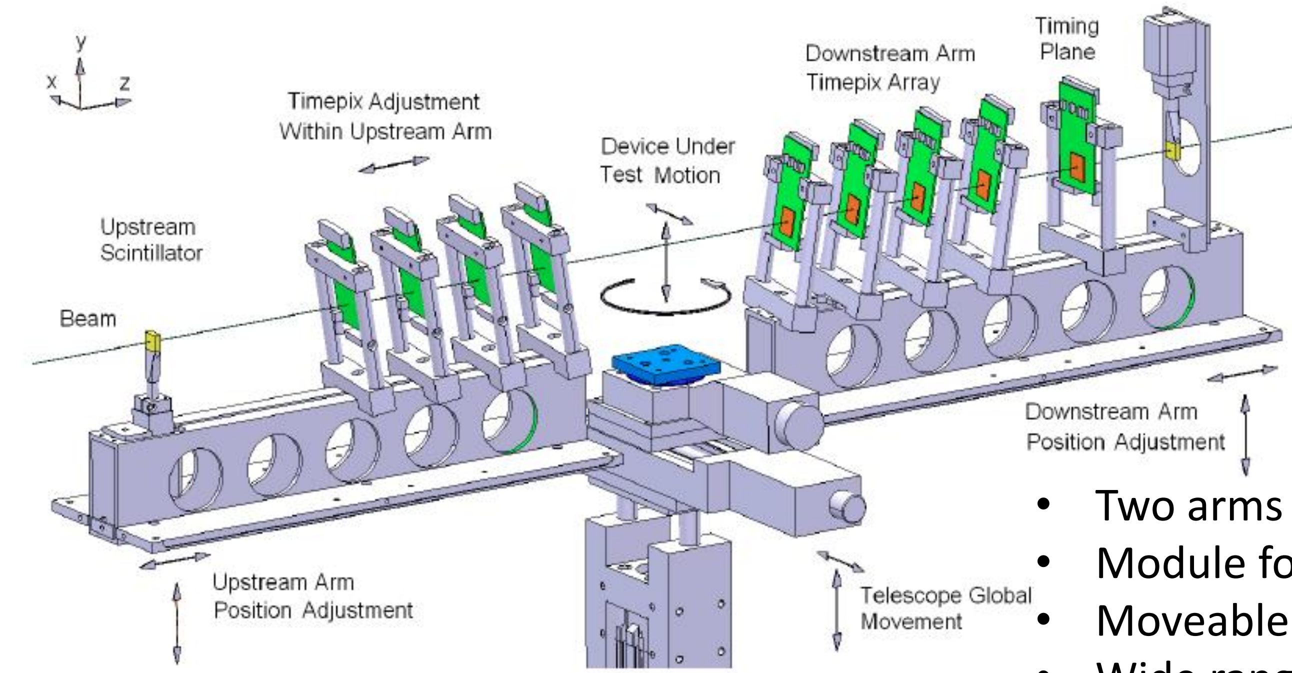
  http://aidasoft.web.cern.ch/node/31

  – Documents alignment papers /methods
    - Current/recent major particle physics experiments

- Testbeam
  - AIDA Testbeam Telescope
- LHCb VELO Alignment

# TimePix Telescope

The University of Manchester




- Two arms with four modules.
- Module for time measurements.
- Moveable DUT centered between arm.
- Wide range of users

- Results led to pixel decision for VELO upgrade (TDR CERN-LHCC-2013-021)

- **AIDA alignment software implemented and supported**
  - Software depended on Telescope geometry

The Timepix Telescope for High Performance Particle Tracking

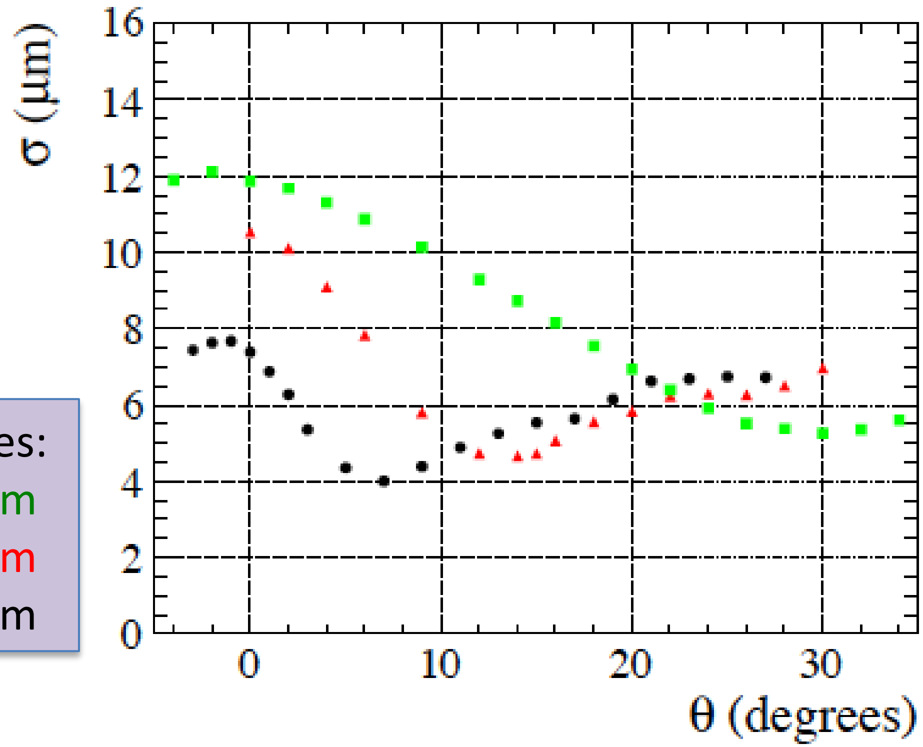
K. Akiba[a], P. Arne-Ronning[b], M. van Beuzekom[c], V. van Beveren[c], S. Borghi[d], H. Boterenbrood[c], J. Buytaert[b], P. Collins[b], A. Dosil Suárez[e], R. Dumps[b], L. Eklund[g], D. Esperante[e], A. Gallas[e], H. Gordon[f], B. van der Heijden[c], C. Hombach[d], D. Hynds[g], M. John[f], A. Leflat[h], Y. Li[f], I. Longstaff[g], A. Morton[g], N. Nakatsuka[b], A. Nomerotski[f], C. Parkes[d], E. Perez Trigo[e], R. Plackett[i,*], M. Reid[j], P. Rodriguez Perez[e], H. Schindler[b], T. Szumlak[k], P. Tsopelas[c], C. Vázquez Sierra[e], J. Velthuis[l], M. Wysokiński[k]

AIDA-PUB-2014-003

# Testbeam Results

The University of Manchester

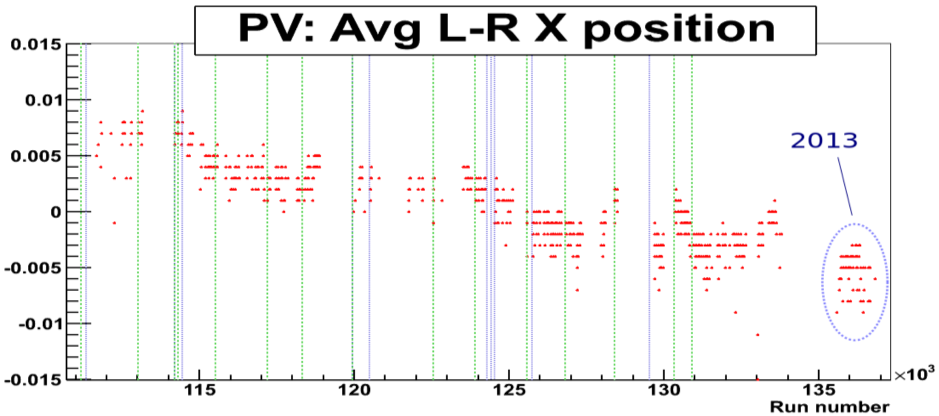
- High resolution obtained from 55x55μm pixel detectors
- Resolution studied as function of sensor thickness, voltage, processing technology
- Paper in preparation

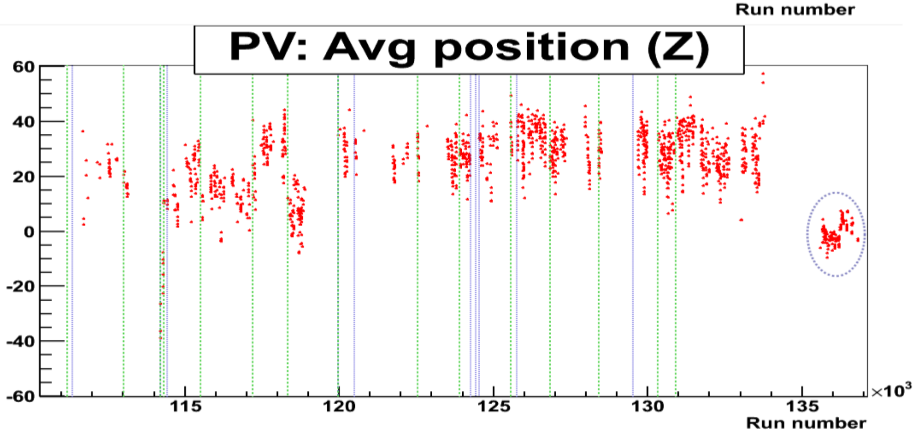


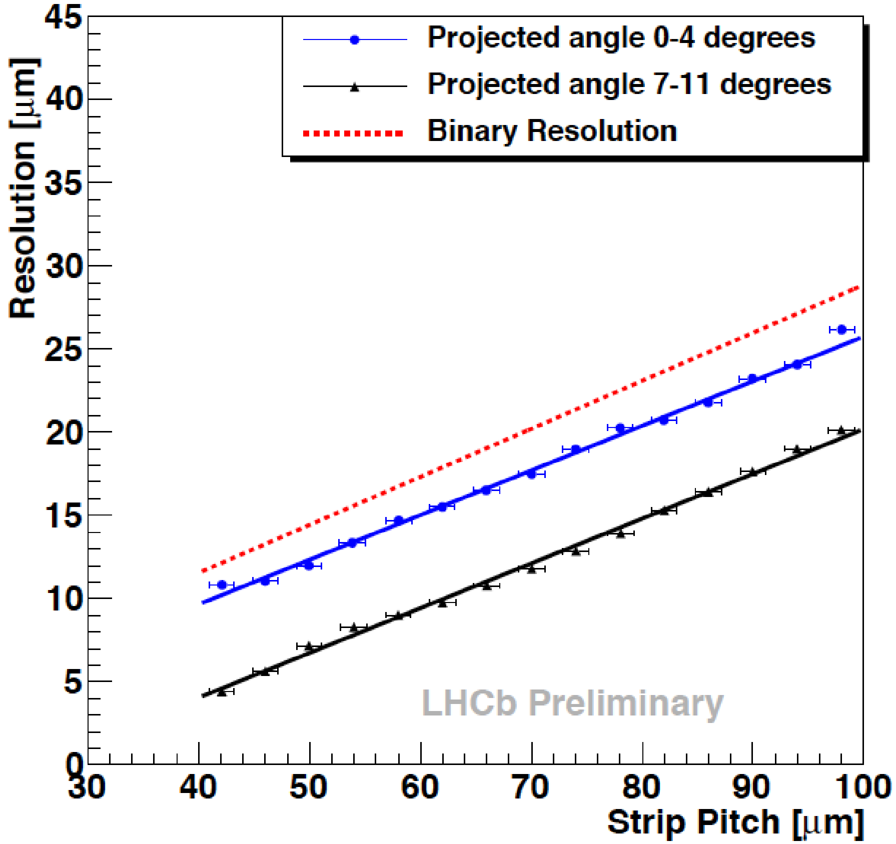

Thicknesses:
300μm
200μm
100μm

# LHCb VELO

- VELO resolution down to ~4 micron
- Low $\chi^2$ sensitivity modes
  - Effects vertex-reconstruction
  - Controlling these effects



- BACH
  - Standalone reconstruction & **alignment** software

# BACH

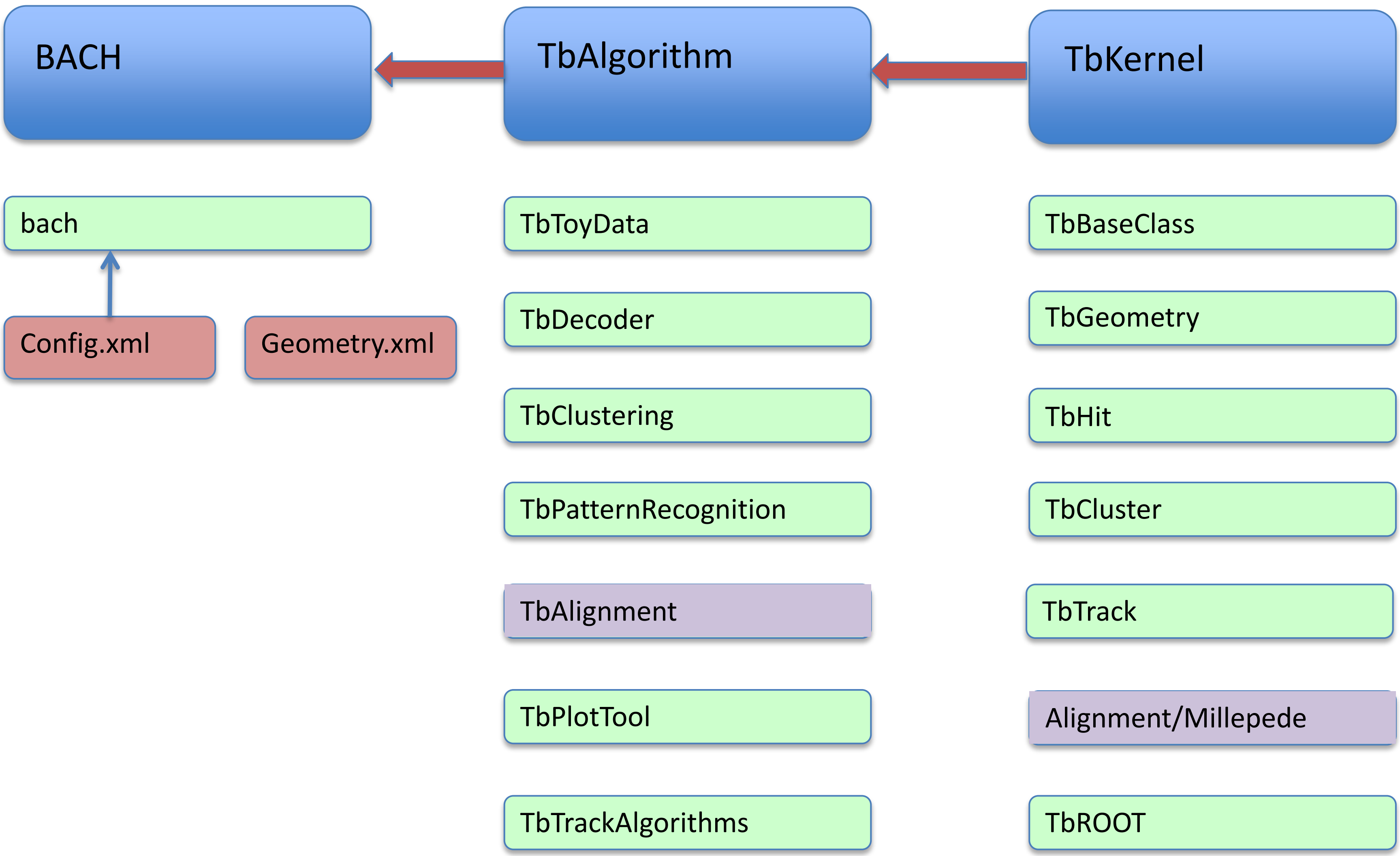
- **BACH** (Basis of Alignment CHain)
- Standalone software, only depends on ROOT and BOOST

- Designed to test and verify alignment algorithms
  - Provides new users a development framework

- Example based on telescope detector design
  - Simple geometry configurable

- Includes complete analysis chain:
- -> Simulation -> Clustering
  -> Pattern Recognition -> Track Fit -> **Alignment**

# BACH software layout



# BACH software layout

- Algorithms and corresponding constants are called in configuration.xml
- Geometry (modules, relative positions, alignment constants) is defined in geometry.xml
- Alignment algorithm produces new geometry.xml

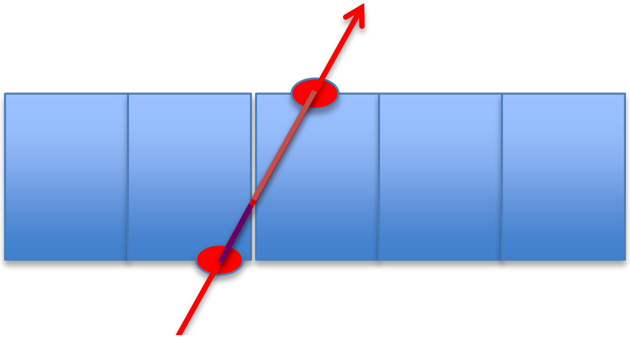
```xml
<?xml version="1.0" encoding="utf-8"?>
<Algorithms NoOfEvt="1000" >



  <TbGeometrySvc>
    <constant name='GeometryFile'  value="geom/Misalign_geom.xml" type="S"/>
    <constant name='PitchX'        value="0.055"                  type="D"/>
    <constant name='PitchY'        value="0.055"                  type="D"/>
    <constant name='NoOfPixelX'    value="256"                    type="I"/>
    <constant name='NoOfPixelY'    value="256"                    type="I"/>
    <constant name='Thick'         value="0.3"                    type="D"/>

  </TbGeometrySvc>

  <TbToyData>
    <constant name="NoOfTracks"    value="25"                      type="I"/>
    <constant name="GeometryFile"  value="geom/Telescope_geom.xml" type="S"/>
    <constant name='MeanX'         value="0."                      type="D"/>
    <constant name='MeanY'         value="0."                      type="D"/>
    <constant name='SigmaX'        value="0.001"                   type="D"/>
    <constant name='SigmaY'        value="0.001"                   type="D"/>
  </TbToyData>
```

```xml
<?xml version="1.0" encoding="utf-8"?>

<Geometry>

        <Module name='C09-W0108' X='0' Y='0' Z='0' RX='0.157' RY='0.157' RZ='0' dX='-0.00591273' dY='0.00137021' dZ='0' dRX='1.46175e-05' dRY='4.80728e-06' dRZ='-6.6633e-05' />
        <Module name='C10_W0108' X='0' Y='0' Z='23' RX='0.157' RY='0.157' RZ='0' dX='-0.0056934' dY='0.00145046' dZ='0' dRX='-3.0659e-06' dRY='-2.76501e-06' dRZ='-6.23762e-05' />
        <Module name='D09-W0108' X='0' Y='0' Z='310' RX='0.157' RY='0.157' RZ='0' dX='0' dY='0' dZ='0' dRX='0' dRY='0' dRZ='0' />
```

# Algorithms



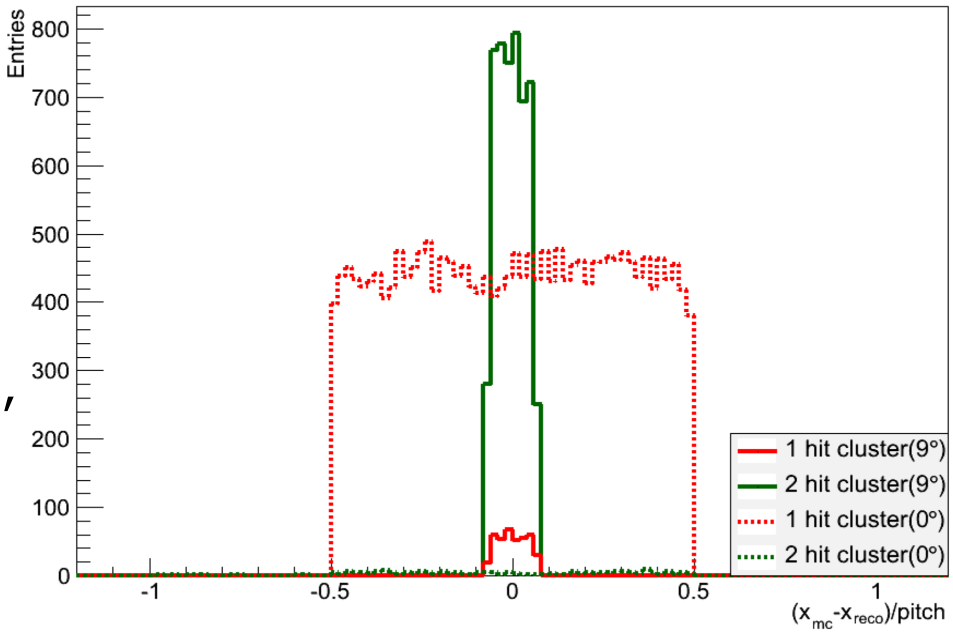
- Simulation:
  - Create random track
  - Determine entry and exit point
  - Allocate charge to each pixel relative to distance, that track passes the pixel



- Clustering:
  - Use centre of gravity (CoG) method

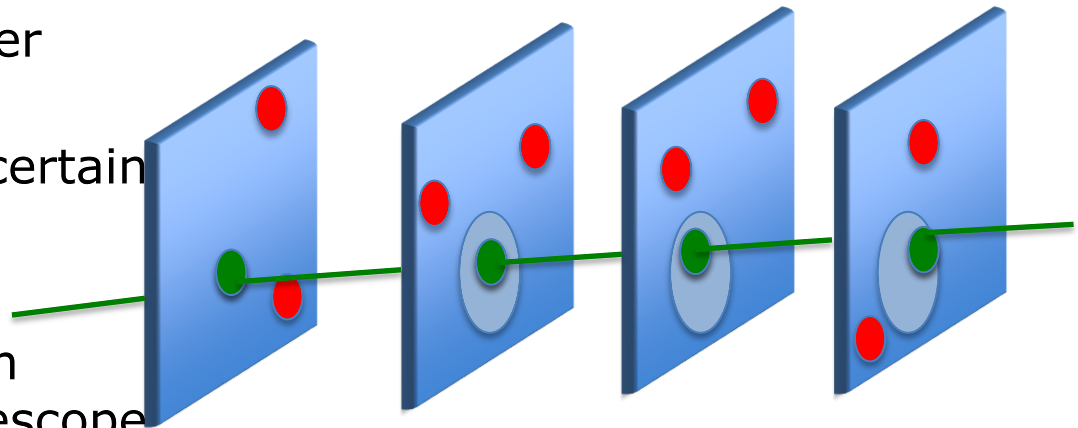
$$\bar{x} = \frac{\sum_i x_i adc_i}{\sum_i adc_i}$$

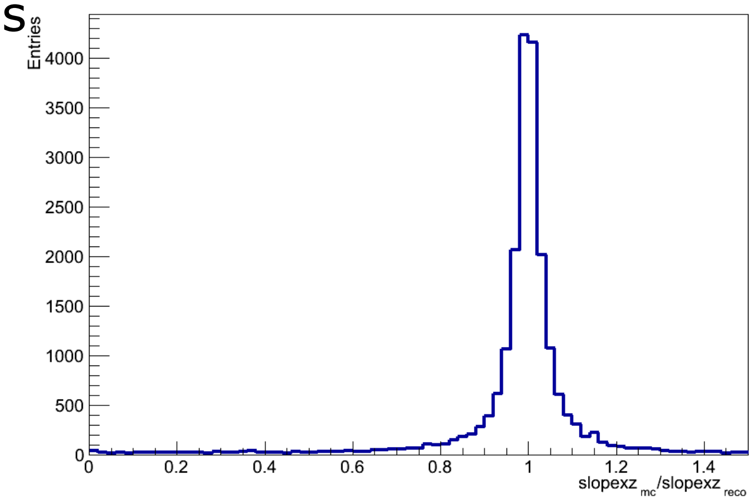
  - If track traverse only one pixel, the centre of the pixel corresponds to cluster position
  - Most precise result for 2 pixel cluster

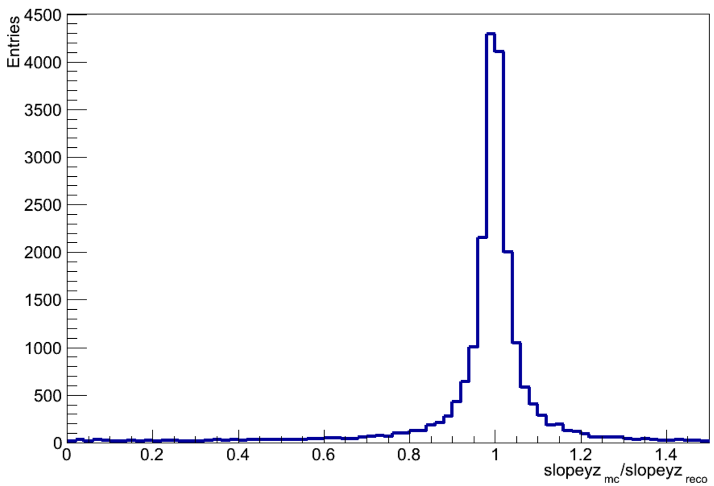


# Algorithms

- **Pattern Recognition**
- Start with a seeding cluster from reference module
- Search for cluster within certain radius on other modules
- Fit Track
- Simple pattern recognition approach, suitable for telescope purposes



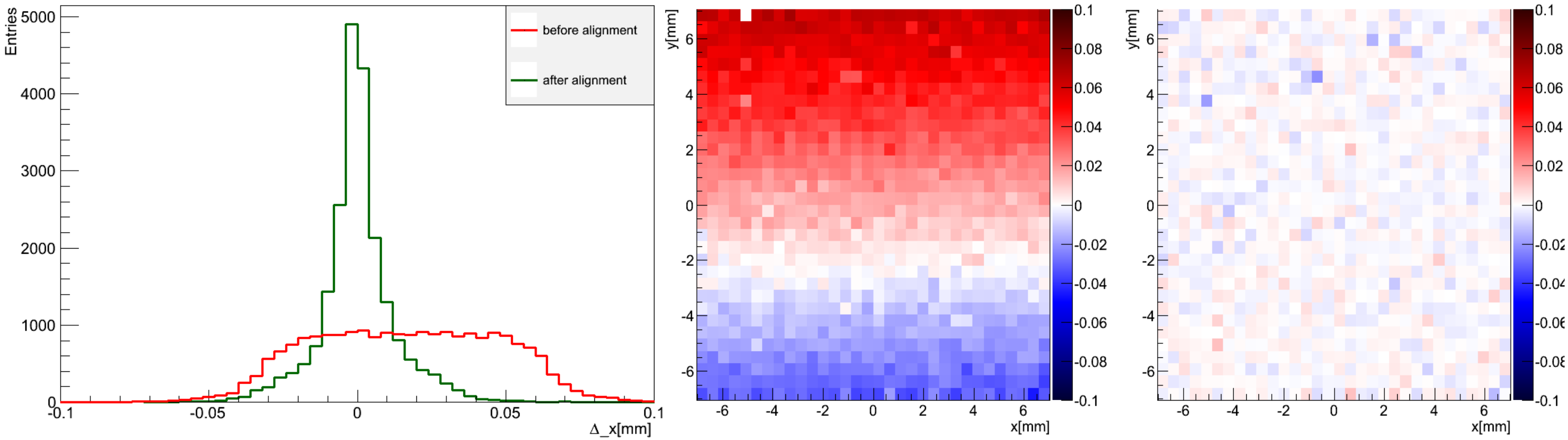


The University of Manchester

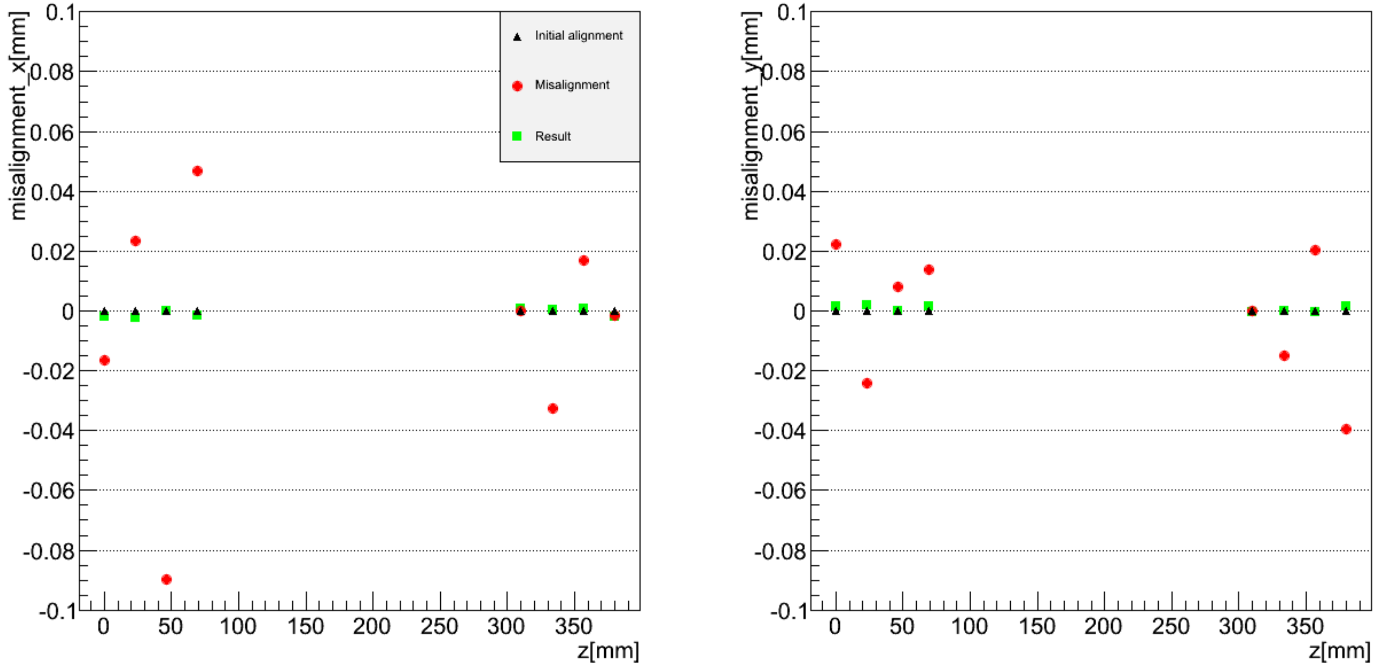
# • Alignment Algorithm

- Feed MILLEPEDE with cluster positions associated with tracks
- Constraints: Fix one module in fit (Defines reference frame)
  Use lagrange muliplier constraints utilising the fact tracks are parallel to z-axis on average
- Fit result are alignment parameter




- Root file histograms for alignment monitoring provided

# Alignment Results




- Alignment algorithm is able to cope with large misalignments
- Difference between determined and true alignment in order of a few micron

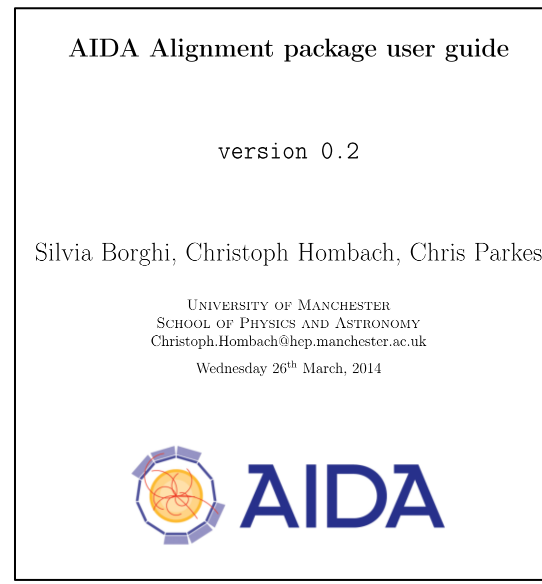
# Summary

- **AIDA Alignment Overview Web Page**
  - http://aidasoft.web.cern.ch/node/31

- **AIDA telescope alignment**
  - Supported range of users
  - First DD4HEP use for real experiment
  - LHCb VELO Upgrade made technology choice

- **LHCb VELO alignment**
  - Improvements for weak mode constraints
  - Future: implement VELO upgrade alignment

- **BACH – standalone** software package for reconstruction and **alignment** for telescope detector geometry
  - Committed to svn, documentation in preparation
  - Future: Implement DD4HEP



AIDA Alignment package user guide

version 0.2

Silvia Borghi, Christoph Hombach, Chris Parkes

University of Manchester
School of Physics and Astronomy
Christoph.Hombach@hep.manchester.ac.uk

Wednesday 26th March, 2014