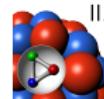


# Concept of the $K_S^0$ Rescue System for the Belle II Pixel Detector

Leonard Koch, Wolfgang Kühn, Sören Lange, and David Münchow



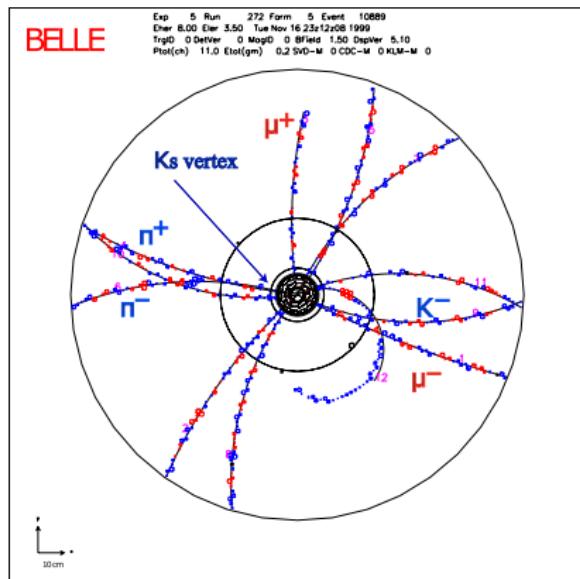
II. Physikalisches  
Institut



Bundesministerium  
für Bildung  
und Forschung

January 30<sup>th</sup> 2015 / Bormio  
53<sup>th</sup> International Winter Meeting on Nuclear Physics

# Motivation



## B physics

- CP violation: eg.  $B^0 \rightarrow J/\psi K_S^0$
- Flavor Changing Neutral Currents: eg.  $b \rightarrow s\gamma$  ( $B^0 \rightarrow K_S^0 \pi^0 \gamma$ , see talk S. Lange)
- $b \rightarrow c \rightarrow s$

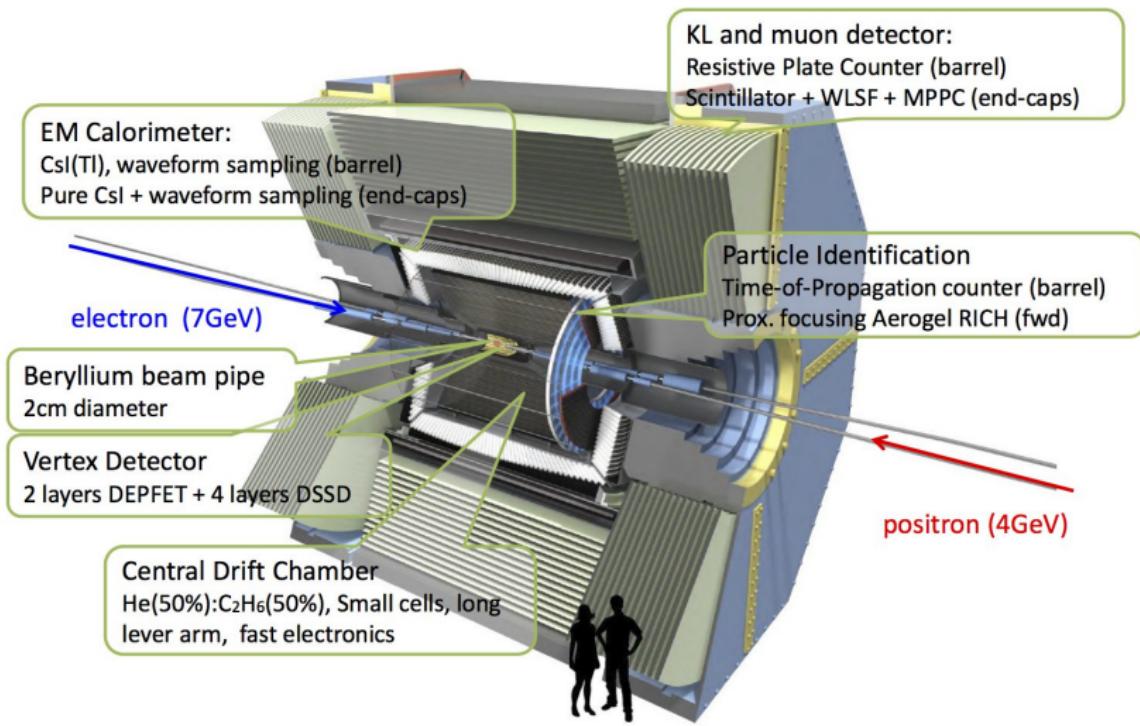
$K_S^0 \rightarrow \pi^+ \pi^-$  are abundant in B meson decays

# SuperKEKB accelerator at KEK

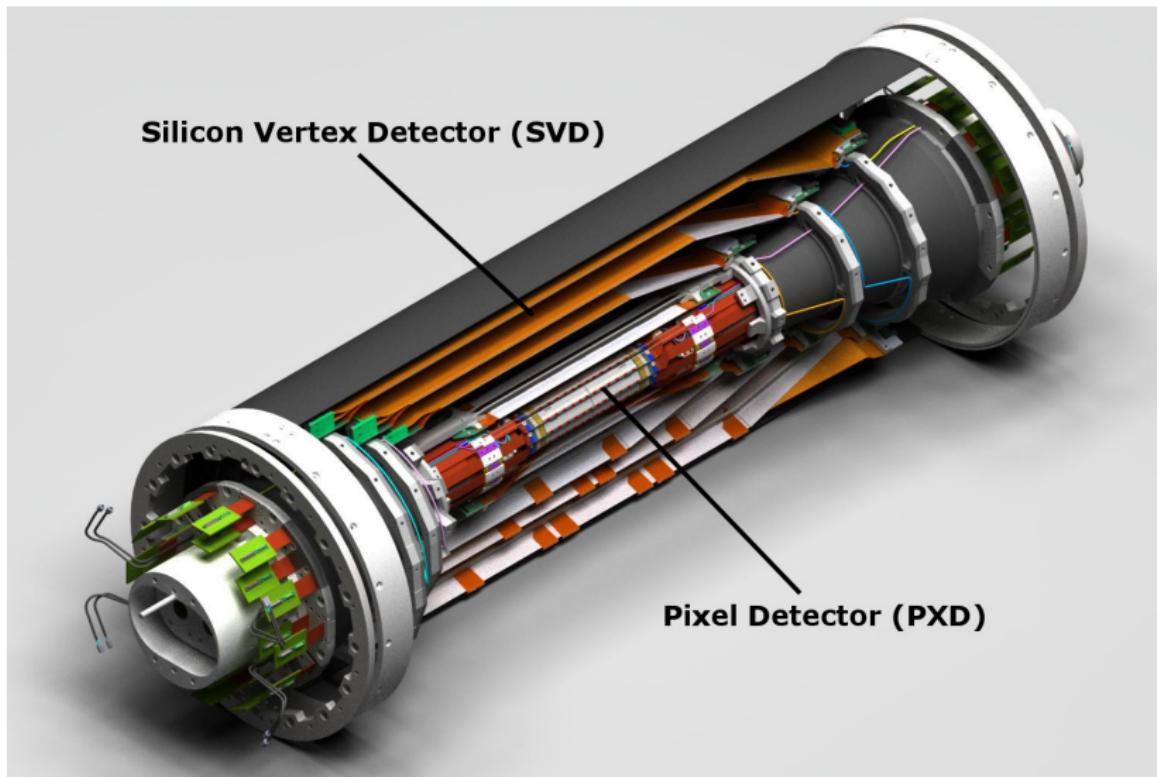
$e^+ e^- \rightarrow \Upsilon(4S) \rightarrow B\bar{B}$  @ $\sqrt{s} = 10.58 \text{ GeV}$



# Belle II Detector

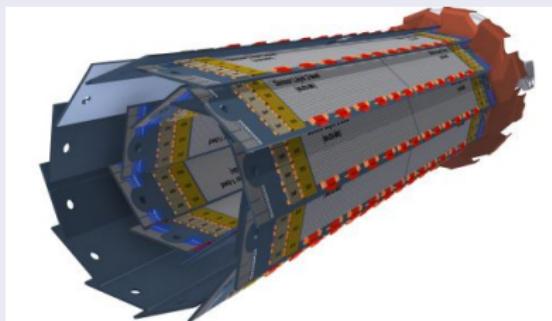


# The Vertex Detector (VXD)



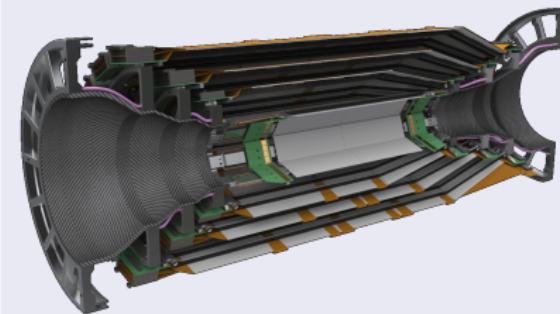
# The Vertex Detector (VXD)

PXD



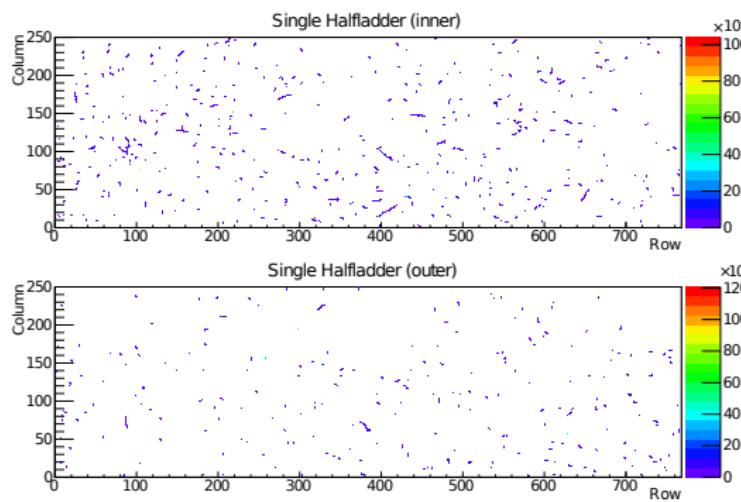
- Radius: 14 - 22 mm
- 2 layers
- $\sim 8$  Mio pixels
- DEPFET technology (see talk J. Ninkovic)

SVD



- Radius: 38 - 140 mm
- 4 layers
- Double sided silicon strip detector

# PXD background



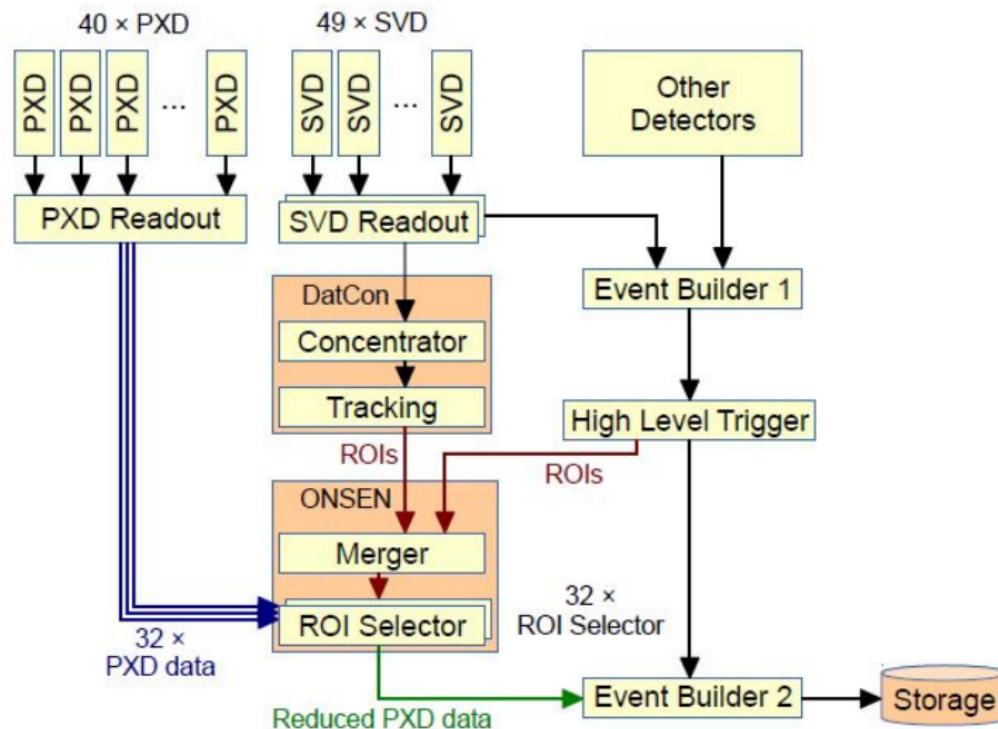
Integration time: 20  $\mu$ s

## Background sources

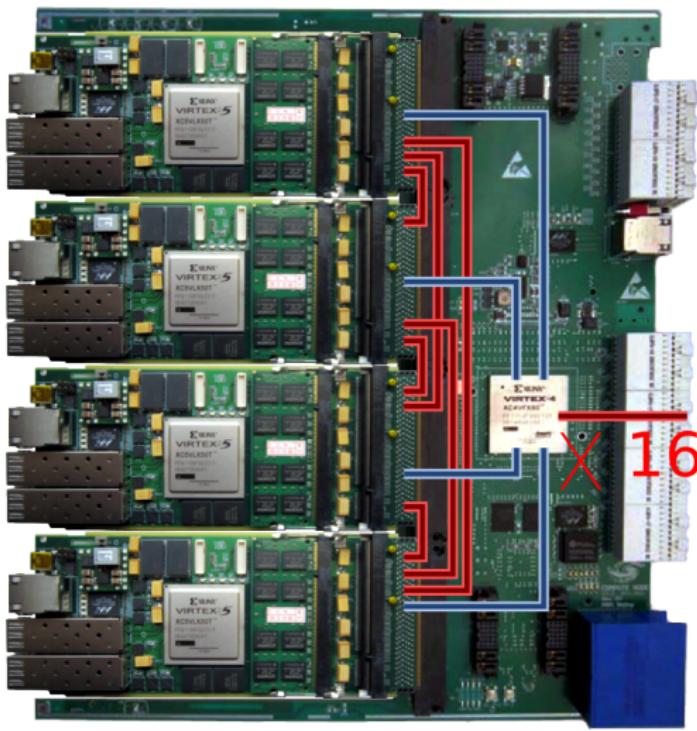
- Beam - gas scattering
- Synchrotron radiation
- Touschek effect
- Radiative Bhabha
- Two photon QED

$\Rightarrow \leq 3\%$  occupancy  
 $\Rightarrow \sim 20\text{ GB/s}$  PXD data output ( $\sim 10\times$  output of rest of detector)

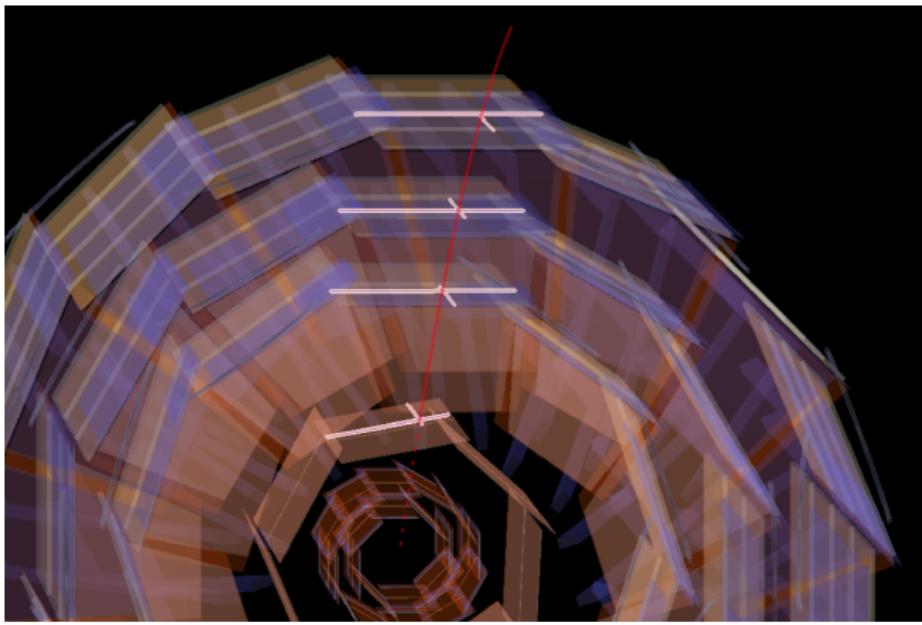
# PXD Readout



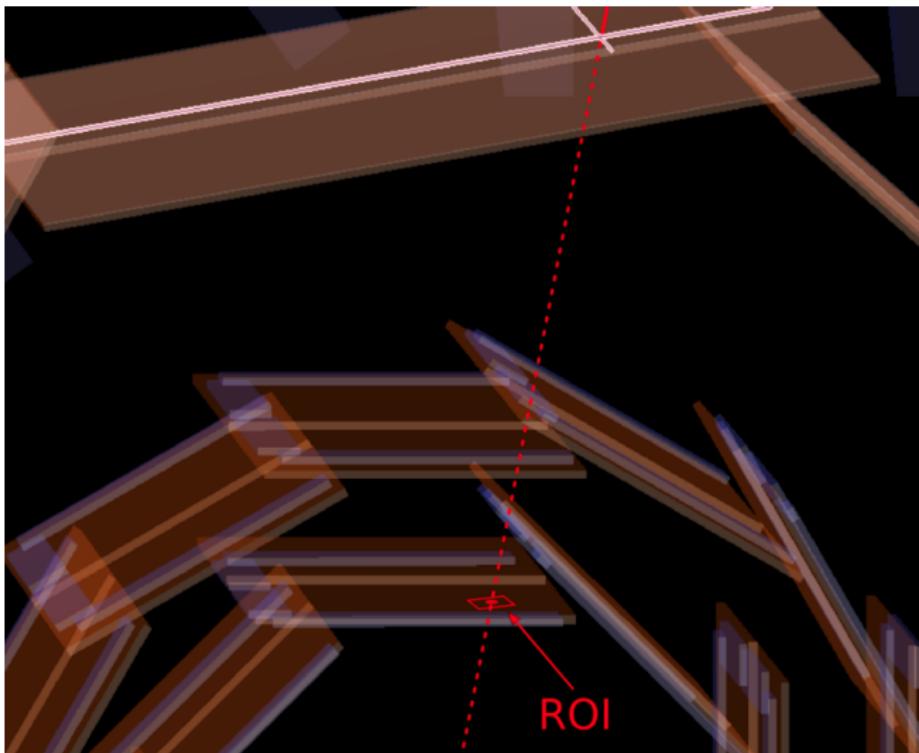
# PXD Readout



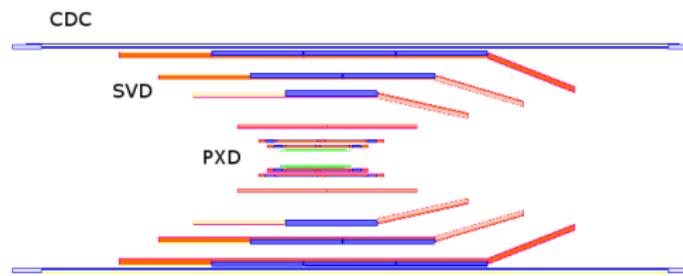
# Regions Of Interest (ROI)



# Regions Of Interest (ROI)

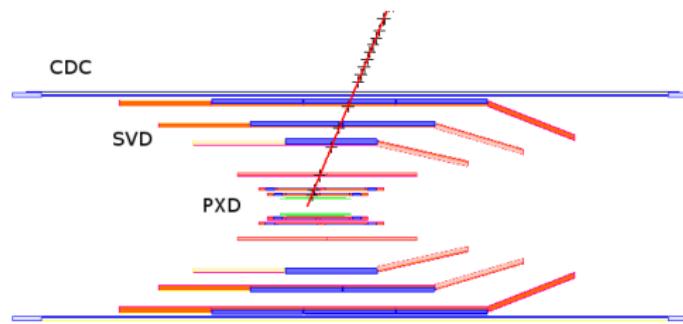


# ROI or no ROI



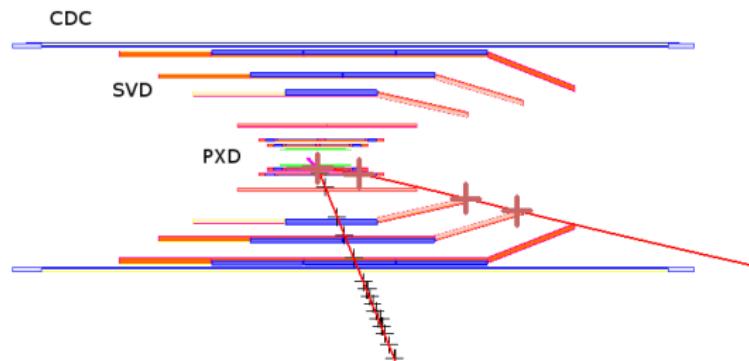
- 'Good' track: ROI
- Low  $p \rightarrow$  high  $dE/dx$  (Bethe-Bloch):  $dE/dx$  Cluster Rescue saves clusters with high seedcharge
- $K_S^0$  ( $c\tau = 2.68\text{ cm}$ ): Displaced vertex
- 1% of all  $K_S^0$  in inclusive  $B$  decays ( $\sim 50\%$  of whole Belle data set): No ROI
- Still trackable with full VXD

# ROI or no ROI



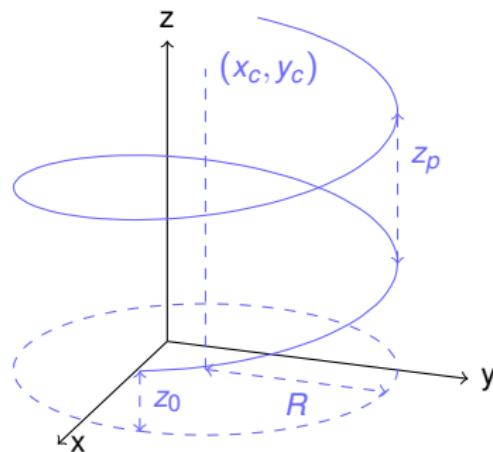
- 'Good' track: ROI
- Low  $p \rightarrow$  high  $dE/dx$  (Bethe-Bloch):  $dE/dx$  Cluster Rescue saves clusters with high seedcharge
- $K_S^0$  ( $c\tau = 2.68\text{ cm}$ ): Displaced vertex
- 1% of all  $K_S^0$  in inclusive  $B$  decays ( $\sim 50\%$  of whole Belle data set): No ROI
- Still trackable with full VXD

# ROI or no ROI



- 'Good' track: ROI
- Low  $p \rightarrow$  high  $dE/dx$  (Bethe-Bloch):  $dE/dx$  Cluster Rescue saves clusters with high seedcharge
- $K_S^0$  ( $c\tau = 2.68$  cm): Displaced vertex
- 1% of all  $K_S^0$  in inclusive  $B$  decays ( $\sim 50\%$  of whole Belle data set): No ROI
- Still trackable with full VXD

# Online Tracking with full VXD



## Parameters of a helix

- Radius:  $R$
- Center:  $(x_c, y_c)$
- Pitch:  $z_p$
- Z-offset:  $z_0$

Platform:

FPGA

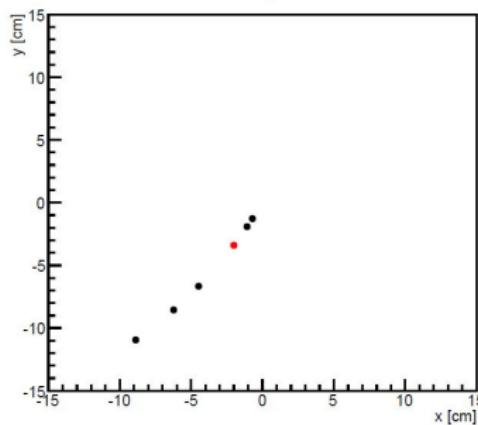
Track finding:

- ➊ X-Y plane:
  - ➊ Conformal transformation
  - ➋ Fast hough transformation
- ➋ Validate track by reconstructing z-component (fast hough transformation)

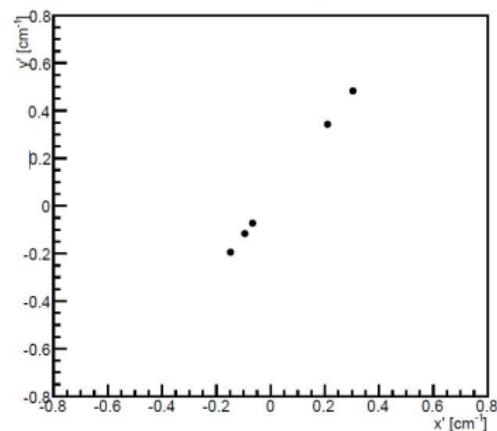
# Conformal Transformation

Convert a circle into a straight line

Real Space



Conformal Space



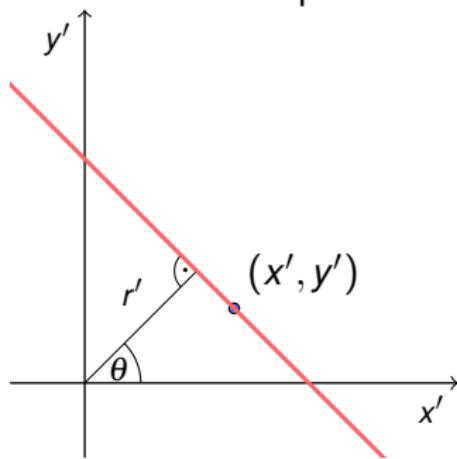
$$x \rightarrow x' = \frac{x - x_0}{r^2}$$

$$y \rightarrow y' = \frac{y - y_0}{r^2}$$

$$r^2 = (x - x_0)^2 + (y - y_0)^2$$

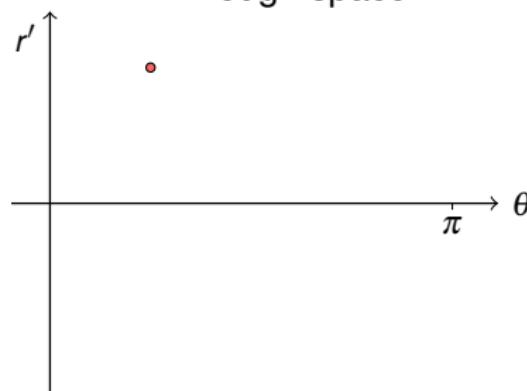
# Hough Transformation

Conformal space



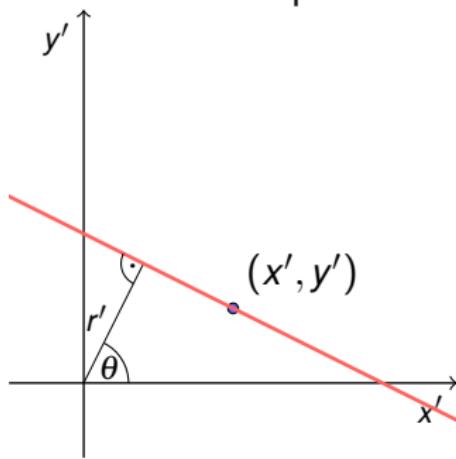
$$r' = x' \cos \theta + y' \sin \theta$$

Hough space



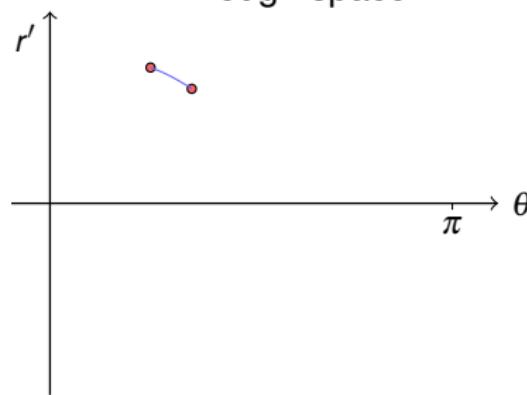
# Hough Transformation

Conformal space

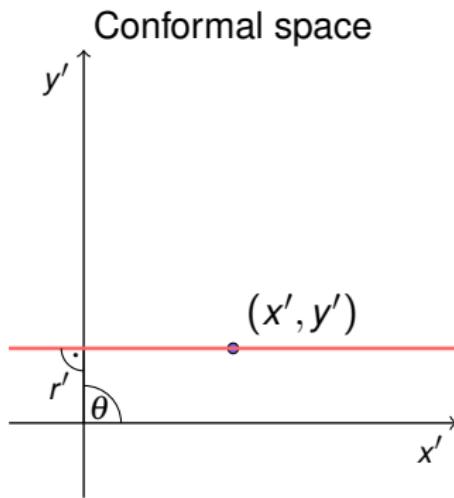


$$r' = x' \cos \theta + y' \sin \theta$$

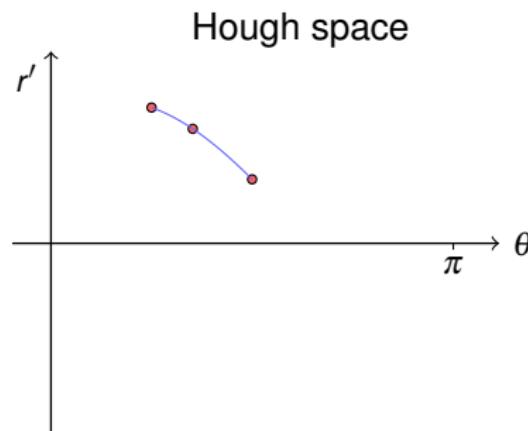
Hough space



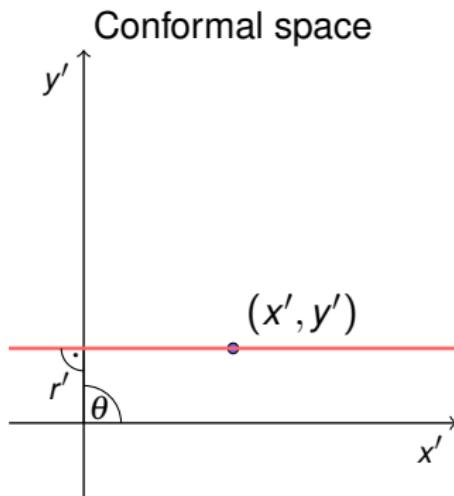
# Hough Transformation



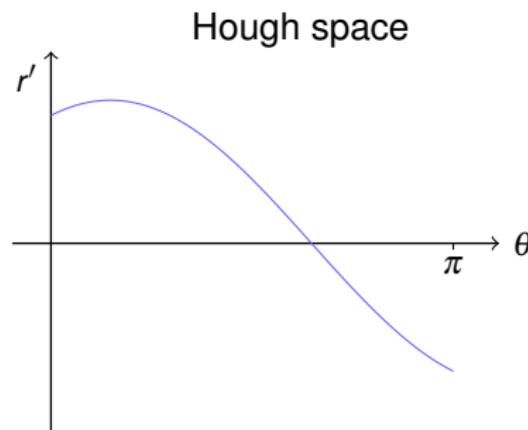
$$r' = x' \cos \theta + y' \sin \theta$$



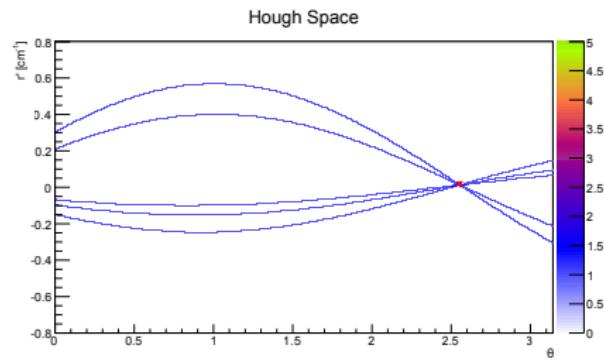
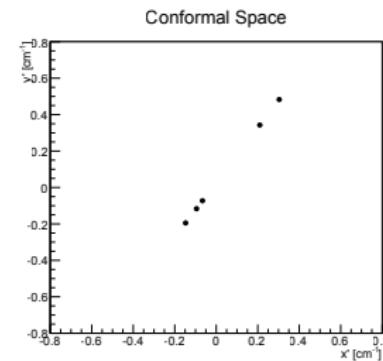
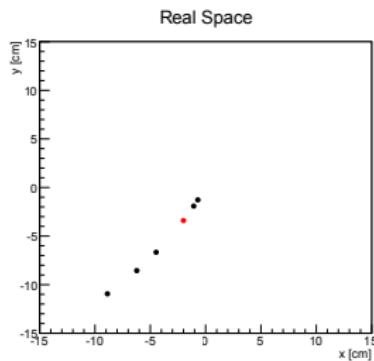
# Hough Transformation



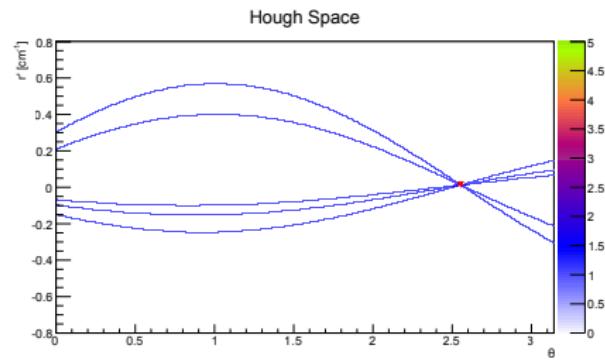
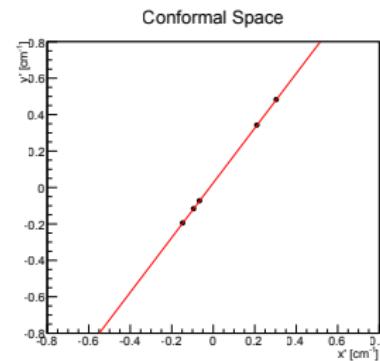
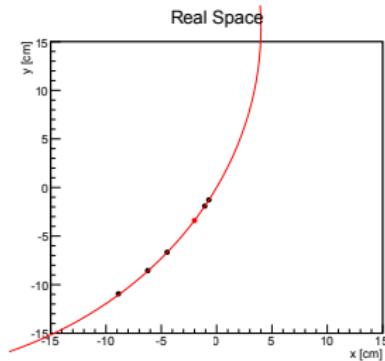
$$r' = x' \cos \theta + y' \sin \theta$$



# Hough Transformation

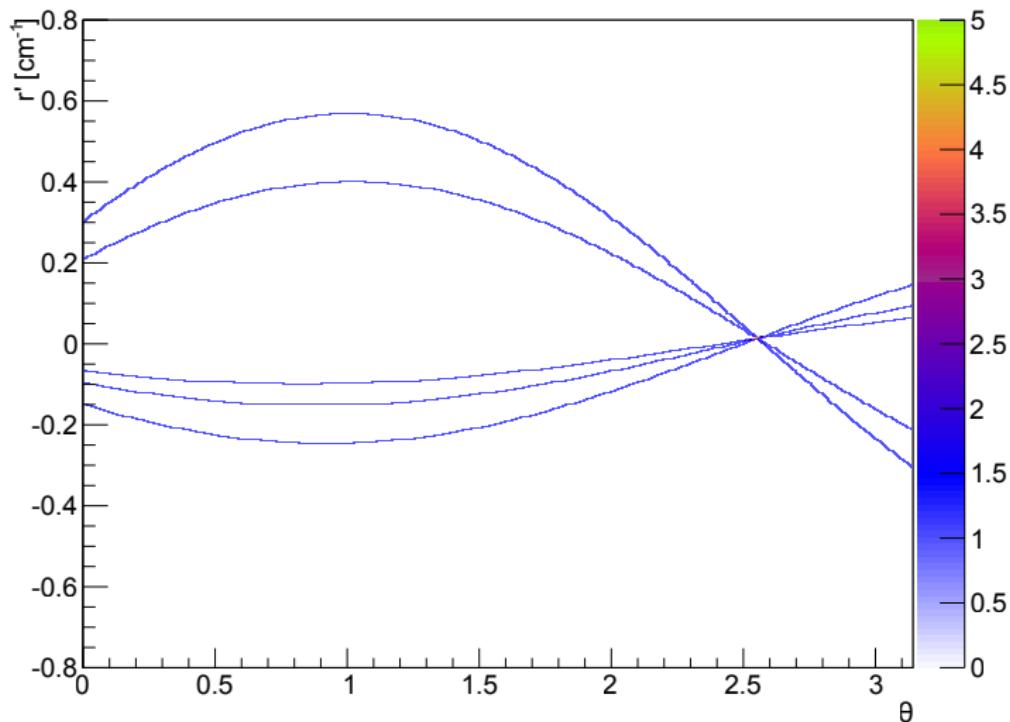


# Hough Transformation



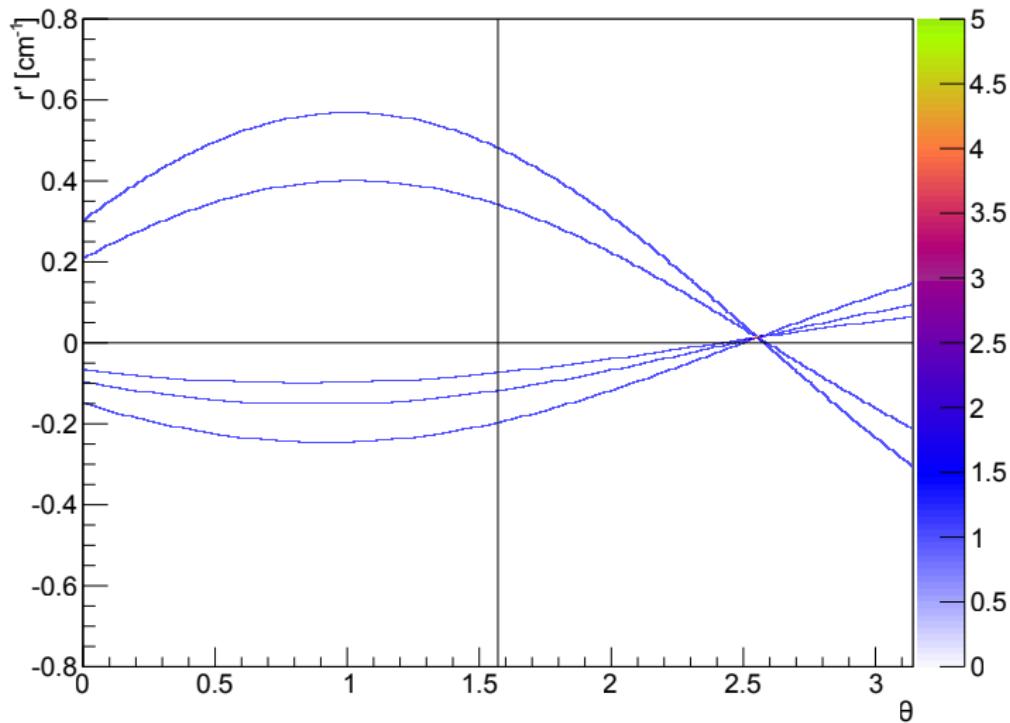
# Fast Hough Transformation

Hough Space



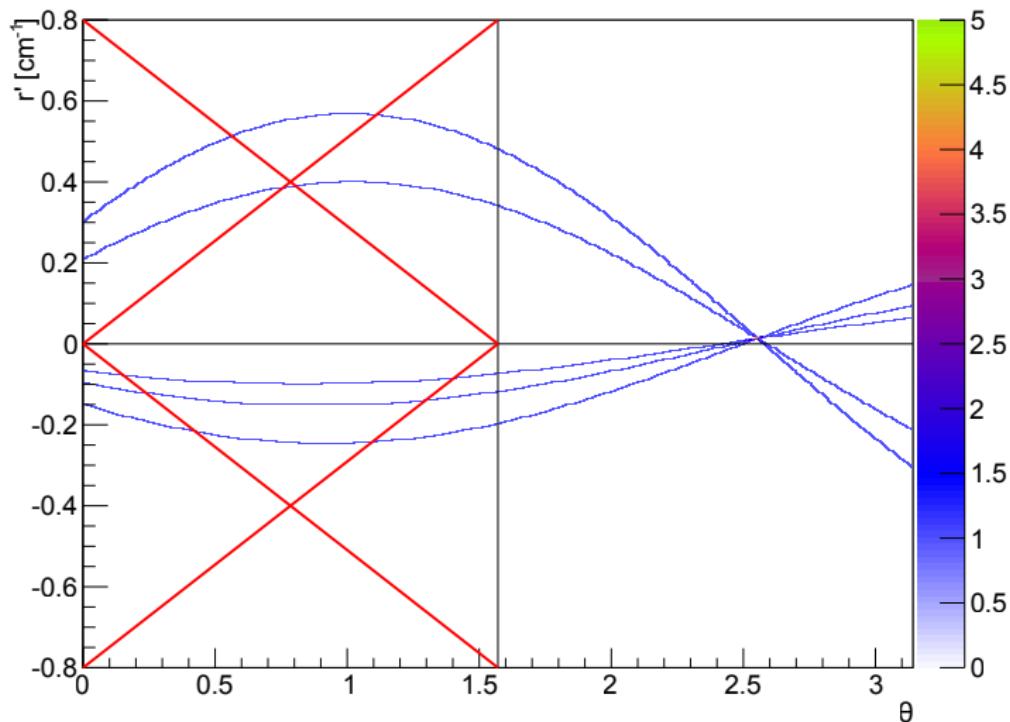
# Fast Hough Transformation

Hough Space



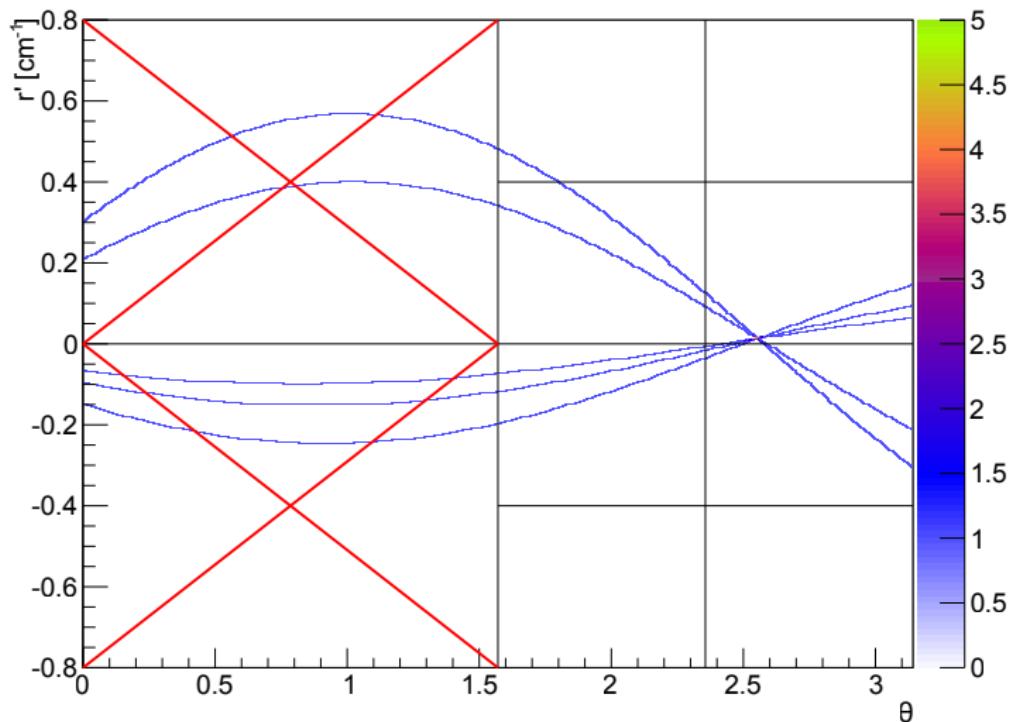
# Fast Hough Transformation

Hough Space



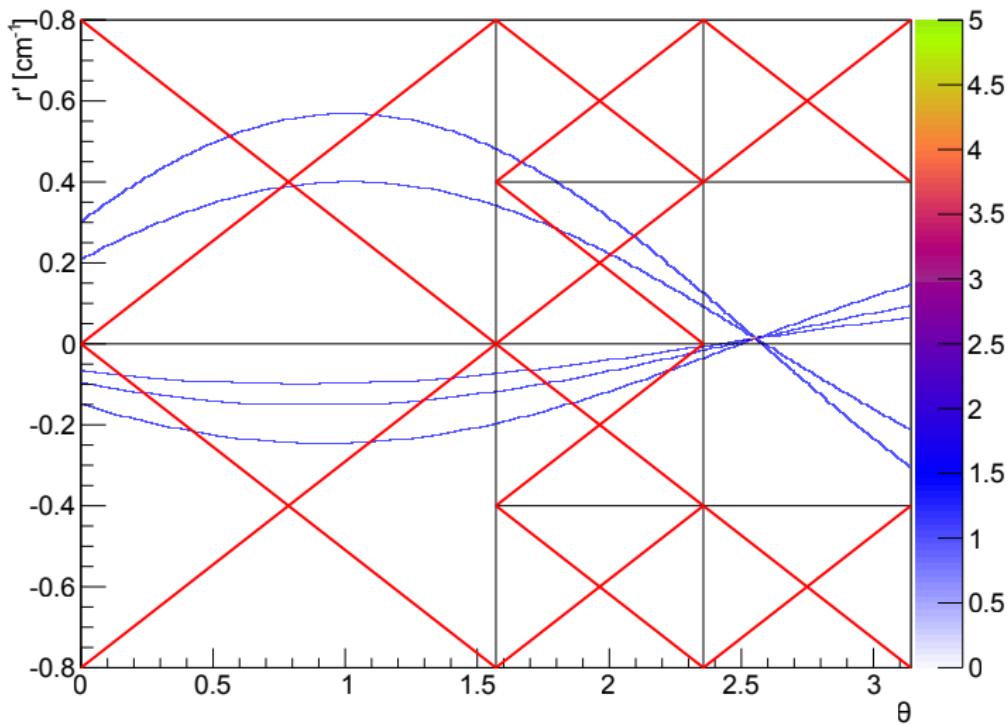
# Fast Hough Transformation

Hough Space



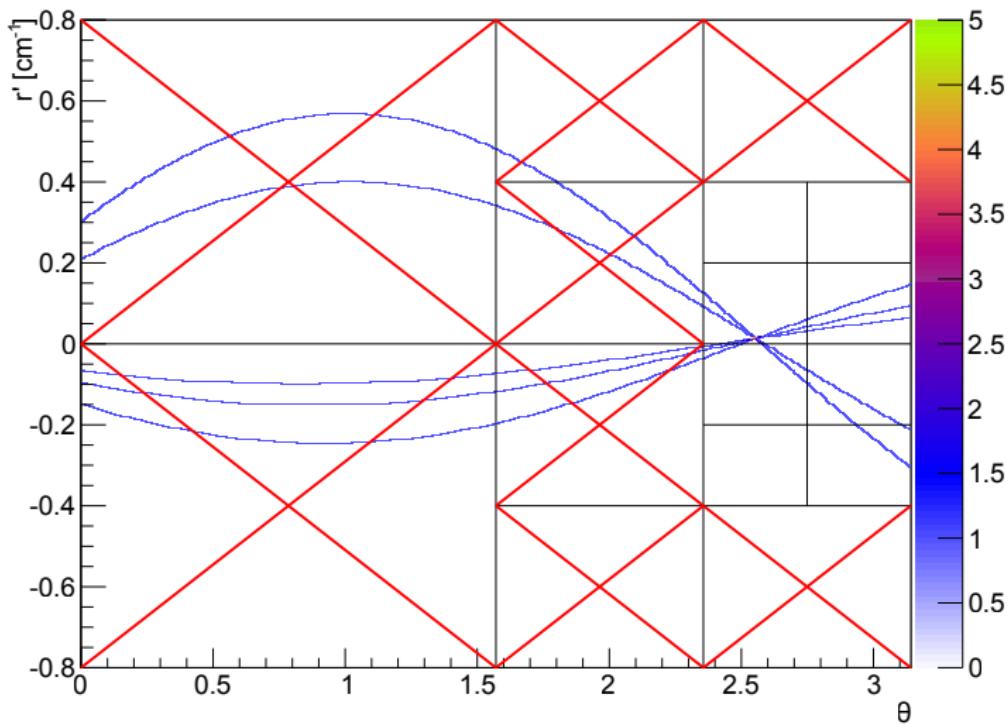
# Fast Hough Transformation

Hough Space



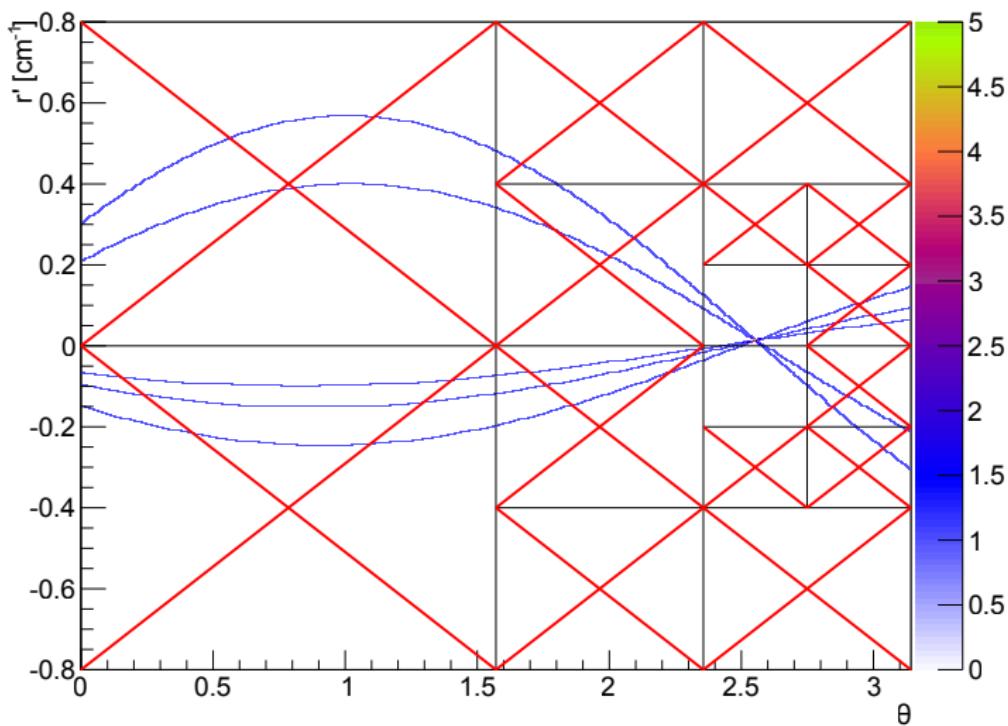
# Fast Hough Transformation

Hough Space



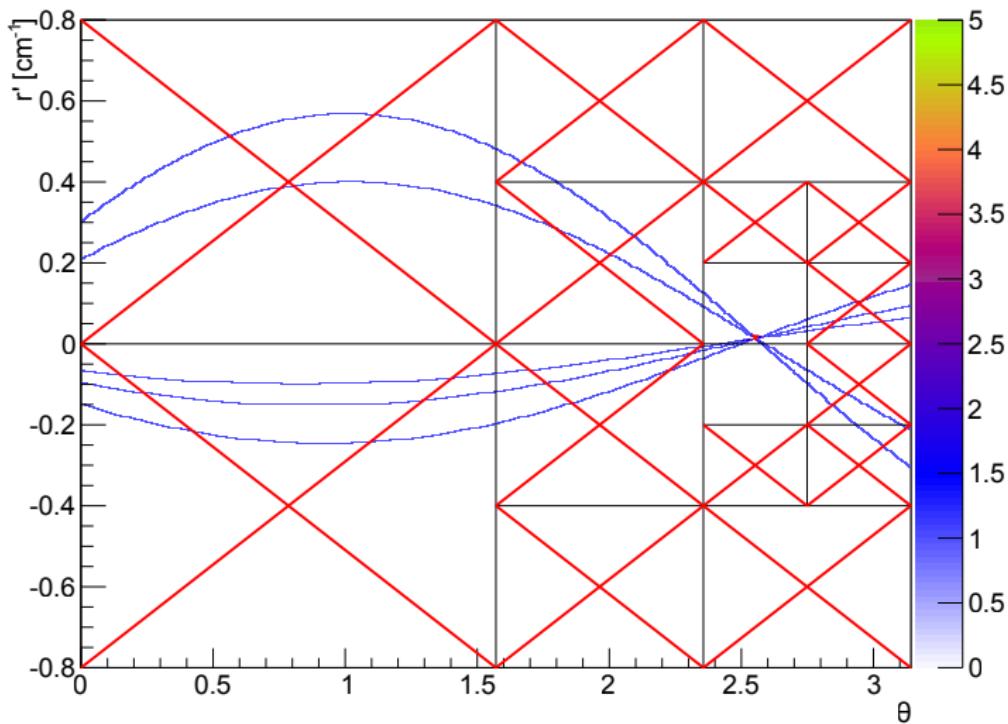
# Fast Hough Transformation

Hough Space



# Fast Hough Transformation

Hough Space



# Reduction of combinatorics

- ➊ High PXD occupancy → huge combinatorics ( $\sim 10^{11}$  hit-hit combinations)
- ➋ Secondary vertices → choose hits in 3<sup>rd</sup> VXD layer (not yet assigned to a DATCON track) as reference point for conformal transformation
- ➌ Idea: Include only PXD hits, that 'fit' to the reference point

## MC simulation

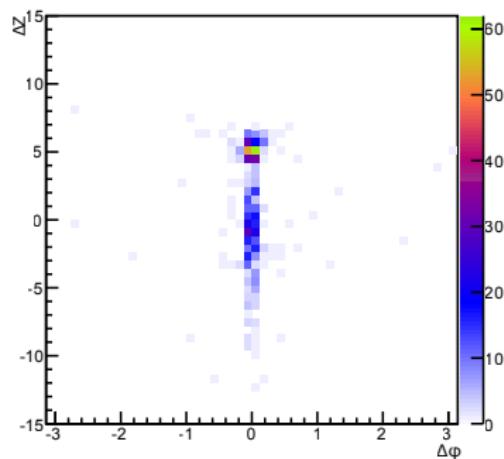
- 10 000 inclusive  $B\bar{B}$  events
- No background included yet
- Select  $\pi^\pm$ :
  - No hits in CDC
  - $\leq 2$  hits in SVD
  - $\geq 3$  hits in VXD
  - Seedcharge < dE/dx Cluster Rescue threshold

# Reduction of combinatorics

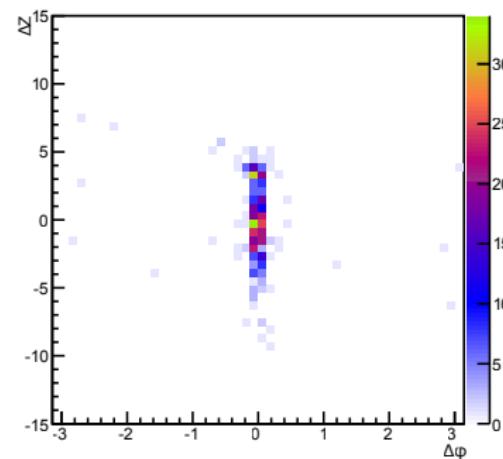
## Observables

- $\Delta z_{i,j} = z_{\text{hit, layer } i} - z_{\text{hit, layer } j}$
- $\Delta \phi_{i,j} = \phi_{\text{hit, layer } i} - \phi_{\text{hit, layer } j}$

Layer 1 - Layer 3



Layer 2 - Layer 3

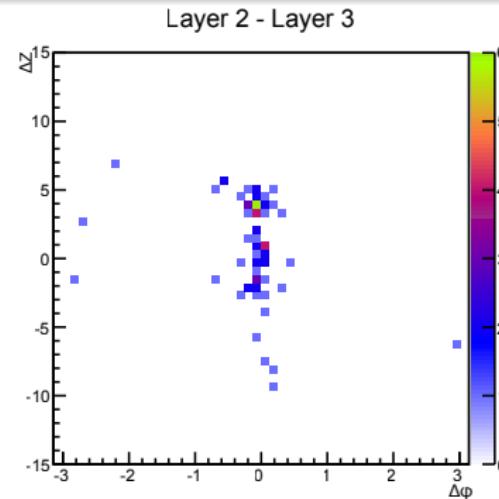
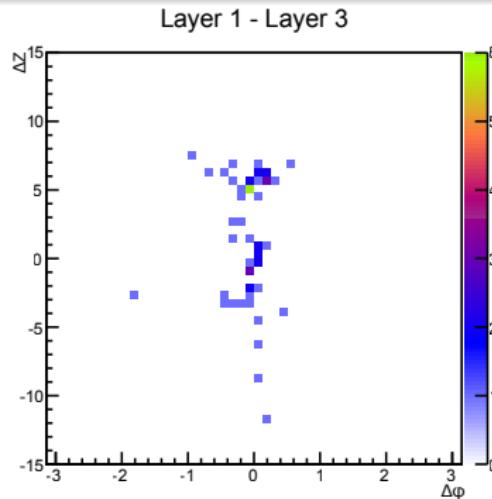


All  $\pi^\pm$

# Reduction of combinatorics

## Observables

- $\Delta z_{i,j} = z_{\text{hit, layer } i} - z_{\text{hit, layer } j}$
- $\Delta \phi_{i,j} = \phi_{\text{hit, layer } i} - \phi_{\text{hit, layer } j}$



Only  $\pi^\pm$  from  $K_S^0$

# Reduction of combinatorics

## More observables of hits

- Coordinates
- Clustersize
- Clustershape
- Clustercharge
- Seedcharge

## Task:

Construct a discriminator to find PXD hits associated to a track through reference point

# Summary

- PXD data reduction via ROIs
- Tracks with too few hits in SVD + CDC → no ROI

Novel approach:  $K_S^0$  Rescue will

- save hits from tracks with displaced vertices (DATCON and HLT have constraints to primary vertex)
- save hits from tracks with  $p > 100 \text{ MeV}/c$  ( $dE/dx$  Cluster Rescue will lose them)

To do:

- Full background
- Efficiency studies and optimization
- Implementation on hardware

# Thank You!