Tracking Sub-atomic Particles Through the Attribute Space

Mohammad Babai[1], Nasser Kalantar-Nayestanaki[1],
Johan G. Messchendorp[1], and Michael H.F. Wilkinson[2(✉)]

[1] KVI-Center for Advanced Radiation Technology, Groningen, The Netherlands
M.Babai@rug.nl

[2] Johann Bernoulli Institute for Mathematics and Computing Science,
University of Groningen, The Netherlands
m.h.f.wilkinson@computer.org

Abstract. In this paper, we present the results of an application of attribute space morphological filters for tracking sub-atomic particles in magnetic fields. For this purpose, we have applied the concept of attribute space and connectivity to the binary images produced by charged particles passing through the tracking detector for the future experiment PANDA. This detector could be considered as an undirected graph with irregular neighbourhood relations. For this project, we rely only on the detector geometry. In addition, we have extended the graph to estimate the z-coordinates of the particle paths. The result is an $O(n^2)$, proof of concept algorithm with a total error of approximately 0.17. The results look promising; however, more work needs to be done to make this algorithm applicable for the real-life case.

Keywords: Attribute space connectivity · Orientation based segmentation · Irregular graph · Sub-atomic particle tracking · Graph morphology

1 Introduction

Many image-analysis problems involve the recognition of thin, line-shaped, oriented structures. When some elongated thin, bright structure needs to be segmented, one possible approach is to remove parts of the image that are neither elongated nor thin. A standard approach is to use of a supremum of openings, where line-shaped structuring elements in different directions could be used [1]. A direct application of this idea will lead to an inefficient implementation of such an operator. One solution to this was proposed by Talbot et al. in [2]. They propose an ordered algorithm for implementing path operators for both complete and incomplete paths with logarithmic complexity as a function of the length of the flexible structuring element used. Most of the proposed solutions to solve the path recognition problems cannot deal with overlapping structures and are not directly capable of grouping pixels into disjoint paths. To solve the problem of overlapping structures, Wilkinson proposed in [3] a new concept of connectivity in higher dimensional spaces, referred to as *attribute spaces*. It is

J.A. Benediktsson et al. (Eds.): ISMM 2015, LNCS 9082, pp. 86–97, 2015.
DOI: 10.1007/978-3-319-18720-4_8

shown that a transformation of an image into a higher dimensional space, for example in an orientation based attribute space, using new computed attributes could be a solution to segment overlapping structures in disjoint segments.

A systematic theory for the construction of morphological operators on graphs was introduced by Toet et al. [4] where structural information could be extracted from graphs using pre-defined probes (structuring graphs). Dilations and erosions were constructed using the graphs neighborhood function and in combination with the probes one could define openings and closings operators on graphs [5].

Here, we present the results of an application of attribute space connected filtering, inspired by graph morphology for recognizing charged particle tracks through the Straw Tube Tracker (STT) for the future experiment PANDA (anti-Proton ANnihilations at DArmstadt). PANDA is one of the experiments at the future Facility for Antiproton and Ion Research (FAIR), which is currently under construction in Darmstadt, Germany. One of the objectives of this experiment is to study the structure of hadronic matter via the annihilation of antiprotons with protons at interaction rates up to 20 MHz. During these collisions, various particles with a large momentum range are produced. The trajectories of charged particles are reconstructed using tracking detectors placed inside a solenoid magnetic field. The curvatures of the reconstructed tracks are used to obtain the momenta of the corresponding particles. The PANDA tracking system consists of central and forward trackers. Here, we consider only one of the central tracker detectors, namely the STT. To cope with the high event rate, the data are processed at runtime and reduced by a factor of about 10^3. To achieve such a reduction factor, intelligent and fast algorithms need to be used that are capable of reconstructing the particle tracks in-situ. There are a number of conventional methods with high precision being used for the recognition of the particle paths through the space [6], but at present none of them is suited for online applications because of their computational complexity. These methods rely on transforming the image points into other spaces (Riemann space, Hough transformation), recognizing the structures, back transformation and fitting curves through the collected space-points where one needs to rely on drift times and corresponding calibrations. In our approach, we use only the geometrical information of the detector. One of the major advantages of our method is that it does not depend on the drift time of the charge through the tubes. Moreover, our algorithm is conceptually easy to implement on embedded architectures such as FPGAs and GPUs with $O(n^2)$ complexity in the number of active pixels.

In this paper, in sec. 2 the geometry of the STT subsystem is discussed, sec. 3 will give a brief summary of the basic concepts of filtering and connectivity, in sec. 3.1 the attribute space and attribute space filters are briefly described. In sec. 4 we present our method and finally we discuss the results and future works in sec. 5.

2 Geometry of the STT

The STT subsystem is the main tracking detector for charged particles and contains 4542 single straw tubes, arranged in a cylindrical volume around the beam-target interaction point. The straws are mounted in the system in two different ways. The axial straws which are parallel to the z-axis (the beam line) and skewed straws which are skewed by a few degrees ($+2.9°$ or $-2.9°$) to the axial direction and are meant to measure the z information of the track. All straws have the same inner diameter of 10 mm and a length of 1500 mm, except for a few outer straws in the border region of each skewed layer, which have a reduced length. The read-out system is designed such that all tubes are read from one side. Figure 1 left shows a schematic view of the STT in the $xy-$plane and the right panel the 3D layout of the STT subsystem [6].

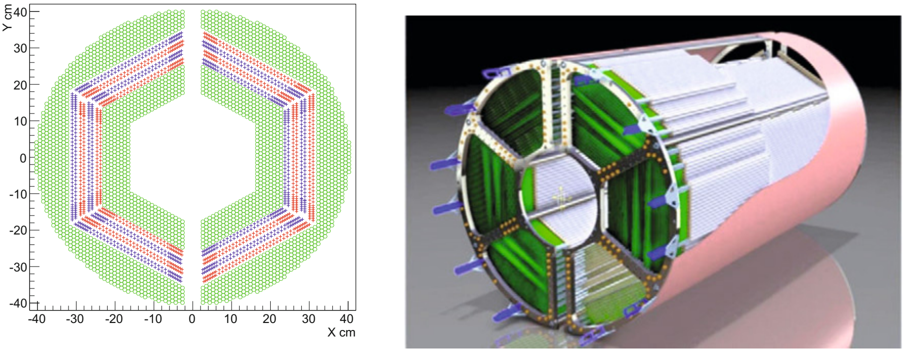

Fig. 1. Left: the front view (along beam direction) of the STT in the xy-plane. The skewed straws are shown in red and blue and green dots represent the axial straws. Right: a 3D view of the STT subsystem.

4A simulation, digitization and reconstruction software package has been developed for this detector. The digitization software provides a realistic model of the hardware readout of the foreseen detector. Here, we use only the data from simulation and digitization parts of this software package. For further details, we refer to the STT technical design report [6].

2.1 STT Graph

The geometry of the STT detector contains the three dimensional coordinates of the center-point of each tube, its half length, the list of its direct neighbors and a direction-vector which describes the slope of the read-out wire; the read-out wire has the same length as the tube itself and passes through the center of the tube. Because of the skewed tubes, the number of neighbors is not equal for all nodes in the graph; it varies between two and 22 neighbours. This way, the detector could be considered as a graph of pixels with varying neighborhood relations. The nodes are described by the tubes and the edges by the neighbouring relations

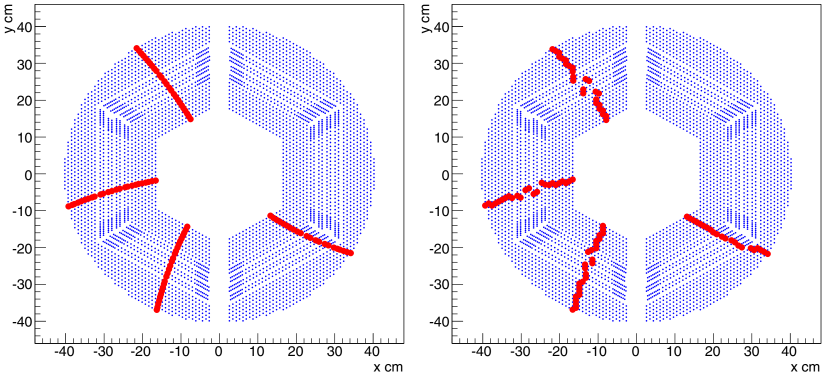

Fig. 2. The position of the STT tubes is shown in blue. Left: the true hit positions (red points) of the particles. Right: red points show the position of the reconstructed hits.

between the nodes. When a charged particle travels through space in a magnetic field, it hits a number of tubes which are turned on while the remaining nodes remain off. This pattern could be considered as a binary image of a number of thin, curved, path-shaped patterns. The curvature, path lengths and multiplicity of the paths depend on the particle type, its momentum and the strength of the magnetic field. An example of such paths is shown in fig. 2. One can see that the axial (z-parallel) straws produce an accurate position in the xy-plane and the skewed nodes have a maximum displacement of ±3 cm based on their lengths and direction. The latter is due to the fact that the exact hit positions, along the tubes, are not defined. Further, as a consequence of the readout and detector efficiency, the generated paths might have gaps of one or more missing pixels (tubes). These effects are visible in the right panel of fig. 2.

To have a better gradual transition between the layers with a different slope, we have extended the STT-grid by 20398 virtual nodes. These nodes are positioned between the tubes with different slopes at the center of the "virtual" intersection volumes. The virtual nodes have exactly two neighbours and have well defined xyz-coordinates. They are turned on if and only if both parent tubes are hit. Adding these nodes reduces the size of the hit displacements in

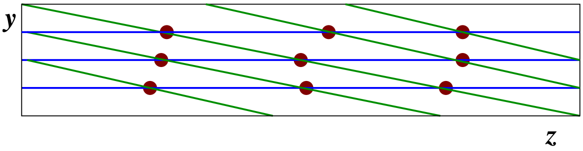

Fig. 3. A schematic view of a number of axial and skewed tubes. Axials are shown in blue, skewed in green. Virtual nodes are represented by red dots. The y and z coordinates of the virtual nodes are determined from the intersection points of the skewed tubes projected onto the plane of the axial tubes. The x coordinate of the virtual nodes is midway between the two planes.

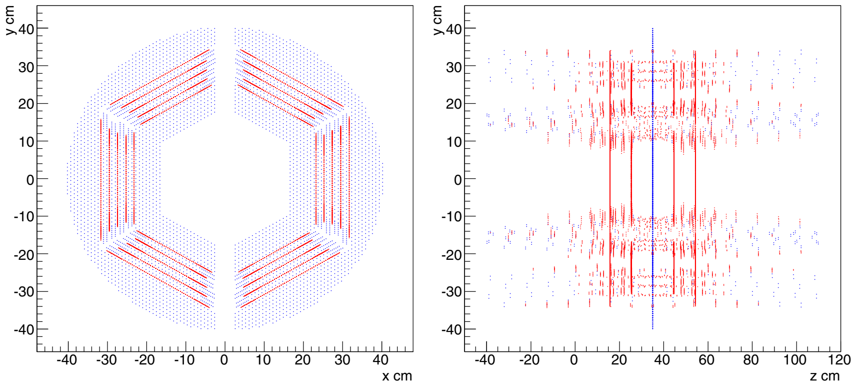

Fig. 4. The structure of the STT graph extended by the virtual nodes. Virtual nodes are shown in red and the center points of the original nodes in blue; the blue dots in the left panel are also the only available information in the z-direction. Left: STT graph in the $xy-$plain. Right: STT graph in the $yz-$plain.

the xy-plane and provides a better estimate of the z-coordinates of the nodes participating in a path. Figure 3 illustrates the positioning of virtual nodes and fig. 4 shows the structure of the grid after the extension.

3 Basic Concepts Filtering and Connections

Consider a universal set E (usually $E = \mathbb{Z}^n$), a binary image X is defined as a subset of E. Define $\mathcal{P}(E)$ to be the power set of E. Connectivity can be defined using the connectivity classes [3,7,8]. Each connectivity class $\mathcal{C} \subseteq \mathcal{P}(E)$ is a set of sets for which holds that the empty set and singletons are connected and for all connected sets with non-empty intersection their union is connected. Using this definition, each image X is a set of connected components C_i with i from some index set I with $\{\forall C_i, \nexists (\mathrm{C} \supset C_i) : \ \mathrm{C} \subseteq X \wedge \mathrm{C} \in \mathcal{C}\}$. A connected component of X is denoted by $\lessdot$. A binary connected opening operator $\varGamma_x$ on X at the point x could be defined as:

$$\varGamma_x(X) = \begin{cases} C_i : x \in C_i \wedge (C_i \lessdot X) & \text{if } x \in X \\ \emptyset & \text{else.} \end{cases} \quad (1)$$

The group of attribute filters is one of the classes of the connected filters. We can define an attribute filter as:

$$\varGamma^T(X) = \bigcup_{x \in X} \varGamma_T(\varGamma_x(X)). \quad (2)$$

Here, T is a criterion with the property of being increasing. T is usually of the form: $T(C) = Attr(\mathcal{C}) \geq \lambda$. $Attr(\mathcal{C})$ is some attribute of $\mathcal{C}$ and λ is a threshold value [9].

3.1 Attribute Space and Attribute Space Filters

Attribute space connectivity and filters were introduced by Wilkinson as a method to improve attribute filtering when connectivities fail to perceptually group correctly [3]. Here, we summarize only the main concept. In this concept, an image $\mathrm{X} \subset E$ is transformed into a higher dimensional space "attribute space" $E \times A$. Here, A is an encoding of the attributes of pixels in X (for example, the local width). The operator $\Omega : \mathcal{P}(E) \to \mathcal{P}(E \times A)$ is defined such that it maps a binary image X into $\mathcal{P}(E \times A)$ space. The increasing inverse operator Ω^{-1} projects $\Omega(X)$ back to X ($\forall \mathrm{X} \in \mathcal{P}(E) : \ \Omega^{-1}(\Omega(\mathrm{X})) = \mathrm{X}$). Using this extension, the attribute space connectivity class $\mathcal{A}$ on a universal set E by the transformation pair (Ω, Ω^{-1}) and connectivity class $\mathcal{C}_{E \times A}$ on the transformed image is defined as:

$$\mathcal{A} = \{\mathcal{C} \in \mathcal{P}(E) \mid \Omega(\mathcal{C}) \in \mathcal{C}_{E \times A}\}. \tag{3}$$

Using this and the transformation pair, a connected filter $\Psi^A : \mathcal{P}(E) \to \mathcal{P}(E)$ could be defined as:

$$\Psi^A(\mathrm{X}) = \Omega^{-1}(\Psi(\Omega(\mathrm{X}))), \tag{4}$$

with $\mathrm{X} \in \mathcal{P}(E)$ and $\Psi : \mathcal{P}(E \times A) \to \mathcal{P}(E \times A)$. Using this framework, different operator-pairs based on, for example width, orientation, even a combination of different attributes can be defined to transform X into the attribute space where connected filters could be used for segmentation.

4 Orientation-Based Attribute Space

Using the framework described in sec. 3.1, one can define the operator $\Omega_\alpha : \mathcal{P}(E) \to \mathcal{P}(E \times A)$ that assigns one or more orientations α_i to every active node x in the graph [3]. This yields a function $f(x, \alpha)$ over $E \times A$. Next, we apply the operator Ω_α on the input image X as:

$$\Omega_\alpha(X) = \{x \in \mathrm{X}, \alpha \in A \mid f(x,\alpha) > \lambda f_{min} \vee f(x,\alpha) = f_{max}\}. \tag{5}$$

Here, f_{min} is minimum value for $f(x, \alpha)$, f_{max} the maximum value and $\lambda \geq 1$ is a tuneable parameter that determines how strict the selectivity is in assigning a node to an orientation space. Using this concept, the orientation-based attribute space is computed for all active nodes in each event in two different ways, namely dynamically and statically. For the static method, a number of angles are predefined to determine the orientation-based attribute space where in the dynamic method the angles are determined for each instance (binary image) in a pre-sensing step. We have observed that using both methods lead to similar results with negligible differences. This could be explained by the fact that all tubes have a diameter of 1 cm and their relative positions are permanent in the space. Due to this arrangement of the nodes, the number of possible orientations is limited and does not have a broad variation.

To determine the attribute space for the STT-graph, we have applied a slightly modified version of the method proposed in [3] by keeping a record of the participating active nodes for all orientations. In this case, we have used the following method for all active nodes in the STT-graph:

1. Compute an opening transform Ω_X^α using a linear structuring element with orientation α and keep a record of all participating active nodes. The structuring element is chosen such, that its length is maximal for the given orientation α.
2. This yields the gray-level function $f(x, \alpha)$ over the domain $(E \times A)$.
3. For all active nodes in X determine f_{min} and f_{max}.
4. Compute $\Omega_\alpha(X)$ using eq. 5.

4.1 Analysis of Attribute Space Connected Component

Transforming the binary image X (one STT readout instance) to the orientation-based Attribute Space using $\Omega_\alpha : \mathcal{P}(E) \to \mathcal{P}(E \times A)$ produces a binary image with an additional dimension, namely the orientation-based dimension. One can redefine the neighborhood relation between the points in the graph using the computed orientation. This way, we can step through the orientation spaces and collect the participating points in each subspace to construct the connected components. In this case the following method is used:

1. Sort all active nodes based on their layer, decreasing order. Insert all nodes in a priority queue (FIFO).
2. For all nodes in the queue: if it is visited for the first time and f_{max} >= minimum required response, create a component candidate and add all members of the orientation sub-space corresponding with f_{max} to the current candidate. Otherwise the node is added to the list of short response nodes.
3. If the node was visited: Find the candidate to which it was added, then step through the neighboring orientation sub-spaces and add the nodes if Ω_α.

The result of the application of this method is shown in fig. 5. This procedure is shown in alg. 4.1. Note that the singletons are kept in a separate set of node list; this way, those grid points could be accessed if necessary.

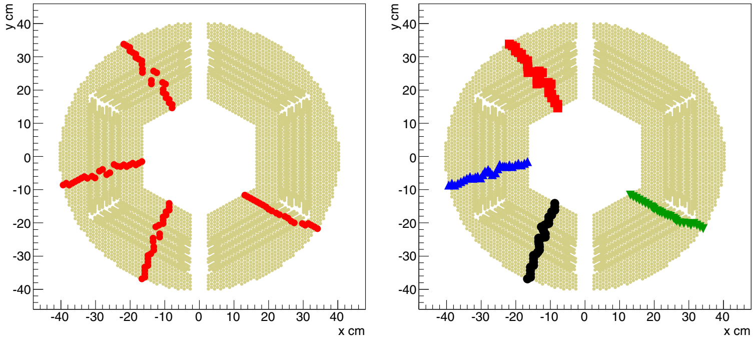

Fig. 5. Left: the digitization output. Right: the composed connected components using alg. 4.1. Each color represents a different path.

Algorithm 4.1 Attribute Space Connected Component

Input: The graph of the tubes, minimum required response.
Output: List of sets "$AllCompList$" containing all connected components.

```
Sort active nodes based on their layers.
Create two FIFOs, ActiveQueue and ShortCompQueue
Insert all active nodes in ActiveQueue
while (¬ empty(ActiveQueue)) do
    Fetch first node "$CurentNode$"
    if ($CurentNode_{f_{max}} < MinimumResponse$) then
        Add node to "ShortCompQueue" and set visited
    else
        if ($CurentNode$ was not visited) then
            Create a new empty set $S$ and add $CurentNode$ to it. Set visited
            for all nodes active in $CurentNode_{f_{max}}$ orientation do
                if (¬ visited) then
                    Add to $S$; mark as visited
                end if
            end for
        else
            Create a list of sets $CompL$ where it was added before
            for all ($cmp \in CompL$) do
                for all ($node \in cmp$) do
                    if (¬ visited ∧ active in current orientation) then
                        Add nodes from neighboring orientations to $cmp$
                    end if
                end for
            end for
        end if
    end if
end while
if (¬ empty(ShortCompQueue)) then
    Create an empty set $Orph$
    Fetch element $sn$
    Find $cmp \in AllCompList$ the node $nn$ with the shortest distance to $en$
    if ($nn$ and $en$ direct neighbors) then
        Add $en$ to $cmp$
    end if
else
    Add $en$ to the set $Orph$
end if
```

4.2 Determination of *z*-coordinate

One of the most important features of each track is the evolution of the path in the z direction. The full reconstruction of each path is used to determine the properties of the passing particle. The presence of the skewed tubes helps to estimate the $z-$coordinates of the points along the path. To have a more accurate estimate of the $z-$coordinates of the points, we introduced virtual nodes as described in sec. 2.1. Using the assumption that all the tracks passing through the STT originate from $(0, 0, 0)$, we could estimate the deviations in the z-direction and generated a rough reconstruction of the whole particle path. Starting from the origin to the first virtual layer, we apply linear interpolation by using a constant step size of $\delta z = z_1$, with z_1 being the z-coordinate of the first virtual node in the track. Between the first and the last virtual layer, an adaptive δz is computed, when passing the layers. From the most outer virtual layer to the most outer node we extrapolate linearly and δz remains constant and equal to the latest determined value. Linear interpolation and extrapolation is used

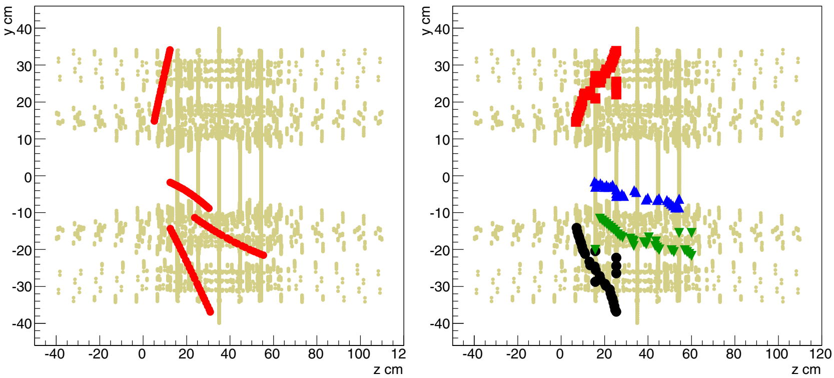

Fig. 6. Left: the true hit positions. Right: the determined paths along the z-axis using linear interpolation, assuming that the paths start at the origin. Because we are not correcting the coordinates of the skewed nodes in the xy-plane, some of the nodes seem to be wrongly positioned.

because there are no forces acting in the z-direction. Figure 6 shows an example of the resulting z reconstruction based on the given description.

5 Results and Discussion

In this section, we present the results of the application of the previously described methods. To obtain the sample test data sets, we have used the PANDA simulation package to produce 10^3 μ^- events with different momenta and a multiplicity of 6 tracks per event[1]. The tracks are generated isotropically around the origin $(0, 0, 0)$. To quantify the segmentation error, we have applied the following method: given an image X with a total area of A, let a reference segmentation divide the image into N regions $\{R_1, \ldots, R_N\}$ with corresponding areas $\{A_1, \ldots, A_N\}$. An example segmentation finds M regions $\{T_1, \ldots, T_M\}$ with areas $\{a_1, \ldots, a_M\}$. For all regions T_j, we find R_k such that $(T_j \bigcap R_k)$ is maximal. One can define the "undermerging" (E_{um}), "overmerging" (E_{om}) and normalized total error (E_{tot}) as:

$$E_{um} = \frac{1}{A} \sum_{j=1}^{M} \frac{\{A_k - \#(T_j \bigcap R_k)\}\#(T_j \bigcap R_k)}{A_k}, \tag{6}$$

$$E_{om} = \frac{1}{A} \sum_{j=1}^{M} \{(a_j - \#(T_j \bigcap R_k)\}, \tag{7}$$

$$E_{tot} = \sqrt{E_{um}^2 + E_{om}^2} \tag{8}$$

[1] The software package "pandaroot" is available from: subversion.gsi.de (Revision: 26379). To make this analysis reproducible, we have fixed the starting seed.

in which $\#C$, denotes the cardinality of C. Here, we have slightly modified the measures described in [10]. The total magnitude of error is obviously equal to zero if the two error types are equal to zero or vanishingly small compared to A. Total error does not give much information on the type of the error. Here, the Monte Carlo (MC) truth particle paths are considered as the reference segmentations to determine the values for both error types. As expected the momentum of the passing particle will affect its behavior during travel-time inside the detector due to a stronger interaction with the magnetic field. Particles with lower momenta have a more curved path while the ones with a higher momentum will be straighter. As a consequence of this momentum dependency, we expect that the value of different error measures will be larger for particles with a lower momentum; because high curvature tends to break up paths.

Figure 7 shows the distribution of error values for 10^3 events and table 1 summarises the mean values for a number of arbitrary selected momenta.

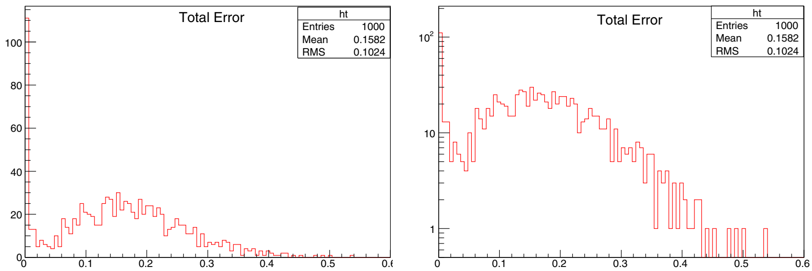

Fig. 7. Distribution of the total error for 10^3, 0.8 GeV events: (left) linear scale; (right) log scale

The under-merging error is mainly caused by the missing nodes or when the paths cross the regions between different sectors where no neighborhood information is available. The over-merging is caused when two or more paths are running through neighboring nodes specially when passing the skewed layers. Because of the fact that there is no direct and accurate information available on the $z-$coordinates when the paths are selected, passing the skewed layers by at least two close neighboring paths will lead to over-merging. Another observation

Table 1. Error estimates for 10^3, μ events. One can observe that E_{tot} shows a small fluctuation and E_{um} drops steadily as a function of momentum

Momentum	E_{um}	E_{om}	E_{tot}
0.4	0.09	0.13	0.18
0.8	0.08	0.12	0.16
1.0	0.08	0.13	0.17
1.5	0.07	0.13	0.16
2.0	0.07	0.12	0.16

is that the value of the total error shows a small fluctuation when moving from low to higher energy particle paths whereas E_{um} drops steadily as momentum increases. The highest values of error are observed for paths of slower particles. These results are preliminary and more statistical tests are needed for a more in depth investigation. Furthermore, we accept a gap-size of zero pixels. Although accepting a larger gap will solve a part of the under-merging problem, it will introduce a more frequent over-merging. Figure 8 shows an event where over- and under-merging effects are visible.

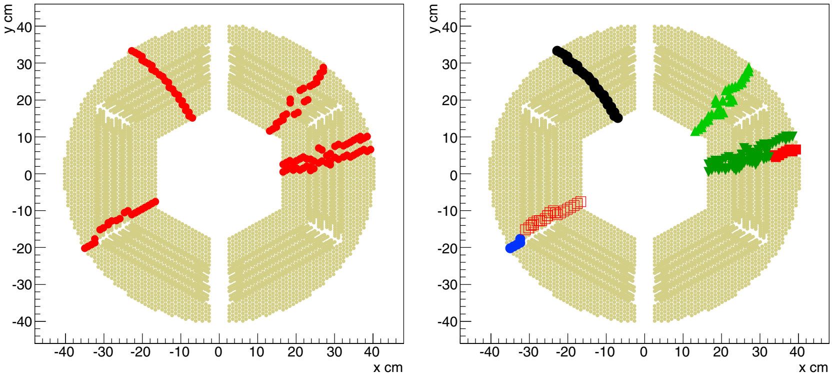

Fig. 8. Left: the detector read-out. Right: the resulting reconstructed paths in $xy-$plain. Each color represents a separate component.

Considering the structure of the graph and the direct available read-out data based on the detector geometry, the results of the applied method look promising; both in recognition of the tracks in 2D and determining the $z-$coordinates for the track points. The separation of the crossing paths is still one of the challenges that need to be addressed which will lead into a smaller over-merging error and a better overall performance. The attribute space framework can handle overlaps intrinsically. One can design attribute space filters to treat the image, based on its determined attributes even if there are overlapping structures in the original image domain. However, the presented results show that the direct application of the framework does not provide a complete solution for this problem. Note that it is necessary, in the case of overlap, that a node is a member of at least two separate paths.
One can estimate the computational complexity of Hough transformation by $P^D + P^{D-1}n$, with, P transformation space parameter, D the dimension of P and n is the number of active pixels. We can observe that this method is less sensitive to n. But when using a sensible number of parameters, the complexity will become $O(n^3)$; improvements could be achieved by parallelization. Our method in the current state has a $O(n^2)$ computational complexity and because of its structure we foresee a complexity reduction to $O(n)$ in the future. Because these methods differ intrinsically, we cannot compare them directly; but, we expect

that for a sound number of parameters and image size, our method will run with at least the same performance. Future work will consist of a modification of the current method by the inclusion of mechanisms for node splitting and separating crossing paths, a better determination of the z-coordinates of the points in each path and modifying the algorithm to reduce the computational complexity to $O(n)$. The recognition of the paths displaced from the origin is also one of the major challenges for our future work.

Acknowledgment. We thank the PANDA collaboration for their support and special thank to the STT developers group.

References

1. Kurdy, M.B., Jeulin, D.: Directional mathematical morphology operations. In: Proceedings of the Fifth European Congress for Stereology, vol. 8(2), pp. 473–480 (1989)
2. Appleton, B., Talbot, H.: Efficient complete and incomplete path openings and closings. Image and Vision Computing 25(4), 416–425 (2007)
3. Wilkinson, M.H.F.: Attribute-space connectivity and connected filters. Image and Vision Computing 25(4), 426–435 (2007); International Symposium on Mathematical Morphology 2005
4. Toet, A., Heijmans, H.J.A.M., Nacken, P., Vincent, L.: Graph morphology. Journal of Visual Communication and Image Representation 3, 24–38 (1992)
5. Cousty, J., Najman, L., Dias, F., Serra, J.: Morphological filtering on graphs. Computer Vision and Image Understanding 117(4), 370–385 (2013); Special issue on Discrete Geometry for Computer Imagery
6. (PANDA Collaboration) Erni, W., Keshelashvili, I., Krusche, B., et al.: Technical design report for the panda straw tube tracker. The European Physical Journal A 49, 25 (2013); Special Article Tools for Experiment and Theory
7. Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. J. Math. Imag. Vis. 8, 41–58 (1998)
8. Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vis. 9(3), 231–251 (1998)
9. Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Comp. Vis. Image Understand. 64(3), 377–389 (1996)
10. Wilkinson, M.H.F., Schut, F.: Digital image analysis of microbes: imaging, morphometry, fluorometry and motility techniques and applications, 1st edn. John Wiley & Sons Ltd., New York (1998)