Towards 3D Tracking of Arbitrary Particle Trajectories in LArTPCs: Part 1

Mitch Soderberg

Feb. 23, 2011

Introduction

- In this talk I will explain some ideas I'm beginning to develop for how one can use the tools already present in LArSoft to create the information necessary for "tracking" of arbitrary particle trajectories (i.e. - no restriction to simple straight-lines with constant slope and/or direction).
  - I am sure something similar to what I'll describe here has been done by previous experiments (ICARUS?)...I haven't looked to verify this, but I don't doubt it.
  - I am in the process of implementing the code for this, so I make no guarantees things will work out.
  - I will try to have preliminary results next time (Part II).

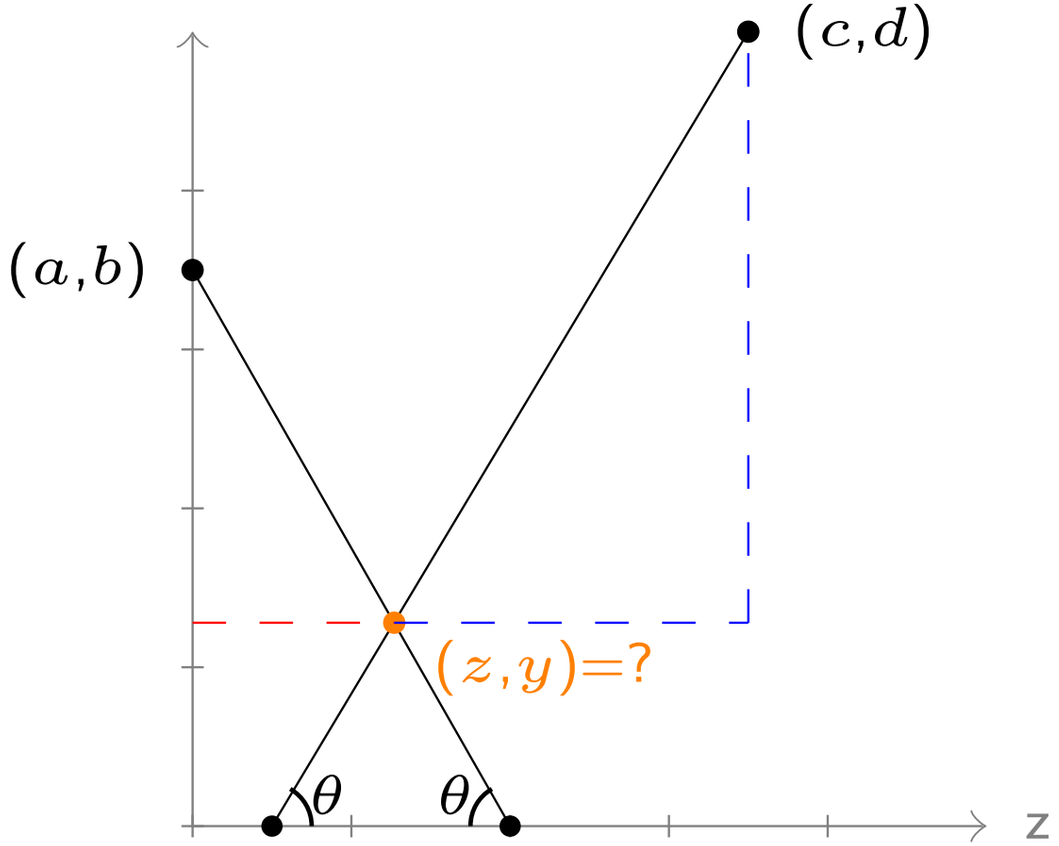
Plato's *Allegory of the Cave* :
"To them, I said, the truth would be literally nothing but the shadows of the images."

WirePlane Geometry I

- Start by understanding geometry of wires within the TPC.
- I've added functions to Geometry package to allow determination of whether two wires, from different planes, intersect. If they do intersect, the point of intersection in the (y,z) plane is returned.

```
void WireEndPoints(unsigned int plane, unsigned int wire, double *xyzStart, double *xyzEnd);

bool ChannelsIntersect(unsigned short c1, unsigned short c2, double &y, double &z);
```



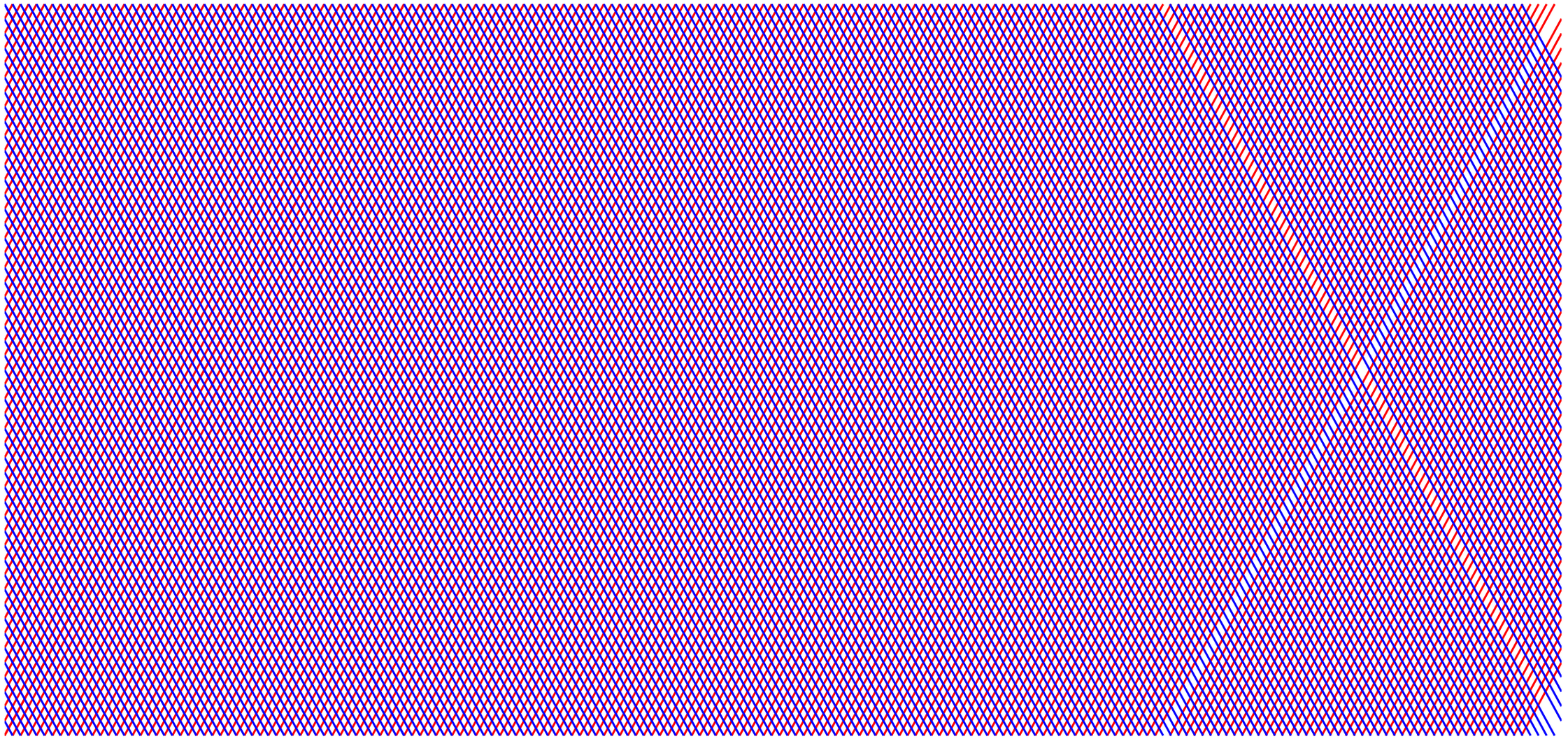
Simple Trig. leads to the intersection coordinate in terms of the upper endpoints of the two wires.

$$y = \frac{1}{2} \cdot \left(b + d + (a - c) \cdot \tan\,\theta \right)$$

$$z = \frac{1}{2} \cdot \left(c + a + \frac{b - d}{\tan\,\theta} \right)$$

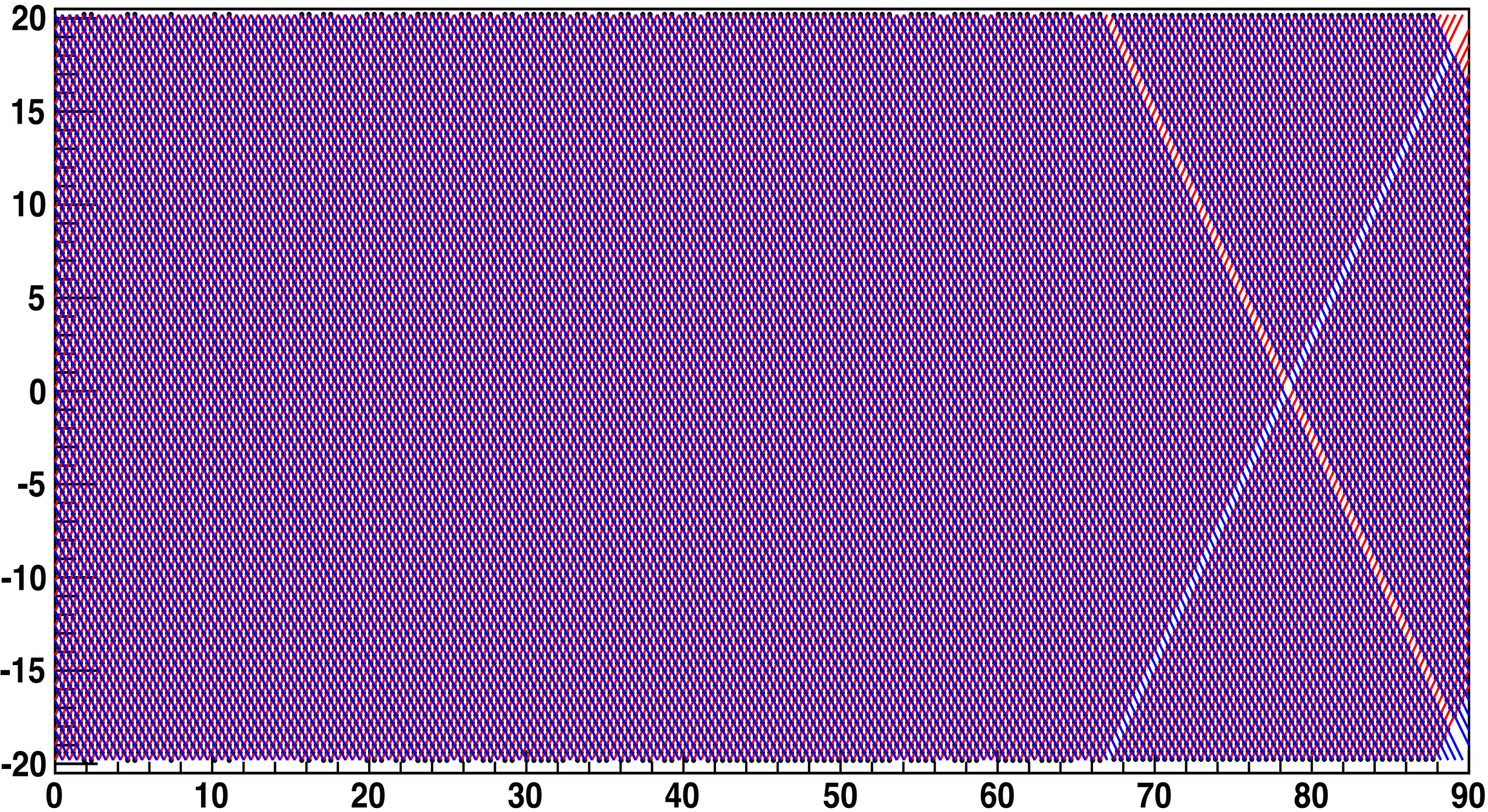
WirePlane Geometry II

- Use these tools to plot the wires in the ArgoNeuT Induction/Planes.
  - Observe the two "X" patterns in the picture...these are a consequence of an error in the GDML description, which overlaps two wires on the upstream end, which causes a gap in the downstream end. Should be simple to fix.
  - Next, notice that the upstream corners are completely filled in, while the downstream corners have empty regions where there are no more wires to fill in the space. In reality, there is an offset in where the first wires appear on the upstream end of the TPC, so really there should be small empty regions in all four corners. Should be simple to fix.



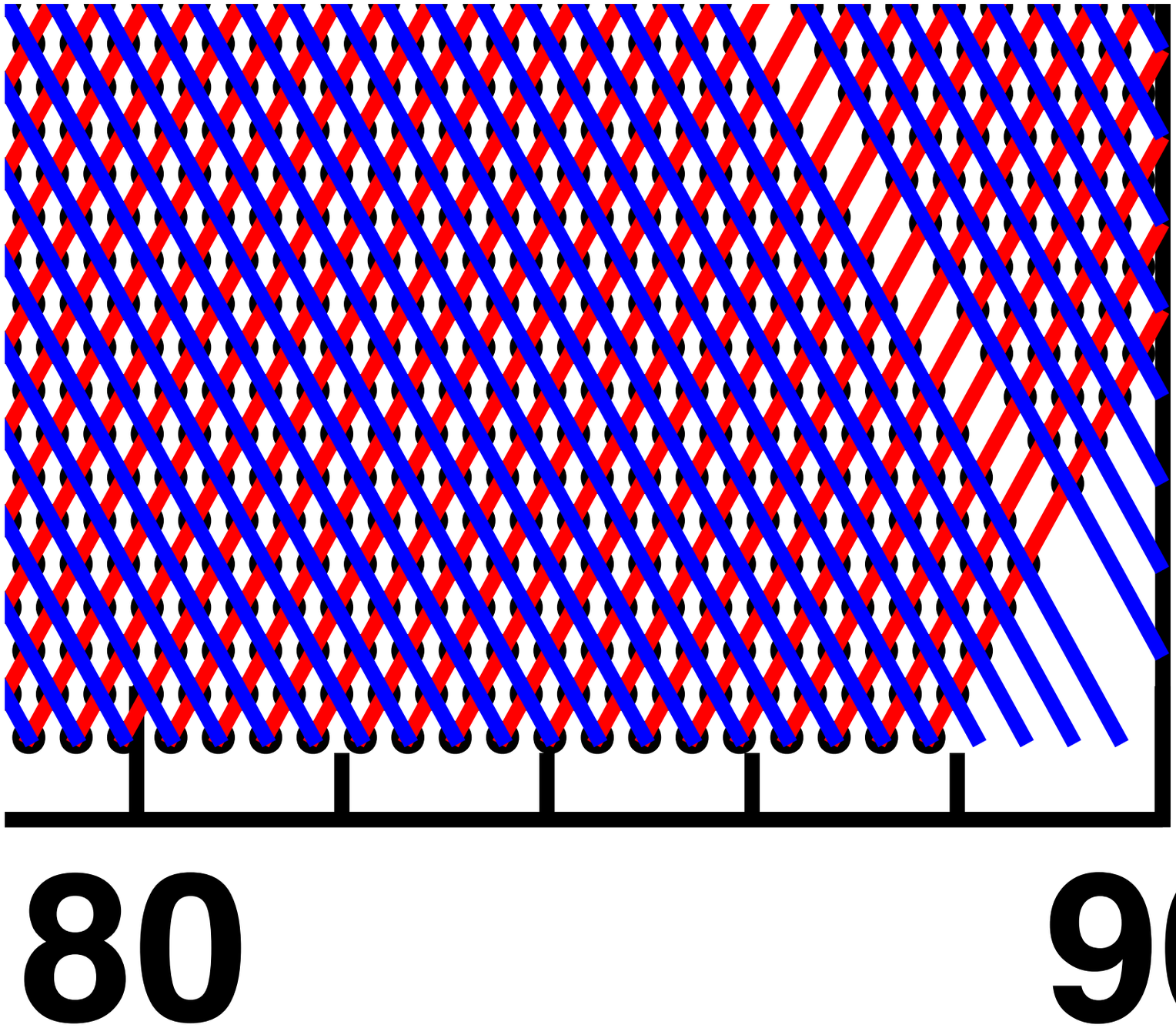
WirePlane Geometry III

- Apply "ChannelsIntersect" function to all wire pairs.
- Find 19508 points of intersection.
  - (a TPC with 240 induction/collection wires oriented at 90-degrees would have 240*240=57600)



WirePlane Geometry IV

- Zoom in on bottom-right corner of previous plot to show intersection points (black circles) returned by ChannelsIntersect function.
- **Moral of the story**: Intersection point for any wire pair can now be quickly determined.

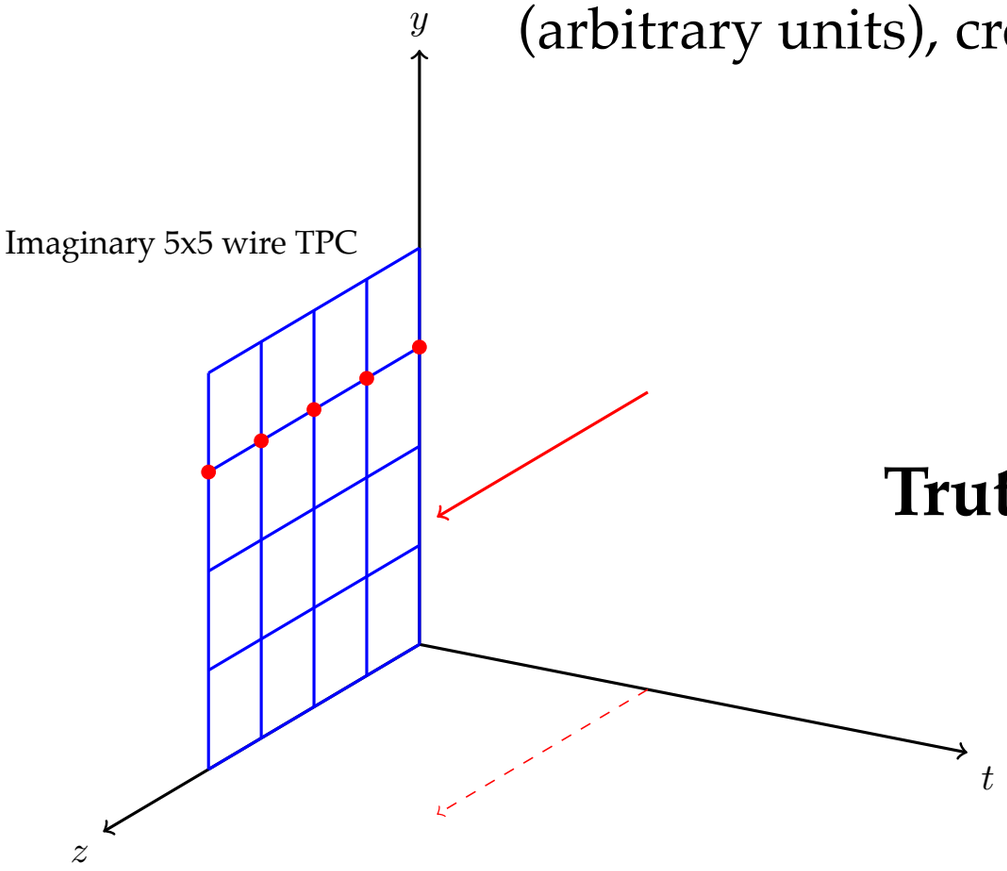


3D Tracking

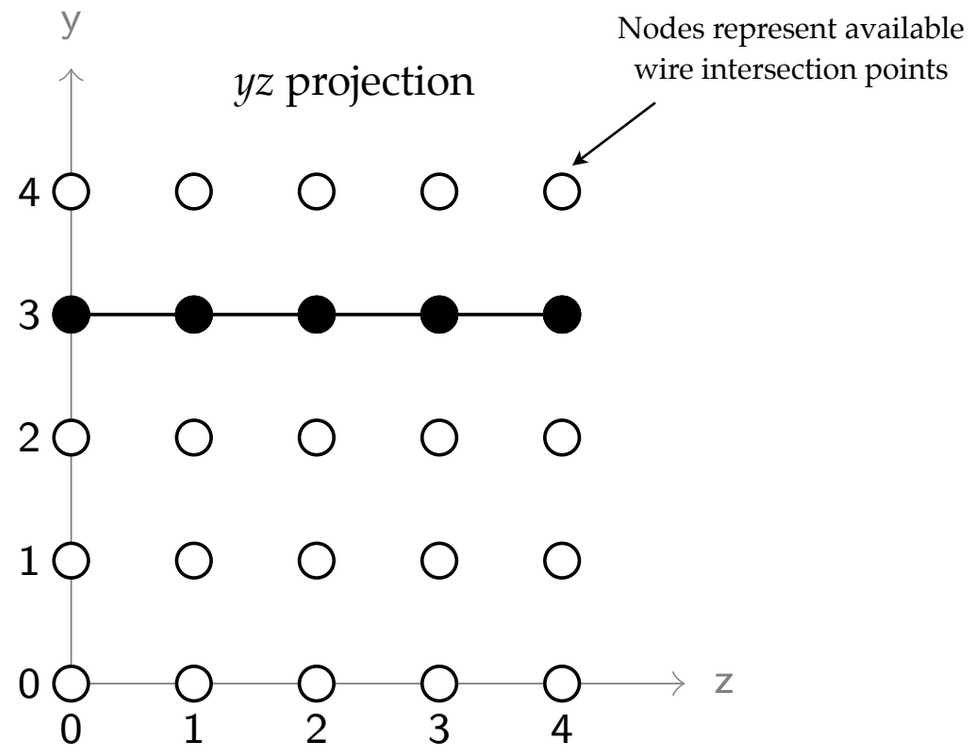
- Would like to have a robust tracking algorithm(s) that can determine the three-dimensional trajectory of all charged particles in an event.
- Thus far in our reconstruction efforts we assume all particles follow a straight-line (i.e. - constant slope/direction) trajectory.
  - This may be a good approximation for some scenarios, but it's certainly not true for very long tracks (due to multiple scattering), and it also doesn't take advantage of all the information available in the event.
- In the following slides I'll describe some simple examples of different particle trajectories, and discuss how we might reconstruct them in 3D.

Ex. 1: Track Parallel to YZ Plane

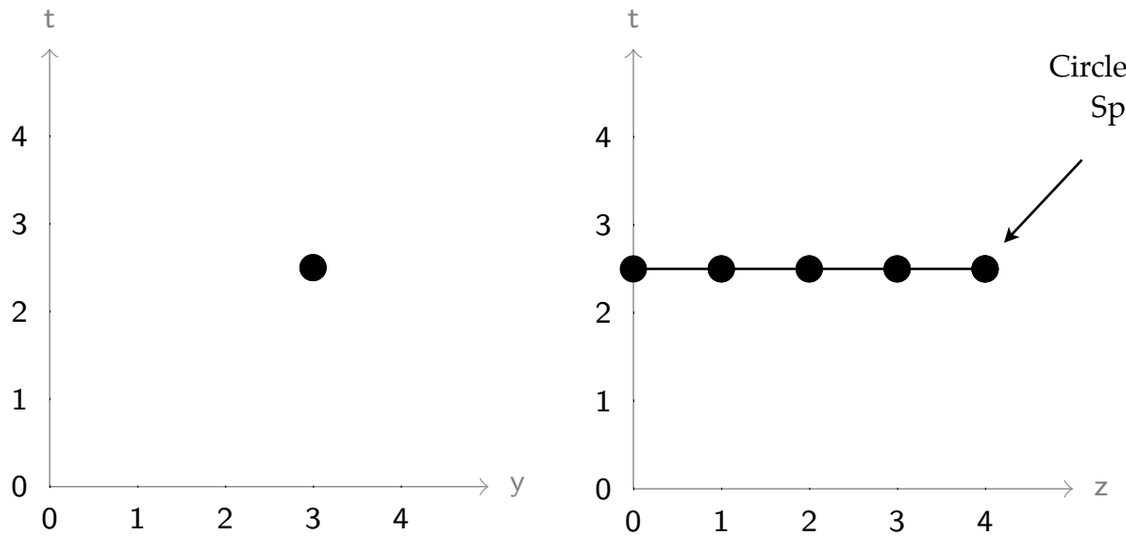
A single track, traveling along the z-direction, at time t=2.5 (arbitrary units), crosses all 5 z wires and only one y wire.



Truth Information

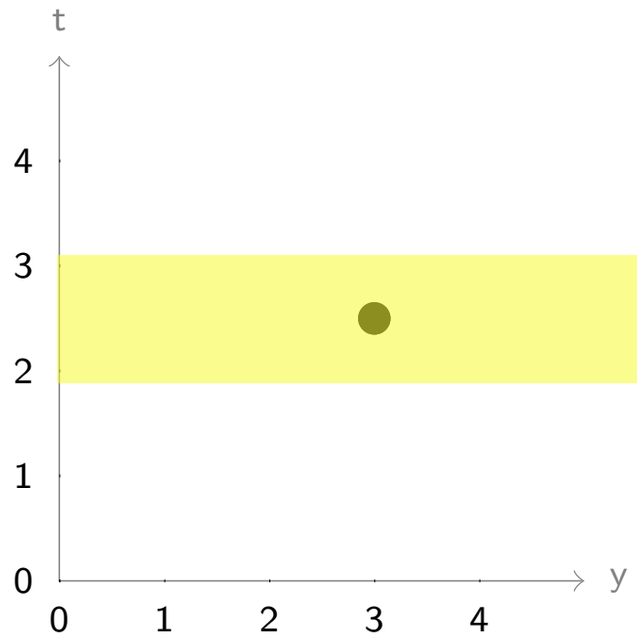


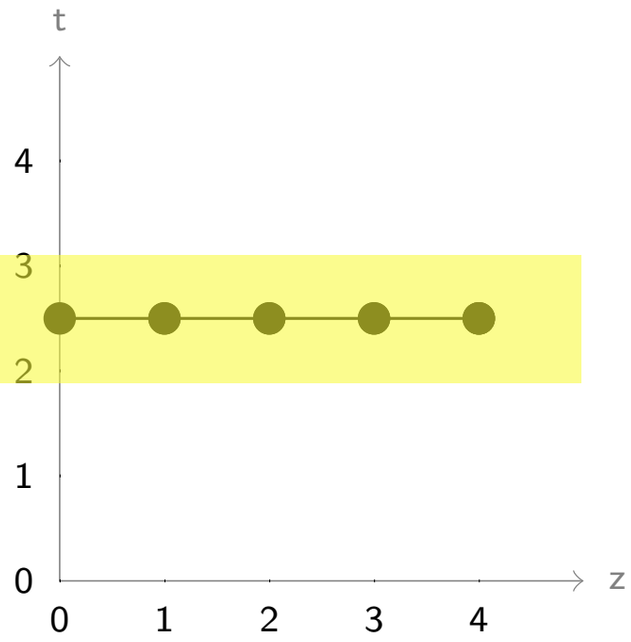
What we actually observe (I'm ignoring Hit characteristics for now).



Ex. 1: Track Parallel to YZ Plane

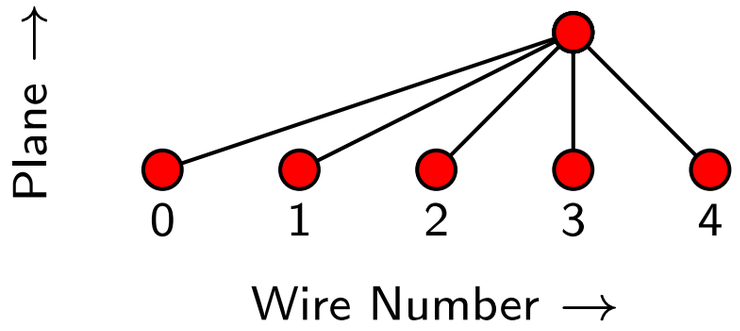
- To recover the true 3D trajectory we need to associate all Hits from Clusters in both planes that occur at the same time, and are on wires that cross. This is what we call a "SpacePoint".
- The set of all such SpacePoints defines allowable trajectories...in this example there is only one.



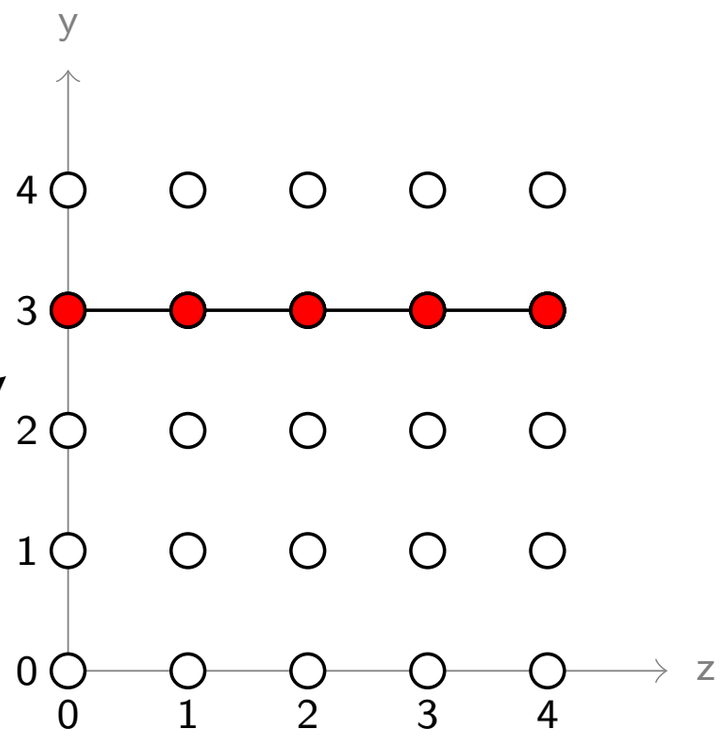


Hits from different planes that are within the same time slice create a SpacePoint.

Within each "SpacePoint" a network of associations between Hits can be created, which defines an allowable trajectory.

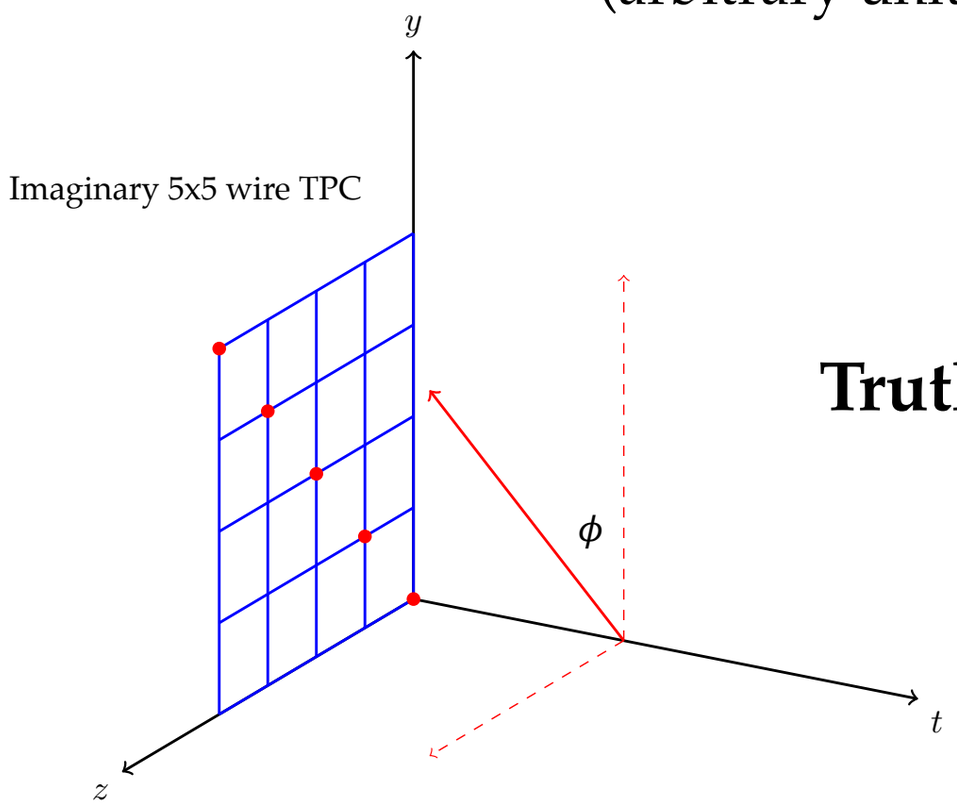


Corresponds to the following trajectory in the *yz* plane

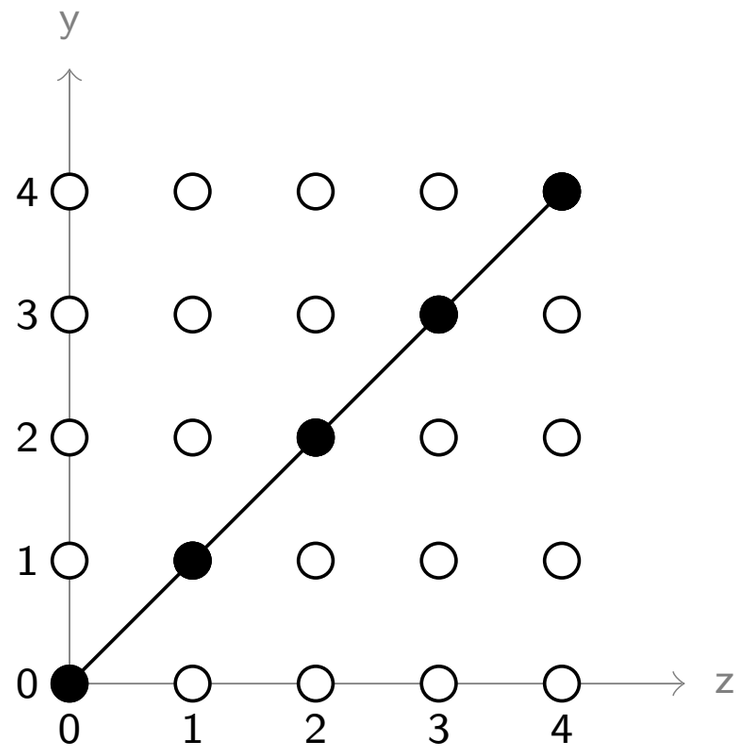


Ex. 2: Angled Track Parallel to YZ Plane

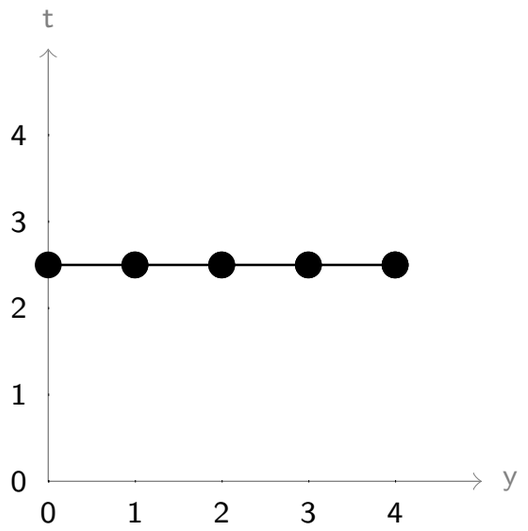
A single track, traveling along the z-direction, at time t=2.5 (arbitrary units), with arbitrary polar angle ϕ.

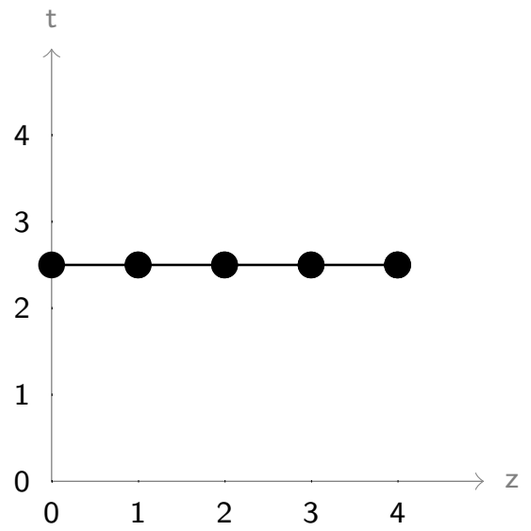


Truth Information



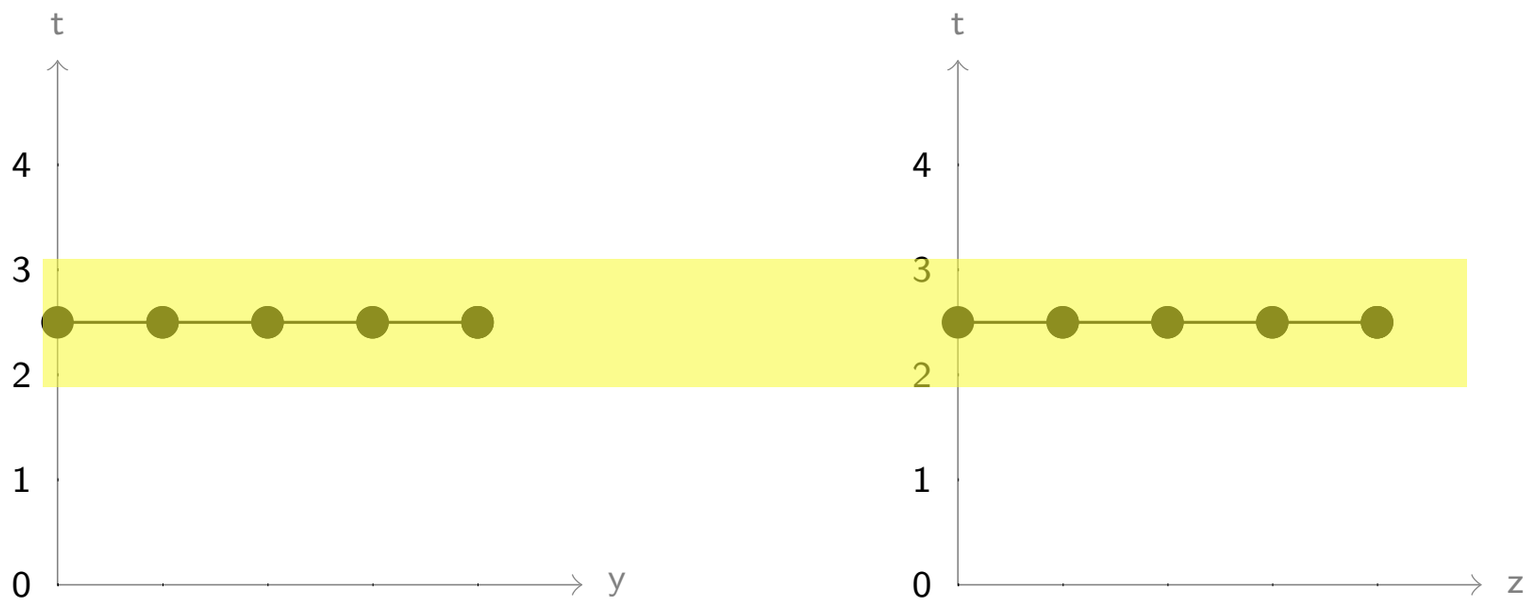
What we actually observe (again ignoring Hit characteristics).



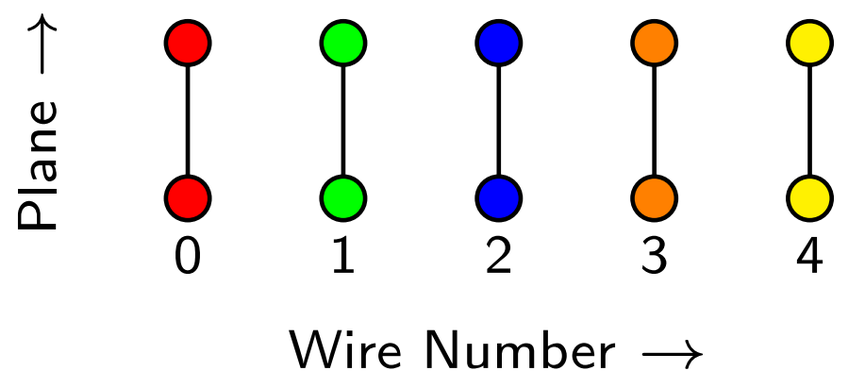


Ex. 2: Angled Track Parallel to YZ Plane

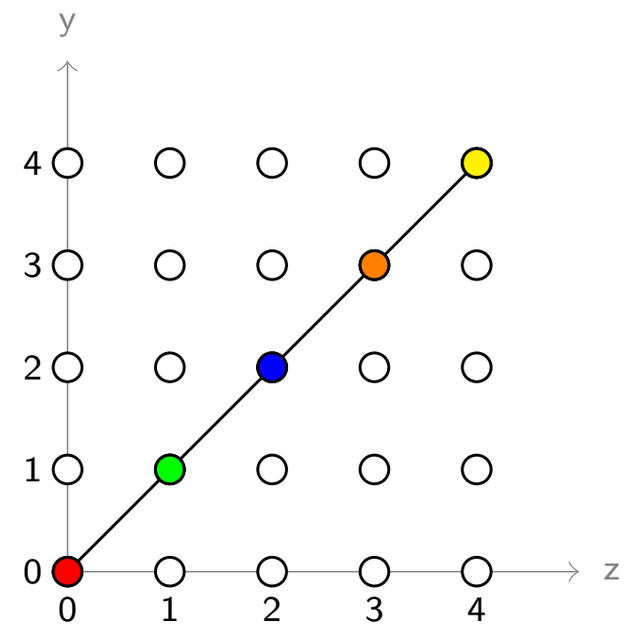
- Again, we associate hits from different views that coincide in time, and are on wires that intersect.
- Again, there is only one SpacePoint for this example...but it's more complex than previous example.



One possible network (which happens to be the right one) within the SpacePoint.

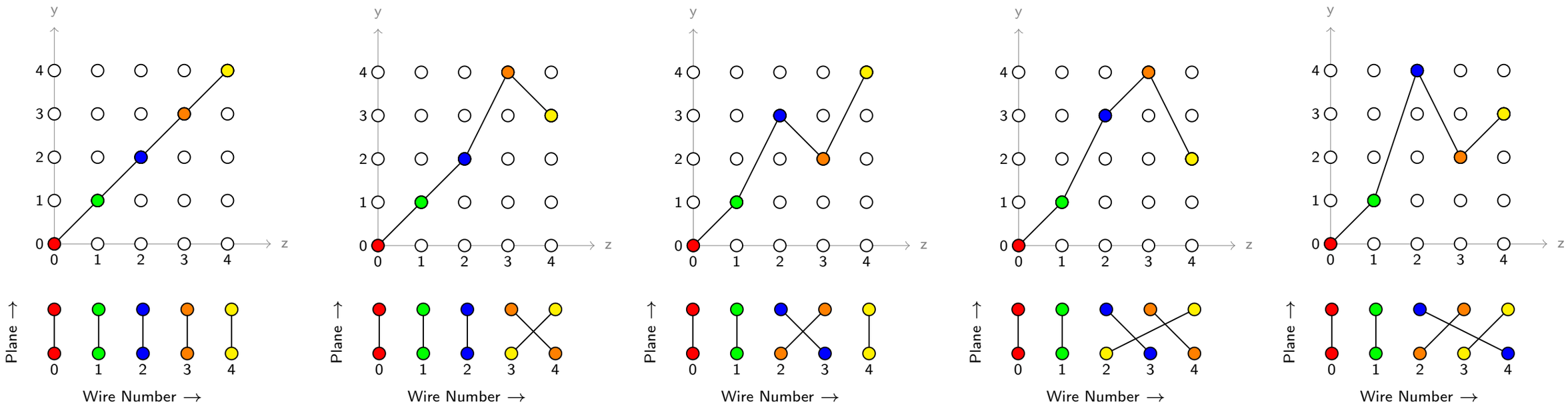


Corresponds to the following trajectory →

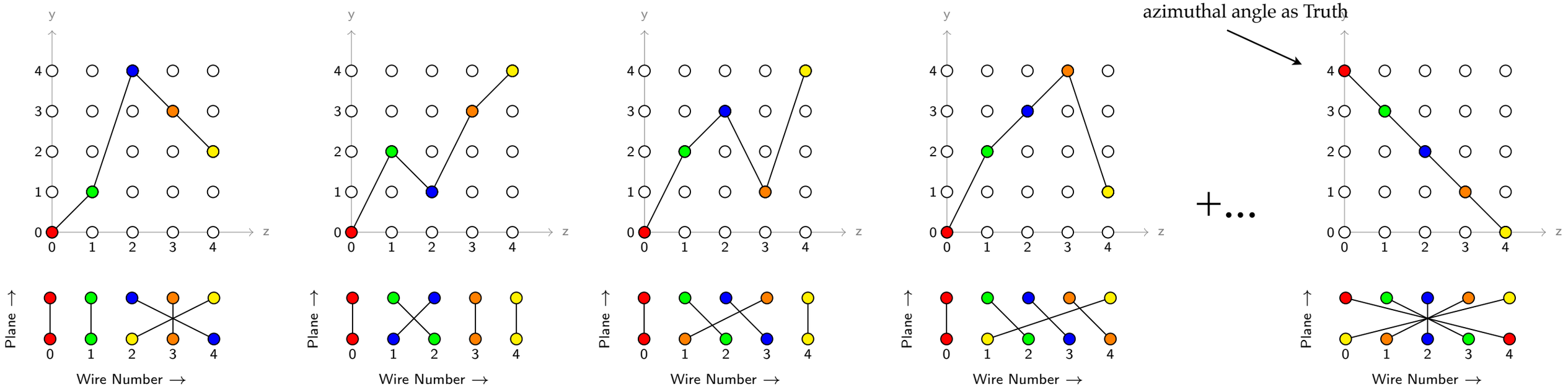


Ex. 2 - Angled Track Parallel to YZ Plane

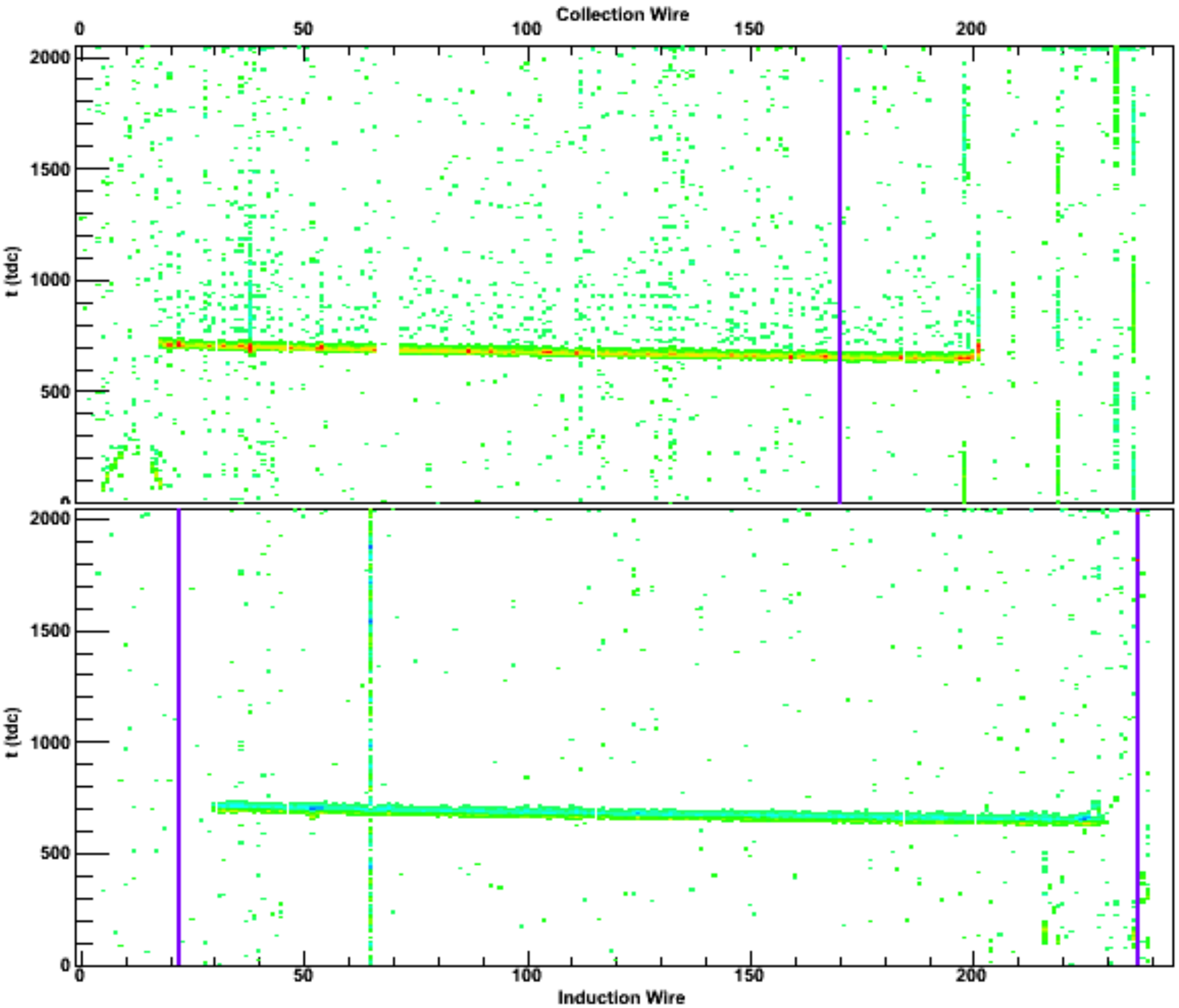
However, there are 120 valid trajectories for this example...



Remember: Each allowable network within the SpacePoint corresponds to a different 3D trajectory.



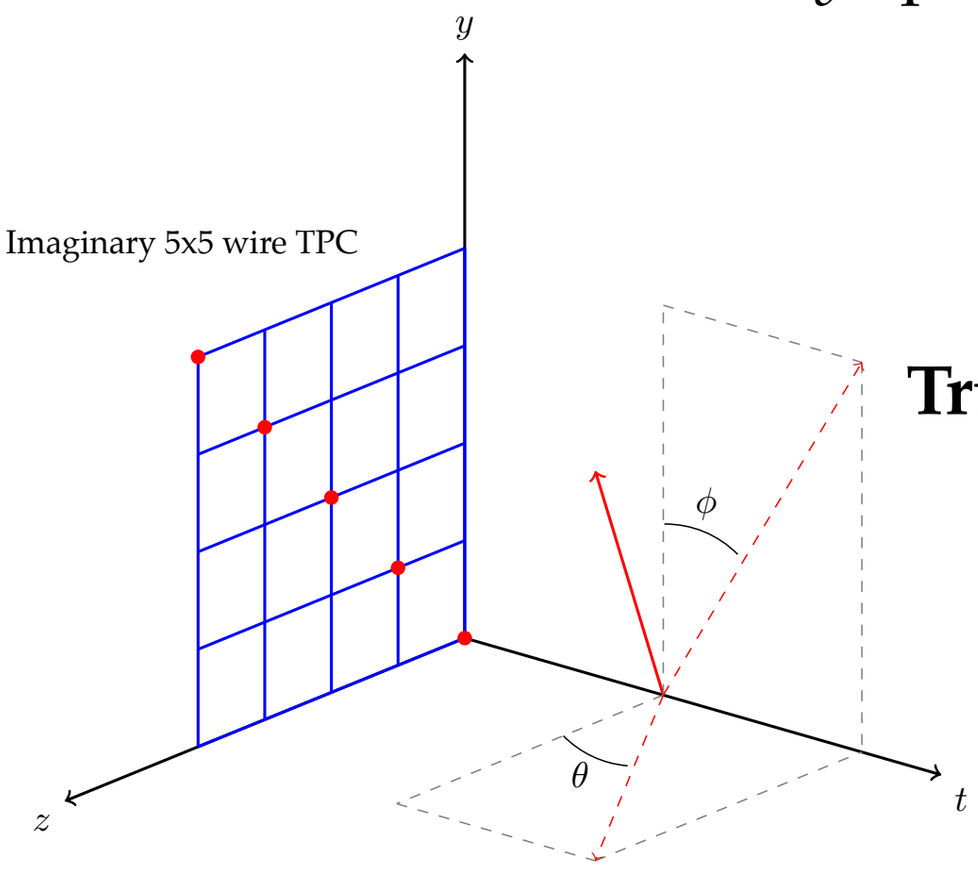
Muon Example



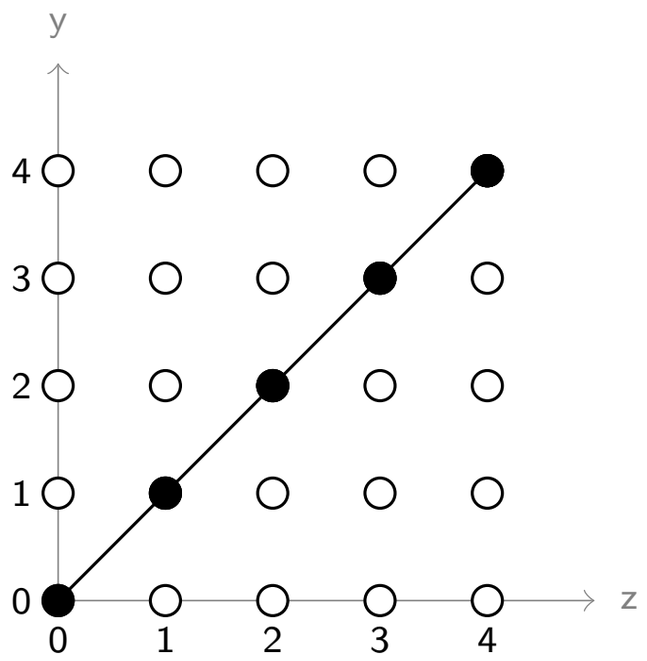
- Muon that is fairly flat in drift direction.
- Angled wireplanes can help us eliminate some potential degeneracy, since not all wire pairs overlap.
- Track is about 180 wires long in Collection view, and 200 wires long in Induction view.
- How do we reconstruct?

Ex. 3: Angled Track

Now add some rotation w.r.t. yz plane...(i.e. - allow a non-zero azimuthal angle, θ)

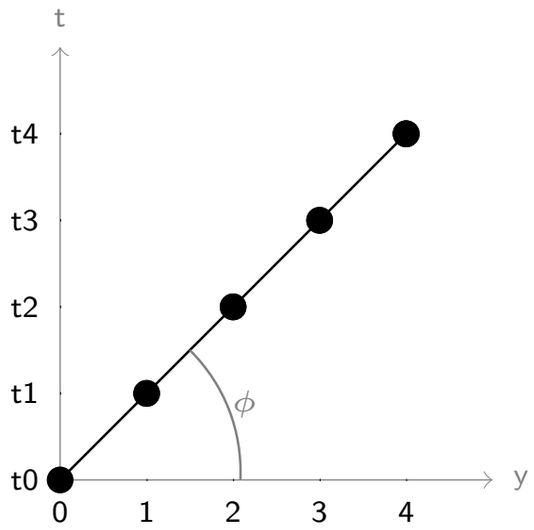


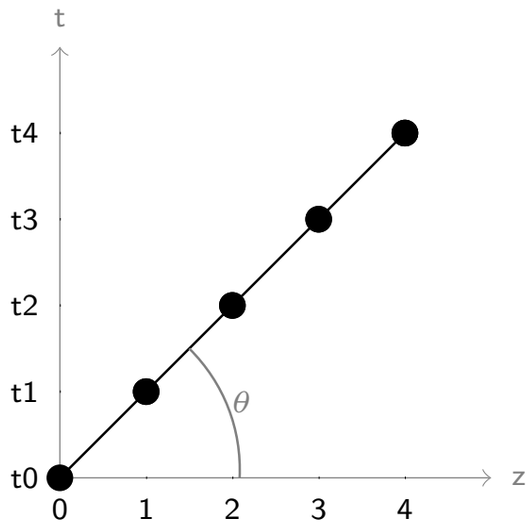
Truth Information



What we actually observe (again ignoring Hit characteristics).

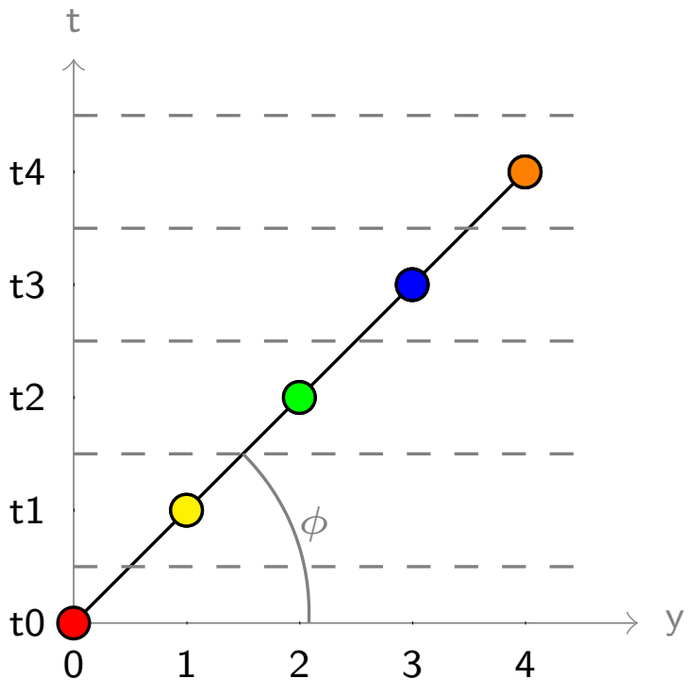
Note: ϕ and θ not necessarily equal.

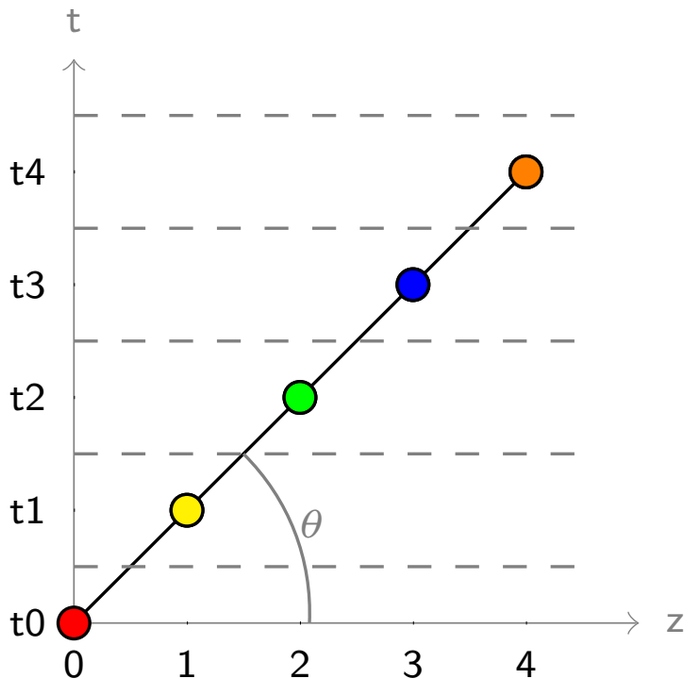




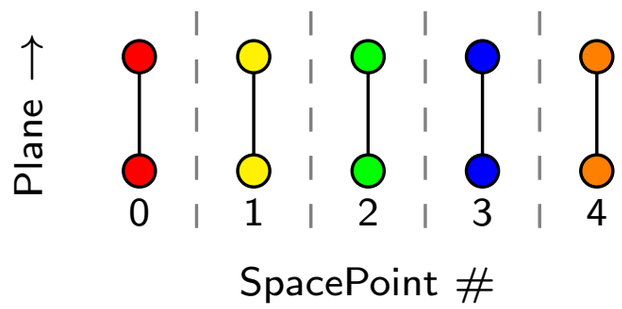
Ex. 3: Angled Track

If orientation angles are sufficient, degeneracy of Ex. 2 is broken since we can break the track up into several SpacePoints.

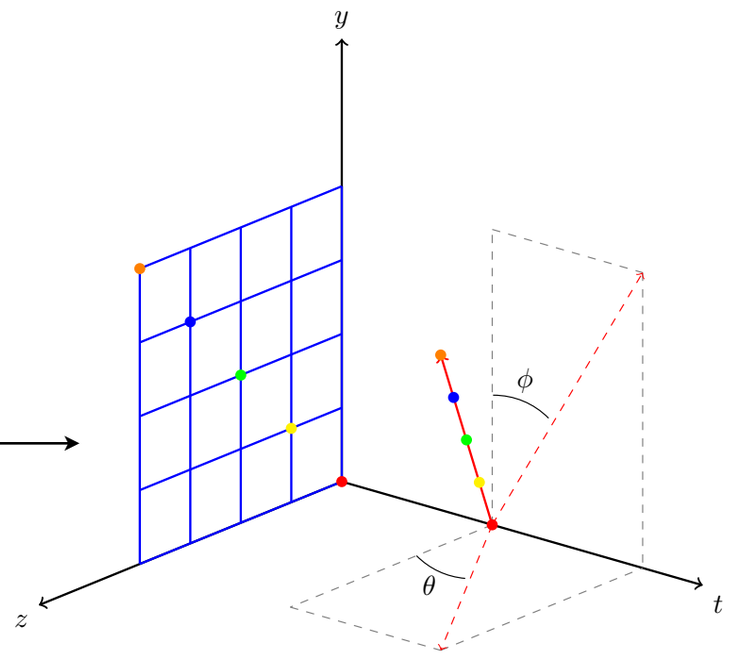




Now we have 5 SpacePoints, each with a trivial network. Immediately can obtain the 3D Trajectory.

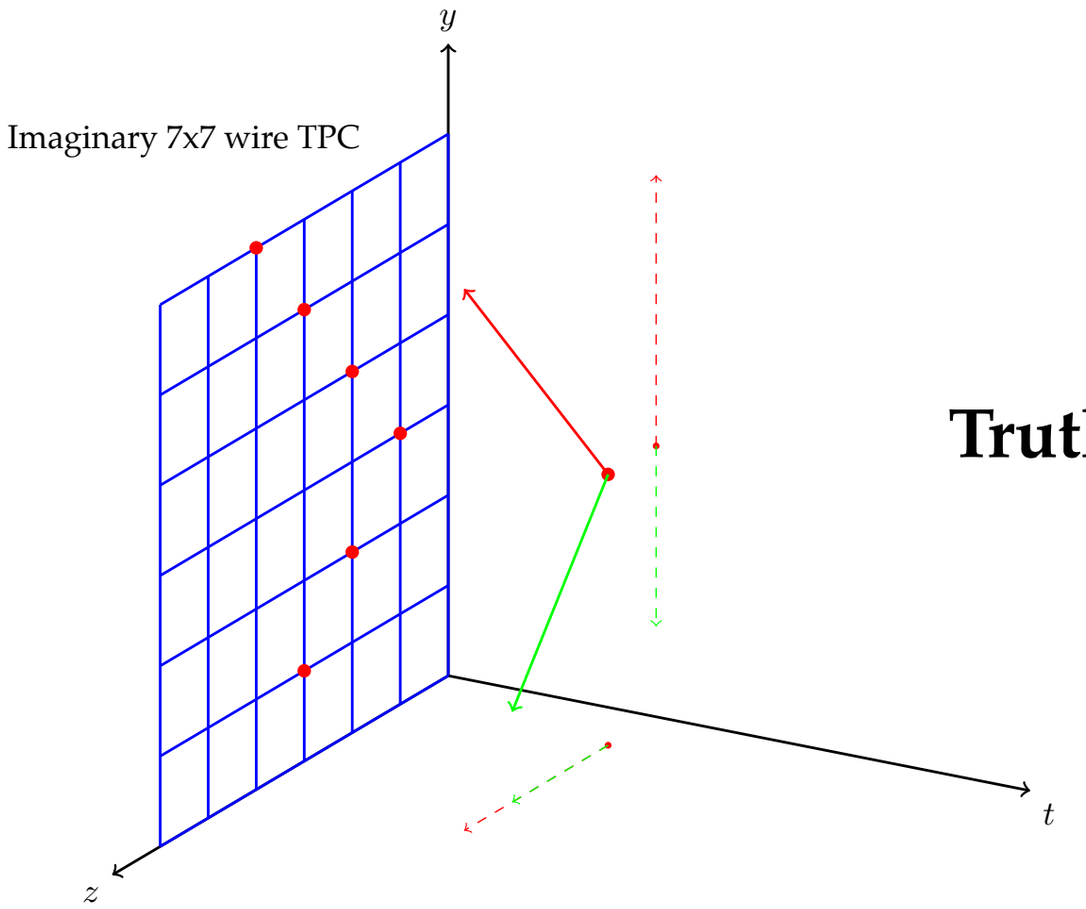


Corresponds to the following trajectory

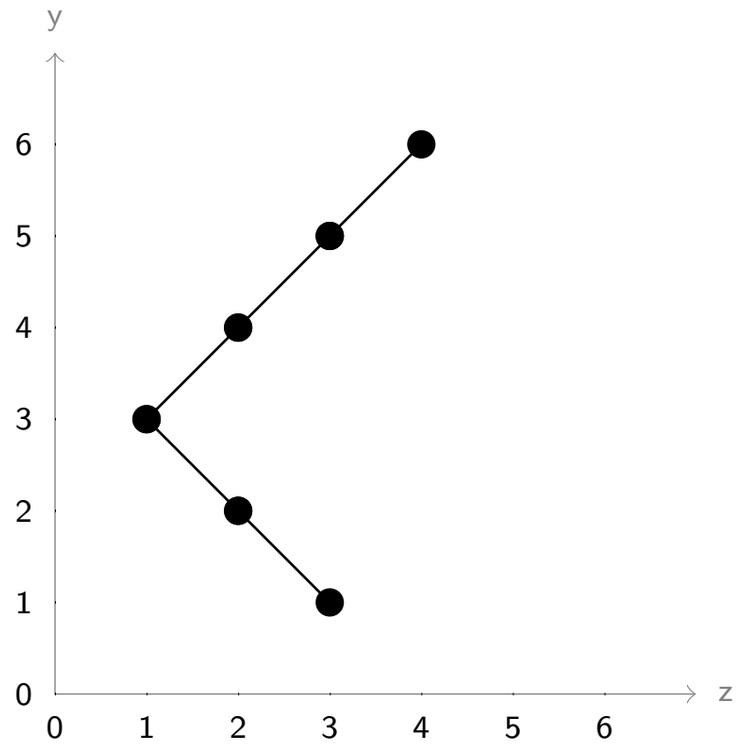


Ex. 4: Two-Track Event Parallel to YZ

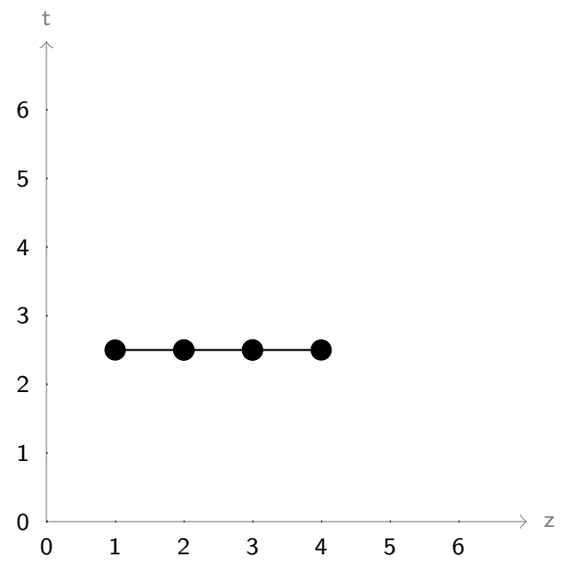
Event with two tracks, occurring at t=2.5 (arbitrary units).

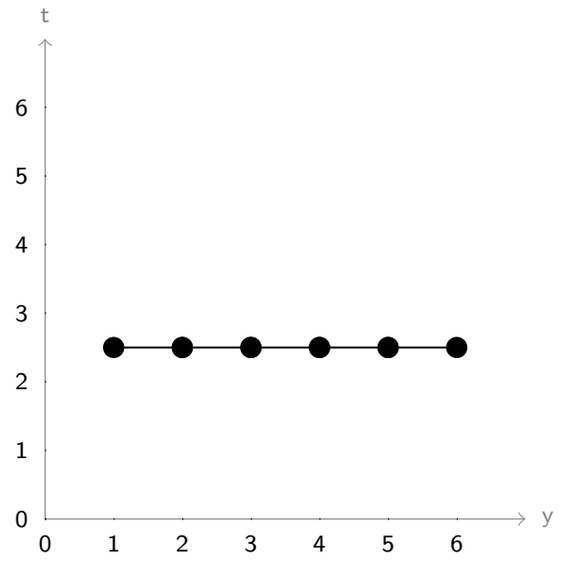


Truth Information



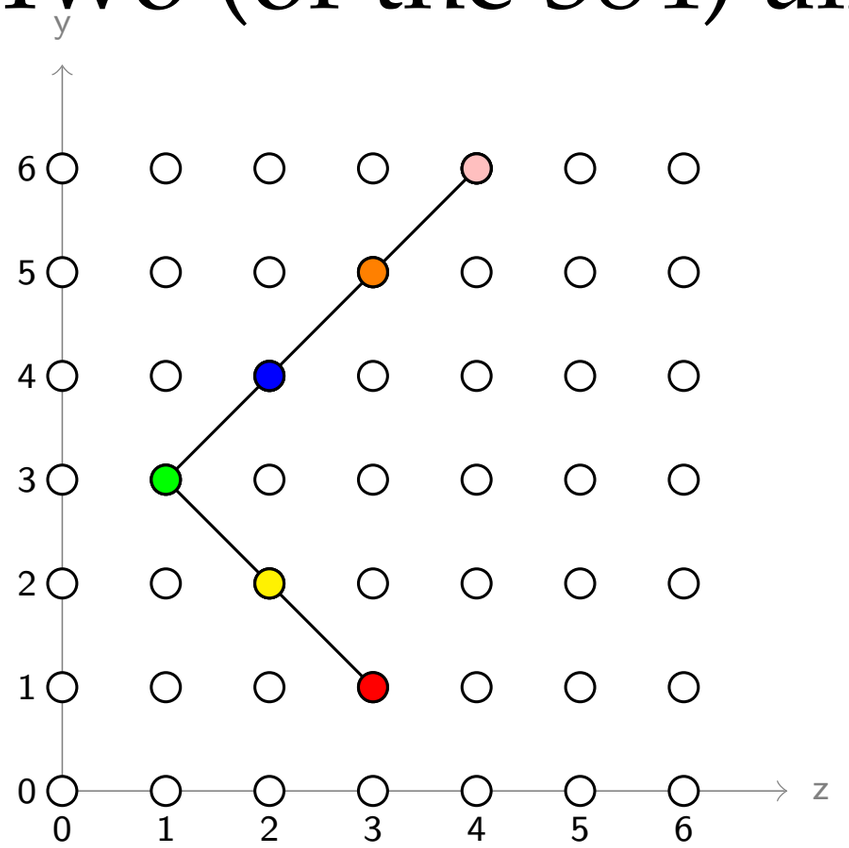
What we actually observe (again ignoring Hit characteristics).

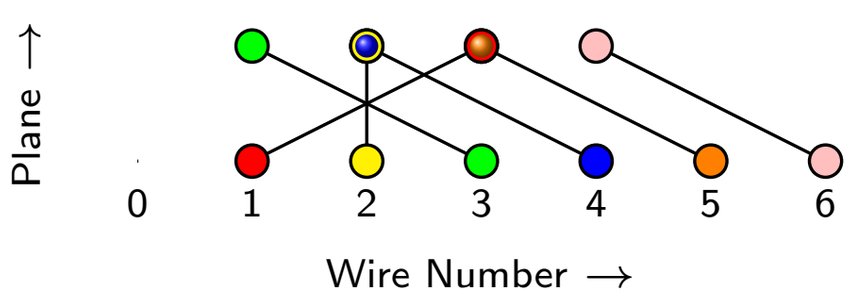


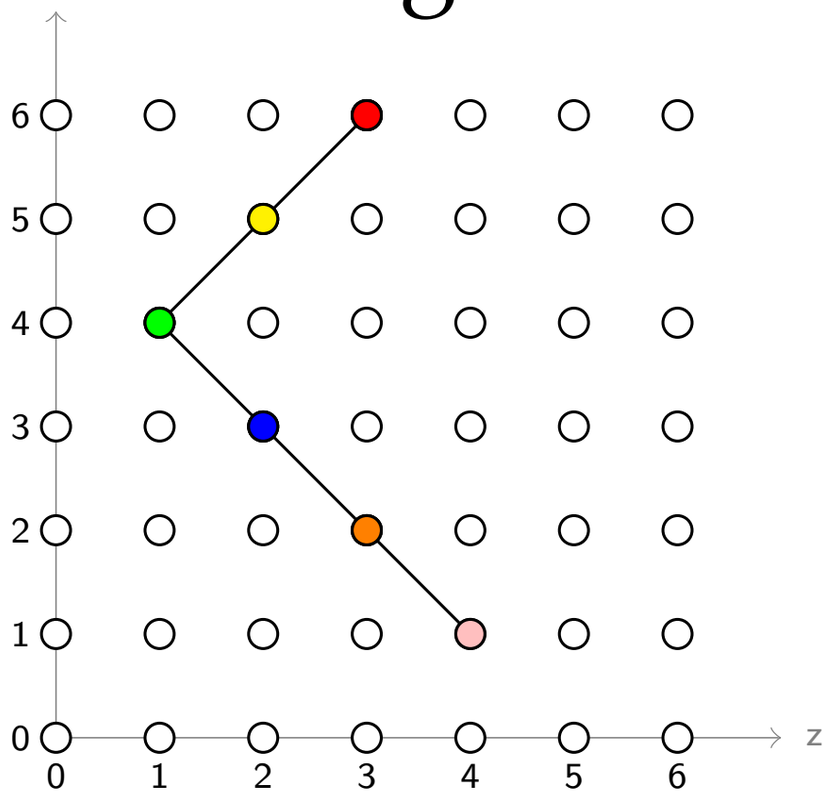


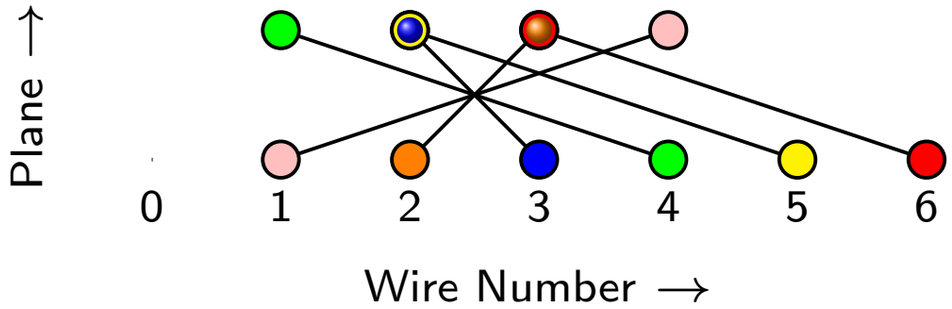
Ex. 4: Two-Track Event Parallel to YZ

Two (of the 384) allowable arrangements...





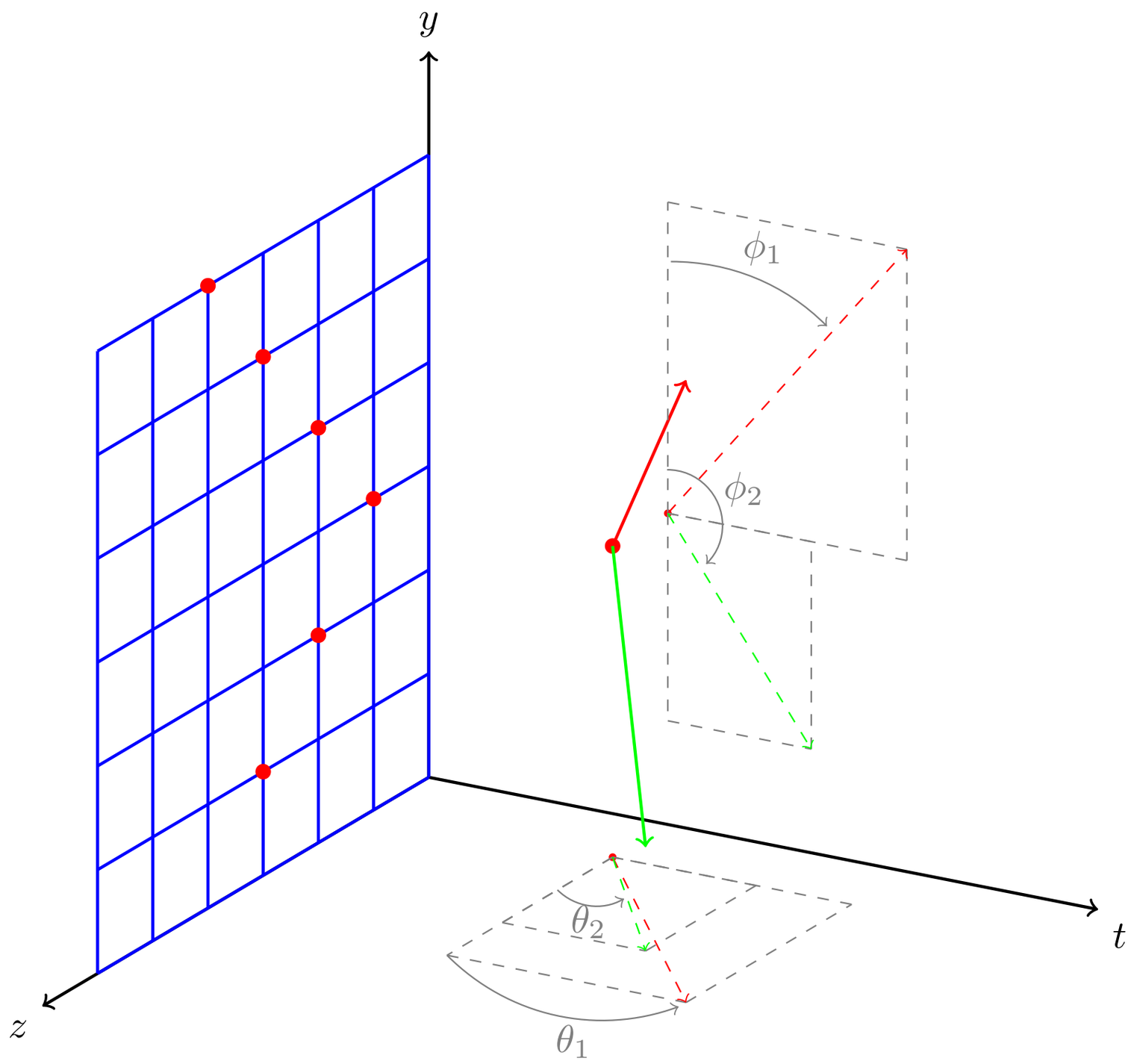




Are we stuck assuming all trajectories are equally valid? Clearly we need additional constraints.

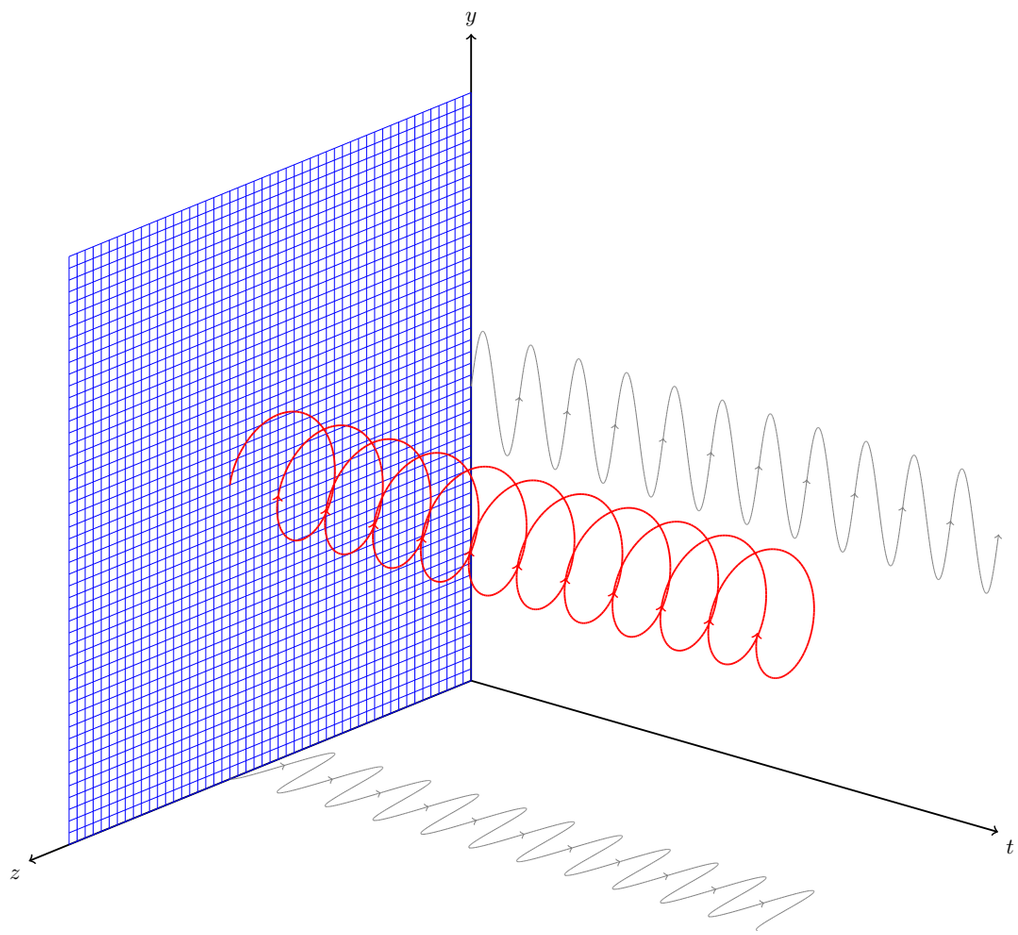
Ex. 5: Two-Track Case

In general the two tracks will have different orientations with respect to the drift axis, which hopefully breaks the degeneracy of the previous example, which should improve reconstruction.

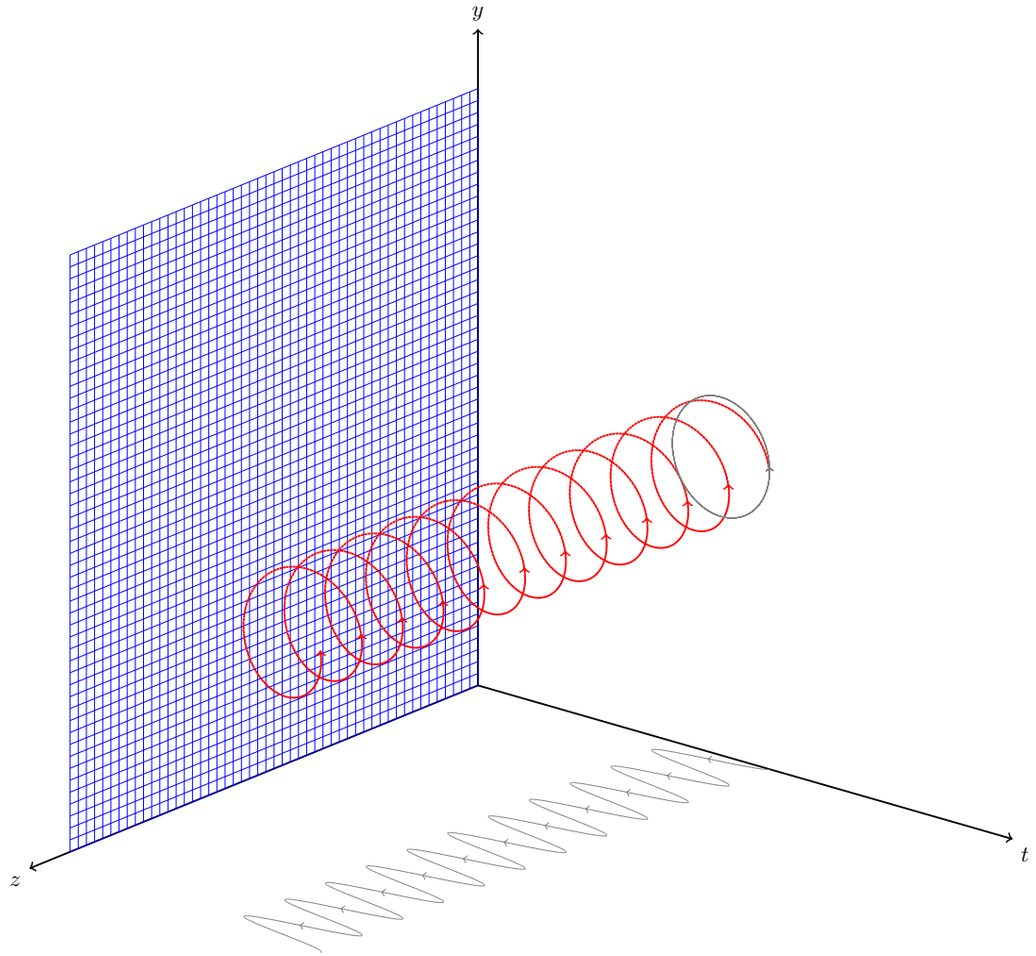


Ex. 6 - Helical Trajectory

- Consider a particle following a helical trajectory traveling through the TPC.
  - This is just an example...obviously we need a magnetic field for this to happen.
- If the "pitch" of the helix is wide enough (e.g. - to spread successive hits in time sample bins further apart than our resolution), and the direction of travel is inclined sufficiently w.r.t. *yz* plane, such a trajectory should be reconstructable by associating SpacePoints from the two views.



Some hope for complete Reconstruction



Much more difficult to Reconstruct due to complicated (circular) projection in *yt* plane.

Generalized 3D Tracking

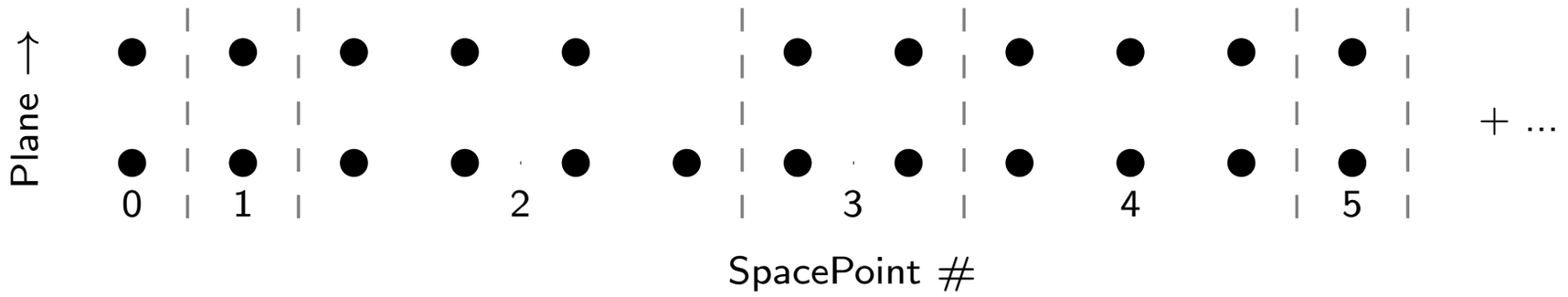
How do we handle more complicated events, potentially with multiple tracks?

- Previous example illustrates the potential to reconstruct a particle following an arbitrary curved trajectory.
- For full 3D event reconstruction we will need to fold in all available constraints to reduce the combanitorics. Such constraints may include:
  - Geometry - which wires within a SpacePoint intersect?
  - Energy - is the energy of the Hits associated within a SpacePoint consistent?
  - Continuity - require neighboring SpacePoints to have trajectories that match on the borders

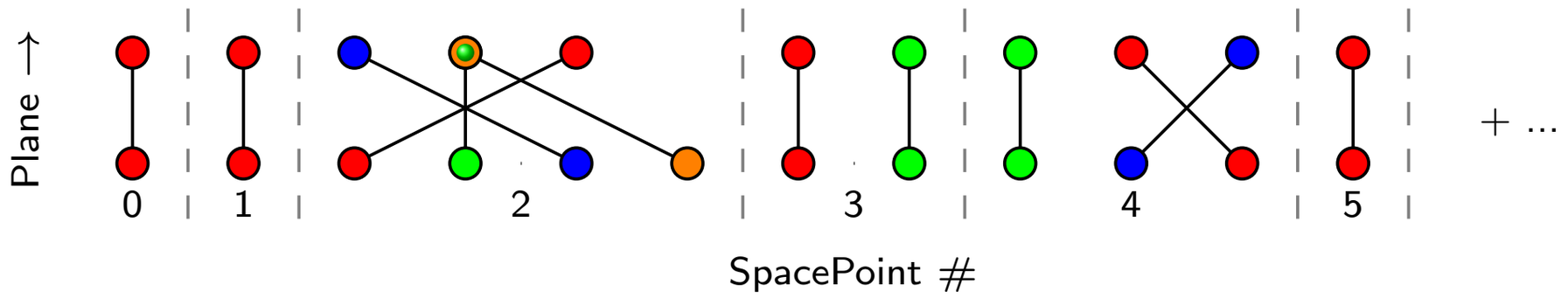
Comments on Reconstruction

- Lessons learned:
  - **A SpacePoint is not always a single point in space...(perhaps we should rename?)**
  - When creating a SpacePoint from input Clusters, we must associate all Hits under consideration that are within a given time slice.
  - A trajectory (a unique path through X,Y,Z space) is created by linking together the Hits within each SpacePoint. Linked Hits must occur on wires that intersect at some point.
  - When forming a trajectory, we must include all Hits in each SpacePoint at least once. We need not require the same number of Hits from different planes under consideration.
- To discuss next time:
  - How do we decide which Clusters from each plane to compare when forming SpacePoints?
  - What are the rules that govern creating a trajectory from a given set of SpacePoints?
  - What constraints can we use in reducing the number of allowable trajectories?

Set of all SpacePoints



Possible Trajectory



Conclusions

- Looking at simple examples gives us ideas for how to proceed with 3D reconstruction.
- Particle trajectories need not follow simple straight lines, and a generalized tracking algorithm should allow for such eventualities.
- For a real size detector (ArgoNeuT...let alone MicroBooNE or LBNE), number of allowable combinations within SpacePoints will pile up quickly for trajectories of very modest lengths oriented fairly perpendicular to the drift axis.
- We need additional constraints to cut out as many meaningless trajectories as possible.