Tracker algorithm based on Hough transformation

Nick Sinev, University of Oregon

Santa Cruz Linear Collider Retreat, June 29, 2002

Why to bother ? Don’t we have good working tracking code?

- Yes, we do. Current algorithm is based on trying every combination of tracker hits in 3 layers. It always will work, and in that sense is “ideal” or “standard” algorithm. It can work for any configuration of tracking detector – barrel, endcap, planes, disks, and so on.
- The only problem with it – huge amount of combination to try. We may be concerned about reconstruction time in the case of large background – say, 1000 hits/layer. In that case number of combination to try is 10^9 and each try is at least 100 flops, so 10^{11} flops/event will take 100 seconds even on fastest “general public” computer. We probably don’t want to use Supercomputers for track reconstruction.

Hough transformation may be faster

- The idea of the method for detecting curves in binary image data was patented by Paul Hough in 1962
- The major idea of the method – mapping “image space” points into “parameter space” curves and looking for set of parameters which satisfies maximum number of image space points.
- This method requires number of operations proportional to the total number of points in the image – not 3rd power of this number, as in the case of “standard” reconstruction algorithm.

BaBar level 1 trigger experience

- Similar ideas were used in the new design of the BaBar level 1 trigger, which does very fast rough reconstruction of tracks to cut on background tracks, not originated from IP. Algorithm uses image – parameter space mapping similar to Hough algorithm, though uses relative position of 2 points in image space, not single point coordinates.
- Algorithm is ideal for specialized processor, based on FPGA circuits and allows tracks reconstruction for an event in less than 5 microseconds .

What I did

- The code has been written to try to speed up tracking in the high background environment. The code is called “Hough Transformation Tracker”, though it is not “pure” Hough Transformation based tracking, rather something between BaBar level 1 design and Hough idea.
- The code uses image – parameter space mapping for first stage of track finding. It uses “seed” hits in one layer, and relative to the seed position of hits in few (up to 10) other layers to estimate parameters of possible track traversing the seed. These parameters are used to limit search for two more hits in another layers, and then such combination of 3 hits selected, “standard” algorithm takes over.

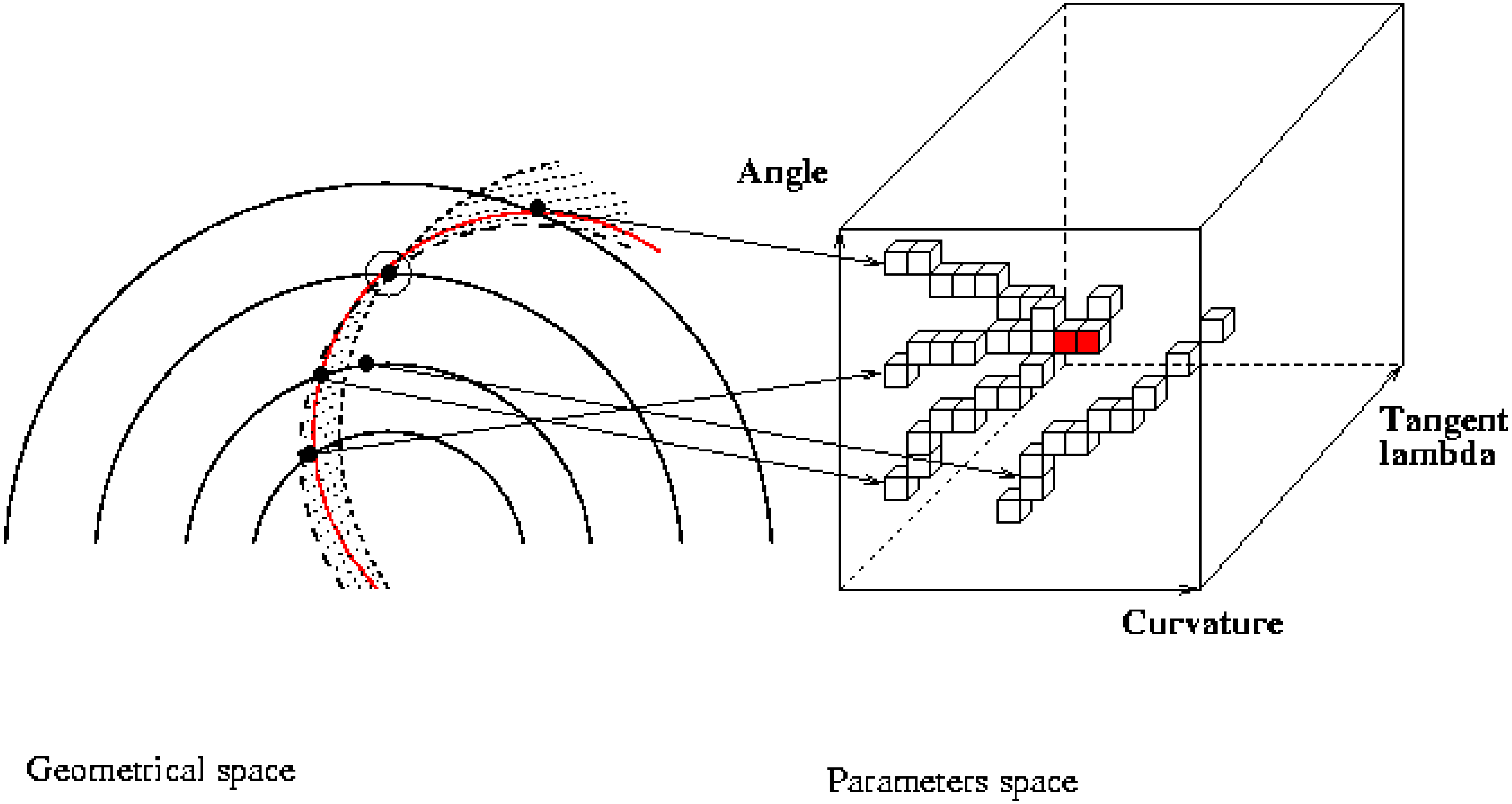
Why not “pure” Hough Transformation

- The “standard” algorithm does not use IP constrains on the tracks. In fact, such constrains would reduce number of combination and it’s grow with background will be only as 2nd , not 3rd power of number of hits. But if we want to avoid such constrains, when “pure” Hough Transformation algorithm would need 5-dimensional parameter space. It requires too much memory for accumulator – of the order of 10 GB. The algorithm with the “seed” hit requires only 3-dimensional parameter space – the track coming through given point may be described with 3 parameters, for example: curvature, tangent lambda and the XY plane angle of the track at seed. 3 dimensional accumulator needs only 1 MB of memory.

Implementation details

- For each hit in seed layer:
  - For each hit in one of the pattern layers:
    - Difference of phi and z relative to the seed hit is calculated
    - For every bin of omega lookup table is used to get track XY plane angle at seed location and track tangent lambda. The hit count in the element in the parameter space, corresponding this set of omega, angle, tangent lambda is increased.
  - After all pattern layers are processed, search is made for element in parameter space with maximum number of hits. This gives the track parameters for the best track hypothesis. These parameters are used to build “track”, and then 3 hits close to it in 3 different layers are used to build more precise track – here the “standard” algorithm kicks in.
- One more pass of the same algorithm is performed with another set of layers

Implementation details



Performance

- It is not optimized yet. Comparing with standard algorithm we need to remember, that I spent almost 6 months optimizing standard algorithm. The Hough Transformation written from scratch in slightly more than a month. So, I believe, that at least 5-fold increase in speed can be achieved, if spend more time on debugging and optimizing.

Time of one T-Tbar event with background reconstruction

Standard		Hough
Without background -	3 sec	--- 45 sec
2x number of hits -	22 sec	-- 140 sec
4x number of hits -	103 sec	-- 480 sec
6x number of hits -	320 sec	-- 800 sec

Conclusion

- The tracking code based on Hough transformation really has quadratic instead of cubic for standard algorithm dependence of the reconstruction time on the number of hits in detector. These may be essential for high background reconstruction.
- Code still need some optimization, but for most detector simulation standard code is faster, and there is no need in using Hough transformation code.
- The code may be used (after optimization) to prove tracking ability for extremely high background (up to 10000 hits/layer).
- For the real detector special tracking processor may be developed, using such kind of code and DSP hardware.