GPU accelerated hough transform for high level trigger application

S. Mohanty[1]* and P. Hristov[2]
[1] Department of Computer Science and Engineering,
Visvesvaraya National Institute of Technology, Nagpur - 440010 and
[2] EP, CERN, CH-1211, Geneva 23, Switzerland

Introduction

The charged particle trajectory recognition is a very complex and time consuming process particularly when the multiplicity becomes very large during the heavy ion collisions at relativistic energies. Since all the events may not be of interest to record, it is essential to have a fast algorithm to select the desired events through online triggering which is known as high level trigger(HLT). ALICE (A Large Hadron Collider Experiment) is a general purpose detector, being used at CERN, LHC to study nucleus-nucleus and proton-proton collisions at different centre of mass energies [1]. The task of the ALICE High level trigger (HLT) is to select the events of interest using various fast track reconstruction algorithms. A fast track-reconstruction algorithm for the time projection chamber of ALICE detector has been discussed in Ref. [2] based on linear Hough Transform (HT). In this work, a similar implementation of HT is done using graphic processing unit(GPU), using both circular and linear HT with simulated data as discussed below.

Hough Transformation

The hough transform is a technique which can be used to isolate features of a particular shape within many patterns, and is most commonly used for the detection of regular curves such as lines, circles, ellipses etc. The simplest of the hough transform is the linear transform for detecting straight lines. In the parameter space, the straight line can be described as $y = mx + b$ and can be graphically plotted for each pair of data points (x_i, y_i). In the hough transform, the main idea is to consider the characteristics of the straight line not as data points (x_i, y_i), but instead, in terms of its hough parameters m and b. Therefore, a striaght line in the parameter space, represnts a single point in hough space. HT makes use of an array called the accumulator to detect the existence of line or curve. The dimensions of the accumulator is equal to the number of unknown parameters of the HT problem. The voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so called accumulator space that is explicitly constructed by the algorithm for computing the HT.

TPC Track Model

The track model of a charged particle trajectory in presence of a magnetic field is a helix. The trajectory can be described by a circle in the R-ϕ plane of the TPC. The circle equation with center a and b passing through the origin is given by, $(x - a)^2 + (y - b)^2 = r^2$ which can be written in the parametric form as,

$$k = \frac{2}{R} sin(\phi - \phi_0) \quad (1)$$

where $k = 1/r$ is the radius of curvature, $R = \sqrt{x^2 + y^2}$, ϕ_0 is the azimuthal angle at $(0, 0)$ and $\phi = tan^{-1} \frac{y}{x}$.

The above circle equation can be transformed into a straight line through conformal mapping with the prescription,

$$u = \frac{x}{x^2 + y^2}, \quad v = \frac{y}{x^2 + y^2} \quad (2)$$

*Electronic address: siddhant9892@gmail.com

If r is fixed by the relation $r^2 = a^2 + b^2$ the following straight line is obtained,

$$v = \frac{1}{2b} - \frac{a}{b}u. \qquad (3)$$

Both Eq.3 and Eq.1 can be used for hough transformation, although a linear model is faster and suitable for HLT application.

Implementation in GPU

We have simulated curve trajectories in the x-y plane to mimic the transverse trajectories in a Time Projection Chamber(TPC) of ALICE detector. To detect the trajectory parameters, we have implemented the hough transformation using both Eq.1 and Eq.3. In case of the circular hough transform, the independent parameters in hough space are ϕ_0 and $p_T(k)$. Similarly under conformal mapping the hough parameters are u and v. Both the methods are implemented using CPU (Intel Core2 Duo, 3.16GHz) and GPU (NVIDIA GeForce GTX 480). For GPU programming, NVIDIA CUDA is used, although it can be implemented in OpenCL as well. While implementing in GPU, each data point is associated with a given thread. So, N threads are used, spread over several blocks corresponding to N data points.

Results and Discussions

To compare the relative performance, we have fixed the block size at 1024 threads which is the maximum possible block size for NVIDIA GeForce GTX 480 and estimated various execution times as a function of number of blocks. Figure 1 shows the execution time versus the number of blocks, for both circular and linear hough transform using CPU and GPU. As expected, circular hough transform takes more time than the linear hough implementation in CPU.

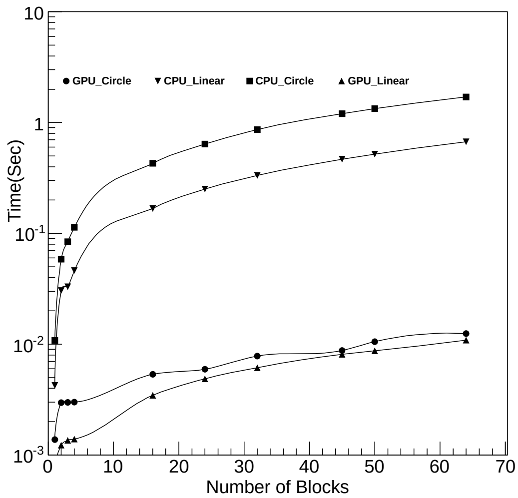

FIG. 1: The block number(unit of 1024 threads) versus execution time in sec.

Conclusion

In conclusion, the GPU implementation for circular hough transform is around 120X times faster than the corresponding CPU implementation. Whereas, it's about 60X times faster in the case of linear transform using conformal mapping. It is interesting to note that the GPU execution time both for circular and linear hough transforms is nearly equal although circular transform takes more time than linear transform in CPU. This is an important observation as circular hough tranform can still be used for the all ϕ ($0 \leq \phi \leq 2\pi$), whereas linear hough transform requires data to be divided into sectors so that the situation corresponding to lines having large slope (angle close to 90^o) can be avoided. The above conclusion is drawn based on simulation and the method will be implemented using ALICE TPC data.

Acknowledgments

S. Mohanty is thankful to F. Carminati for inviting to work as a summer intern with ALICE offline group at CERN.

References

[1] ALICE Collaboration 2000 2002 **TDR of Time Projection Chamber** (CERN/LHCC/2000-001).

[2] C Cheshkov 2006 **Nuclear Instruments and Methods in Physics Research** A566 p 35-39.