

CPU vs. GPU: Initial timing measurements of the hough transform algorithm

Victoria Xu
INFN GAP Meeting
July 17, 2014

Hough transformation algorithm

1. Transform cartesian coordinates into parameter space (polar plane) - **GPU**
 - a) For each point (x,y): plot the family of lines which could pass through it (becomes a sinusoid in the polar plane)
$$r = x \cos(\theta) + y \sin(\theta); \quad r = [-r, r]; \quad \theta = [0, 180]$$
 - b) Fill the polar plane (a grid of theta (horizontal axis), radius (vertical axis)) with the family of lines that could pass through each point
 - c) An intersection of sinusoids in the polar plane = points which belong to the same line
2. Search through the parameter space for local maxima & extract theta, radius coordinates - **GPU**
 - a) Search through the parameter space grid for threshold crossings
 - b) Check if the threshold crossing is a local maxima (check if the weight of the bin in parameter space is greater than the weight of neighboring bins. if so, then the bin is a local maxima)
3. Construct and draw lines corresponding to the locations of local maxima in the cartesian plane, for ease of comparison between detected lines and existing lines

A current timing measurement (very preliminary...)

n_entries = 112	GPU	CPU
Filling parameter space	1.915 ms	1.582 ms
Searching for maxima	1.289 ms	0.522 ms

n_entries = 352	GPU	CPU
Filling parameter space	2.385 ms	4.079 ms
Searching for maxima	1.290 ms	0.538 ms

cpu_hough_transform.C

```
Reading hits data...

Hough Transform: Fill parameter space from 352 entries...
Real time 0:00:00.004079, CP time 0.000

Searching for maxima in parameter space with threshold = 10...
Real time 0:00:00.000538, CP time 0.000

Reconstructing the lines found...
y(x) = 0.726543x + 2.03393
y(x) = 1.03553x + 2.75807
y(x) = 1.03553x + 3.47785
y(x) = 1.48256x + 4.81068

Time for entire macro (w/o plotting & creating data):
Real time 0:00:00.004738, CP time 0.000
```

1. Fill parameter space (TH2D, 2-D Histogram)
2. Search for maxima in parameter space
 - a) Find threshold crossings
 - b) Check if threshold crossings are local maxima
3. Reconstruct a line in the cartesian plane from the radius and theta coordinates of local maxima

gpu_hough_transform.C

```
Simulating hits data...
Plotting  $y(x) = 1.5x + 5...$ 
Plotting  $y(x) = 1.05x + 3...$ 
Plotting  $y(x) = 0.735x + 1.8...$ 

Reading hits data into arrays...

Hough Transform: Fill parameter space from 352 entries...
Real time 0:00:00.002385, CP time 0.000

Searching for maxima in parameter space with threshold = 10...
Real time 0:00:00.001290, CP time 0.000
Found 4 threshold crossings

Reconstructing the lines found...
 $y(x) = 0.726543x + 1.8541$ 
 $y(x) = 1.03553x + 3.59889$ 
 $y(x) = 1.48256x + 5.36487$ 
 $y(x) = -1.03553x + 119.483$ 

Time for entire macro (w/o plotting & creating data):
Real time 0:00:00.003917, CP time 0.010, 2 slices
```

1. Fill parameter space (1-D array) kernel: 1 thread = 1 theta - GPU
2. Search for maxima in parameter space - GPU
 - a) Kernel: Find threshold crossings & check if threshold crossings are local maxima
 - b) Stream compaction - only copy back the array holding the values of maxima
3. Reconstruct the line from the radius and theta coordinates of local maxima - CPU

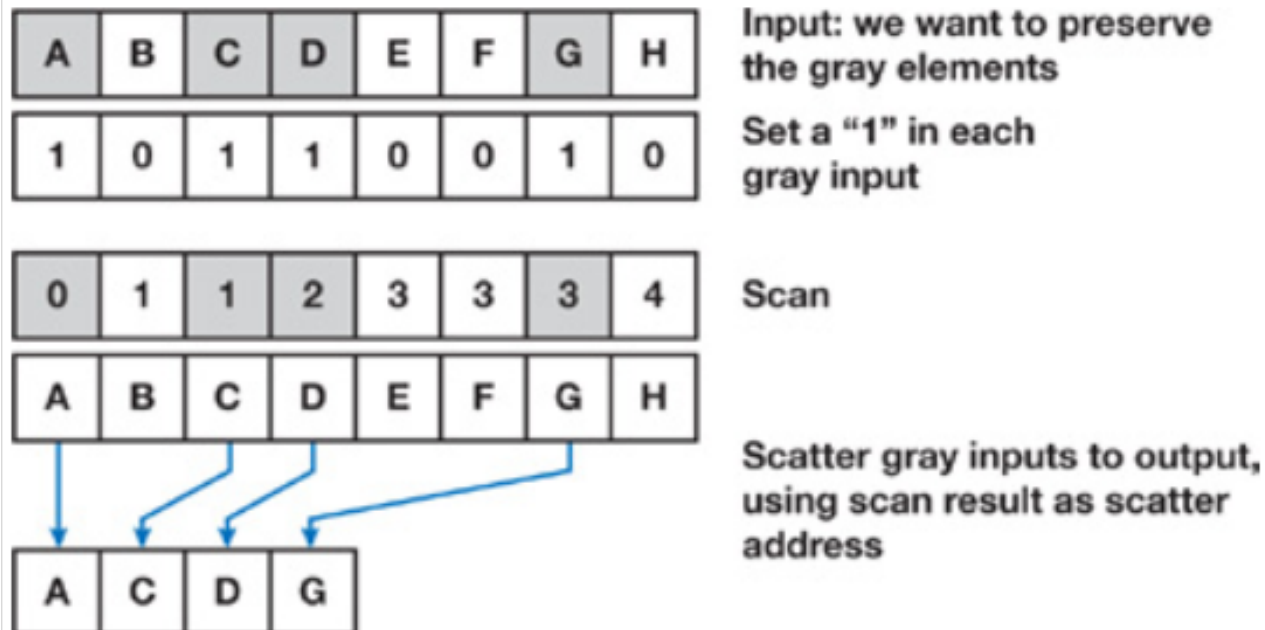
Stream Compaction: CUDA C thrust library

Figure 39-9 Stream Compaction Example

Stream compaction requires two steps, a scan and a scatter.

1. The first step generates a temporary vector where the elements that pass the predicate are set to 1 and the other elements are set to 0. We then scan this temporary vector. For each element that passes the predicate, the result of the scan now contains the destination address for that element in the output vector.
2. The second step scatters the input elements to the output vector using the addresses generated by the scan.

Figure 39-10 shows this process in detail.



gpu_hough_transform.C

```
Simulating hits data...
Plotting  $y(x) = 1.5x + 5...$ 
Plotting  $y(x) = 1.05x + 3...$ 
Plotting  $y(x) = 0.735x + 1.8...$ 

Reading hits data into arrays...

Hough Transform: Fill parameter space from 112 entries...
Real time 0:00:00.001915, CP time 0.000

Searching for maxima in parameter space with threshold = 7...
Real time 0:00:00.001289, CP time 0.000
Found 3 threshold crossings

Reconstructing the lines found...
 $y(x) = 0.726543x + 2.47214$ 
 $y(x) = 1.03553x + 3.59889$ 
 $y(x) = 1.48256x + 5.36487$ 

Time for entire macro (w/o plotting & creating data):
Real time 0:00:00.003388, CP time 0.000, 2 slices
```

cpu_hough_transform.C

```
Reading hits data...
```

```
Hough Transform: Fill parameter space from 112 entries...
```

```
Real time 0:00:00.001582, CP time 0.000
```

```
Searching for maxima in parameter space with threshold = 7...
```

```
Real time 0:00:00.000522, CP time 0.000
```

```
Reconstructing the lines found...
```

```
 $y(x) = 0.726543x + 2.03393$ 
```

```
 $y(x) = 1.03553x + 3.47785$ 
```

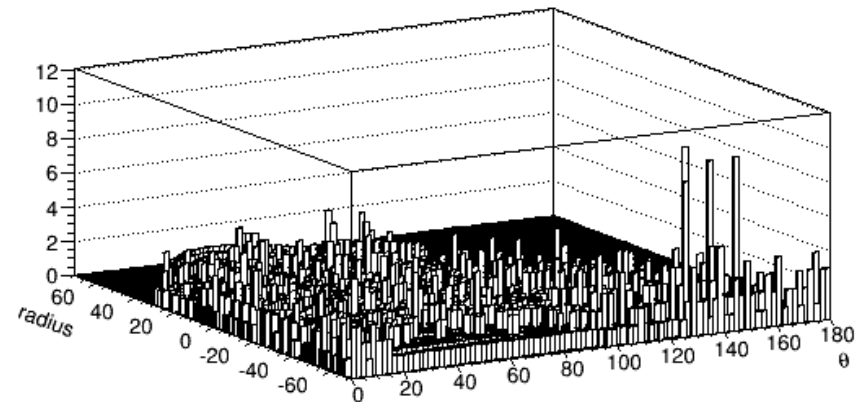
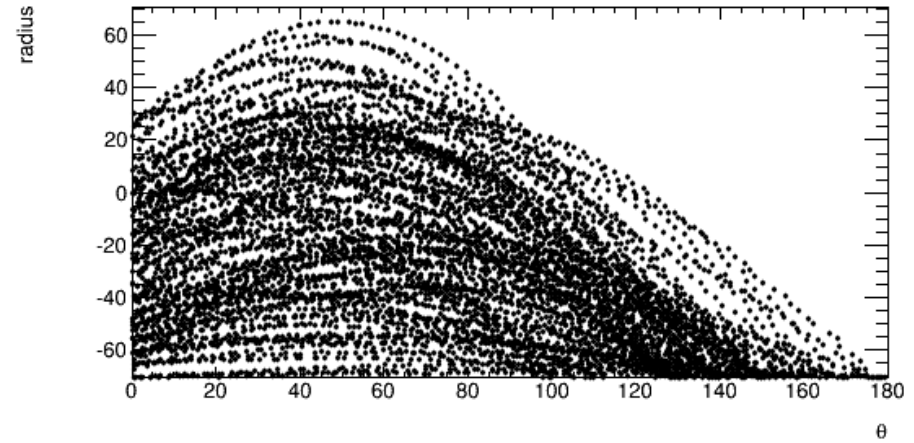
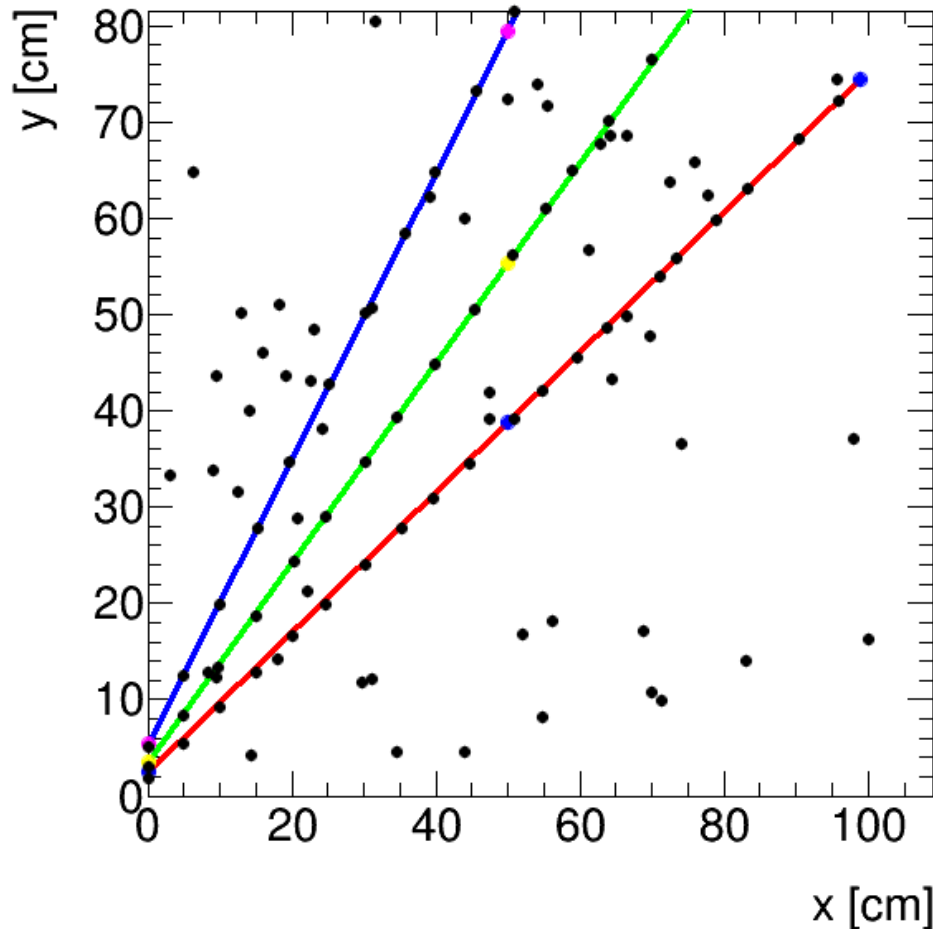
```
 $y(x) = 1.48256x + 4.81068$ 
```

```
Time for entire macro (w/o plotting & creating data):
```

```
Real time 0:00:00.002213, CP time 0.000
```

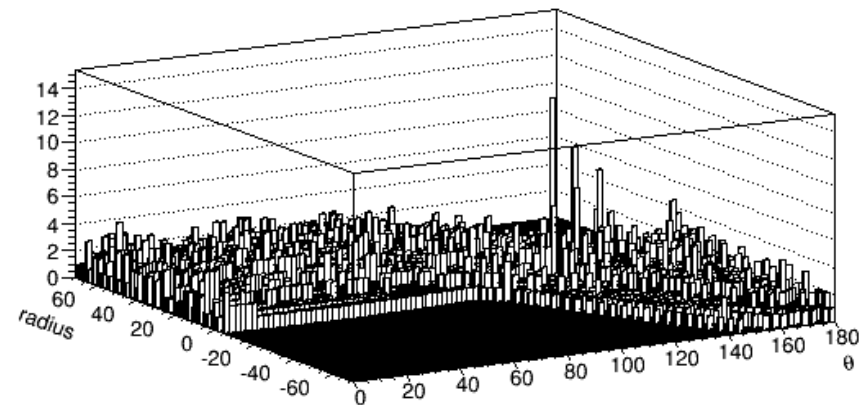
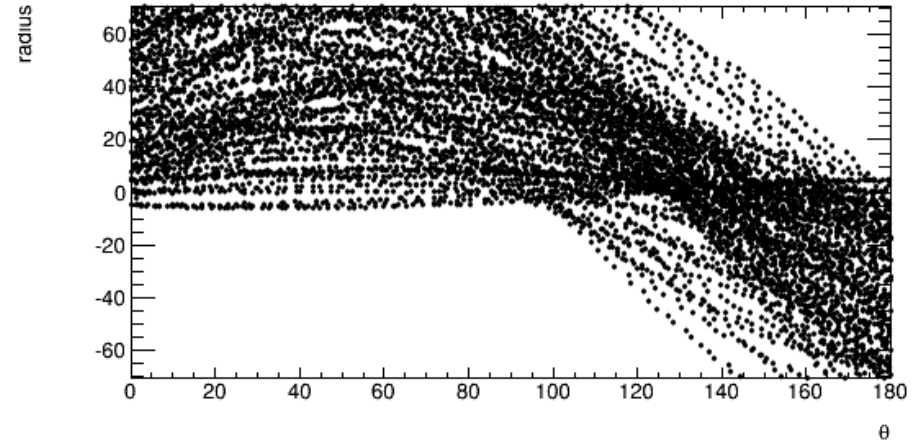
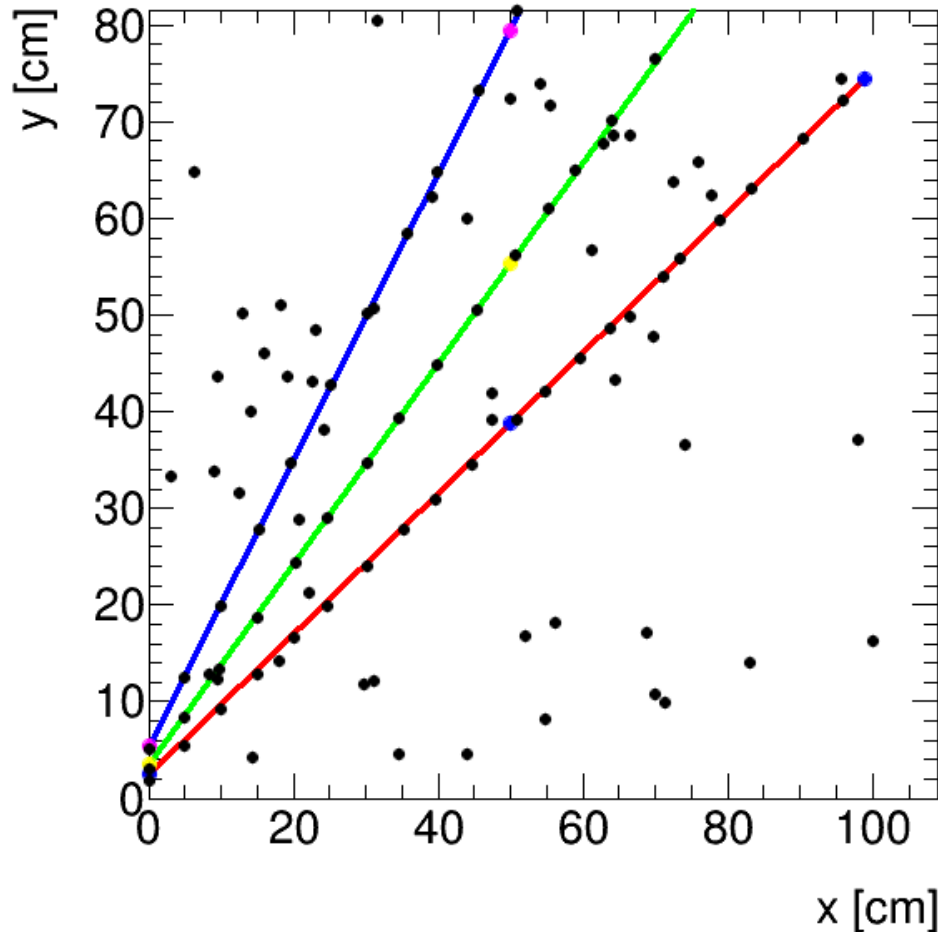

Output plots: GPU (112 hits)

sim. hit distribution | parameter space



Output plots: CPU (112 hits)

sim. hit distribution | parameter space



A current timing measurement (very preliminary...)

n_entries = 112	GPU	CPU
Filling parameter space	1.915 ms	1.582 ms
Searching for maxima	1.289 ms	0.522 ms

n_entries = 352	GPU	CPU
Filling parameter space	2.385 ms	4.079 ms
Searching for maxima	1.290 ms	0.538 ms