Homework 04

Scattering

David C. Banks

Electrical Engineering and Computer Science
University of Tennessee

2009

Textbook

Hand in at the beginning of class in 1 week
Chapter 14
Problem 1: Compute the average value of the function xyz in the unit cube $(x, y, z) \in [0, 1]^3$.
Problem 2: Compute the average value of r on the unit disk $(r, \phi) \in [0, 1] \times [0, 2\pi)$.

Programming components

Scene: source, target, sensor centered at target
Sensor prints particle position and velocity on entry, exit
Position relative to sensor's center
Read man page for atan2
Create a histogram of frequency of occurrence

```
int  freq[numAngles][numSpeeds][numAngles][numSpeeds];
```

Programming components

Read the output file from the sensor as input to the sensor
Discretize angle, energy (speed) into bins
Key-value pair
sensorDataIn filename
numAngles int **numSpeeds** int
speedMin float **speedMax** float
Note: assume angleMin=0, angleMax=2π
Increment count for the bin containing a sample

```
anglei0 = (numAngles)*(angle0-anglemin)/(anglemax-anglemin)
speedi0 = (numSpeeds)*(speed0-speedmin)/(speedmax-speedmin)
...
freq[anglei0][speedi0][anglei1][speedi1] ++;
```

Programming components

Compute max value of histogram

```
maxHist = 0;
for ( angle0 = 0; angle0 < numAngles; angle0++ )
  for ( speed0 = 0; speed0 < numSpeeds; speed0++ )
    for ( angle1 = 0; angle1 < numAngles; angle1++ )
      for ( speed1 = 0; speed1 < numSpeeds; speed1++ )
        {
        newMax = freq[angle0][speed0][angle1][speed1];
        if ( newMax > maxHist )
          maxHist = newMax;
        }
```

Programming components

Convert histogram to probability
Divide by the total number of occurrences for $\sum P = 1$
Divide by the max number of occurrences for $P \leq 1$

```
prob = (float)freq[angle0][speed0][angle1][speed1]/maxHist;
```

Programming components

Generate random output angle and speed using rejection
Pick a random velocity v_1 from disk of radius speed_{max}
Compute direction θ_1, speed_1 from v_1
Pick a random number rnd $\in$ [0,1]
If rnd > $P[\theta i_0][\text{speed} i_0][\theta i_1][\text{speed} i_1]$
Try again with new angle, speed, rnd
Else, place particle at edge of sensor and release
particle.position.x = sensorCenter.x + sensorRadius*cos(θ_1)
particle.position.y = sensorCenter.y + sensorRadius*sin(θ_1)
particle.velocity = v_1

Webpage

Create scene files, put image of scene on webpage
Shoot at least 10,000 particles at targets in sensor
Use 5 values of gravity G: 2 negative, 2 positive, zero
Record time for each run (see /usr/bin/time)
Write sensor output to file
Read file as input, run scattering simulator
Record time for each run
Record animations of particles scattering from sensor
Report speedup in scattering model versus gravity simulation