CS451
Ray Tracing
BVH and sampling

Jyh-Ming Lien
Department of Computer Science
George Mason University

Most materials are from “Realistic Ray Tracing” by Shirley and Morley
Review

- What we learned last week
  - Ray tracing framework
  - Reflect
  - Refraction
  - Soft shadow, reflection, refraction
- Remember that a pixel defines an area, not a point, in an image
Today

- More on ray tracing
  - Bounding volume hierarchy
  - Sampling strategy
  - Monte Carlo ray tracing

- Why do we need so many different versions of ray tracers?
Sampling

- Uniform distribution is good, BUT
- Correlation between samples is bad (jaggy rendering)
  - So, using points on a grid is bad (called Regular Sampling)
- Random Sampling (with uniform random number generator)

\[
\begin{align*}
\text{for } i &= 0 \text{ to } N - 1 \\
    x_i &= \text{random}(0, 1) \\
    y_i &= \text{random}(0, 1)
\end{align*}
\]
Simple Random Sample

regular samples

random samples
Good Samples

- A good set of samples has the following properties
  - uniformly distributed over the unit square without regular spacing
  - Some minimum distance is maintained between samples points
  - 1D x and y projections of unit square are also uniformly distributed

- In general, a very large sample size satisfy all these properties
- BUT, it will take you a lot of time to render
Low-Discrepancy Sampling

- Given a sample $P$ of size $k$ create from a measure space $X$
- The discrepancy of $P$ w.r.t a range space $R$ is

$$D(P) = \max_R \left| \frac{\|P \cap R\|}{k} - \frac{\mu(R)}{\mu(X)} \right|$$
Jittered Sampling

- Jittered (stratified) sampling
- Prevent clumping of samples

```
for i = 0 to n_x - 1
  for j = 0 to n_y - 1
    k = i * n_x + j
    x_k = randfrom(i/n_x, (i + 1)/n_x)
    y_k = randfrom(j/n_y, (j + 1)/n_y)
```
Problem of Jittered Sampling

- Jittering suffers when these samples are projected onto the $X$- or $Y$-axis.

Projections on the $X$- or $Y$-axis are not uniformly distributed.
$n$-rook Sampling

1. Place $n$ rooks on a $n$ by $n$ chessboard
2. Shuffling the $y$ (or $x$) coordinates of the rooks
**n-rook Sampling**

- Problem: 1D projections is fixed, but at the cost of ruining the 2D distribution!
Multi-Jittered Sampling

- Multijittered sampling
  - Combine jittering and rook sampling
  - Create a canonical arrangement

```c
for (int j = 0; j < n; ++j) {
    for (int i = 0; i < m; ++i) {
        p[j * m + i].x = (i + (j + drand48()) / n) / m;
        p[j * m + i].y = (j + (i + drand48()) / m) / n;
    }
}
```
Multi-Jittered Sampling

- Multijittered sampling
  - Create a canonical arrangement
  - Shuffle the X coordinates
  - Shuffle the Y coordinates

```cpp
for (int j = 0; j < n; ++j) {
    for (int i = 0; i < m; ++i) {
        int k = j + drand48() * (n - j);
        std::swap(p[j * m + i].x,
                   p[k * m + i].x);
    }
}

for (int i = 0; i < m; ++i) {
    for (int j = 0; j < n; ++j) {
        int k = i + drand48() * (m - i);
        std::swap(p[j * m + i].y,
                   p[j * m + k].y);
    }
}
```
Poisson-Sampling

\begin{align*}
i &= 0 \\
&\text{while } i < N \\
&\quad x_i = \text{randfrom}(0, 1) \\
&\quad y_i = \text{randfrom}(0, 1) \\
&\quad \text{reject} = \text{false} \\
&\quad \text{for } j = 0 \text{ to } i - 1 \\
&\quad \quad \text{if } [(x_i - x_j)^2 + (y_i - y_j)^2] < d^2 \\
&\quad \quad \quad \text{reject} = \text{true} \\
&\quad \quad \text{break} \\
&\quad \text{if not reject} \\
&\quad i = i + 1
\end{align*}
Centroidal Voronoi Diagram (CVD)

- Definition: Voronoi diagram
- Definition: centroidal Voronoi diagram
- CVD provides sampling with low discrepancy
Centroidal Voronoi Diagram (CVD)

- Definition: Voronoi diagram
- Definition: Centroidal Voronoi diagram
- CVD provides sampling with low discrepancy

CVD is very useful in computer graphics, for example in generating stippling drawing
Blue Noise Sampling

- The term “blue-noise” refers to any noise with minimal low-frequency components and no concentrated spikes in energy.
- Blue-noise sampling generates randomized uniform distributions.
- https://www.youtube.com/watch?v=mJOvenBySGs
Sampling and Filtering

- Sample from this 2D function

\[ L(x, y) = \frac{1}{2} (1 + \sin \frac{x^2 + y^2}{100}) \]

- [http://www.cc.utah.edu/~ndn1/pete-rt/ray1_old.html](http://www.cc.utah.edu/~ndn1/pete-rt/ray1_old.html)