Realistic Image Synthesis

- Monte Carlo Sampling -

Roman Ďurikovič

Overview

- **Today**
  - Blind Monte Carlo Integration
  - Intelligent Monte Carlo Integration
  - Discrepancy and Basic Quasi Monte-Carlo Sampling
  - Direct Lighting Computation

Blind Monte Carlo Integration

- **Blind Methods**
  - No information about integrand
- Goal:
  - Fast numerical integration
  - Low variance
  - At low sampling rates
  - Maximize Efficiency = 1/ (Variance * Cost)

- **Algorithms**
  - Crude Monte Carlo Sampling
  - Rejection Sampling
  - Sequential Tests
  - Blind Stratified Sampling (Jittering)
  - Weighted Monte Carlo Sampling
  - Quasi Monte Carlo Sampling

Blind Monte Carlo Integration

- **Crude Monte Carlo Integration**
  - Computing the area under f(x)

$$\theta = \int_0^1 f(x)dx$$

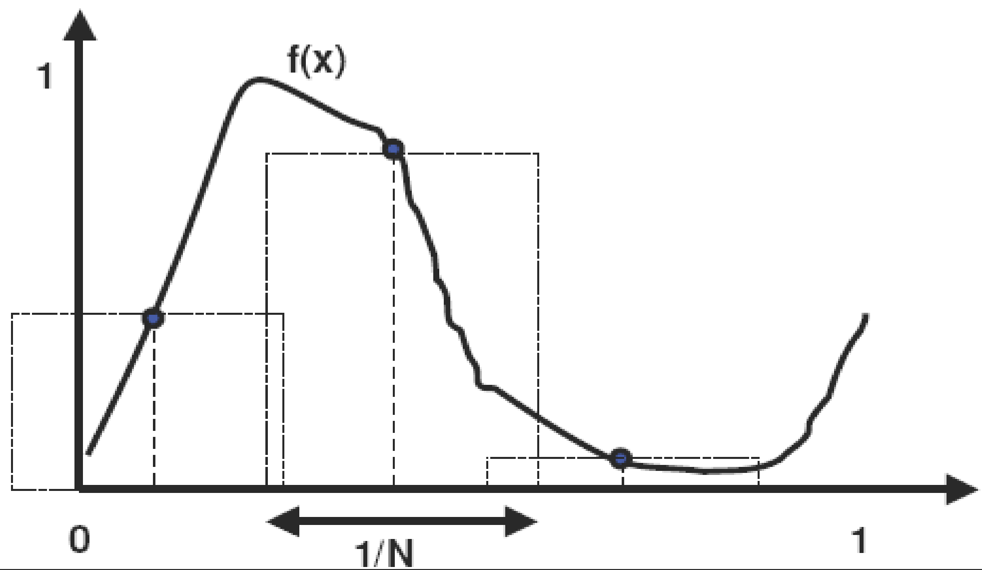
  - Law of large numbers provides for independent and uniformly distributed random variables ξ_i in [0..1]

$$\theta \approx \bar{f} = \frac{1}{N}\sum_{i=1}^{N} f(\xi_i)$$

  - Standard deviation

$$\sigma_{\bar{f}} = \frac{\sigma_f}{\sqrt{N}}$$

„diminishing return“



Blind Monte Carlo Integration

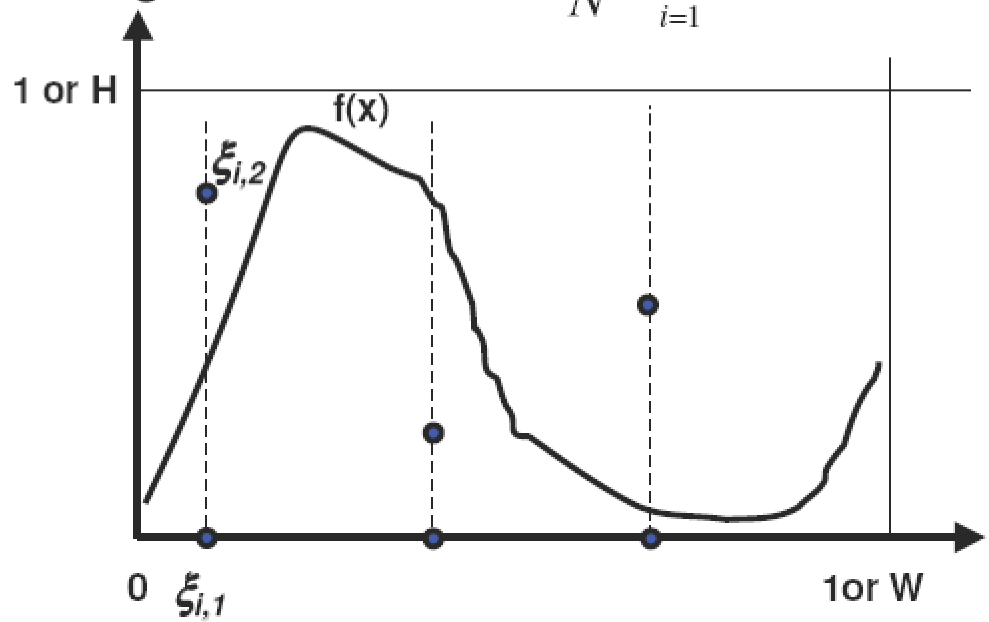
- **Rejection Sampling**
  - Define

$$g(x,y) = \begin{cases} 0 & y > f(x) \\ 1 & y \le f(x) \end{cases}$$

  - Numerical integration
    - Like crude Monte Carlo, with two random variables

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g(\xi_{i,1},\xi_{i,2}) \quad \text{or in general} \quad \bar{g} = \frac{HW}{N}\sum_{i=1}^{N} g(\xi_{i,1},\xi_{i,2})$$

  - Variance is always worse as for crude MC
  - Rule:
    Wherever possible use exact values instead of estimates



Blind Monte Carlo Integration

- **Sequential Sampling**
  - Central Limit Theorem

$$\lim_{N\to\infty} \Pr\left\{ F_N - E[Y] \le \frac{t\sigma}{N^{1/2}} \right\} = \frac{1}{2\pi} \int_{-\infty}^{t} e^{-x^2/2} dx$$

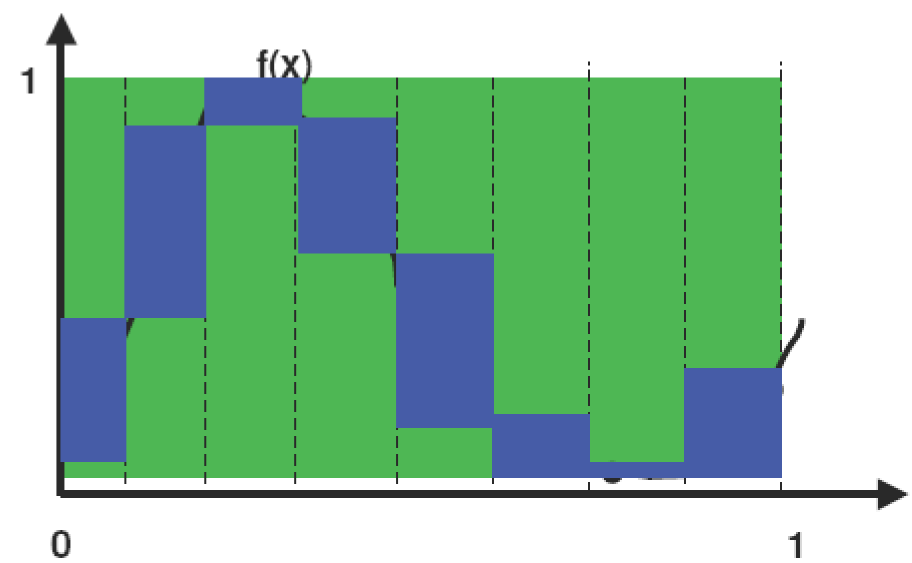
- **Approach**
  - Send rays until confidence in the estimate is high enough
    - Student t-distribution [Purgathofer, 1987]
    - Chi-squared distribution [Lee et al, 1985]
  - Problem:
    - Usually too conservative
    - Requires too many samples

Blind Monte Carlo Integration

- **Blind Stratified Sampling (Jittering)**
  - Goal: uniform distribution of samples
  - Subdivision of domain of the function into k *strata*

$$\overline{f} = \sum_{i=1}^{k} \sum_{j=1}^{N_i} \frac{a_i - a_{i-1}}{N_i} f(a_{i-1} + (a_i - a_{i-1})\xi_j)$$

  - With independent random variables
    - Variance of sum is sum of variances
  - In general lower variance of f over smaller intervals



Blind Monte Carlo Integration

- **Example for Stratification**
  - Shadow boundary

$$F_N = \frac{1}{N}\sum_{i=1}^{M} N_i F_i \quad \text{with } N = \sum N_i$$

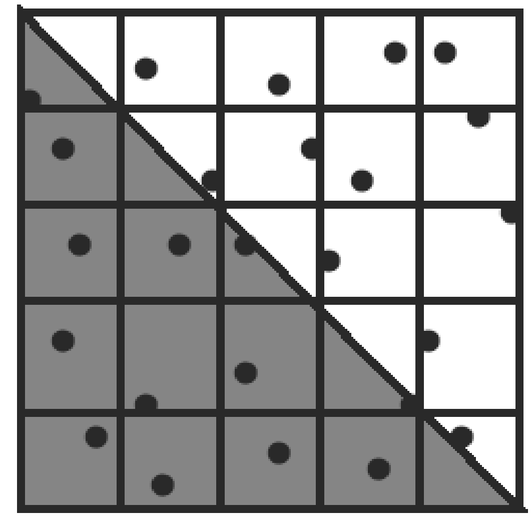
  - N strata
  - One sample per stratum (N_i=1)

$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{N} V[F_i] =$$

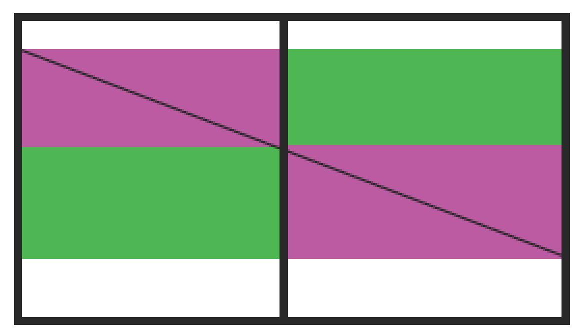
$$= \frac{1}{N^2}\sum_{m=1}^{\sqrt{N}} V[F_i]$$

$$= \frac{V[F_i]}{N^{1.5}}$$

  - Quadratic improvement of efficiency with number of strata for smooth functions



$$V = \sum_{i=1}^{N} (X - E(X))^2$$

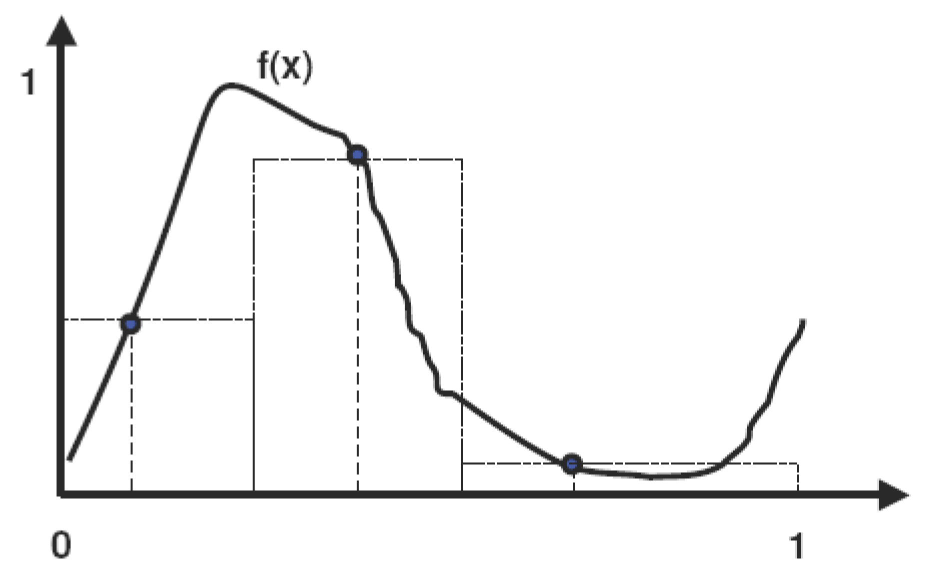


Blind Monte Carlo Integration

- **Weighted Monte Carlo Sampling [Yakowitz`78]**
  - Goal: Accurate coverage of range of f(x)
  - Weighting with area of Voronoi region of each sample

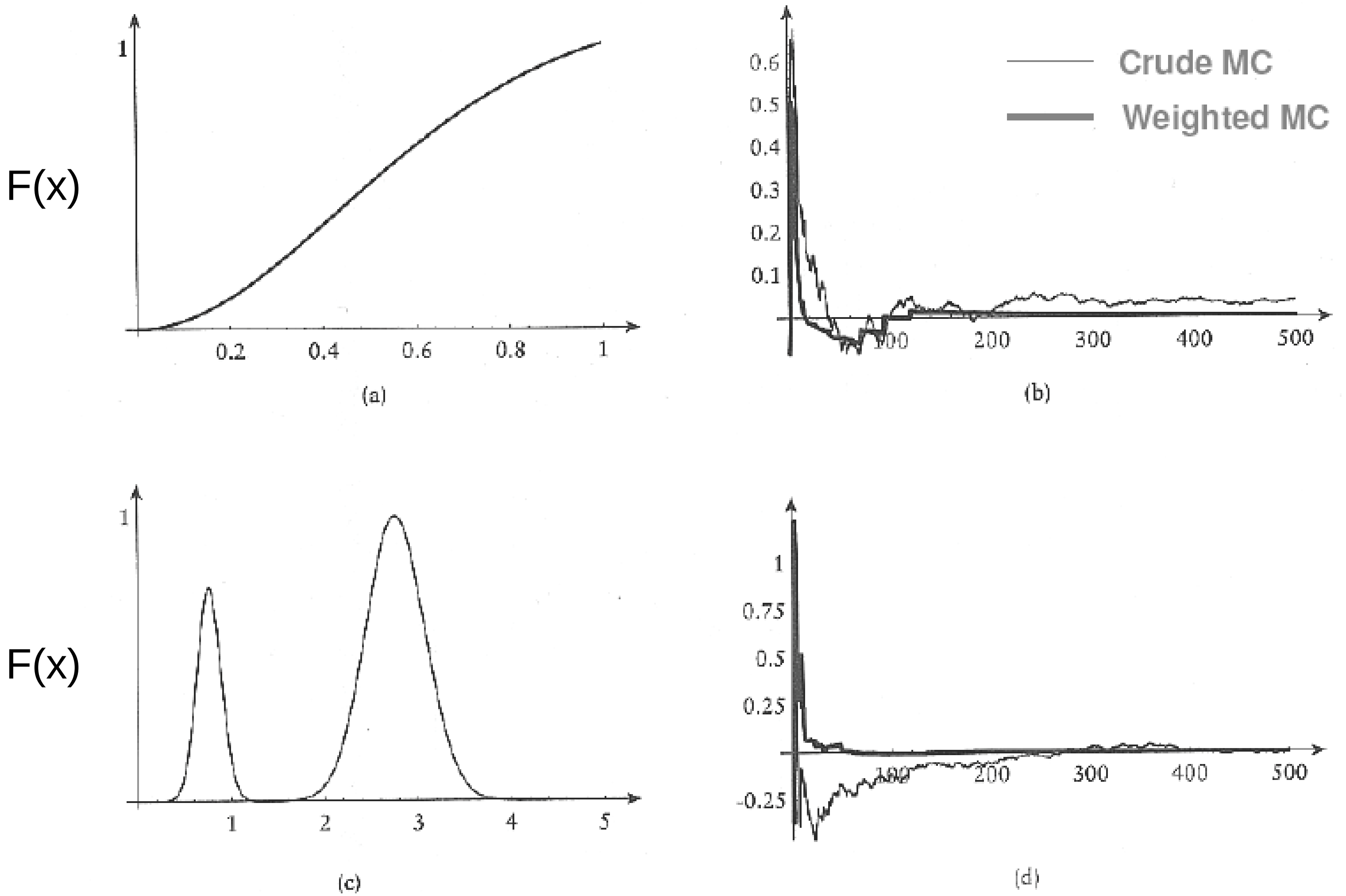
$$\overline{f} = \sum_{i=1}^{N} w_i f(\xi_i) / \sum_{i=1}^{N} w_i$$

  - Voronoi region of a point p_i:
    - All points p that are closer to p_i than any other point
  - Better convergence at low dimensions
    - $O(1/N^{2/d})$
    - if f has continuous second derivative



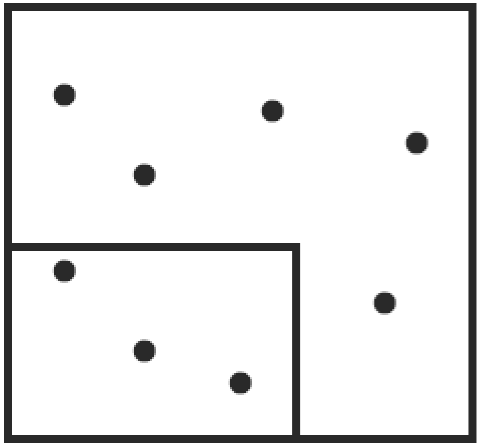
Blind Monte Carlo Integration

- **Two Examples for weighted Monte Carlo**



Blind Monte Carlo Integration

- **Quasi-Monte-Carlo**



  - Goal:
    - Uniform coverage of range
    - Low „Discrepancy“ (D*)

$$D^*(P_n) = \underbrace{\sup_{J=\prod_{j=1}^{s}[0,a_j)\subset I^S}}_{\text{For all rectangle anchored at origin}} \Big| \underbrace{\int_{I^S} \chi_J(x)dx}_{\text{Area}} - \underbrace{\frac{1}{N}\sum_{i=0}^{N}\chi_J(x_i)}_{\text{Estimate of Area}} \Big|$$

    - s: Number of dimensions, I: unit interval,
      χ: characteristic function, P_N: sequence of points

- **Identical algorithms except for random number generator**

Blind Monte Carlo Integration

- **Halton-Sequence (N-dimensional)**

  $$\xi_m = \left(\phi_2(m), \phi_3(m), \phi_5(m), \ldots, \phi_{p_N}(m)\right)$$

  – $\phi_r(m)$: radical inverse
    - Reflect bit pattern of m in basis r at decimal point
    - $\phi_2(26_{10}) = \phi_2(11010_2) = 0.01011_2 = 11/32$
    - $\phi_3(19_{10}) = \phi_3(201_3) = 0.102_3 = 11/27$

  – p_N: N-th prime number

  – σ: O(1/N) for smooth functions

  – Uniform distribution:
    - More significant bit vary faster
    - Visit all intervals of 2^{-k} before intervals of $s^{-(k+1)}$

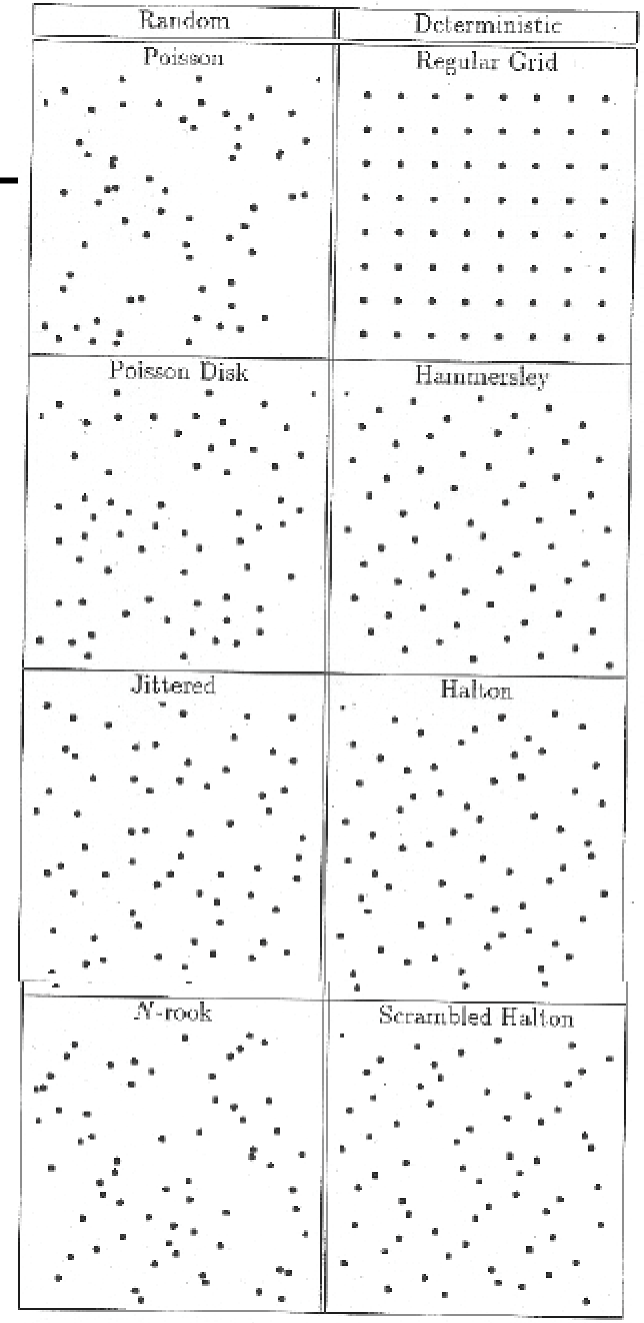
- **Hammersley-Sequence (N-dimensional)**

  $$\xi_m = \left(m / N, \phi_2(m), \phi_3(m), \phi_5(m), \ldots, \phi_{p_N}(m)\right)$$

- **Zaremba-Sequenz, …**

Sample-Distributions

- **Visual evaluation of discrepancy across several random distributions**



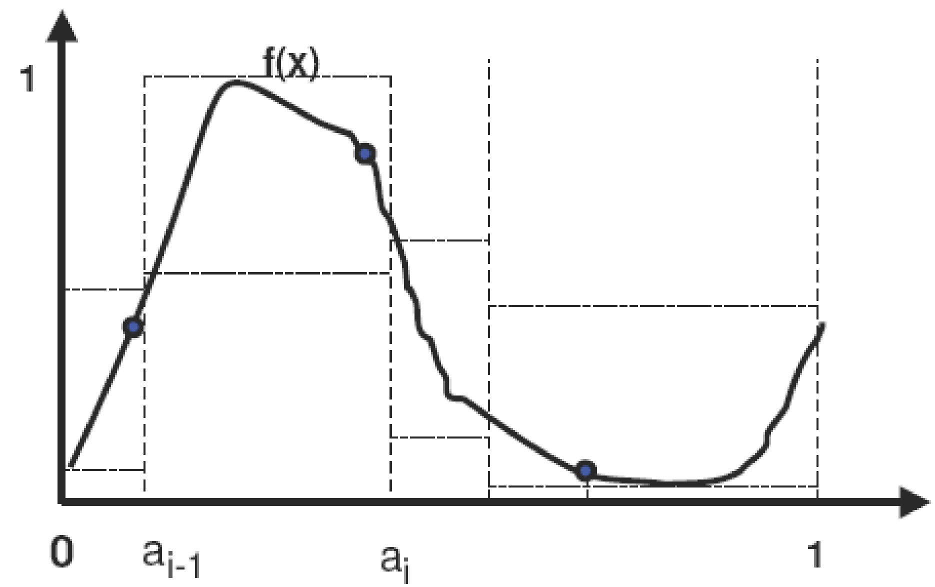
Intelligent Monte Carlo Integration

- **Goal**
  - Exploit knowledge about integrand
  - Intelligent placement of samples
- **Algorithm**
  - Intelligent stratified sampling
  - Importance sampling
  - Weighted importance sampling
  - Separation of the main part (Control Variates)

Intelligent Monte Carlo Integration

- **Intelligent Stratified Sampling**
  - Goal: Low variance with few samples
  - Suitable placement of strata
- **Approach**
  - Choose strata such that variance is equally distributed
  - Choose number of samples N_i

$$N_i \propto (a_i - a_{i-1})\,\mathrm{var}_i(f)$$



Intelligent Monte Carlo Integration

- **Importance Sampling**
  - Goal: Distribute samples such as to minimize variance
- **Approach**

$$\bar{f} = \int f(x)dx = \int \frac{f(x)p(x)}{p(x)}dx = \int \left[\frac{f(x)}{p(x)}\right]\underbrace{p(x)dx}_{\text{sampling}}$$

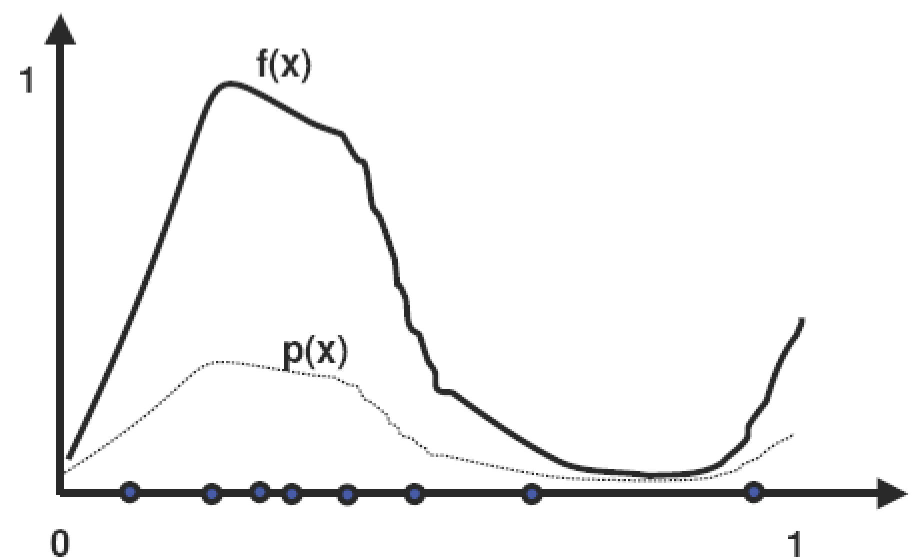
  - Choose p(x) such that
    - p is a probability density
    - $f(x)/p(x) < \infty$
  - Ideally: $p(x) \propto |f(x)|$ but

$$p(x) = Cf(x)$$

$$1 = \int p(x)dx = C\int f(x)dx$$

$$\Rightarrow C = 1/\int f(x)dx$$

  - But any function p(x) that has a shape similar to f(x) helps



Intelligent Monte Carlo Integration

- **Combining multiple importance distributions**
  - Idea: One function p(x) is too inflexible
  - Use multiple functions in parallel
- **Approach with two estimators and weights w_i ($\Sigma w_i = 1$)**

$$V[w_1 S_1 + w_2 S_2] = w_1^2 V[S_1] + w_2^2 V[S_2] + 2 w_1 w_2 Cov[S_1, S_2]$$

$$Cov[S_1, S_2] = E[S_1 \cdot S_2] - E[S_1]E[S_2] \quad \text{(zero if independent)}$$

$$\Rightarrow \frac{w_1}{w_2} = \frac{V[S_2] + Cov[S_1, S_2]}{V[S_1] + Cov[S_1, S_2]}$$

  - Similar results for multiple estimators
- **A-priori weighted integration**
  - Weight two estimators
  - Weights are determined analytically or are estimated (manually)

$$I = \sum_{m=1}^{M} \frac{w_m}{N_m} \sum_{i=1}^{N_m} \frac{f(\xi_i)}{p_m(\xi_i)}$$

Intelligent Monte Carlo Integration

- **A-posteriori multiple importance sampling**
  - Choose samples
  - Assign weights according to probabilities of each estimator

$$I = \frac{1}{N} \sum_{m=1}^{N} \sum_{i=1}^{M} w_m \frac{f(\xi_i)}{p_m(\xi_i)} \text{ with } \sum_{\mathrm{m=1}}^{\mathrm{M}} w_m = 1$$

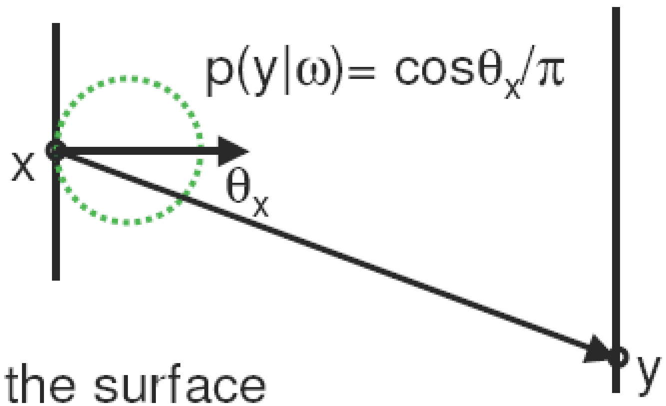
- **Balance Heuristics**

$$w_i(x) = \frac{p_i(x)}{\sum p_j(x)}$$

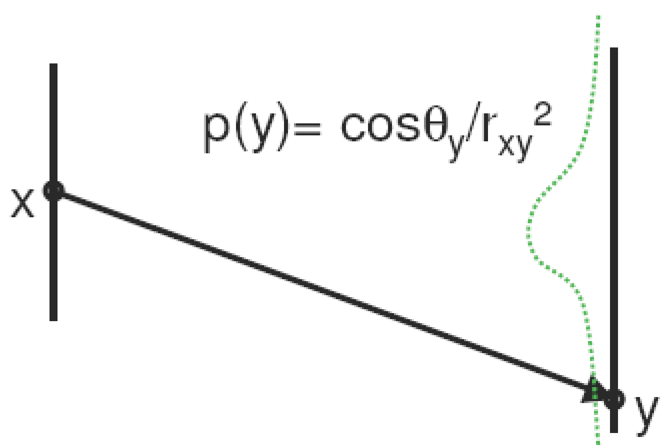
  - No other combination can be much better [Veach 97]
  - Motivation
    - Samples with low probability boost the variance with $1/(p_i)$
    - Assign larger weights to samples with higher probability
  - Must be able to evaluate probability of sample according to other estimate

Intelligent Monte Carlo Integration

- **Example: Different Probabilities**
  - Sampling directions



  - Sampling the surface



Intelligent Monte Carlo Integration

- **Other weighting heuristics**
  - Variance is additive – may have impact on already good estimators
  - Try to sharpen the weighting
- **Cutoff and Power Heuristic**

$$w_i = \begin{cases} 0 & \text{if } p_i < \alpha p_{\max} \\ \dfrac{p_i}{\sum_k \{p_k \mid p_k \geq \alpha p_{\max}\}} & \text{otherwise} \end{cases}$$

$$w_i = \frac{p_i^\beta}{\sum_k p_k^\beta}$$

  - Reduced weight for samples with low probability
- **Maximum Heuristic**

$$w_i = \begin{cases} 1 & \text{if } p_i \text{ is maximum} \\ 0 & \text{otherwise} \end{cases}$$

  - Adaptively partitions the integration domain according to $p_i(x)$
  - But too much samples are thrown away

Intelligent Monte Carlo Integration

- **Separation of the main part (Control Variates)**
  - Goal: Low variance through approximation with analytically solvable function
- **Approach**

$$\bar{f} = \int f(x)dx = \underbrace{\int g(x)dx}_{\text{analytically solvable}} + \underbrace{\int \left[f(x) - g(x)\right]dx}_{\text{lower variance than f(x)}}$$

  - g(x) should be a good approximation to f(x)

Direct Lighting Computation

- **Need to compute the integral**

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\underline{y}\in S} \underbrace{f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L(y,-\underline{\omega}_i)}_{\text{usually low variance if mostly diffuse}} \underbrace{V(\underline{x},\underline{y})}_{\substack{\text{unknown}\\ \text{variance}}} \underbrace{\frac{\cos\theta_x \cos\theta_y}{\|\underline{x}-\underline{y}\|^2}}_{\text{high variance}} dA_y$$

- **See also**
  - Shirley et al.: MC-Techniques for Direct Lighting Calculations
- **Single light source, not too close (>1/5 of its radius)**
  - Small:
    - $1/r^2$ has low variance
    - $\cos\theta_x$ has low variance
  - Planar:
    - $\cos\theta_y$ has low variance too
  - Choose samples uniformly on light source geometry
    - Sampling directions has very high variance unless we have huge lights

Direct Lighting Computation

- **Importance sampling of many light sources**
  - Cost grows with number of lights
- **Approaches**
  - Equal probability (1/NL)
  - Fixed weights according to total power of light

$$p_i = \frac{\Phi_i}{\sum \Phi_i}$$

    - Sample as discrete probability density function
  - Fixed spatial subdivision
    - Estimate the contribution in each cell (e.g. octree)
  - Dynamic and adaptive importance sampling
    - Compute a running average of irradiance at nearby points
    - Use the relative contribution as the importance function
    - Should use coherent sampling
    - Might need to estimate separately for primary and secondary

Direct Lighting Computation

- **Sampling thousands of lights interactively**
  - At each pixel send random path into the scene and towards light
    - Low overhead since we already trace many rays per pixel
  - Gives a rough estimate of light contribution to the entire image
    - Take maximum contribution of each light at any pixel
    - Might want to average over several images (less variance)
  - Use this estimate for importance sampling
    - Make sure every light is sampled eventually
    - Might ignore lights with very low probability (but has bias)
  - Trace samples ONLY from the eye
    - Avoids touching the entire scene
    - Minimizes working set for very large scenes