Blue Noise – An Introduction

Hao He

Table of Content

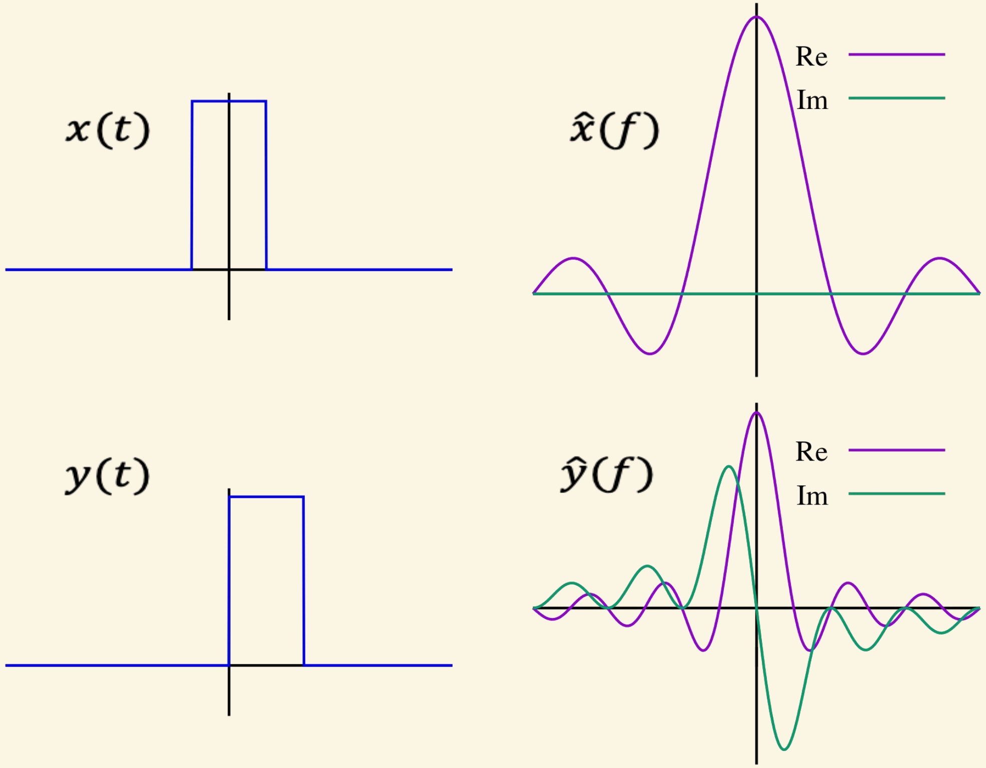
- Signal Processing Prerequisites
  - Fourier Transform
  - Power Spectrum of a Signal
  - 2D Discrete Fourier Transform
- Blue Noise Sampling
- Different Blue Noise Sampling Methods
  - Poisson Disk Sampling
  - Relaxation Based Sampling
  - Patch/Tile-Based Sampling
- Multiclass Blue Noise Sampling

Fourier Transform

- Decompose signal $x(t)$ into frequency components

$$\hat{x}(f) = \int_{-\infty}^{+\infty} e^{-2\pi i f t} x(t) dt$$

- Transform waveform into spectrum



Power Spectrum of a Signal

- The power spectrum $S_{xx}(f)$ of $x(t)$ describe the distribution of power into frequency components.
- For signal with finite energy

$$E = \int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |\hat{x}(f)|^2 df$$

$\hat{x}(f)$: Fourier transform of $x(t)$

- Then , we define Energy Spectral Density as

$$S_{xx}(f) = |\hat{x}(f)|^2$$

Power Spectral Density(PSD) of a Signal

- For signal with infinite energy, consider its average power

$$P = \lim_{T\to\infty} \frac{1}{T} \int_0^T |x(t)|^2 dt$$

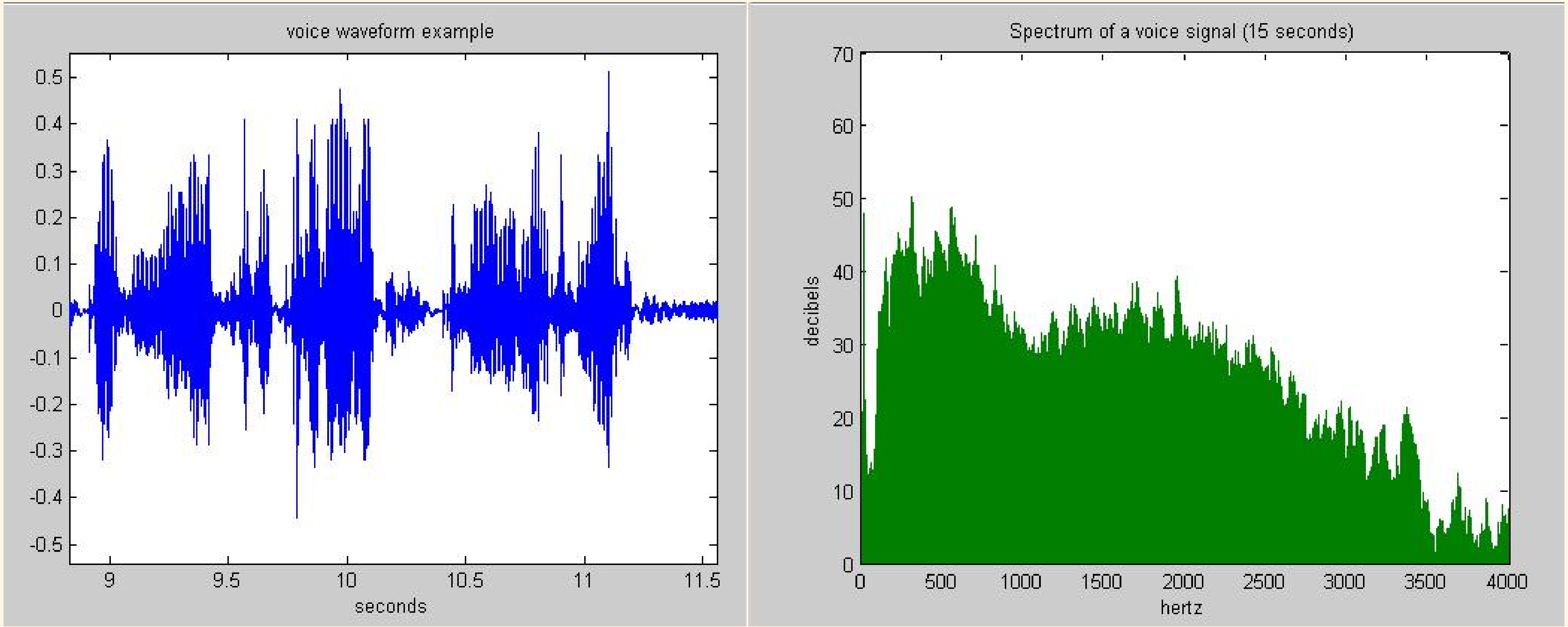
- Then we define the Power Spectral Density as

$$S_{xx}(f) = \lim_{T\to\infty} \boldsymbol{E}[|\hat{x}(f)|^2]$$

$$\hat{x}(f) = \frac{1}{\sqrt{T}} \int_0^T x(t) e^{-2\pi i f t} dt$$

$\hat{x}(f)$: truncated Fourier transform, $\boldsymbol{E}$: expected value.

Example of Power Spectrum



https://en.wikipedia.org/wiki/Spectral_density#/media/File:Voice_waveform_and_spectrum.png

2D Discrete Fourier Transform

- Signal can also be 2D, like an image
- Fourier transform can be applied to N*N 2D image(or other samples) to generate a 2D spectrum

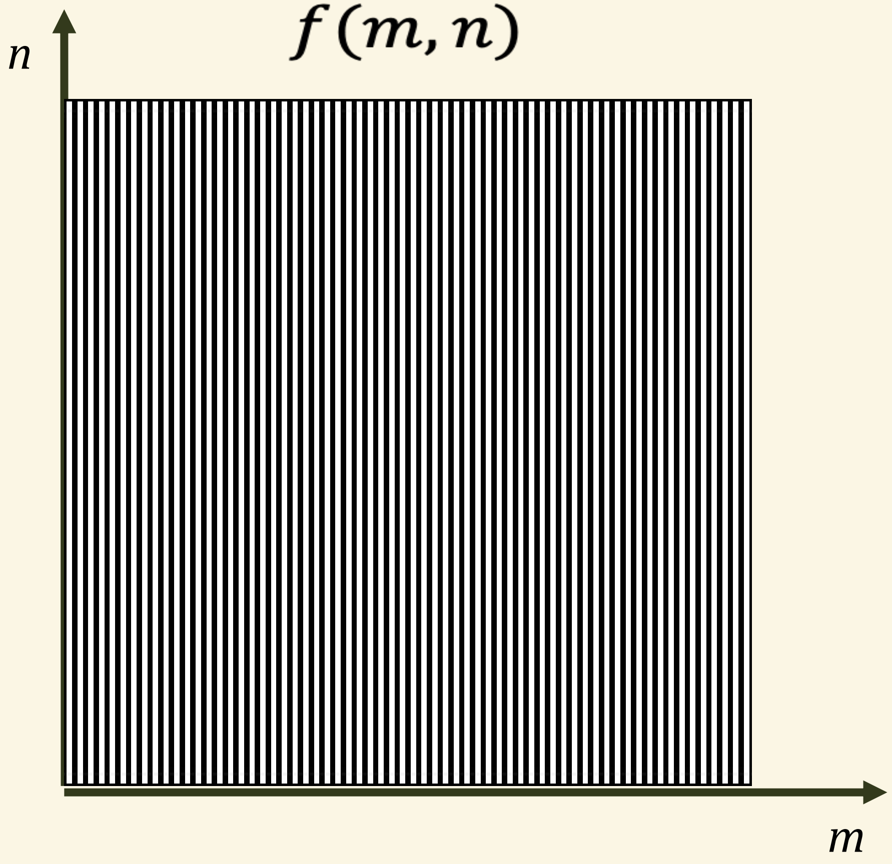
$$F(u,v) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) e^{-i\left(\frac{2\pi um}{N} + \frac{2\pi vn}{N}\right)}$$

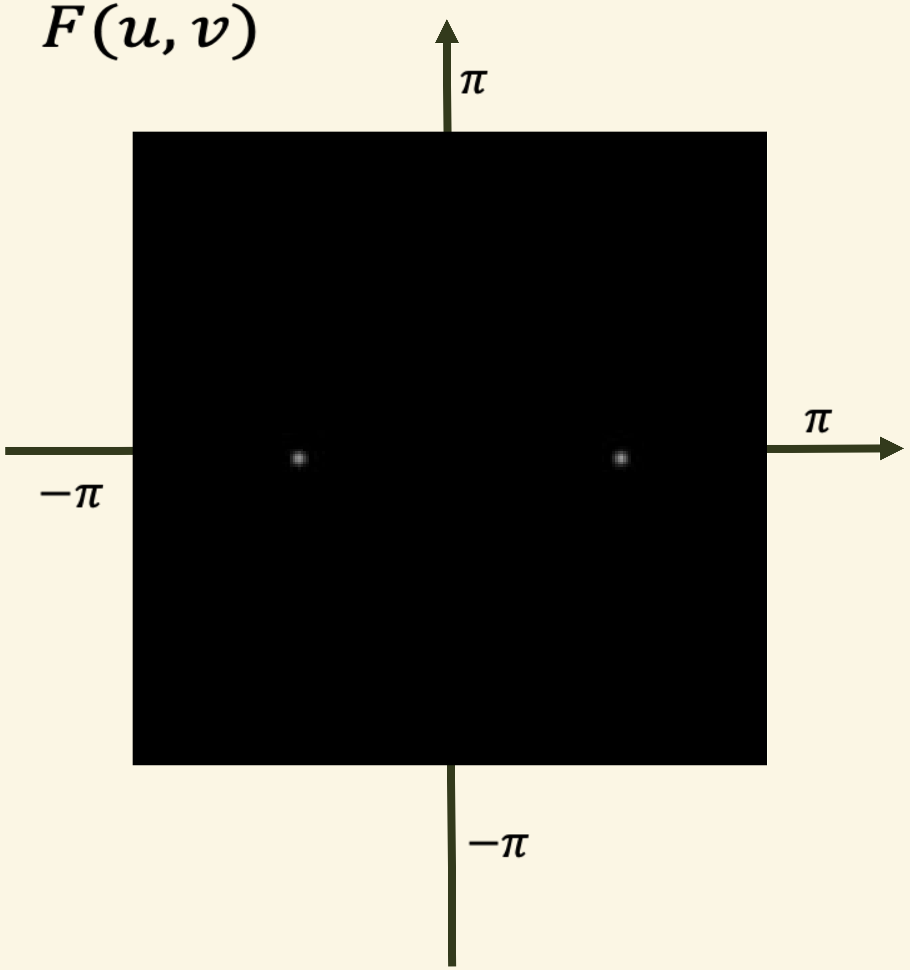
$f(m,n)$: sampled power value at (m,n)

$2\pi u/N$: vertical frequency

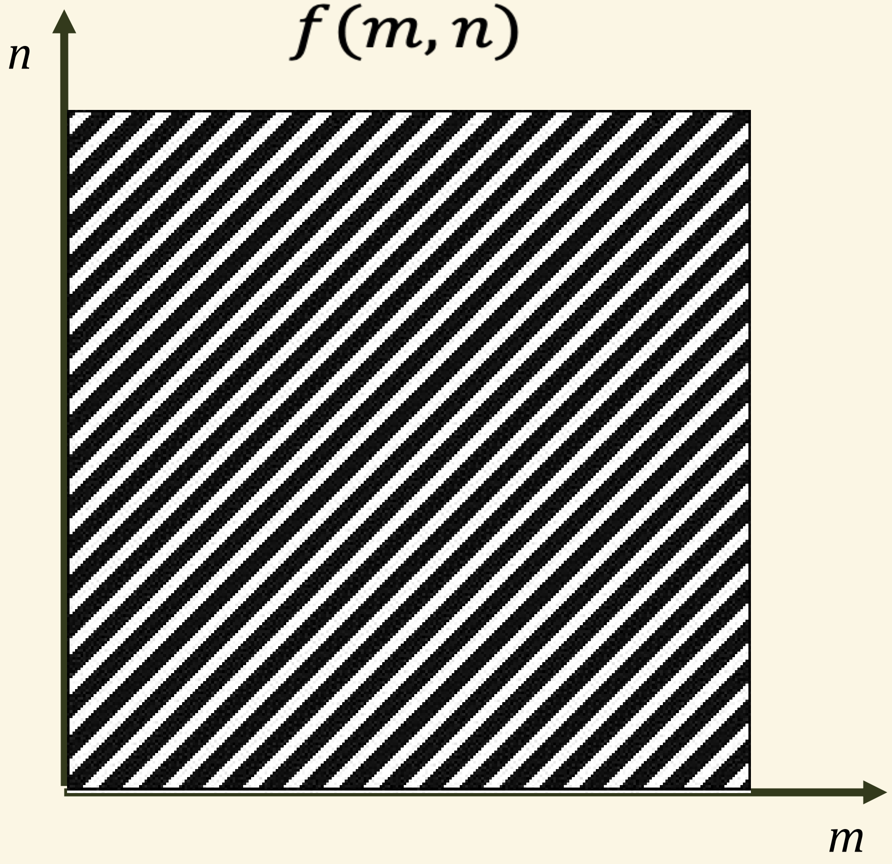
$2\pi v/M$: horizontal frequency

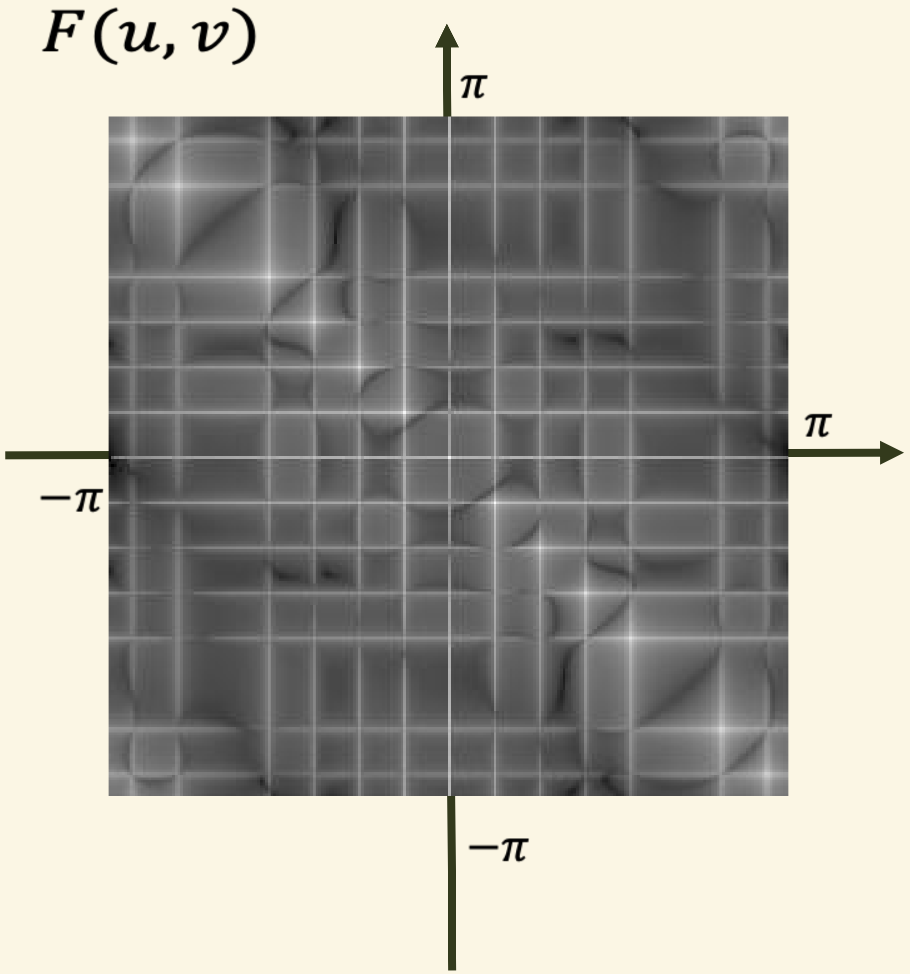
2D DFT Examples



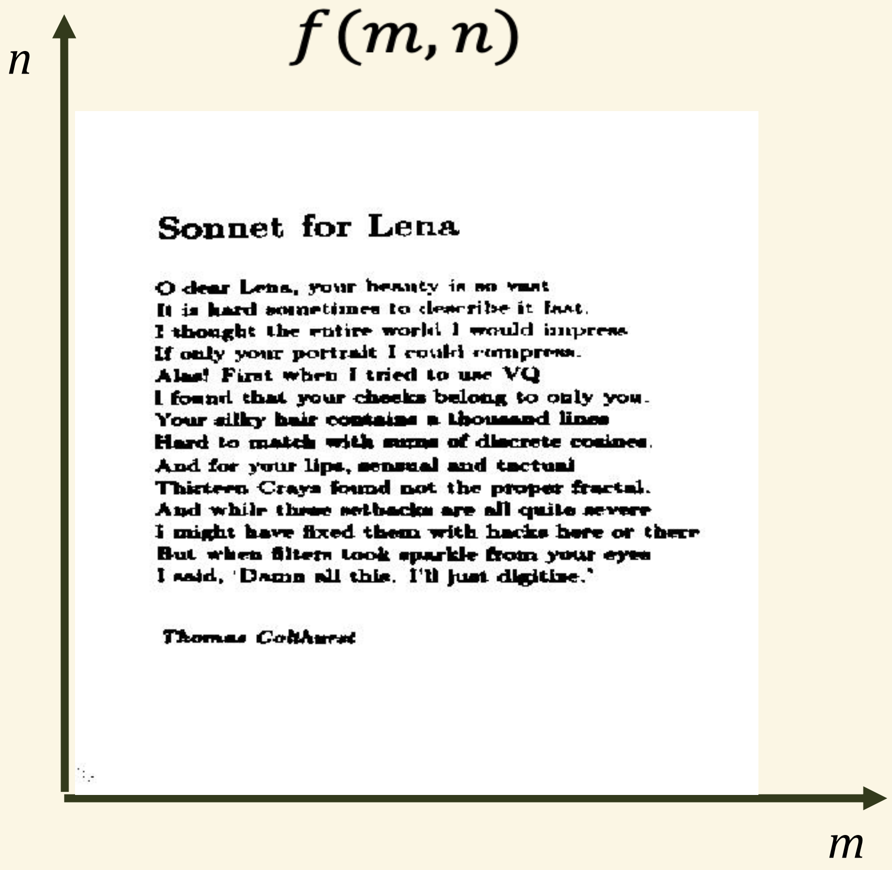


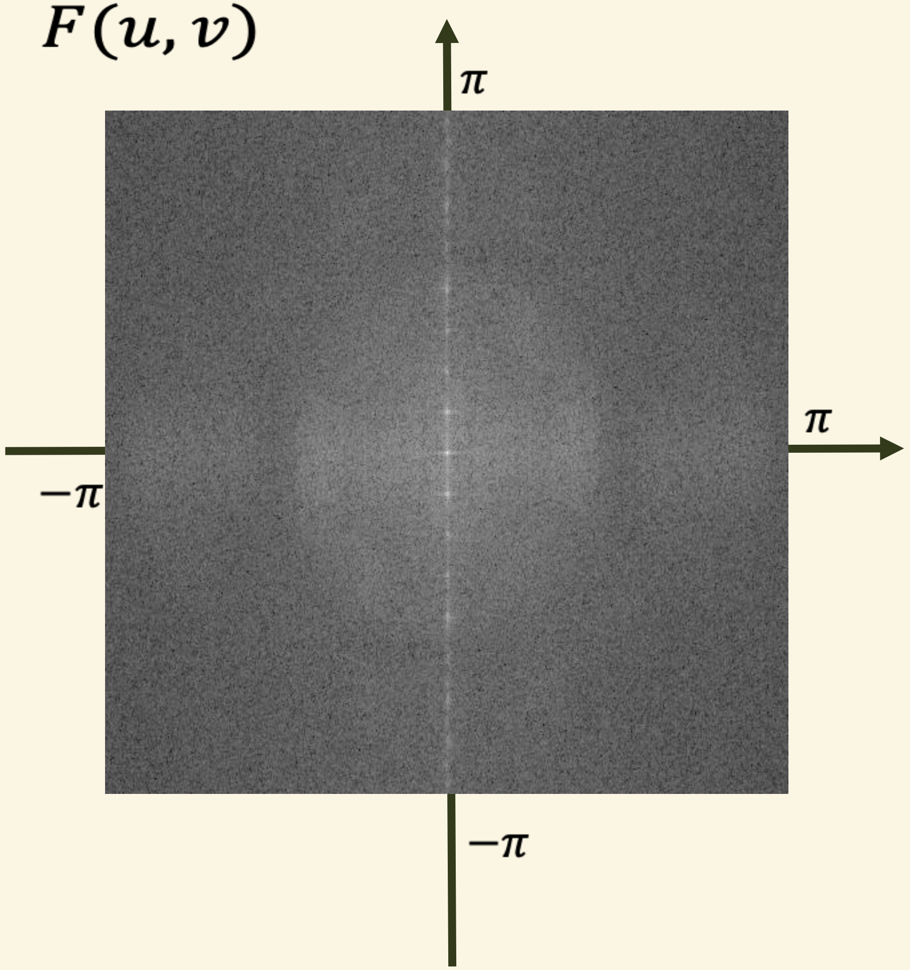
2D DFT Examples



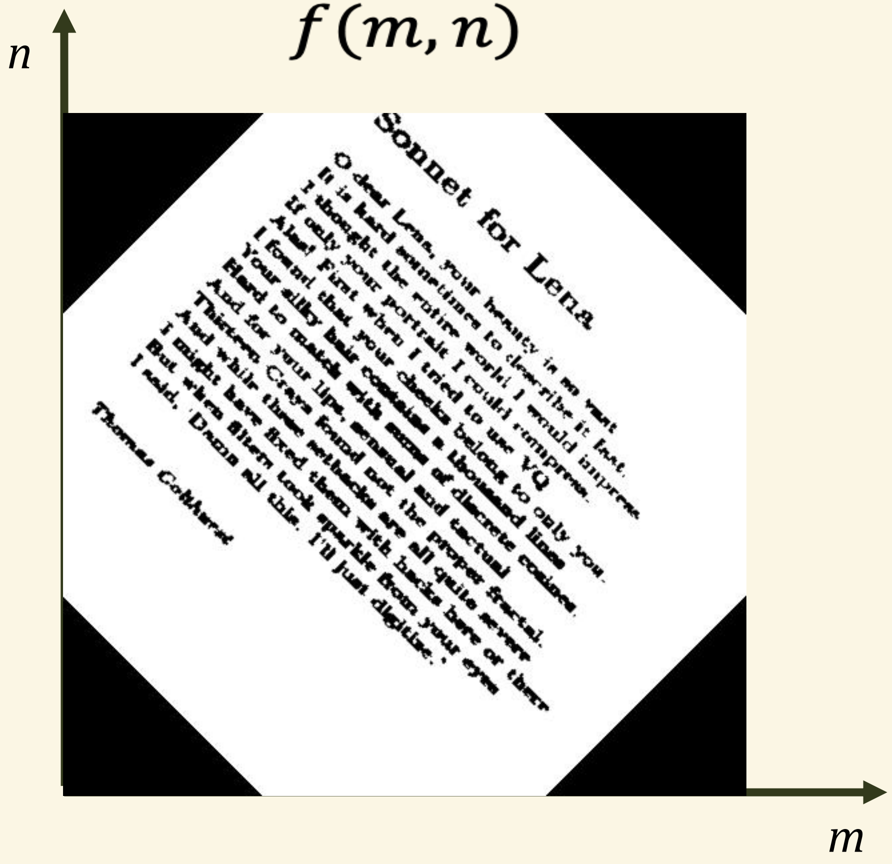


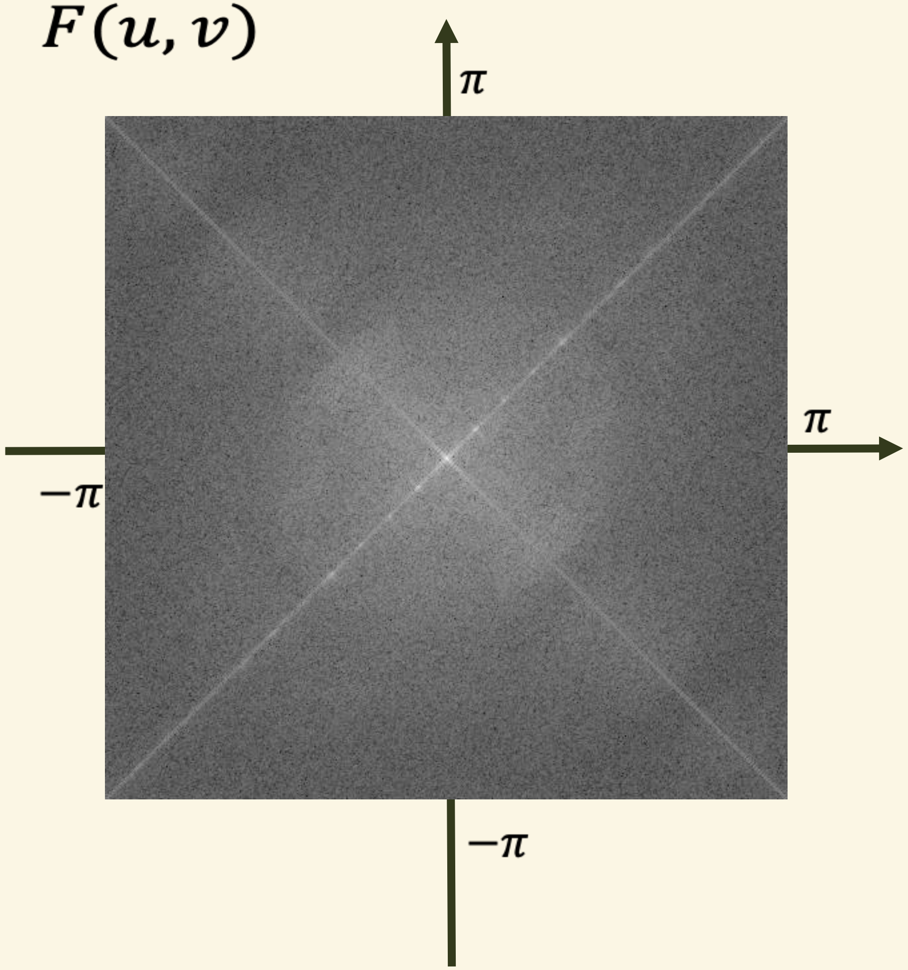
2D DFT Examples



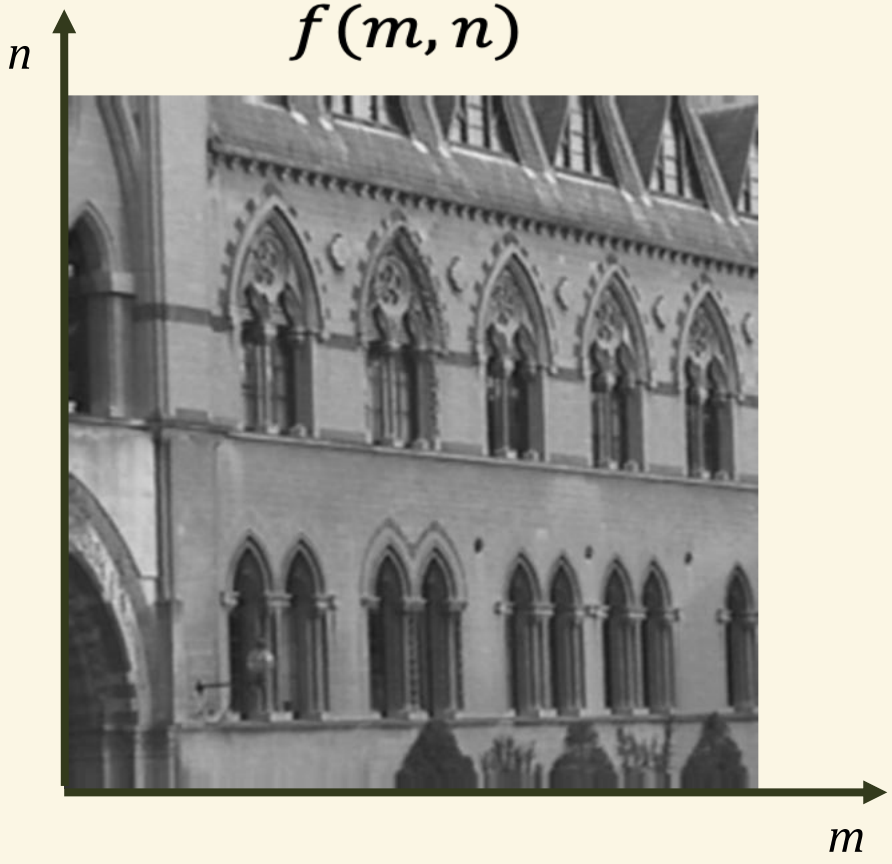


2D DFT Examples





2D DFT Examples



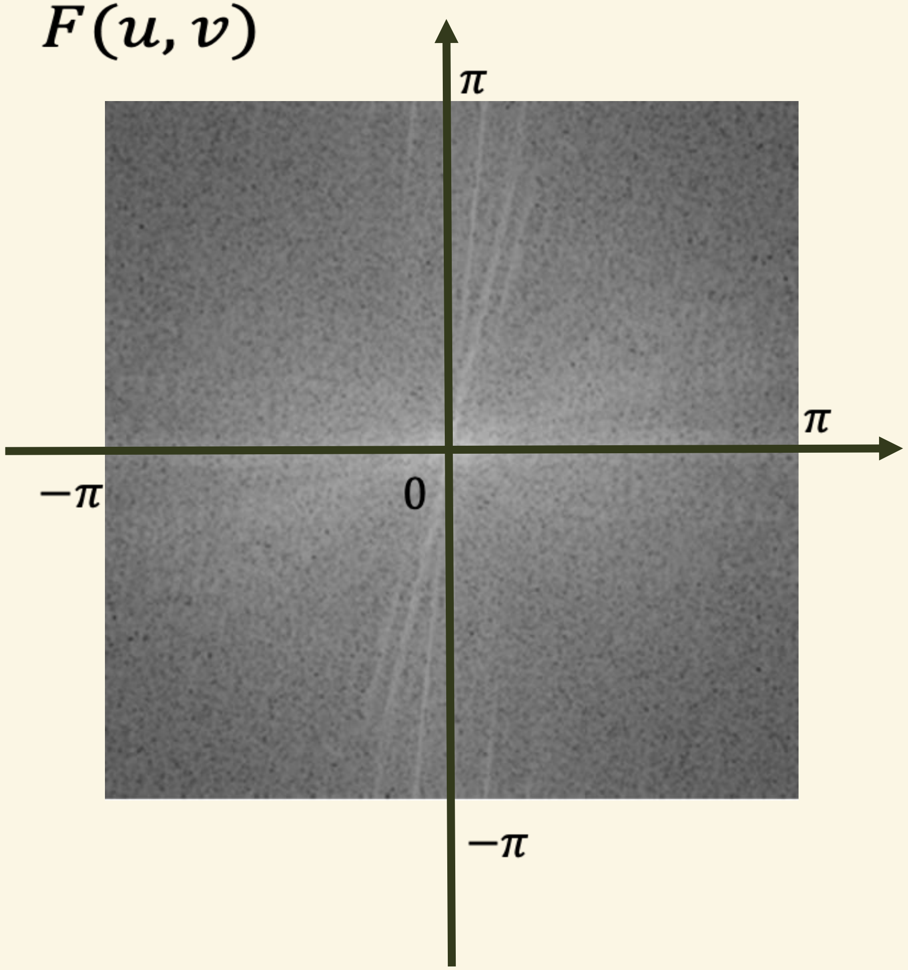


Table of Content

- Signal Processing Prerequisites
  - Fourier Transform
  - Power Spectrum of a Signal
  - 2D Discrete Fourier Transform
- Blue Noise Sampling
- Different Blue Noise Sampling Methods
  - Poisson Disk Sampling
  - Relaxation Based Sampling
  - Patch/Tile-Based Sampling
- Multiclass Blue Noise Sampling

What is Noise?

- Noise is a random signal.
- In discrete sense, it is a sequence of random variables (i.e. A random process)

$$X_1, X_2, \dots, X_n, \dots$$

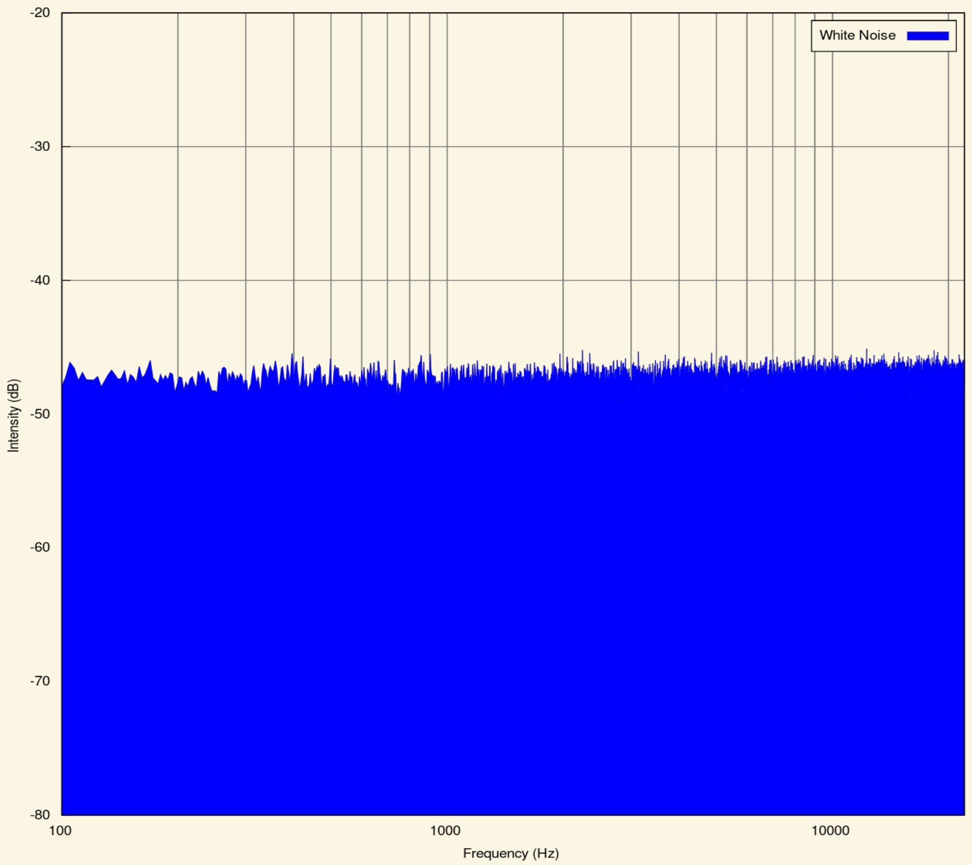
- Different types of noise have different characteristics
  - For example, a white noise satisfies

$$\begin{cases} EX_i = \mu \\ Cov(X_i, X_j) = \sigma^2 \text{ for any } i \neq j \end{cases}, \text{ where } \mu \text{ and } \sigma^2 \text{ are constants}$$

  - White noise have equal intensity at different frequencies

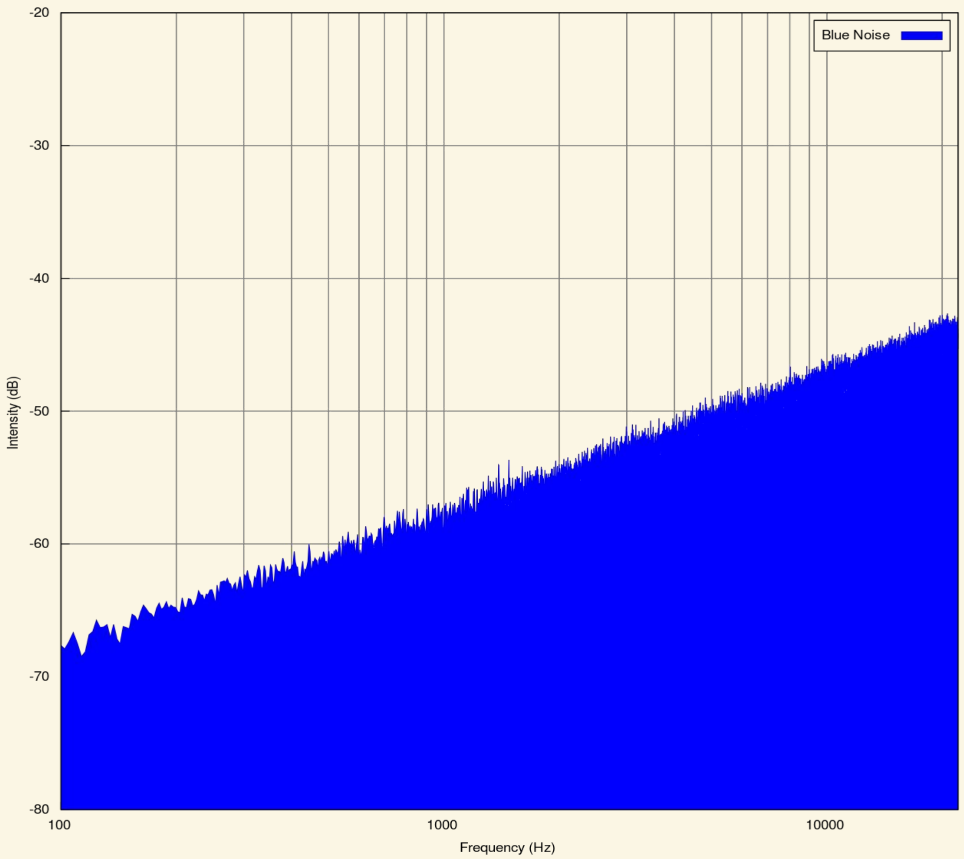
Power Spectrum of Noise

White Noise



Constant Power Spectral Density

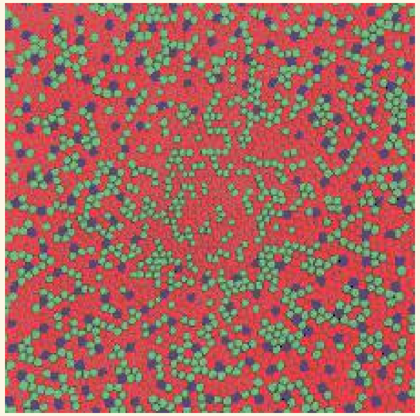
Blue Noise



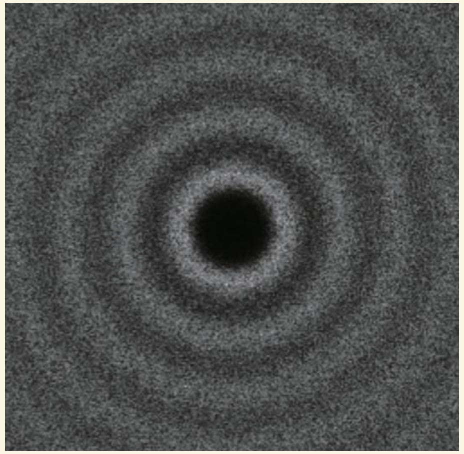
Low Frequency – Less Power
High Frequency – Higher Power

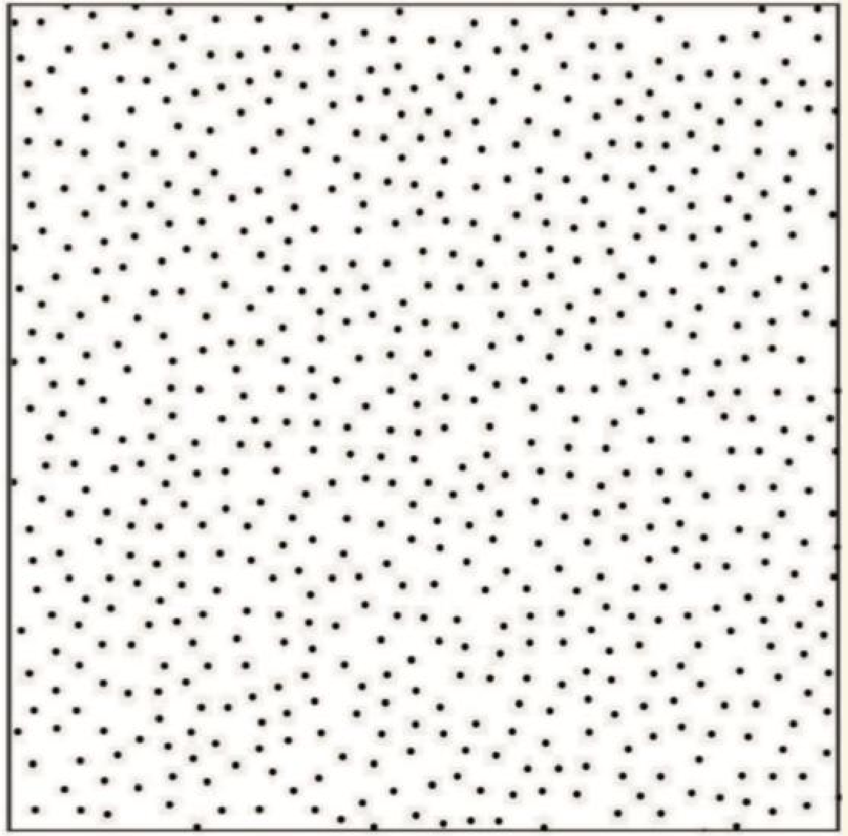
Blue Noise Sampling in Computer Graphics

- Inspired from the distribution of primate retina cells
- Refer to sampling methods that produce randomly located but spatially uniform sampling results



Human retina cells

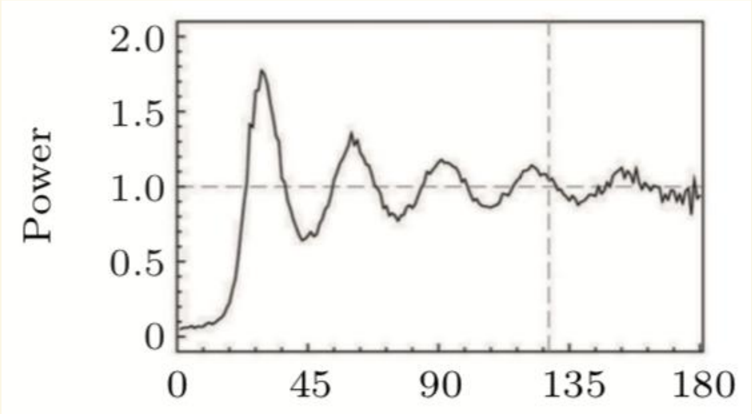


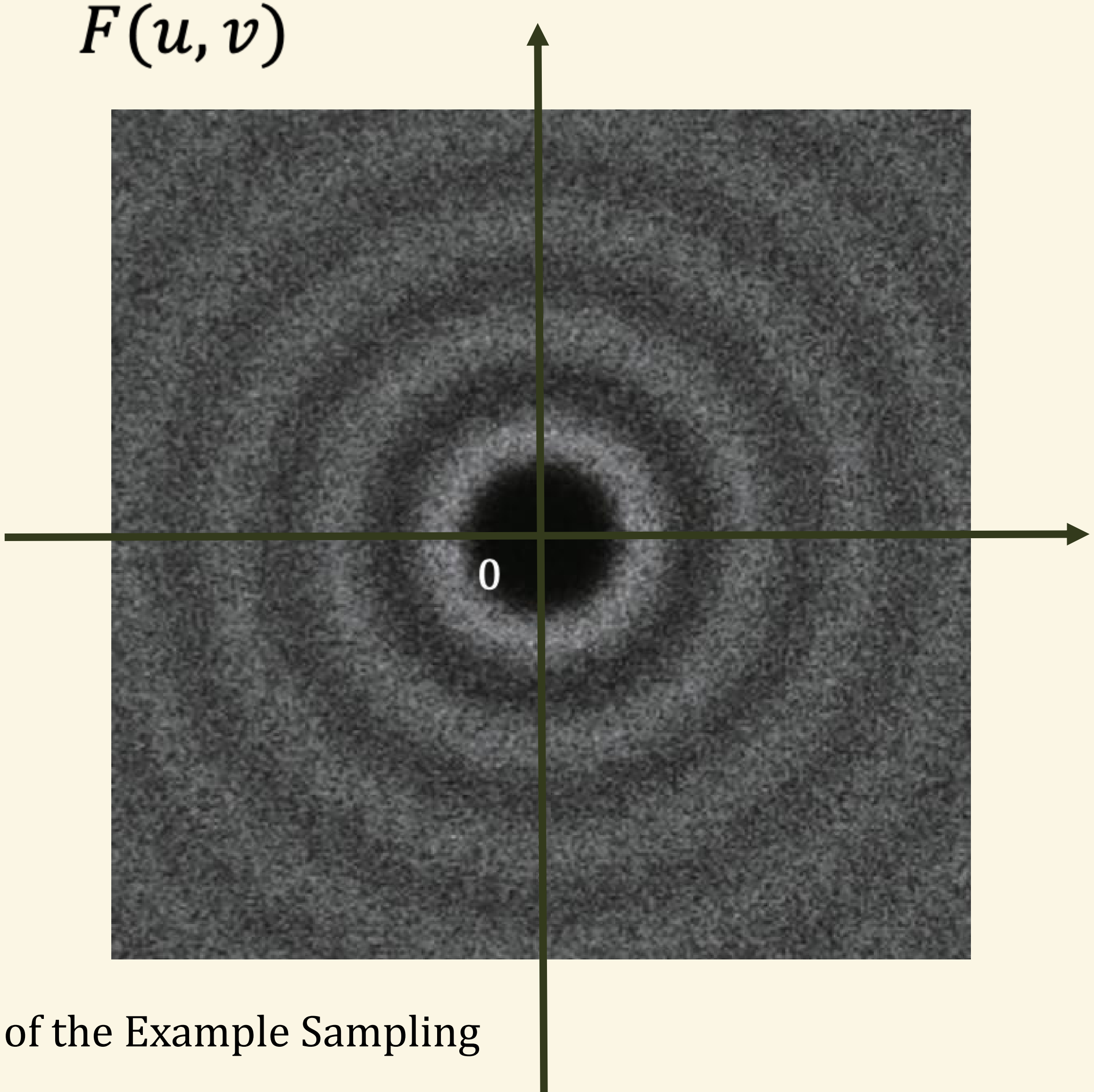


Example Sampling on 2D

Power Spectrum of Blue Noise Sampling

- The power spectrum of blue noise sampling
  - Lacks low frequency energy
  - Has structure residual peaks





Power Spectrum of the Example Sampling

The Blue Noise Sampling Example

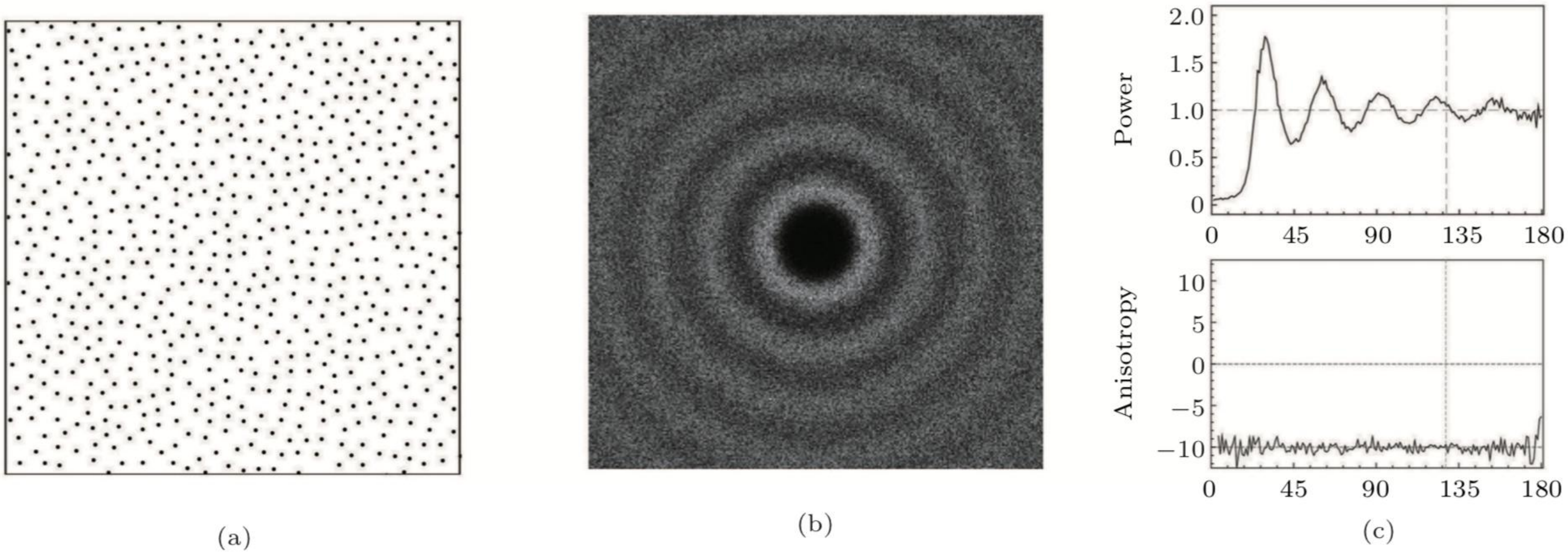
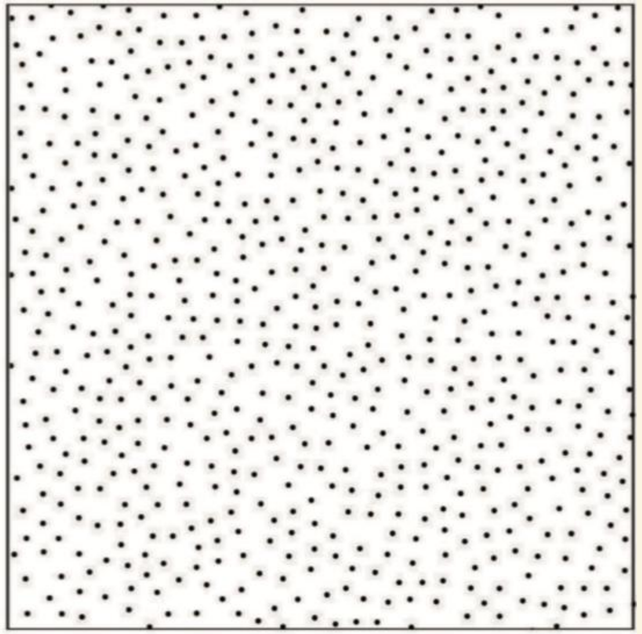


Fig. 1. Example of Poisson-disk sampling and its spectral analysis. (a) A sampled point set. (b) Power spectrum from this point set. (c) Radial means and normal anisotropy.

Different Sampling Domains

- 2D Euclidean Domain
- 3D Euclidean Domain
- High Dimensions
- 3D Surface



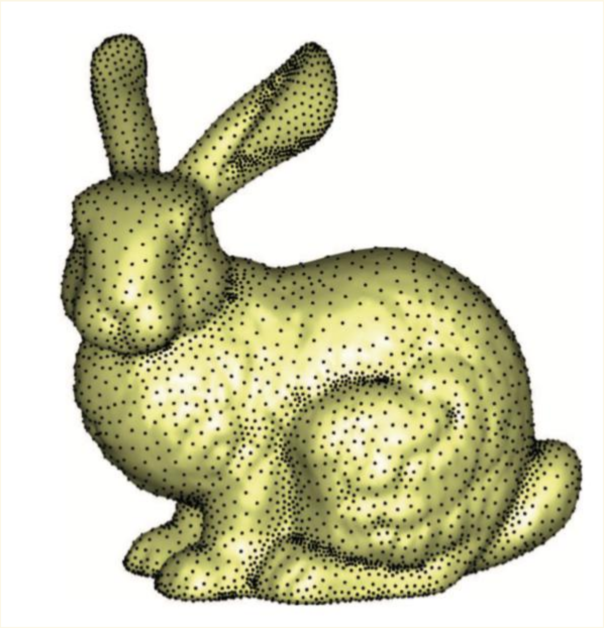
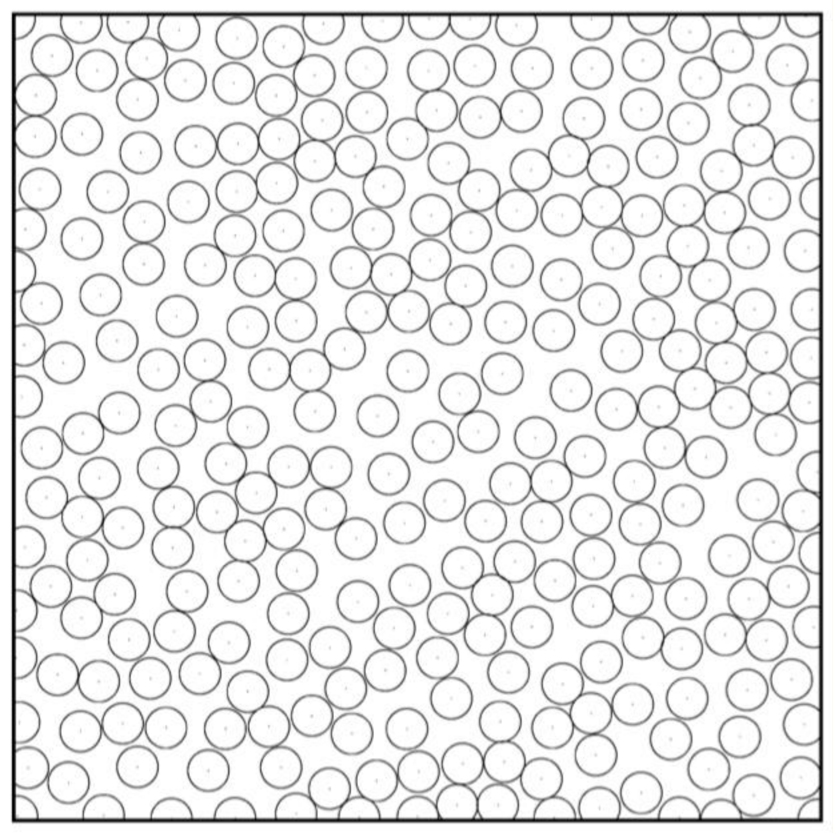


Table of Content

- Signal Processing Prerequisites
  - Fourier Transform
  - Power Spectrum of a Signal
  - 2D Discrete Fourier Transform
- Blue Noise Sampling
- Different Blue Noise Sampling Methods
  - Poisson Disk Sampling
  - Relaxation Based Sampling
  - Patch/Tile-Based Sampling
- Multiclass Blue Noise Sampling

Poisson Disk Sampling

- An ideal Poisson-Disk sampled point set $X = \{(\boldsymbol{x_i}, r)\}_{i=1}^{n}$ in sample domain Ω should satisfy
  - Minimal distance property
    $$Dist(\boldsymbol{x_i}, \boldsymbol{x_j}) > 2r$$
  - Unbiased sampling property
    $$\forall \boldsymbol{x}_i \in X, S \subseteq \Omega, P(\boldsymbol{x_i} \in \mathrm{S}) = \int dS$$
  - Maximal sampling property
    $\bigcup(\boldsymbol{x_i}, 2r) \supseteq \Omega$



An Example Sampling of Disks

Poisson Disk Sampling

- Algorithm: Dart Throwing
  - Randomly place a disk and check whether it satisfies constraints
  - Complexity for the first proposed algorithm $O(n^2)$
  - Very slow to converge
- Many algorithms to accelerate
  - By maintaining data structures about disk information
  - Have $O(n)$ and $O(n\log n)$ algorithms
  - However, fast algorithms often tend to break constraints
  - See Reference 4 for details

Poisson Disk Example

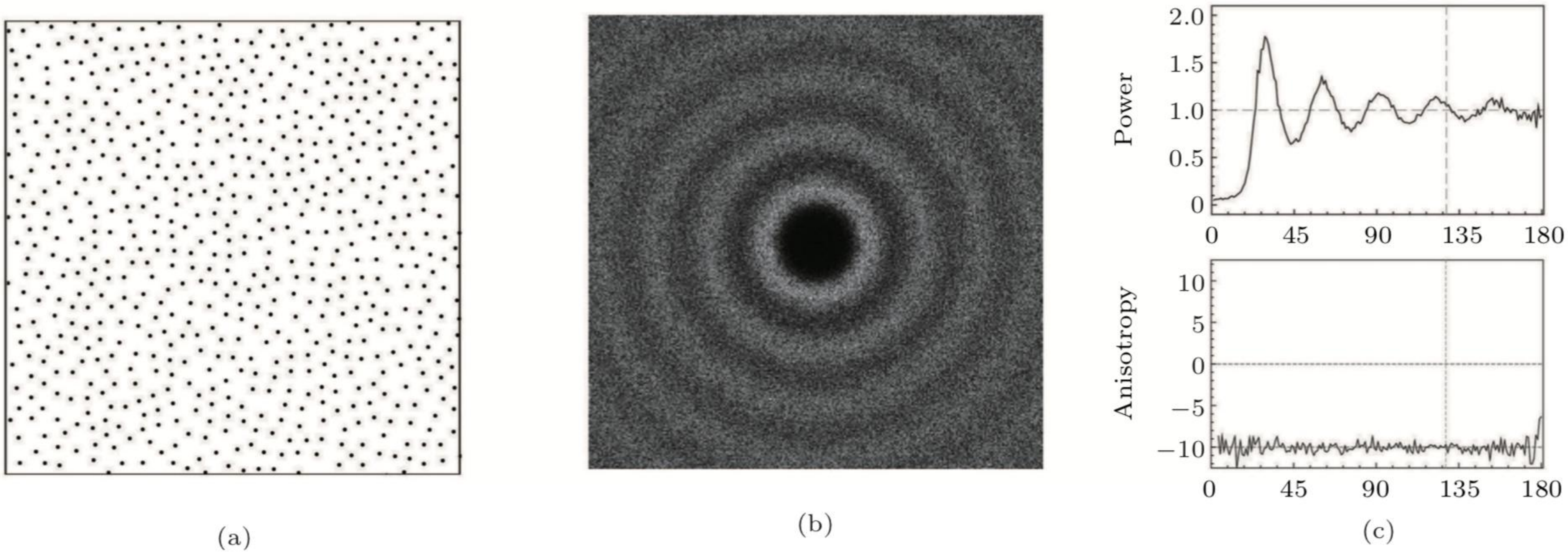
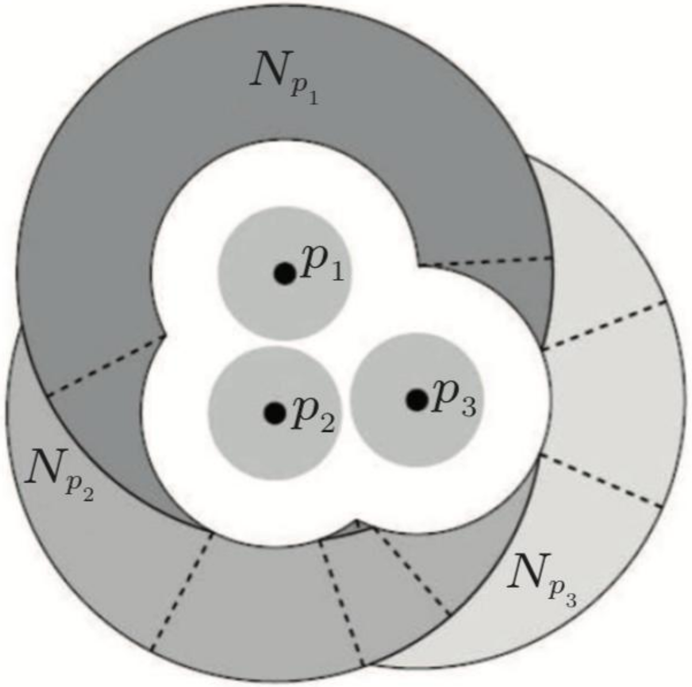
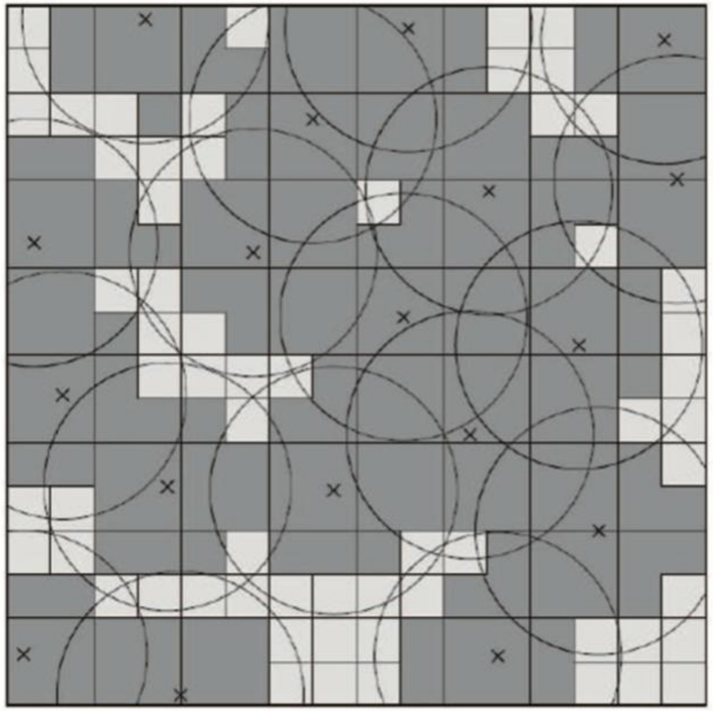


Fig. 1. Example of Poisson-disk sampling and its spectral analysis. (a) A sampled point set. (b) Power spectrum from this point set. (c) Radial means and normal anisotropy.

Examples of Acceleration Data Structures



(a)



(b)

Fig.2. Data structures used for accelerating Poisson-disk sampling. (a) Scalloped sectors[19]. (b) Quad-tree[13].

Examples of Gap Computation Methods

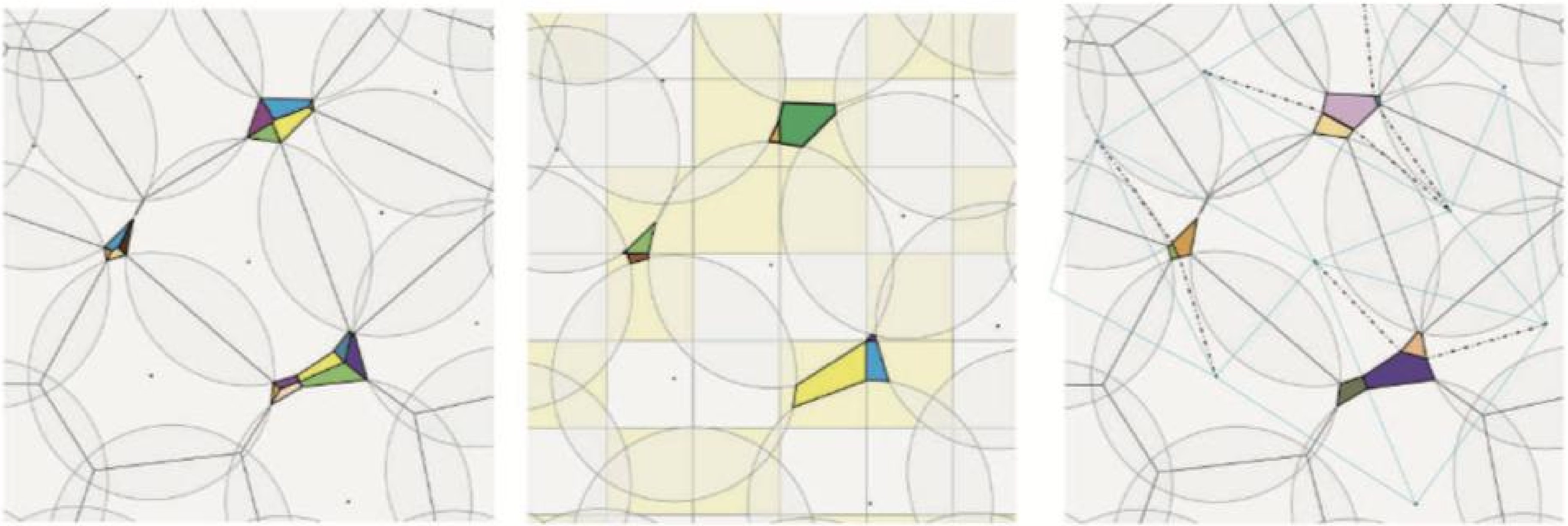


Fig.3. Comparison of three representative algorithms for gap computation. (a) Voronoi diagram[24]. (b) Uniform grid[12]. (c) Regular triangulation and power diagram[25].

Relaxation Based Sampling

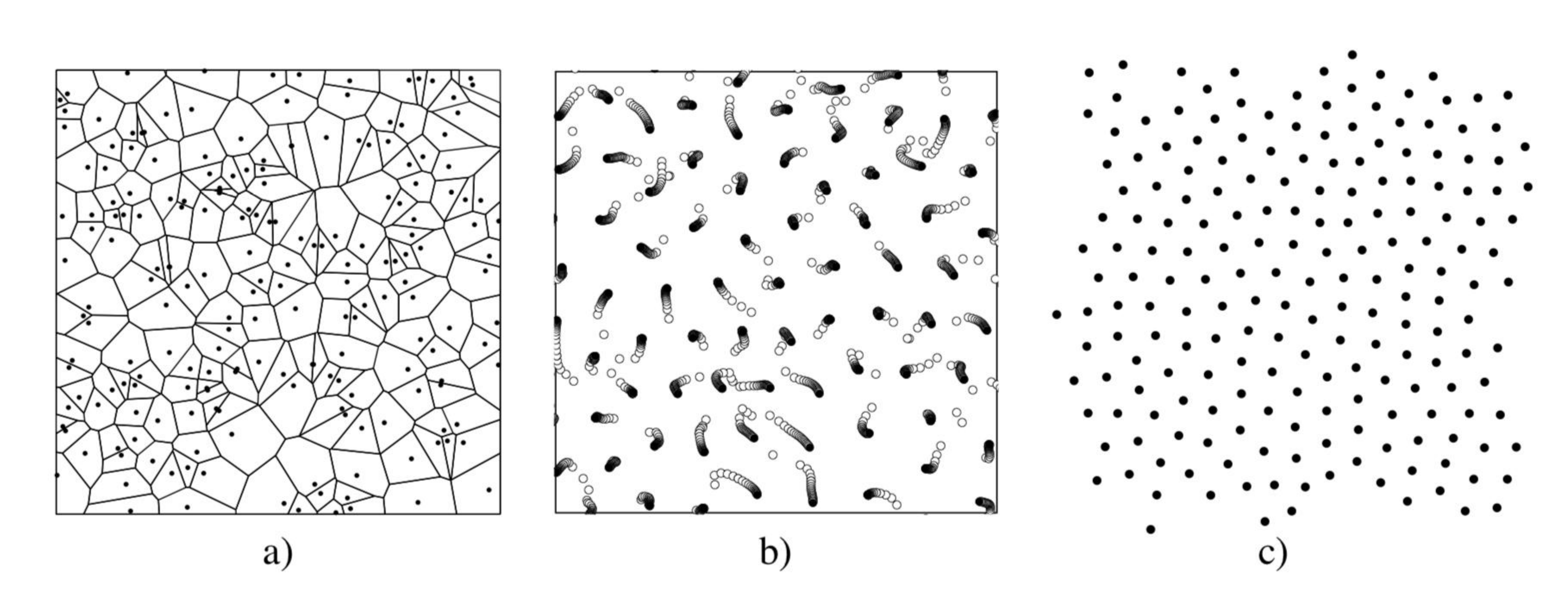
- Two steps
  - Generating an initial point set $X = \{\boldsymbol{x_i}\}_{i=1}^{n}$
  - Optimizing point positions using Lloyd iterations until convergence
- Minimize the energy function of centroidal Voronoi Tessellation

$$E_{CVT}(X) = \sum_{i=1}^{n} \int_{V_i} \rho(\boldsymbol{x}) \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 d\boldsymbol{x}$$

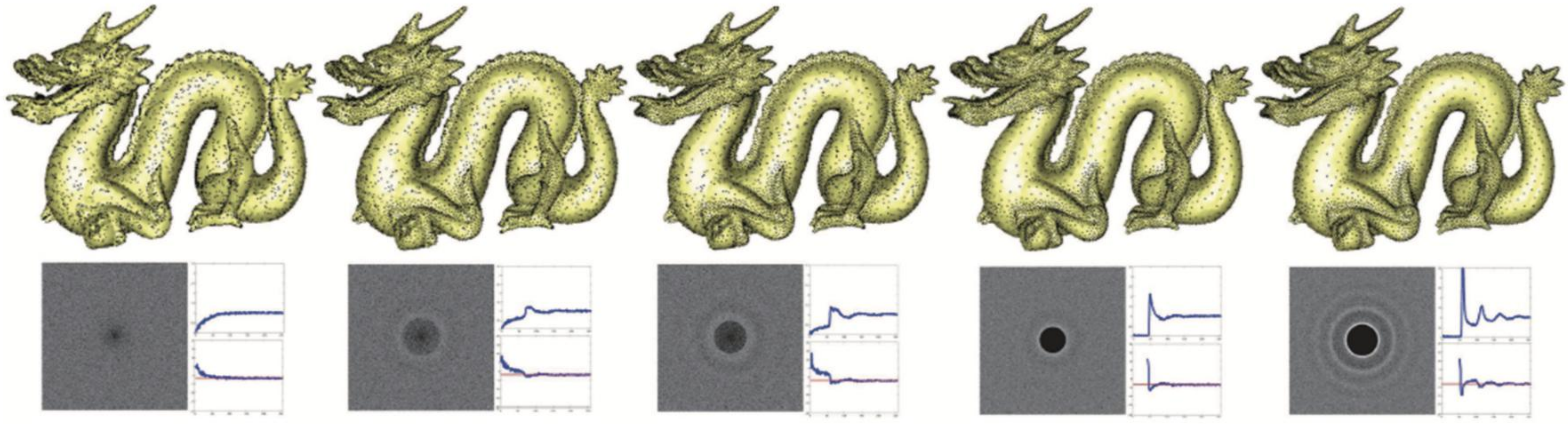
V_i: Voronoi Cells, $\rho(x)$: Density Function

- Many research into this

Relaxation Based Sampling

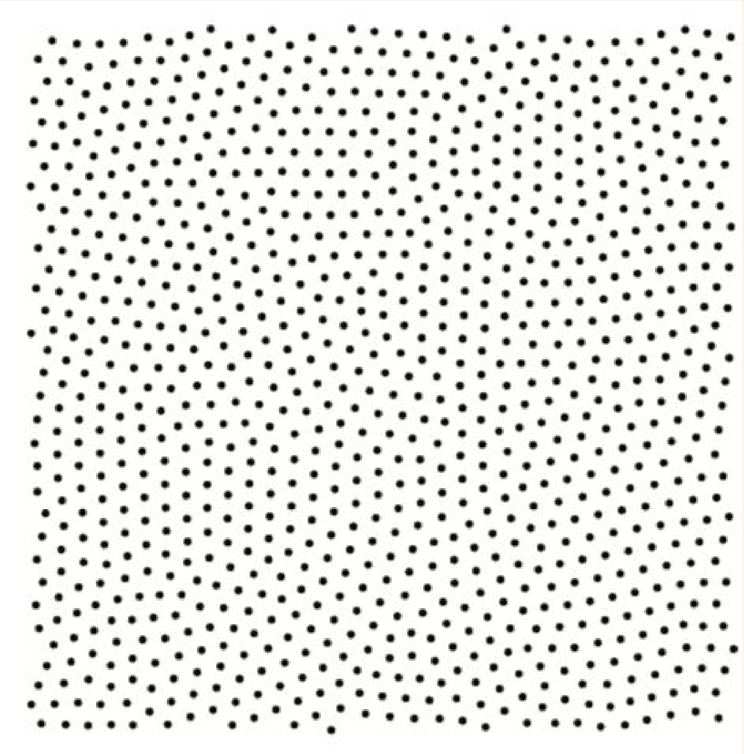


Relaxation Based Sampling

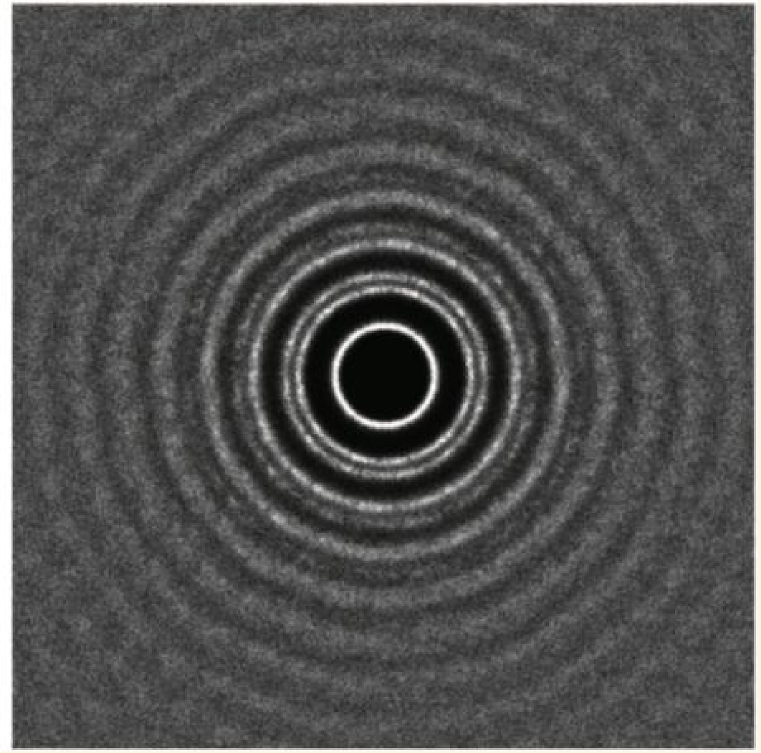


(a) (b) (c) (d) (e)

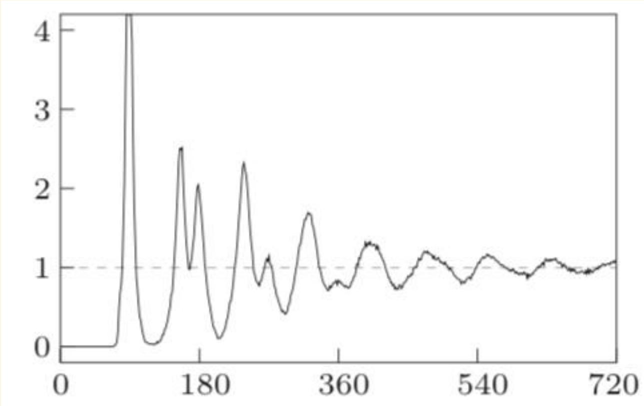
Relaxation Based Sampling



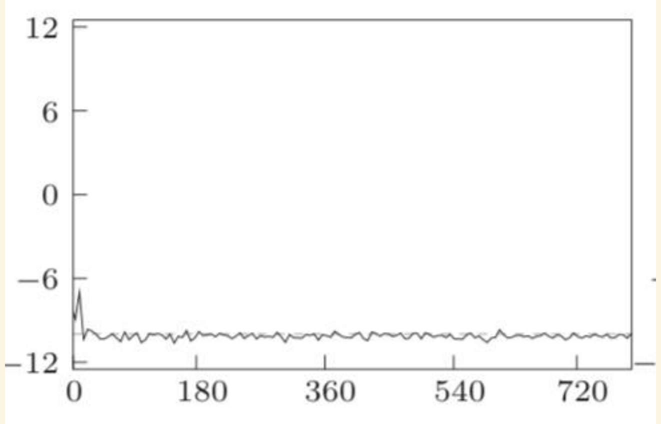
Sampled Points



Power Spectrum



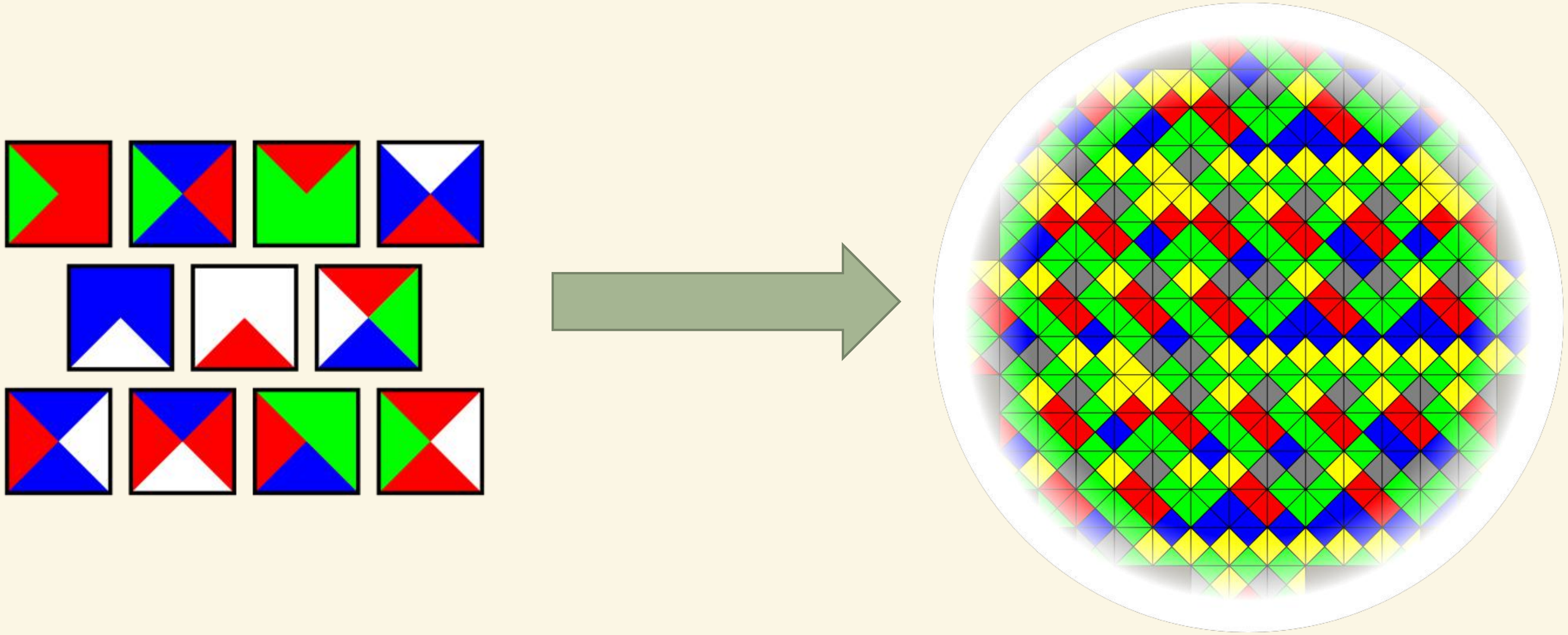
Power Spectrum



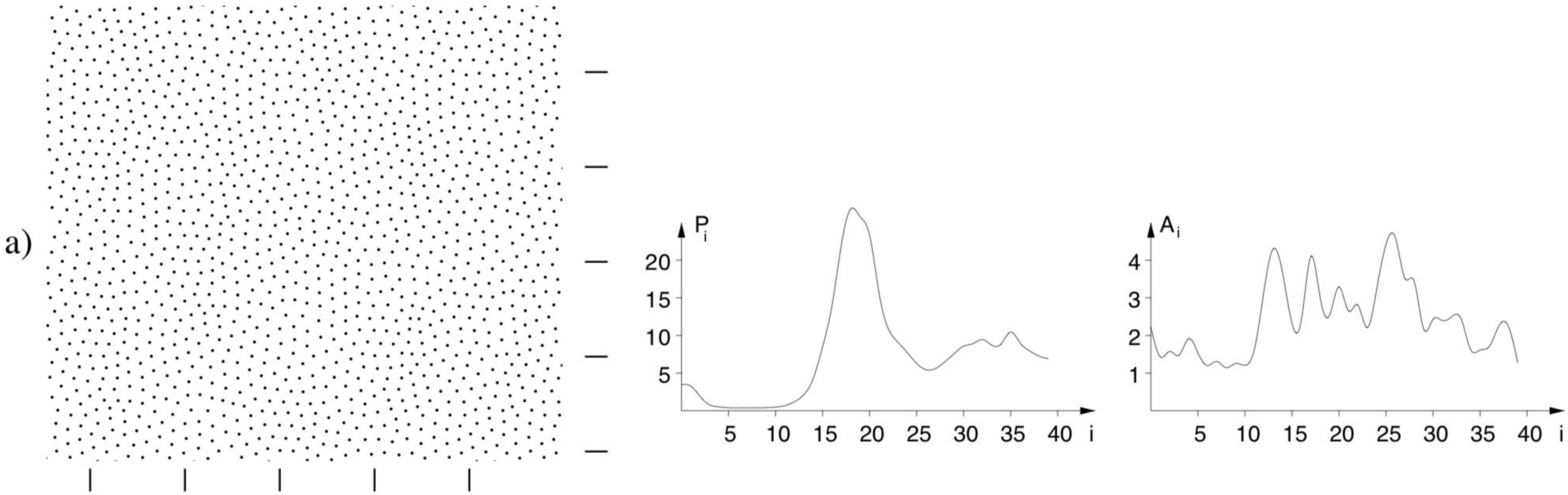
Anisotropy

Patch/Tile-Based Sampling

- One or more tiles are pre-computed and then placed next to each other to form point sets of arbitrary sizes
- For example, use Wang tiles



Patch/Tile Based Sampling (Wang Tile)



Summarizing These Sampling Methods

- Poisson-Disk Sampling
  - Can specify point density
  - Extensively studied and many methods of different quality and speed
- Relaxation Based Sampling
  - Can specify number of points
  - Slow but good quality
- Patch/Tile Based Sampling
  - Fast, can run in real-time
  - Sacrifice sampling quality
- What to use depend on the specific application

Table of Content

- Signal Processing Prerequisites
  - Fourier Transform
  - Power Spectrum of a Signal
  - 2D Discrete Fourier Transform
- Blue Noise Sampling
- Different Blue Noise Sampling Methods
  - Poisson Disk Sampling
  - Relaxation Based Sampling
  - Patch/Tile-Based Sampling
- Multiclass Blue Noise Sampling

Multiclass Blue Noise Sampling

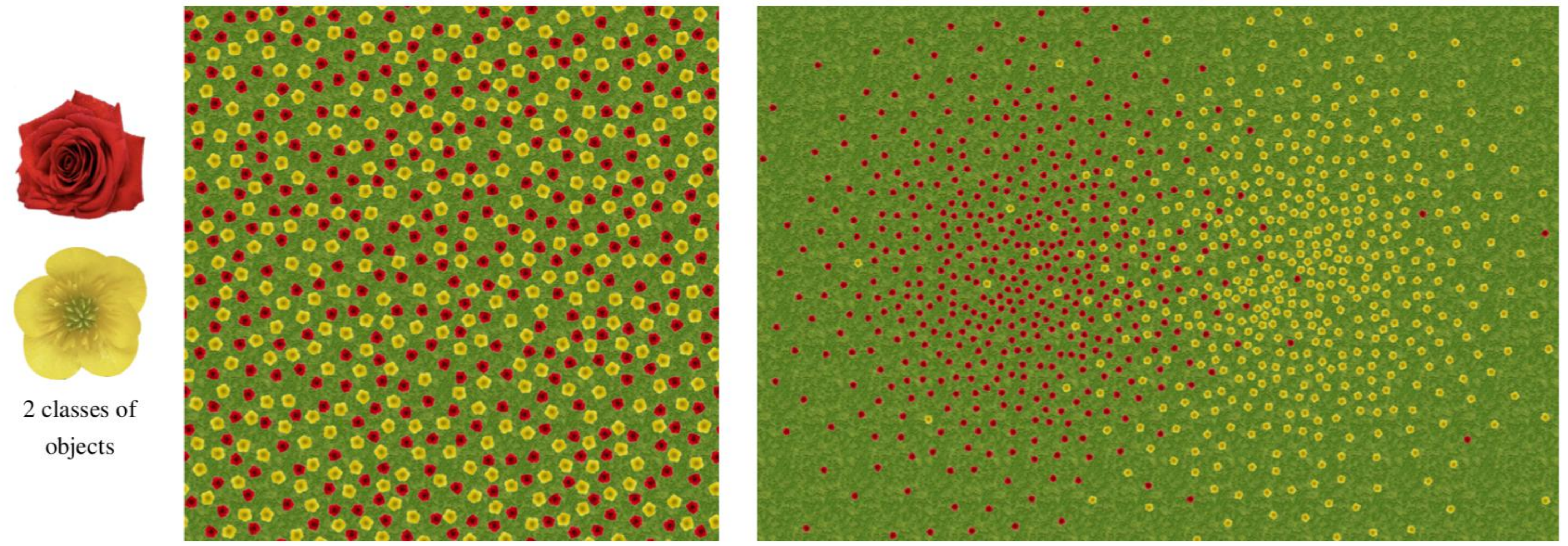


Figure 1: Object placement by multi-class blue noise sampling. Our algorithms can produce both uniform (middle) and adaptive (right) sampling results.

References

1. Wikipedia, “Fourier Transform”, “Power Spectrum”, “Noise”, “Color of Noise”, “Wang tiles”.
2. Fourier Transform, https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm
3. 何书元，《概率论》，北京大学出版社。
4. Yan, Dong-Ming, et al. "A survey of blue-noise sampling and its applications." *Journal of Computer Science and Technology*30.3 (2015): 439-452.
5. Wei, Li-Yi. "Multi-class blue noise sampling." *ACM Transactions on Graphics (TOG)*. Vol. 29. No. 4. ACM, 2010.
6. Hiller, Stefan, Oliver Deussen, and Alexander Keller. "Tiled blue noise samples." *Vision, Modeling, and Visualization (VMV)*. 2001.
7. Du, Qiang, Vance Faber, and Max Gunzburger. "Centroidal Voronoi tessellations: Applications and algorithms." *SIAM review* 41.4 (1999): 637-676.

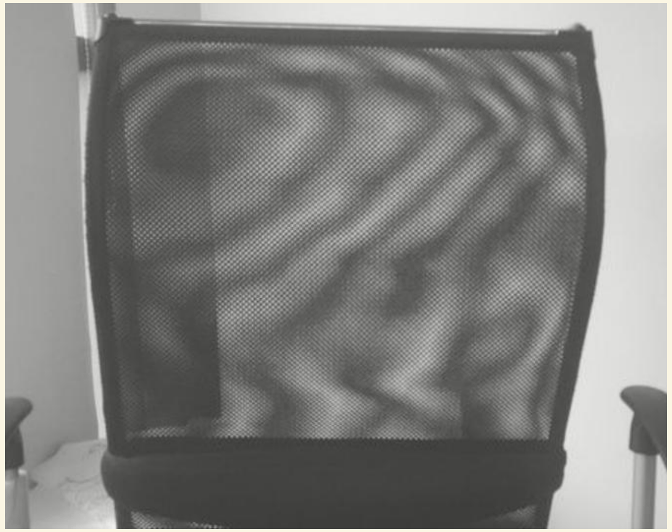
Thank you!

Blue Noise Application

Shixun Wu


无双
PROJECT GUTENBERG
钞级大骗 正在热映


知乎 @高胜禹




「每周一知」
“莫列波纹”
Moiré Pattern
日本平面设计师倉嶌隆广的扫描动画书《Poemotion》
亦称摩尔纹、莫尔条纹，是多组栅栏状条纹重叠后产生的干涉影像。在移动过程中，影像会产生图案变化。
莫列波纹广泛应用在钞票防伪、海洋勘测等领域。
21 星期三
NOV
关注顺丰科技UED
每周一条设计知识

Rendering

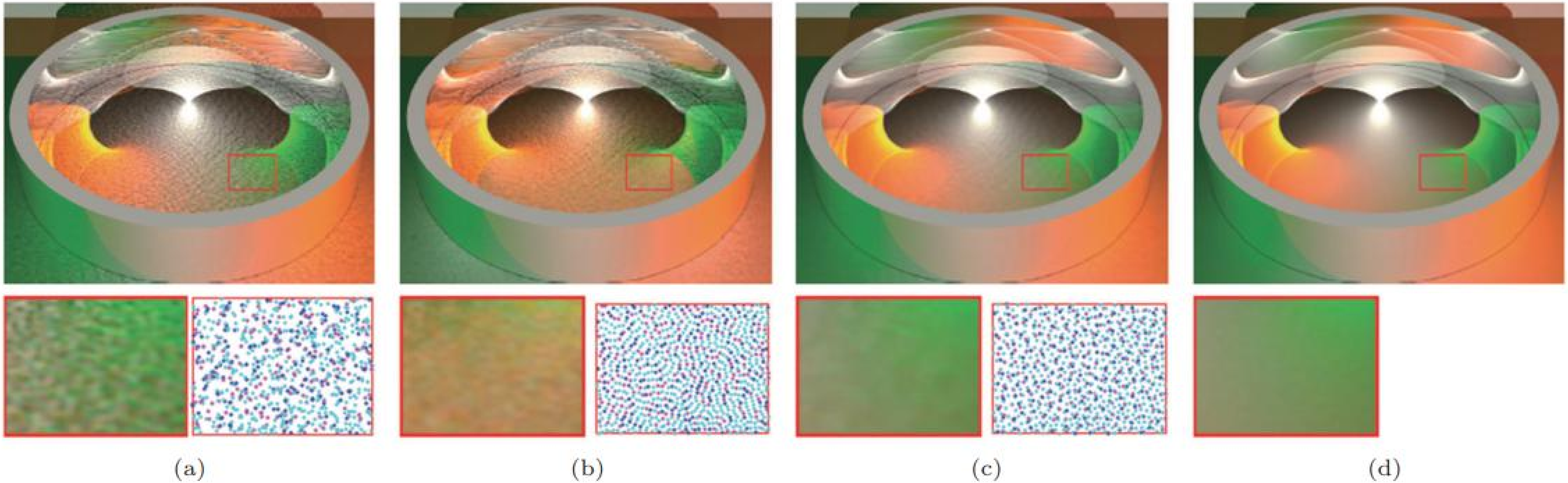
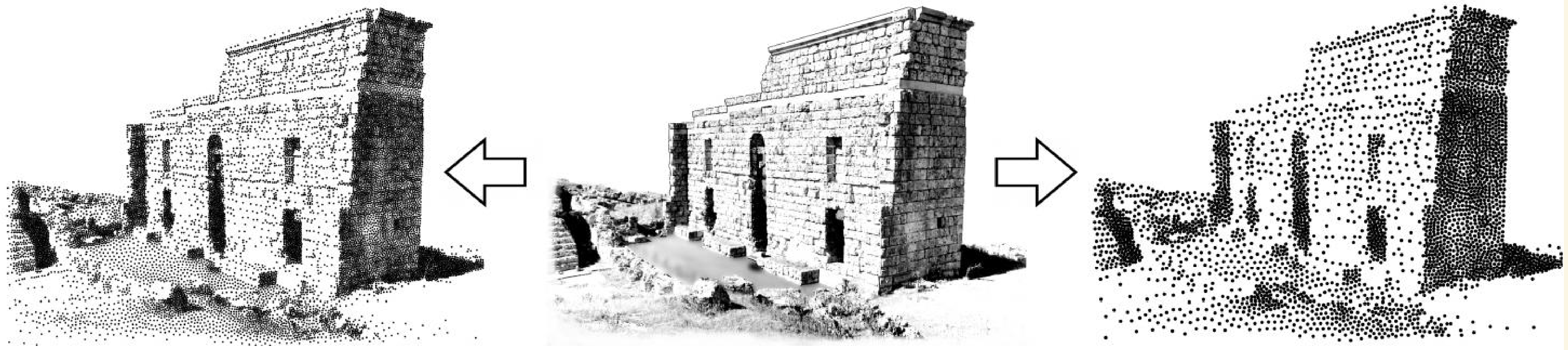


Fig. 7. Comparison of photon mapping results of different methods. (a) Unrelaxed result. (b) Result of [87]. (c) Result of [59]. (d) Reference 40X photons.

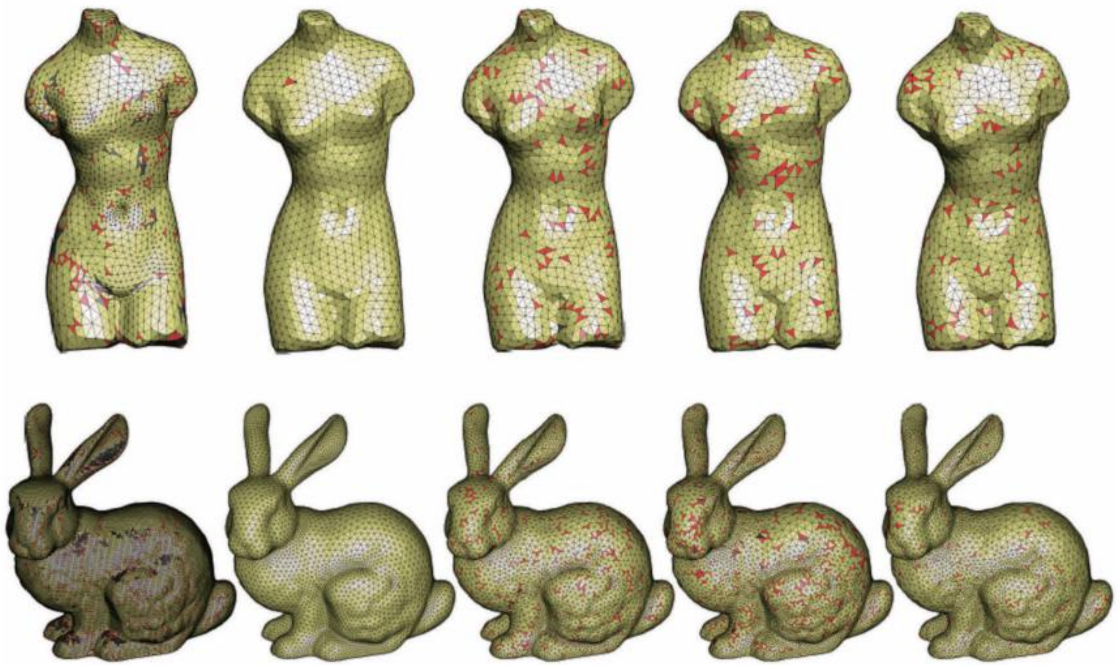
Image/Video Stippling

(a) (b)

Fig.8. Blue-noise sampling for image and video stippling. (a) Image stippling[54]. (b) Video stippling[90].



Meshing



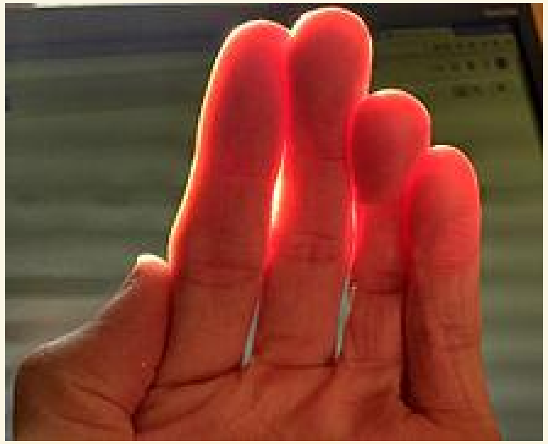
(a) (b) (c) (d) (e)

Fig. 11. Blue-noise surface remeshing. Top row: uniform remeshing. Bottom row: adaptive remeshing. The result of CVT always has the best meshing quality but lacks blue-noise features, while the other blue-noise remeshing methods are able to generate competitive results. The red triangles have angles larger than 90°, and the gray triangles have angles smaller than 30°. (a) Input. (b) CVT. (c) CapCVT. (d) FPO. (e) MPS.

Photon Mapping

- Caustics
- Diffuse interreflection
- Subsurface scattering





The blue noise sample on Photon Mapping

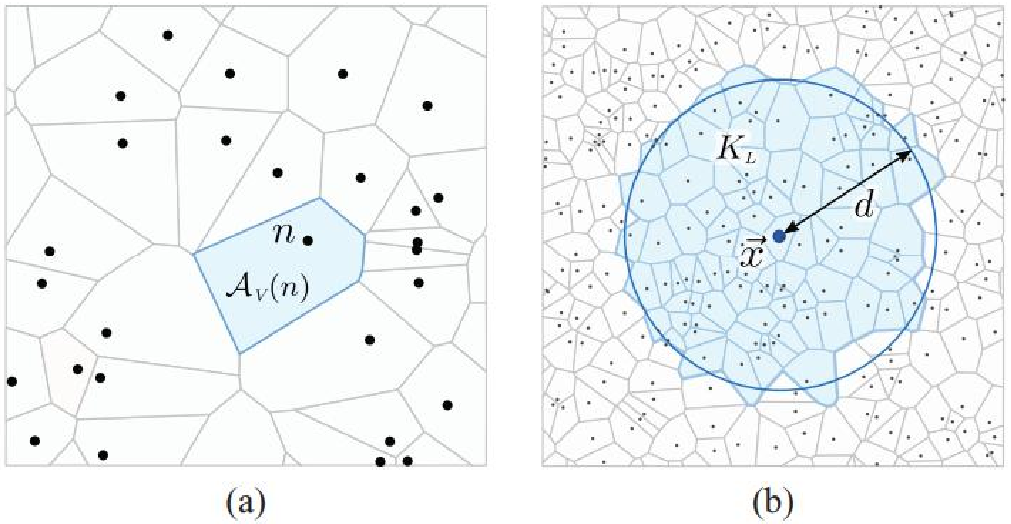


Fig. 2. (a) The Voronoi tessellation links flux density with the discrete set of photon sites. (b) A k-NN estimator reconstructs exitant radiance area using the disk of radius d spanning the subset of K_L-nearest photons local to the query point $\vec{x}$.

Sampling and Estimation

- Random Sample → A lot of noise

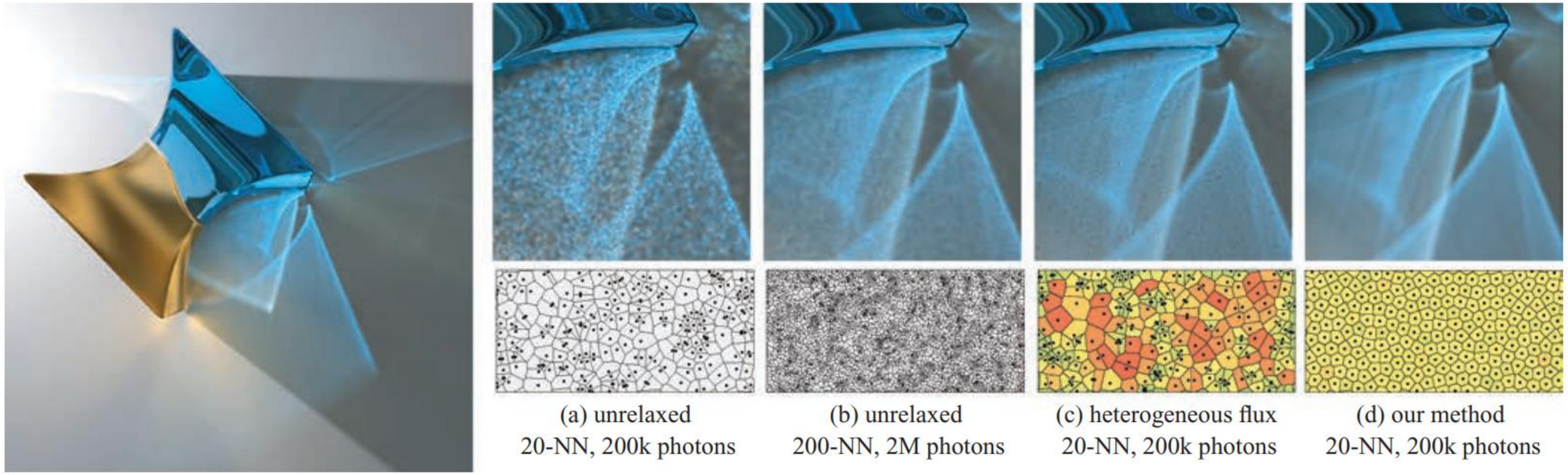


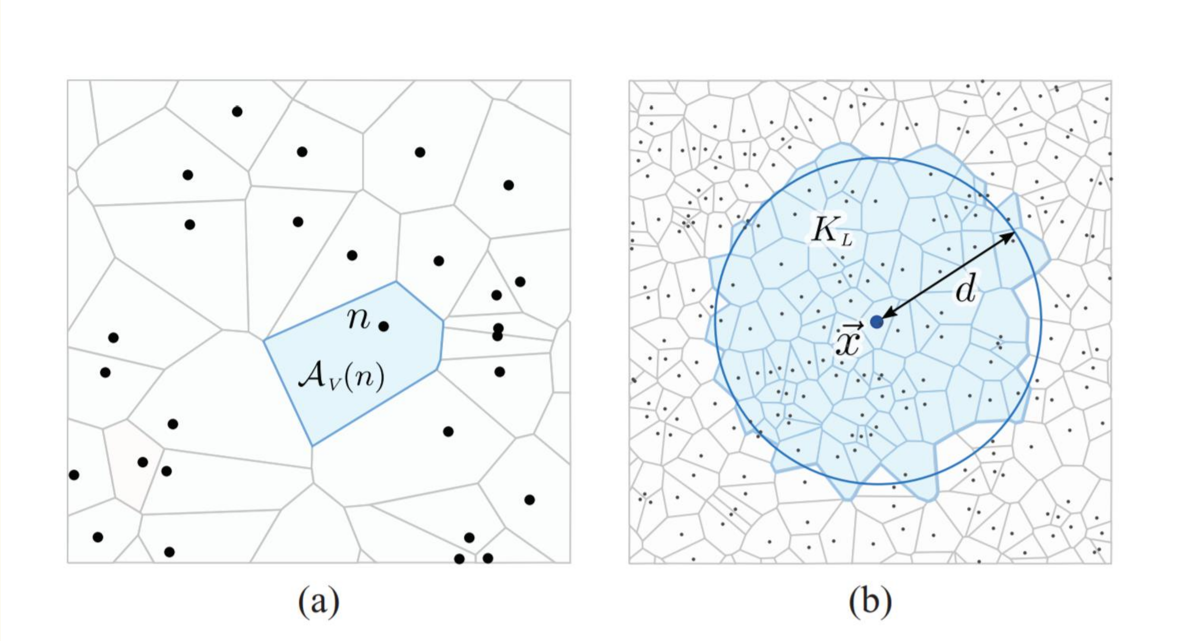
Fig. 4. A complex dielectric object generating both specular and glossy caustics. (a) An unrelaxed photon map exhibits high levels of noise. (b) Noise reduction by using an order of magnitude more photons in both the map and k-NN kernel. (c) Integrating flux density over each photon counterbalances variance but leaves residual error. (d) Our method relaxes the photon map thereby minimizing error due to discrepancy.

Where the noise comes from?

- The estimation is not accurate
- Estimation of continuous points not continuous

How to deal with it?

- Increase the sampling points
- Increase the estimation area
- It is a trade off

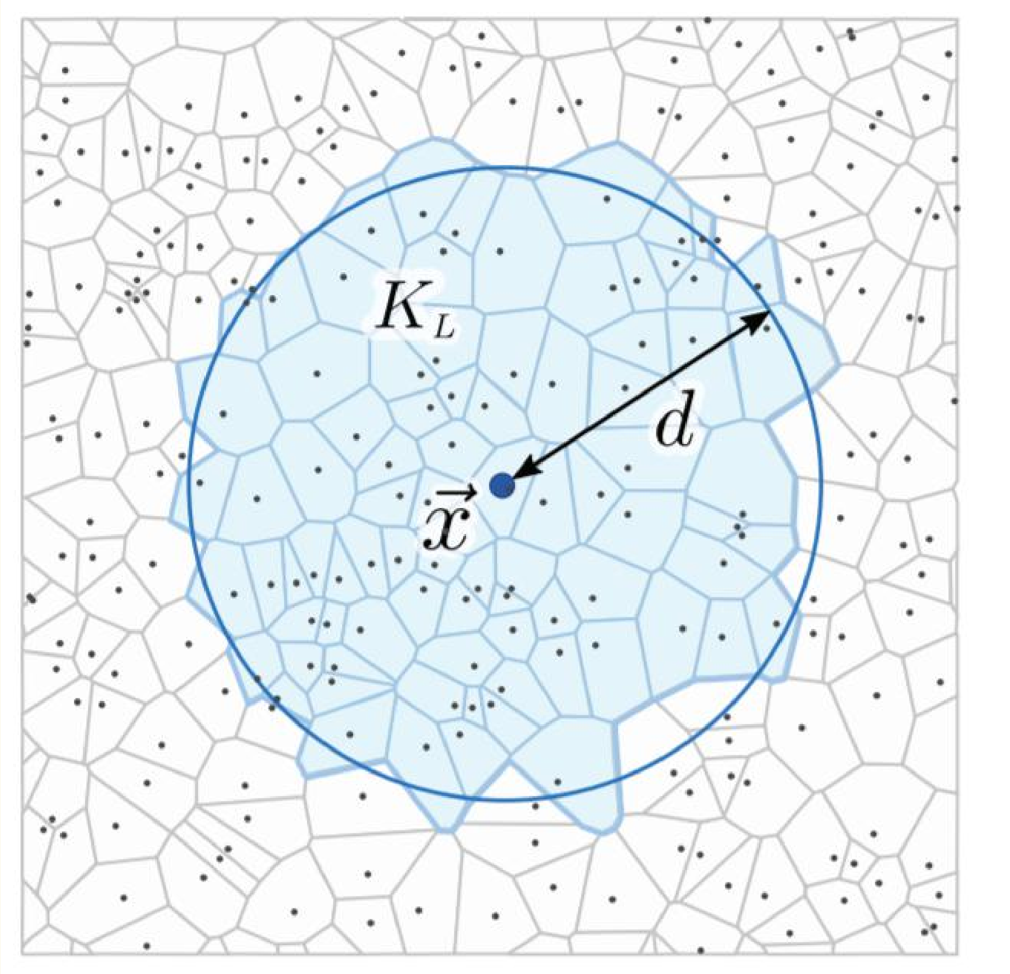


Disadvantages

- Increasing sampling points is a pressure on **storage**

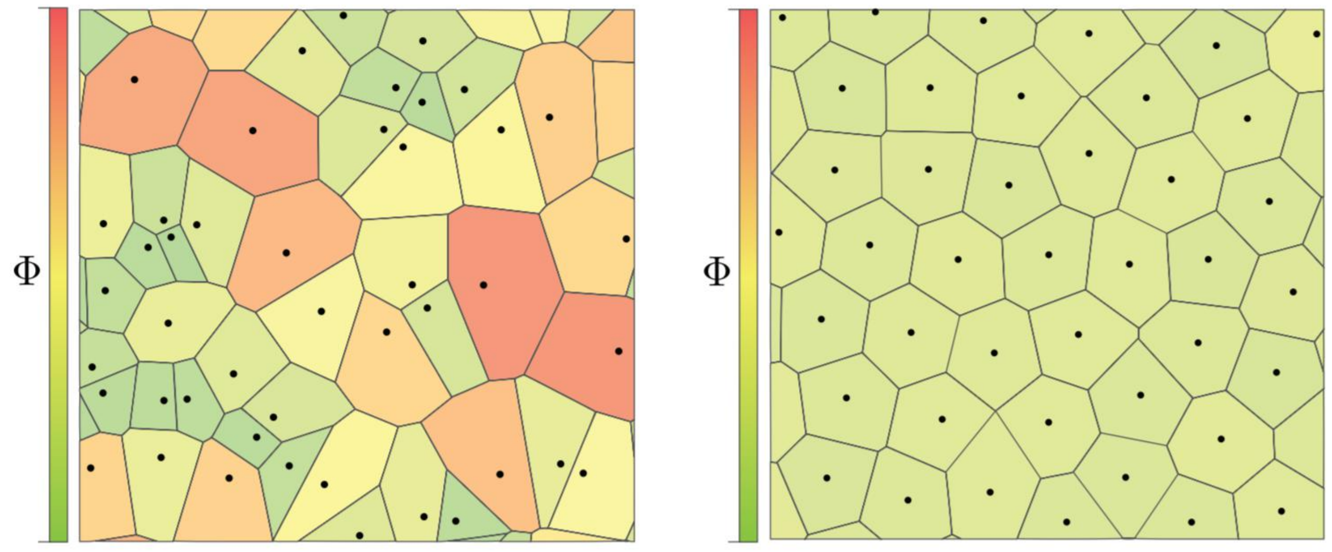
Voronoi cell vs. Single Phonton

- Estimate by the flux of a Voronoi cell
- Not by the flux of a single phonton



New Problem:

- Approximation error on the edge
- Caused by the irregular shape of veronoi cell



(a) Heterogeneous flux (b) Homogeneous flux

Blue Noise Distribution

- Minimize the variance

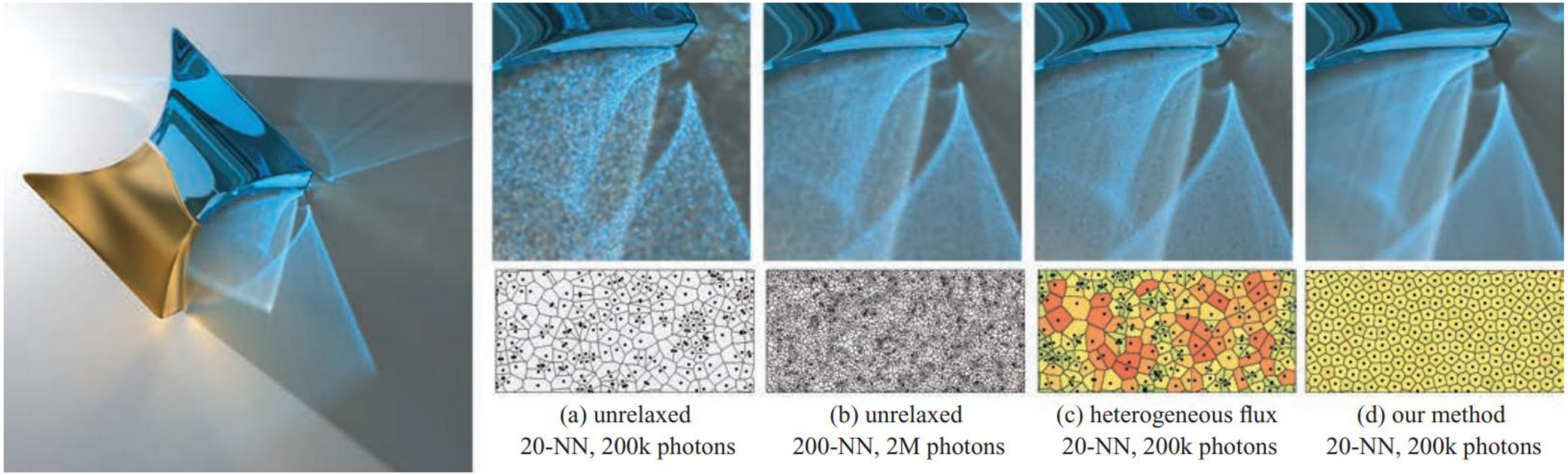


Fig. 4. A complex dielectric object generating both specular and glossy caustics. (a) An unrelaxed photon map exhibits high levels of noise. (b) Noise reduction by using an order of magnitude more photons in both the map and k-NN kernel. (c) Integrating flux density over each photon counterbalances variance but leaves residual error. (d) Our method relaxes the photon map thereby minimizing error due to discrepancy.

Algorithm

ALGORITHM 1: ProgressiveRelax()

$N \leftarrow$ CastPersistentPhotons()
$T = 1$
$\delta t = \frac{1}{|N|}$
repeat
  $T_{i+1} = \alpha T$ $\{$*Eq. 13*$\}$
  while $T < T_{i+1}$ ***do***
    $m \leftarrow$ CastTransientPhoton()
    if *m is stored* ***then***
      $T = T + \delta t$
      $n \leftarrow$ FindIntersectedCell(m)
      $\Sigma_\Phi(n) = \Sigma_\Phi(n) + \Delta\Phi_p(m)$
    end if
    DeletePhoton(m)
  end while
  for all $n \in N$ ***do***
    $r_V(n) \leftarrow$ EstimateLocalArea(n)
    $r_\gamma(n) = r_V(n)\sqrt{\hat{\gamma}(n)^{\mathcal{W}(n)}}$ $\{$*Eqs. 8 and 15*$\}$
  end for
  for all $n \in N$ ***do***
    $\vec{x}(n) = \vec{x}(n) + \mathcal{W}(n)\vec{f}(n)$ $\{$*Eq. 15* $\}$
  end for
until User stop

Monte Carlo Estimation of Flux

$$\int_{\mathcal{A}_V(n)} B(a)da = \int_{\mathcal{A}_V(n)} \frac{d\Phi(a)}{da} da \,.$$

$$\int_{\mathcal{A}_V(n)} \frac{d\Phi(a)}{da} da \equiv \lim_{\frac{1}{|M|}\to 0} \sum_{m\in V(n)} \Delta\Phi_p(m)$$

$$\int_{\mathcal{A}_V(n)} \frac{d\Phi(a)}{da} da \approx \frac{1}{T} \sum_{m\in V(n)} \Delta\Phi_p(m)$$

$$\gamma(n) = \frac{1}{|N|} \int_{\sigma} B(a)da \left[\int_{\mathcal{A}_V(n)} B(a)da\right]^{-1}$$

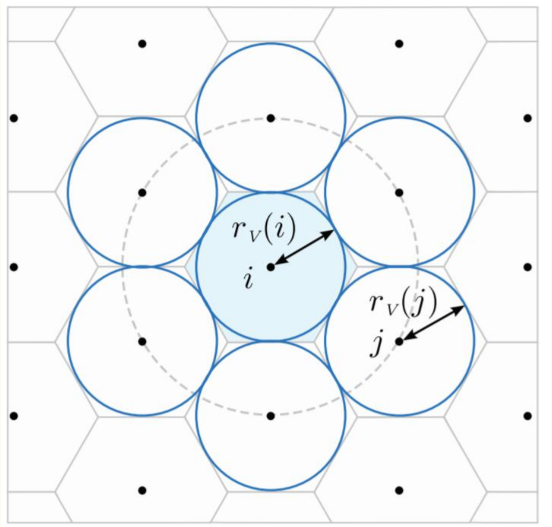
$$\hat{\gamma}(\mathrm{n}) = \frac{1}{|N|} \sum_{m\in M} \Delta\Phi_p(m) \left[\sum_{m\in V(n)} \Delta\Phi_p(m)\right]^{-1}$$

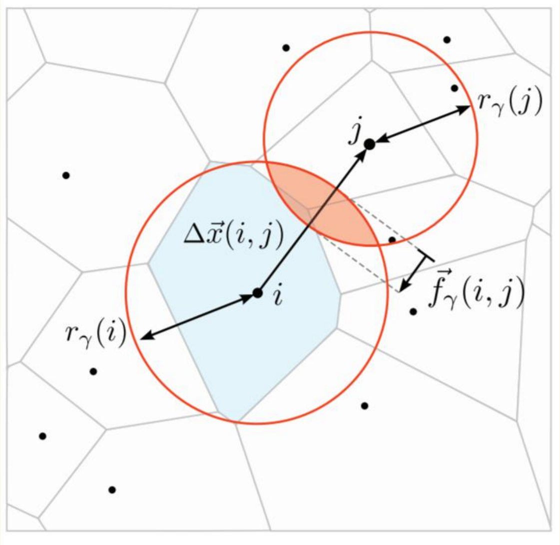
Relaxation Operator

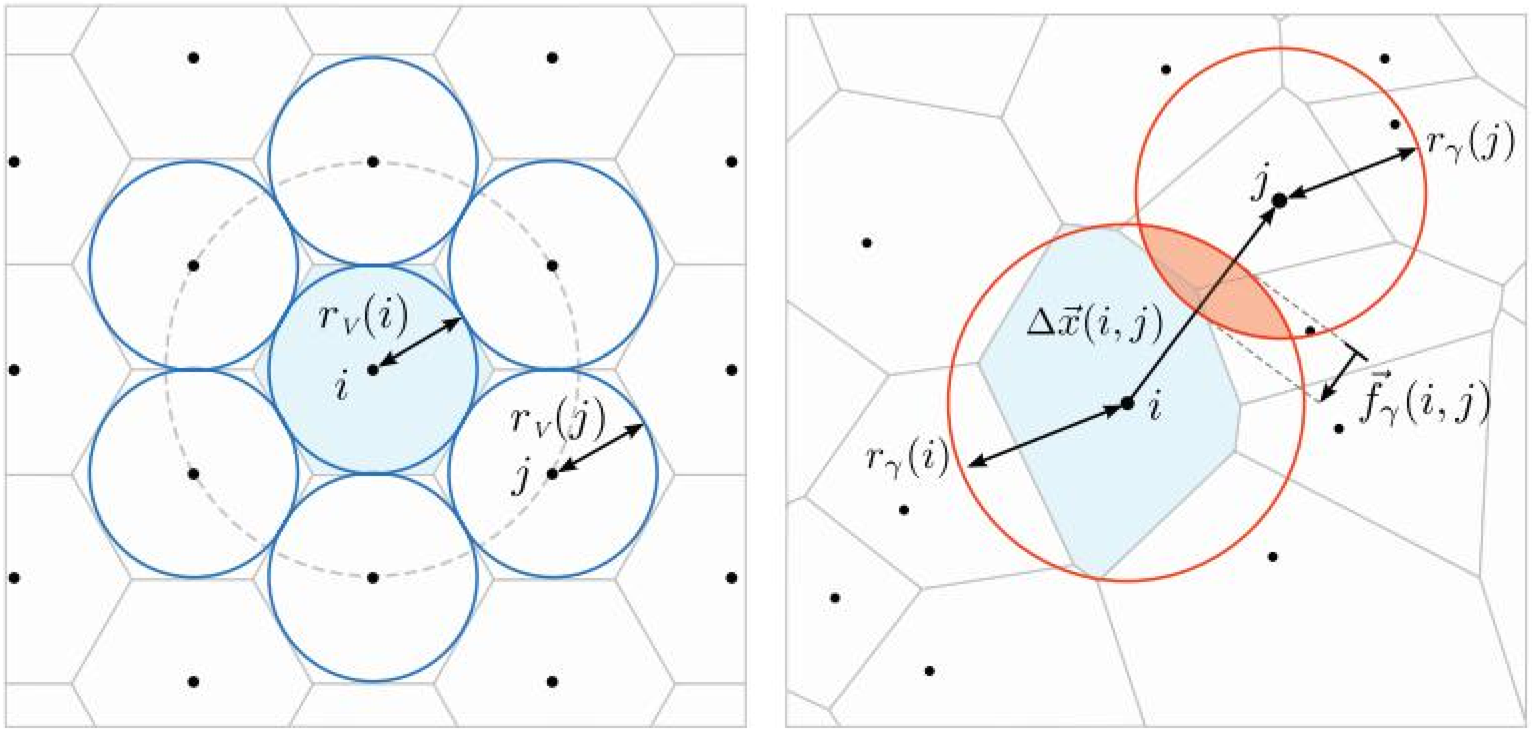
$$f_\gamma(i, j) = \|\Delta\vec{x}(i, j)\| - (r_\gamma(j) + r_\gamma(i)) .$$

$$f_\gamma(i, j) = \|\Delta\vec{x}(i, j)\| - (r_\gamma(j) + r_\gamma(i)) .$$

$$f_V(i, j) = r_V(j) - r_V(i) .$$

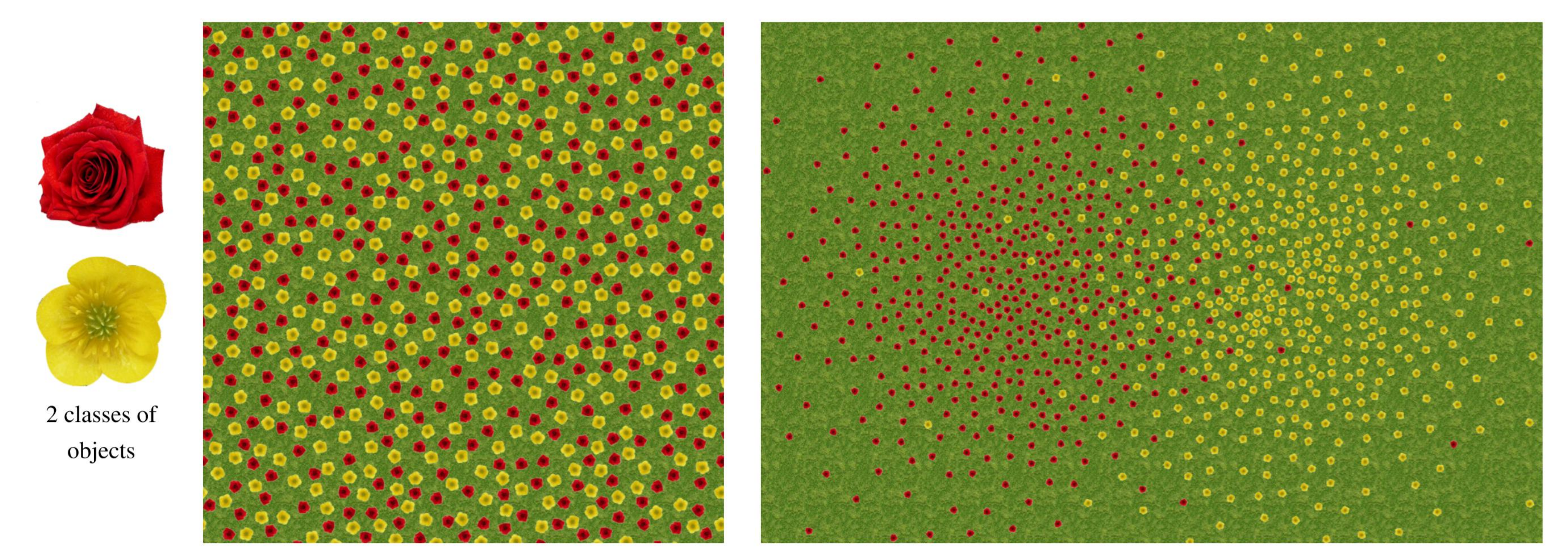






(a) (b)

Multi Class Blue Noise Sampling

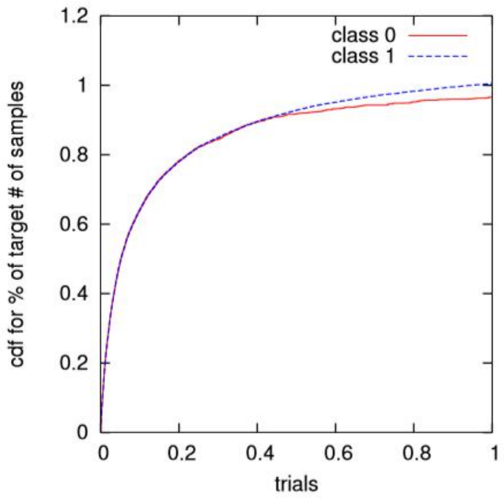


Sample Class

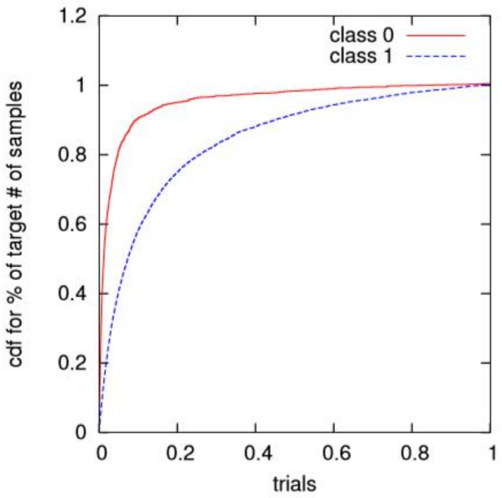
- Fill rate

$$N_i = N \frac{\frac{1}{r_i^n}}{\sum_{j=0}^{c-1} \frac{1}{r_j^n}}$$

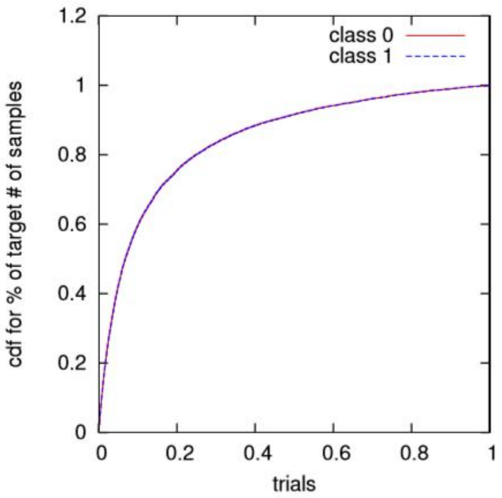
Sample Control



247810.7 trials
0 killed
237444.7 rejected
10366 accepted



240310.5 trials
0 killed
229944.5 rejected
10366 accepted



157388.3 trials
1023.7 killed
145998.6 rejected
11389.7 accepted

R-Matrix

function $\mathbf{r} \leftarrow \text{BuildRMatrix}(\{r_i\}_{i=0:c-1})$

  // $\{r_i\}$*: user specified per-class values*

  // *c: number of classes*

  for i = 0 to c-1

    $\mathbf{r}(i,i) \leftarrow r_i$ // *initialize diagonal entries*

  end

  sort the c classes into priority groups $\{\mathbf{P}_k\}_{k=0:p-1}$ with decreasing r_i

  // *classes in the same priority group have identical* r *values*

  $C \leftarrow \emptyset$ // *the set of classes already processed*

  $D \leftarrow 0$ // *the density of the classes already processed*

for k = 0 to p-1

  $C \leftarrow C \bigcup \mathbf{P}_k$

  foreach class $i \in \mathbf{P}_k$

    $D \leftarrow D + \frac{1}{r_i^n}$ // n *is the dimensionality of the sample space*

  end

  foreach class $i \in \mathbf{P}_k$

    foreach class $j \in C$

      if $i \neq j$

        $\mathbf{r}(i,j) \leftarrow \mathbf{r}(j,i) \leftarrow \frac{1}{\sqrt[n]{D}}$ // $\mathbf{r}$ *is symmetric*

    end

  end

end

return r

Overlapping