matfyz

Computer Graphics Charles University

Anti-aliasing and Sampling

© 1996-2020 Josef Pelikán

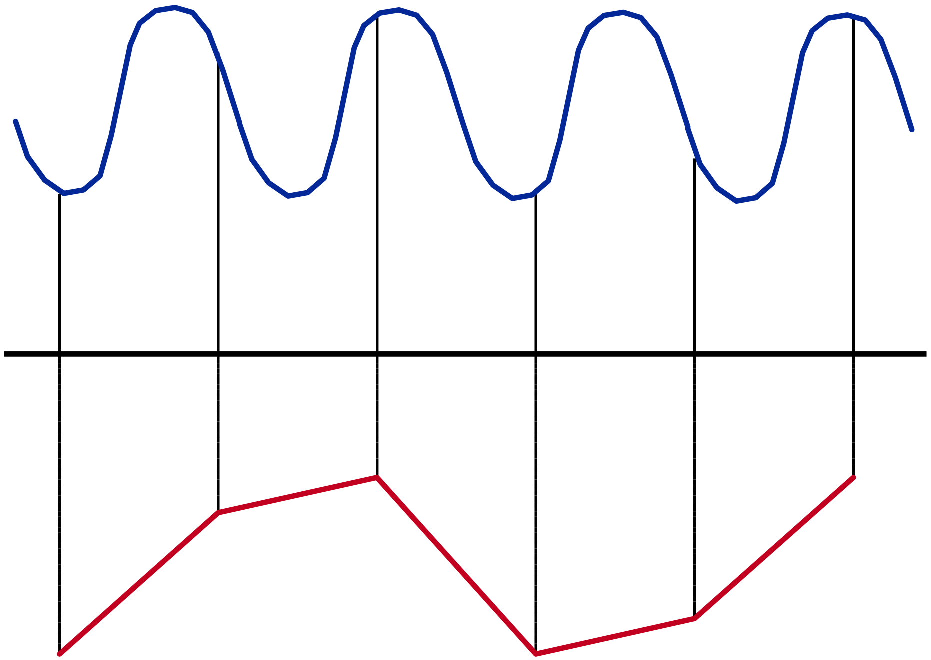
CGG MFF UK Praha

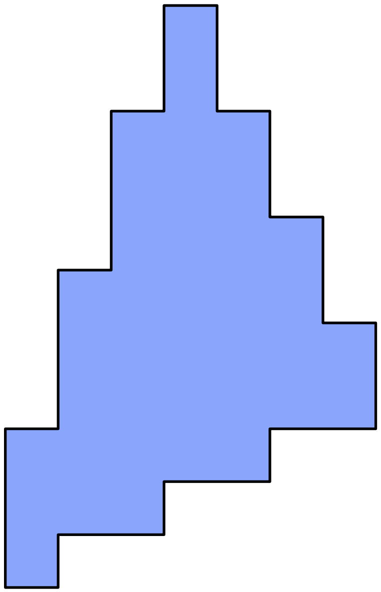
pepca@cgg.mff.cuni.cz

https://cgg.mff.cuni.cz/~pepca/

Spatial and temporal alias

Alias – artefacts caused by an unsufficient (regular) sampling





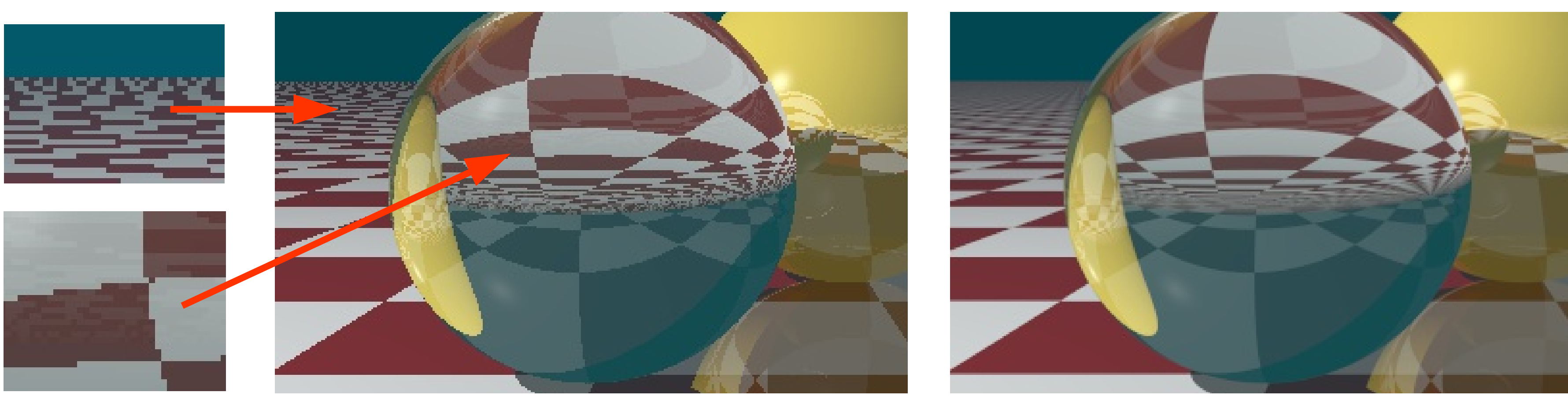
Spatial alias

Jagged oblique lines

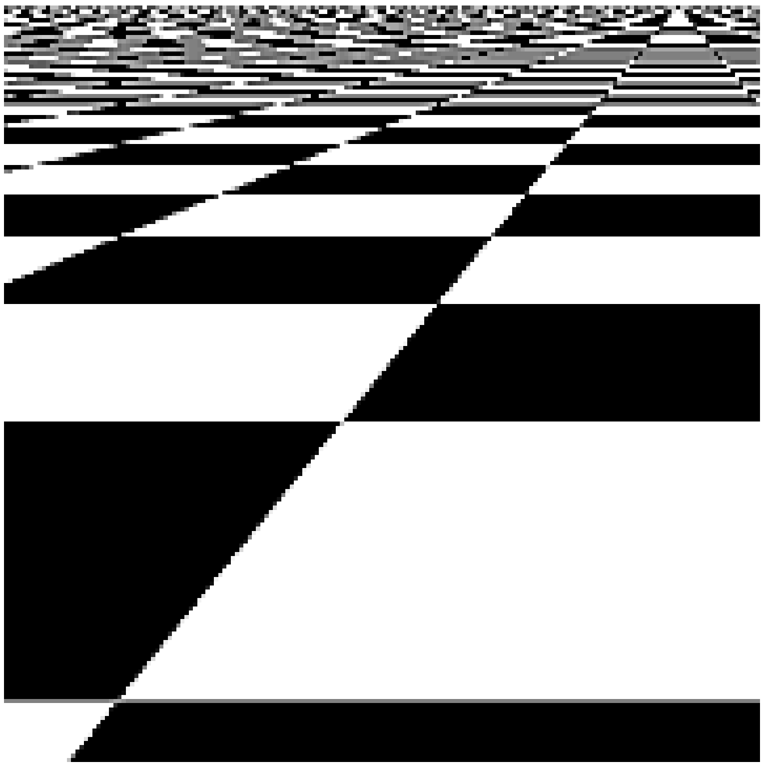
- regular dense system of lines or stripes on a texture can lead to "Moiré effect"

Interference of fast periodic image changes with a pixel raster

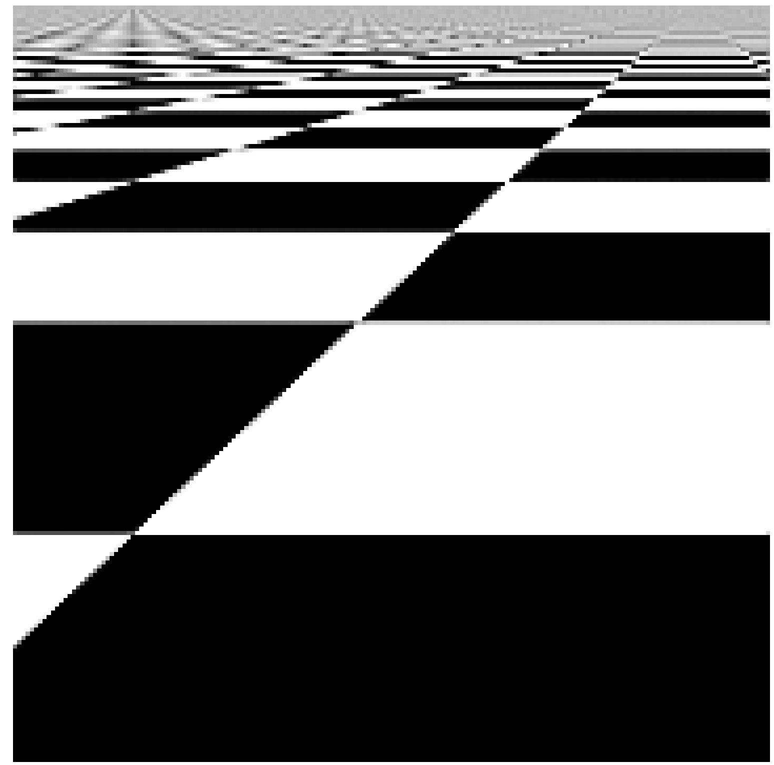
- example – picket fence in perspective projection
- too fine or too distant regular texture (checkerboard viewed from distance)



Spatial alias – checkerboard



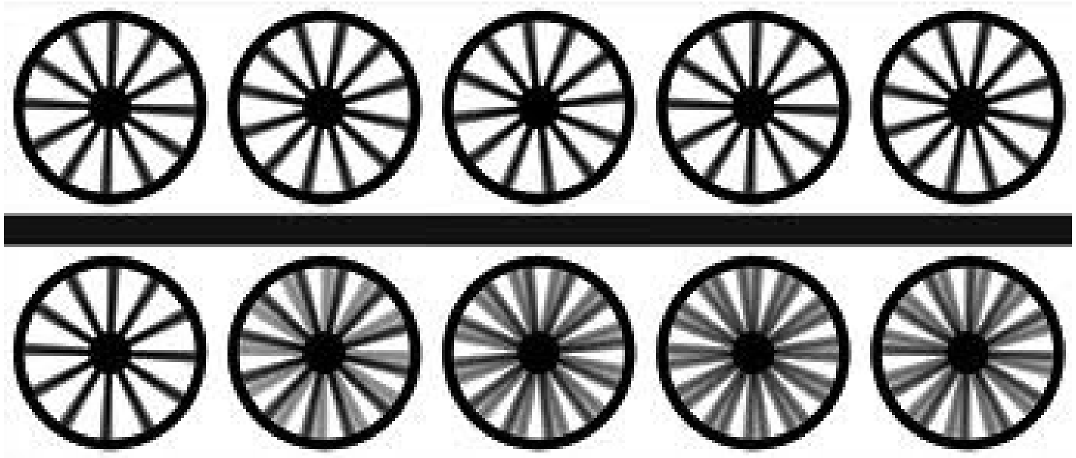
1 sample per pixel



256 spp (jittering)

Temporal alias

- Shows in slow motion animation
- Blinking pixels on contours of moving objects
  - the whole small objects can blink
- Interference of a periodic movement with a frame frequency
  - spinning wheel seems to be still or even rotating the other direction



© 2017, Tony Davis

Real world

Human visual system has no alias

Alias manifests only mildly in photography

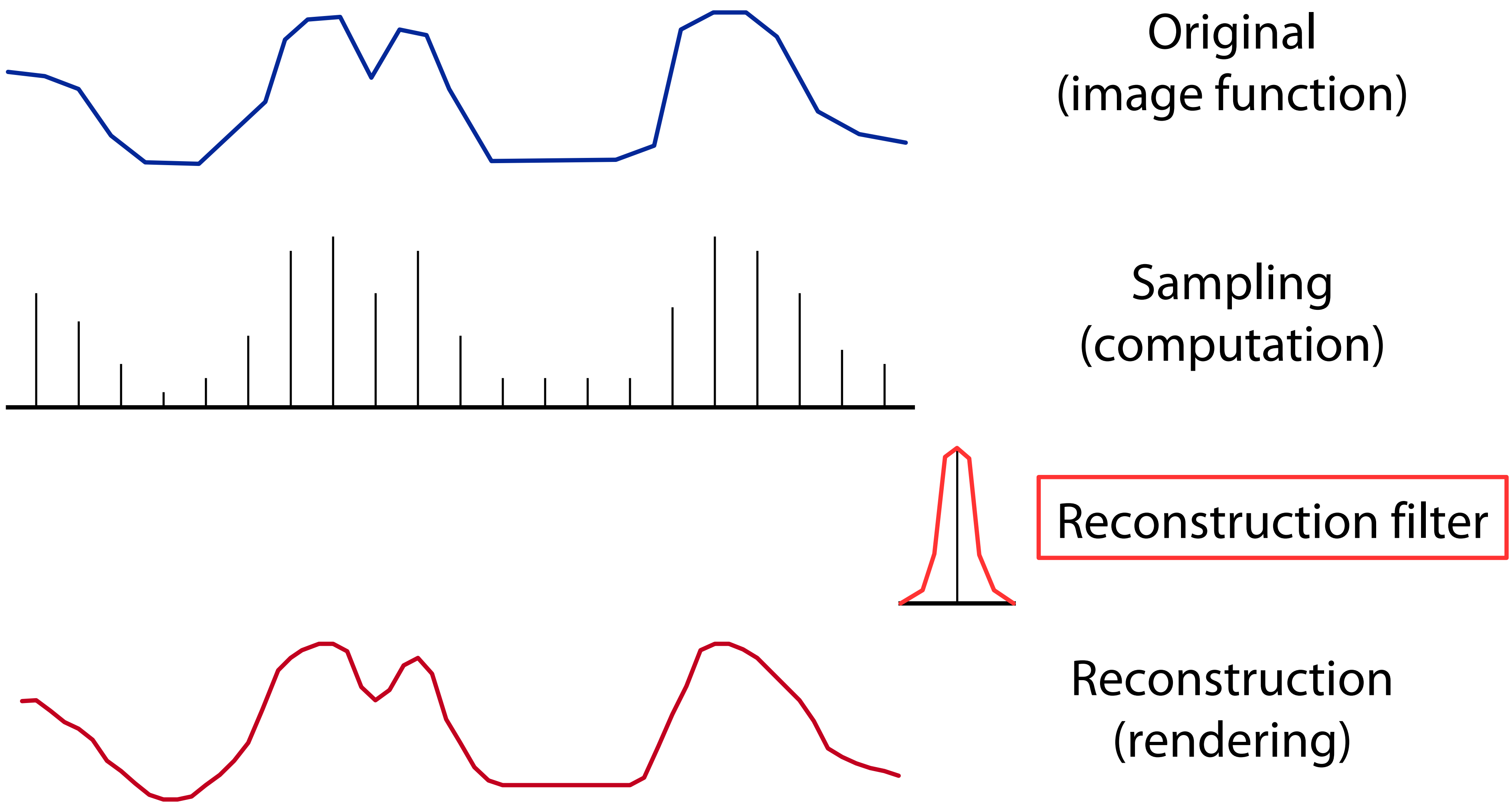
Objects smaller than **resolution of a sensor** are without details (blurry)

– fence from large distance is percieved as an average-colored area (mix of background + foreground colors)

Too fast movement generates **fuzzy** (blurred) **perception**

© 1984, Cook et al.

Reconstruction in raster context



Sampling and reconstruction

Image sampling or **computing** of image function

- higher frequencies should be reduced/removed from an image before sampling
- low pass filter (convolution – window averaging)
- image synthesis can reduce higher frequencies directly (anti-aliasing by pixel supersampling)

Reconstruction filter is defined by an output device

- e.g. neighbour CRT monitor pixels overlap
- LCD pixels behave differently (almost ideally separated)

Anti-aliasing by supersampling

Image function with continuous domain and unlimited spectrum $\mathbf{f}(\mathbf{x}, \mathbf{y})$

Anti-aliasing filter (function with limited support) $\mathbf{h}(\mathbf{x}, \mathbf{y})$

Pixel color [i,j]

$$\mathbf{I}(\mathbf{i}, \mathbf{j}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{h}(\mathbf{x} - \mathbf{i}, \mathbf{y} - \mathbf{j}) \, \mathbf{dx} \, \mathbf{dy}$$

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Simple variant

Assuming a **box smoothing filter** and **unit square pixel**

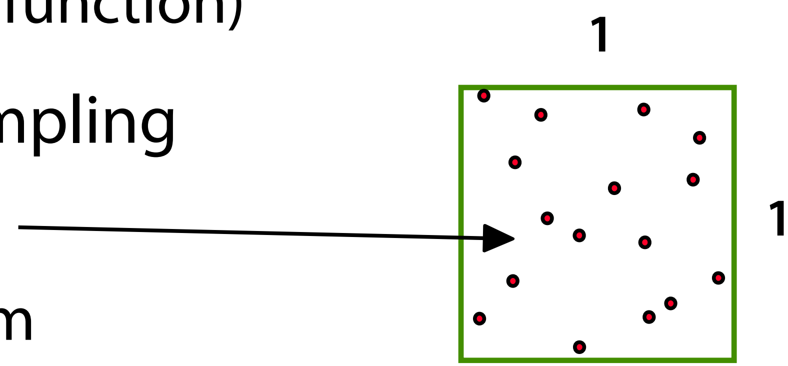
$$I(i,j) = \int_{j}^{j+1} \int_{i}^{i+1} f(x,y)\, dx\, dy$$

(integral average value of the image function on the pixel area)

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Quadrature

- ❶ **Analytic** (close form solution)
  - in rare cases (simple image function)
- ❷ **Numeric solution** using sampling
  - finite set of samples $[x_i, y_i]$
  - integral estimate by the sum



$$I(i,j) = \frac{\sum_k f(x_k, y_k) \cdot h(x_k, y_k)}{\sum_k h(x_k, y_k)}$$

  - stochastic sampling – **Monte-Carlo method**

Sampling methods

Sampling is a **mapping** $\mathbf{k} \rightarrow [\mathbf{x_k}, \mathbf{y_k}]$

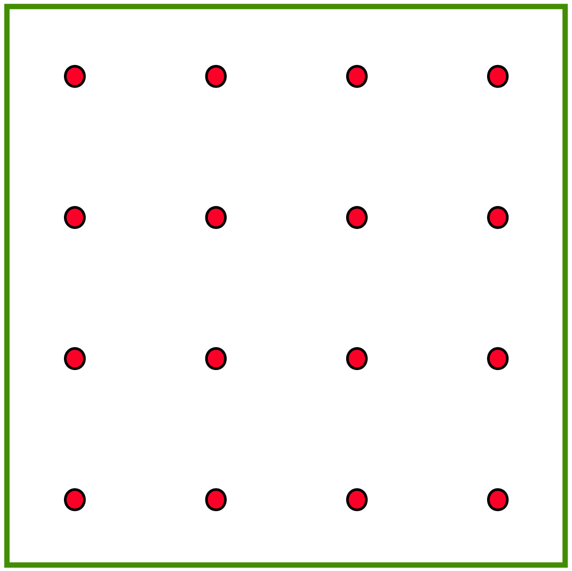
- sample is selected from the given 2D region (domain)
  - » usually rectangular, square or circular
- sampling in higher dimensions (e.g. dim > 8)

Required **properties** of sampling algorithms

- uniform probability over a domain
- high regularity is not desirable (interference)
- efficient computation

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Uniform sampling

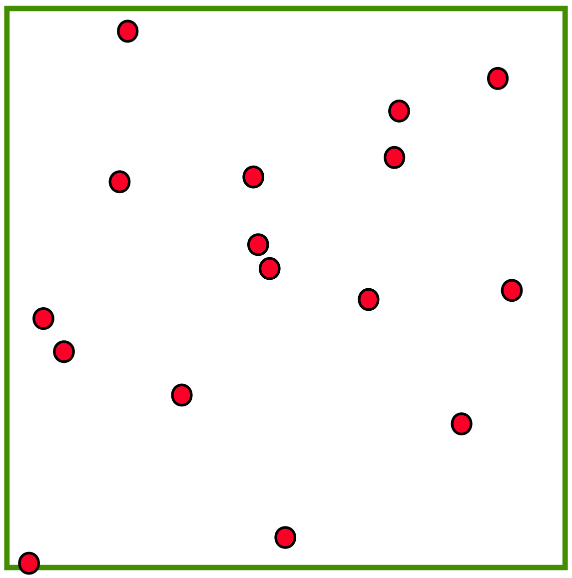


Does not reduce **Moire / interference**

(interference still appears in higher frequencies)

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Random sampling



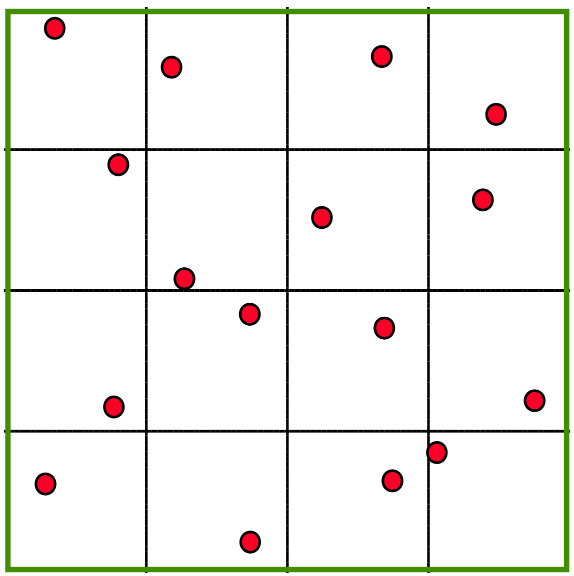
N **independent** random samples with uniform probability density (PDF)

Samples tend to form **clusters**

Too much noise in a result image

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Jittering



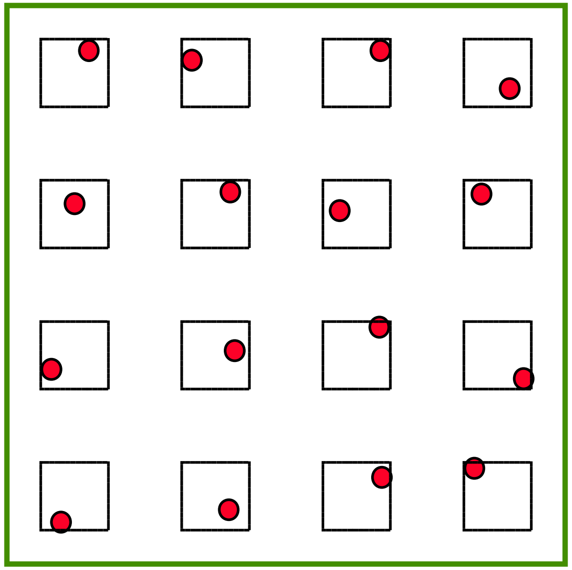
K × K **independent** random samples in K × K **equally sized** sub-regions covering the original domain completely

Big clusters are not possible

More regular coverage of a domain

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Partial jittering, “semi-jittering”

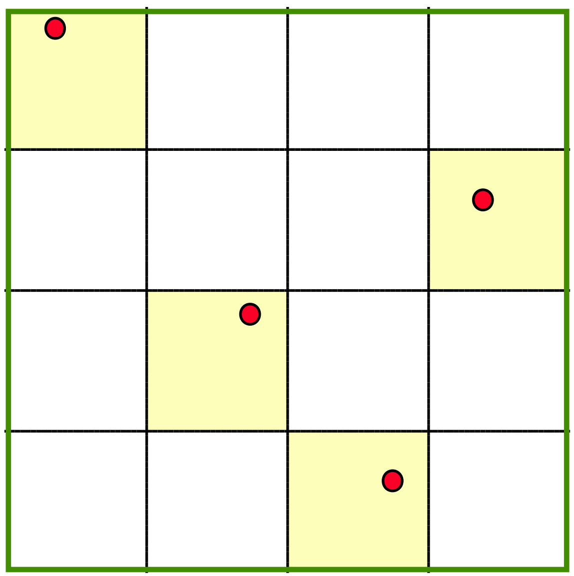


K × K **independent** random samples in K × K **equally sized** sub-regions **not covering** the original domain

Clusters are impossible but **too** regular (interference)

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

“N rooks” (“uncorrelated jitter”)



“Low-cost jittering”

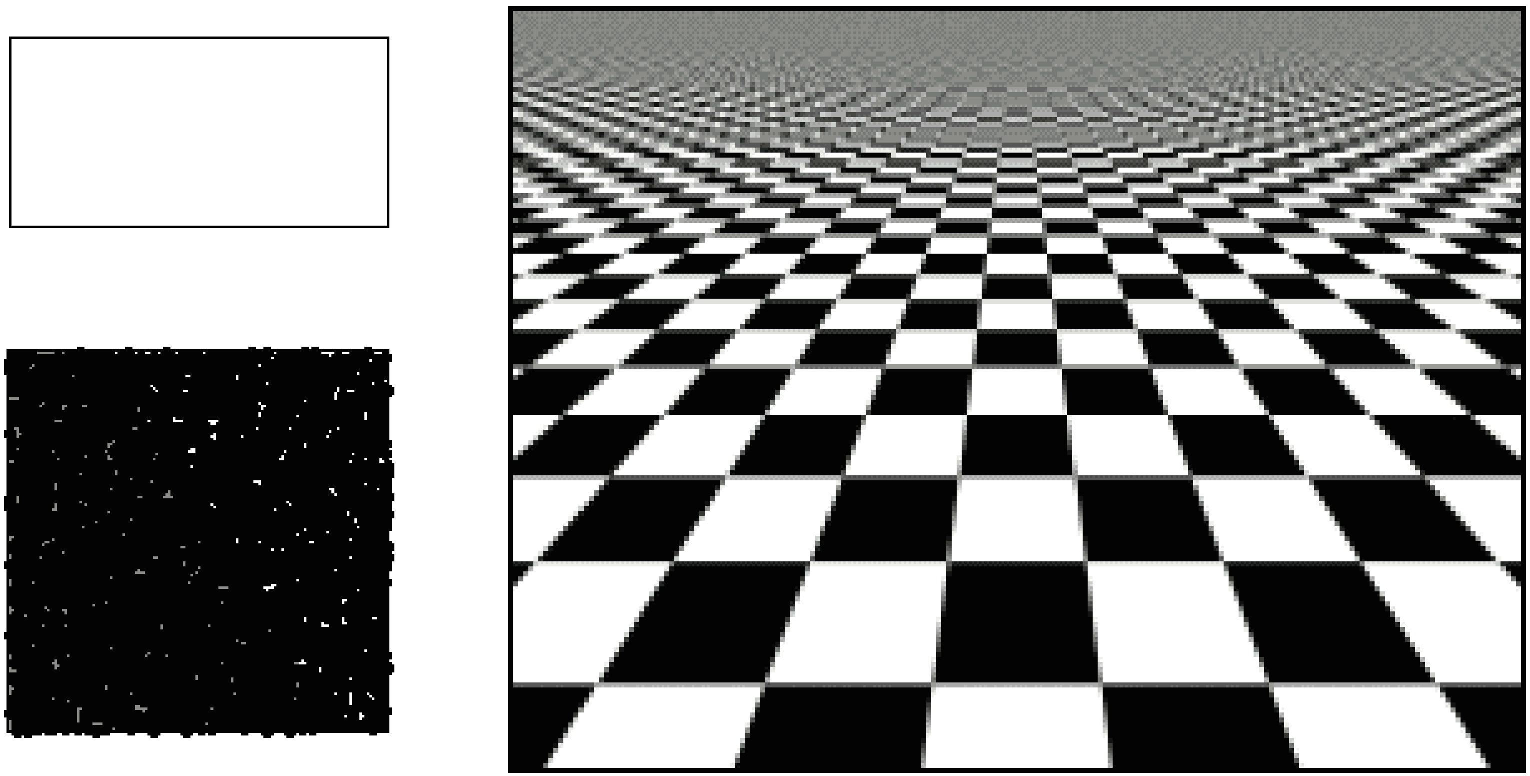
there is exactly one sample in each row and in each column

Random permutation of a diagonal

Good properties of “jittering” are preserved

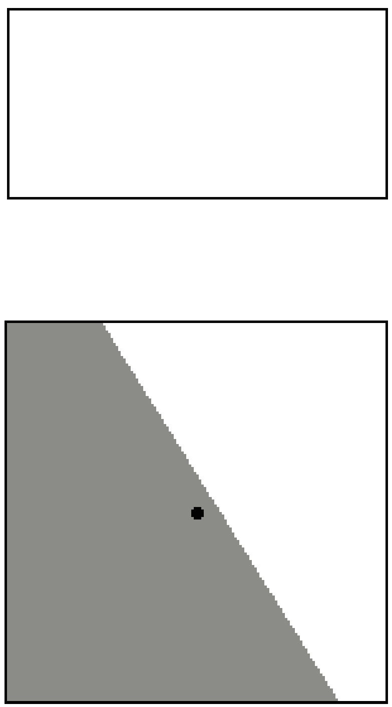
Higher **efficiency** (especially in high dimension D > 5)

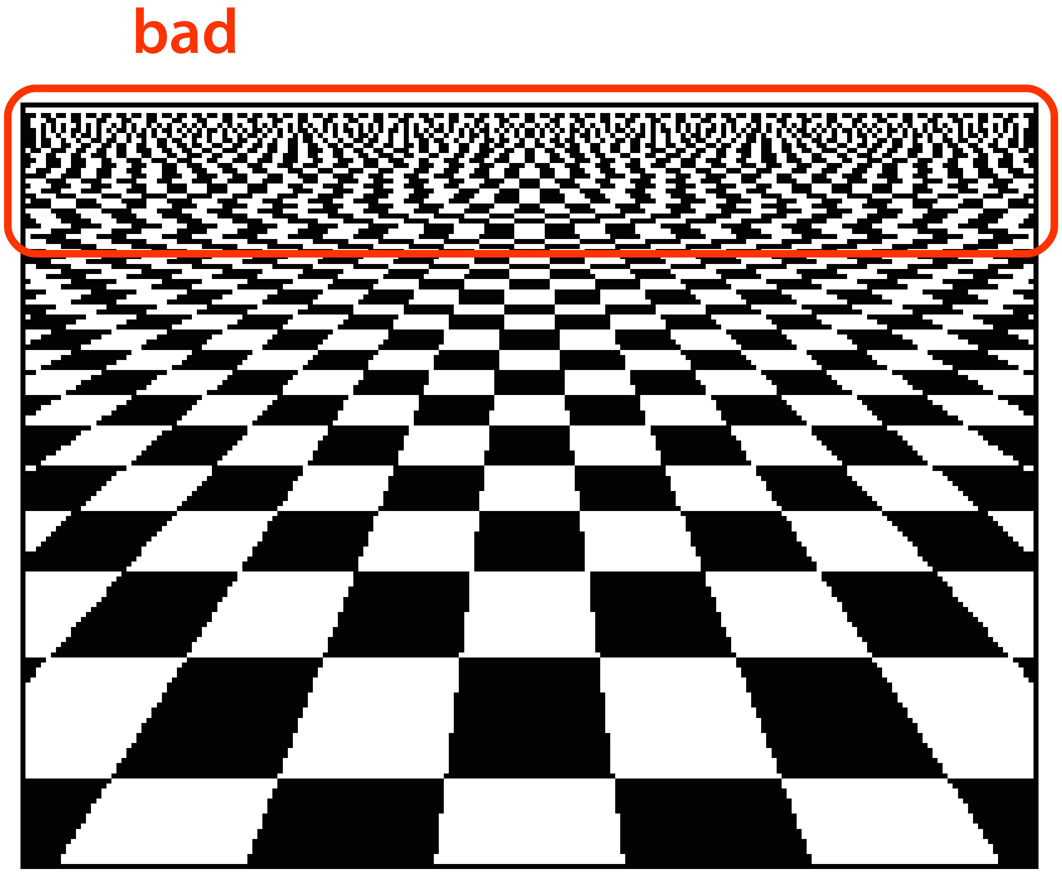
Example – converged



2500 spp (samples per pixel)

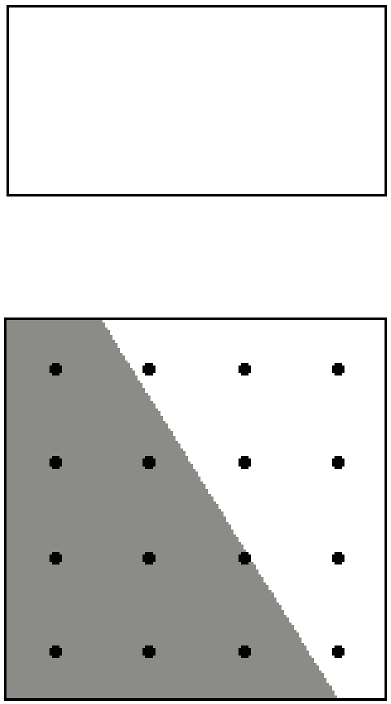
Example – 1spp

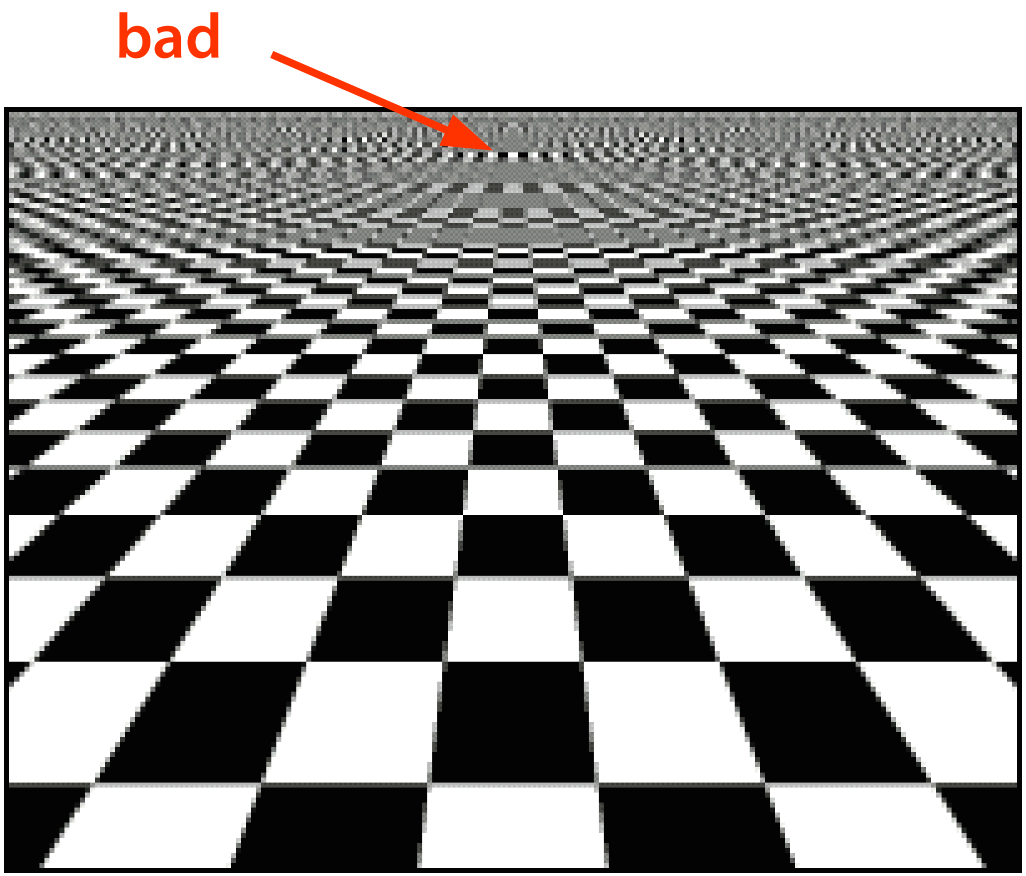




1 spp

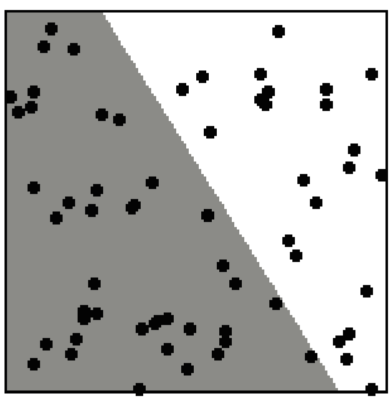
Example – regular

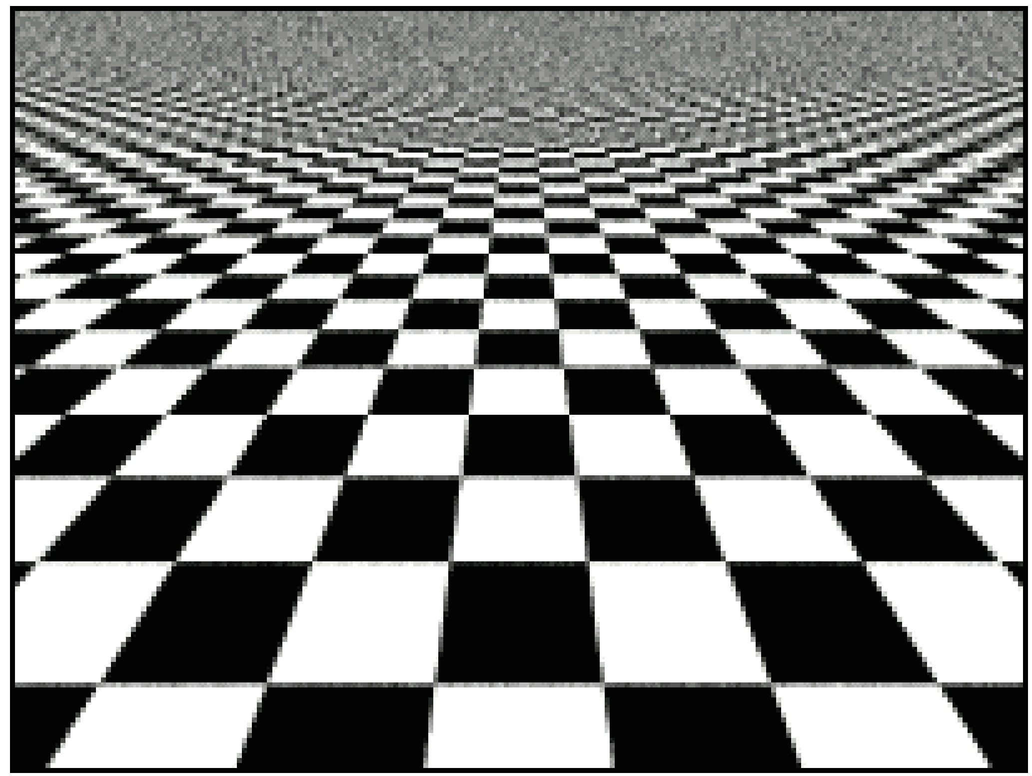




16 spp

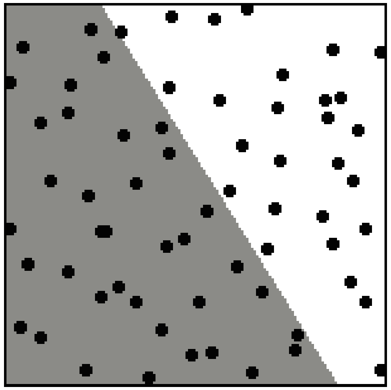
Example – random

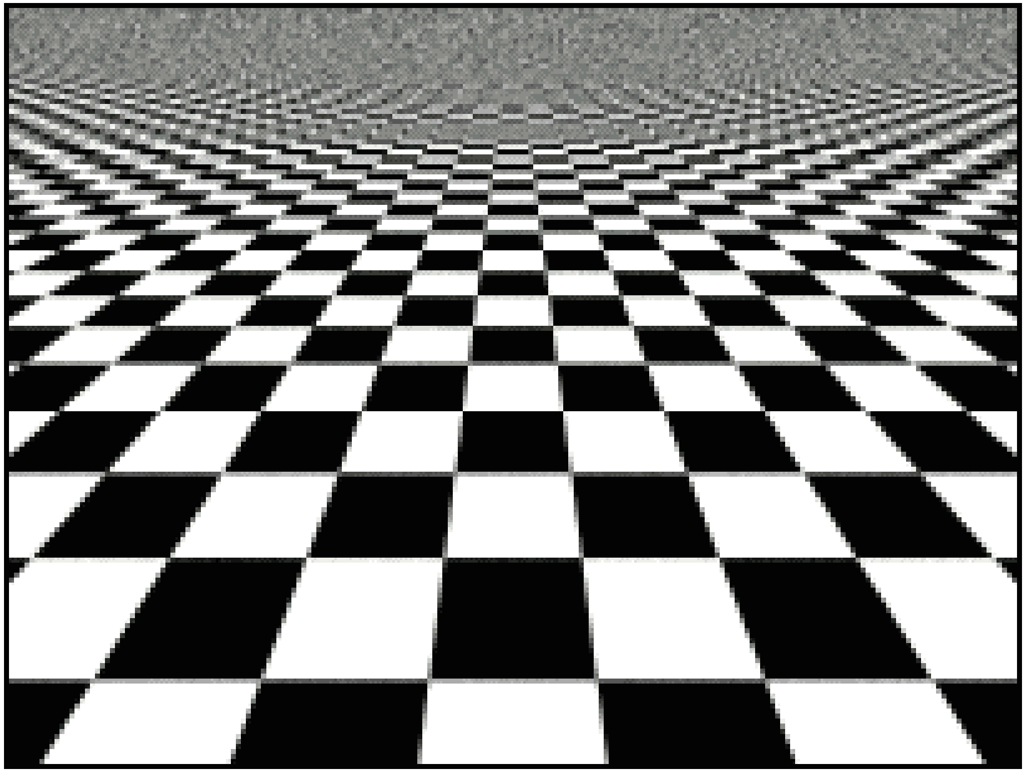




64 spp

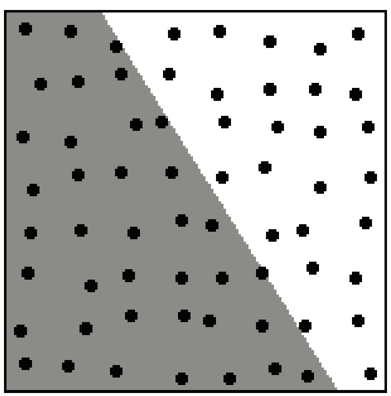
Example – jittering





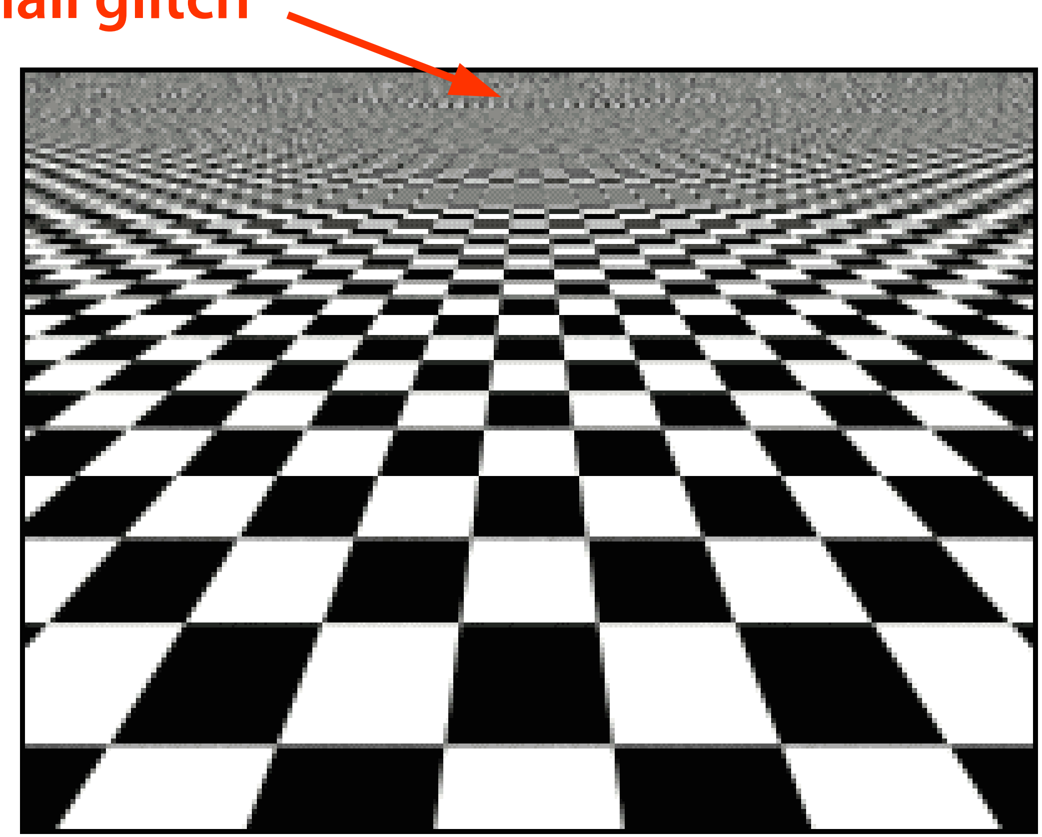
64 spp

Example – semi-jittering



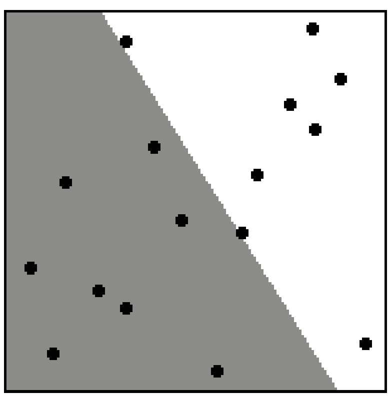
64 spp

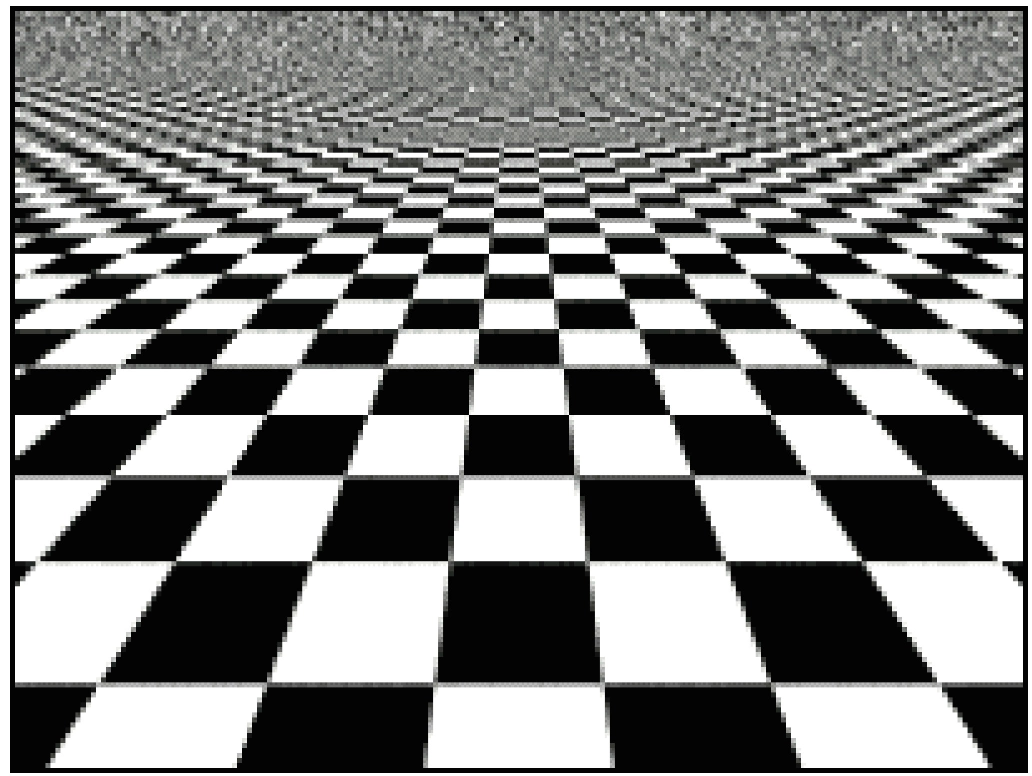
small glitch



© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

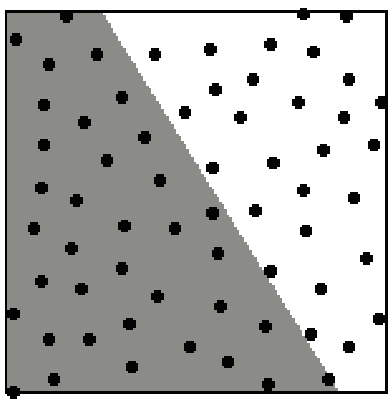
Example – N rooks

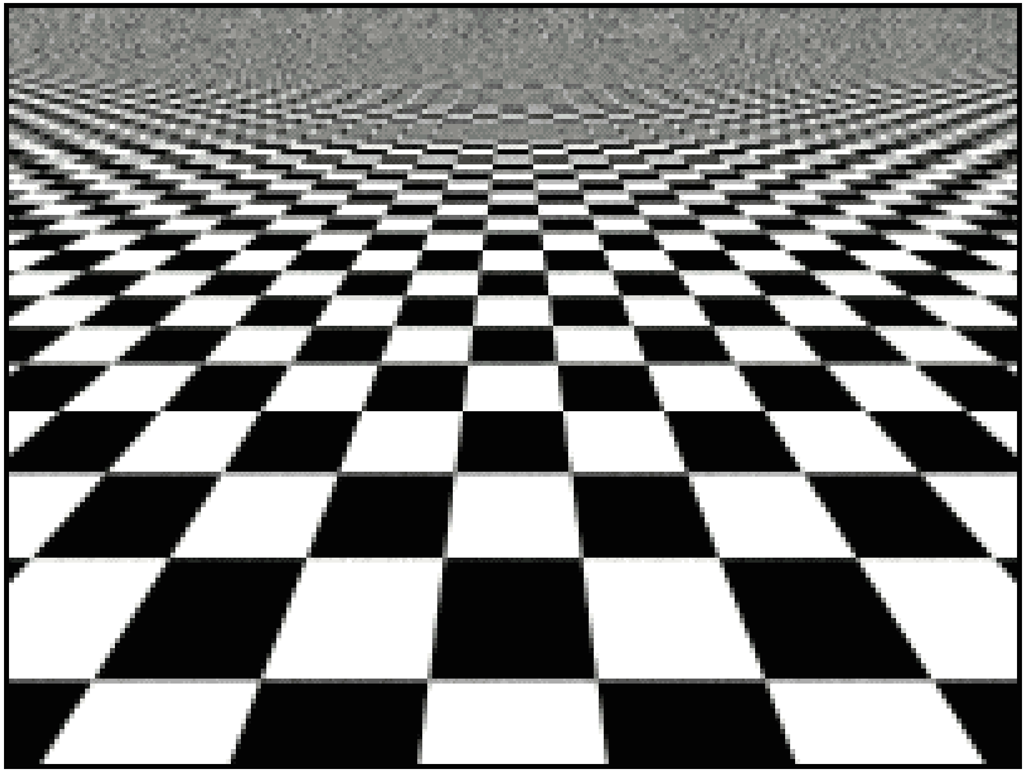




16 spp

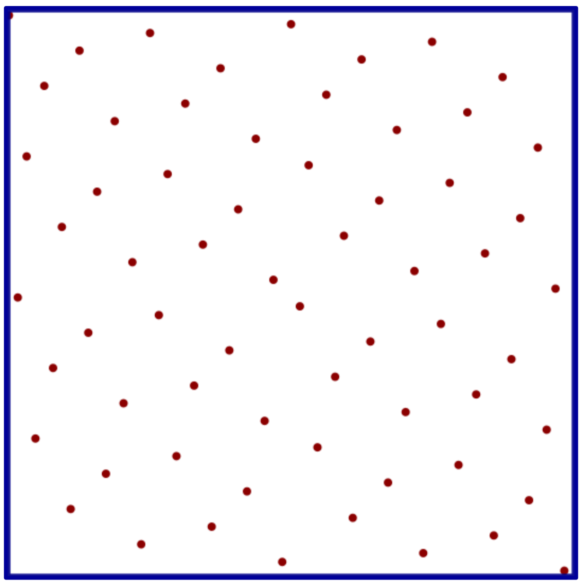
Example – Poisson disk





64 spp

Hammersley



+ excellent discrepancy

+ deterministic

+ very fast

- difficult adaptive refinement
- bad spatial spectrum

Famous **Halton sequence** is based on similar principles..

Deterministic sequences

Based on similar principles

- Halton, Hammersley, Larcher-Pillichshammer

For a prime number **b** let **n** be positive integer expressed using b-representation

$$n = \sum_{k=0}^{L-1} d_k(n)\, b^k$$

then there is an unique number from **[0,1) range**

$$g_b(n) = \sum_{k=0}^{L-1} d_k(n)\, b^{-k-1}$$

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Halton, Hammersley

Famous **Halton** sequence (e.g. $b_1 = 2, b_2 = 3$)

$$x(n) = \left[g_{b_1}(n), g_{b_2}(n) \right]$$

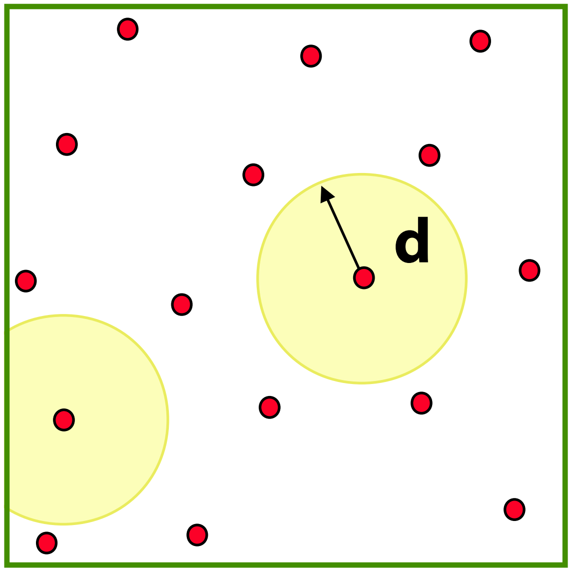
Hammersley sequence (e.g. $b = 2$)

$$x(n) = \left[\frac{n}{N}, g_b(n) \right]$$

Larcher-Pillichshammer sequence uses **XOR** operation instead of addition (inside the $g_b(n)$)…

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Poisson disk sampling



N random samples

meeting condition

$\| [x_k,y_k] - [x_l,y_l] \| > d$

for given value **d**

Prevents creating **clusters**, imitates **distribution of light-perception cells** in retina of mammals

Difficult efficient **implementation**!

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Implementation

Rejection sampling

- candidate sample is rejected if too close to any previous accepted sample
- less efficient for higher number of samples

Choice of **value d is problematic**

- maximum number of placeable samples depends on **d**

Difficult **adaptive refinement**

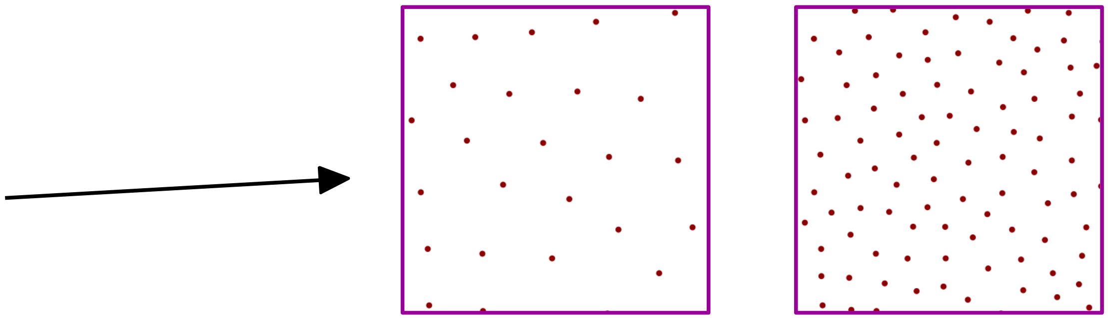
- additional samples to a existing set of samples

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Mitchell's ("best candidate") algorithm

Generates **gradually refined sample set** (from Poisson sampling)

- no problems with **d**
- intrinsic **refinement**



Compute-intensive algorithm

- sample set can be **precomputed and reused**
- to **reduce dependency** between neighbour pixels random rotation and translation can be used

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Mitchell's algorithm

❶ The 1^{st} **sample** is chosen randomly

❷ Choice of the $(\mathbf{k+1})^{th}$ **sample**

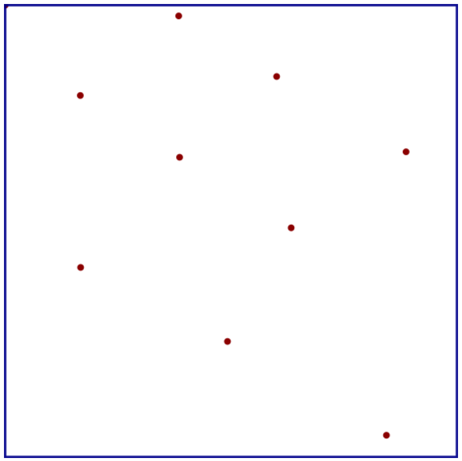
- generate $\mathbf{k \cdot q}$ independent candidates (**q** determines sample-set quality)
- the **most distant candidate** (from all previous **k** accepted samples) is selected and accepted

For higher **q** we get better quality set

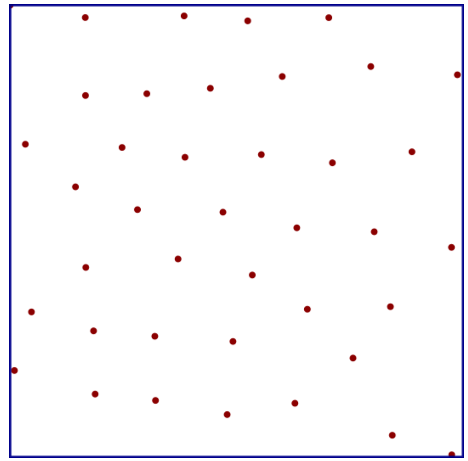
- choose $\mathbf{q} > \mathbf{10}$ in demanding situations

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

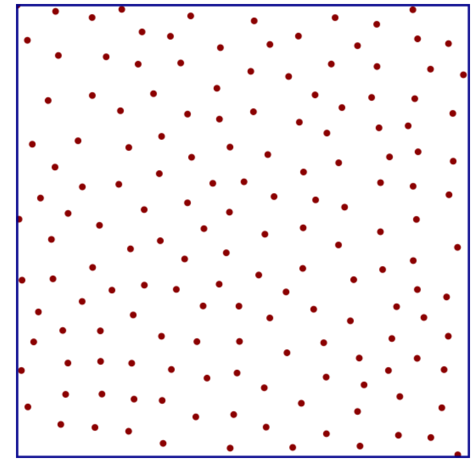
Incremental example



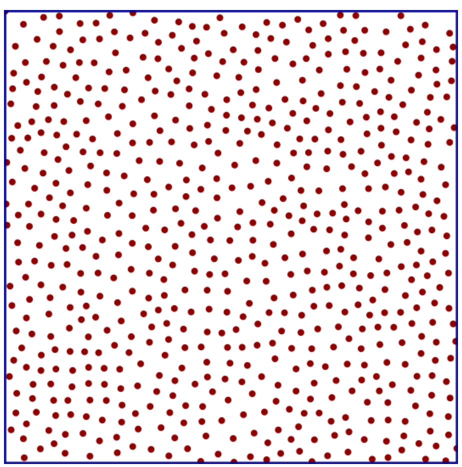
10



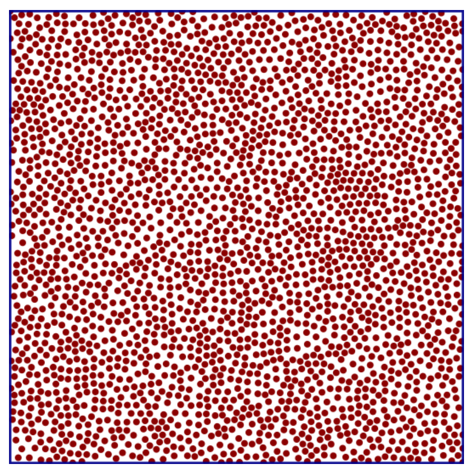
40



160



640



2560

"Quality coefficient"
k = 10

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Adaptive refinement

Sampling based on **local importance** (importance sampling) or **interest**

- some regions have higher weight (higher probability)
- regions with **higher variance** should be sampled more densely

"Importance" or "interest" **need not be known in advance** (explicitly)

- algorithm has to adopt to intermediate results (adaptability)

 © Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Modification of static methods

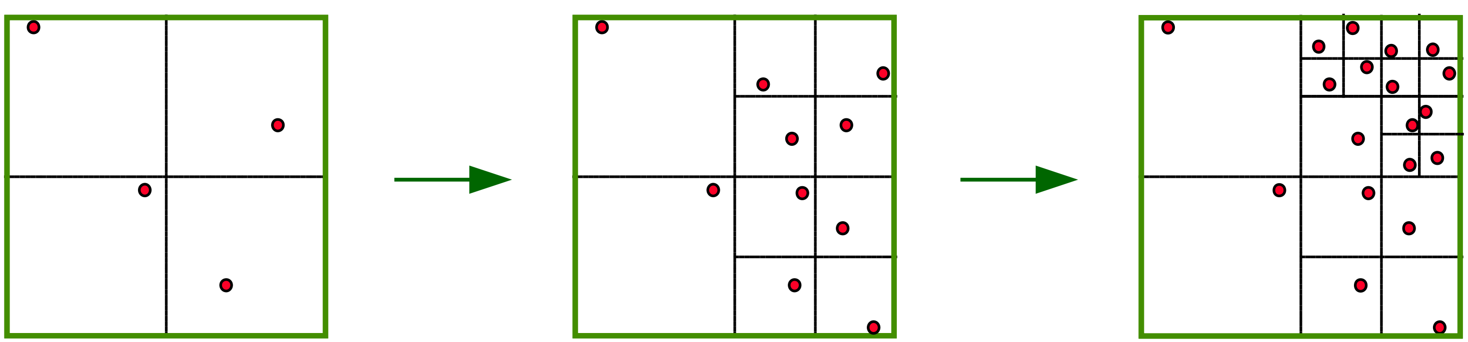
❶ **Initial phase**

- compute small set of test samples (1 to 5)
- define **refinement criterion** based on previous samples

❷ **Refinement phase**

- sampling is refined in regions of higher need (criterion)
- efficiency – we should reuse all generated samples!

Almost every sampling can be **reformulated** in that way



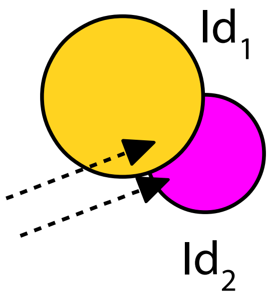
Refinement criteria

Function values (difference, variance, gradient)

- difference between neighbour samples…

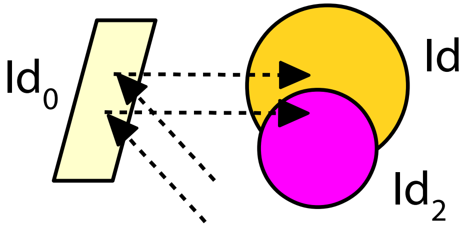
Id's of hit solids (Ray-tracing specific)

- higher priority
- textures with repeated patterns – use of **signatures**

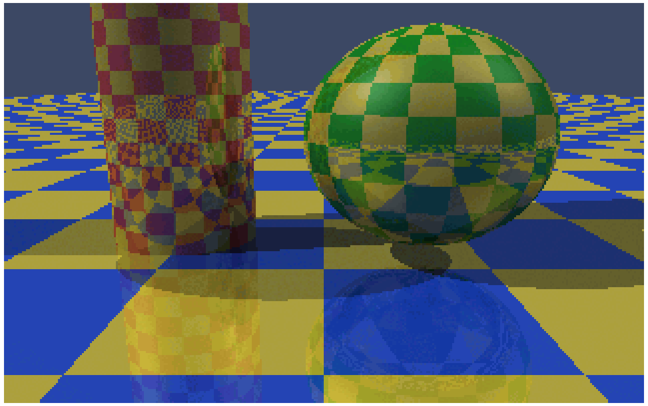


Trace tree (recursive ray-tracing)

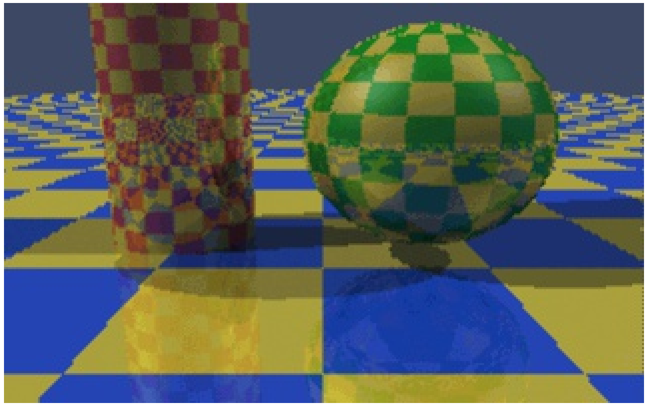
- topologic comparison of complete or limited trace trees
- **tree identifier** – recursive hash function using solid Id's, texture signatures, shadow / light…



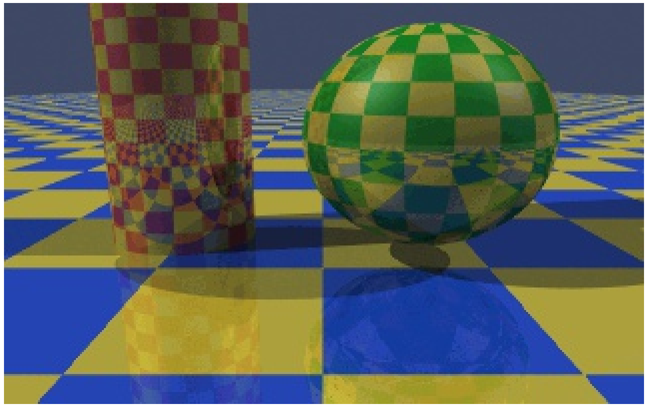
Adaptive resampling example



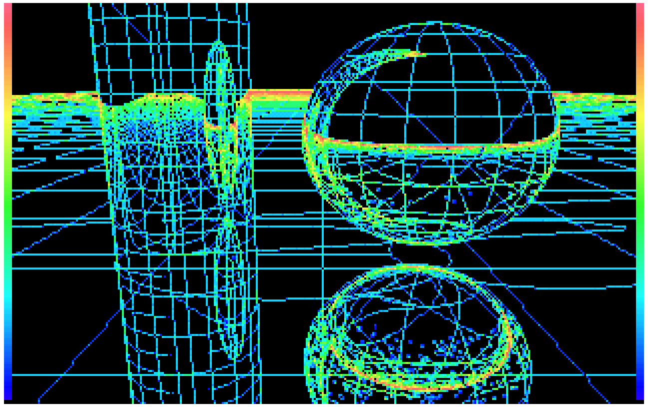
1 spp



1/2 spp



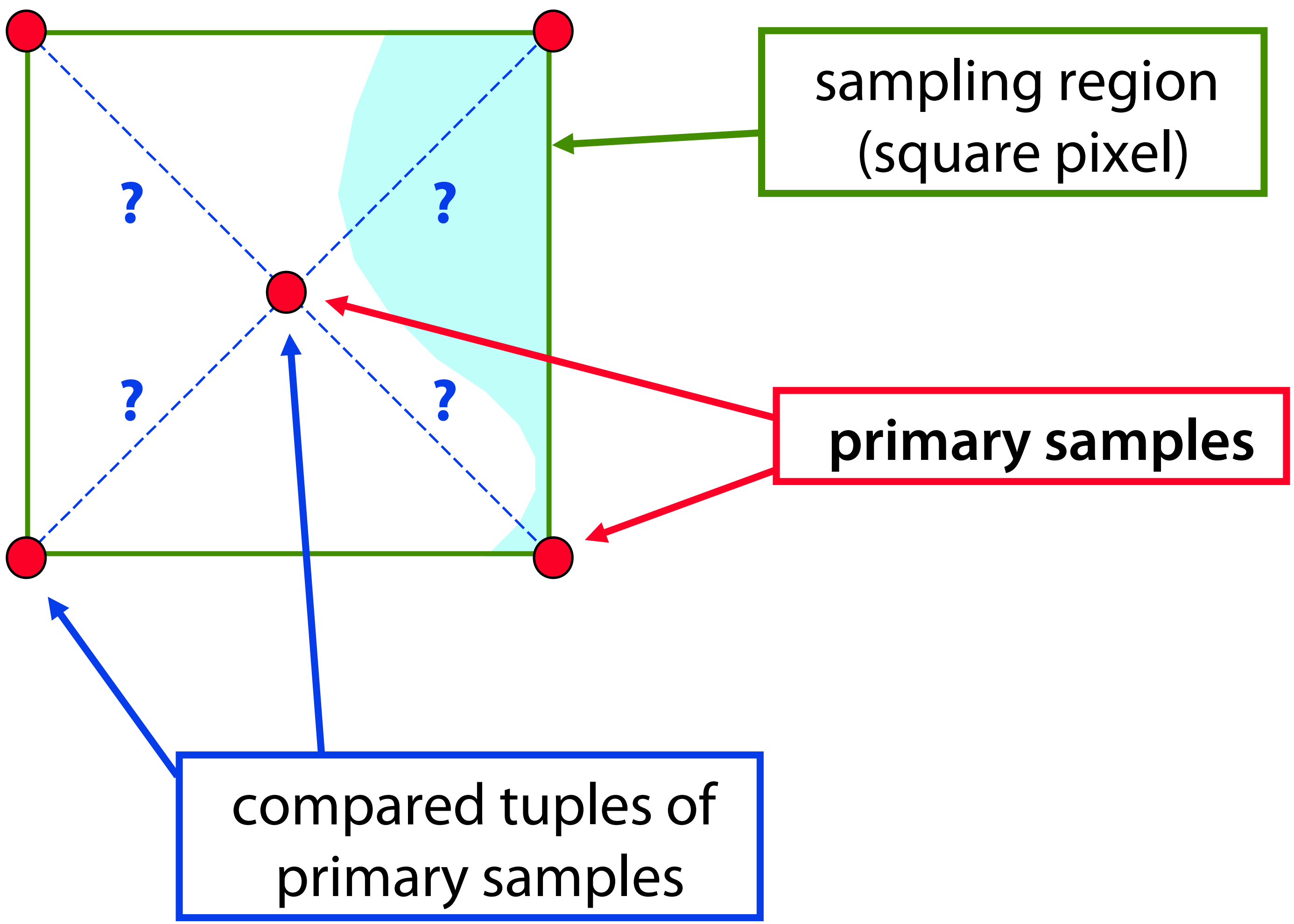
adaptive



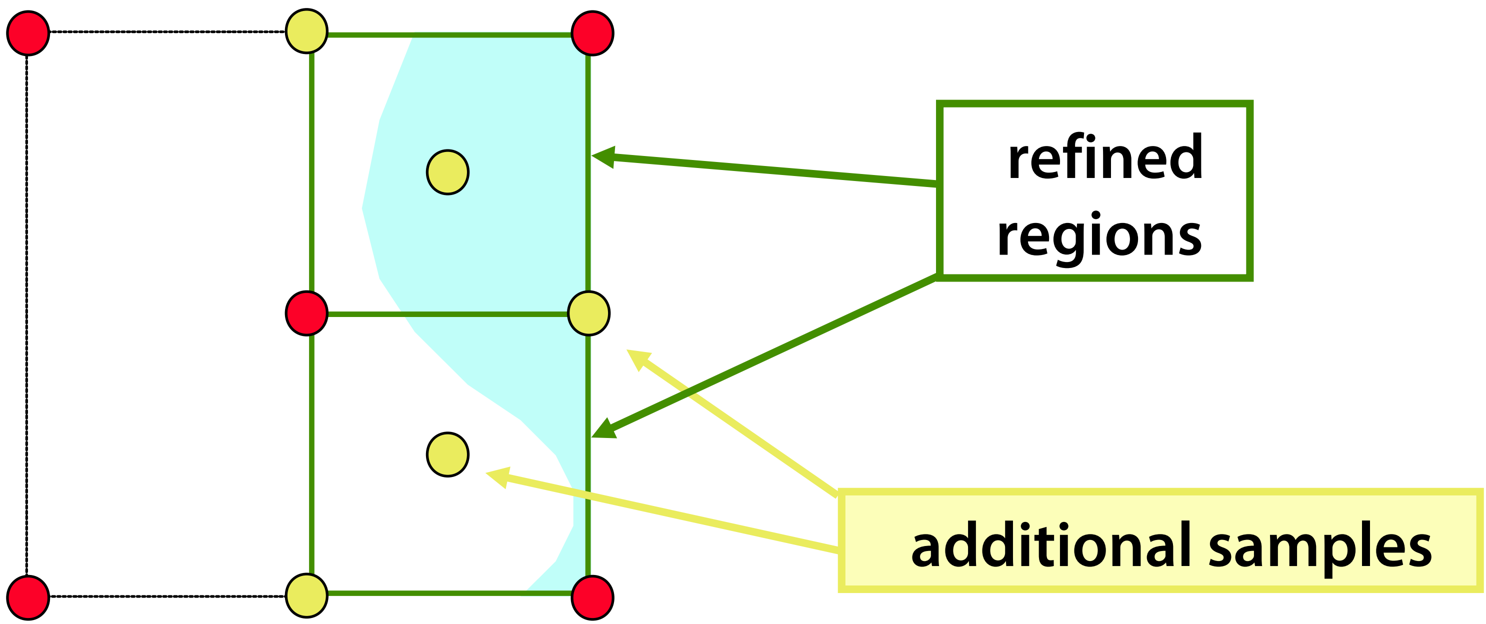
refinement map

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca

Recursive refinement (Whitted)

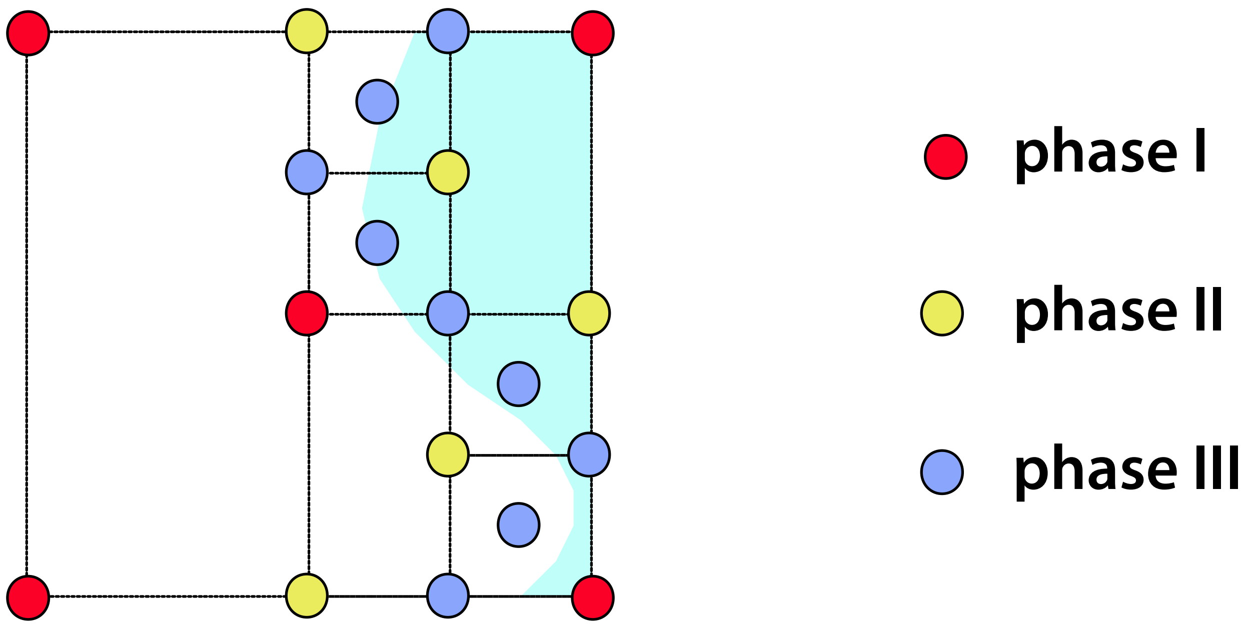


Refinement phase



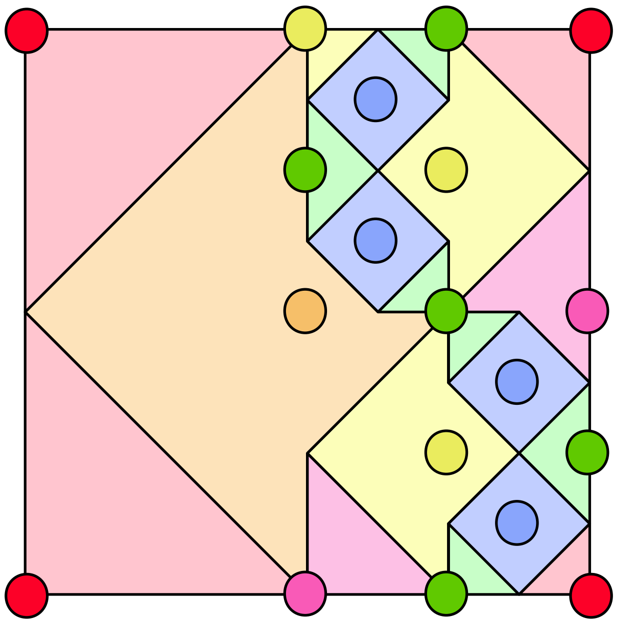
The same procedure is executed in refined regions **recursively** (up to the declared maximum level)

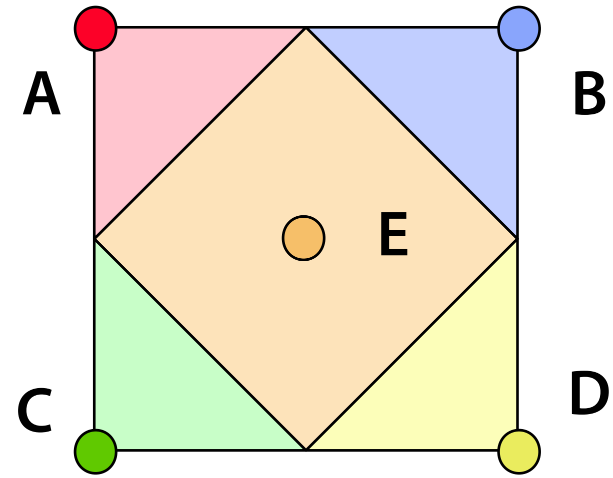
Result sample set



Evaluated: 5 + 5 + 9 = **19 samples**
(from total number of **41**)

Pixel value reconstruction





$$\frac{1}{2}E + \frac{1}{8}\left[A + B + C + D\right]$$

If the refinement stops in a specific square,
its area is split to two triangles (diagonal samples)

Literature

A. Glassner: ***An Introduction to Ray Tracing***, Academic Press, London 1989, 161-171

A. Glassner: ***Principles of Digital Image Synthesis***, Morgan Kaufmann, 1995, 299-540

J. Pelikán: ***Náhodné rozmisťování bodů v rovině*** (Random point placement), CSGG 2014, slides & paper available online

© Josef Pelikán, https://cgg.mff.cuni.cz/~pepca