Anti-aliasing and sampling

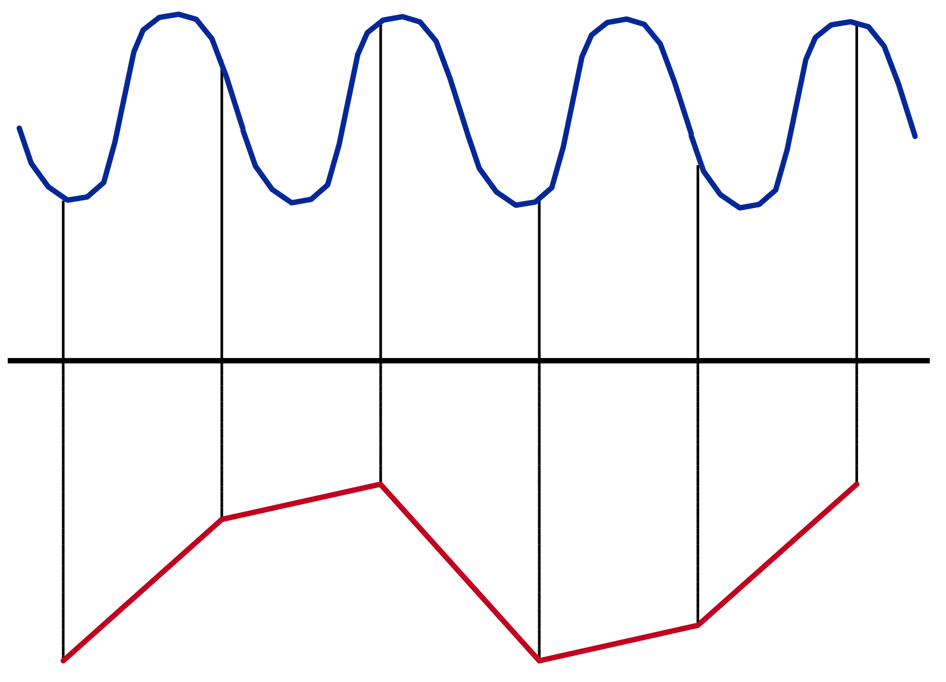
CGG MFF UK Praha

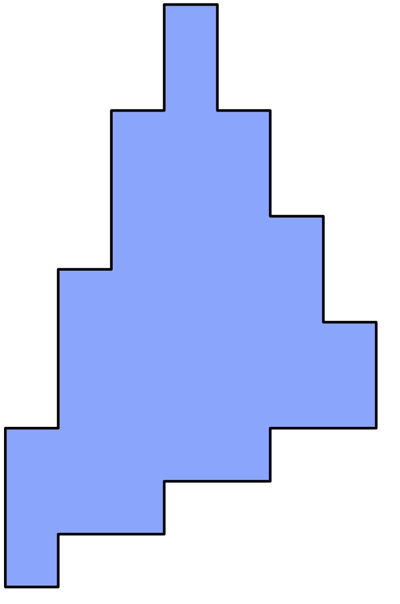
pepca@cgg.mff.cuni.cz

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

Spatial and temporal alias

Artefacts caused by a regular-grid
or regular sampling

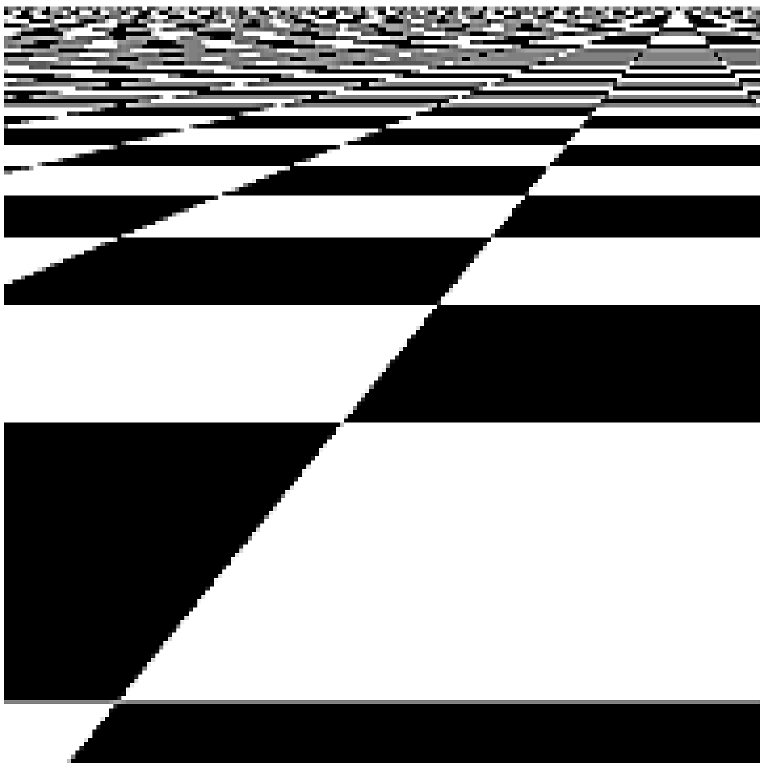




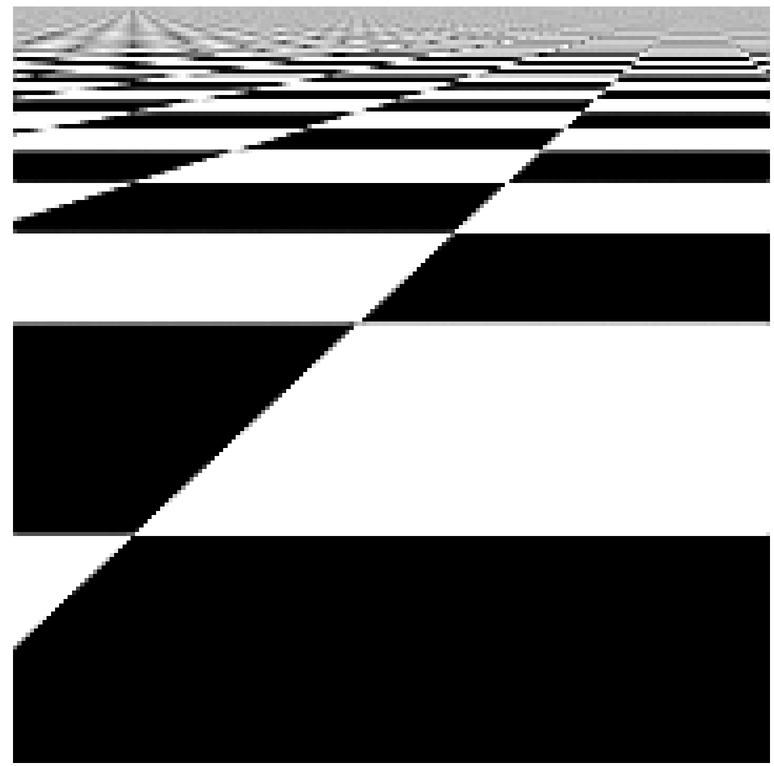
Spatial alias

- **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

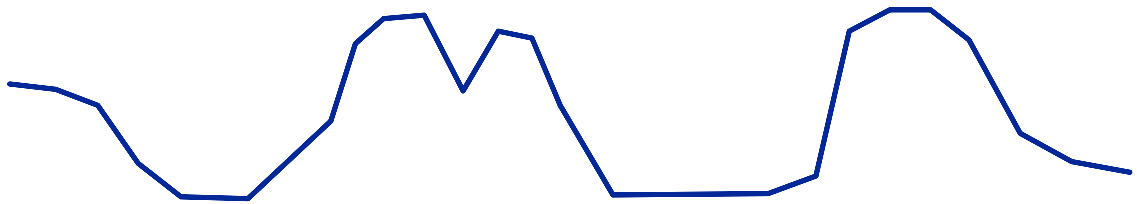
Real world

Human visual system has no alias.

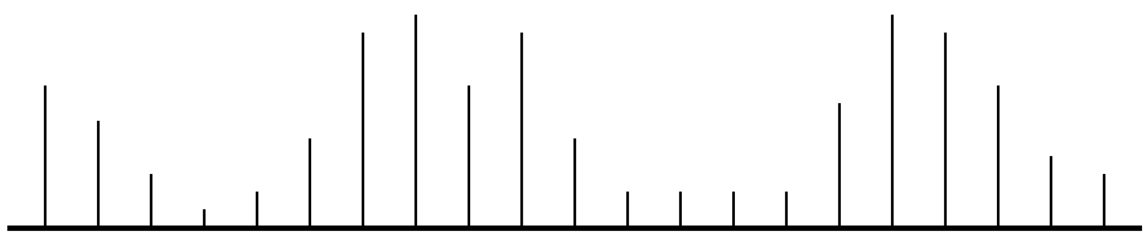
Alias manifests only mildly in photography.

- objects smaller than **resolution of a sensor** are without details / blurry
  - fence from large distance is percieved as an average-color area (mix of background + foreground colors)
- too fast movement generates **fuzzy** (blurred) **perception**

Reconstruction in raster context

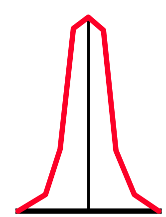


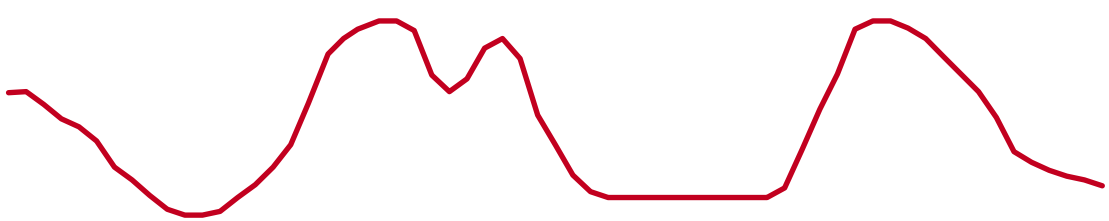
original
(image function)



sampling
(computation)

reconstruction filter:





reconstruction
(rendering)

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

Anti–aliasing by supersampling

Image function with continuous domain and unlimited spectrum: $f(x, y)$

Anti-aliasing filter (function with limited support)**:** $h(x, y)$

Pixel color [i,j]:

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

Simple variant

Assuming **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)

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

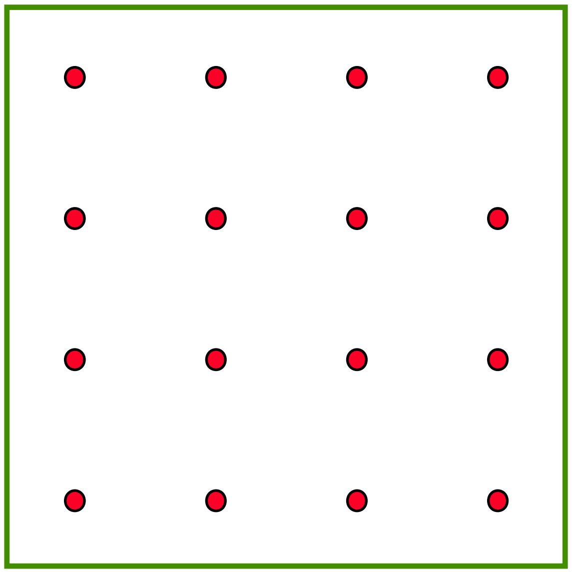
$$I(i,j) = \frac{\sum_k f(x_k, y_k) \cdot h(x_k, y_k)}{\sum_k h(x_k, y_k)}$$

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

Sampling methods

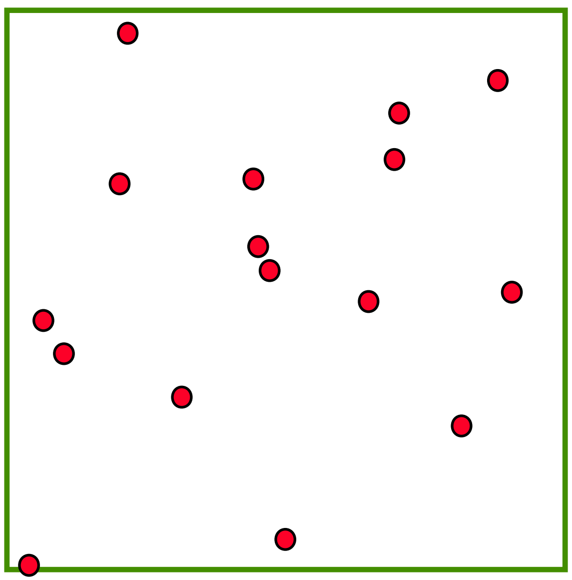
- **mapping:** $k \rightarrow [\, x_k, y_k \,]$
  - sample is selected from the given 2D region (domain): usually rectangular, square or circular
  - sampling in higher dimensions (dim=10)
- required **properties** of sampling algorithms
  - uniform probability over a domain
  - high regularity is not desirable (interference)
  - efficient computation

Uniform sampling



Does not reduce **Moire / interference**
(interference still appears in higher frequencies)

Random sampling

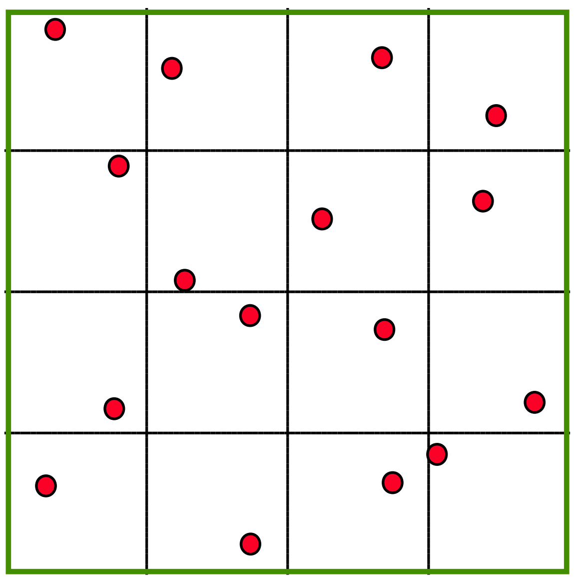


N independent random samples with uniform probability density (PDF)

Samples tend to form **clusters**

Too much noise in a result image

“Jittering”

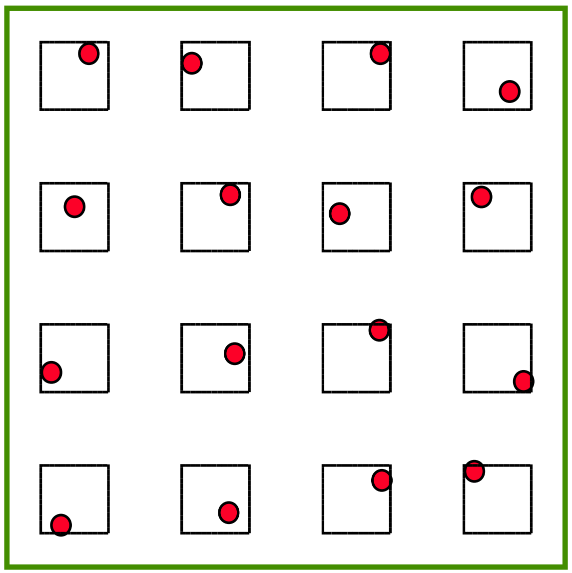


K × K independent random samples in K × K equally sized sub-regions covering the original domain completely

Big clusters are not possible

More regular coverage of a domain

Partial "jittering", "semi–jittering"

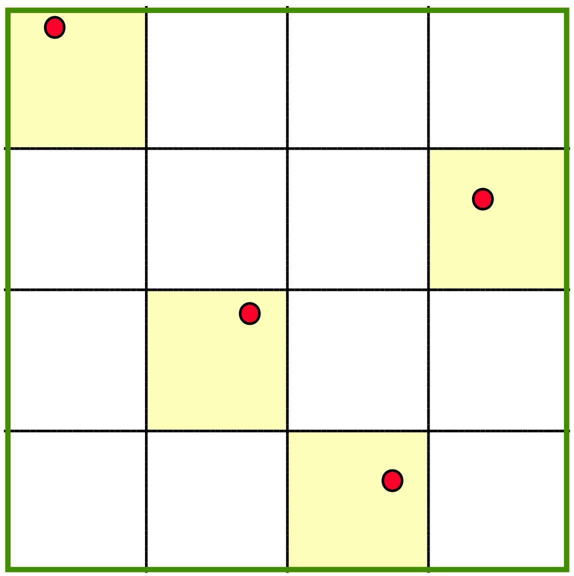


K × K independent random samples in K × K equally sized sub-regions not covering the original domain

Clusters are impossible

Too regular (interference)

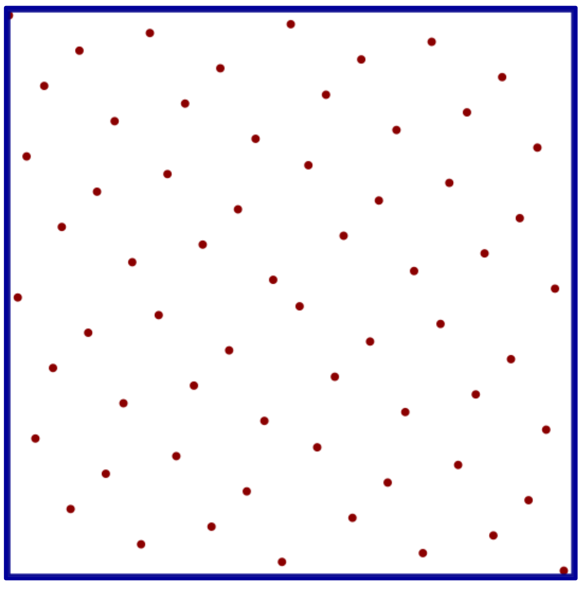
"N rooks" ("uncorrelated jitter")



"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 dimensions)

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 a number from [0,1) range:

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

Halton, Hammersley

- famous Halton sequence (e.g. $b_1=2$, $b_2=3$):

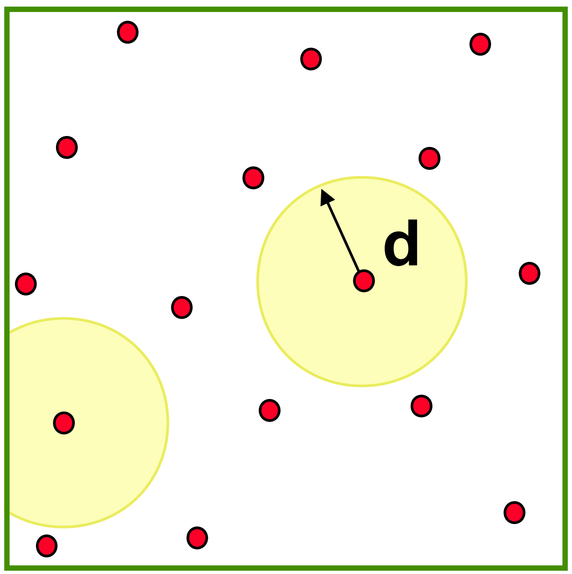
$$x(n) = \left[g_{b_1}(n), g_{b_2}(n) \right]$$

- 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 $g_b(n)$)..

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 a mammal.
Difficult efficient **implementation**!

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

Mitchell's algorithm

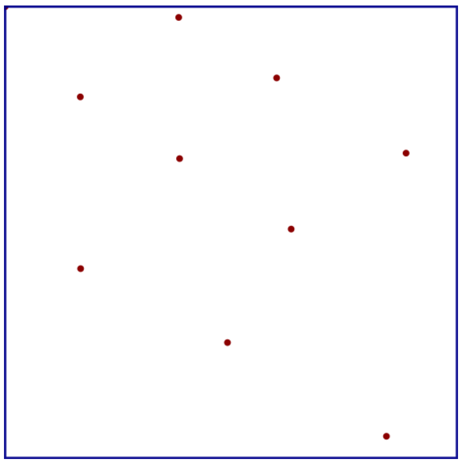
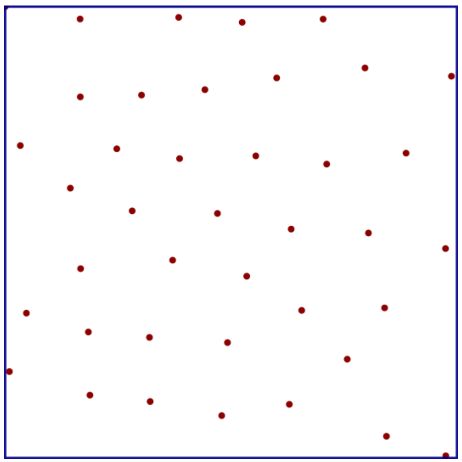
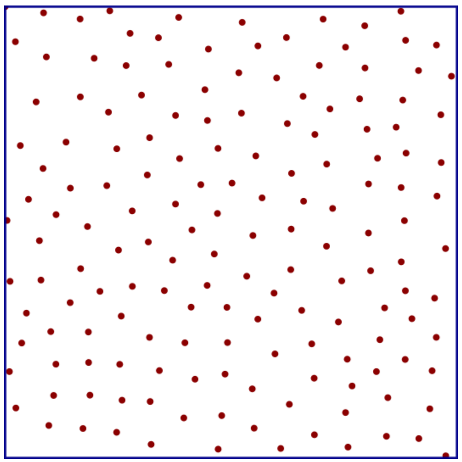
("best candidate" algorithm)

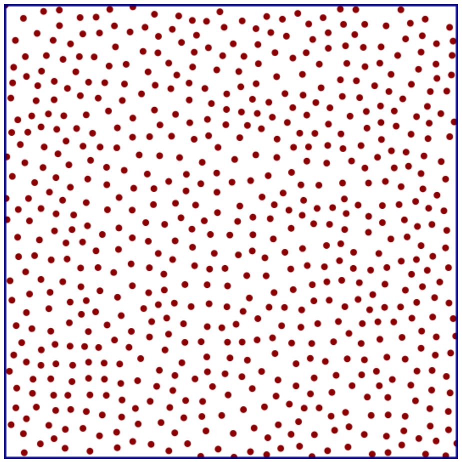
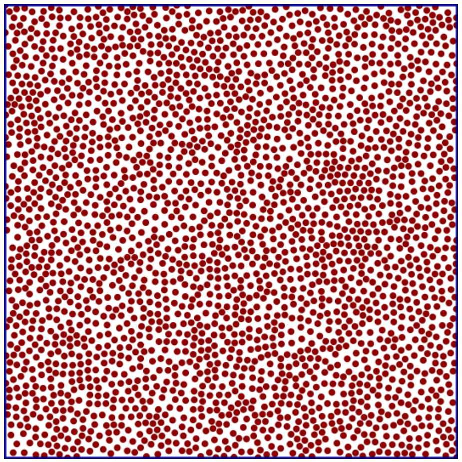
- generates **gradually refined sample set** (from Poisson sampling)
  - no problems with **d**
  - intrinsic refinement
- **compute-intensive** algorithm
  - sample set can be **precomputed and reused**
  - to **reduce dependency** between neighbour pixels random rotation and translation can be used

Mitchell's algorithm

- ❶ the **1st sample** is chosen randomly
- ❷ choice of the **(k+1)th sample**:
  - generate **k· 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 **q > 10** in demanding situations

Incremental example

Sample numbers:

10, 40, 160,
640, 2560

K = 10

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)

Modification of static methods

❶ **initial phase**:
- compute small set of test samples (1-5)
- define **refinement criterion** based on previous samples

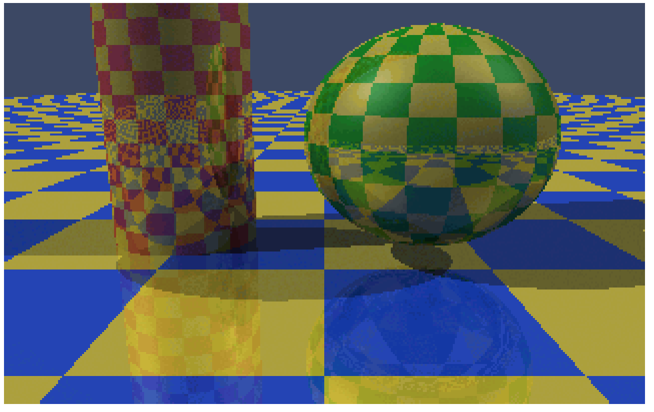
❷ **refinement phase**:
- sampling is refined in regions of higher need (criterion)
- efficiency: if we can reuse all generated samples

➡ almost every sampling algorithm can be **reformulated** in that way

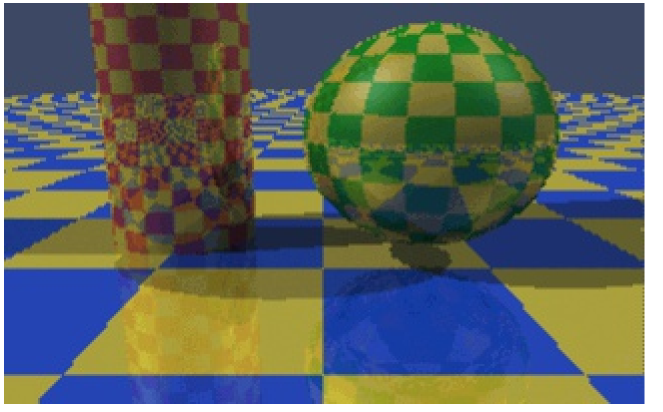
Refinement criteria

- **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 <u>signatures</u>
- **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, ..

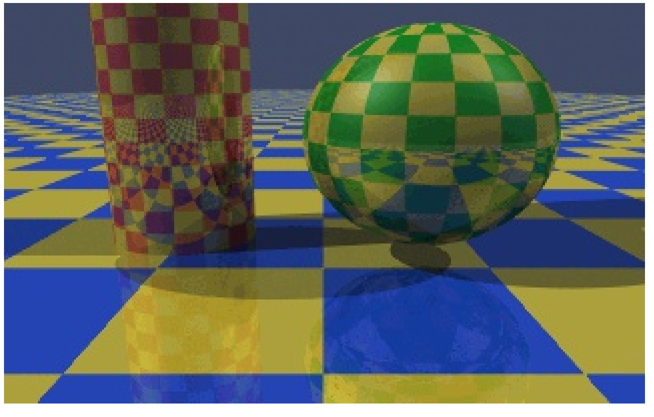
Adaptive resampling example



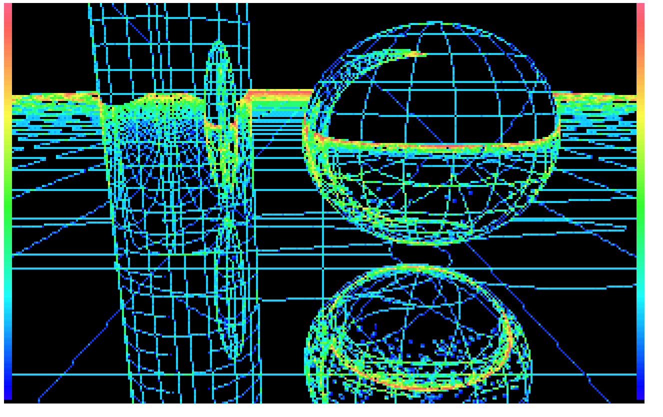
1 spp



1/2 spp

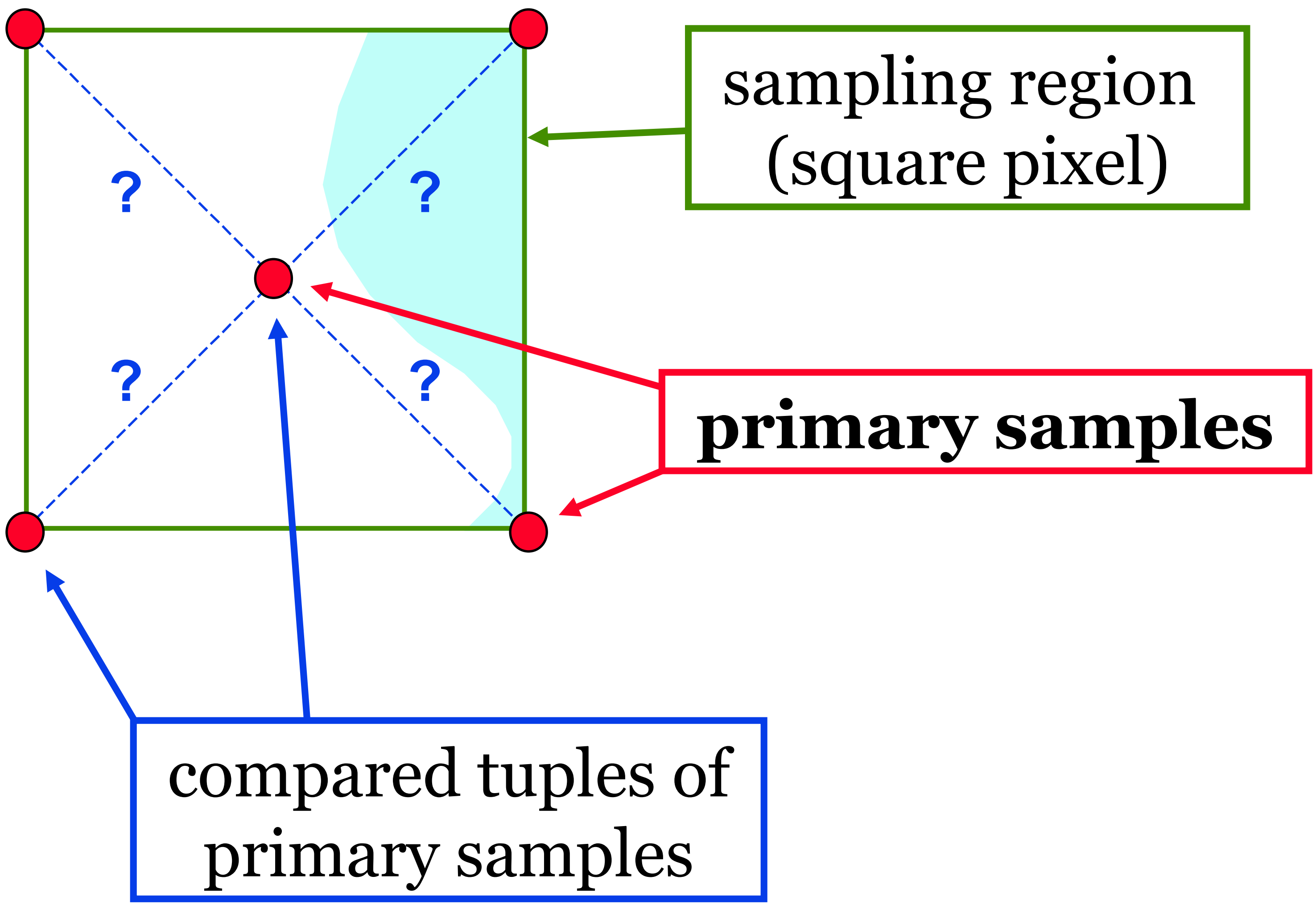


adaptive

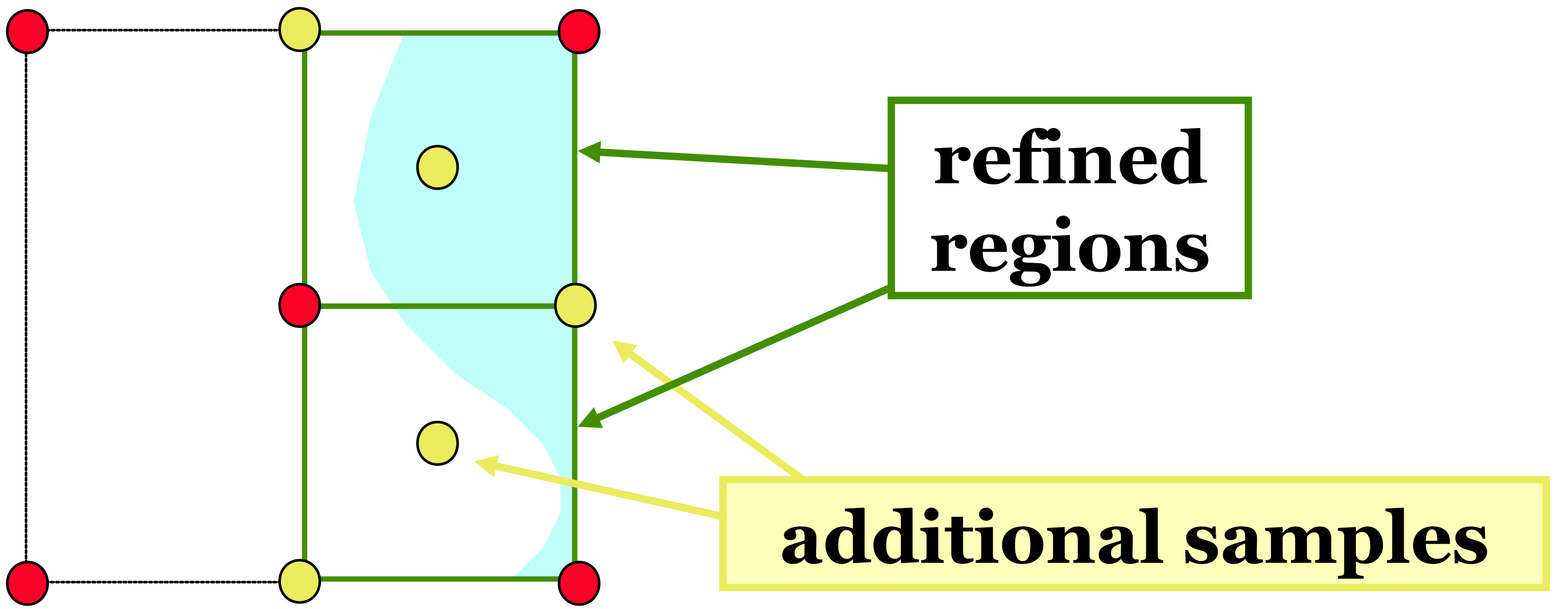


refinement map

Recursive refinement (Whitted)

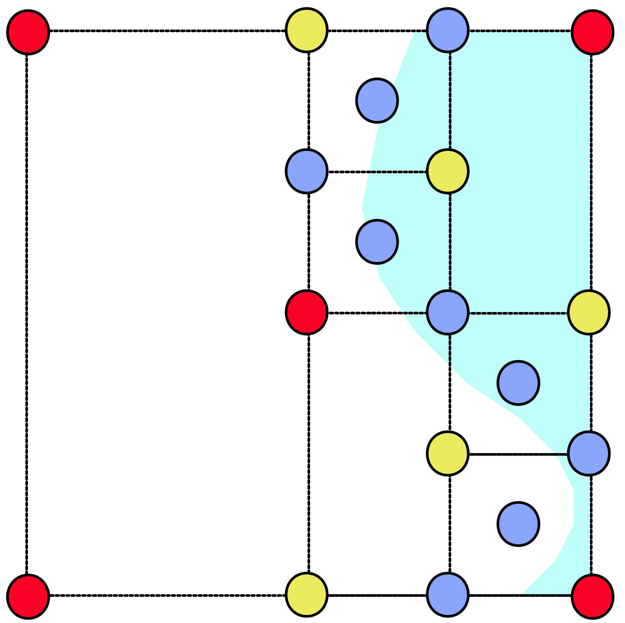


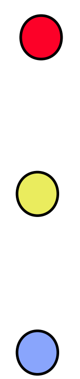
Refinement phase



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

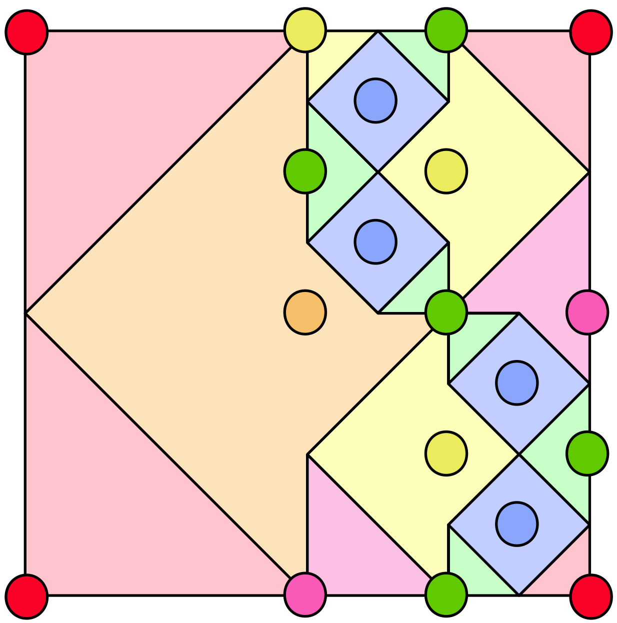
Result sample set

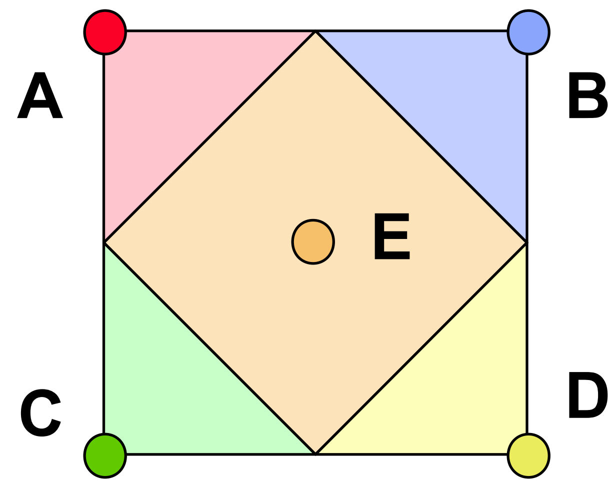




Evaluated: 5+5+9 = **19 samples**
(from total number of **41**)

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)

References

- **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