Stereological techniques for Solid Textures

Jan Tore Stølsvik

Overview

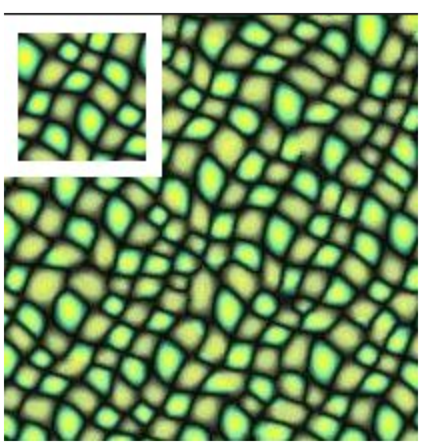
- Goal
- Why
- Stereology
- Solid Textures
- Estimating 3D Distributions
- Reconstructing the Volume
- Results
- Conclusions

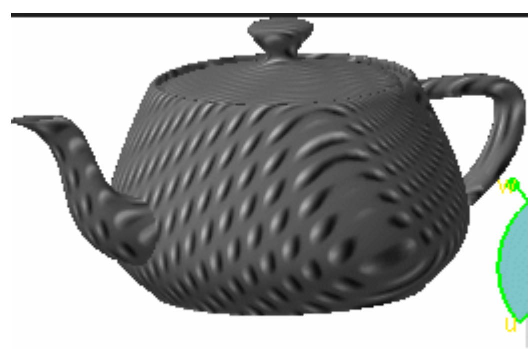
Goal

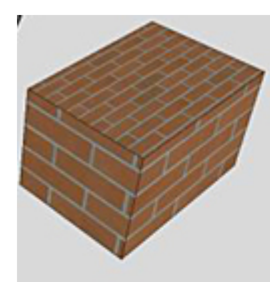
- Automatic Texture mapping for arbitrary shapes
- Textures with particle distribution like asphalt, rocks and also sponges (particles are void space)
- Should require 2D image input of material only
- Show Video

Why

- It is a difficult and tedious task to get multiple 2D textures to form a consistent visual appearance on a model without it looking bad
- 2D texture synthesis works bad for objects cut out of 3D spatially varying materials
- For use in computer graphics applications

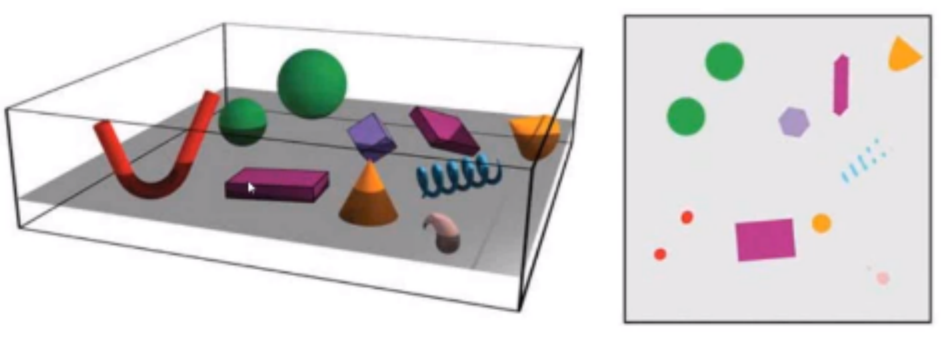






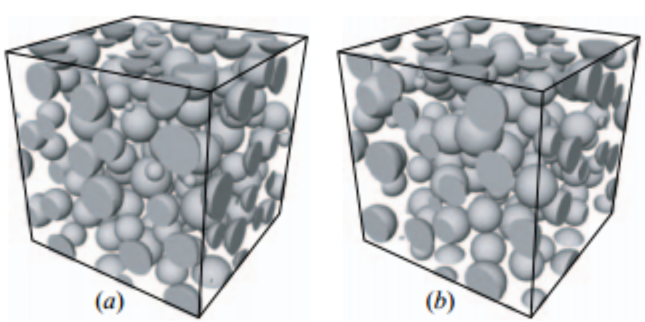
Stereology

- Used in biology and material science
- Try to guess the 3D shapes of a cross section



Solid Textures

- Make an object you want to texture
- To render this object each pixels gives its 3D position to the algorithm which tells it the color it should have
- It is difficult and tedious to get multiple 2D textures to form a consistent visual appearance on a model without looking bad.



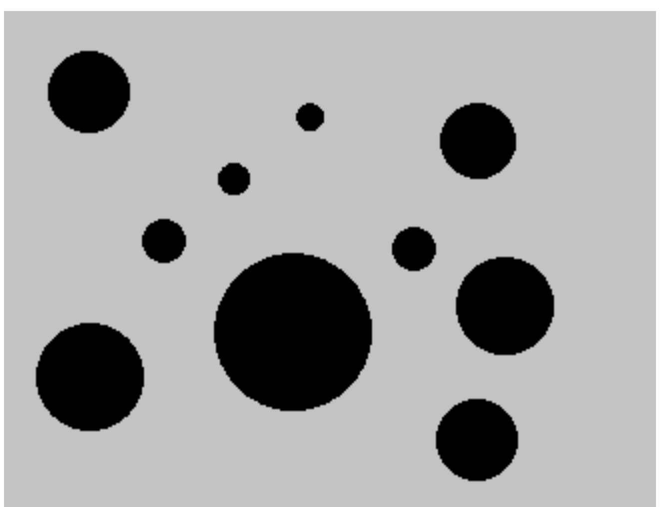
Estimating 3D Distribution

- As in stereology, but we try to estimate the distribution of different particles in a medium from a 2D slice
- Our objective is to establish a relationship between the size distribution of 2D circles in circles pr unit area and 3D spheres in spheres pr unit volume
- For any distribution of identical convex particles the particle density is expressed as

$$N_A = \bar{H} N_V$$

Spheres

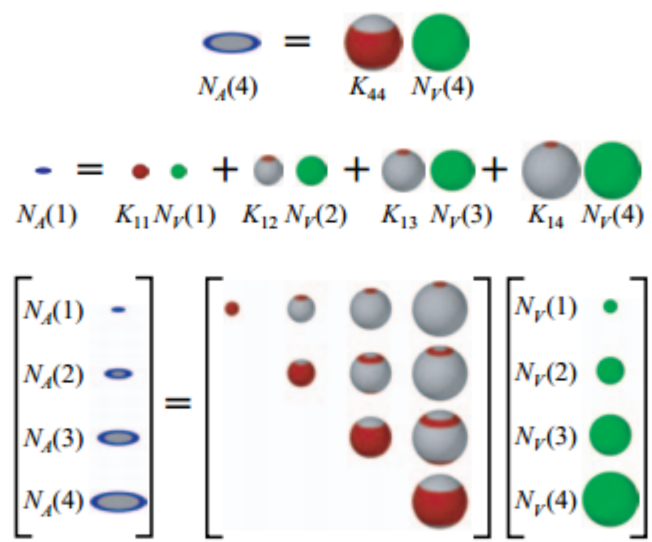
For most volumes it is unlikely that the particles will all be of the same size, so we use a histogram approach



$$N_A = \bar{H} N_V$$

$$N_A = d_{max} K N_V$$

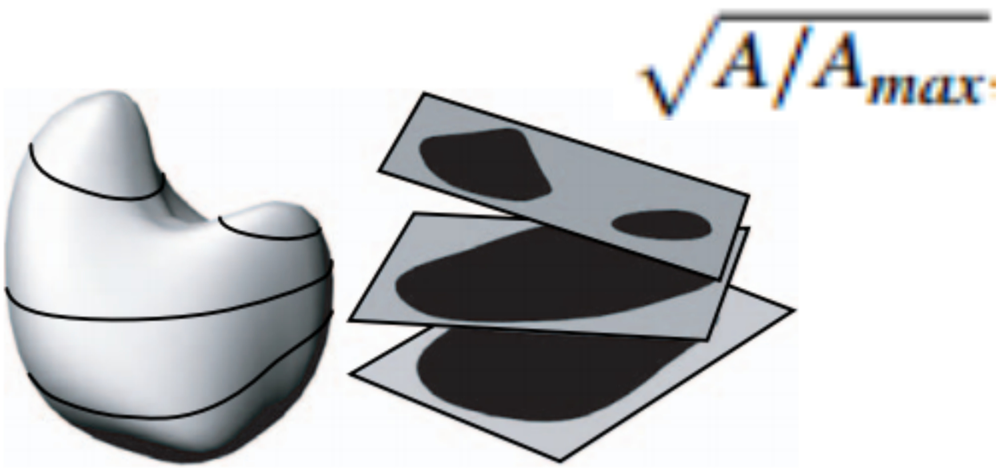
$$N_V = \frac{1}{d_{max}} K^{-1} N_A$$

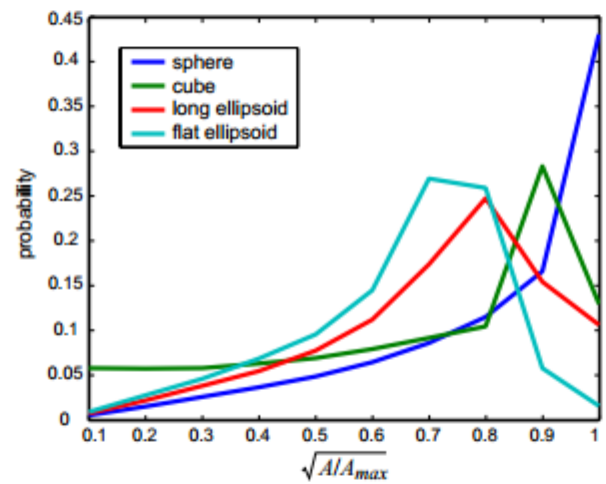


$$K_{ij} = \begin{cases} \frac{1}{n}\left(\sqrt{j^2-(i-1)^2} - \sqrt{j^2-i^2}\right) & \text{for } j \geq i \\ 0 & \text{otherwise} \end{cases}$$

Other Particles

- We cannot easily classify the profile size according to its diameter
- Kij represents the relative probability that a particle in column j exhibits a profile in row i





Other Particles

- To compute the K matrix they use the Monte Carlo routine to take advantage of the GPU
- Randomly orient the polygon mesh
- Render such that the near clipping plane of an orthographic camera cuts through the particle at random depth
- Read of the stencil buffer to find the area
- Keep track of the biggest area
- Record all these and create probabilities
- Run around 100.000 times
- Can also be used to find the mean caliper diameter

Other Particles

Before this data can be used we need a scale factor s to relate the size of particle P to the size of the particles seen in our input image

$$s = \sqrt{A_{img}/A_{Pmax}}$$

Aimg is the maximum area in the image and Amax is the Maximum area in our calculations

Calculate the H by multiplying s with the mean caliper diameter then solve as usual

$$N_V = \frac{1}{\bar{H}} K^{-1} N_A$$

Managing Multiple Particle Types

If there are multible types of particles of different shape each particle type i will have its own mean caliper diameter, matrix Ki and distribution Nvi

$$N_A = \sum_i \left(\bar{H}_i K_i N_{Vi}\right)$$

We can assume that the distribution of size is independant of type, so we can express this as

$$\begin{aligned} N_A &= \sum_i \left(\bar{H}_i K_i P(i) N_V\right) \\ &= \sum_i \left(\bar{H}_i K_i P(i)\right) N_V \end{aligned}$$

where Nv is the sum of all Nvi’s and P(i) is the probability that particle p is of type i

Reconstructing the Volume

- Once we have the particle densities Nv a volume can be constructed, the process establishes particle positions and colors
- Create particles according to the distribution
- Place all the particles inside a volume
- Run simulated annealing to resolve collisions
- This repeatedly searches for all collisions and then relaxes particle positions to reduce collisions

Reconstructing the Volume

- If size and color are uncorrealated then assign colors to the particles randomly
- If this is not the case use a k-means clustering algorithm to set the mean profile colors. Then run the stereology algorithms on each color and then combine the results
- To add fine details the mean is subtracted from the original picture and an existing similar method is used to add these details

Results

- Recreation of synthetic volumes, very successful
- Work with physical data was hard because the algorithm depends on you creating the particles you think will be in the volume
- Test volume, they carved a shape out of real material and challenged the algorithm to do the same
- Physical inputs, they tried to replicate simple 2D pictures of materials

Conclusions

- The methods expand on the class of 3D solid textures that can be synthesized from 2D photographs
- Building on stereology and existing literature the methods manage to perform accurate predictive rendering with a sound statistical approach
- Future work includes automated estimation of 3D particle shapes and support for greater variety of input textures