Rendering Smoke & Clouds

Game Design Seminar 2007

Jürgen Treml

tum.3D

computer graphics & visualization

Talk Overview

1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
    1. **Generating Clouds**
    2. **Rendering Clouds using Volume Rendering**
    3. **Example: Clouds à la Dobashi**
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. **Generating / Designing Clouds**
    2. **Rendering Clouds**
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

Where are we?

1. **Introduction to Clouds**
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. Virtual Clouds – An Artistic Approach
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

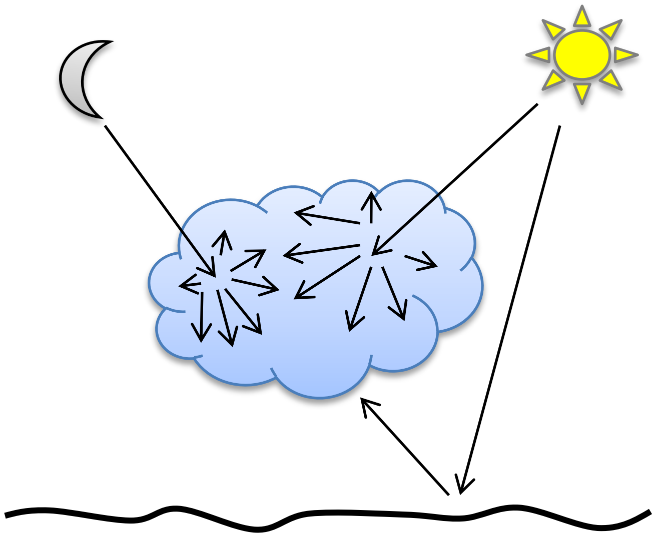
computer graphics & visualization

1. Introduction to Clouds

- What are clouds?
- Mass of visible water droplets
- Technically speaking: Continuous 3D density field of (condensed) water in the air
- Form when warm air cools down and condensates
- Formation influenced by: Temperature, Pressure, Humidity Ratio Conensation / Evaporation

1. Introduction to Clouds

- Why do we see clouds?



- Color depends on
  - Spectrum of incoming light
  - Atmosphere
  - Angle to sun
  - Angle to viewer
  - ...

Where are we?

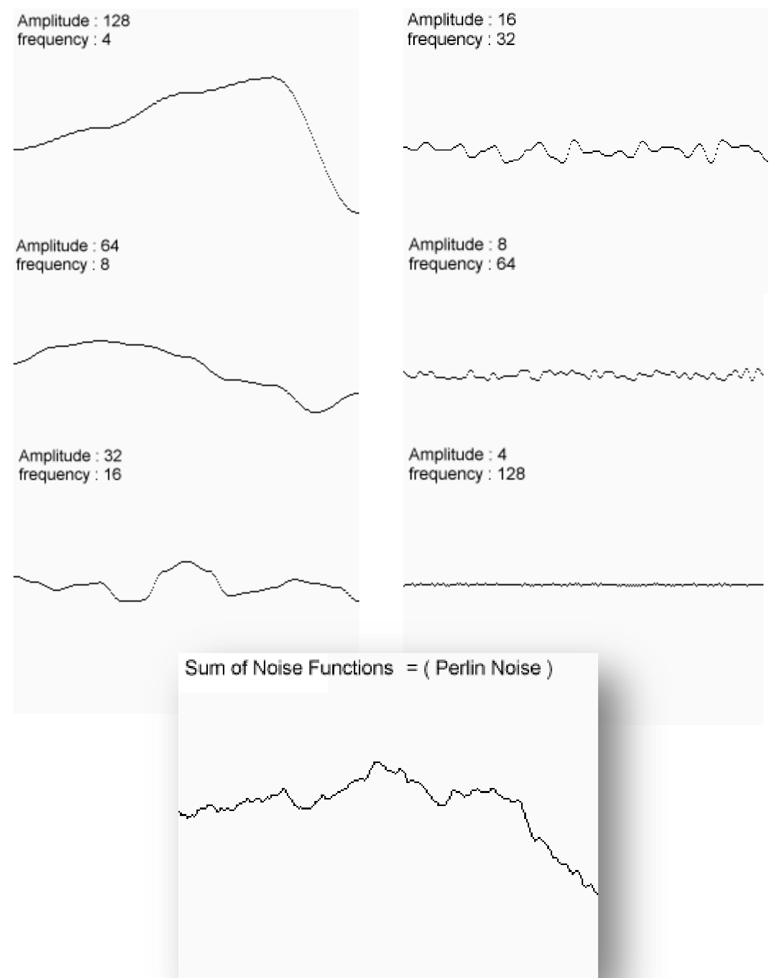
1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
   1. **Generating Clouds**
   2. Rendering Clouds using Volume Rendering
   3. Example: Clouds à la Dobashi
   4. Extending Dobashi: Multiple Forward Scattering
   5. A few Notes on Cloud Animation
3. Virtual Clouds – An Artistic Approach
   1. Generating / Designing Clouds
   2. Rendering Clouds
   3. Performance Tweaks
   4. A few Notes on Animation
   5. Evaluation

2.1.1 Meteorological / Physical Model

- Implement a meteorological model
  - Simulating and modeling environment: air pressure, temperature, humidity and saturation
  - Account for
    - Potential temperature
    - Buoyant force
    - Environmental lapse rate
    - Saturation mixing ratio
    - Water continuity
    - Thermodynamics
    - Vorticity confinement
    - Fluid flow
    - ...
  - Clouds creation, movement and dissipation as an ad-hoc result

2.1.2 Noise Based Model

- 1/f-noise
  - Functional noise, i.e. no memory footprint
  - Fast (Faster than simulation)
  - Arbitary number of dimensions
  - Natural look
    - stochastic, self-similar
    - 1/f: decreasing amplitude with increasing frequency



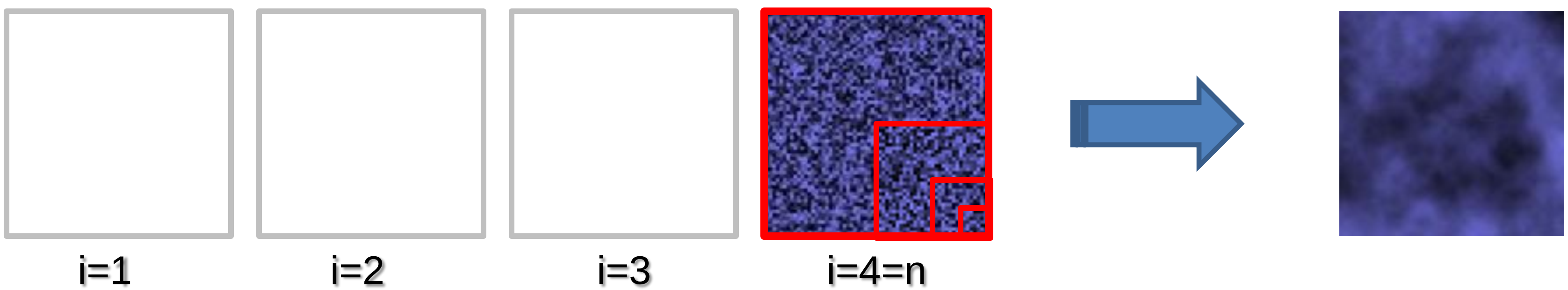
tum.3D computer graphics & visualization

2.1.2 Noise Based Model

- In mathematical terms:

$$N(x,y) = \sum_{i=1}^{n} \frac{1}{2^i} B\left(\frac{1}{2^{n-i}} x, \frac{1}{2^{n-i}} y\right)$$

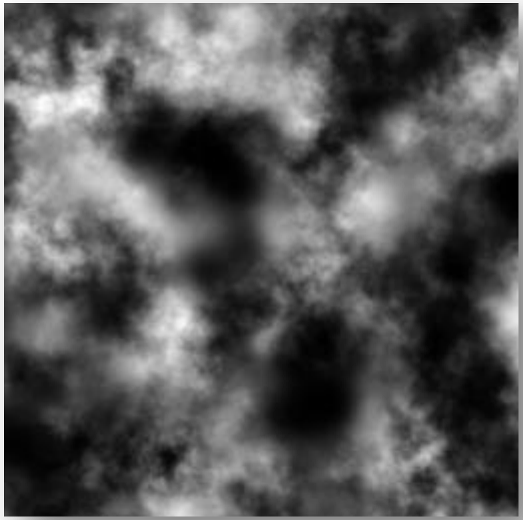
$N(x,y)$: Synthetic noise value
$B(x,y)$: Base function
n : Number of octaves

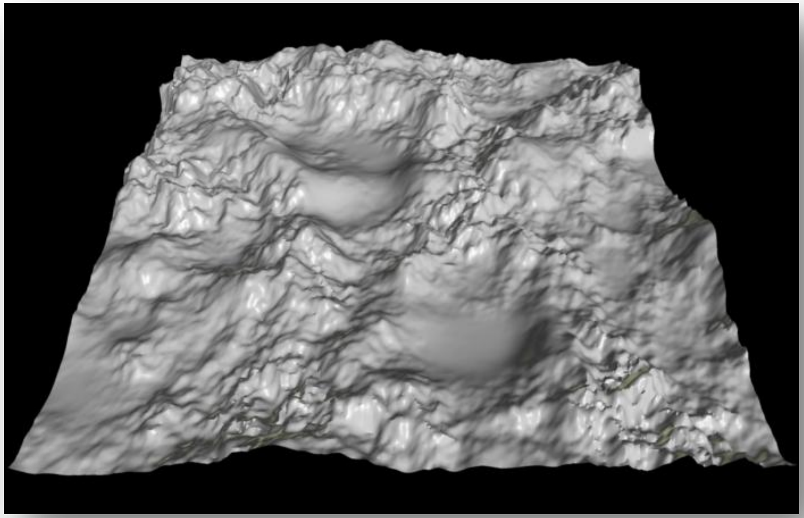


- Base Function:
  - white noise, Perlin noise, image, etc.
  - Pre-created and stored or pseudo-random function

2.1.2 Pseudo-Random Noise Based Model

- Noise interpretation:
  - 2D noise, e.g. height map (terrain)
  - 3D noise, e.g. Volume density field (clouds)
  - 4D noise, e.g. for time-animated 3D fields



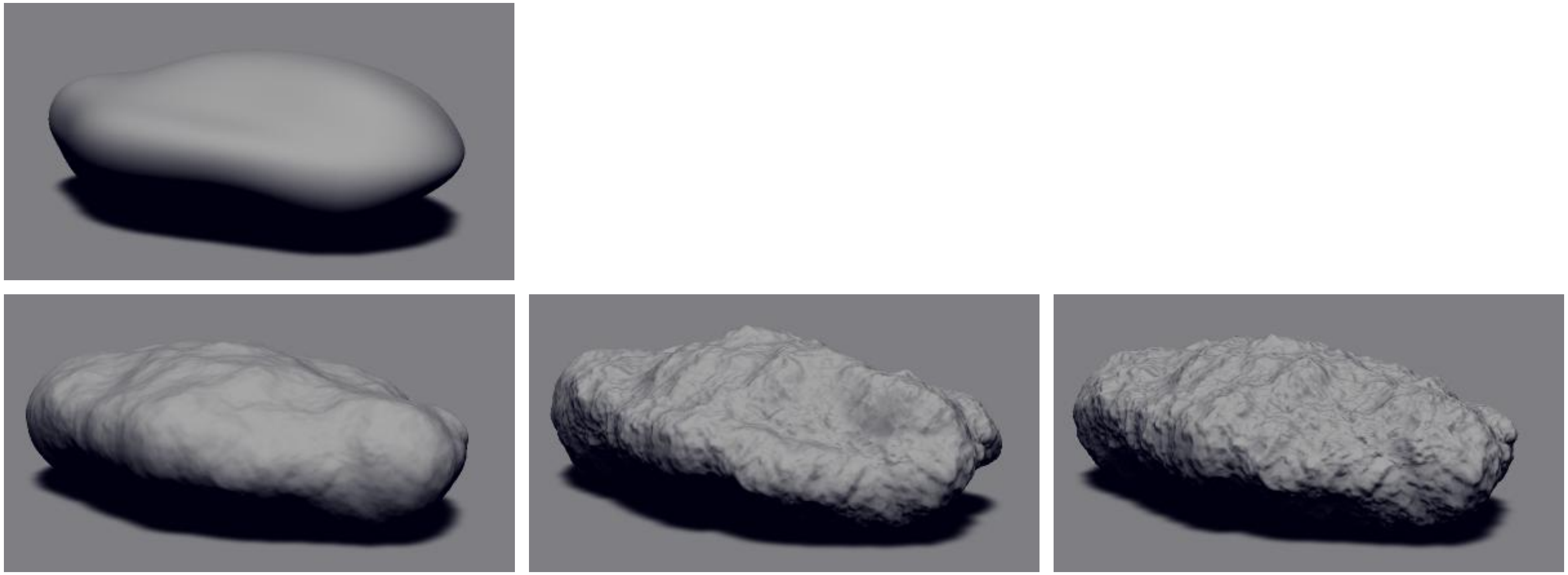


2.1.2 Pseudo-Random Noise Based Model

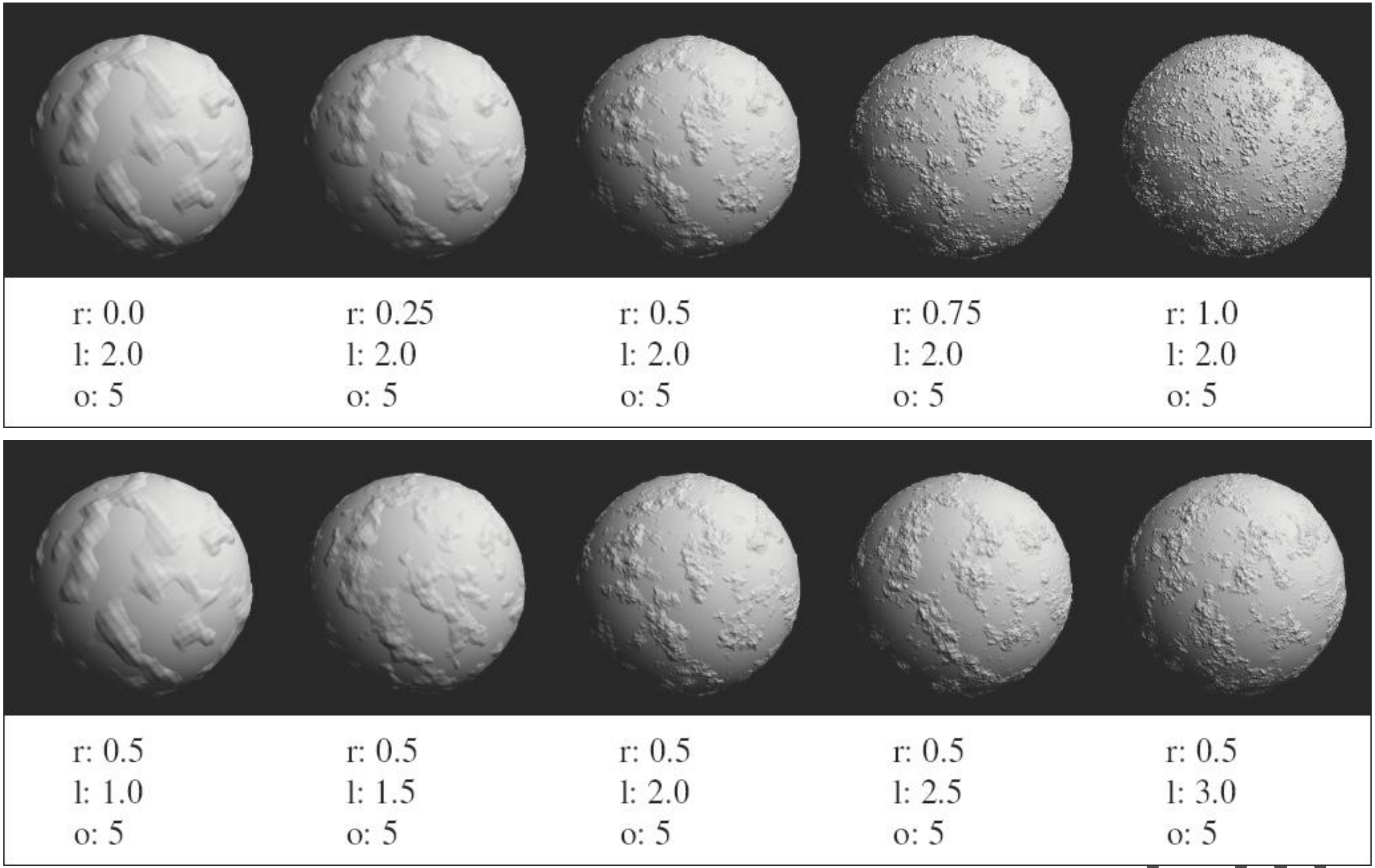
- Tweaking the noise function

$$N(x,y)=\sum_{i=1}^{n}\frac{1}{2}r^{i}B\left(\frac{1}{l^{n-i}}x,\frac{1}{l^{n-i}}y\right)$$

r : roughness
l : fractal gap (lacunarity)
n : Number of octaves

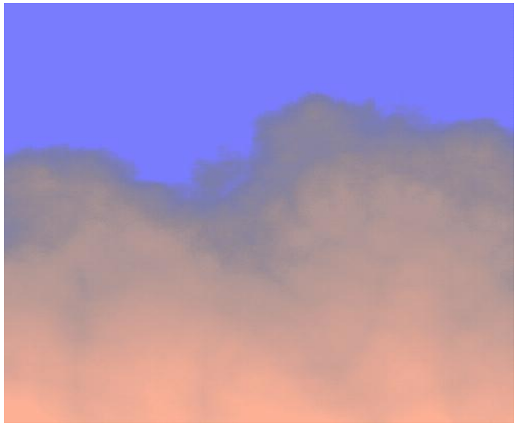


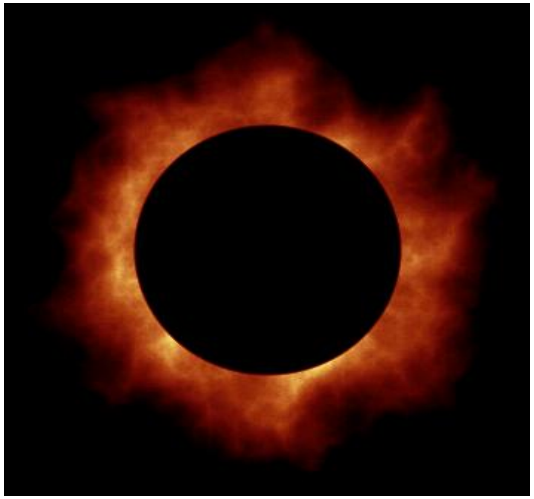
2.1.2 Pseudo-Random Noise Based Model

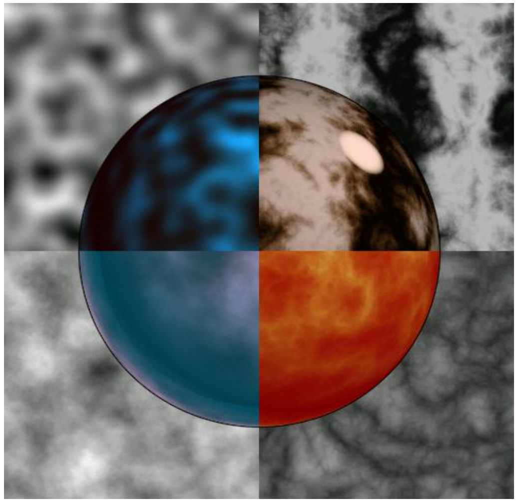


2.1.2 Pseudo-Random Noise Based Model

- Create various effects, changing...
  - Base function
  - Number of octaves
  - Roughness, lacunarity
  - ...

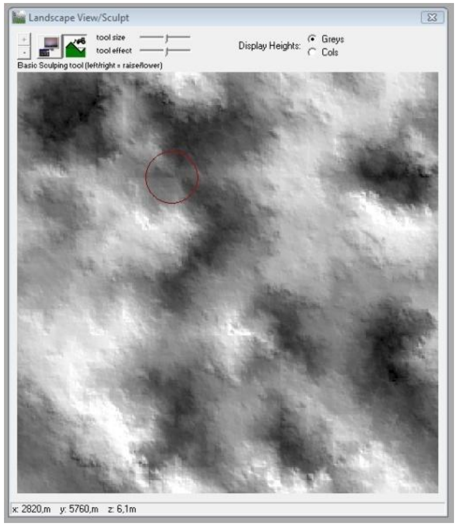


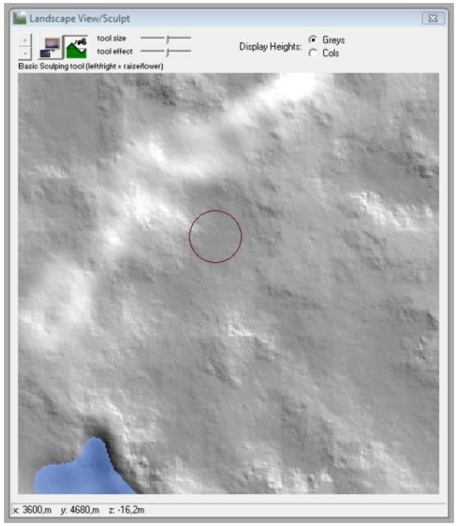




2.1.3 Noise-Based Editing Model

- Create basic noise field
- Let user edit the field
  - User may define parameters of the noise function before generating the noise field
  - User can edit the generated noise (like a brush or eraser in photoshop or paint, etc.)
- Used in Terragen for example





Where are we?

1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
    1. Generating Clouds
    2. **Rendering Clouds using Volume Rendering**
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. Virtual Clouds – An Artistic Approach
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

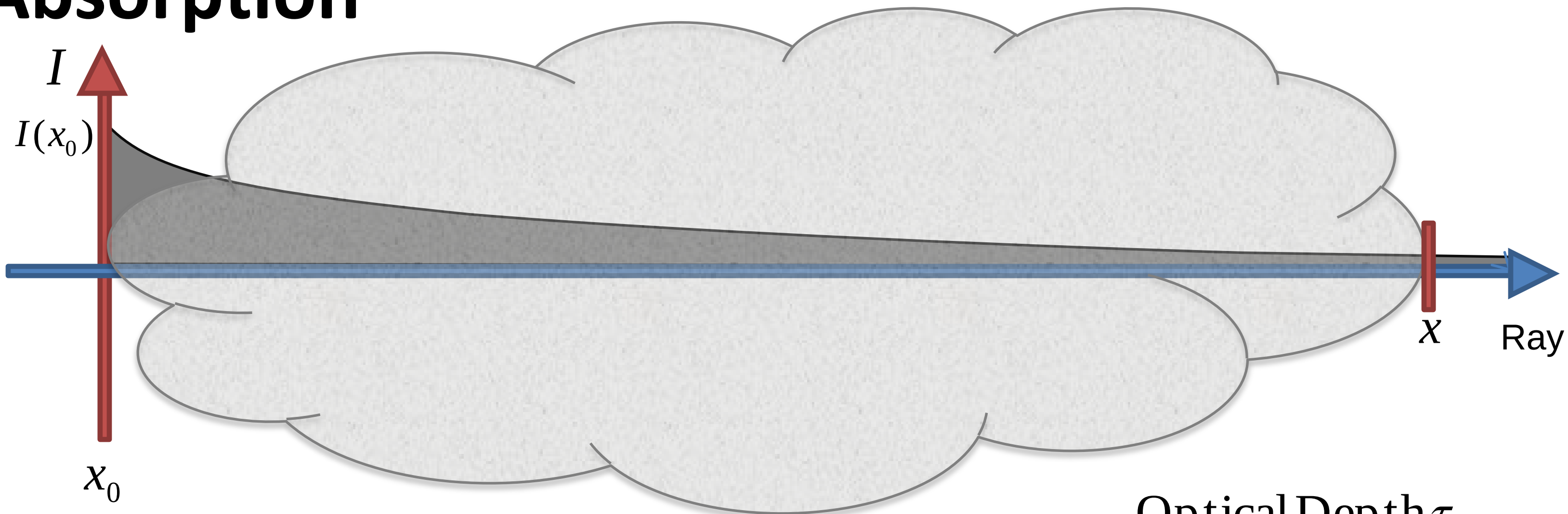
2.2.1 The Volume Rendering Integral

- Optical Model:
  Light (particles) travelling through / interacting with density volume
- Effects:
  - Absorption
  - Emission
  - (Multiple) Scattering
  - (Shadows)

2.2.1 The Volume Rendering Integral

- Physical Model: Emission and absorption only

Absorption



Optical Depth τ

Absorption κ

$$\tau(x_1, x_2) = \int_{x_1}^{x_2} \kappa(x)\,dx$$

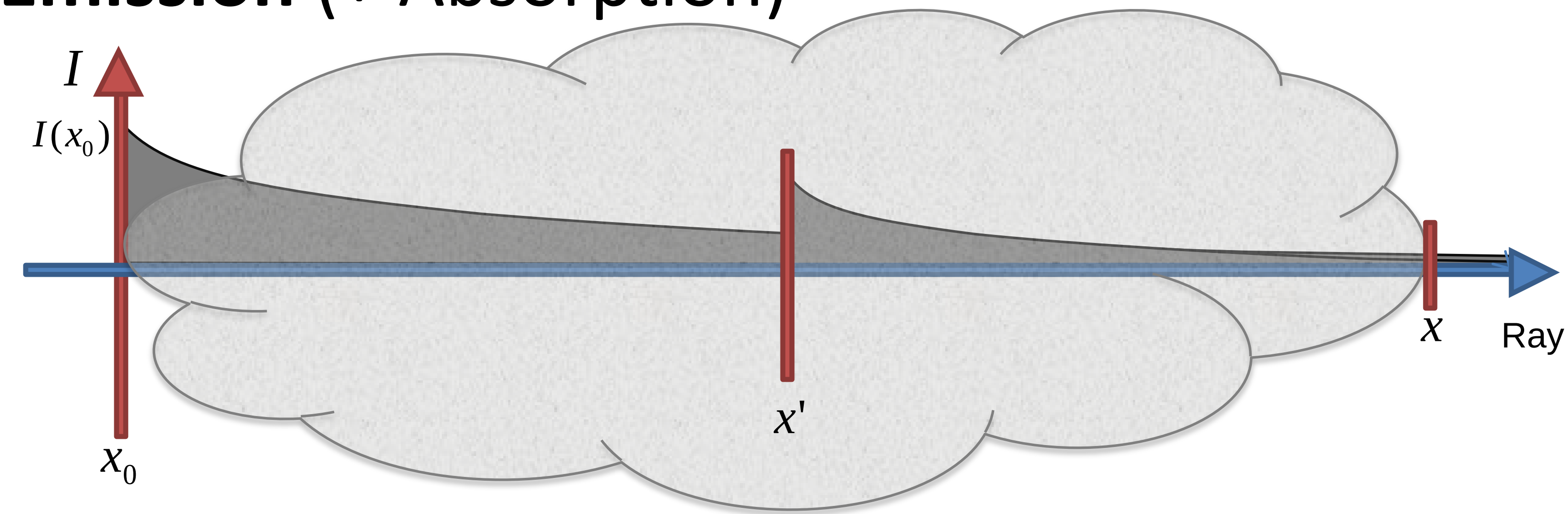
$$I(x) = I(x_0)e^{-\tau(x_0,x)}$$

Initial Intensity *Attenuation along [x_0,x]*

2.2.1 The Volume Rendering Integral

- Physical Model: Emission and absorption only

Emission (+ Absorption)



$$I(x) = I(x_0)e^{-\tau(x_0,x)} + \int_{x_0}^{x} q(x')e^{-\tau(x',x)}dx'$$

Initial Intensity at x' *Attenuation along [x',x]*

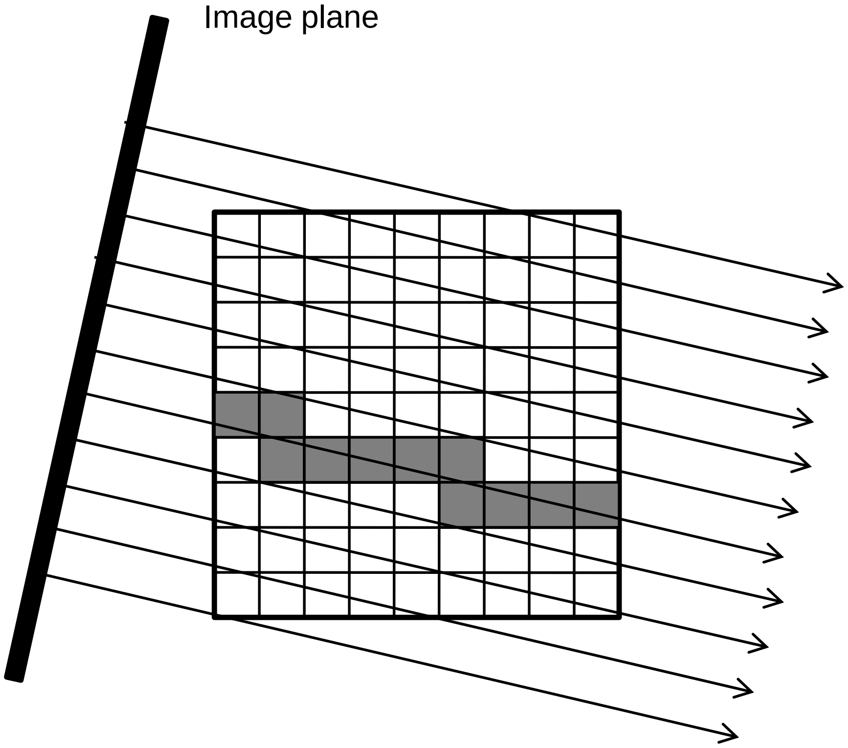
4.2.1 The Volume Rendering Integral

- No general closed form solution
- Approximation by discrete sum:

→ Ray Casting

2.2.2 Backward VR: Ray Casting

- Image space algorithm
- Pixel by pixel
- Cast rays into volume
- Sample volume at discrete intervals

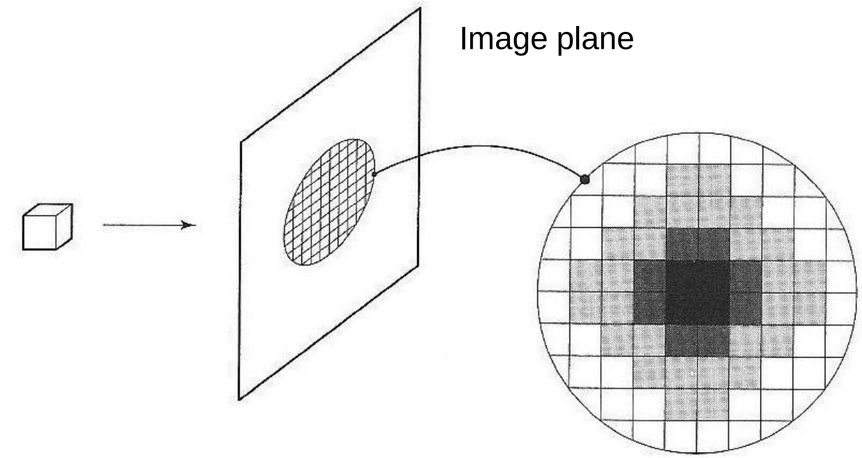


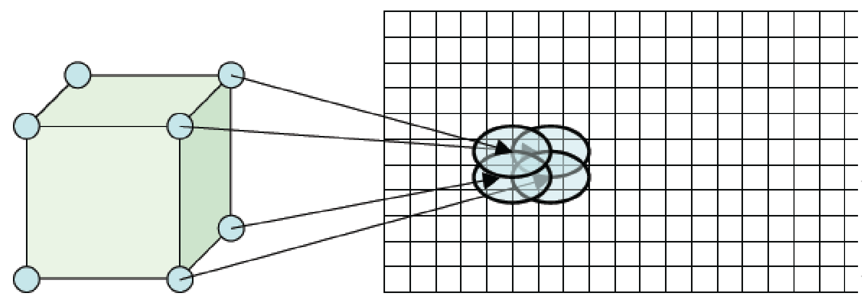
2.2.2 Backward VR: Ray Casting

- Usually resampling required (ray doesn‘t hit voxel centers)
  → interpolate / filter (trilinear, splines, ...)
- Accumulate color and opacity along the ray

2.2.3 Forward VR: Splatting

- Object space algorithm
- Voxel by voxel
- Project each voxel onto the image plane
- One voxel usually influences several pixels





2.2.3 Forward VR: Splatting

- Apply filter when projecting voxel on pixels (e.g. Gaussian)
- Example: Cloud rendering algorithm proposed by Dobashi (2000)

computer graphics & visualization

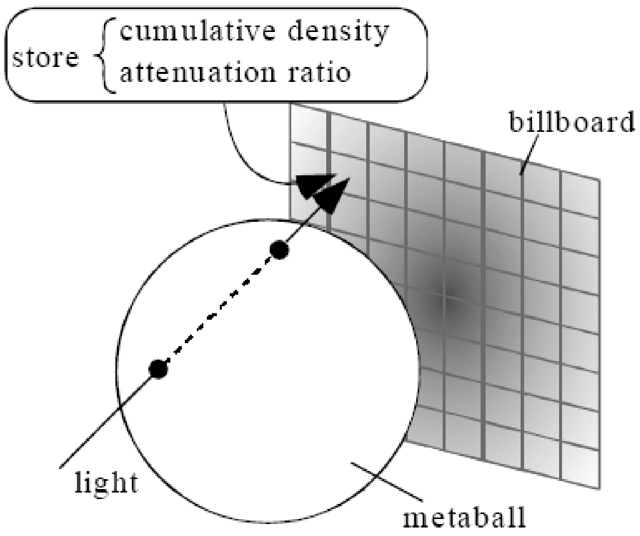
Where are we?

1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. **Example: Clouds à la Dobashi**
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. Virtual Clouds – An Artistic Approach
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

2.3 Example: Clouds à la Dobashi

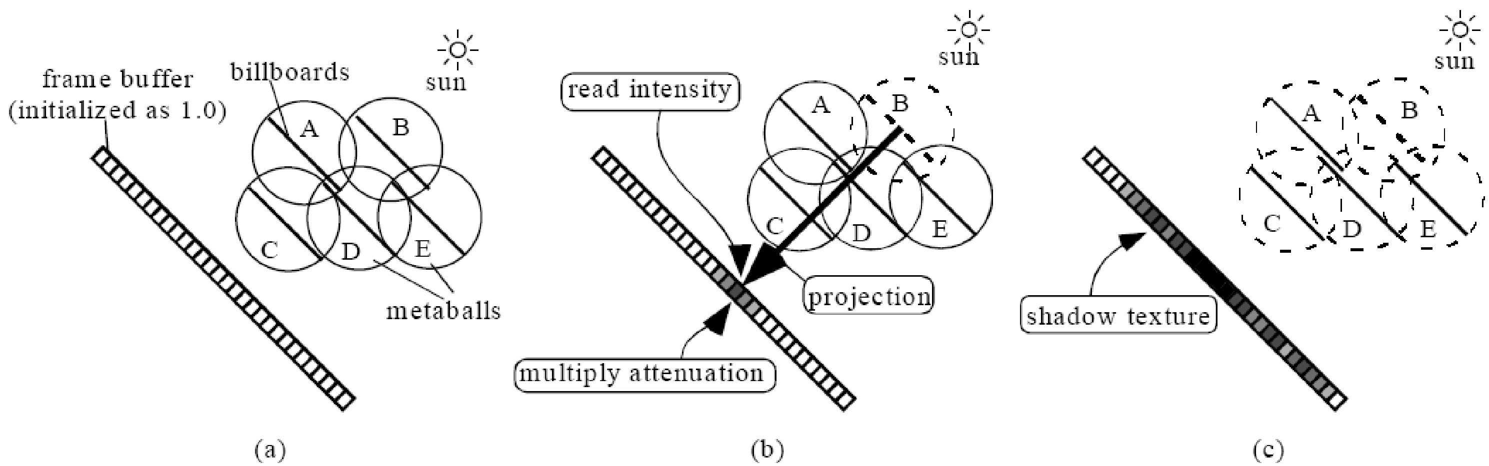
- Given (for now):
  Discrete density distribution (voxel grid)

- Project each voxel on a billboard:
  filter using **metaballs** (similar to Gauss but effective radius of influence)



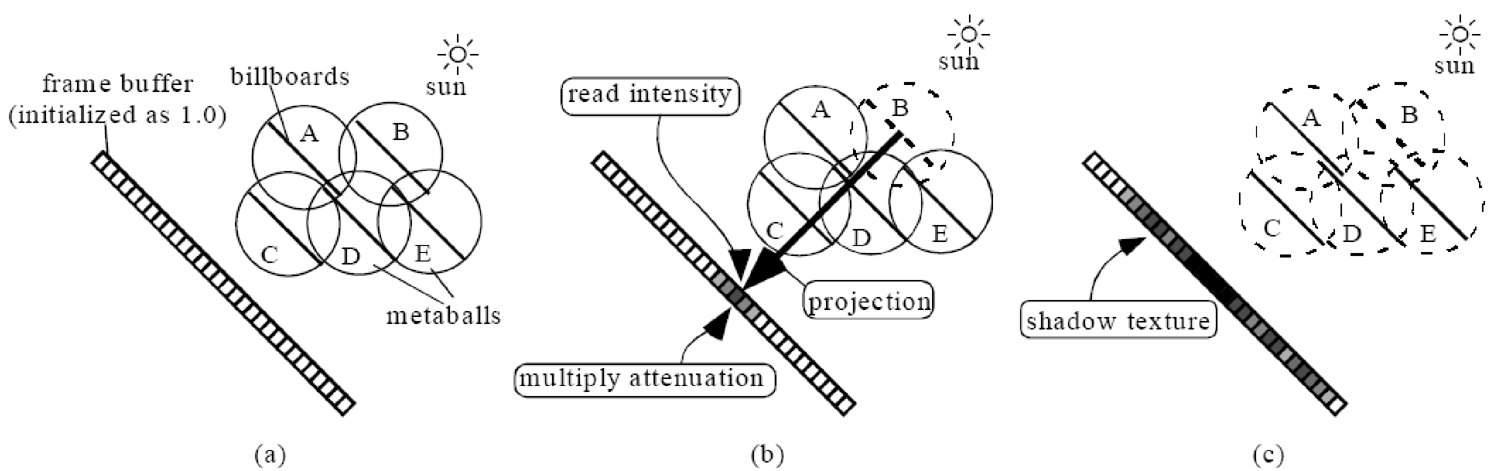
2.3 Example: Clouds à la Dobashi

- Render image as viewed from sun
  - Orient billboards towards sun
  - Starting from closest to sun:
    for each metaball, render billboard to framebuffer (multiply attenuation)



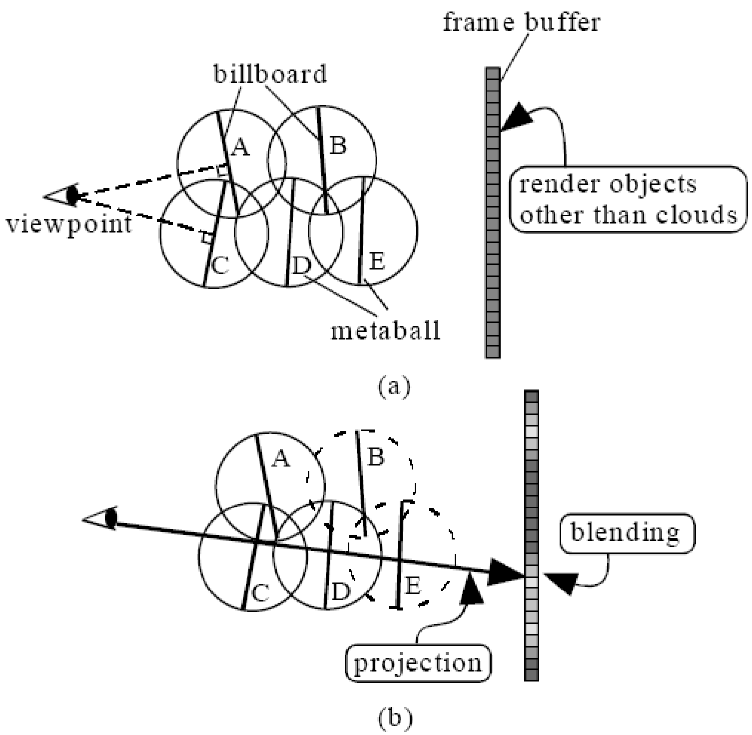
2.3 Example: Clouds à la Dobashi

- Render image as viewed from sun
  - Read pixel corresponding to metaball center from framebuffer
    → attenuation between metaball and sun
  - Multiply by sunlight color
    → metaball color



2.3 Example: Clouds à la Dobashi

- Render image from user perspective
  - Render all objects besides clouds
  - Orient billboards towards viewpoint
  - Project on image plane (back to front):
    - multiply framebuffer color by attenuation ratio
    - Add metaball color



2.3 Example: Clouds à la Dobashi

A few notes:

- Attenuation „texture“ created during second step can be used as shadow map (for the ground)
- Method accounts for single scattering of light and shadows

Where are we?

1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
   1. Generating Clouds
   2. Rendering Clouds using Volume Rendering
   3. Example: Clouds à la Dobashi
   4. **Extending Dobashi: Multiple Forward Scattering**
   5. A few Notes on Cloud Animation
3. Virtual Clouds – An Artistic Approach
   1. Generating / Designing Clouds
   2. Rendering Clouds
   3. Performance Tweaks
   4. A few Notes on Animation
   5. Evaluation

2.4 Next Step: Multiple Forward Scattering

- Proposed by Harris and Lastra 2001
- Just a short overview

- Single scattering similar to Dobashi
- Extending the VRI to account for multiple forward scattering:

$$I(P,\omega) = I_0(\omega) e^{-\int_0^{Dp} \tau(t)dt} + \int_0^{Dp} g(s,\omega) e^{-\int_s^{Dp} \tau(t)dt} ds$$

2.4 Next Step: Multiple Forward Scattering

$$I(P,\omega) = I_0(\omega)e^{-\int_0^{Dp}\tau(t)dt} + \int_0^{Dp} g(s,\omega)e^{-\int_s^{Dp}\tau(t)dt} ds$$

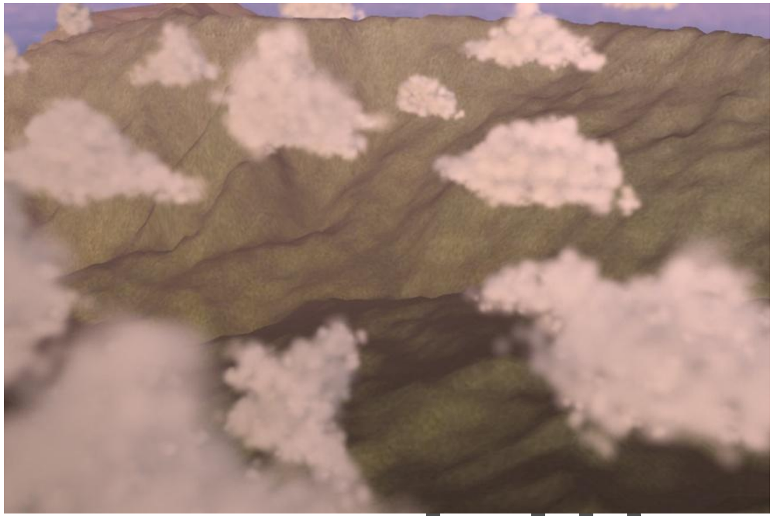
$$g(x,\omega) = \int_{4\pi} r(x,\omega,\omega')I(x,\omega')d\omega'$$

- What‘s the message?
  - Each particle besides light from outside a cloud also receives light scattered by other particles
  - Amount is a function of the angle (spatial angle)
  - Caracterized by the BSDF and phase function (e.g. Rayleigh scattering)

2.4 Next Step: Multiple Forward Scattering

- Again solved by discrete approximization
- Forward scattering accounts most for the optical perception of clouds
  →restrict calculations to small angle around the forward direction
- Assuming BSDF and other factors being constant (due to small angle)
- Split light path into small number of discrete directions
- …

2.4 Next Step: Multiple Forward Scattering



Where are we?

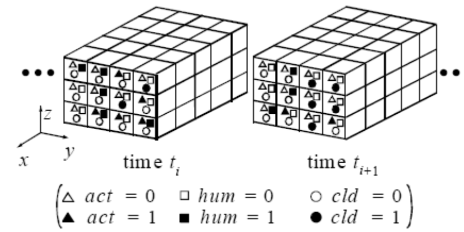
1. Introduction to Clouds
2. **Virtual Clouds based on physical Models**
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. **A few Notes on Cloud Animation**
3. Virtual Clouds – An Artistic Approach
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

2.5 A few Notes on Animating Clouds

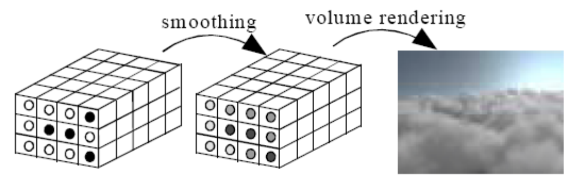
- Based on meteorological model (as with cloud generation)
- Account for all physical phenomena
  → Clouds creation, movement and dissipation as an ad-hoc result

2.5 A few Notes on Animating Clouds

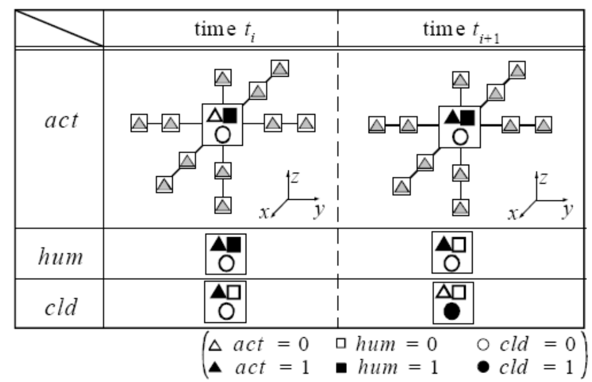
- Simple model: **cellular automata**
  - E.g. Used by Dobashi
  - Cells of an automaton correspond to voxels and carry **state variables**: Vapor/humidity, Clouds, Phase transition
  - **Binary states**!
  - Set of **transition functions**
  - **Smoothing** before redering



(a) Simulation process.



(b) Rendering process.



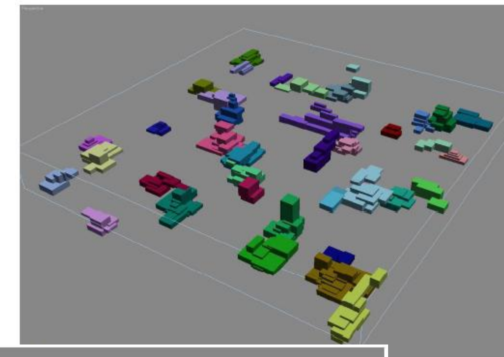
tum.3D computer graphics & visualization

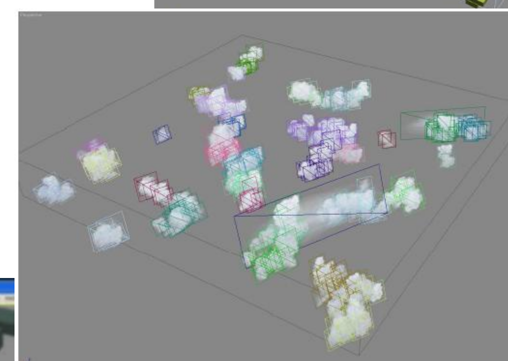
Where are we?

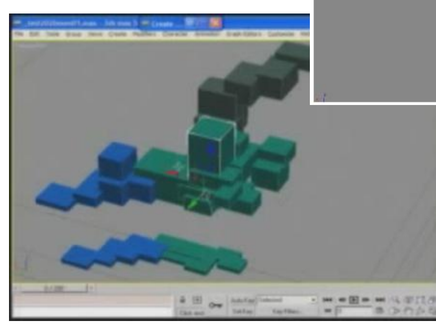
1. Introduction to Clouds
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. **Generating / Designing Clouds**
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

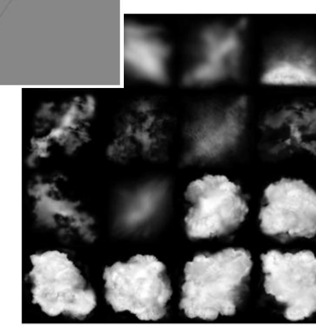
3.1 Generating / Designing Clouds

- Artist designs clouds with a GUI based tool
- Use simple shapes (boxes, spheres) to model the basic cloud shapes
- Fill boxes randomly with (textured) sprites



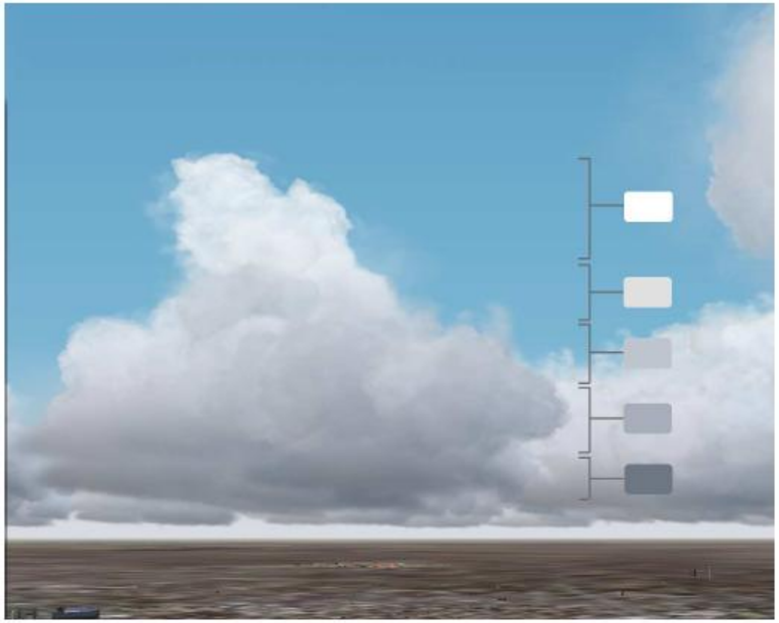






3.1 Generating / Designing Clouds

- Artist can specify density, etc.
- Artist specifies cloud coloring and shading
  - Percentage of ambient color (time of day)
  - Vertical color levels and colors
  - Shading groups
  - Directional colors (time of day)
- Store only sprite center points and sizes (+ coloring information)



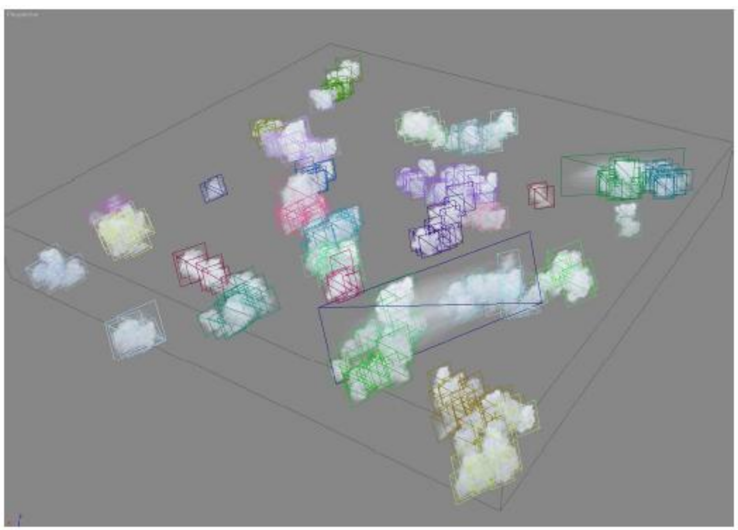
computer graphics & visualization

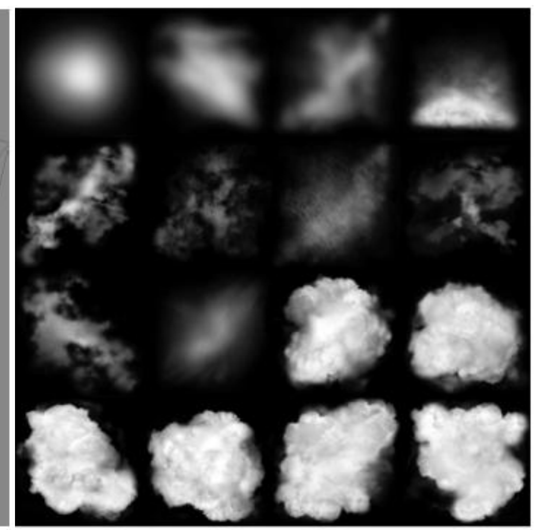
Where are we?

1. Introduction to Clouds
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. Generating / Designing Clouds
    2. **Rendering Clouds**
    3. Performance Tweaks
    4. A few Notes on Animation
    5. Evaluation

3.2 Rendering Clouds

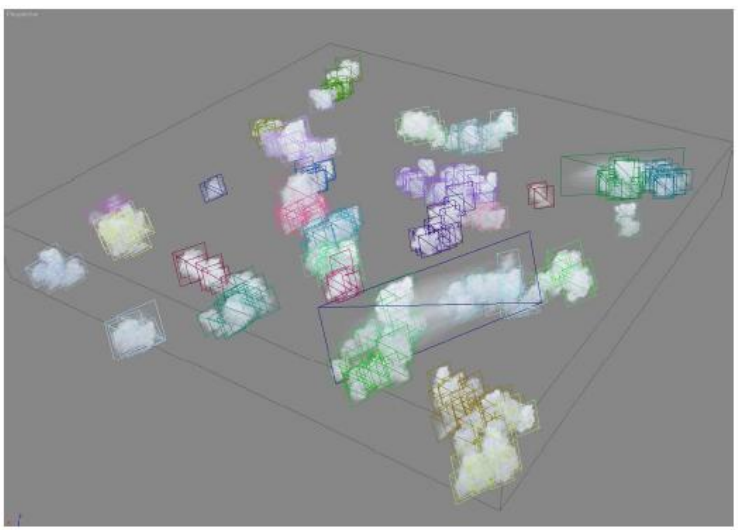
- Load sprite center points, sizes and color information from disk
- Draw quads around sprite centers
- Map textures to quads (random rotation in quad plane)

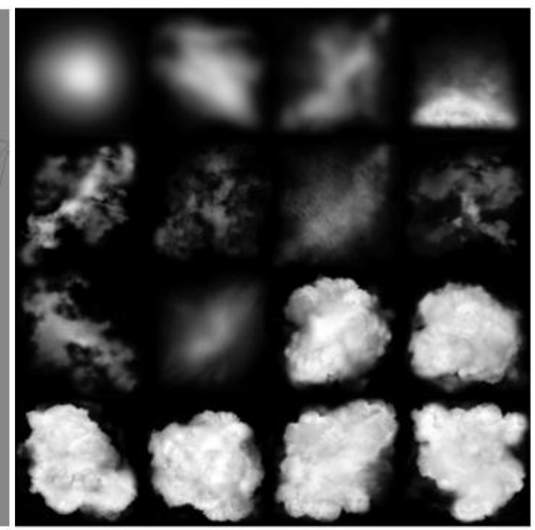




3.2 Rendering Clouds

- Rotate quads towards camera
- Calculate shading:
  function of angle between vector *shading-group center <> sun* and *shading group center <> sprite*
- Take into account directional and vertical color levels:
  interpolate between discrete levels specified by artist
- Render sprites to frame buffer



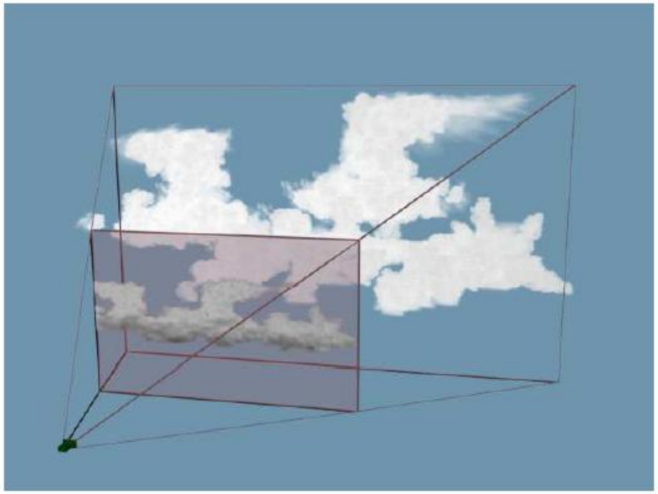


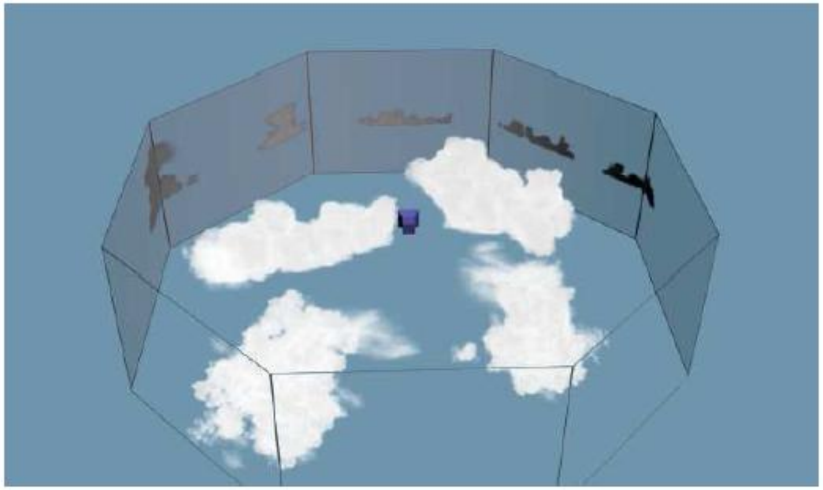
Where are we?

1. Introduction to Clouds
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. **Performance Tweaks**
    4. A few Notes on Animation
    5. Evaluation

3.3 Performance Tweaks

- Use impostors for clouds outside a certain range to the user aircraft
  - *Octogonal ring*
  - *Switch between impostors as user changes viewing direction*
  - *Visual imperfection vs. gain of speed*





Where are we?

1. Introduction to Clouds
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. **A few Notes on Animation**
    5. Evaluation

3.4 A few Notes on Animating Clouds

- Not much animation done actually
- Only cloud formation and **dissipation**
  - Done by slowly increasing transparency
  - Start with sprites at the borders till finally reaching the innermost sprites
  - Cloud formation is just the opposite

tum.3D computer graphics & visualization

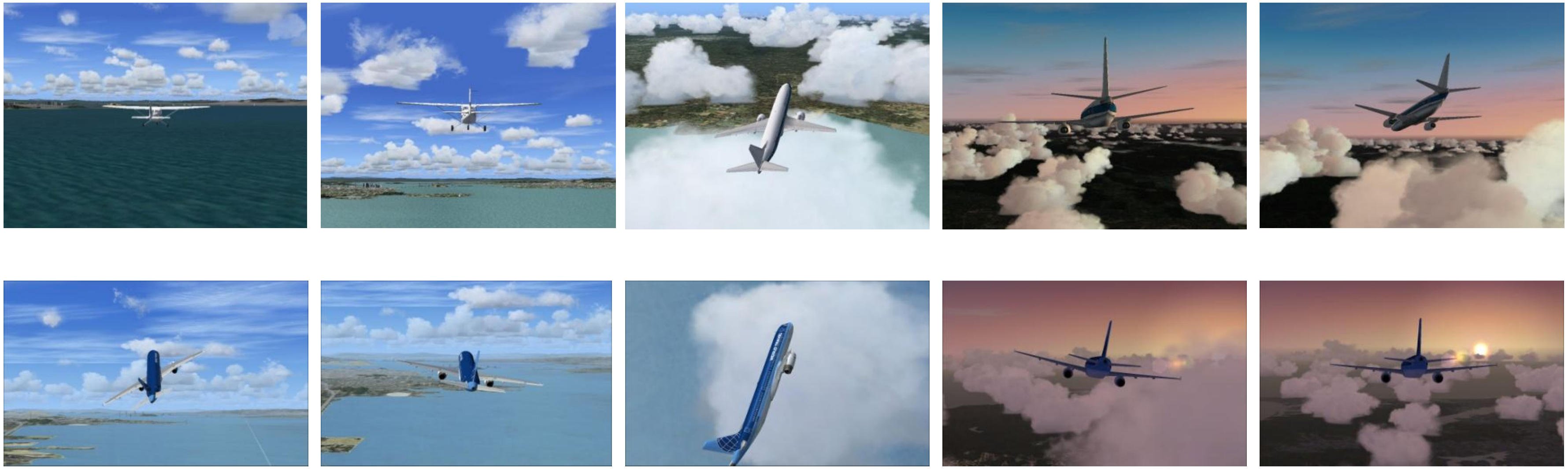
Where are we?

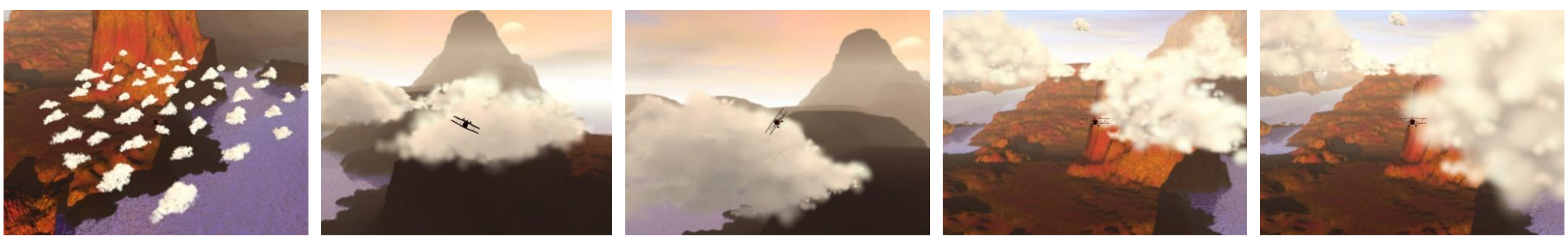
1. Introduction to Clouds
2. Virtual Clouds based on physical Models
    1. Generating Clouds
    2. Rendering Clouds using Volume Rendering
    3. Example: Clouds à la Dobashi
    4. Extending Dobashi: Multiple Forward Scattering
    5. A few Notes on Cloud Animation
3. **Virtual Clouds – An Artistic Approach**
    1. Generating / Designing Clouds
    2. Rendering Clouds
    3. Performance Tweaks
    4. A few Notes on Animation
    5. **Evaluation**

3.5 Evaluation / Comparison

- Extremely rough approximization of the real physics (e.g. Vertical shading levels)
- Inaccurate shading
- No self-shadowing or any shadowing at all
- Extremely flexible in controlling the appearance of clouds
- Pretty fast (even on older PCs)
- Visually not totally unconvincing ;-)

3.5 Evaluation / Comparison





Talk Summary

- What are Clouds?
- Effects to consider when dealing with clouds
- Two different approaches on creating and rendering clouds
  - Random noise + volume rendering
  - Artistic models
- Few hints on cloud animation

Questions?!?