



# Realistic rendering of clouds in realtime

Antoine Bouthors

Advisor: Fabrice Neyret

Co-advisor: Nelson Max, UC Davis



INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE

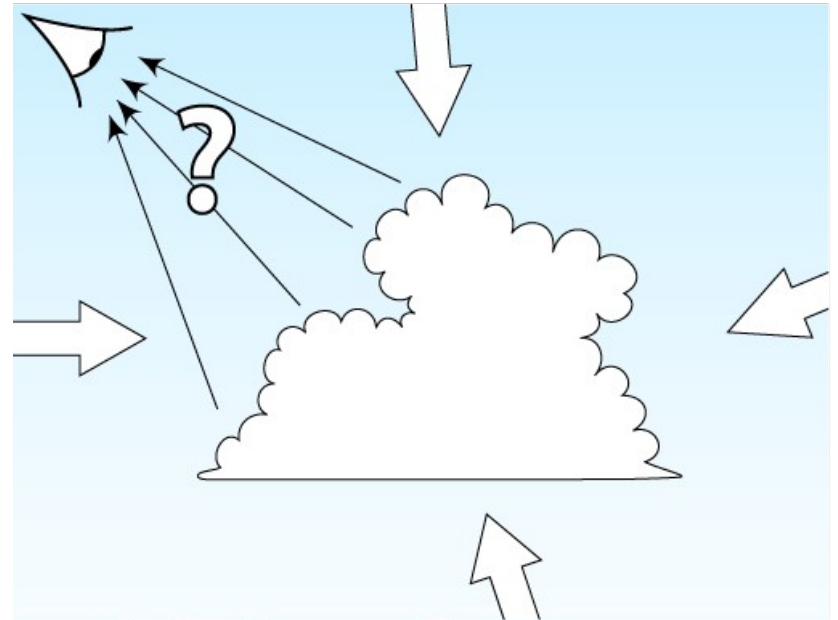


# Motivation



# Goals

- The rendering problem
  - Given an lit cloud
  - What is the **luminance** at every visible point ?



# Goals

- Render clouds
  - Realistically
  - At high resolution (high detail)
  - In real-time (30 frames per second)
  - That can be animated



# What is realism ?

- Ambiguous question
- Same **visual features** as real clouds

Are these clouds realistic or not ?



# Overview

- Part I:
  - Identify visual features
  - Study previous works
- Part II:
  - Study of light transport in clouds
- Part III:
  - New rendering methods for clouds

# What is a cloud ?

1. The human point of view
1. The physics point of view
2. The CG point of view

# Shape



Cumulonimbus



Cirrus



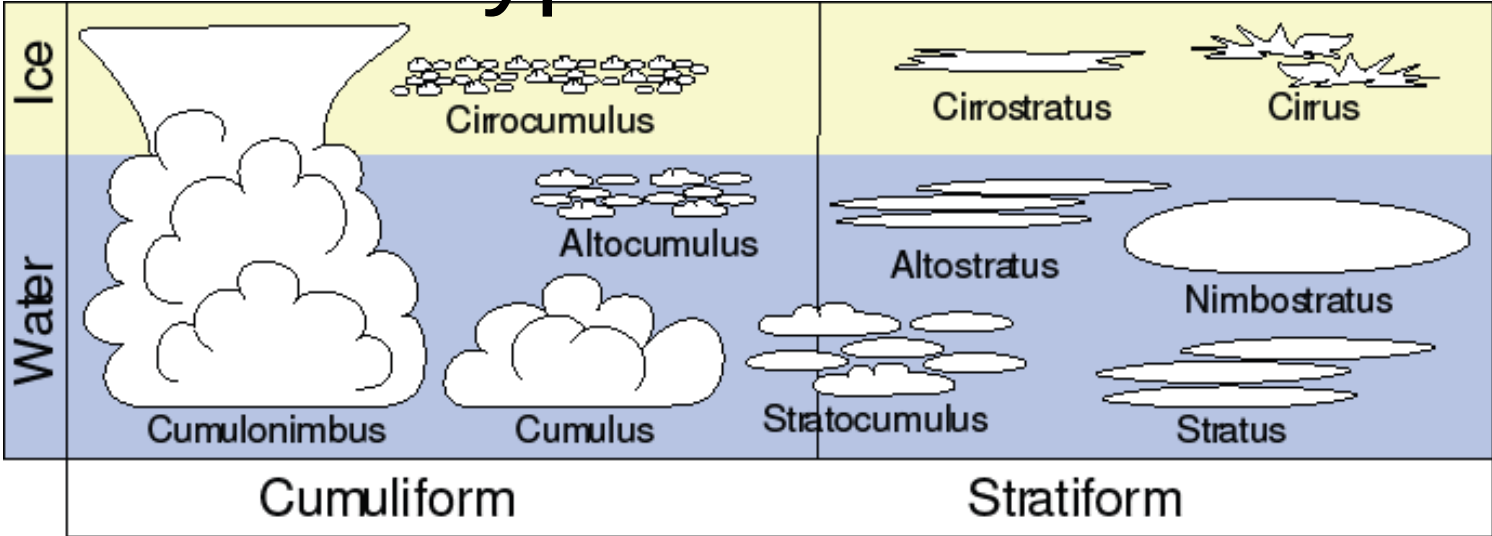
Cumulus



Altostratus



# Types of clouds



Cumuliform



Stratiform

# Edges



Hard, puffy



Wispy



Soft



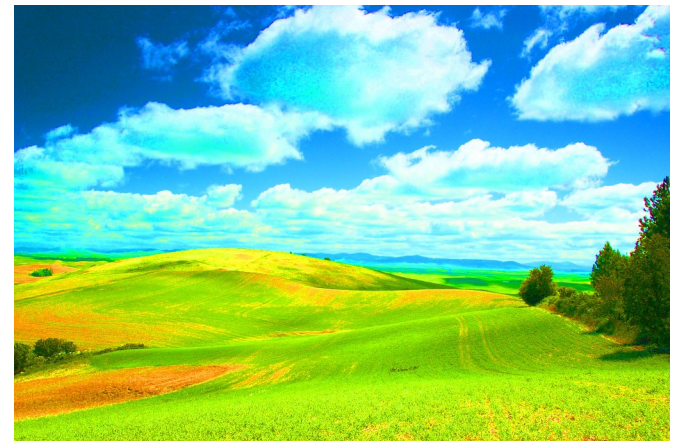
Close-up on a wisp



# Colors



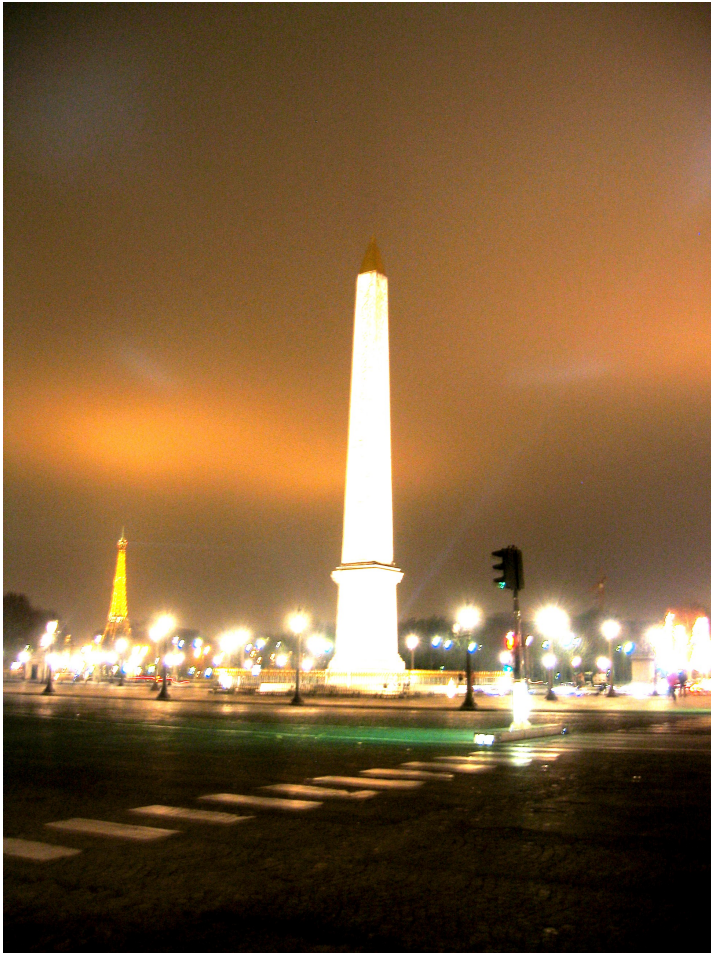
Sun + sky



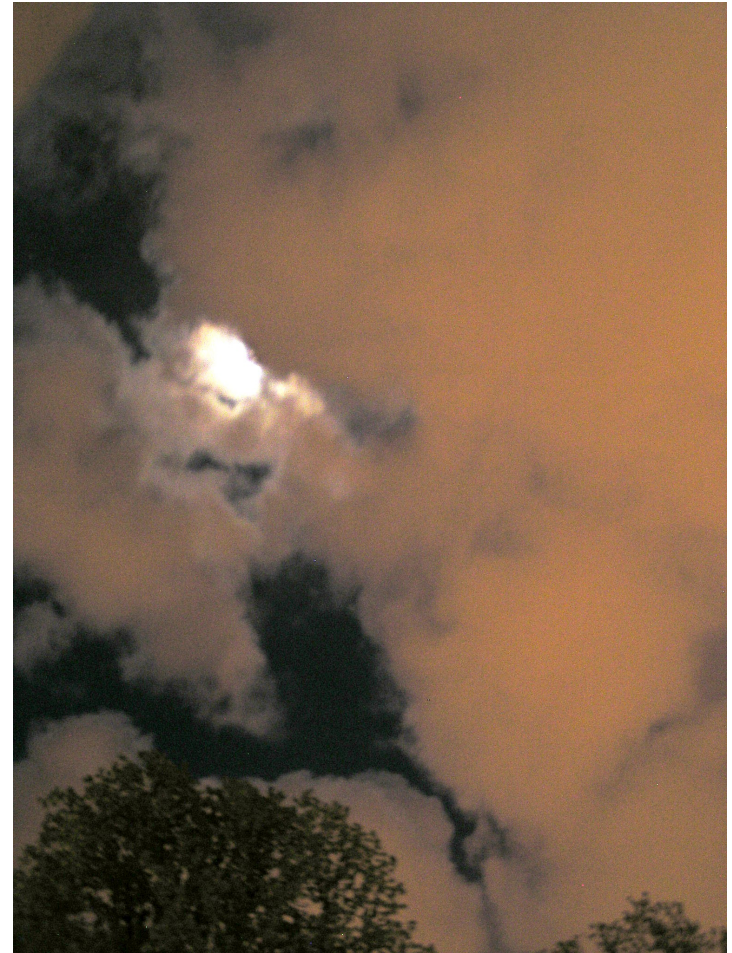
Sun + sky + ground



# Colors



Ground



Moon + sky + ground

# Colors



Sky + ground

# Colors

- Extreme cases: ice blink and water sky
  - Inter-reflections btw ground & clouds



Ice blink



Water sky



# Visual features

- Bright, crispy lit side
- Dark, soft unlit side



# Visual features

- Lit side:
  - Bright creases (faults between lobes)
  - Dark edges
  - Detailed





# Visual features

- Unlit side:
  - Silver lining (details!)
  - Dark, softer core



# Visual features

- Unlit side:
  - Silver lining (details!)
  - Dark, softer core



# Visual features

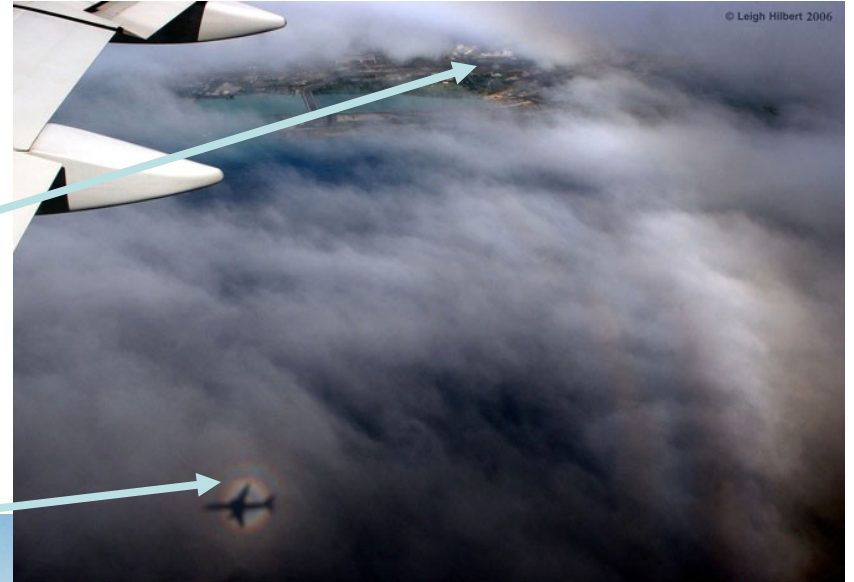
- Stratiform clouds:
  - "Pseudo-specular" effect



# Visual features

- Fogbow

- Glory



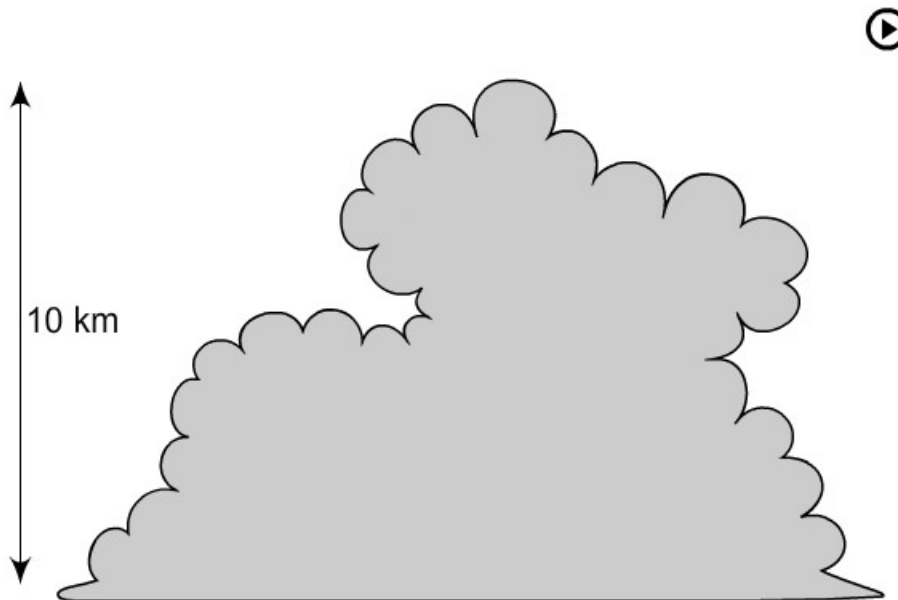
# What is a cloud ?

- 1. The human spectator point of view
- 1. The physics point of view**
- 1. The CG point of view



# Atmospheric physics

- A cloud = collection of microscopic water droplets
  - **Size of a cloud:** up to 20 km
  - **Concentration / density:** ~1 million droplets / liter
  - **Radius:** 5  $\mu\text{m}$  – 15  $\mu\text{m}$





# Scattering

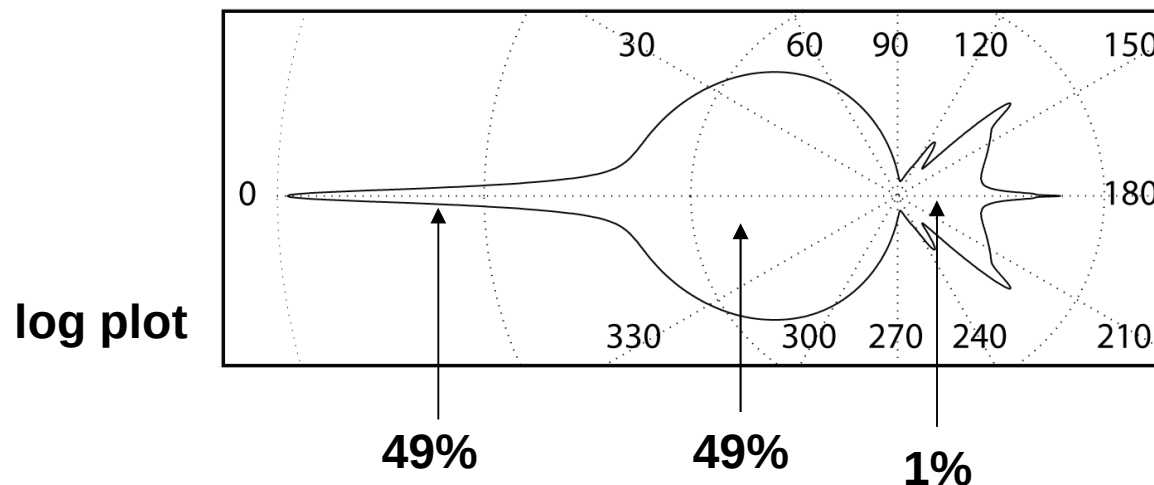
- Droplets scatter light
  - No absorption



# Scattering

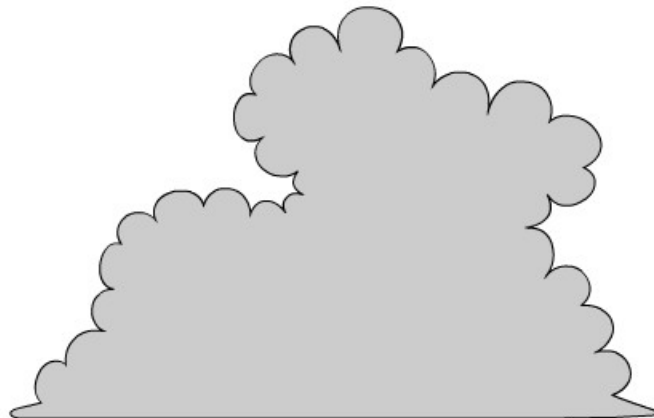
- Phase function

- Defined by Mie theory (validated reference)
- Strongly anisotropic
- Expensive to compute
- Depends on droplet size, temperature, wavelength...



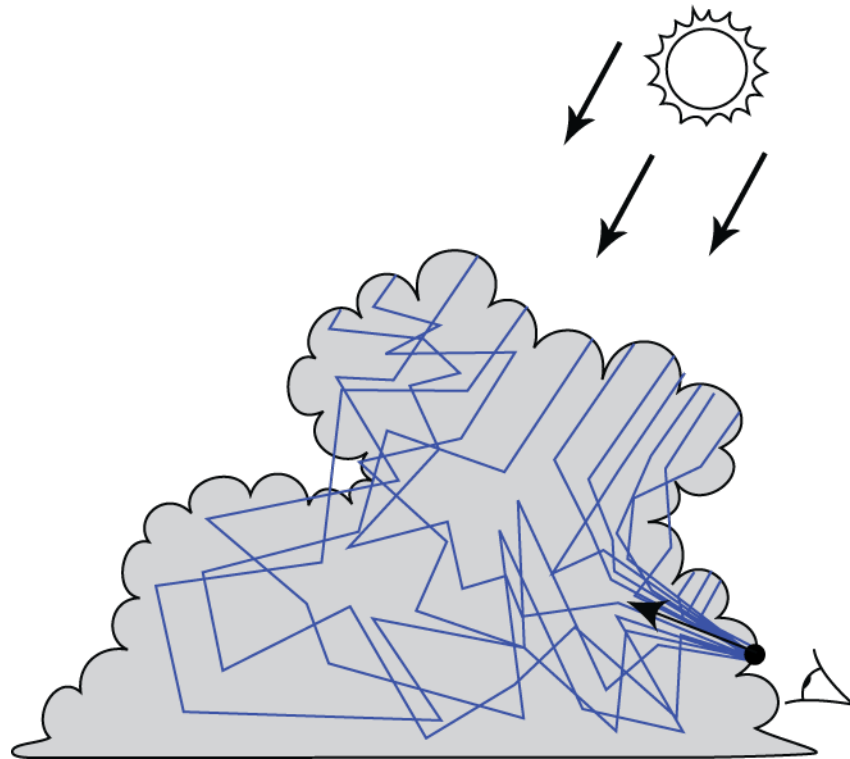
# Multiple scattering

- Path of a photon through a cloud



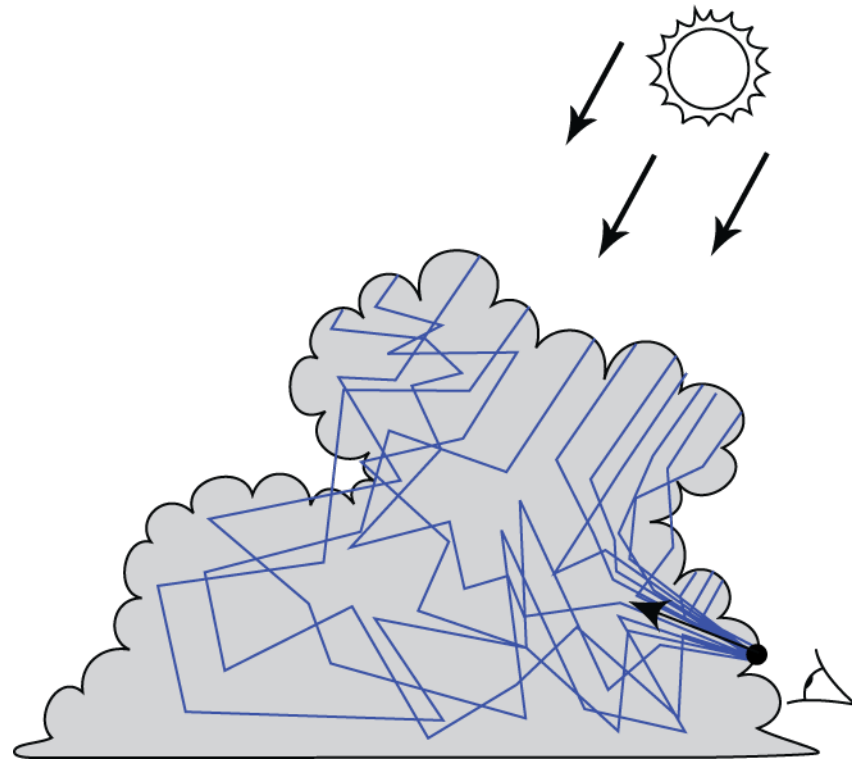
# Multiple scattering

- Many photons come from many paths



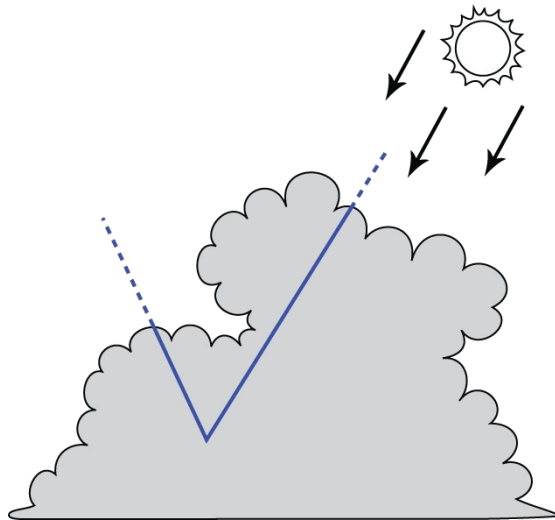
# Multiple scattering

- Rendering problem = how much light all these paths carry ?

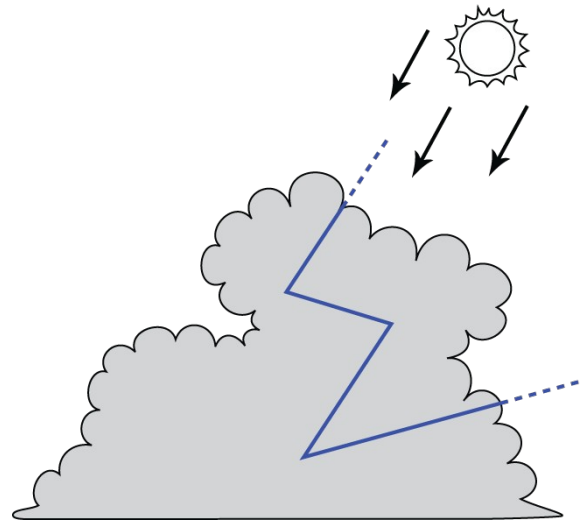


# Multiple scattering

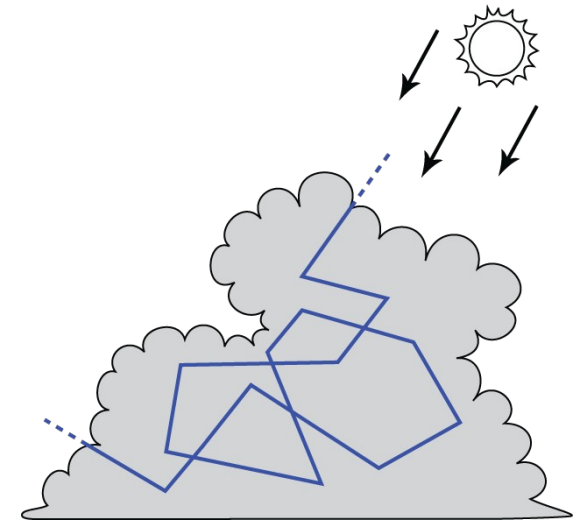
- Order of scattering
  - Number of bounces of a given path



Order 1



Order 3



Order 15



# What is a cloud ?

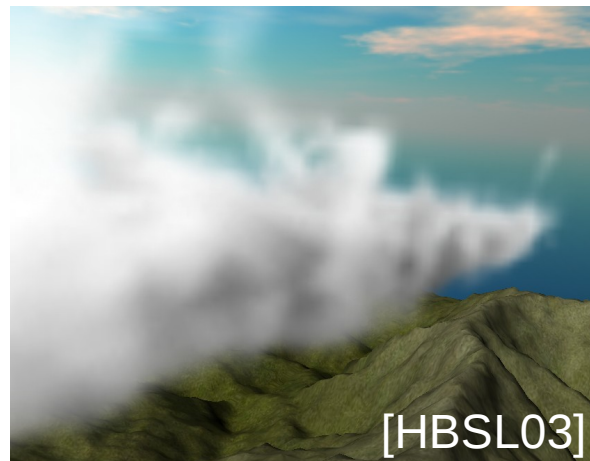
1. The human spectator point of view
2. The physics point of view
1. The CG point of view

# Computer graphics

- CG methods need:
  - A shape (i.e. droplet density field)
  - A phase function
  - A light transport model

# Shape models

- Radar data, simulation
  - 3D grids
  - Current resolutions:  $256^3$
  - Necessary resolution:  $>1000^3$
  - Too coarse
  - Blurry results



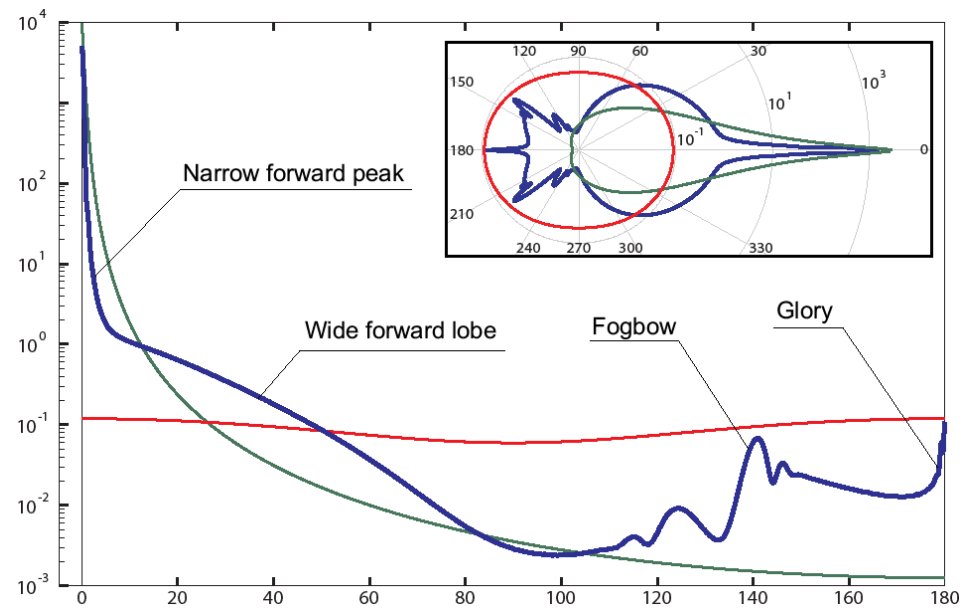
# Shape models

- Procedural methods
  - No resolution problems
  - Details added with Perlin noise



# Phase function

- **Mie function** [REHL03]
  - Precise
- **Henyey-Greenstein or Gaussian** [PAS03]
  - Less contrast, no glory, no fogbow
- **Rayleigh or constant** [MYND01,HBSL03]
  - No silver lining, less contrast



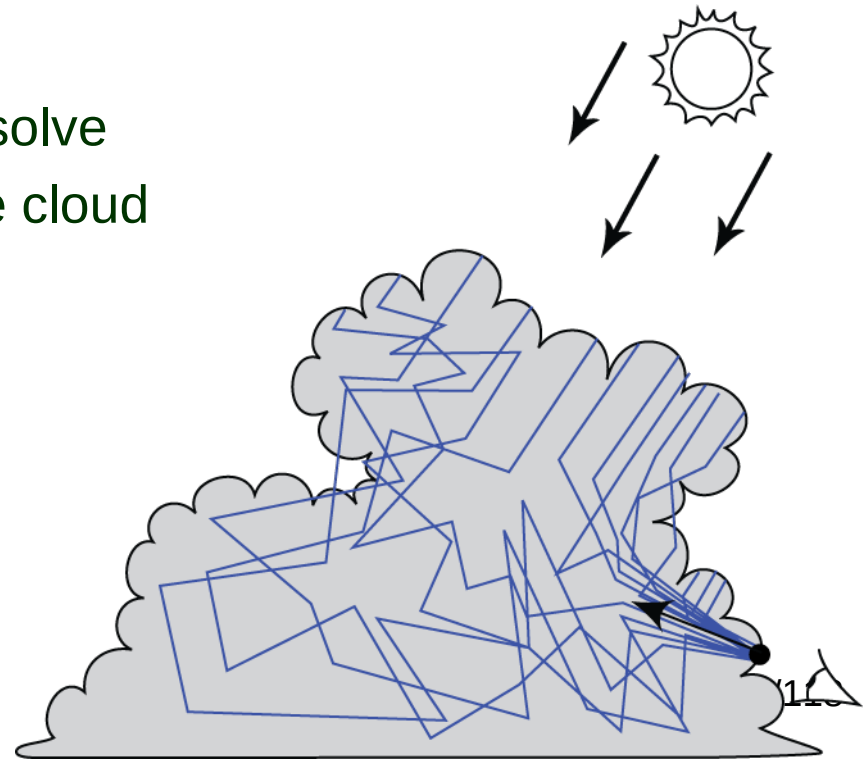
# Light transport models

- Solving the radiative transfer equation

$$-\frac{1}{\kappa_e(p)}(\vec{s} \cdot \nabla)L(p, \vec{s}) = L(p, \vec{s}) - \frac{1}{4\pi} \int_{\|\vec{s}'\|=1} P(\vec{s}, \vec{s}')L(p, \vec{s}')d\vec{s}'$$

- Very costly
  - 6D integro-differential system to solve
  - = computing all paths through the cloud

→ Not realtime



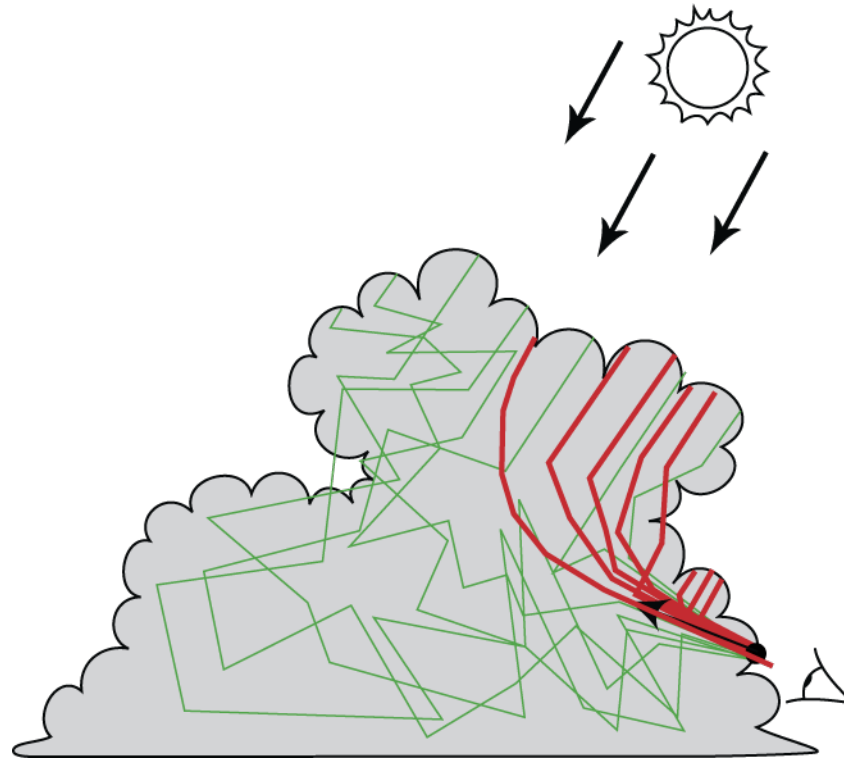
# Realtime light transport models

- Simplify the problem
  - Ignore sky and ground contributions
- Find an approximate solution
  - Do not consider all paths

# Realtime light transport models

- Consider only multiple forward scattering
  - i.e. short, low-turning paths

[DKY<sup>+</sup>00, HL01, REHL03, PAS03, KPHE02, HBSL03, SSEH03, REK<sup>+</sup>04]





# Realtime light transport models

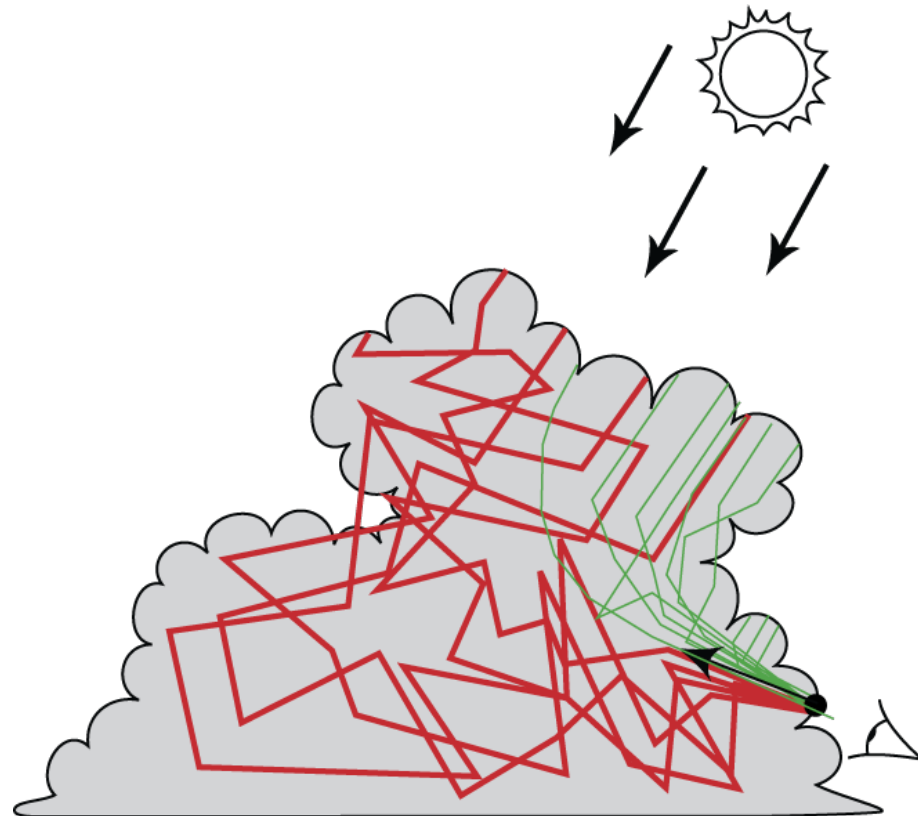
- Consider only multiple forward scattering
  - Reproduces silver lining
  - Underestimates diffusion effects
    - No bright lit side
    - No dark soft unlit side
    - No strong contrast



# Realtime light transport models

- Consider only diffusion
  - i.e. long, high-order paths

[Sta95]



# Realtime light transport models

- Consider only diffusion
  - Reproduces diffusion effects
  - Ignores view-dependent effects
    - No silver lining



# Realtime light transport models

- Summary
  - Existing models are either
    - Not reproducing all features
    - Too costly
    - Too blurry
  - Realism & realtime not reached yet

# Realtime light transport models

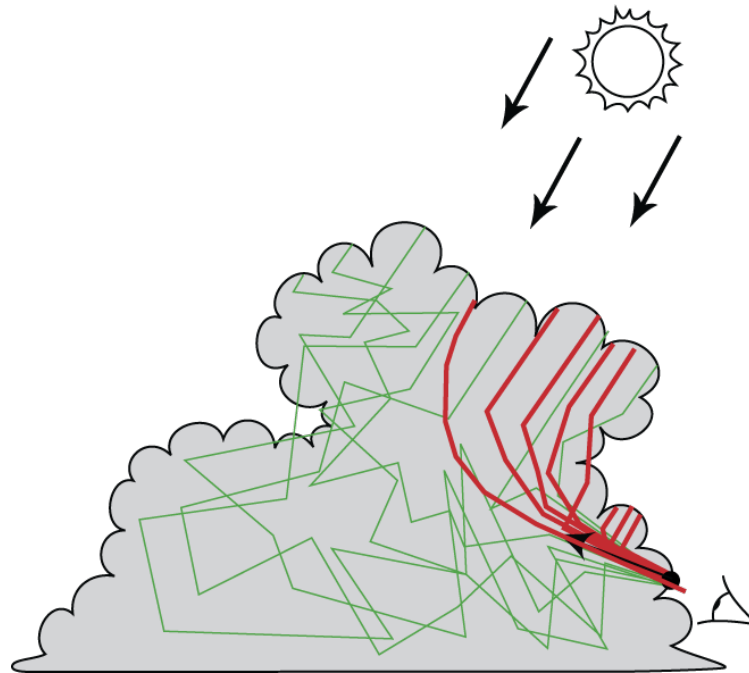
- We need a better light transport model that
  - Reproduces all visual features
  - Is efficient
  - Handles detailed complex shapes

# Our approach

General idea

# General idea

- Existing methods
  - Rely on microscopic models
  - Compute paths inside the cloud
- The eye does not see inside the cloud



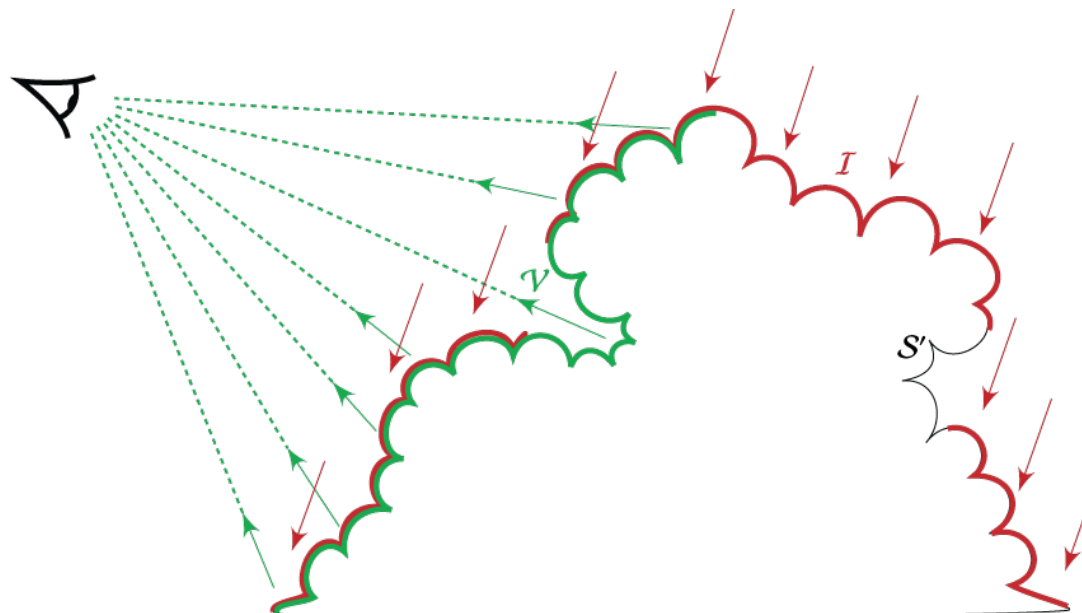
# General idea

- Existing methods
    - Rely on microscopic models
    - Need to compute paths inside the cloud
  - The eye does not see inside the cloud
- This is costly and redundant
- We want to avoid computing paths inside the volume



# General idea

- Our solution: consider **surface to surface** light transport
  - From **lit surface** to **viewed surface**
  - Do not walk through the volume
  - Rely on a **mesoscopic** model



# Overview

- Part I:
  - Identify visual features
  - Study previous works
- Part II:
  - Study of light transport in clouds
- Part III:
  - New rendering methods for clouds

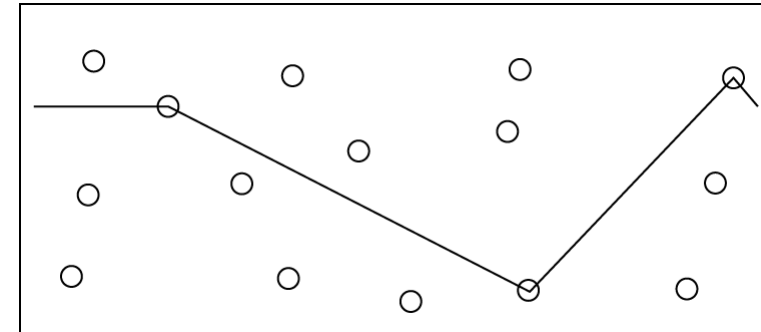
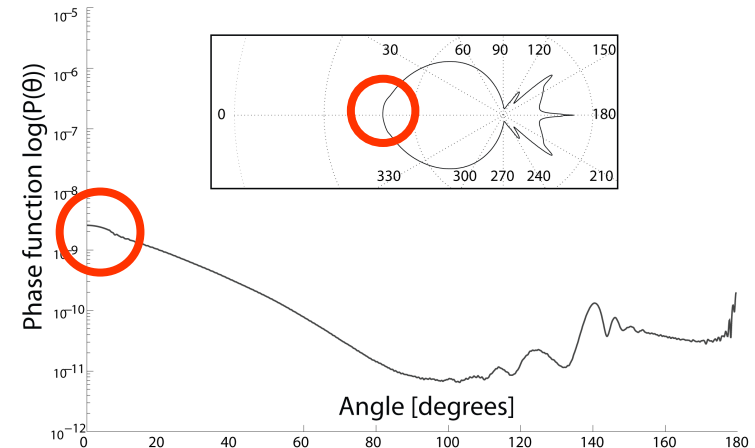
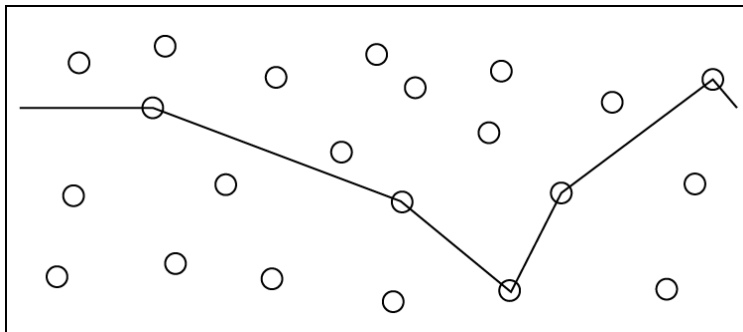
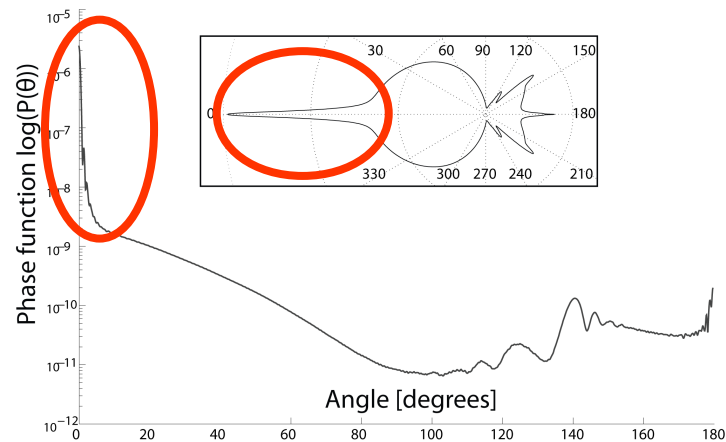
# Overview

- Study light transport in a simple shape: a slab
  - Homogeneous density
  - Phase function: modified-Mie



# Phase function

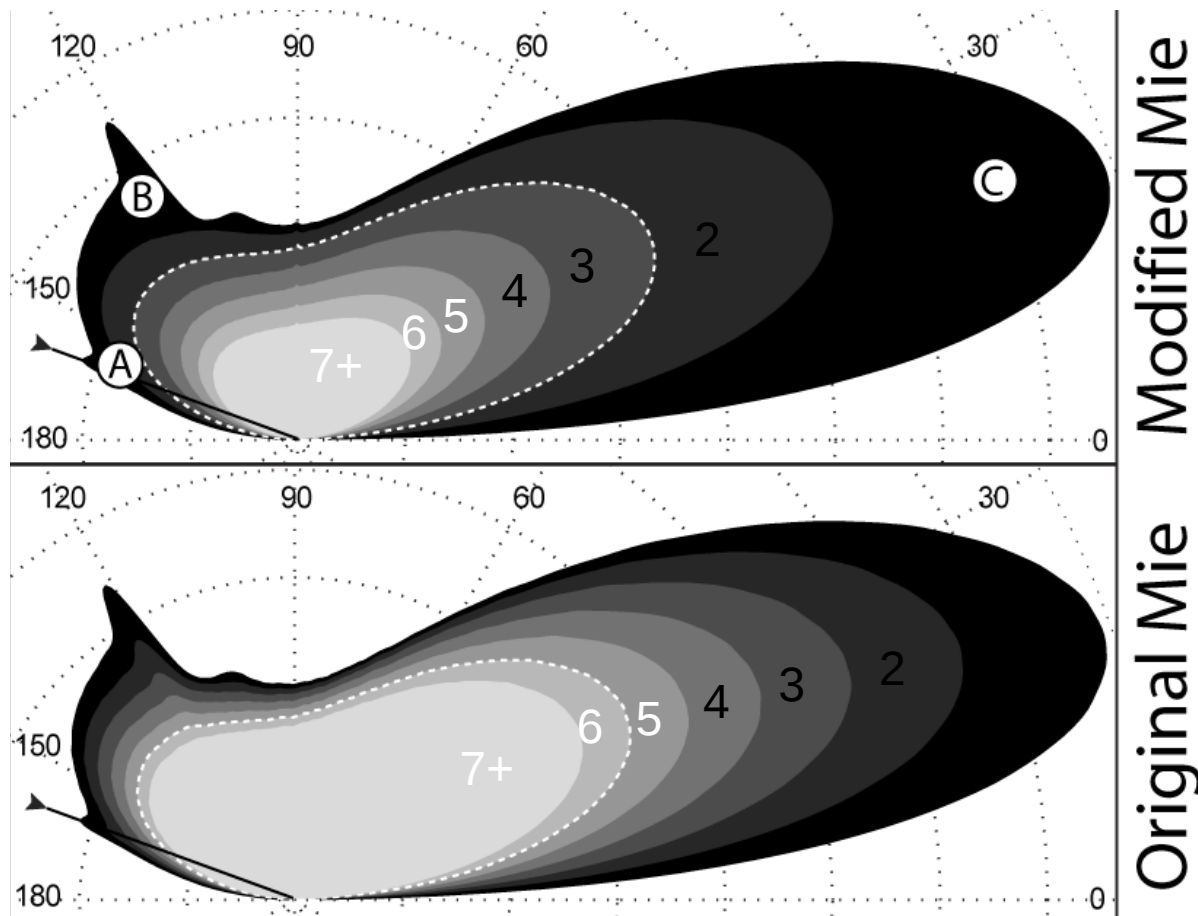
- Our Modified-Mie model
  - Strong forward peak → transparency



# Phase function

- Modified-Mie model validation

BRDF of  
a slab

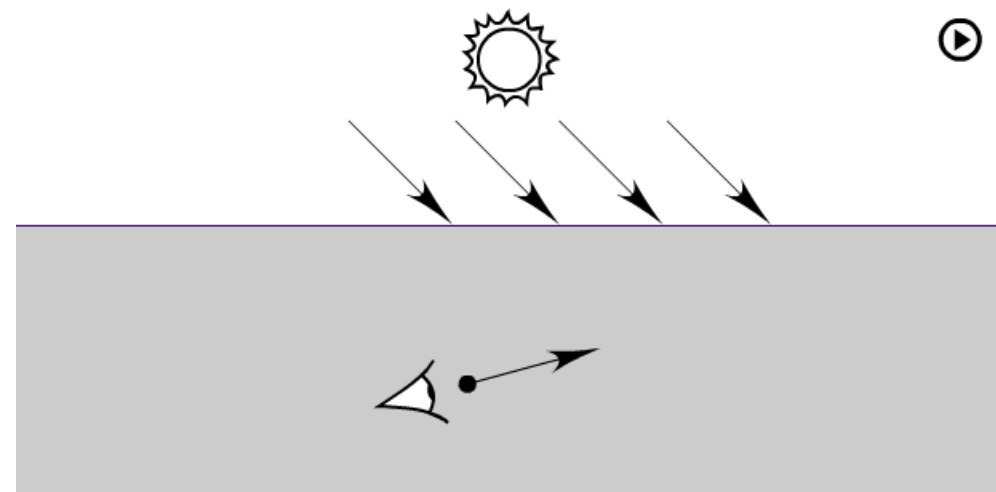




# Overview

- What we study:
  - For every possible
    - Slab thickness
    - Lighting angle
    - View angle
    - Viewpoint depth
    - Scattering order
  - Measure
    - Luminance
    - Where light came from

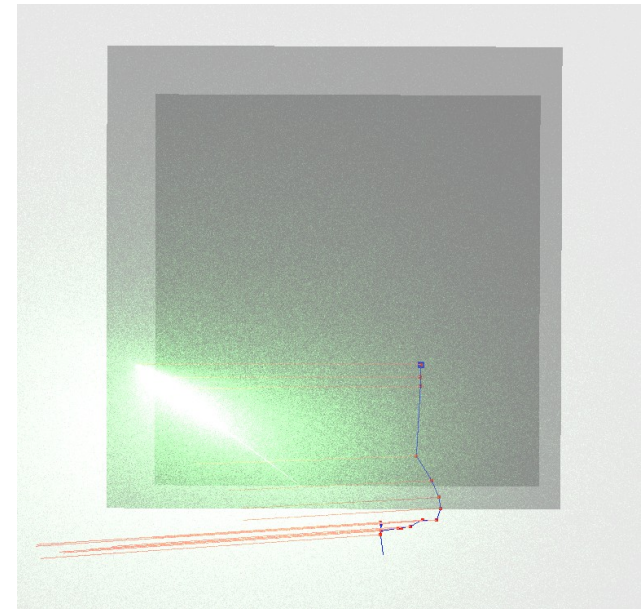
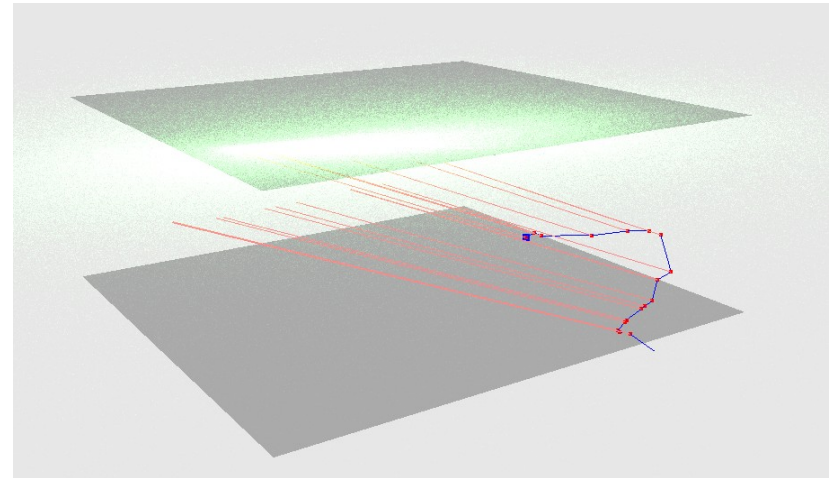
6D → 5D function



# Overview

- What we study:
  - For every possible
    - Slab thickness
    - Lighting angle
    - View angle
    - Viewpoint depth
    - Scattering order
  - Measure
    - Luminance
    - Where light came from

6D → 5D function



# Experimentation

- Parallel Monte Carlo path tracing
- On supercomputers (clusters)
- Storing results in a database
- A few weeks of computations
- A few dozens of gigabytes of data

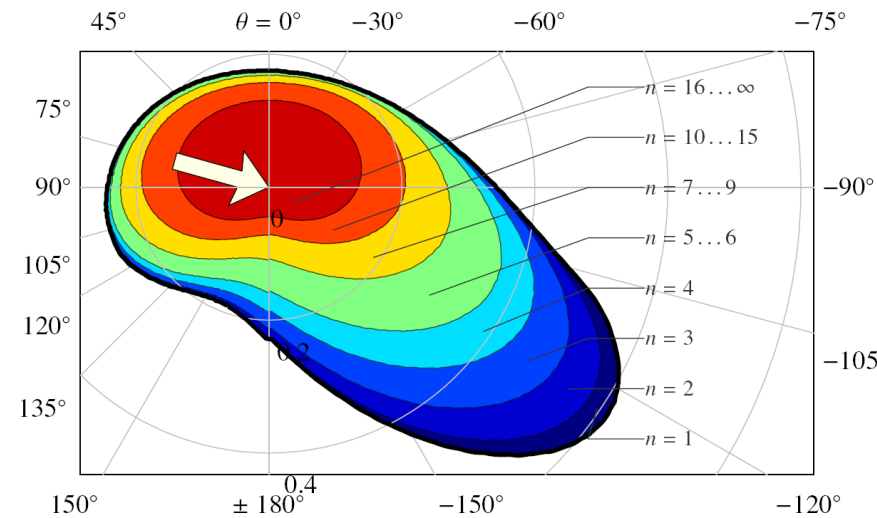
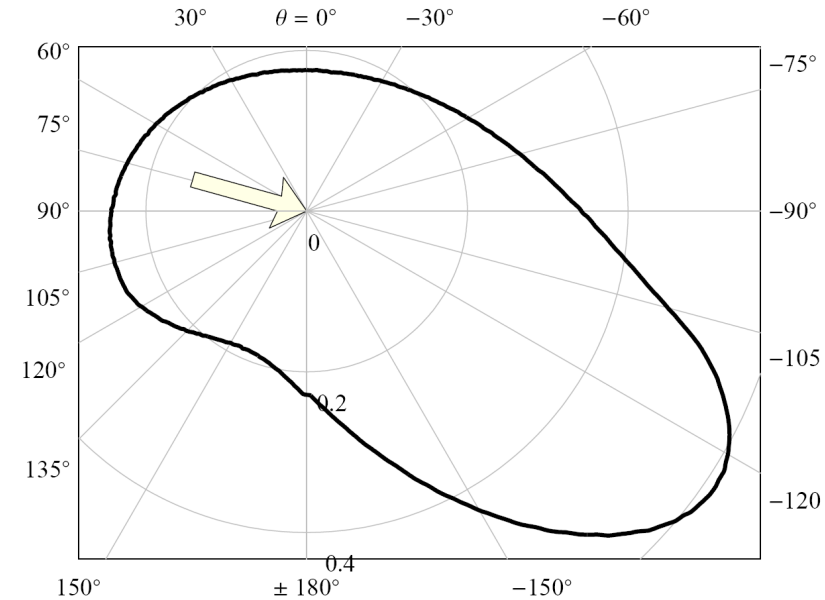
# Results

- Mesoscopic model of light transport in a slab
    - For any lighting condition
    - Luminance at any point
    - No need for complex computations
    - The answer is immediate
- + A lot of interesting information

# Analysis of results

- Tools for analysis
  - BSDF (Bidirectional Scattering Distribution Function)  
(Similar to a BRDF)

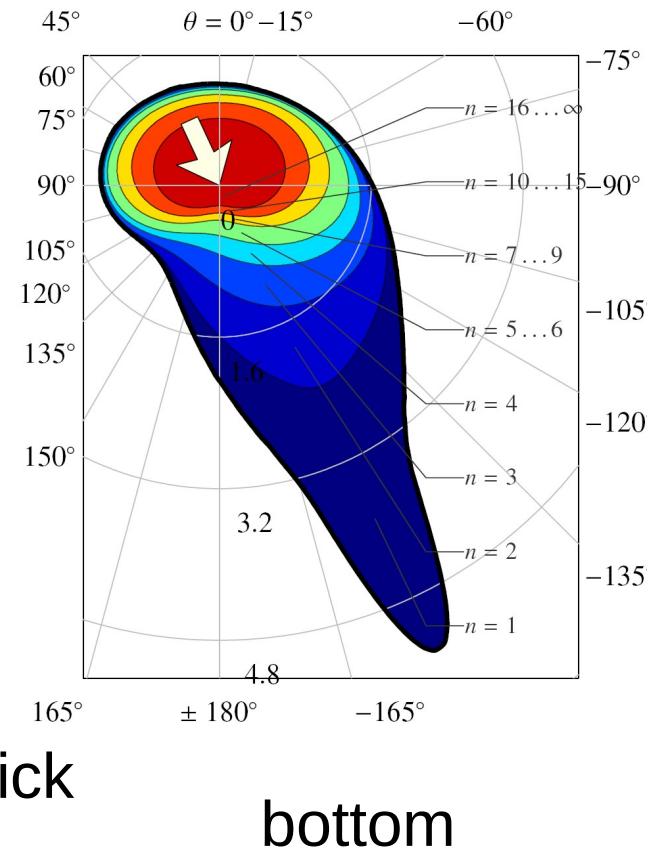
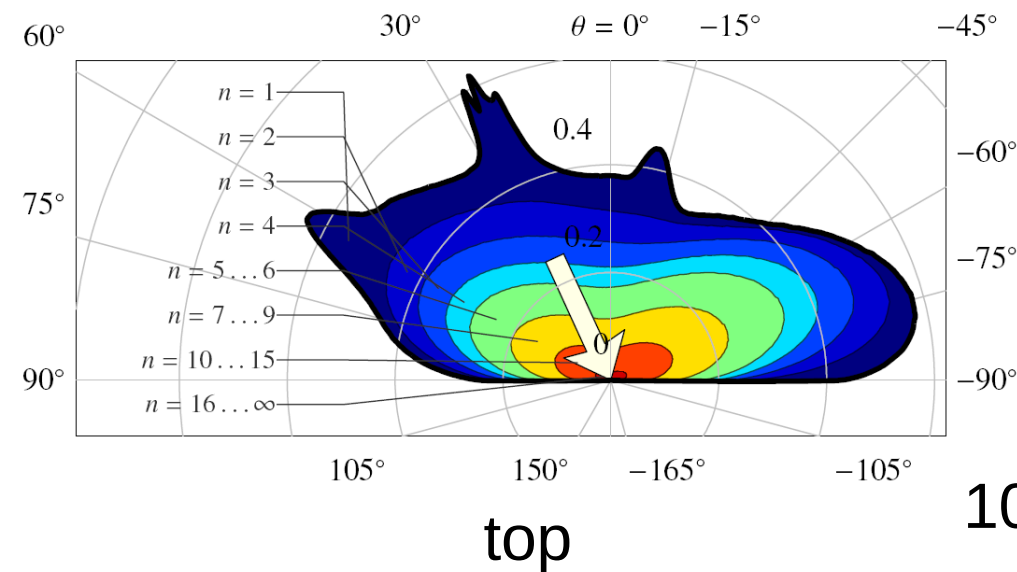
- With order-dependence





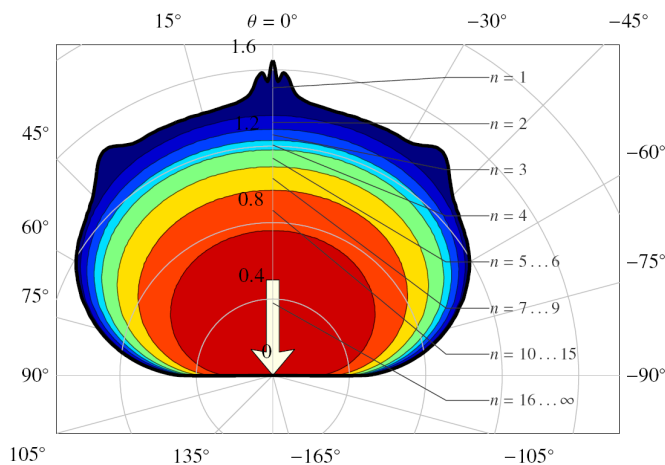
# Analysis of BSDF

- In thin slabs
  - Low orders
  - Very anisotropic
  - Strongly forward



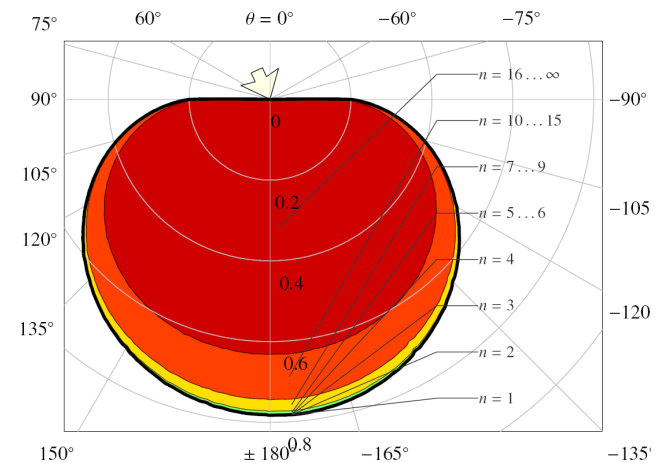
# Analysis of BSDF

- In thick slabs
  - At bottom
    - High orders
    - Very diffusive
  - On top
    - Low and high orders
    - Diffusive + anisotropic features



top

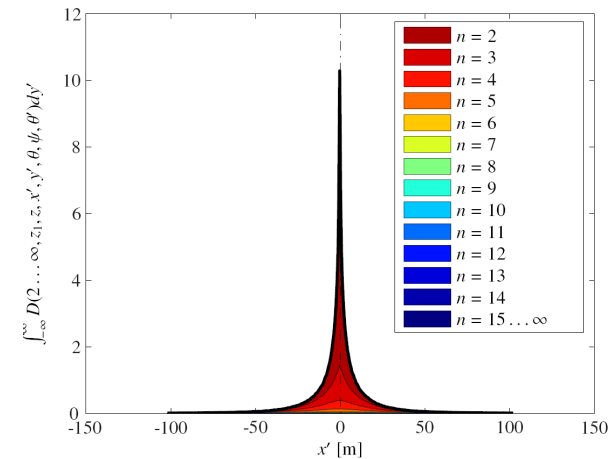
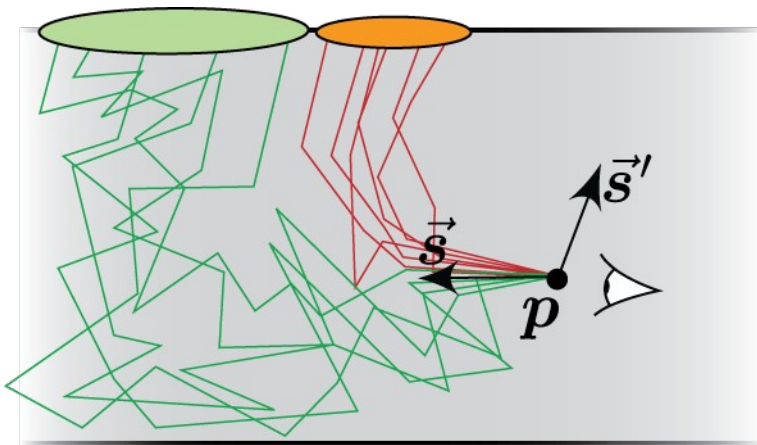
500m thick



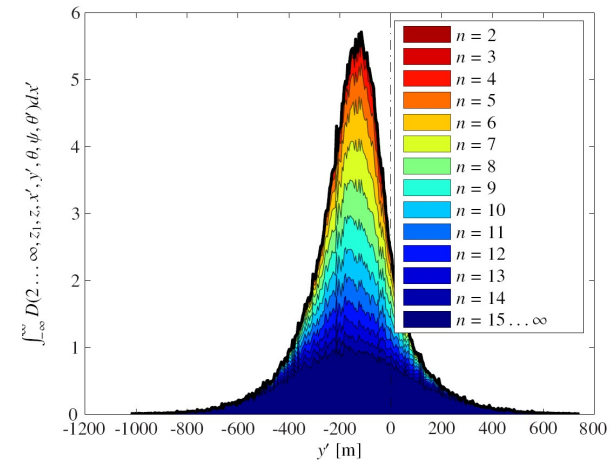
bottom

# Analysis of entry points of light

- For each order
  - Limited area
  - Disk-shaped
  - Gaussian distribution



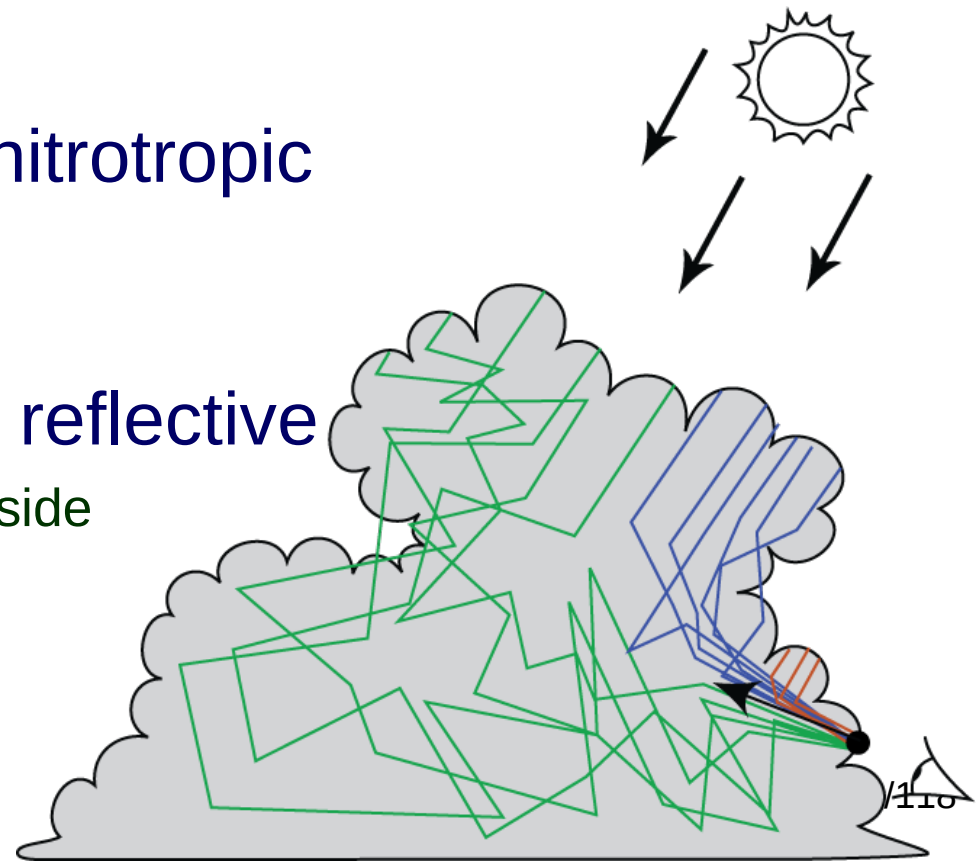
100m



200m

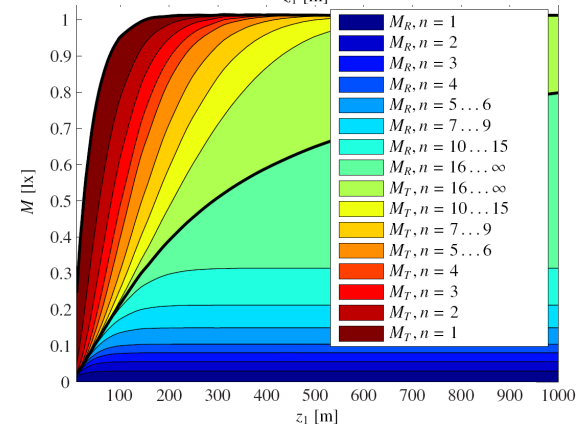
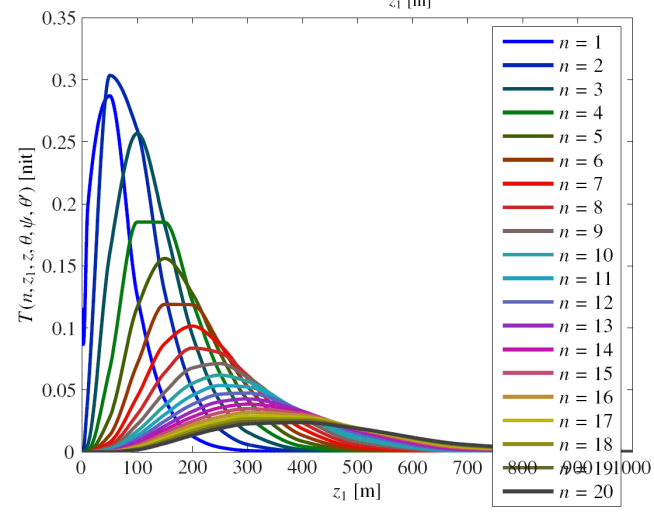
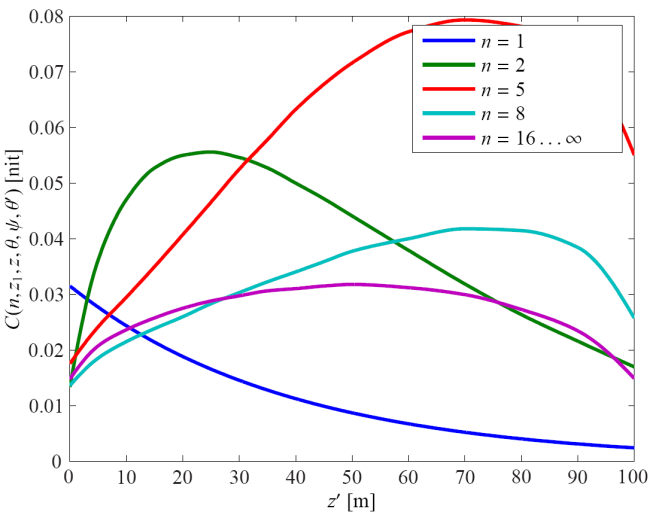
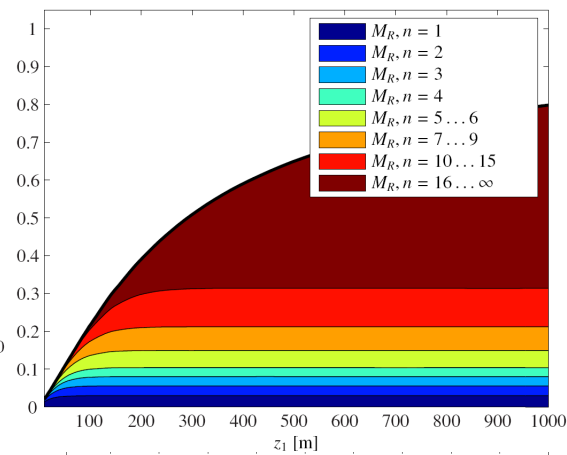
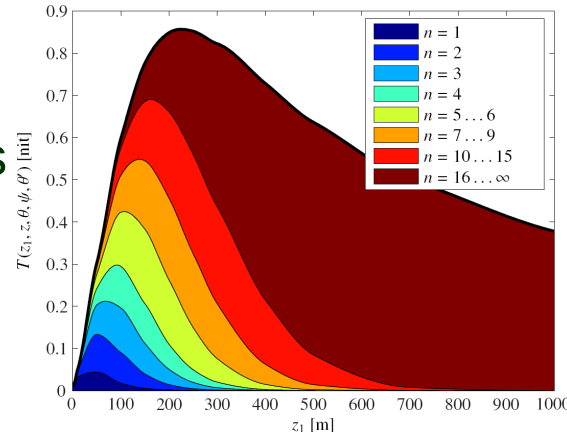
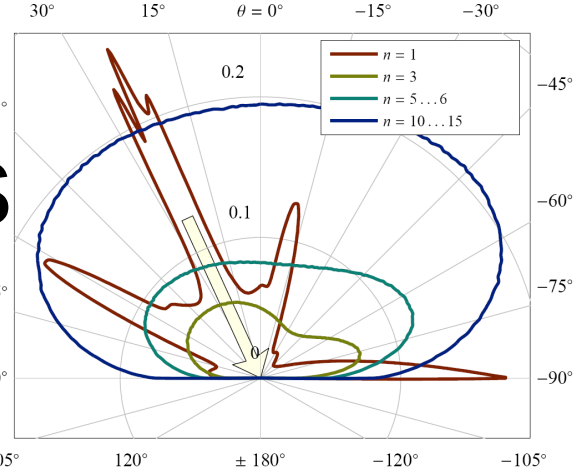
# For real clouds

- **Each order**
  - Has a different behavior
  - Brings different features
- **Low orders: short & anitotropic**
  - Silver lining, dark edges
- **High orders: diffusive, reflective**
  - Bright lit side, soft dark unlit side



# Other analyses

- Many other observations
  - Asymptotic regimes
  - Common behaviors
  - Logarithmic behaviors
  - etc...



# Using the results

- Mesoscopic model of light transport in a slab
  - $\Psi$ : 5D → 6D
  - In a **huge** table (20GB)
  - Does not fit on GPU
- Other representations are possible
  - Procedural function
  - Standard compression
  - Decomposition (Wavelet / Sph. harmonics / Fourier)
  - ...

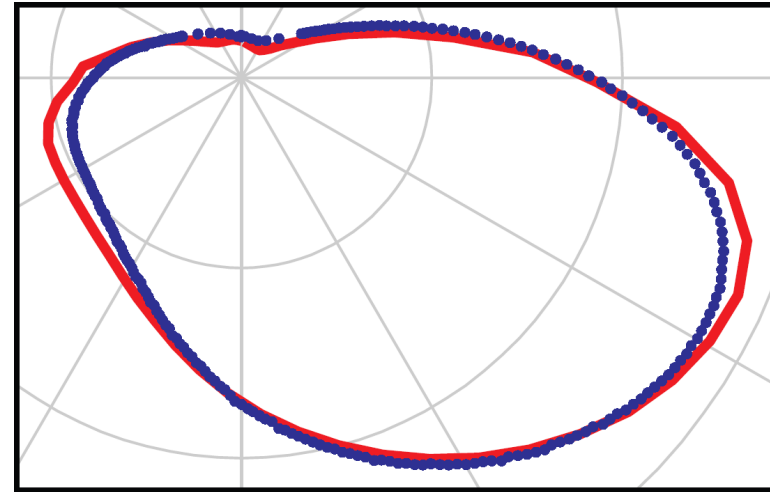
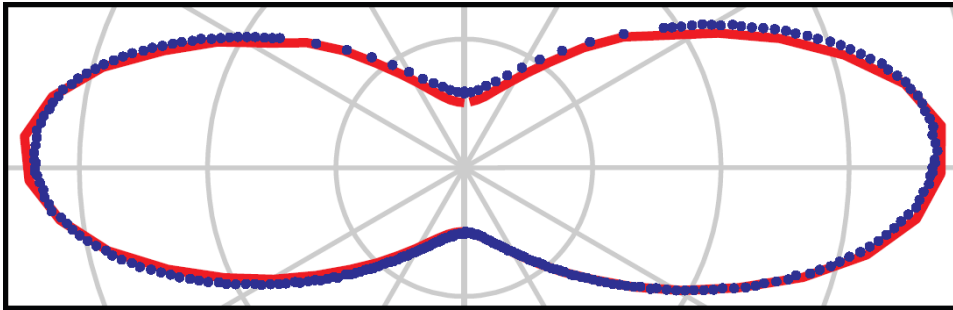
# Using the results

- We found something
  - Procedural model
    - A few functions
    - Several 2D tables
  - 20GB → 2MB (10,000:1)

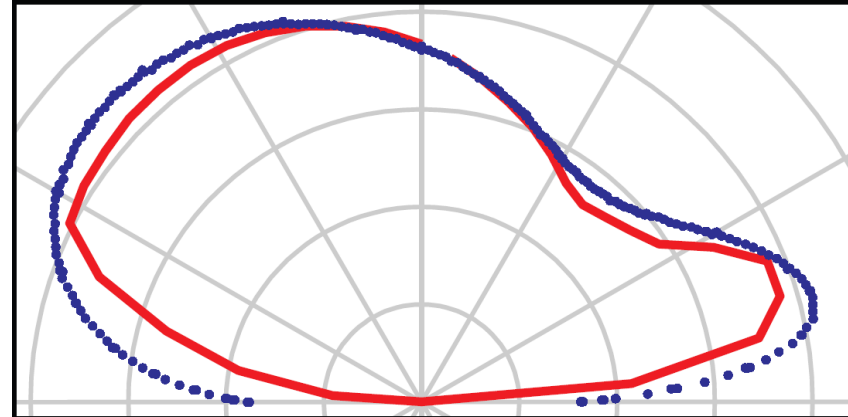
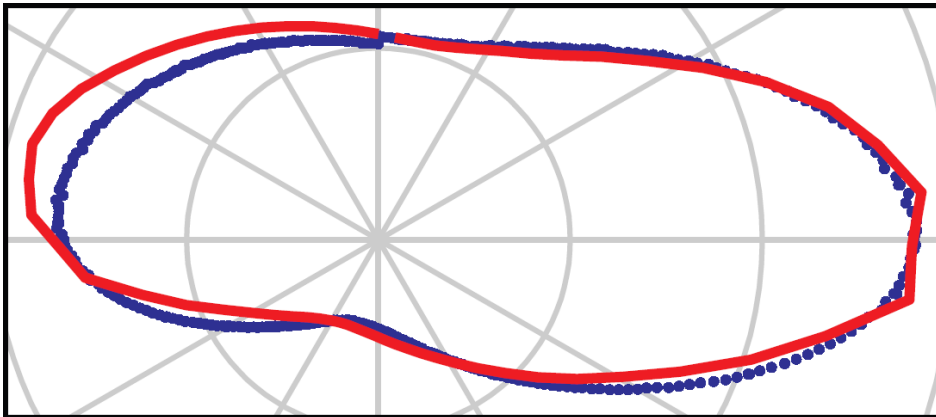


# Using the results

- We found something
  - Fits the data well



**Monte-Carlo vs. Our compression**



# Summary

- Lots of information
- Complete mesoscopic model of light transport **in a slab**
- How to make clouds with this ?
- Not published (yet), physics journal ?

# Overview

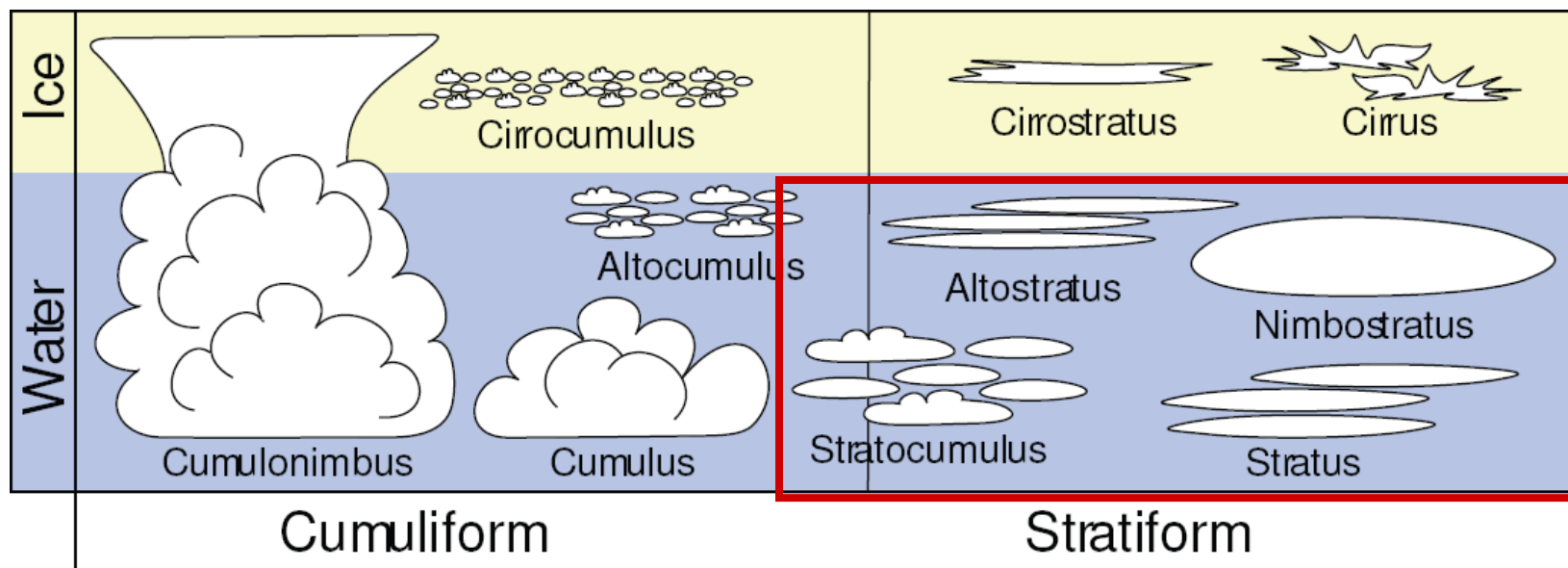
- Part I:
  - Identify visual features
  - Study previous works
- Part II:
  - Study of light transport in clouds
- **Part III:**
  - New rendering methods for clouds

# Part III

Two new clouds rendering models

# New clouds rendering models

1. Model for stratiform clouds [BNL06]
1. General model [BBNM07]

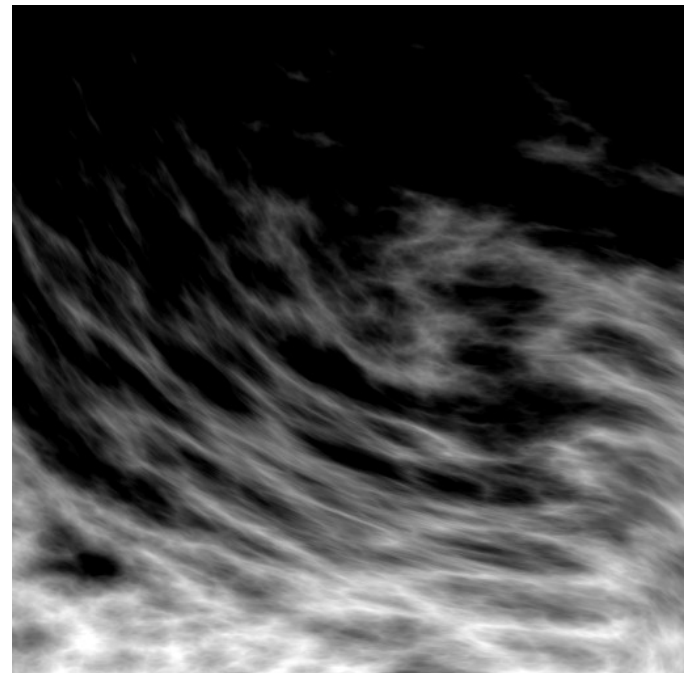
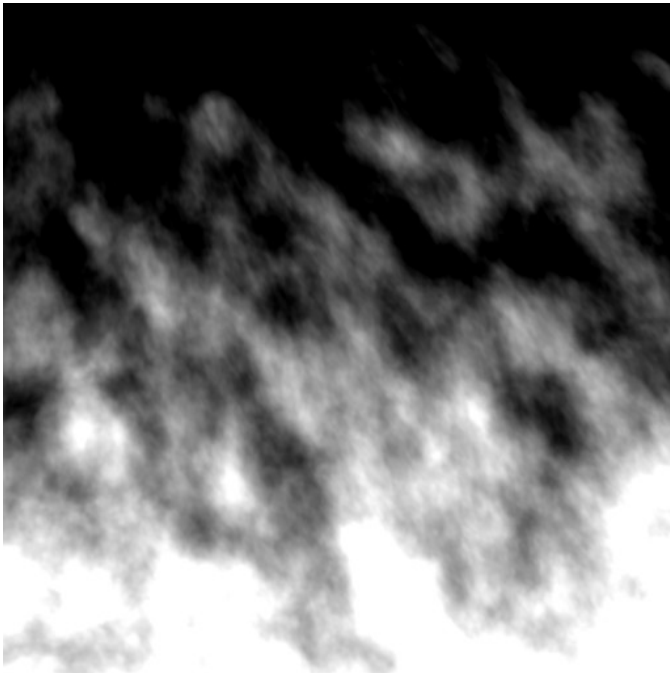


# Overview

- Shape model
- Light transport model

# Shape

- Height field
  - Suitable
  - Lot of detail
  - Low memory
  - Animatable



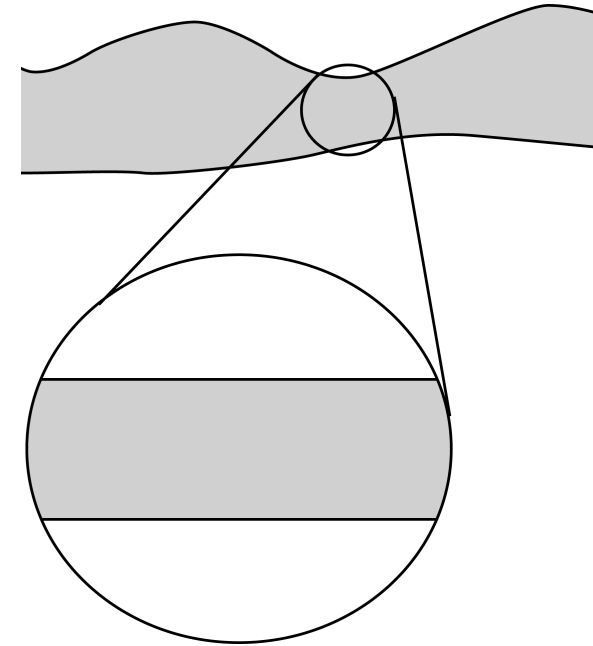


# Light transport model

- Assumptions

- Homogeneous
- Stratiform

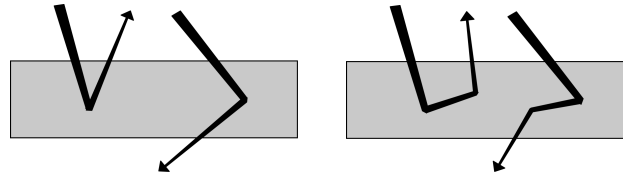
→ **locally equivalent to a slab**



## General idea

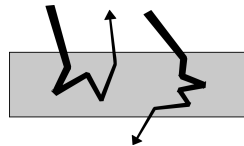
- Treat separately different orders
  - Low orders → highly anisotropic features

- Order 1 & 2

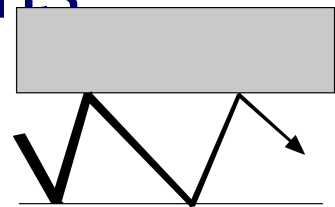


- High orders → diffusive features

- Orders > 3

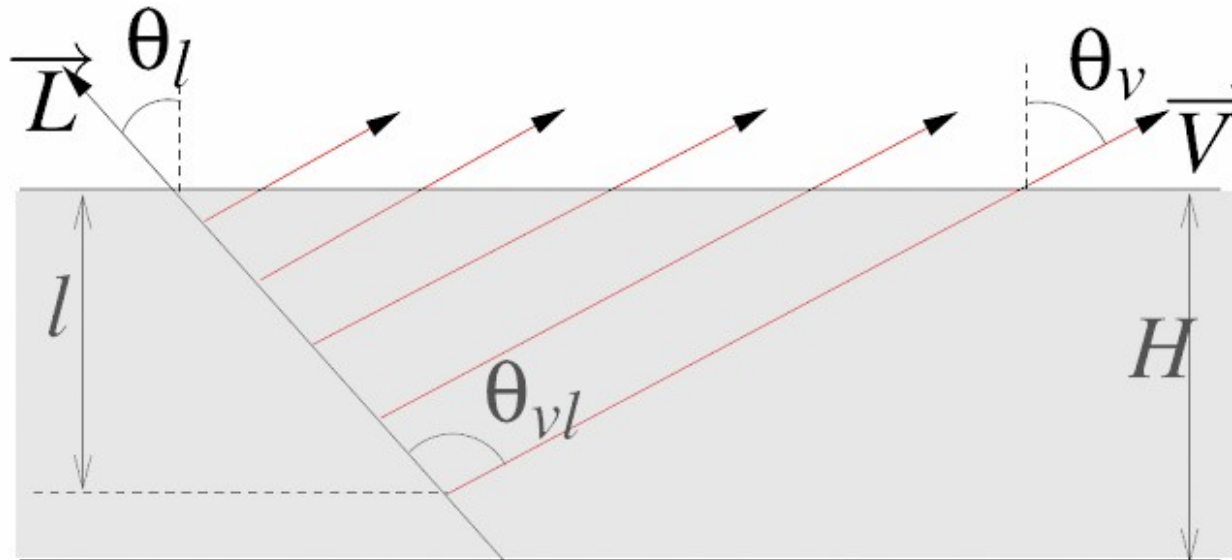
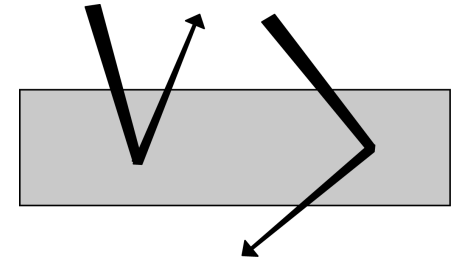


- Account for sky and ground illumination
- Account ground-clouds inter-reflections



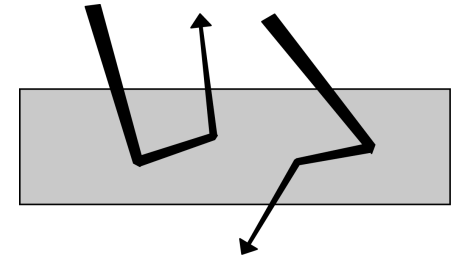
# Order 1

- Analytical solution

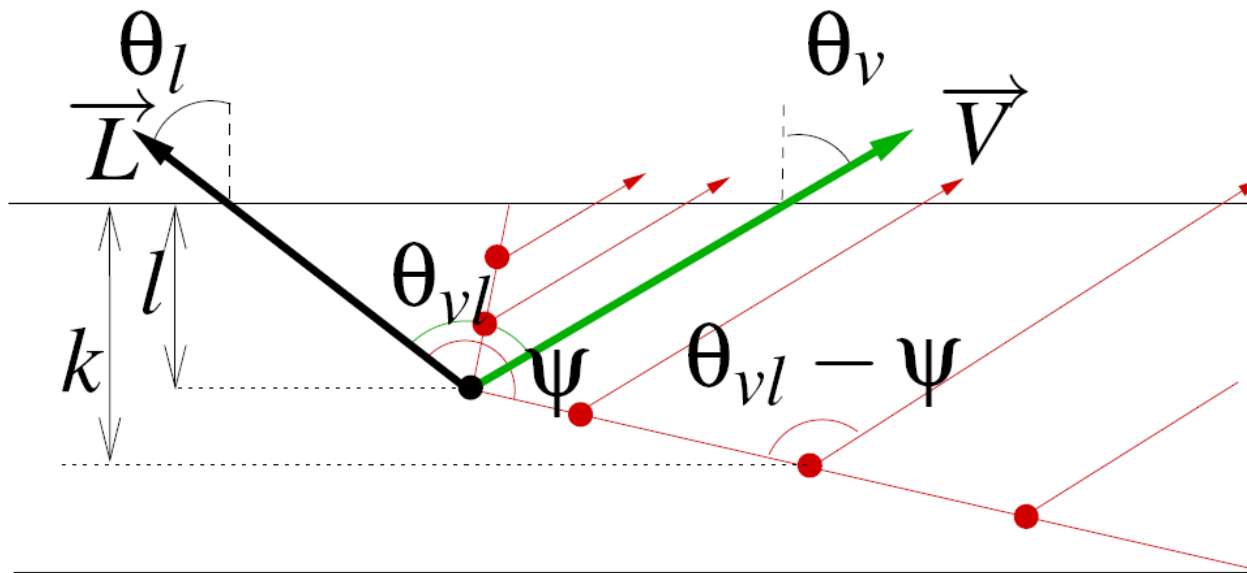


$$\frac{\kappa P(\theta_{vl}) \mu_l}{\mu_v + \mu_l} (1 - \tau(H_l + H_v))$$

## Order 2



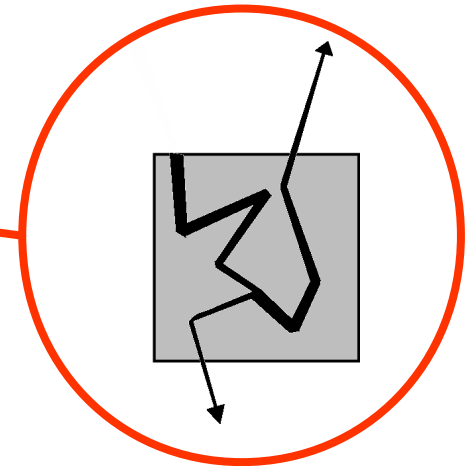
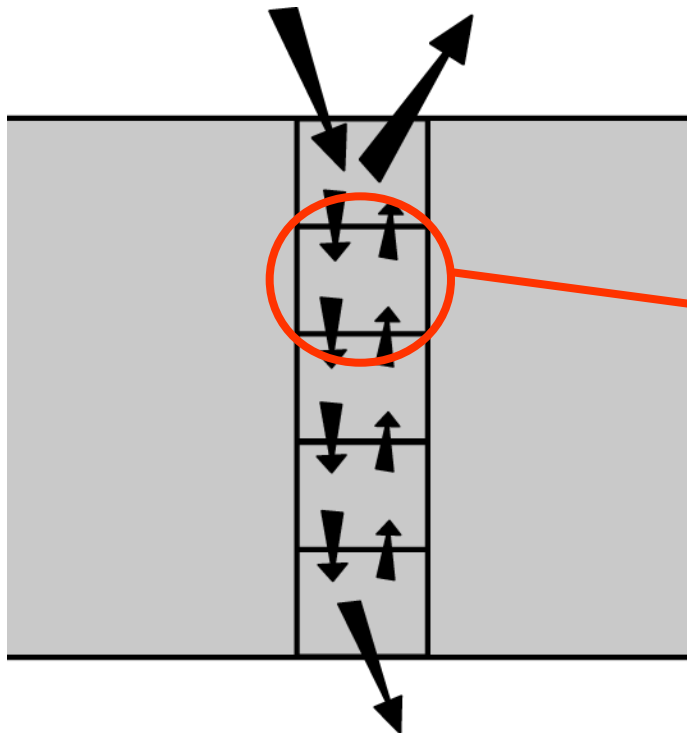
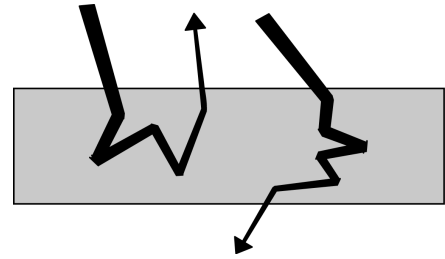
- Approximation by convolution



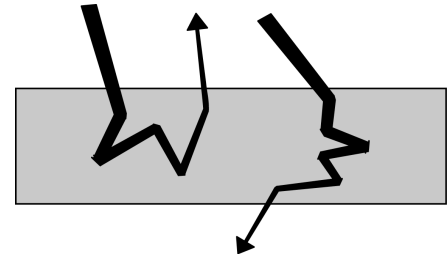
$$\frac{\kappa^2 \bar{P}^2(\theta_{vl}) \mu_l}{\mu_v + \mu_l} (1 - \tau(H_l + H_v))$$

## Orders $> 3$

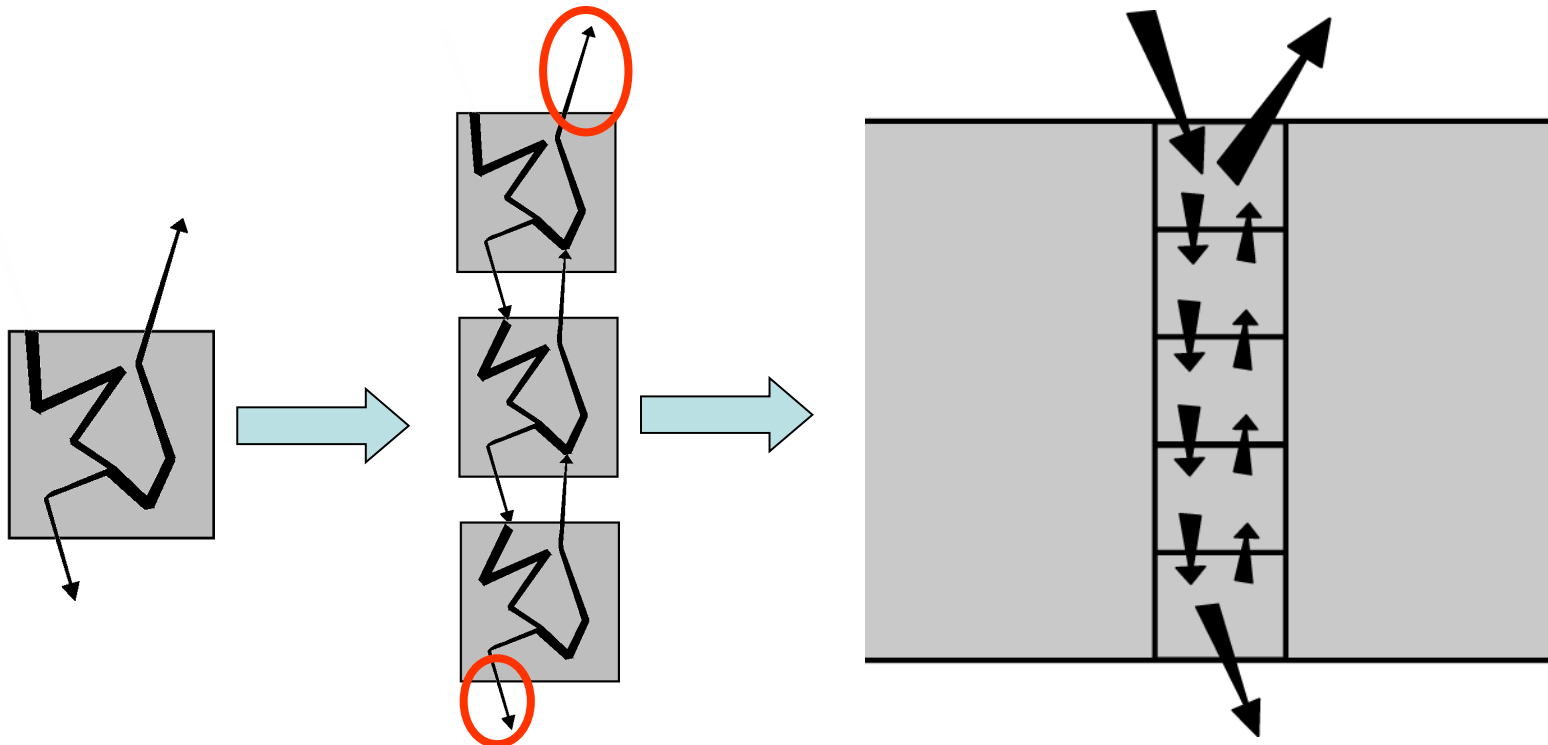
- The "doubling-adding" method
  - Simple method for slabs
  - A slab = a column of cells



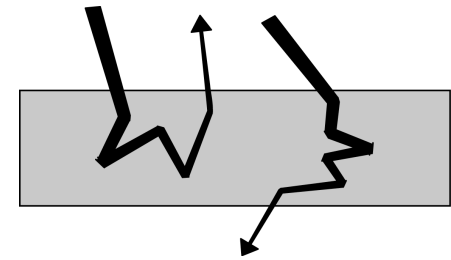
## Orders $> 3$



- The "doubling-adding" method
  - Diffusion → simple interactions between cells
  - Know one cell → know the whole column



## Orders $> 3$



- The "doubling-adding" method
  - Analytic & efficient
  - But **limited** to simple phase functions
- "Corrected" doubling-adding method
  - Accounts for anisotropy
  - Deduced from study

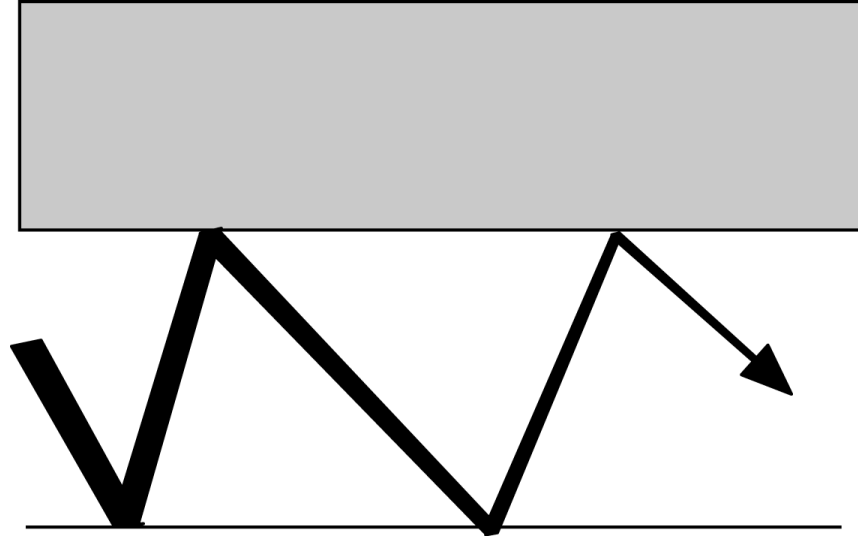


# Sky and ground sources

- Also use the doubling-adding method
- Use diffuse sources

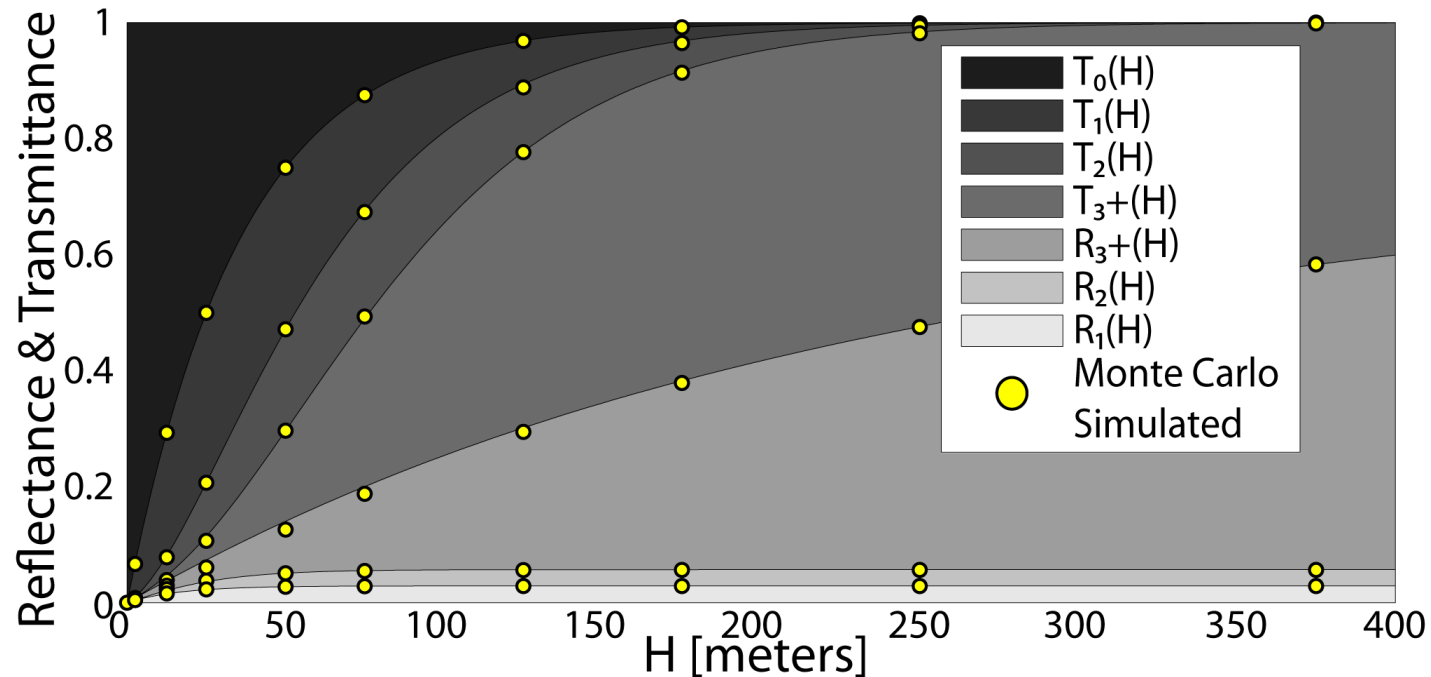
# Ground-clouds inter-reflections

- New radiosity method
  - Realtime
  - For parallel textured planes
  - On GPU



# Validation

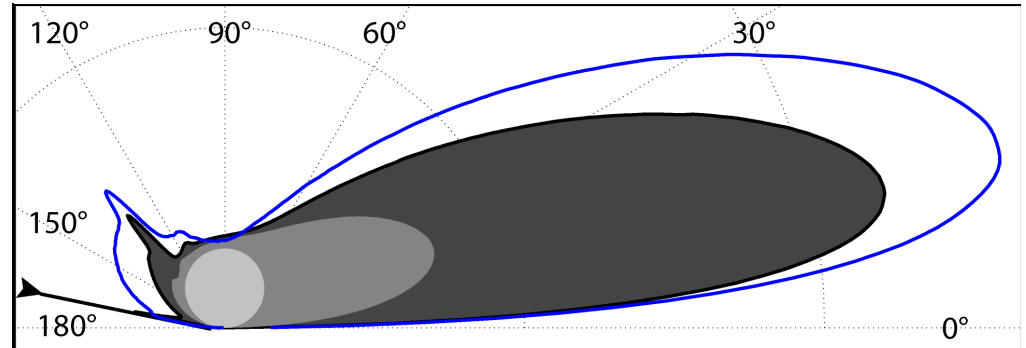
- Corrected doubling-adding method



# Validation

- Light transport model

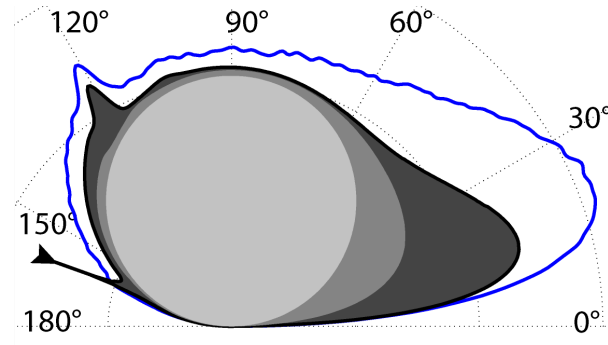
5m-thick slab



BRDFs

- Monte-Carlo
- Our model

100m-thick slab



# Implementation

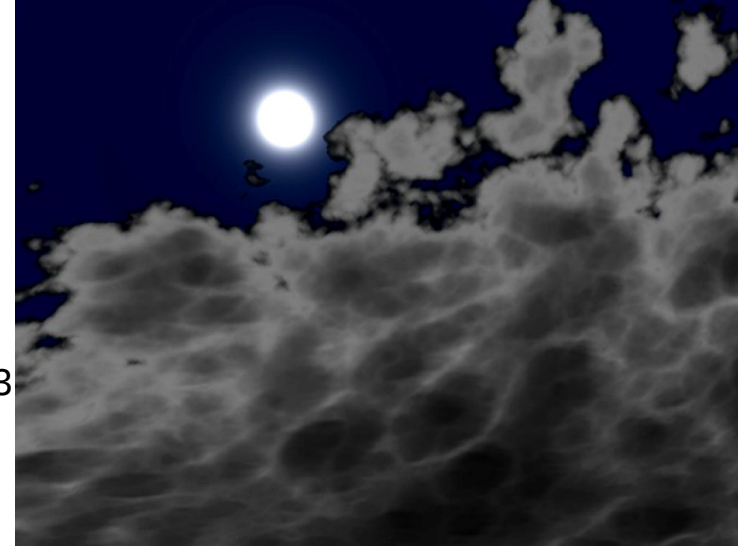
- Height field: advected textures
  - 16km-wide landscape
  - 2km-height, 500m-thick cloud layer
- Lighting & rendering on GPU
  - Efficient shaders (GLSL)
  - Clouds, ground, shadows: textures
  - Radiosity: 16x16 textures
- 20 – 40 FPS on current hardware

# Results

# Bottom view: contributions



Orders 1 & 2



Orders >3



Ground Illumination



Sky illumination

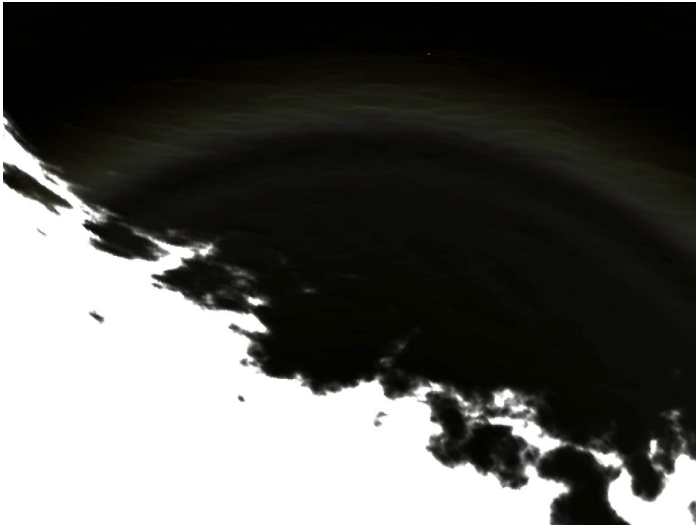


# Bottom view: summing it all

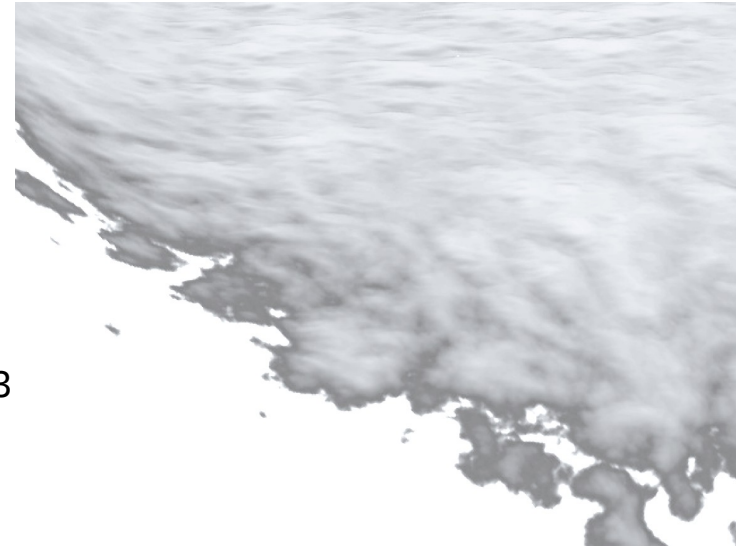


20 – 40 fps

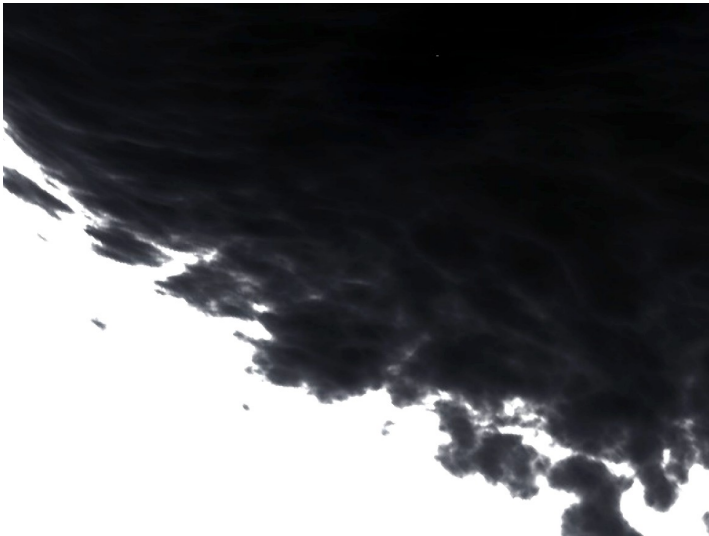
# Top view: contributions



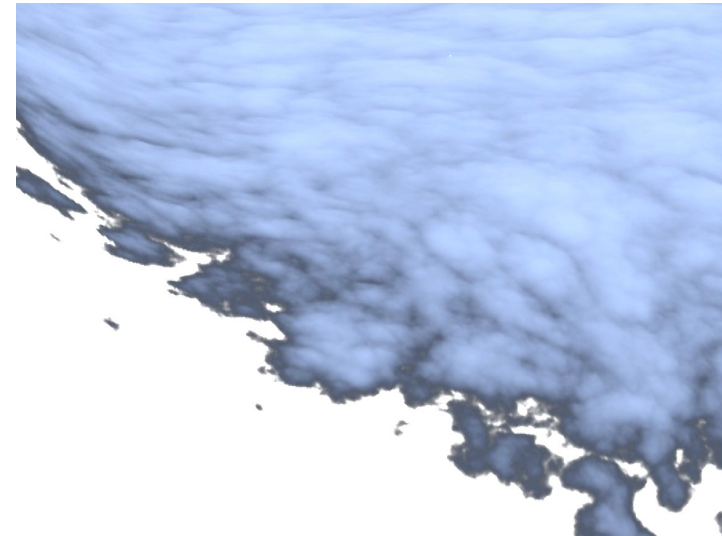
Orders 1 & 2



Orders >3

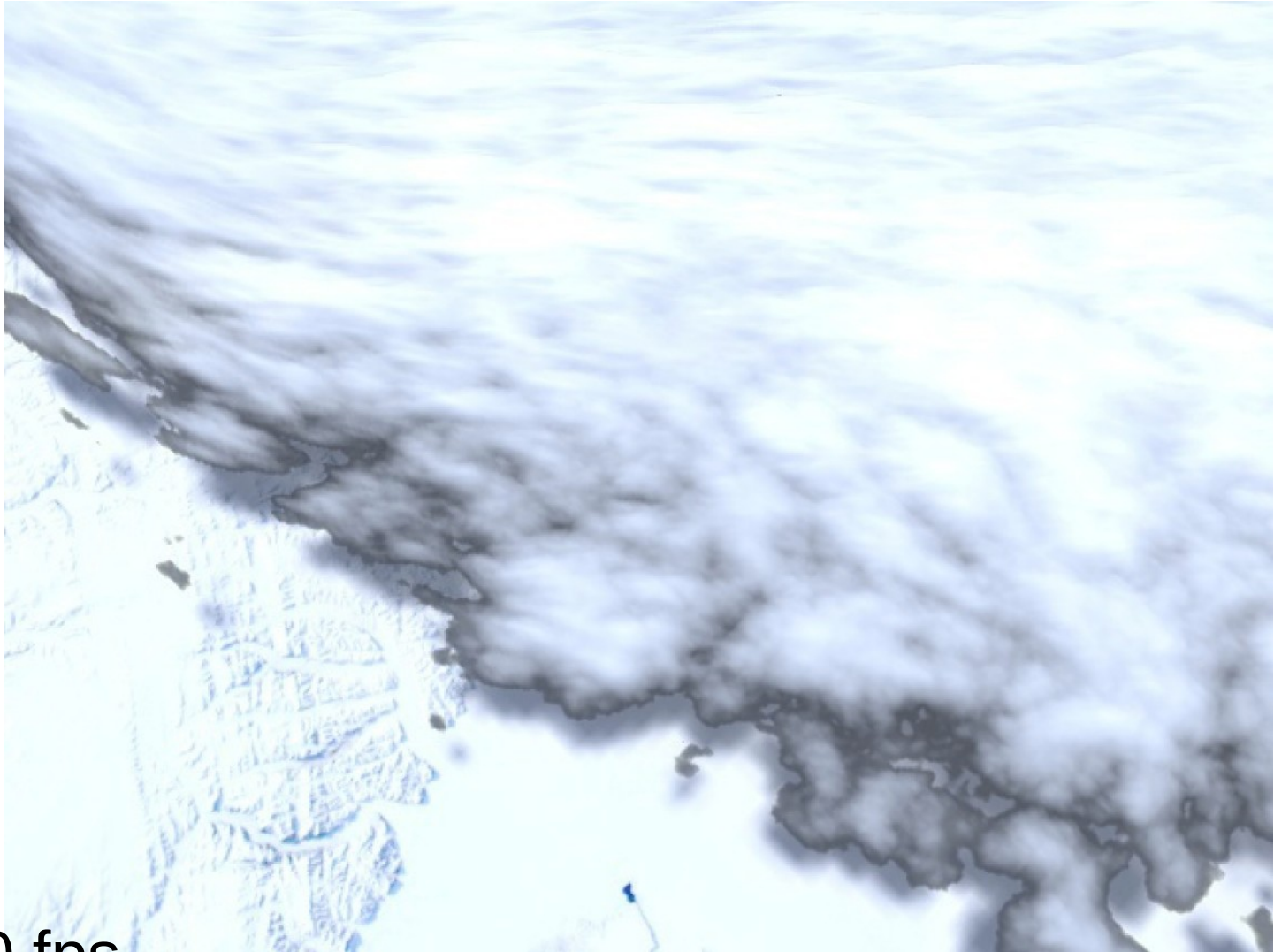


Ground Illumination



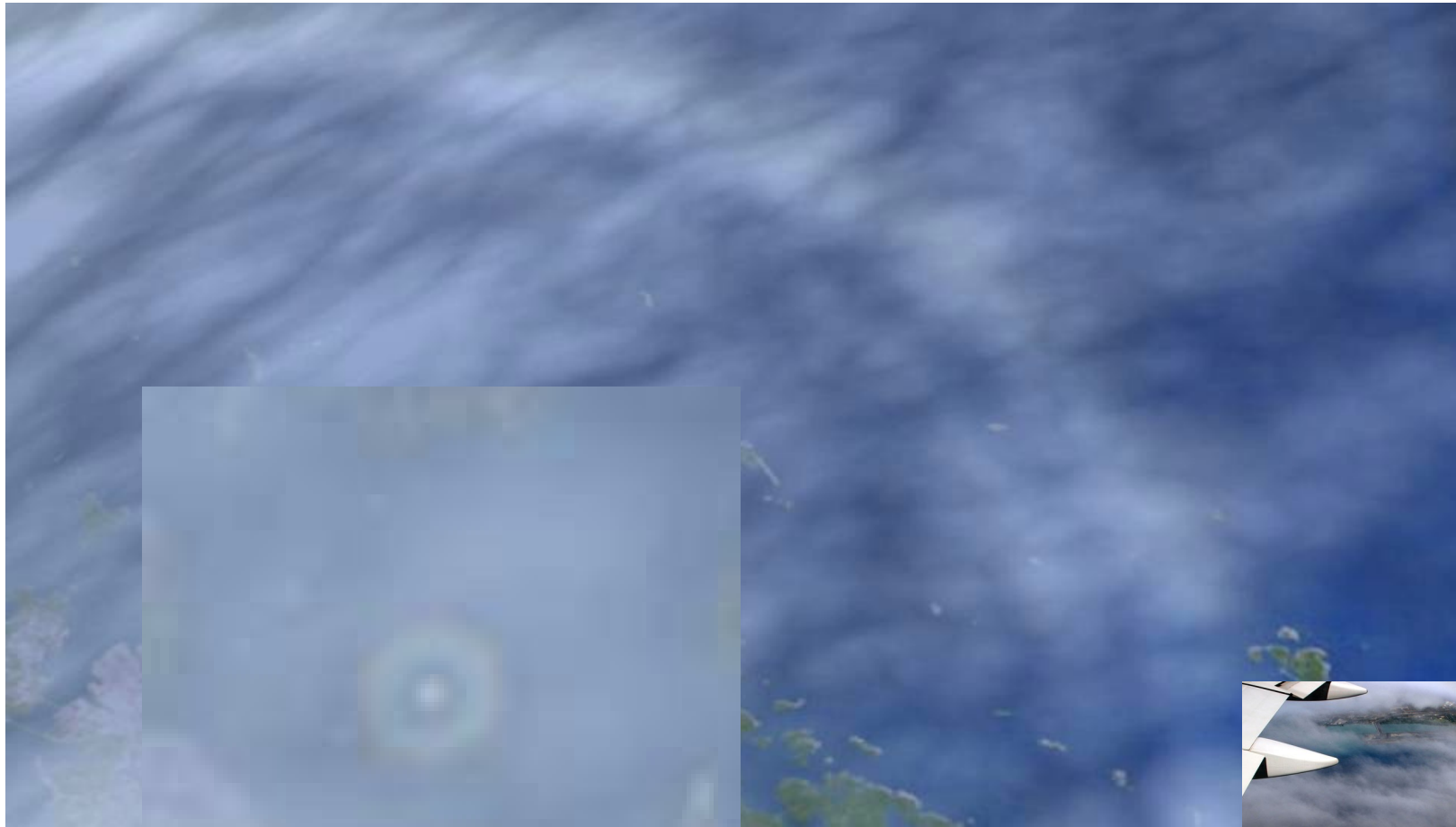
Sky illumination

# Top view: summing it all



20 – 40 fps

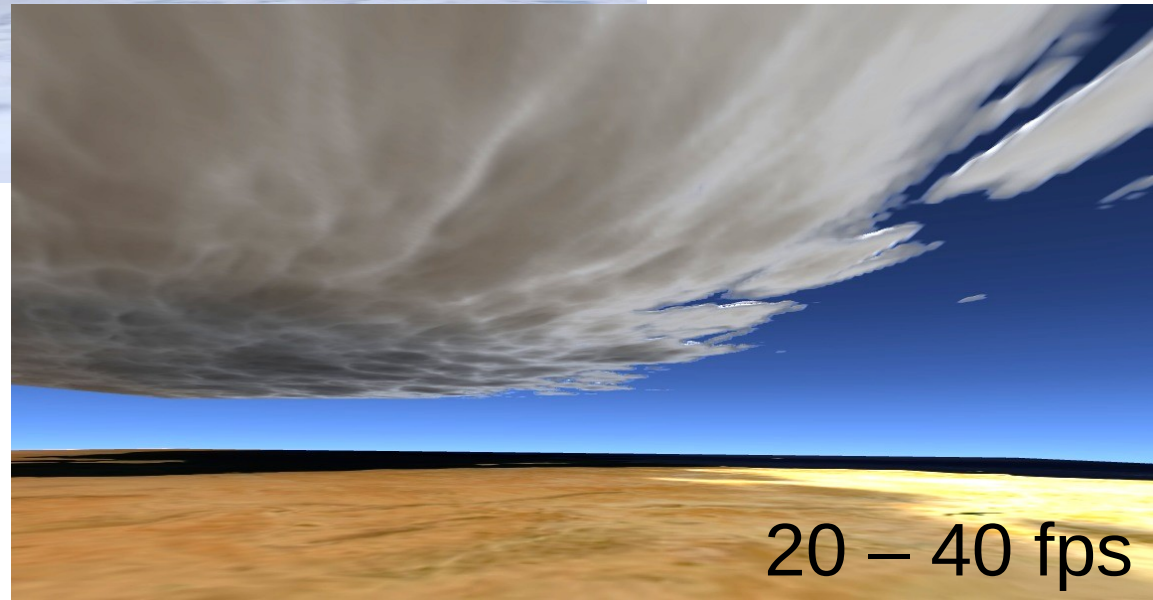
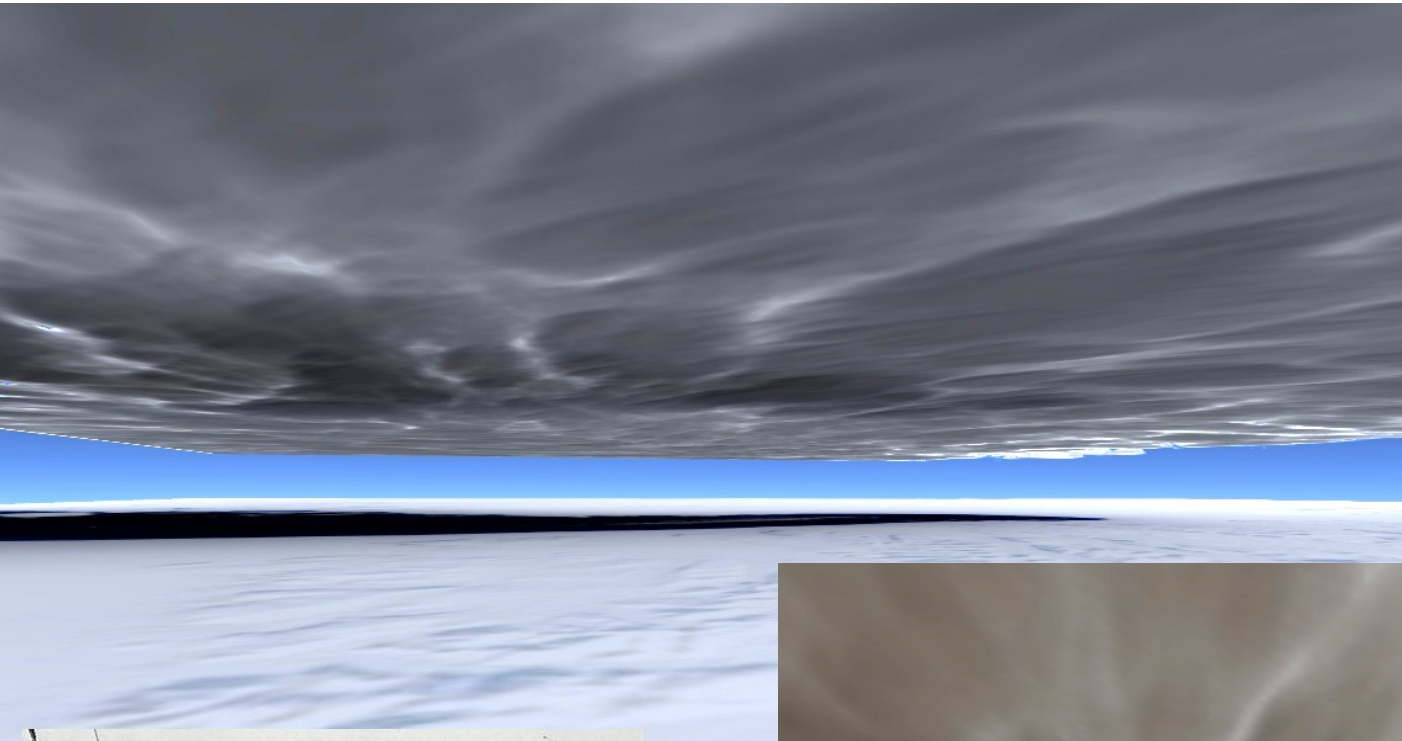
# Features: glory and fogbow



20 – 40 fps

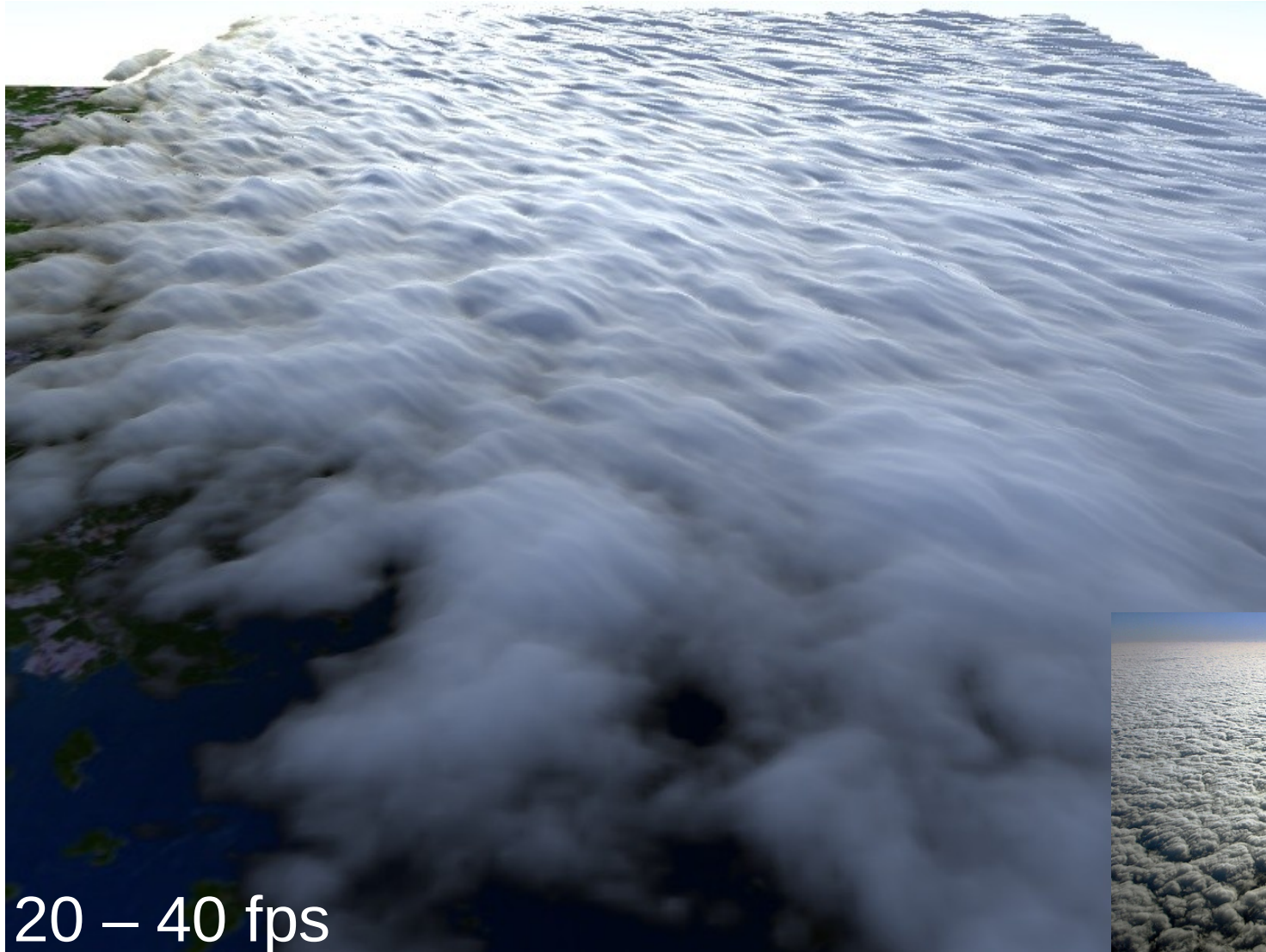


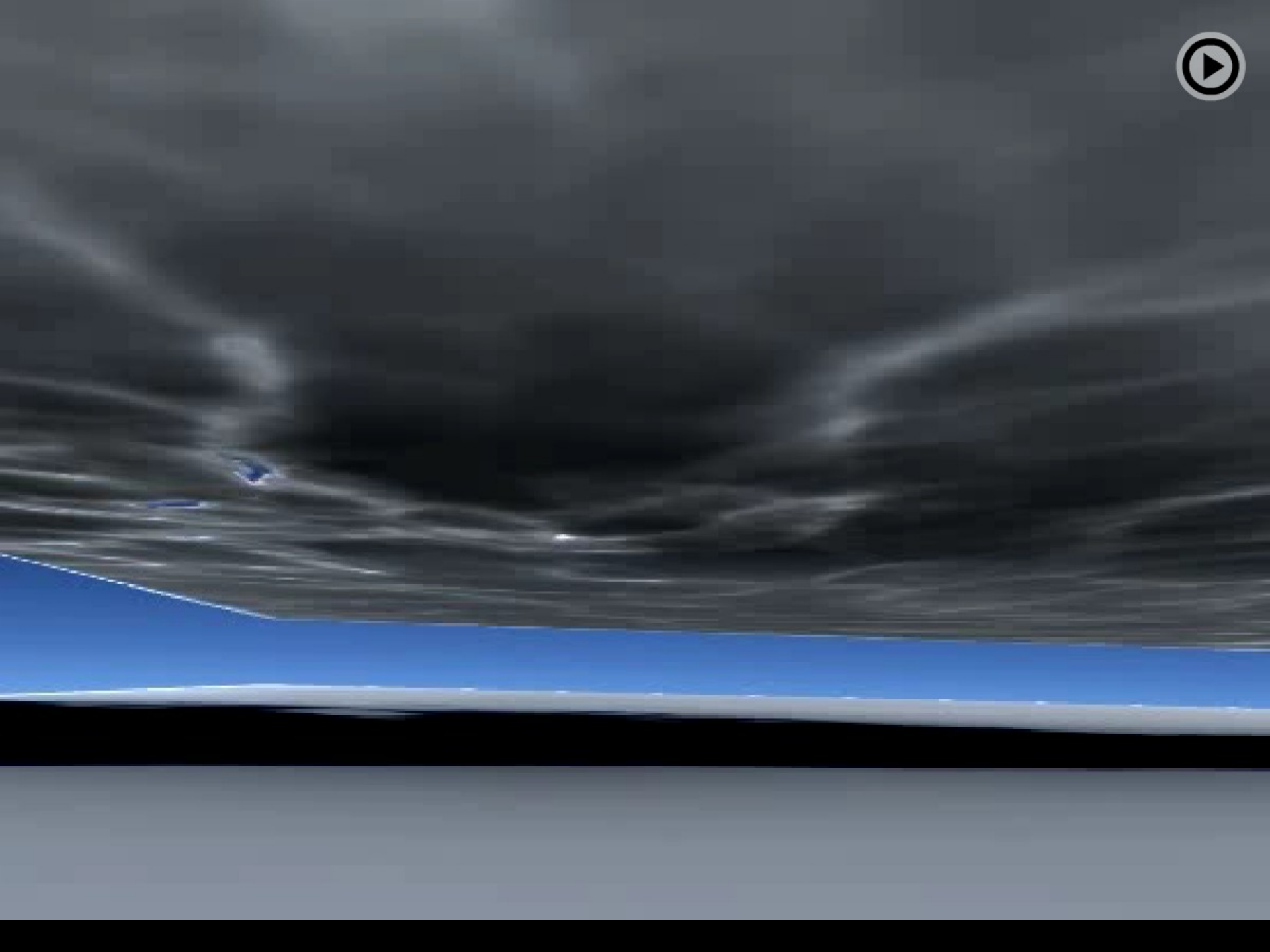
# Features: water sky



20 – 40 fps

# Features: pseudo-specular effect





# Discussion

- Advantages

- Realistic
- Real-time
- Animation-friendly

- Disadvantages

- Accuracy can be improved
- Limited to stratiform clouds
- "Hard edges" aspect

- Published at EGWNP'06

Antoine Bouthors, Fabrice Neyret, and Sylvain Lefebvre. ***Real-time realistic illumination and shading of stratiform clouds***. In *Eurographics Workshop on Natural Phenomena*, sep 2006.

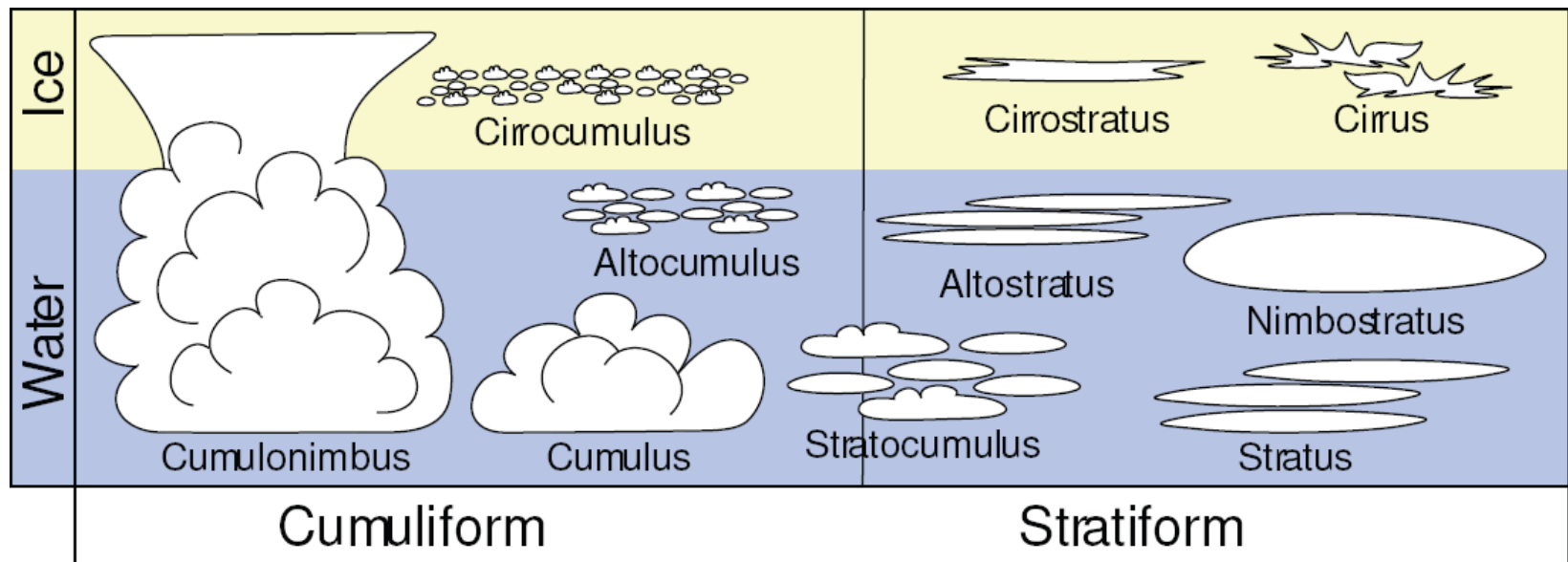
- Implemented in upcoming commercial game



# New clouds rendering models

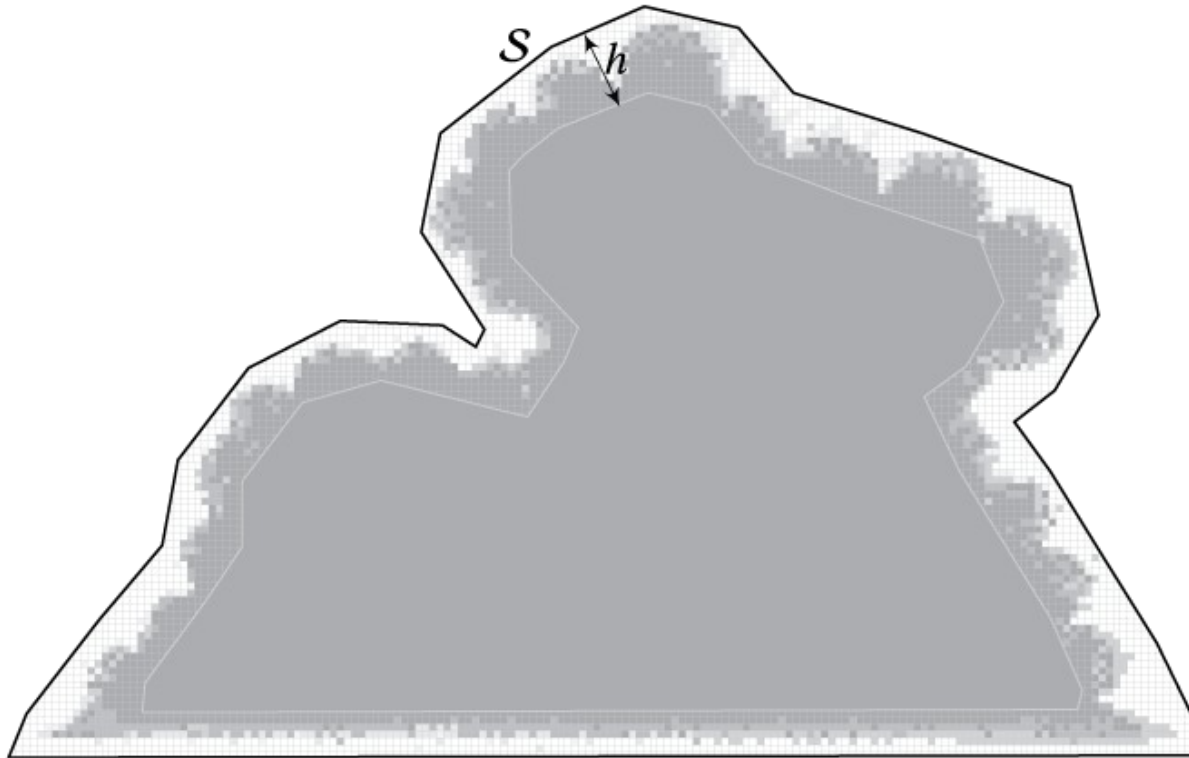
1. Model for stratiform clouds [BNL06]

1. General model [BBNM07]



# Shape model

- Inhomogeneous only on edges (optional)
- Homogeneous core

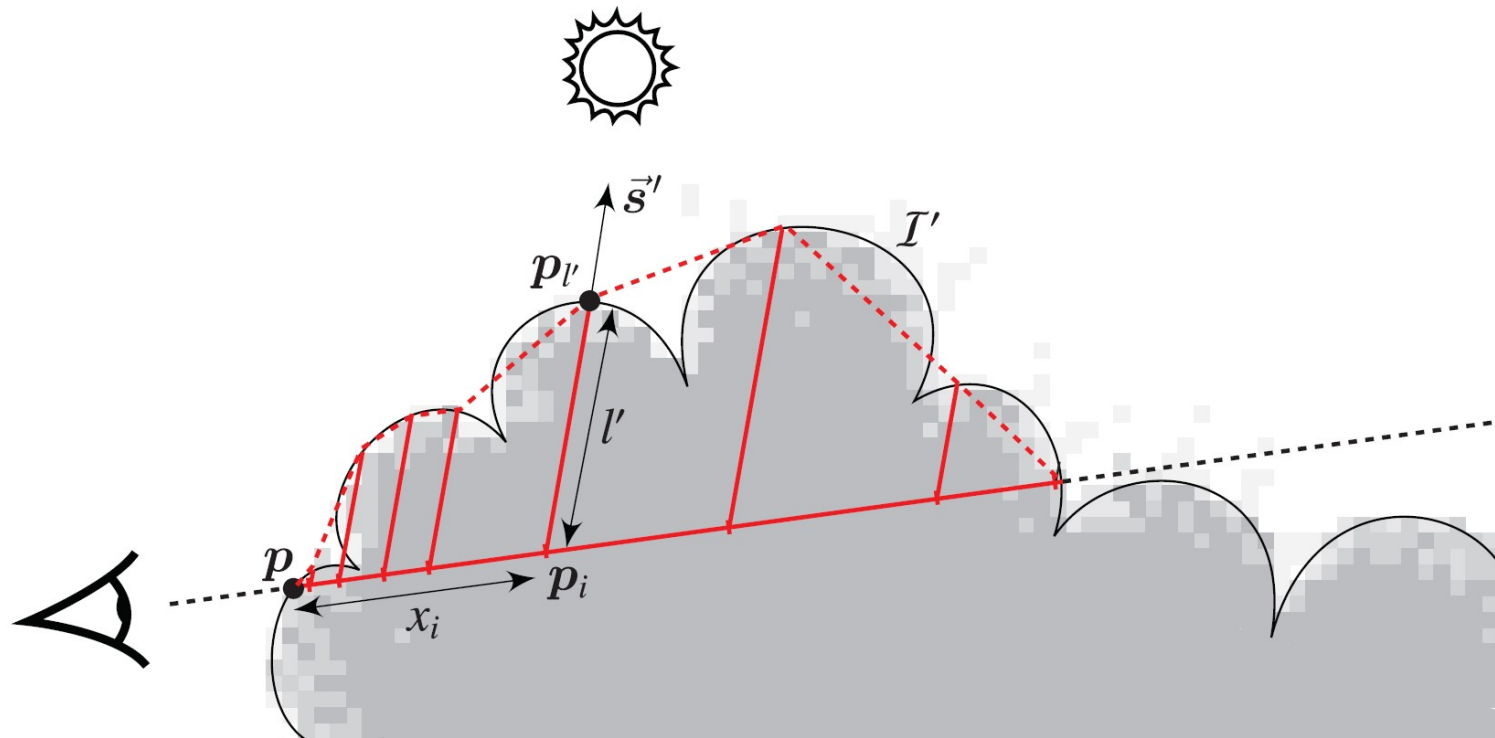


# Light transport model: overview

- Assumptions:
  - Homogenous core
- Our approach:
  - Separate contributions of different orders
  - Use our study

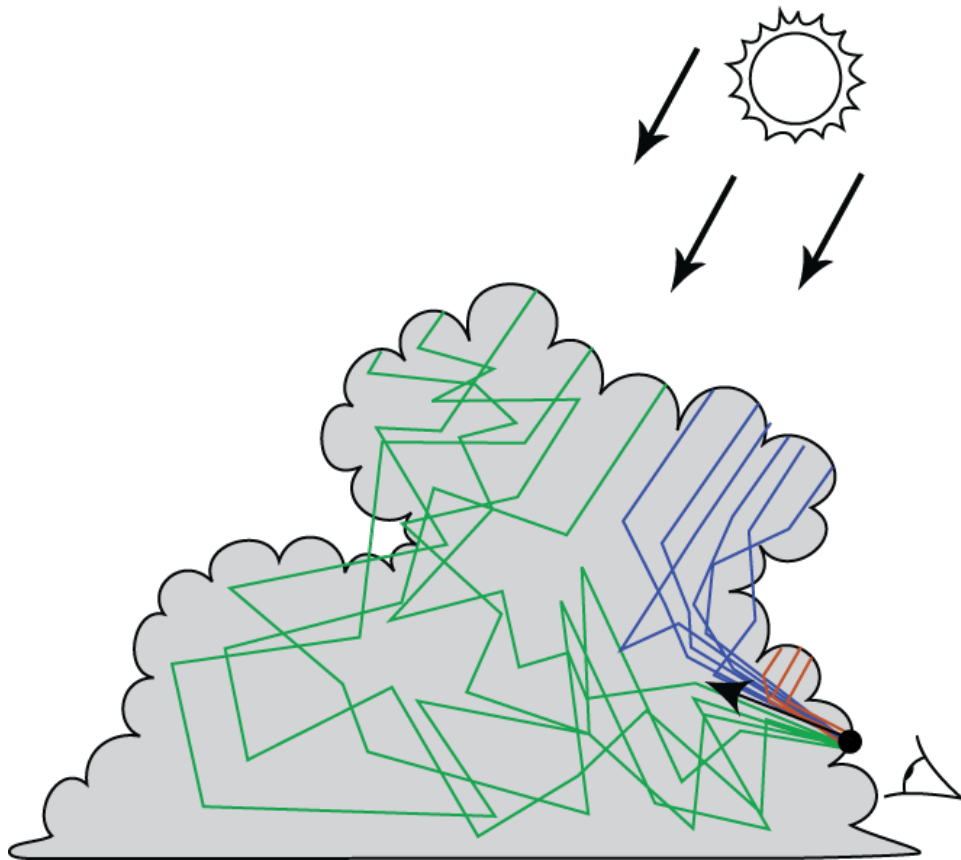
# Single scattering

- GPU ray marching



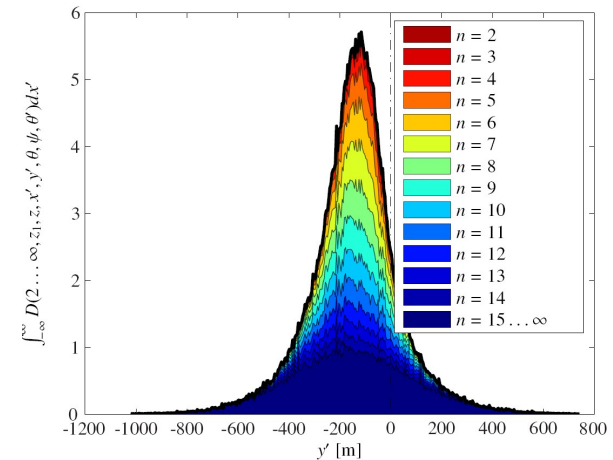
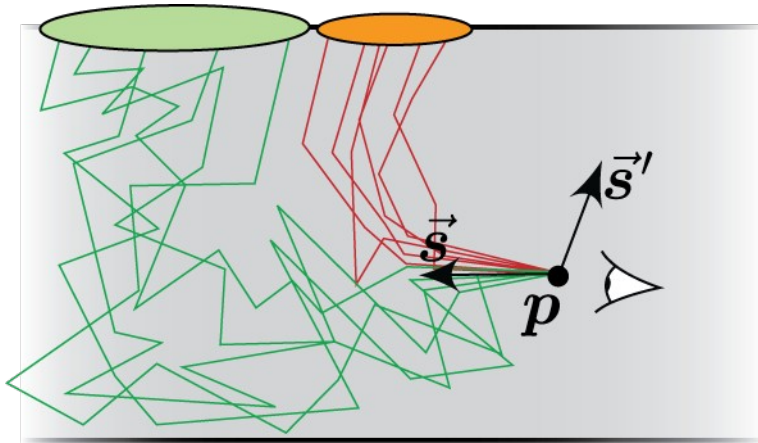
# Multiple scattering

- Split by groups of orders of scattering
- Compute luminance for each group



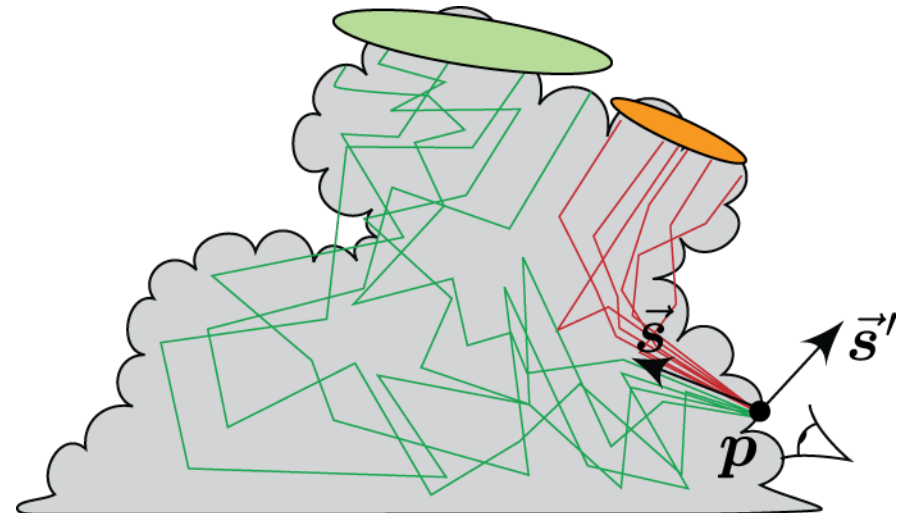
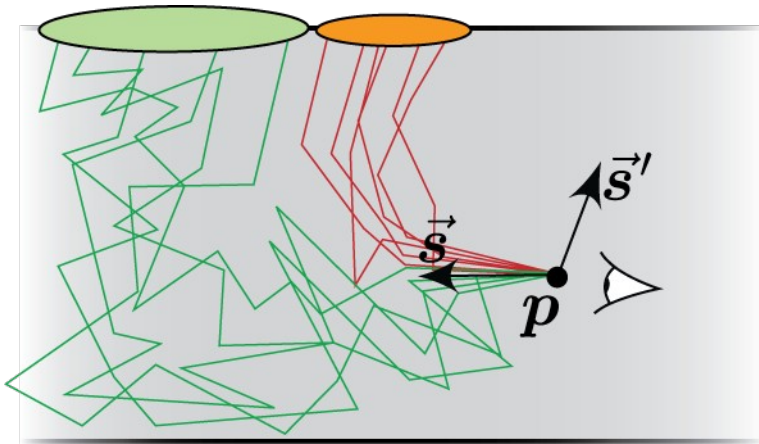
# Multiple scattering

- Defining the **collector area**
  - Area through which 95% of the light enters
  - In slabs, collector = disk
  - $\Psi$  gives it for a slab



# Multiple scattering

- For an arbitrary cloud
  - Assume collector = disk



# Multiple scattering

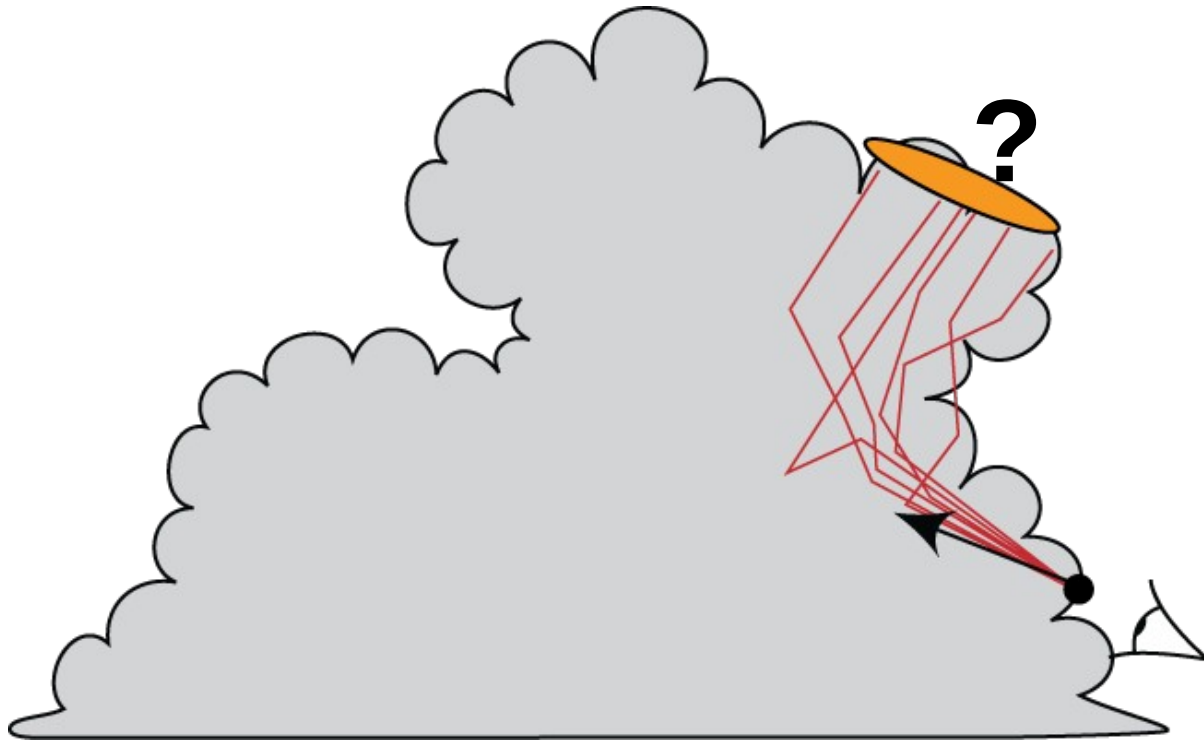
- Surface outside the collector is negligible
  - Fit a slab using the collector
  - Use  $\Psi$  to compute luminance





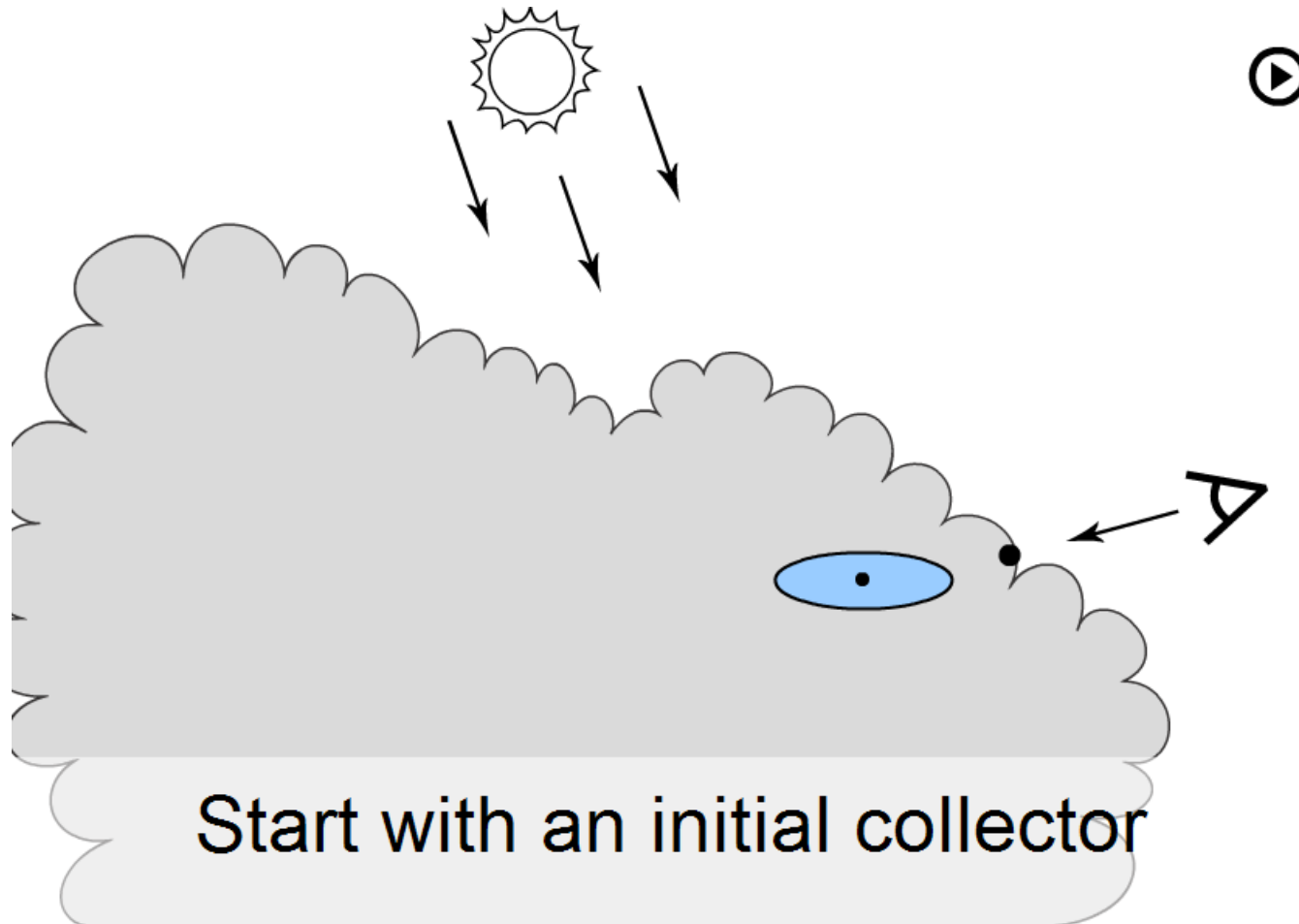
# Multiple scattering

- One issue left: where is the collector ?



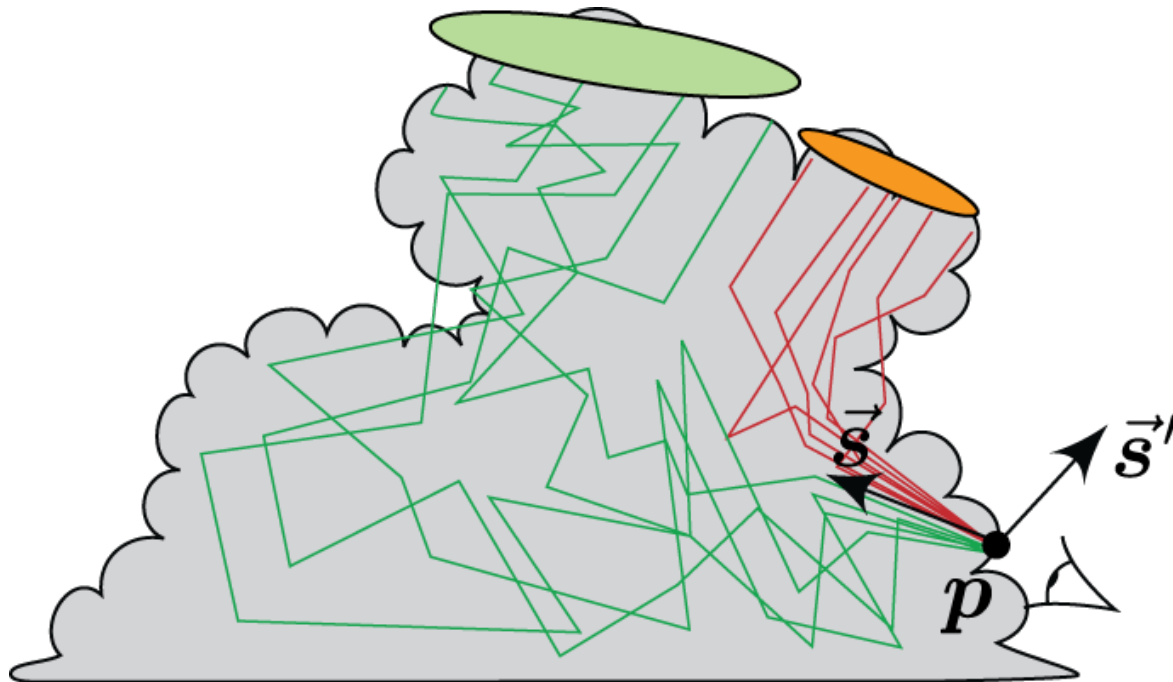
# Finding the collector

- Our solution: iterative algorithm



# Light transport model: summary

- For each order of scattering
  - Find the collector
  - Compute luminance using  $\Psi$



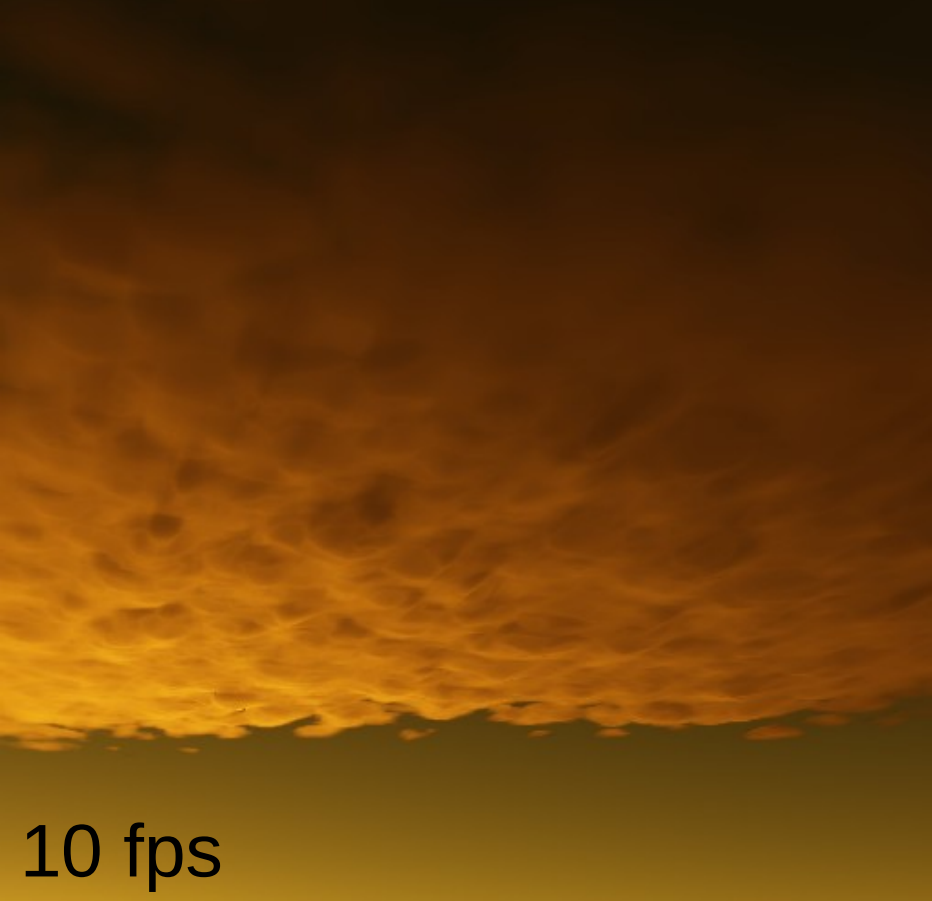
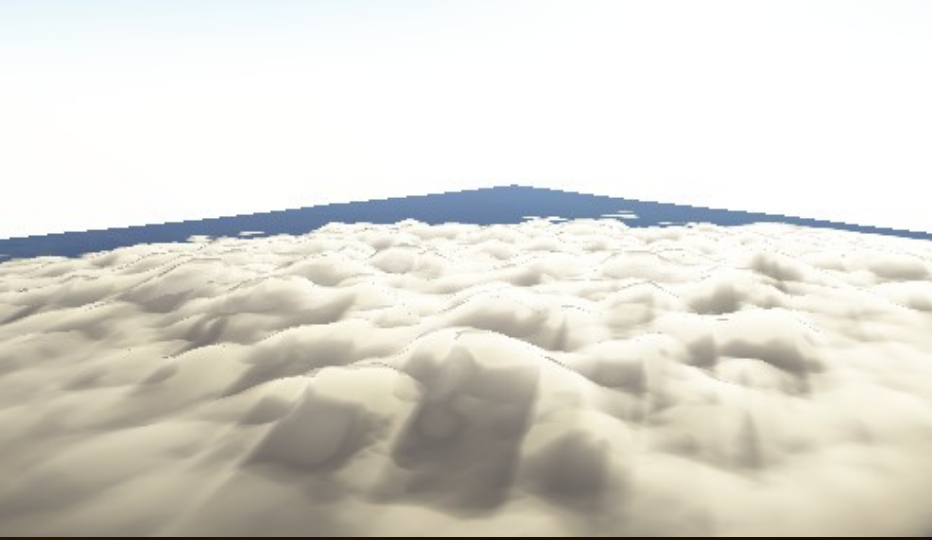
# Sky and ground

- Additional sources for sky and ground

# Implementation

- On GPU
  - Efficient shaders (GLSL)
  - Render-to-texture, deferred shading, depth maps
  - Mesoscopic model = functions + textures

# Results



10 fps

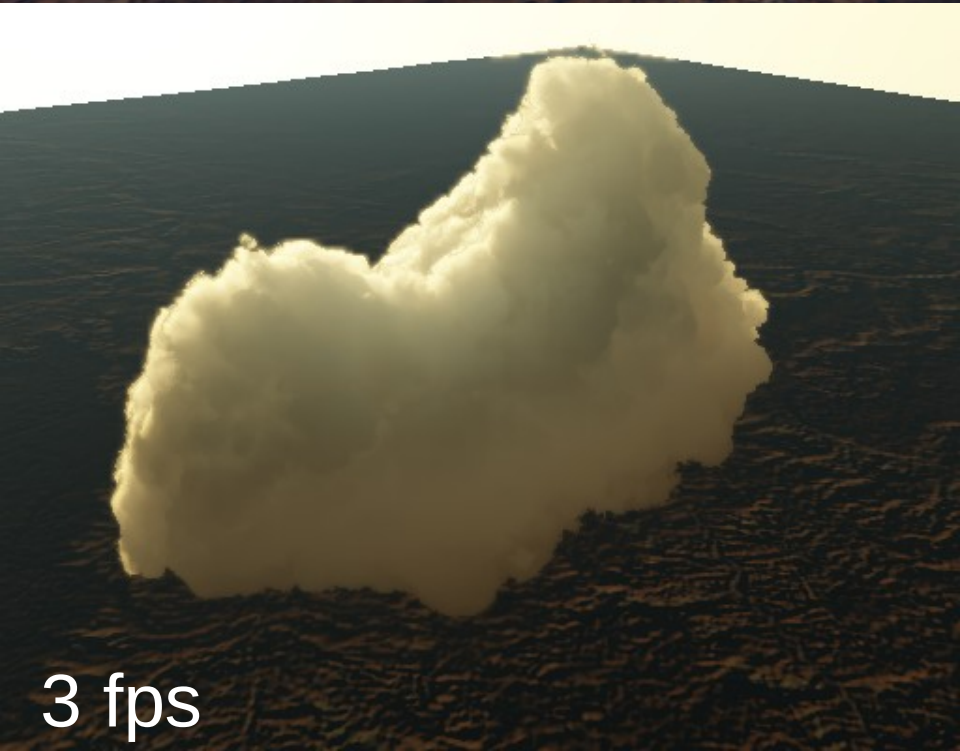


10 fps



10 fps



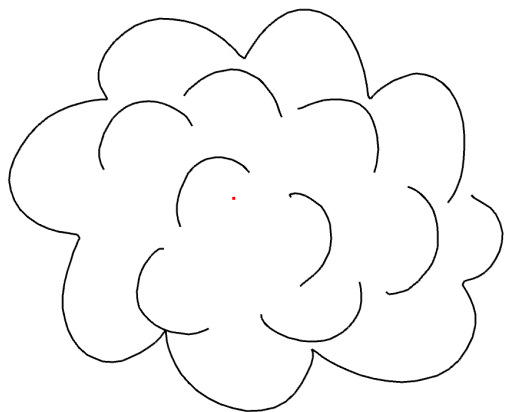


3 fps

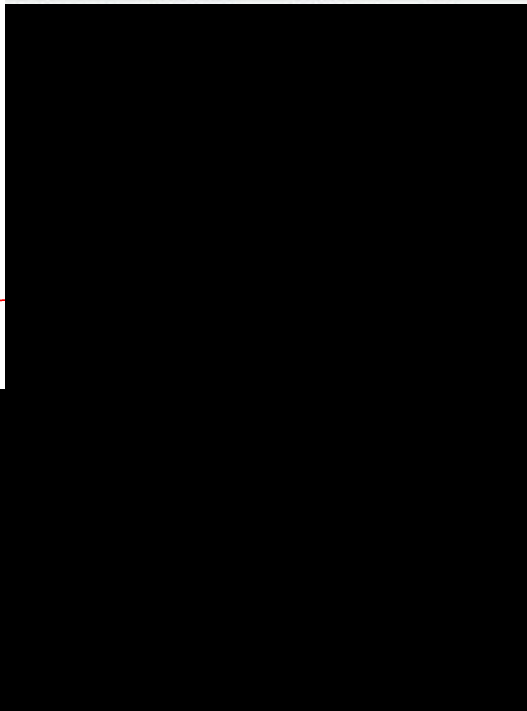
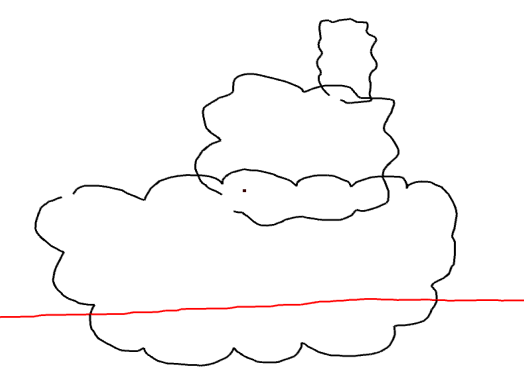
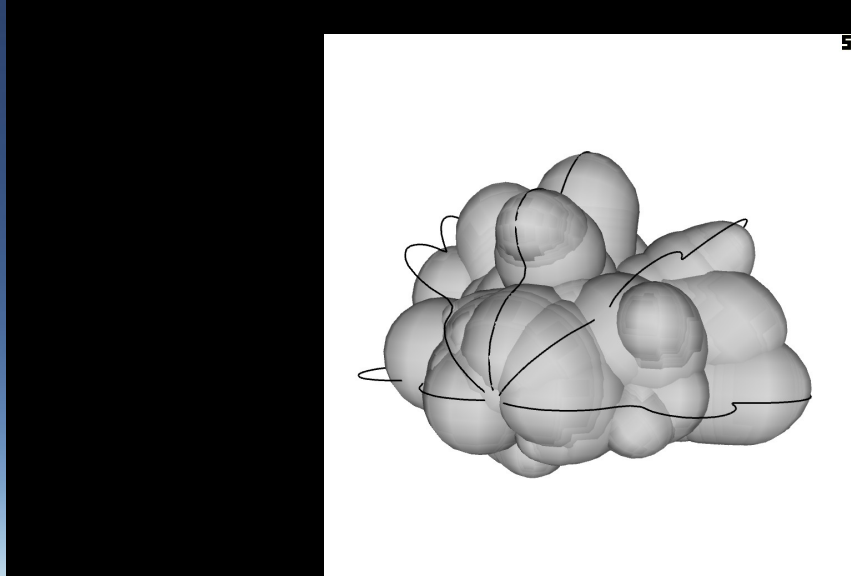




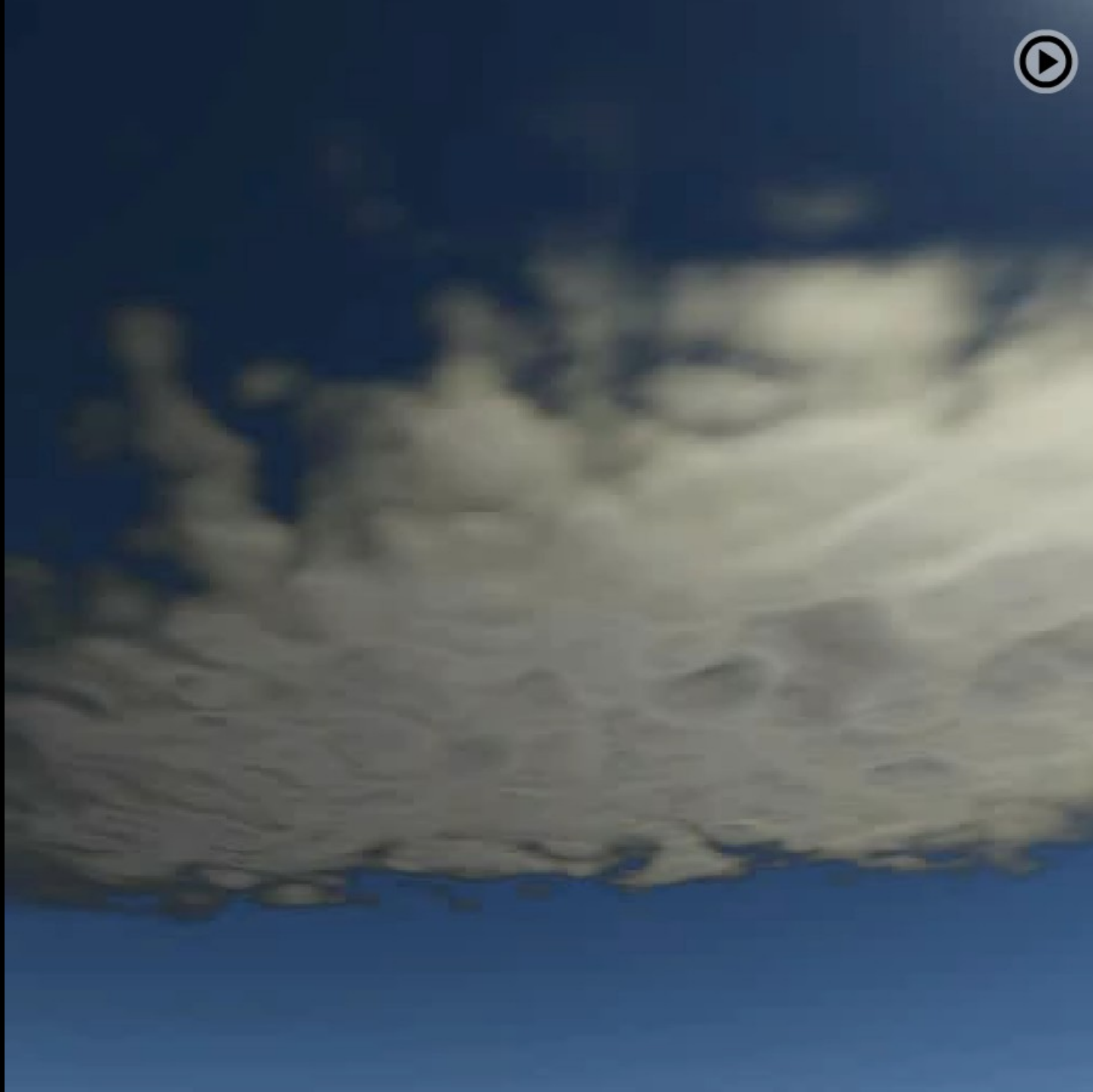
3 fps



3 fps



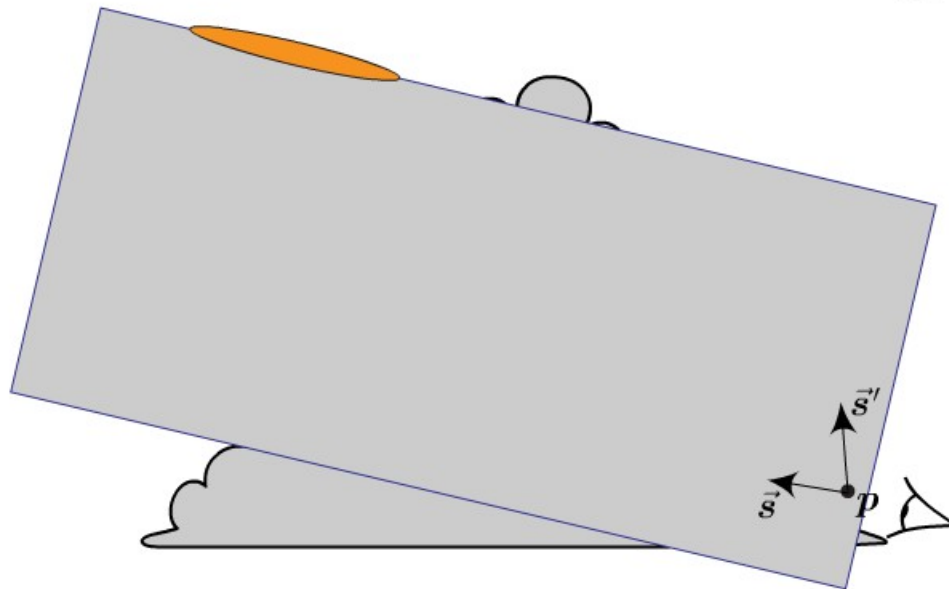
3 fps





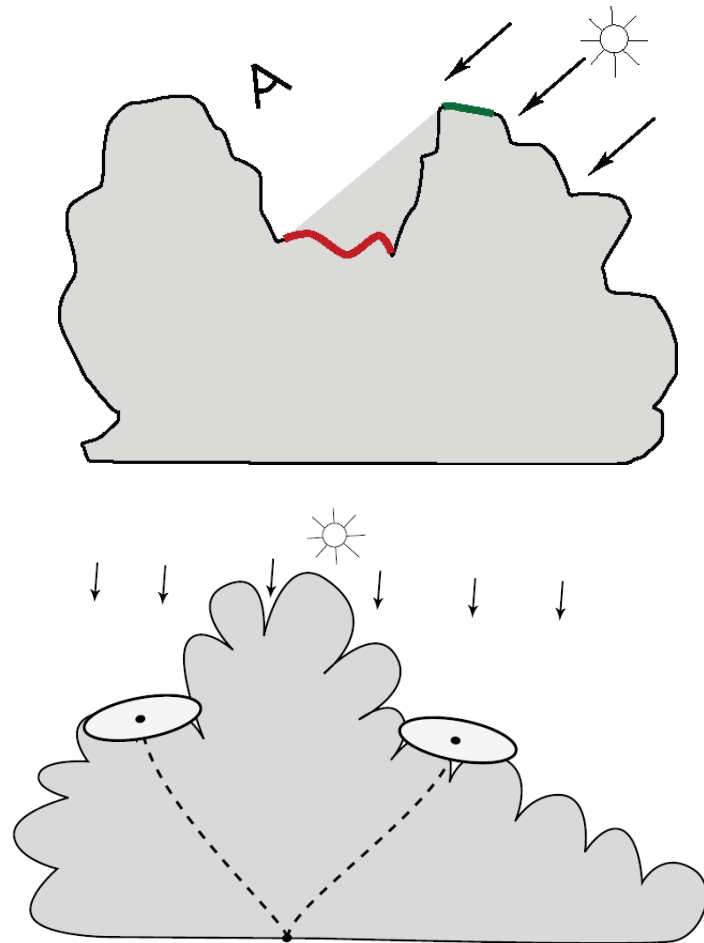
# Discussion

- Iterative algorithm
  - May have periodic points
  - Use relaxation
  - (solver issue)



# Discussion

- Issues with:
  - Non-convex shapes  
(implementation issue)
  - More than one collector  
(solver issue)  
→ multiple root finding





# Discussion

- Advantages

- Realistic
- Detailed
- Interactive (tunable cost)
- Animation-friendly
- Works on any shape

- Disadvantages

- Complex to implement

- Published at I3D'08

Antoine Bouthors, Fabrice Neyret, Nelson Max, Eric Bruneton, and Cyril Crassin.  
***Interactive multiple anisotropic scattering in clouds.*** In *ACM SIGGRAPH Symposium on Interactive 3D graphics and games (I3D)*, 2008.

# Conclusions

- New way to compute radiative transfer
  - High quality
  - High speed
  - High detail
  - Animatable
  - Compatible with GPU

# Conclusions

- Future research
  - More than one collectors
  - Non-convex shapes, full sceneries
  - Inhomogeneous clouds
- Other applications
  - Subsurface scattering





Thank you !  
Questions ?



# What about convergence ?

- Bounded search space  
→ No divergence
- Periodic points  
→ Use relaxation
- Start with large collector
  - "Refine" cloud surface

