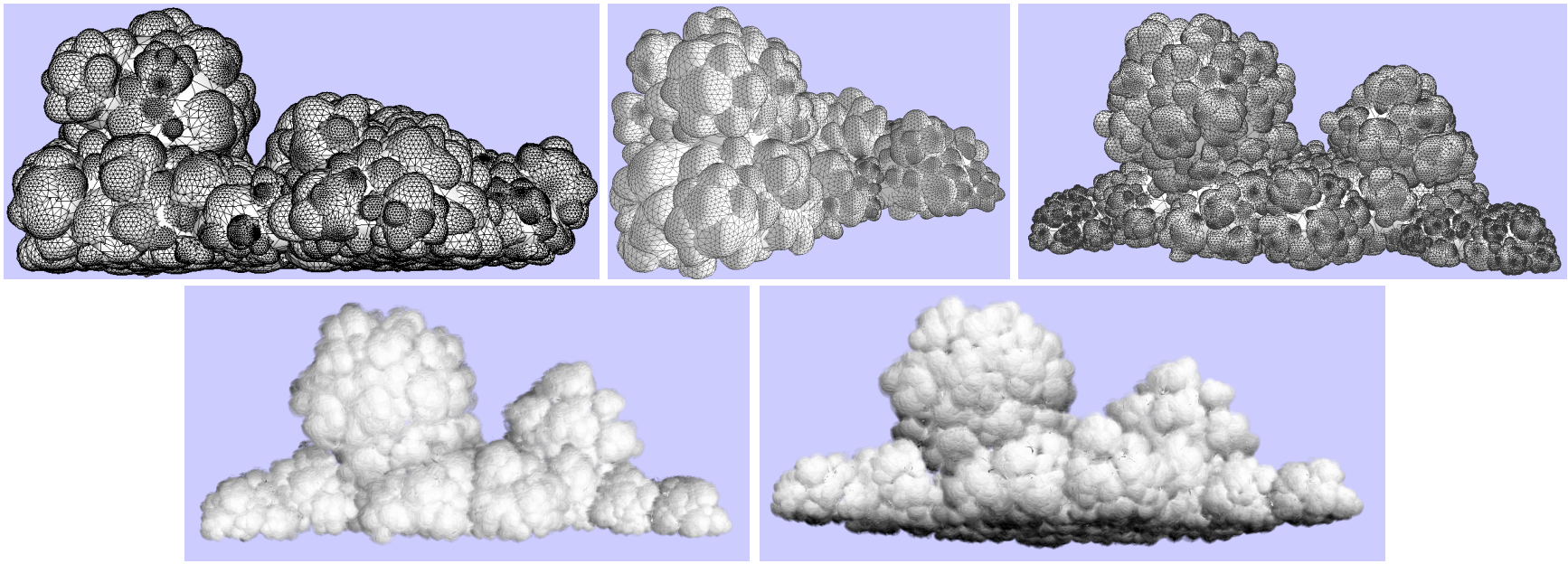
# Modeling Clouds Shape

*Antoine Bouthors* *Fabrice Neyret*

Laboratoire GRAVIR

Grenoble, France



http://www-imagis.imag.fr/Publications/2004/BN04/

# 1 Introduction

- Target: well contrasted clouds (*i.e.,* cumulus clouds)
- Existing CG models:
  - Adapted for other types of clouds
  - Not yet realistic enough in real-time for cumulus
- Long term goal : realistic, animated, real-time
- For now: cloud shape

# 1.1 Case study: Cumulus shape caracteristics

- Multiscale set of stacked bubbles
- Very dense core
- Low density only in a thin cloud/air interface
- Flat bottom

# 1.1 Case study: Cumulus visual caracteristics

- Core highly reflective
- More scattering than reflection in the corolla

# 1.1 Case study: our hypothesis

- The surface plays the main role in lighting
- Clear silhouette

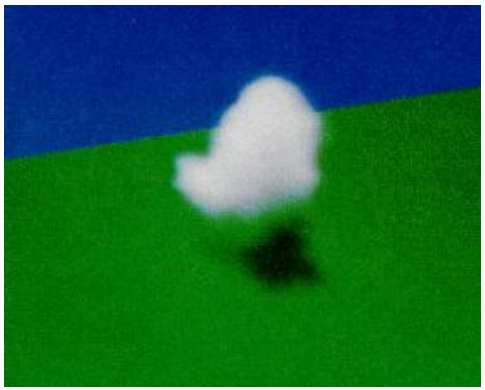
$\Rightarrow$ Well identified quasi-surface, which is:

- Multiscale
- Having much geometric details

# 1.2 Previous work

| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.2 Previous work



[Kajiya *et al.* , 1984]

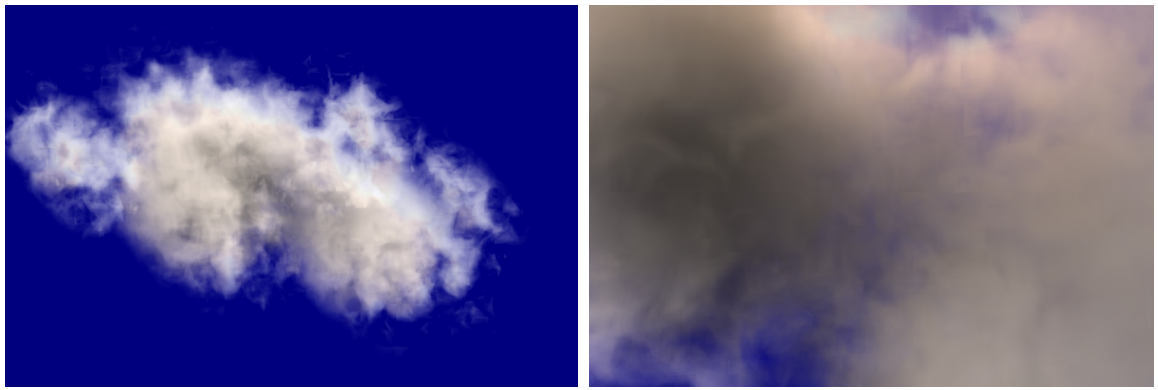
| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.2 Previous work

[Harris *et al.* , 2003]

| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.2 Previous work



[Ebert *et al.* , 1997]

| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.2 Previous work

[Nishita *et al.* , 1996]

| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.2 Previous work

[Gardner, 1985]

| **Shape representation** | **Shape generation** | **Rendering method** | **Rendering speed** |
|---|---|---|---|
| Volume | Simulation | Volume rendering | Real-time |
| Surface | Procedural | Slicing | Fast |
| | Implicit | Impostors | Slow |
| | From real data | Mesh ray-tracing | |

# 1.3 Our approach

- Surface like in [Gardner, 1985]
- Implicit like in [Nishita, 1996]
- Hierarchical
- High level of detail

# Plan

# 2 Our method

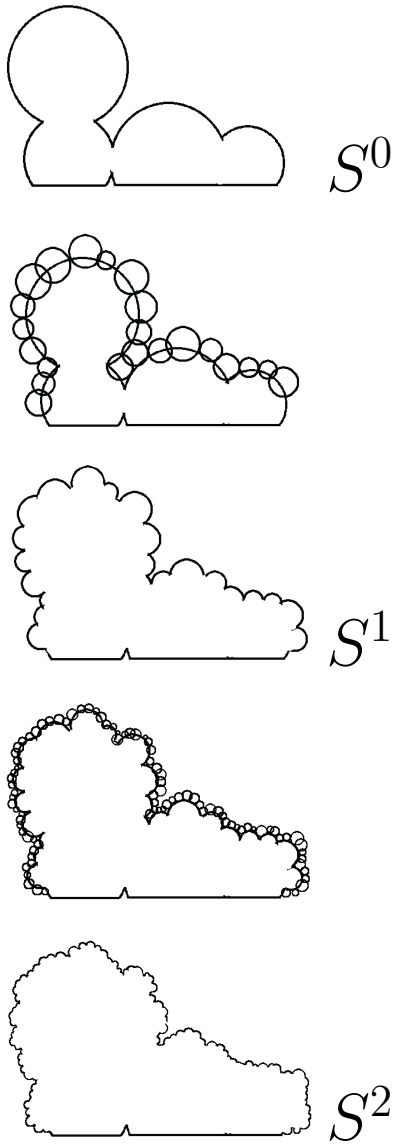
The user defines a root level $S^0$

Blobs $\mathbf{P}_i^1$ are created on $S^0$

Surfaces $S_i^1$ of these blobs define $S^1$

Blobs $\mathbf{P}_i^2$ are created on $S^1$

And so on...



## 2.1 Our representation

- Level $l$: set of blobs (position $\mathbf{P}_i^l$, radius $r_i^l$)
- Surface $S_i^l$ of blob $\mathbf{P}_i^l$: implicit function $f_i^l(\mathbf{P})$
- Level surface $S^l = \bigcup_i S_i^l$ of this level

# 2.1 Our representation

- Level $l$: set of blobs (position $\mathbf{P}_i^l$, radius $r_i^l$)
- Surface $S_i^l$ of blob $\mathbf{P}_i^l$: implicit function $f_i^l(\mathbf{P})$ $\longrightarrow$ 2.2
- Level surface $S^l = \bigcup_i S_i^l$ of this level

# 2.1 Our representation

- Level $l$: set of blobs (position $\mathbf{P}_i^l$, radius $r_i^l$)
- Surface $S_i^l$ of blob $\mathbf{P}_i^l$: implicit function $f_i^l(\mathbf{P})$
- Level surface $S^l = \bigcup_i S_i^l$ of this level $\longrightarrow$ 2.3

# 2.1 Our representation

- Level $l$: set of blobs (position $\mathbf{P}_i^l$, radius $r_i^l$) $\longrightarrow$ 2.4
- Surface $S_i^l$ of blob $Po_i^l$: implicit function $f_i^l(\mathbf{P})$
- Level surface $S^l = \bigcup_i S_i^l$ of this level

## 2.2 Defining the blob surface (*i.e.*, $f_i^l(\mathbf{P})$)

Base: spherical shape

Ptotential $f_i^l(\mathbf{P})$ Altered to match our observations

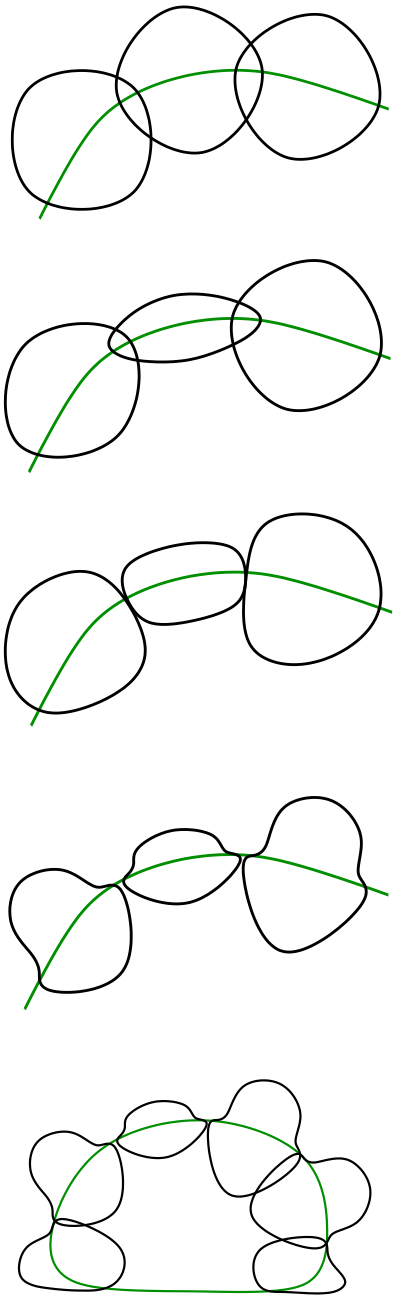
Base: spherical shape

Random flattening term

Mutual repulsion: ”contact surface” inspired from [Gascuel *et al.* 93]

The blob enlarges near its base (*i.e.*, near $S^{l-1}$)

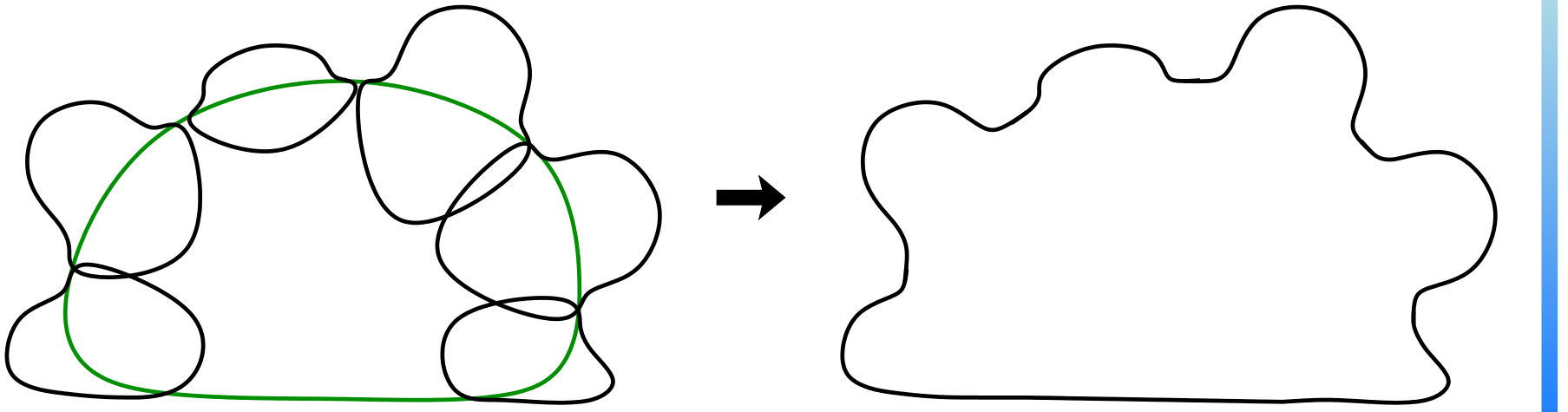
$S_i^l$ does not go below a given height



# 2.3 Defining the level surface

$$f^l(\mathbf{P}) = \max\left(f^{l-1}(\mathbf{P}), \max_i f_i^l(\mathbf{P})\right)$$

- $S^l$ is the union of the $S_i^l$'s and $S^{l-1}$
- The cloud surface is the surface of the last level



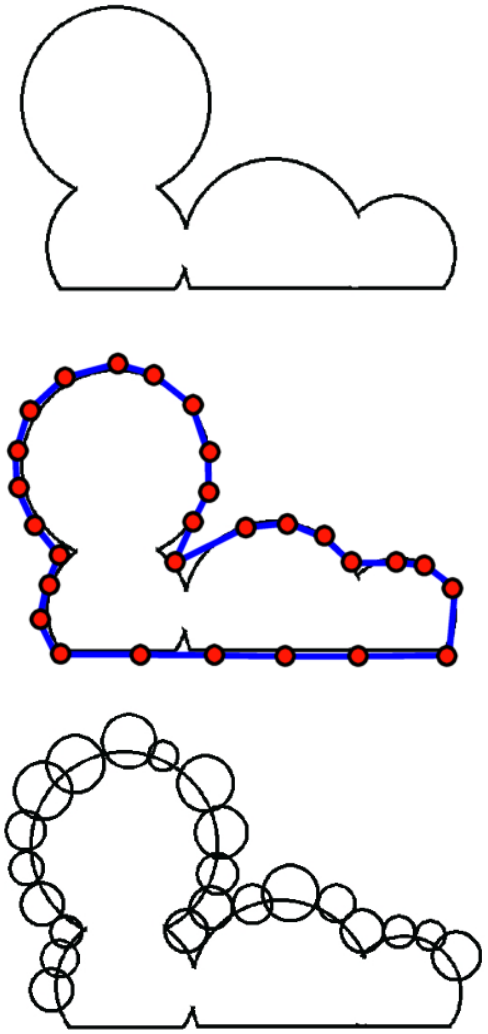
# 2.4 Setting the blobs

$S^l$ (implicit) is discretized using particles [Witkin *et al.* 94] [Crossono *et al.* 97]

Each particle has random variation of repulsion radius

Particles centers $\longrightarrow$ blobs centers

Particles repulsion radius $\longrightarrow$ blobs radius

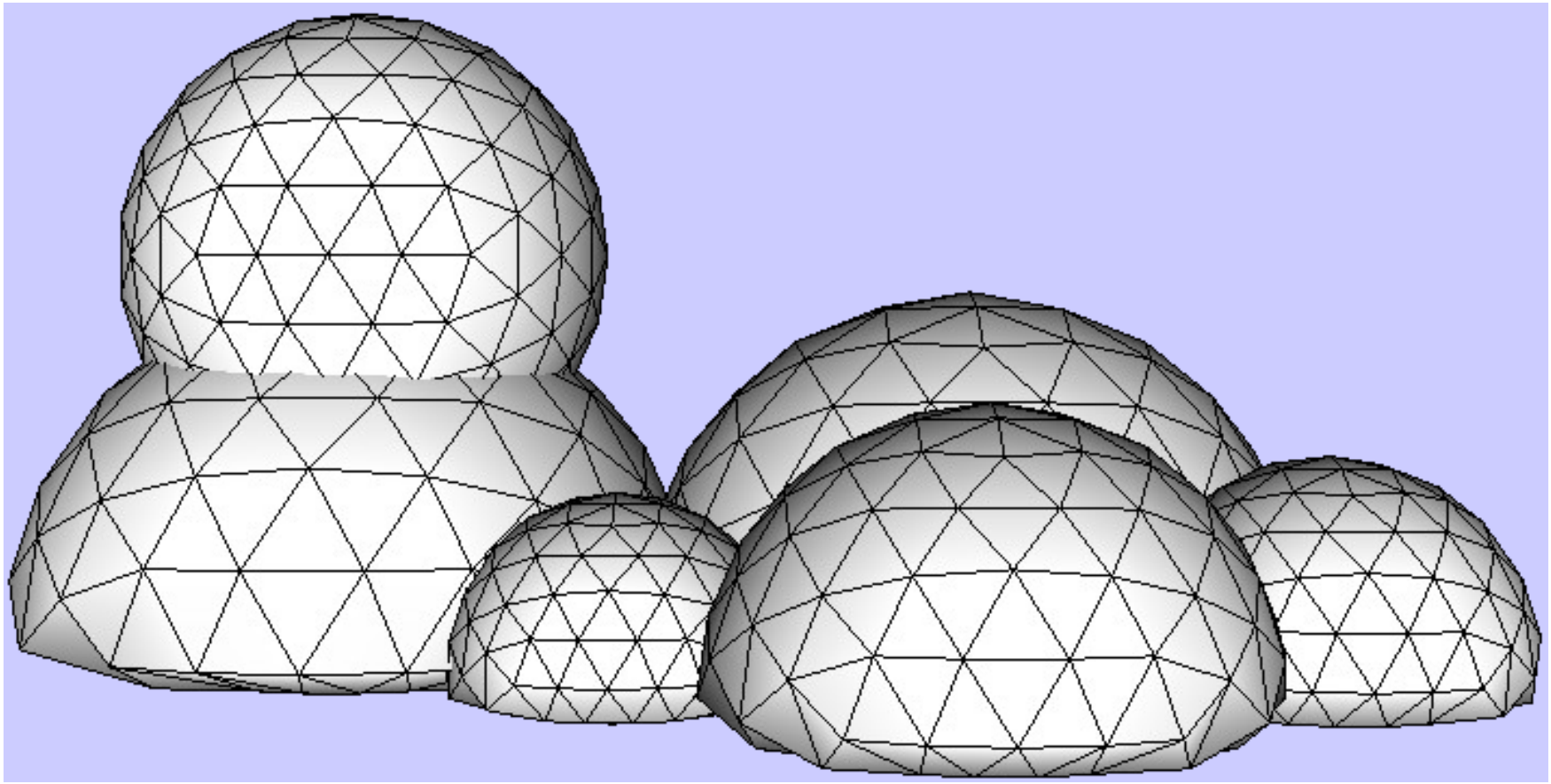


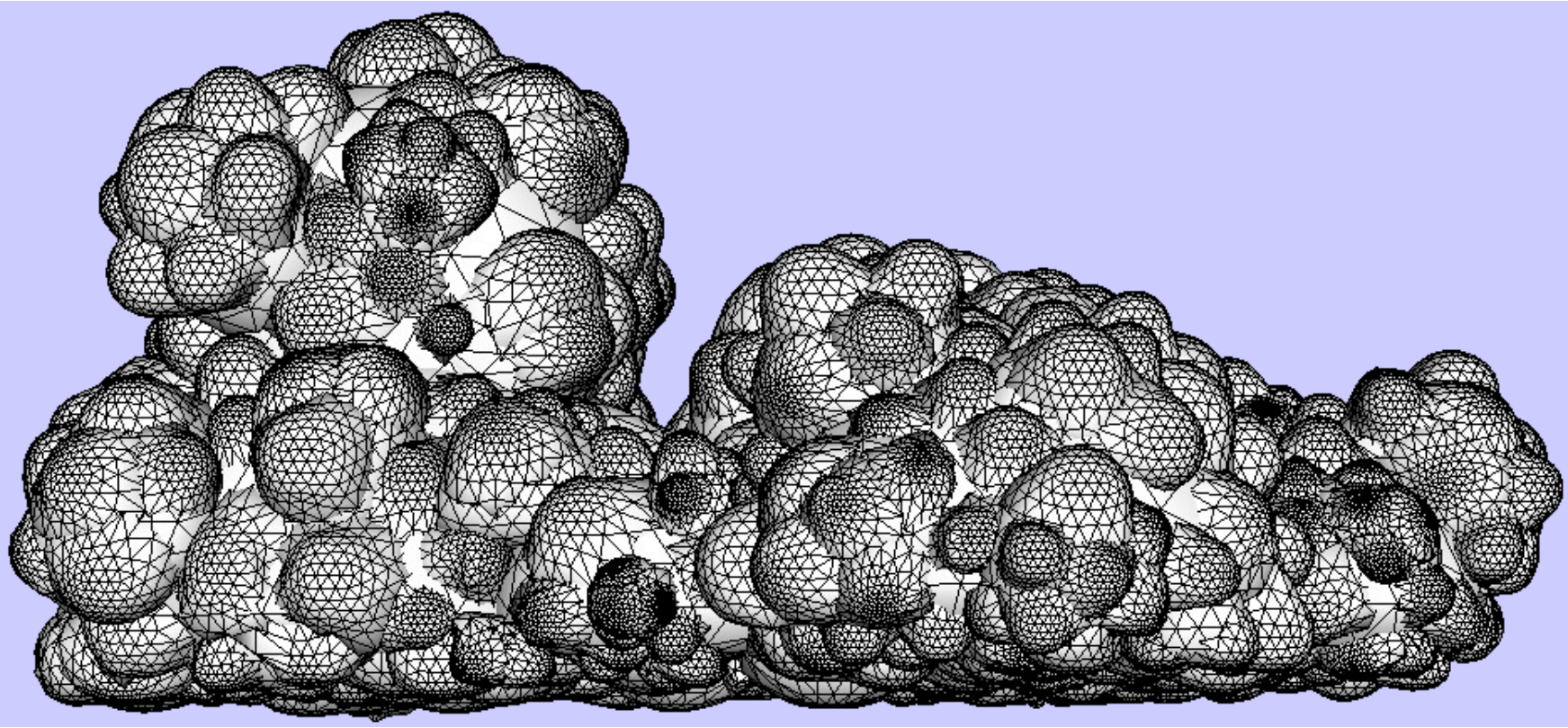
# 3 Rendering

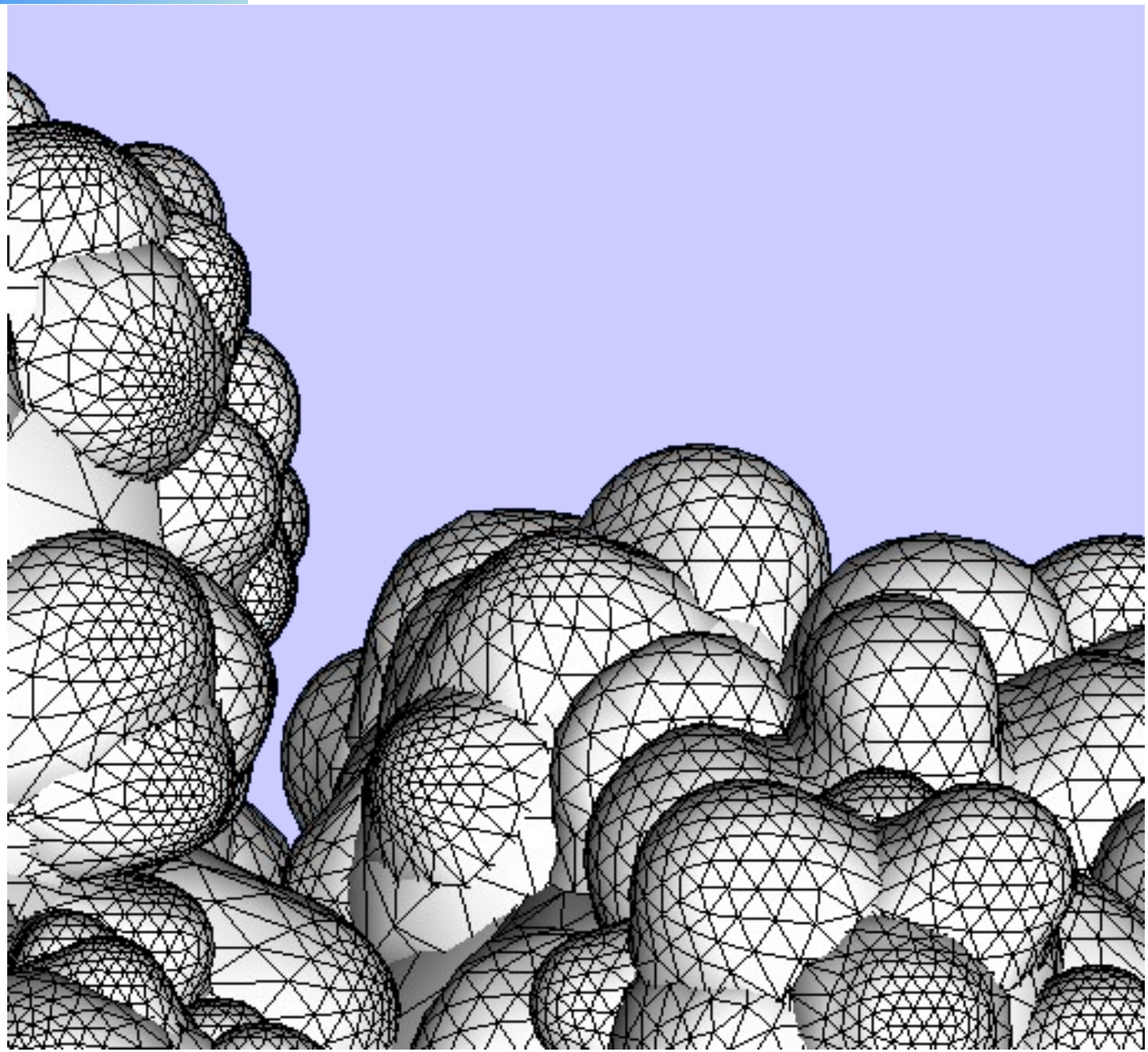
Not the purpose of this paper: minimal

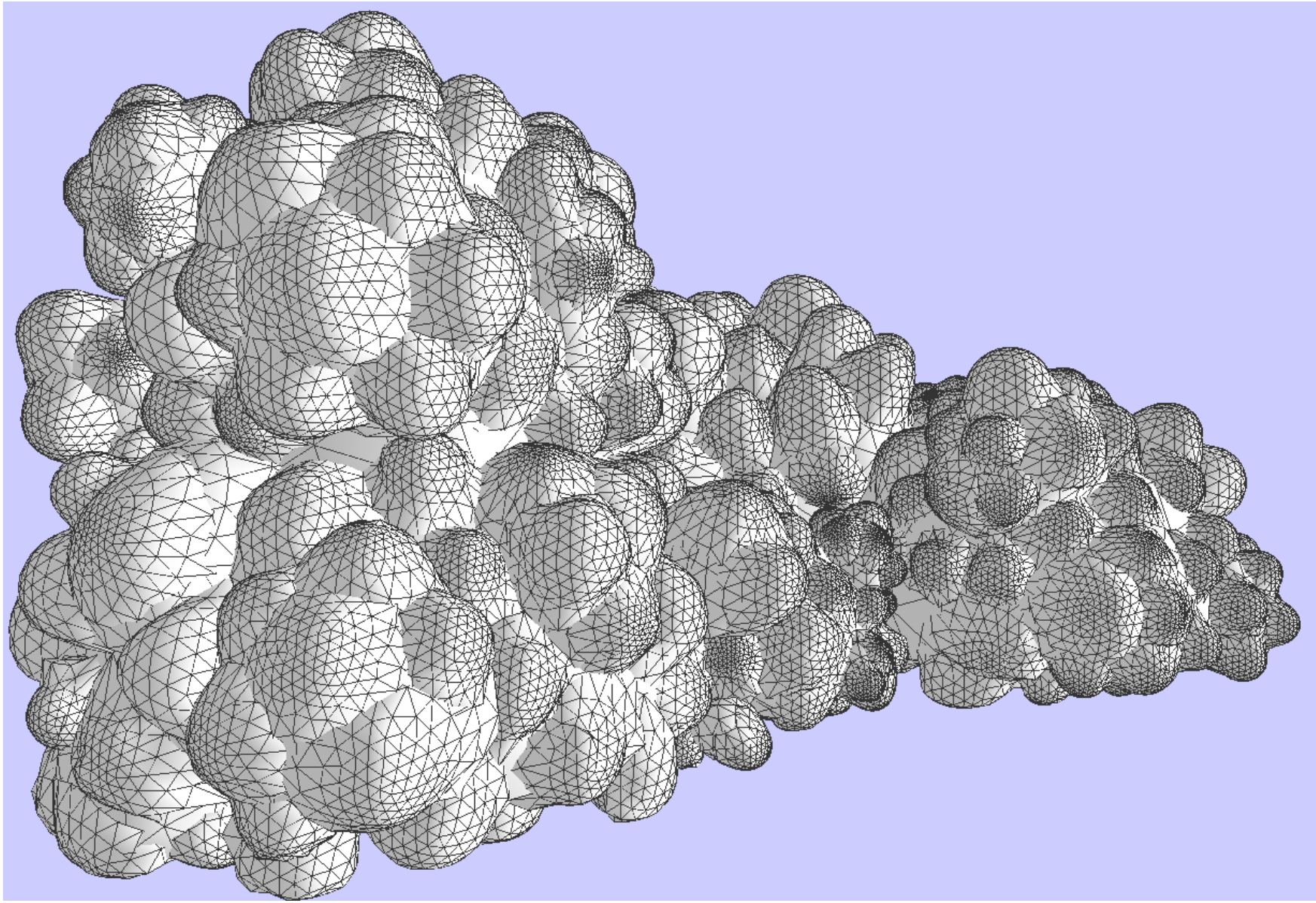
- Model inspired from Gardner's
- Texture simulating higer levels

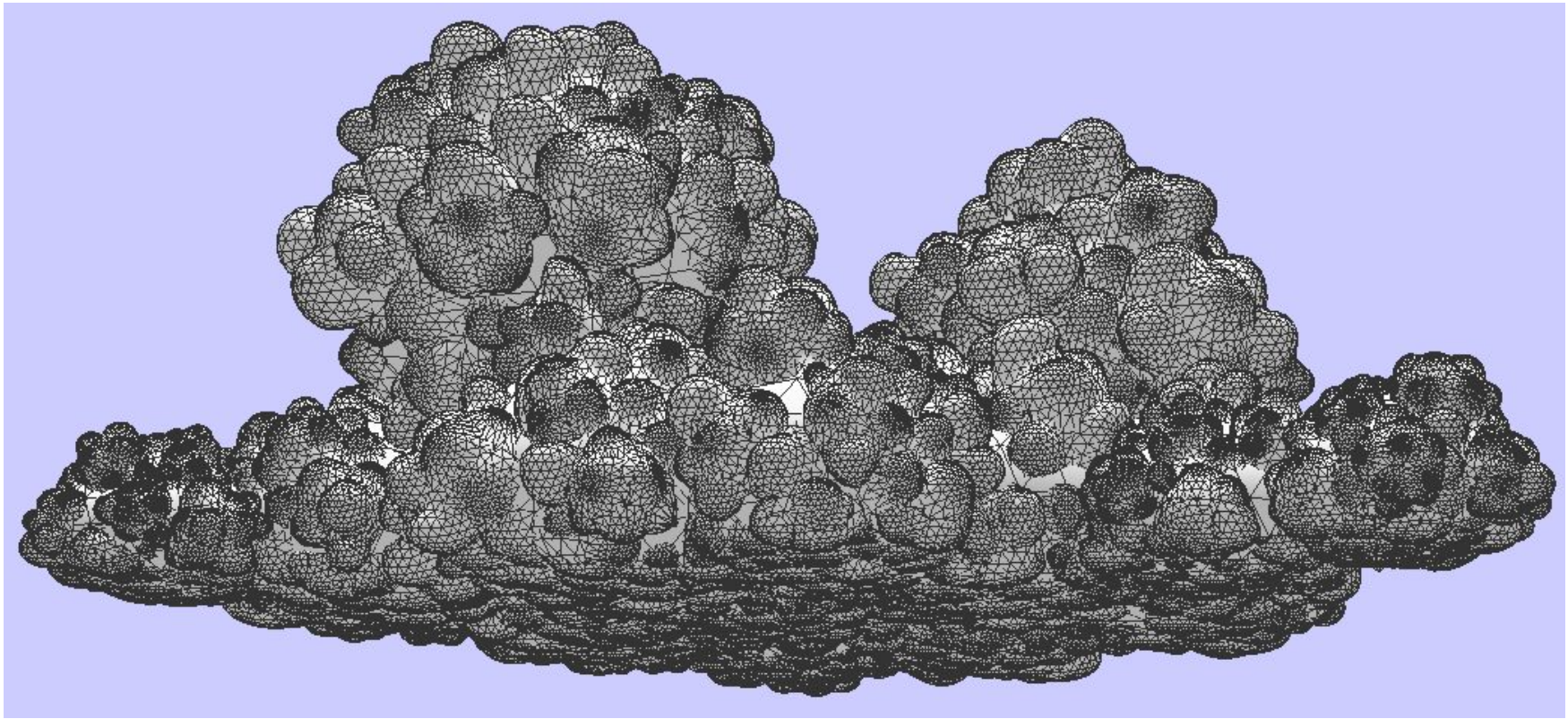
# 4 Results

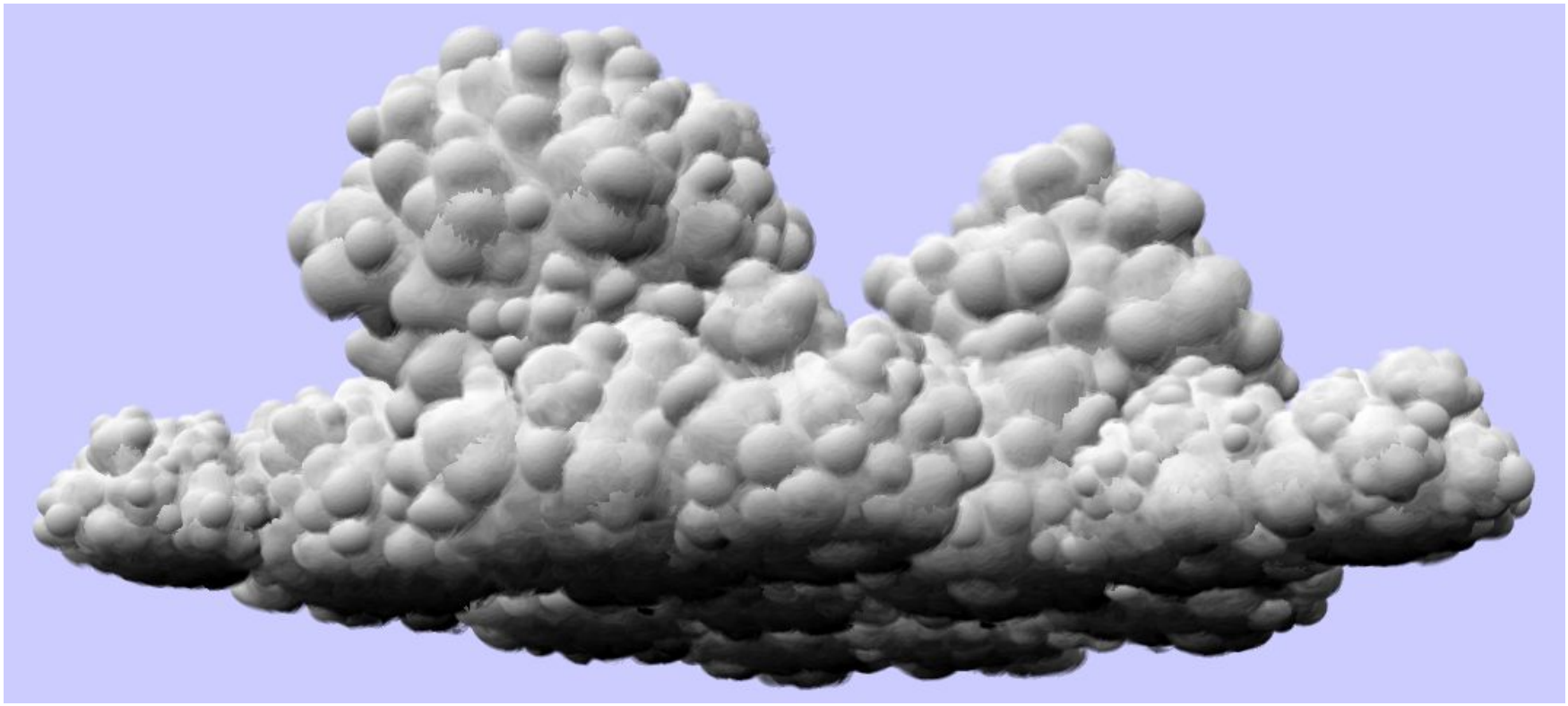


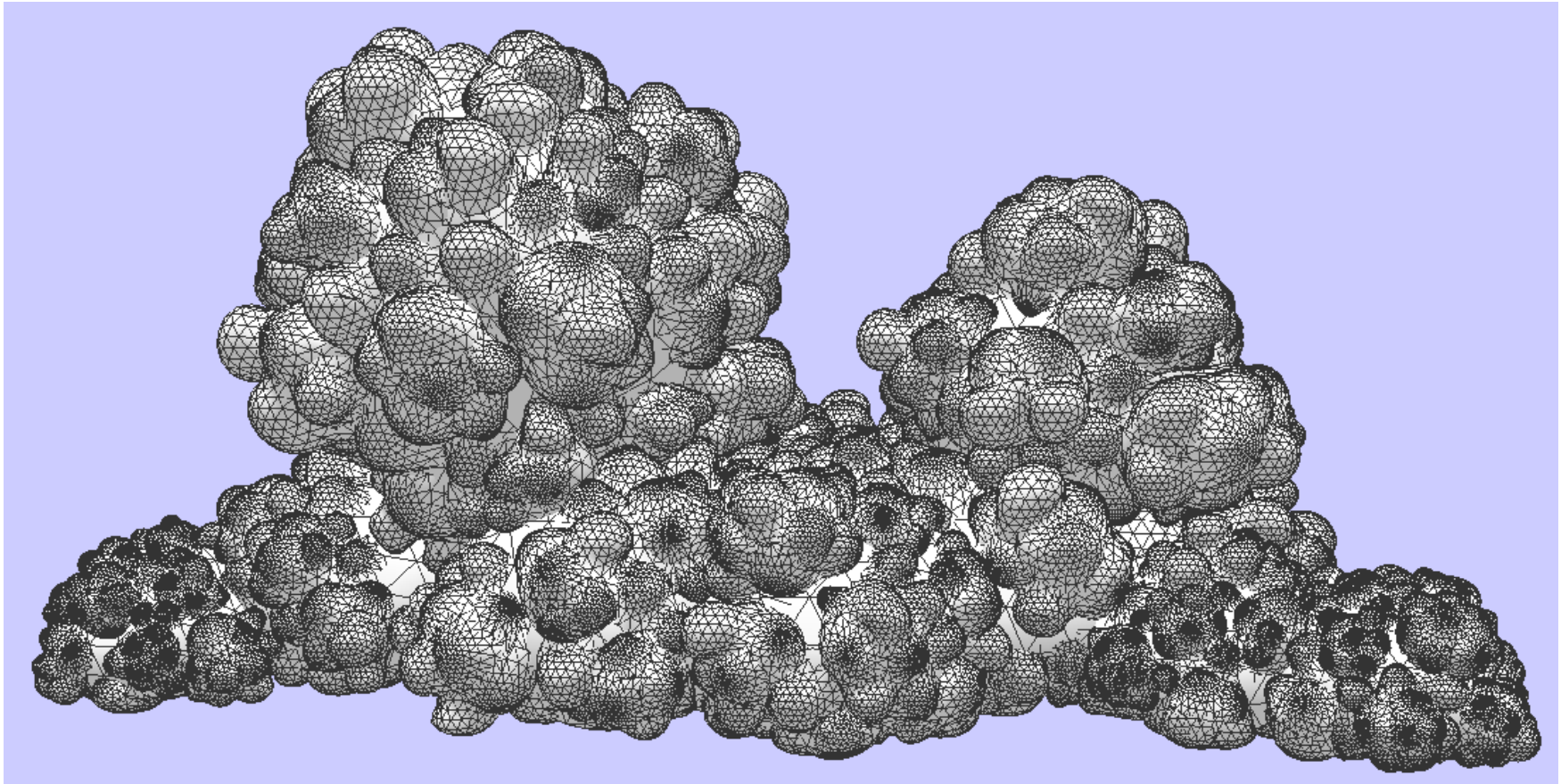


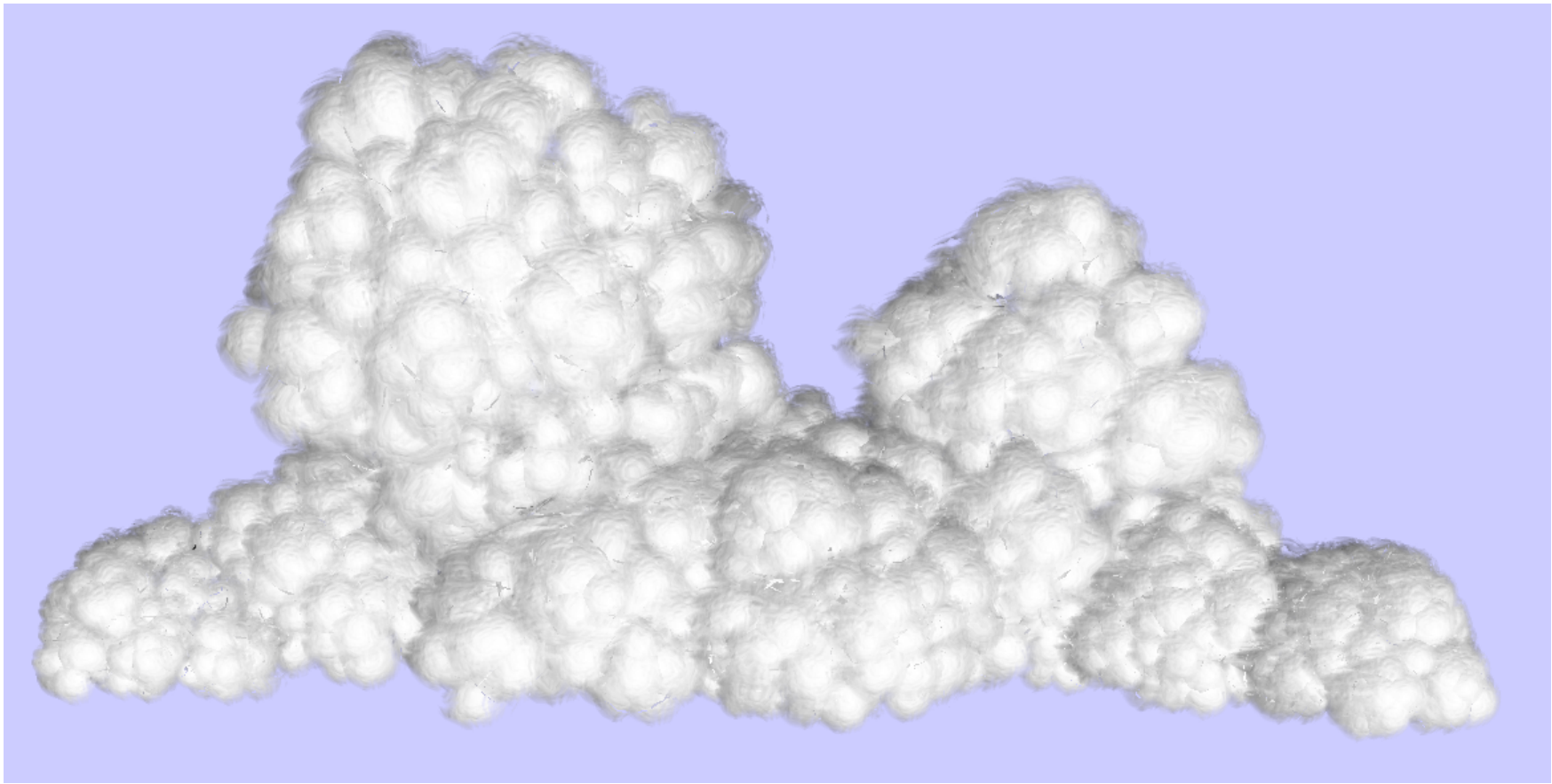


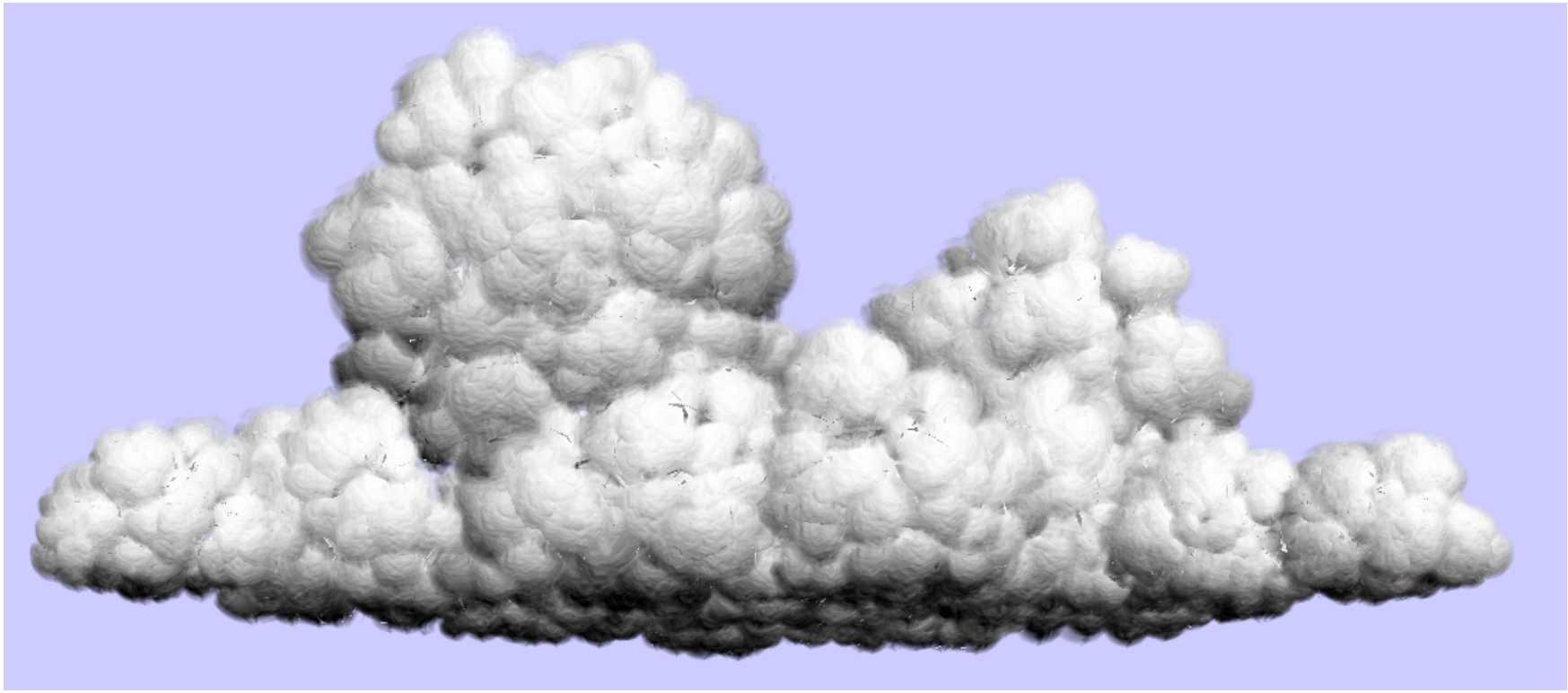












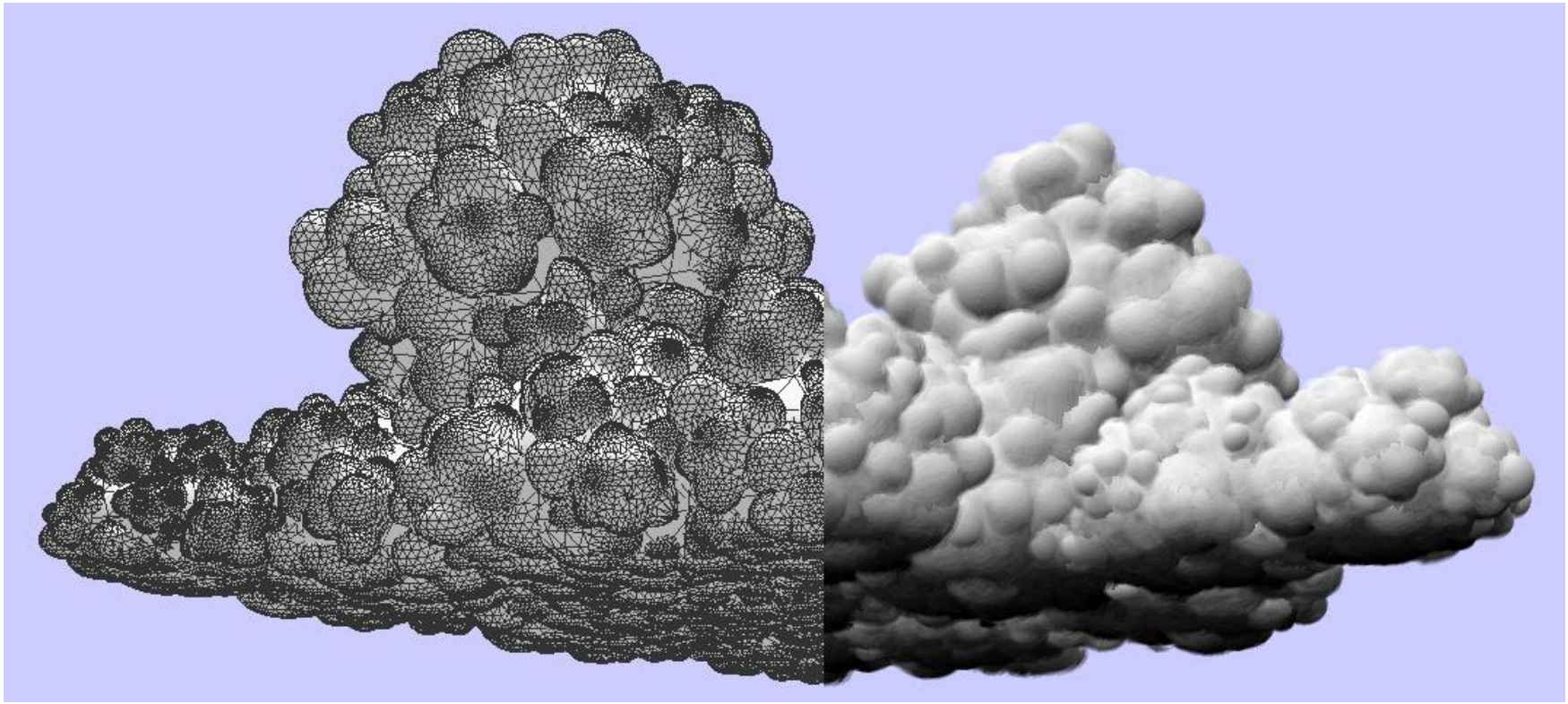
# 5 Conclusion

+ Fast rendering
+ Much detail
+ Animatable [Neyret 97]
- Slow generation

## Future work:

- Huge geometry $\longrightarrow$ adaptive mesh
- Shaders $\longrightarrow$ realistic rendering, complex effects
- Animation
- Long term : cloudy sky, volcano smoke...

# Questions ?



# Some math

$$\text{Blob surface: } S_i = \{\mathbf{P} \in \mathbf{R}^3 / f_i(\mathbf{P}) = 1\} \quad (1)$$

$$\text{Implicit function: } f_i(\mathbf{P}) = g_i(\mathbf{P}) + m_i(\mathbf{P}) + n_i(\mathbf{P}) + o_i(\mathbf{P}) \quad (2)$$

$$\text{Flattened sphere: } g_i(\mathbf{P}) = \exp\left(1 - \frac{\mathrm{d}_i}{r_i(1 - e_i \mathrm{d}_{l-1})}\right) \quad (3)$$

$$\text{Contact surface: } m_i(\mathbf{P}) = \sum_j m_i^j(\mathbf{P}) \quad (4)$$

$$m_i^j(\mathbf{P}) = (1 - \epsilon - g_j(\mathbf{P})) \min(1, g_j^2(\mathbf{P})) \quad (5)$$

$$\text{Base enlarging: } n_i(\mathbf{P}) = b \ \min\left(1, e^{-\frac{I d_{l-1}}{r_i}}\right) e^{1 - \frac{d_i}{r_i}} \quad (6)$$

$$\text{Flat base: } o_i(\mathbf{P}) = g_i(\mathbf{P}) \min\left(0, \frac{\mathsf{height}(\mathbf{P}) - h_0}{\alpha_h}\right) \quad (7)$$