Procedural cloudscapes

Antoine Webanck[1], Yann Cortial[2], Eric Guérin[2], Eric Galin[1]

[1] Université Lyon 1, CNRS, LIRIS, France

[2] INSA-Lyon, CNRS, LIRIS, France

EUROGRAPHICS 2018

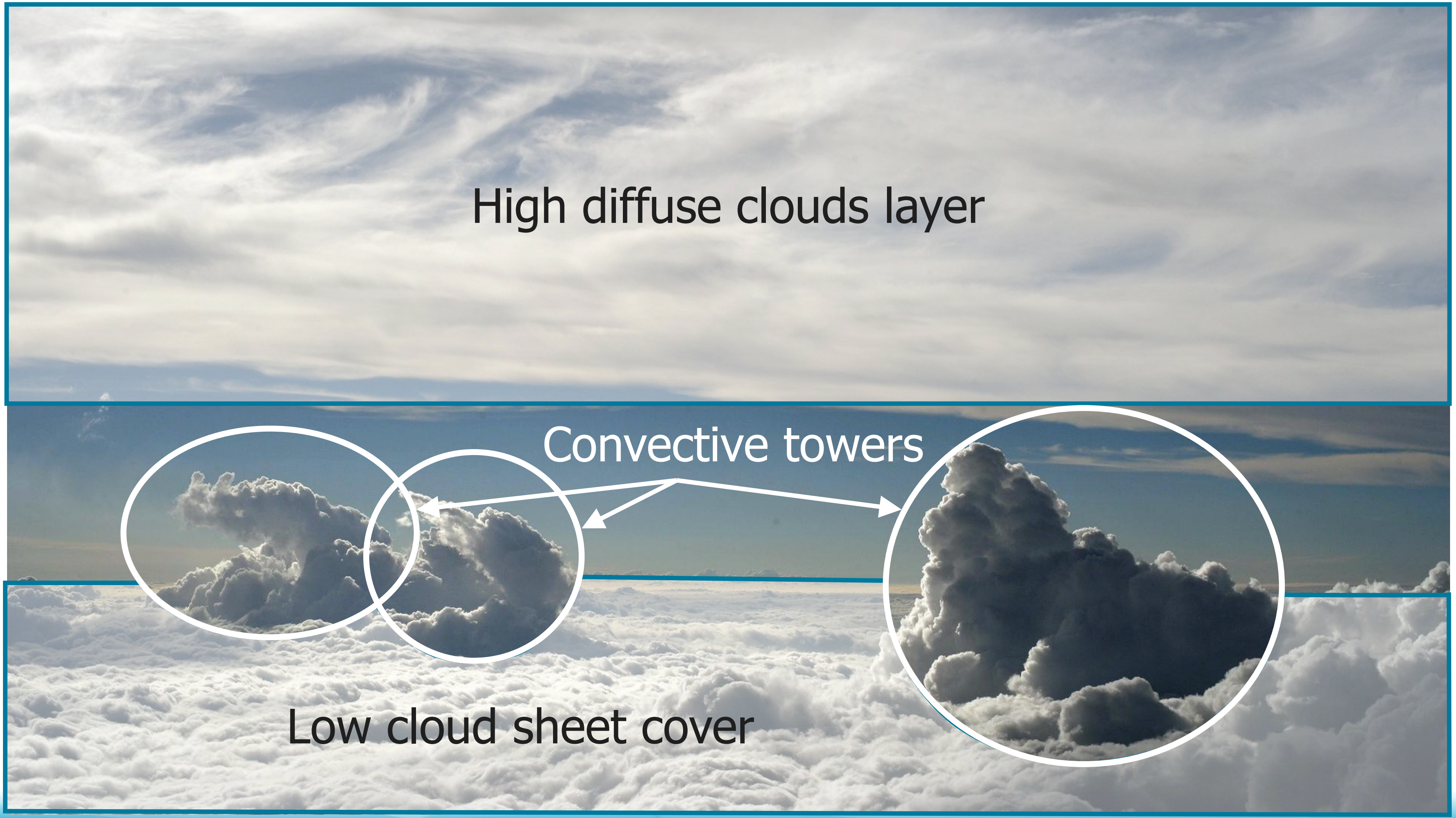
Introduction

1. Cloudscape model
2. Cloudscape morphing algorithm

Results

Conclusion

Real example



Problem

- **Modeling** and **animation** of large scale cloudscapes
- User control

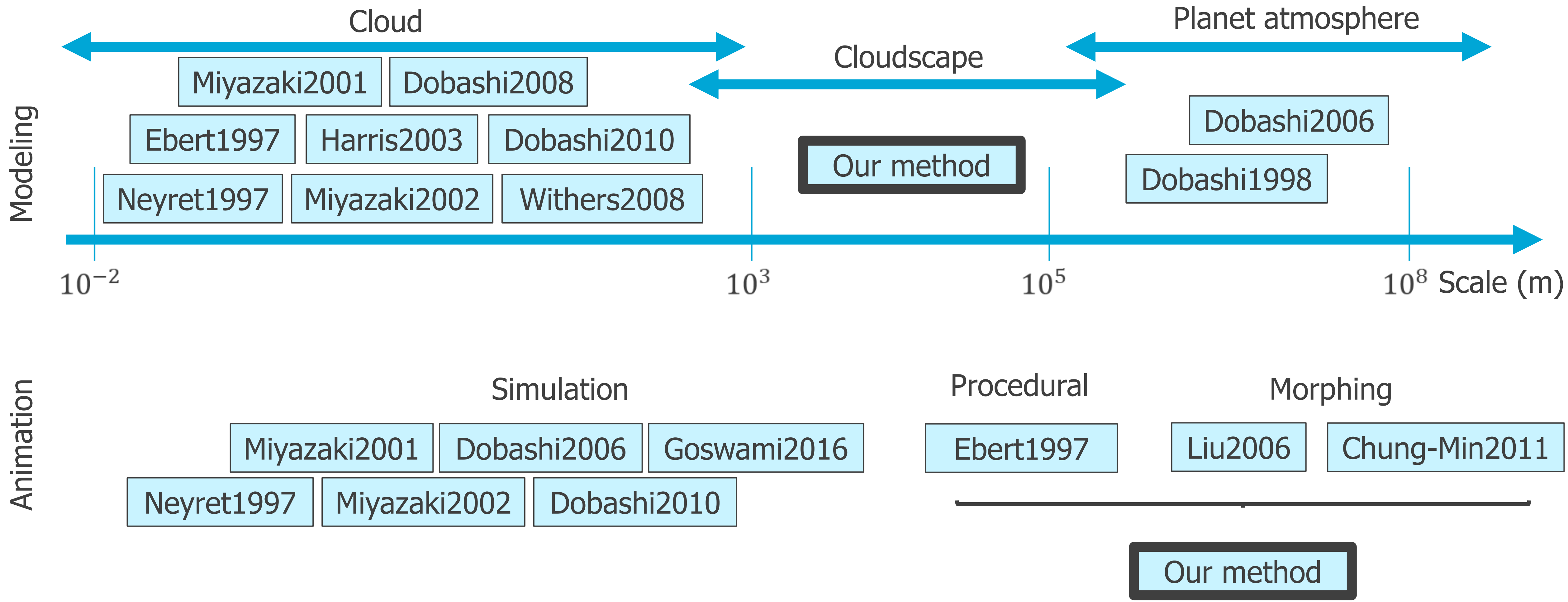
Scientific challenges

- Define a 4D (3D + t) function

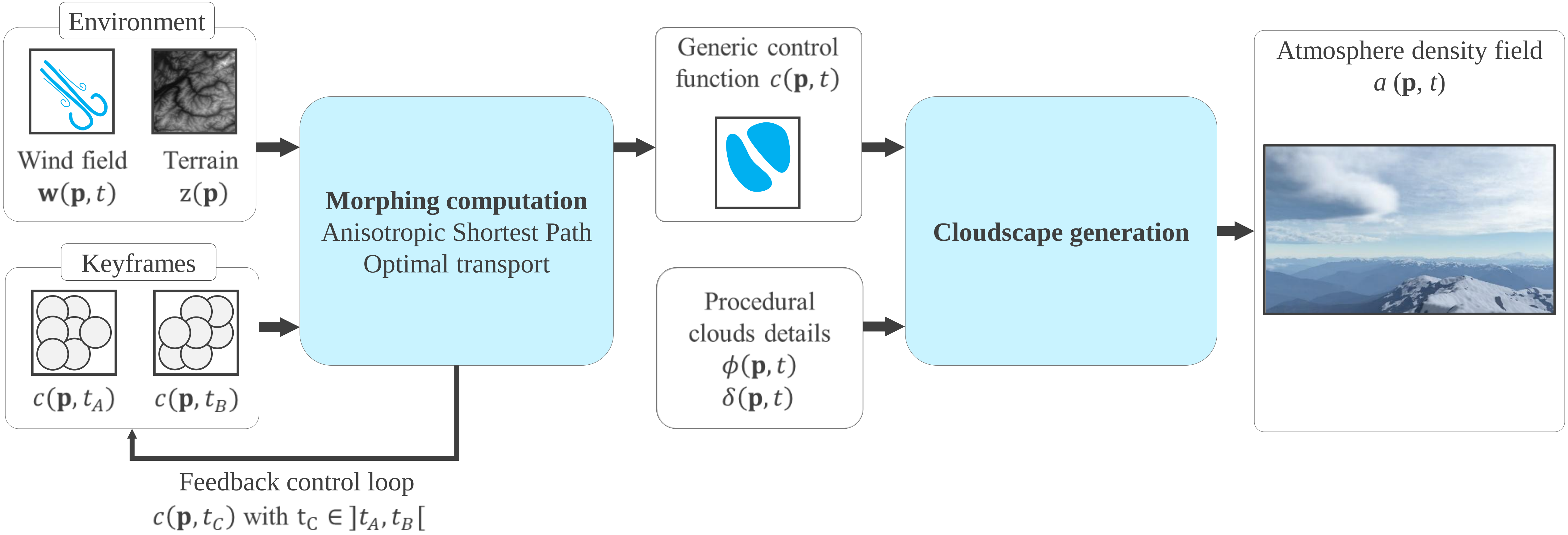
$$a(\mathbf{p}, t): \mathbb{R}^4 \to \mathbb{R}^+$$

- Authoring such a function is complex
- Chaotic natural phenomena (thermodynamics, fluid physics)

Related work: atmosphere animation



Pipeline preview



Introduction

1. Cloudscape model

2. Cloudscape morphing algorithm

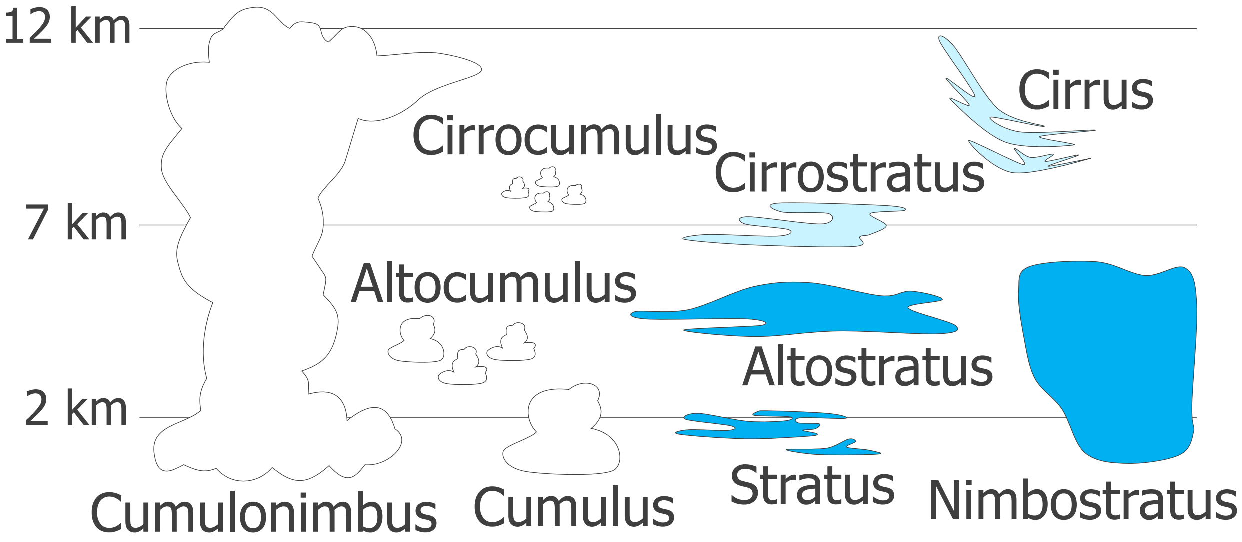
Results

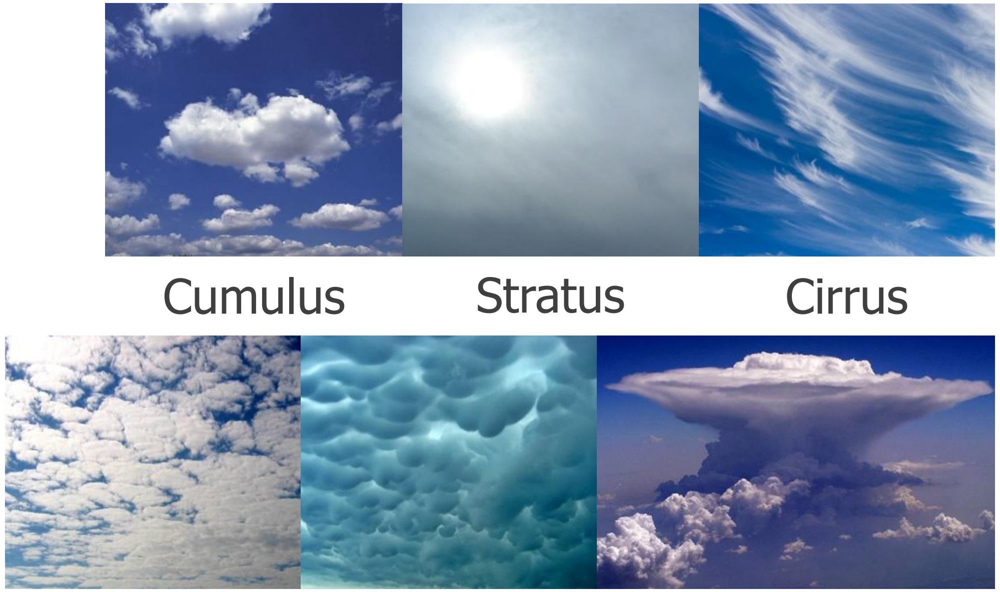
Conclusion

Clouds in meteorology

Clouds names refer to their shape and altitudes

- Cumuliform clouds have prominent convective bulbs
- Stratiform clouds are almost homogeneous layers
- Cirriform clouds are high and look like hair





Cumulus Stratus Cirrus

Altocumulus Mammatus Cumulonimbus

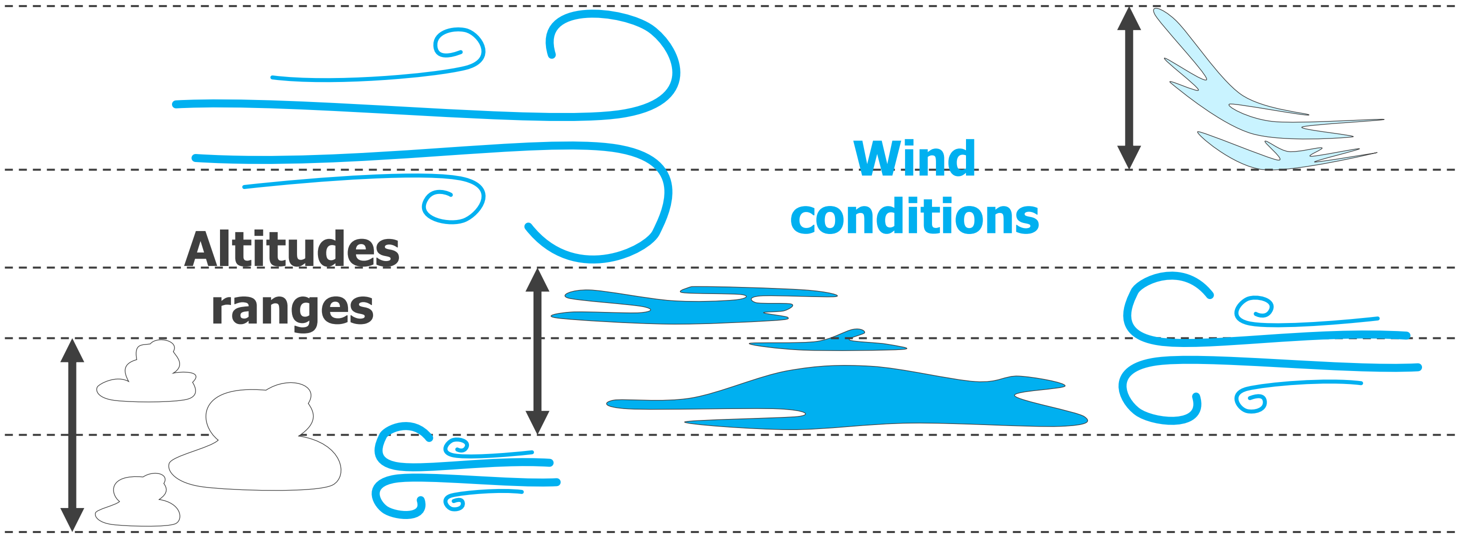
Different clouds at different altitudes

Type	Range (km)
Cirrus	1 to 2
Stratus	0.2 to 0.8
Cumulus	0.1 to 5
Cumulonimbus	7 to 12

Global cloudscape model

- Cloud density function $a(\mathbf{p}, t): \mathbb{R}^4 \to \mathbb{R}^+$

- Each cloud type in a different layer
  - altitude range $[a, b]$
  - wind field $\mathbf{w}(\mathbf{p}, t)$
  - **control function**
  - **shape and details functions**



- Cloud density function of layer i

  $a_i(\mathbf{p}, t): \mathbb{R}^4 \to \mathbb{R}^+$

- Blend of all layers

  $a(\mathbf{p}, t) = \sum_i a_i(\mathbf{p}, t)$

Control function

- **Large-scale control function** for the cloud cover

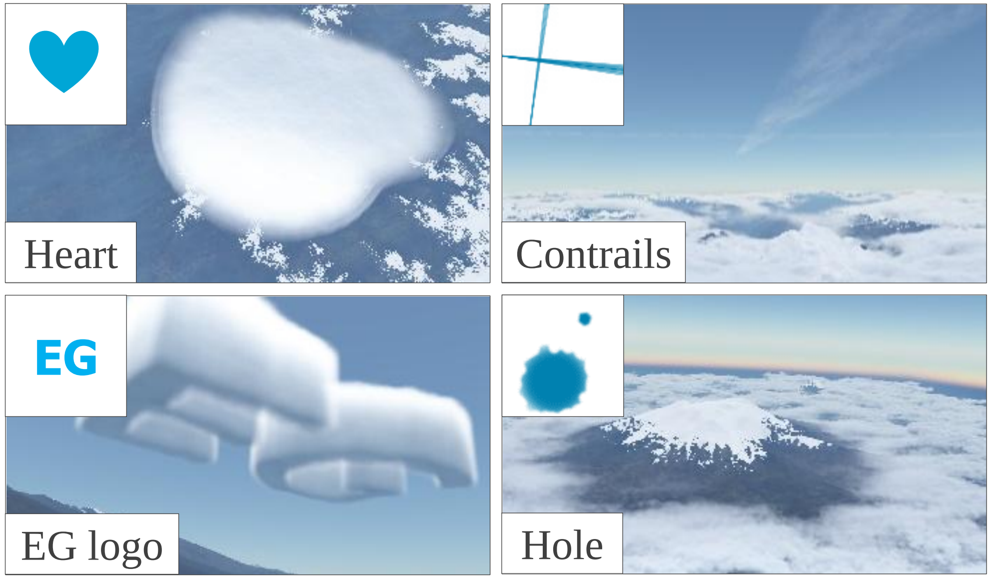
  $$c(\mathbf{p},t): \mathbb{R}^4 \rightarrow \mathbb{R}^+$$

- Defined manually for special effects

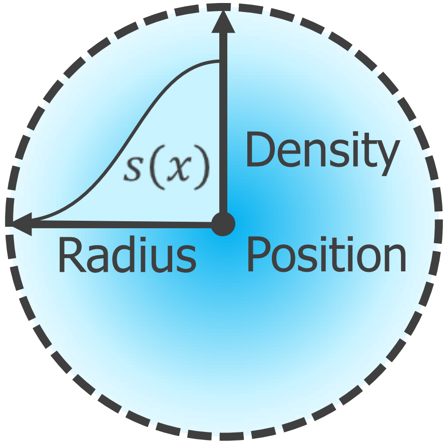
- Defined with primitives for morphing

  $$c(\mathbf{p},t) = \sum_j c_j(\mathbf{p},t)$$

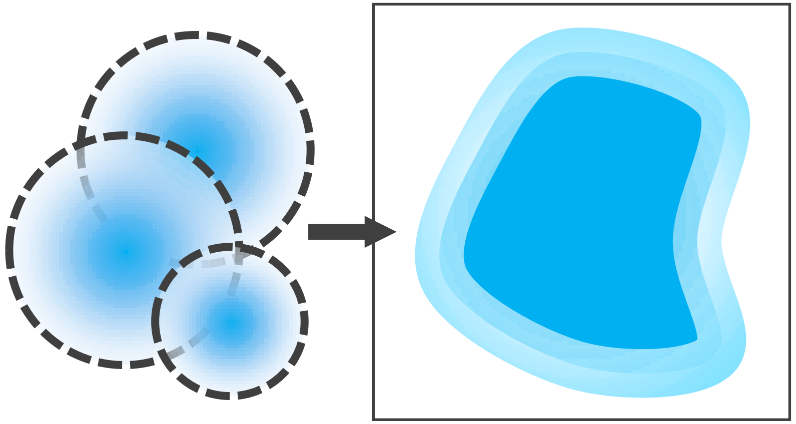
- Primitives parameters
  - Maximum density
  - Position
  - Radius



Cover special effects



Control primitive


Primitives
Control function

Clouds shape and details

- Control function and procedural noise

$$a_i(\mathbf{p},t) = f\big(c(\mathbf{p},t), \phi(\mathbf{p},t), \delta(\mathbf{p},t)\big)$$

- Smooth control function

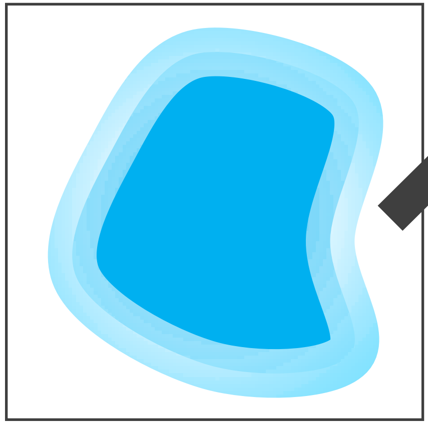
$$c(\mathbf{p},t) : \mathbb{R}^4 \to \mathbb{R}^+$$

- Large shape

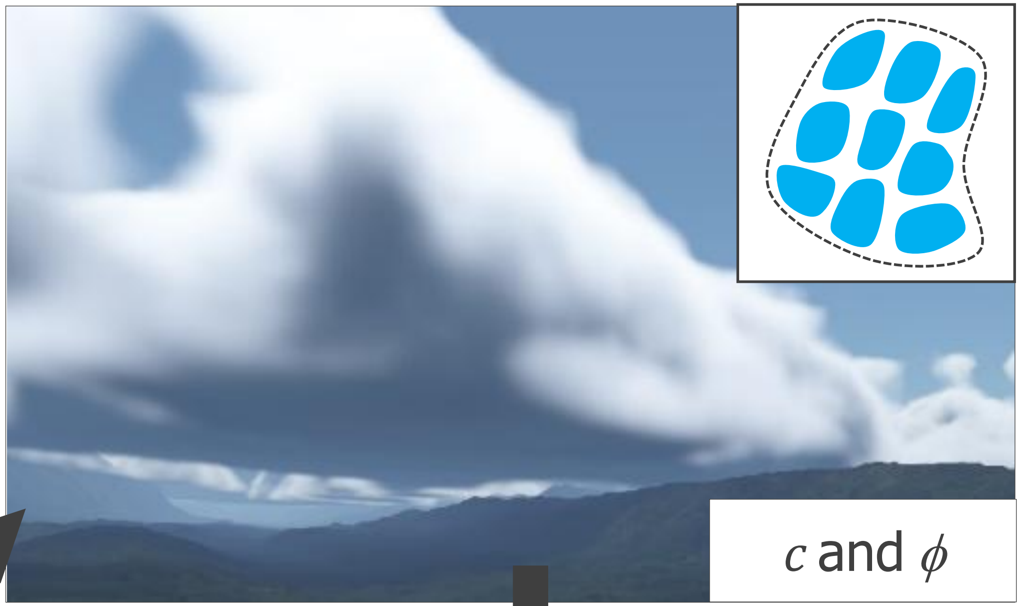
$$\phi(\mathbf{p},t) : \mathbb{R}^4 \to \mathbb{R}^+$$

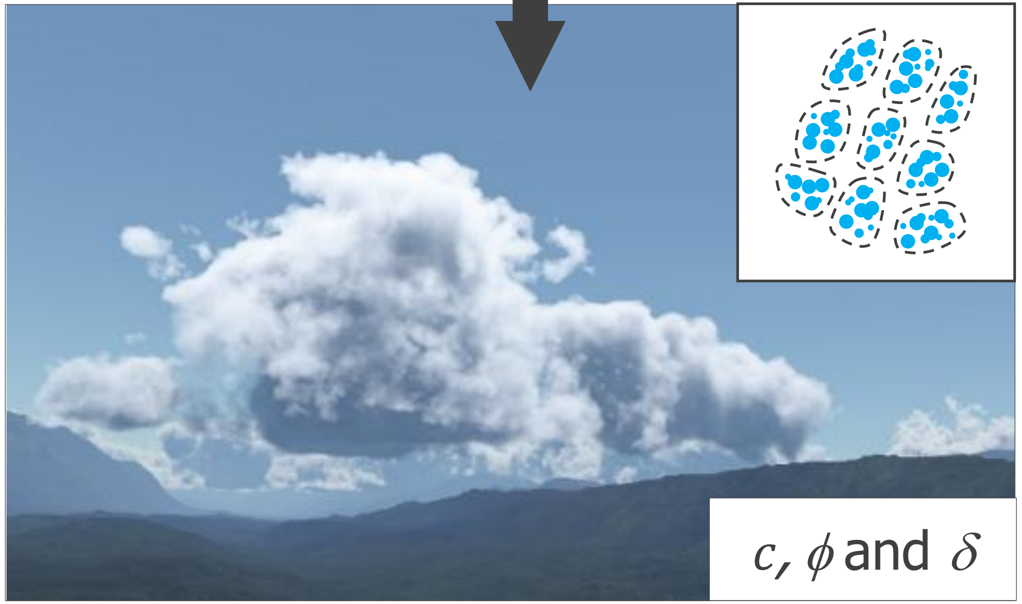
- Smaller details

$$\delta(\mathbf{p},t) : \mathbb{R}^4 \to \mathbb{R}^+$$



c





Introduction

1. Cloudscape model

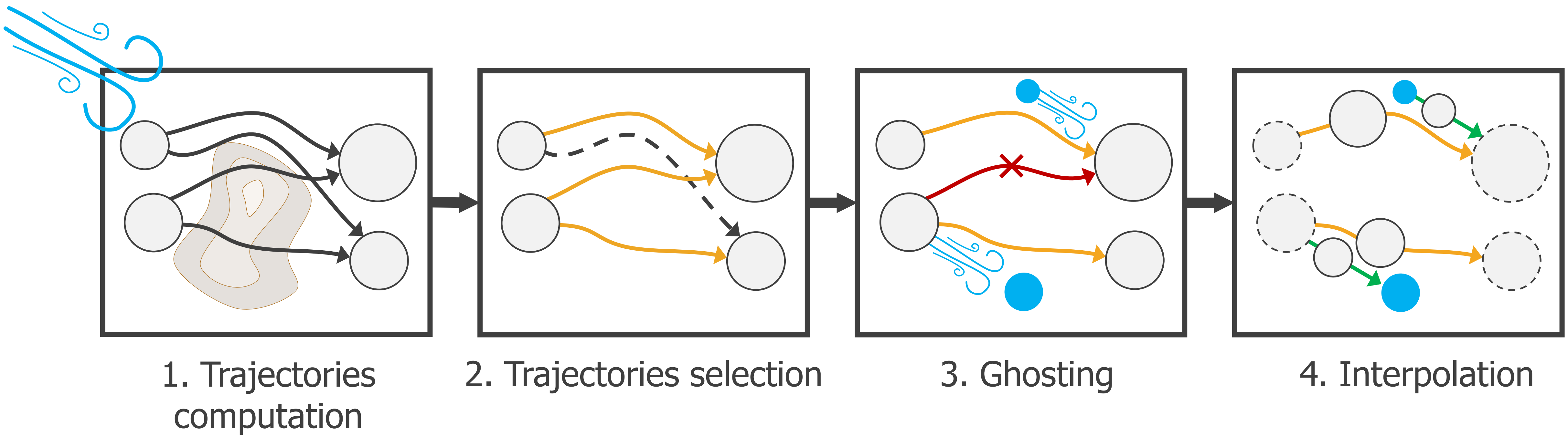
2. Cloudscape morphing algorithm

Results

Conclusion

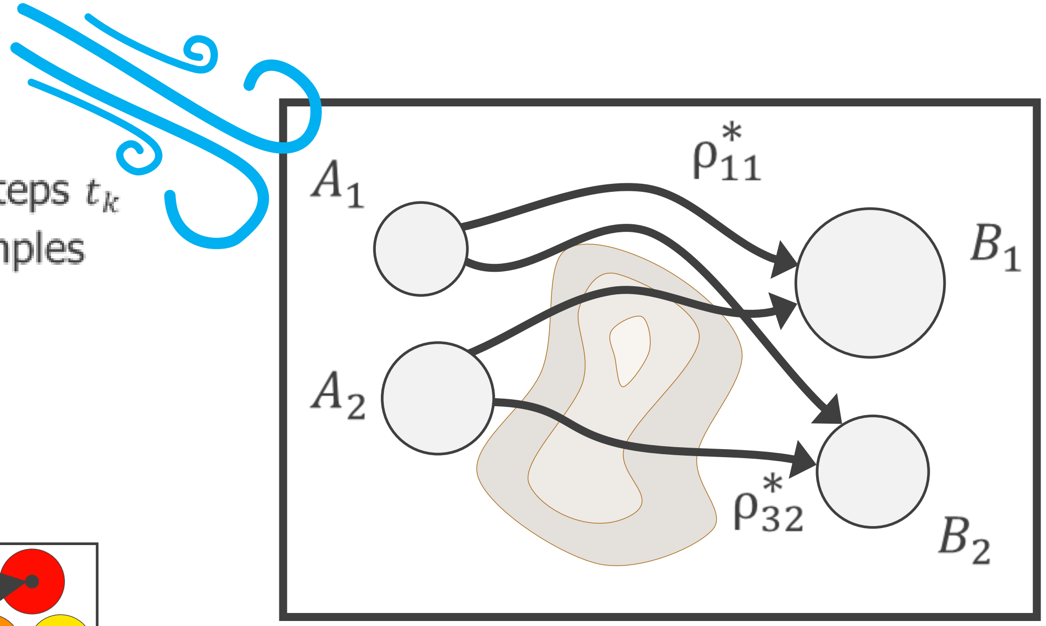
Cloudscape morphing

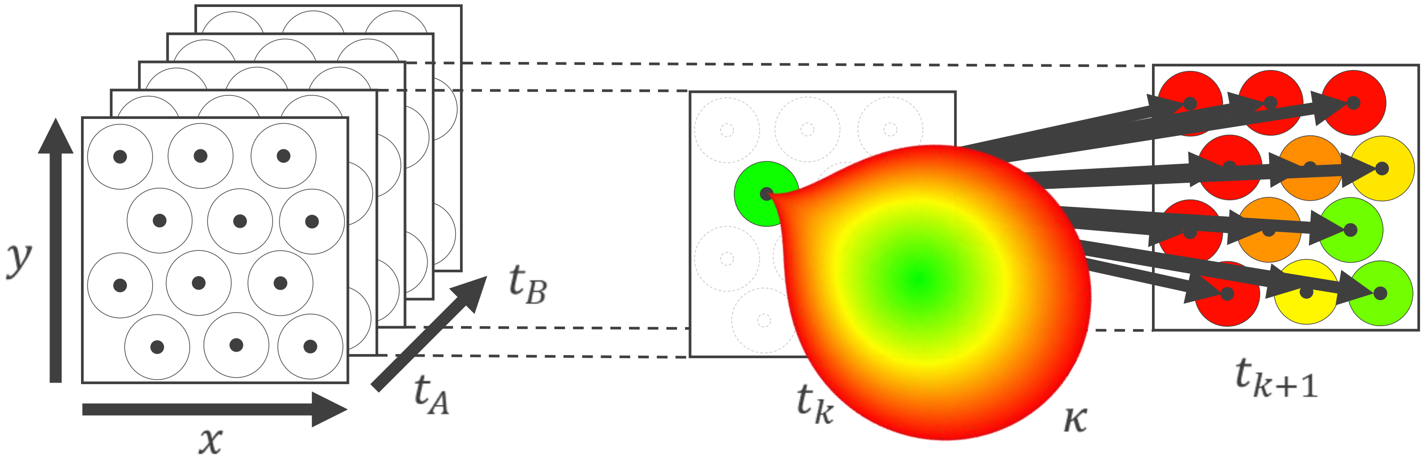
- Animate each cloud layer separately
- Morph between subsequent keyframes primitives



Step 1: trajectories computation

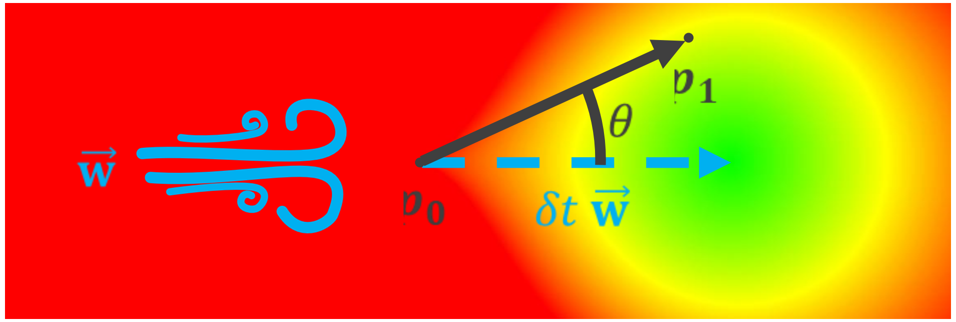
- Compute all **anisotropic shortest paths** ρ_{ij}^*
  - Sample the XY domain
  - Discretize time between keyframes $\Delta t = [t_A, t_B]$ in n time steps t_k
  - Define anisotropic cost function κ between subsequent samples
  - Anisotropic shortest path between subsequent samples





Step 1: anisotropic cost function

- Anisotropic cost function
  - $\kappa(\mathbf{p_0}, t_0, \mathbf{p_1}, t_1): \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^+$
  - $\kappa(\mathbf{p_0}, t_0, \mathbf{p_1}, t_1) = \underbrace{\lambda_w \kappa_w}_{\text{Wind}} + \underbrace{\lambda_s \kappa_s + \lambda_e \kappa_e}_{\text{Terrain slope and elevation}}$



$$\kappa_w(\mathbf{p_0}, \mathrm{t}_0, \mathbf{p_1}, \mathrm{t}_1) = \underbrace{\|\mathbf{p_1} - \delta t\, \vec{\mathbf{w}}\|}_{\text{Drag distance}} \cdot \underbrace{\left(2 - cos(\theta)\right)}_{\text{Angle}}$$

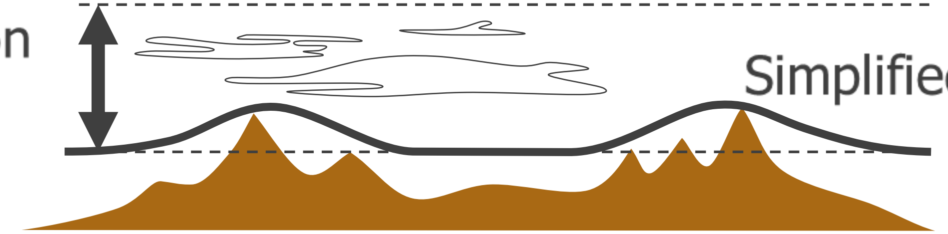
$$\kappa_s(\mathbf{p_0}, \mathbf{p_1}) = \nabla \tilde{z}(\mathbf{p_0}) \cdot \frac{\mathbf{p_1} - \mathbf{p_0}}{\|\mathbf{p_1} - \mathbf{p_0}\|}$$

Slope cost

$$\kappa_e(\mathbf{p_0}, \mathbf{p_1}) = \frac{\tilde{z}(\mathbf{p_0}) - a}{b - a}$$

Overlap cost

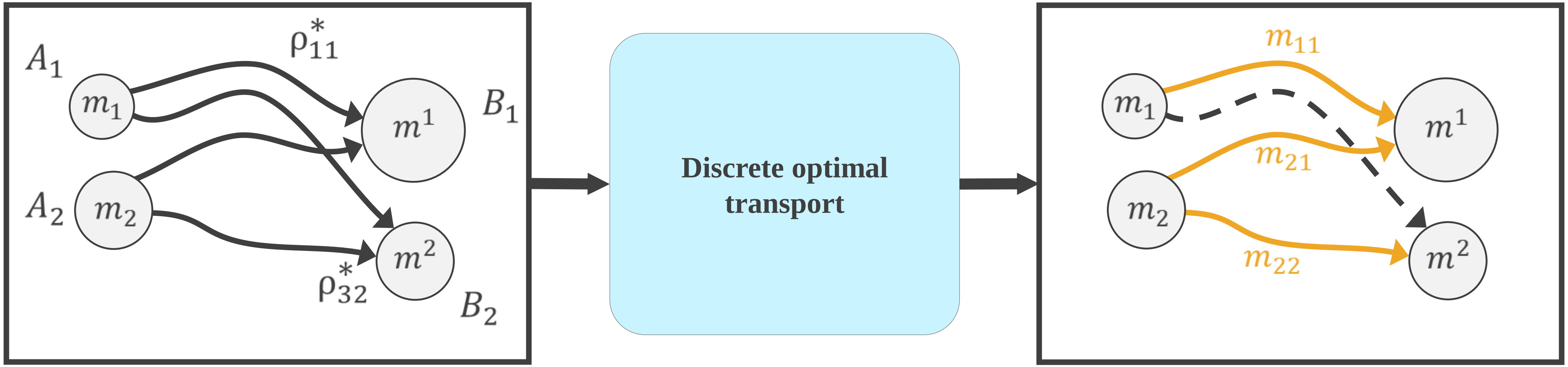
Elevation range $[ab]$



Simplified terrain $\tilde{\mathbf{z}}(\mathbf{p})$

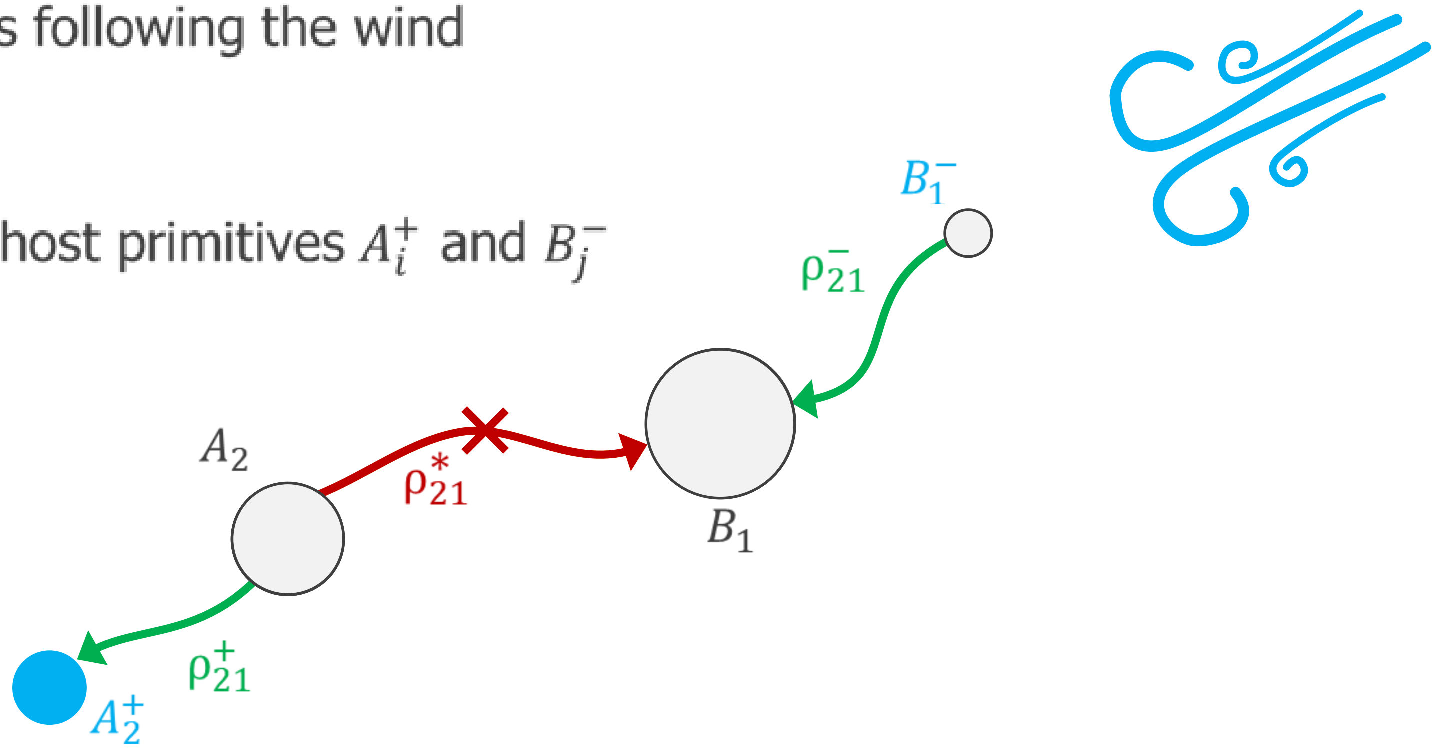
Step 2: trajectories selection

- $n \times m$ known trajectory costs
- Compute mass of initial and final primitives m_i and m^j
- Apply optimal transport to optimize cost per unit of mass
  - Less than $n + m - 1$ trajectories subset
  - Masses m_{ij}



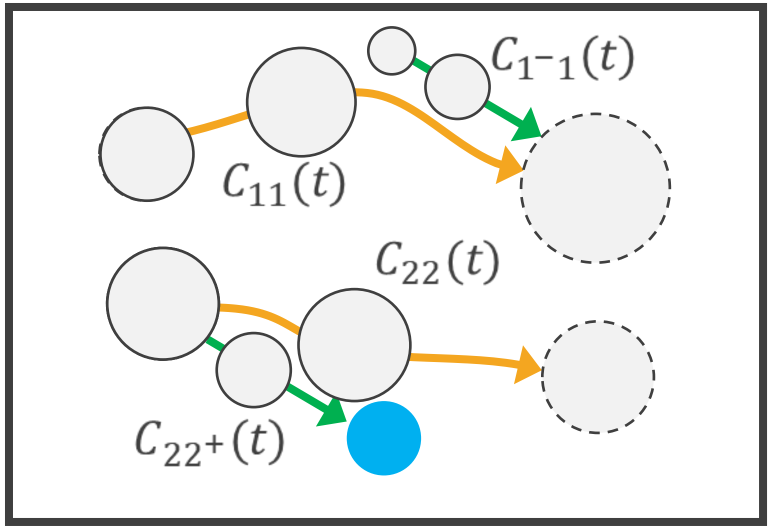
Step 3: ghosting

- Ghost primitives of null density: appearing or vanishing primitives
- Discard expensive trajectories ρ^*_{ij}
- Extrapolate trajectories following the wind
  - Backward (fade in) ρ^-_{ij}
  - Forward (fade out) ρ^+_{ij}
- Redistribute mass to ghost primitives A^+_i and B^-_j



Step 4: interpolation

- One primitive per trajectory
- Interpolate
  - Position following the trajectory
  - Radius linearly
  - Density linearly weighted by relative mass



Introduction

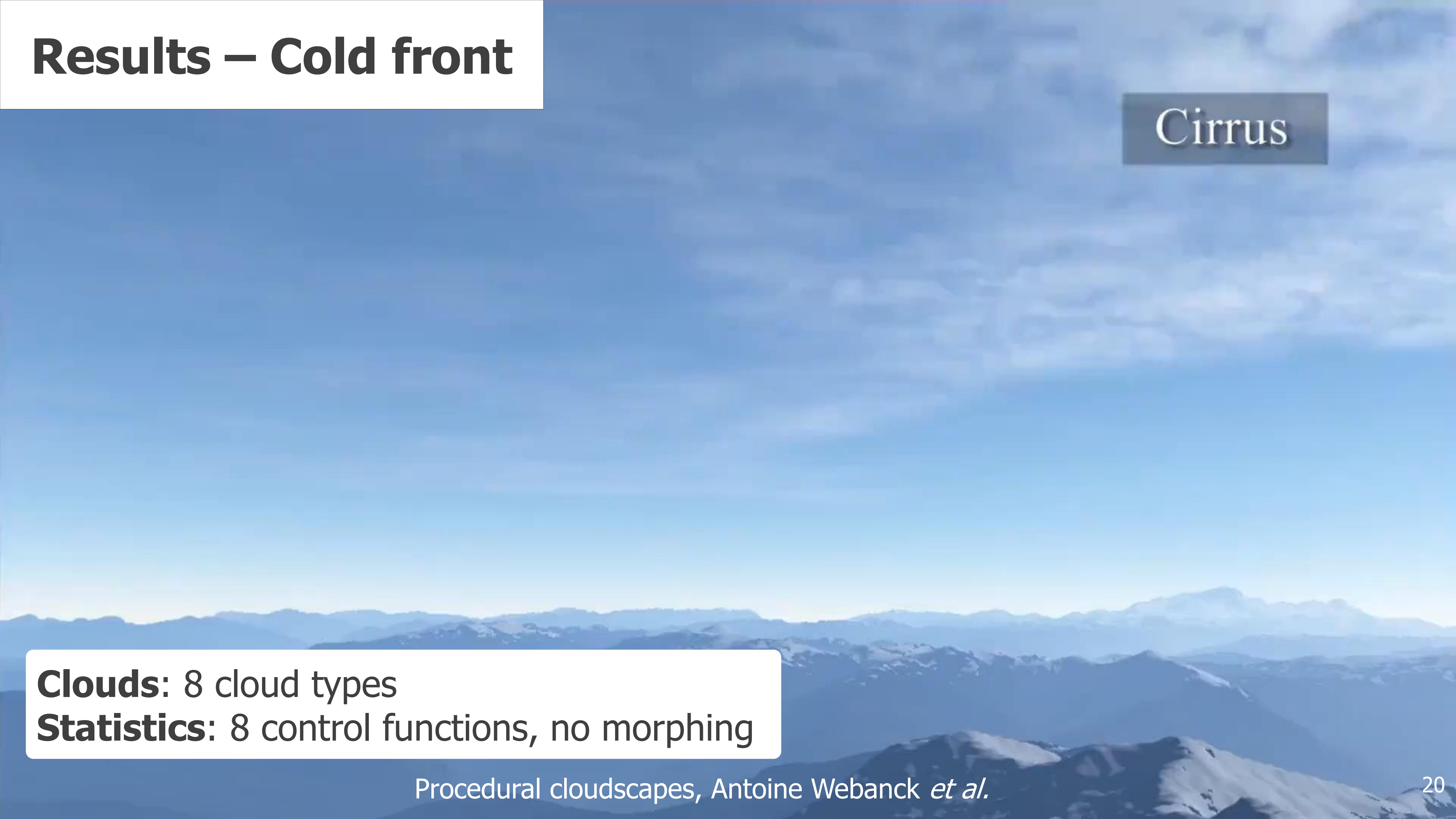
I. Cloudscape model

II. Cloudscape morphing algorithm

Results

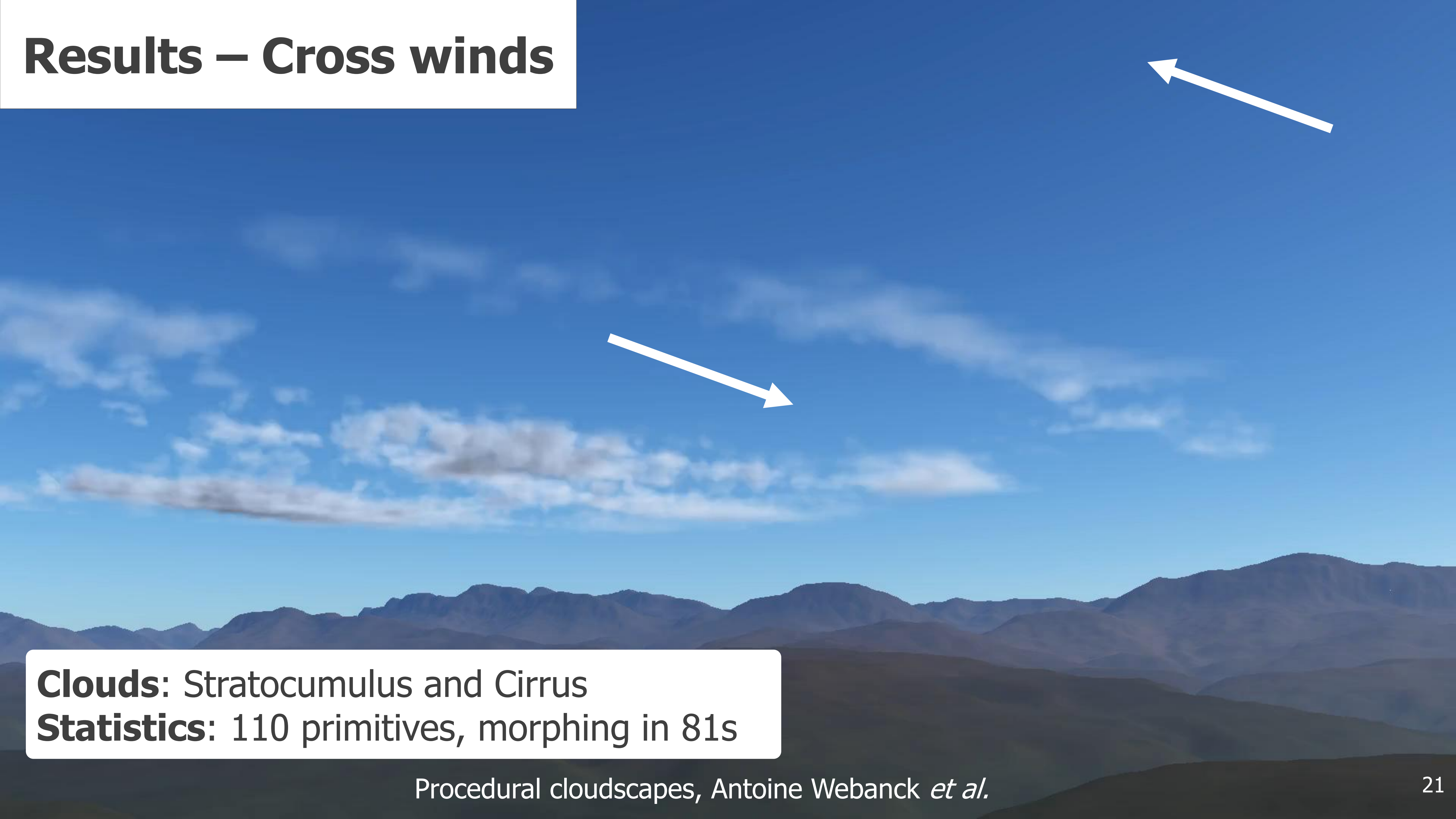
Conclusion

Results – Cold front



Clouds: 8 cloud types
Statistics: 8 control functions, no morphing

Results – Cross winds



Clouds: Stratocumulus and Cirrus
Statistics: 110 primitives, morphing in 81s

Results – Good weather

Clouds: Stratus and Cumulus
Statistics: 460 primitives, morphing in 68s

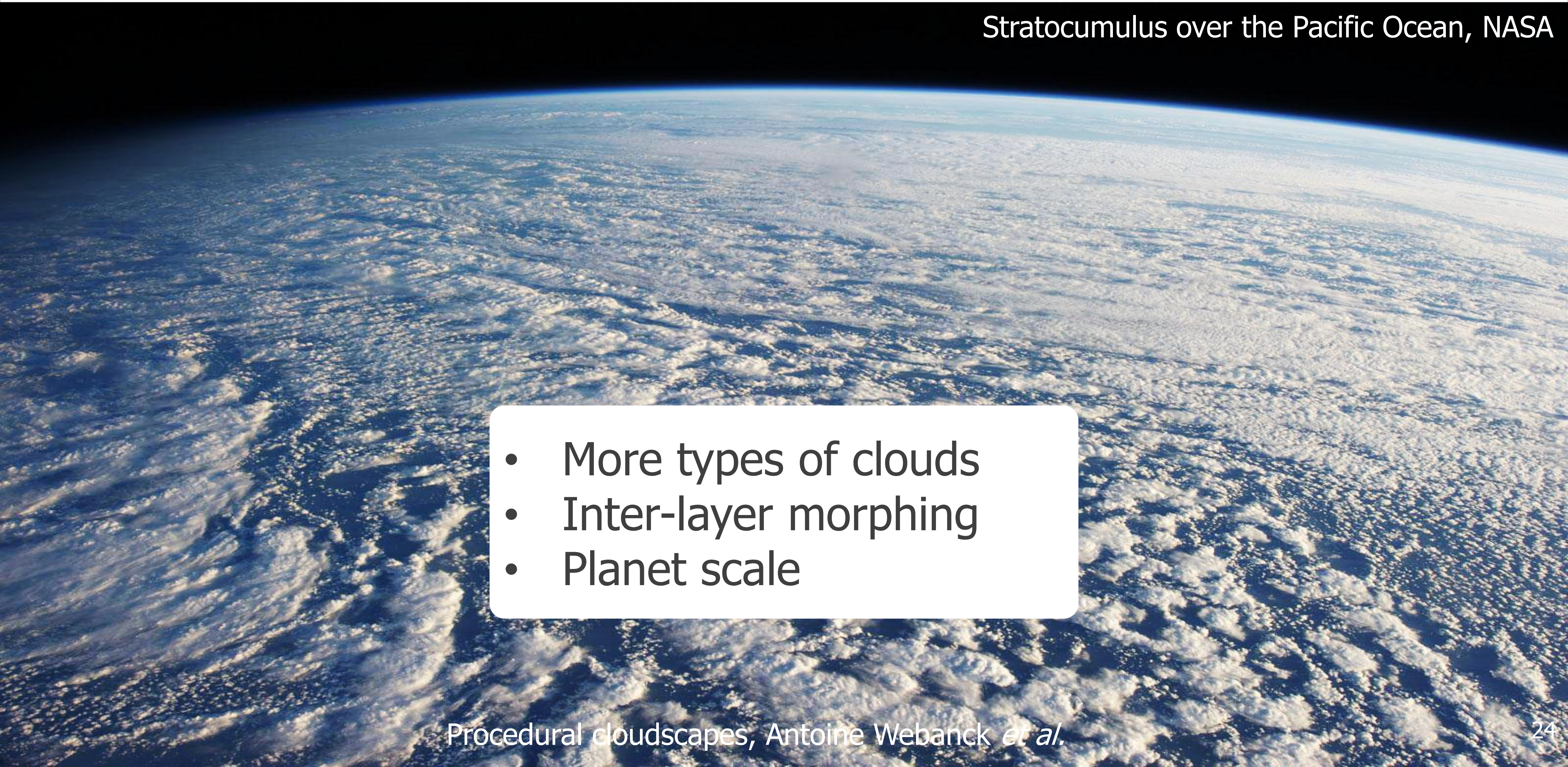
Procedural cloudscapes, Antoine Webanck *et al.*

Conclusion

Controllable procedural animated cloudscape

- Efficient model with procedural details
- Controllable by intuitive key-framing
- Coherent morphing accounting for environment

Future work



Stratocumulus over the Pacific Ocean, NASA

- More types of clouds
- Inter-layer morphing
- Planet scale

Procedural cloudscapes

Antoine Webanck[1], Yann Cortial[2], Eric Guérin[2], Eric Galin[1]

[1] Université Lyon 1, CNRS, LIRIS, France

[2] INSA-Lyon, CNRS, LIRIS, France

EUROGRAPHICS 2018