Procedural cloudscapes

Antoine Webanck\textsuperscript{1}, Yann Cortial\textsuperscript{2}, Eric Guérin\textsuperscript{2}, Eric Galin\textsuperscript{1}

\textsuperscript{1} Université Lyon 1, CNRS, LIRIS, France
\textsuperscript{2} INSA-Lyon, CNRS, LIRIS, France
Introduction

1. Cloudscape model
2. Cloudscape morphing algorithm

Results

Conclusion
Real example

High diffuse clouds layer

Convective towers

Low cloud sheet cover

Procedural cloudscapes, Antoine Webanck et al.
Problem

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

Scientific challenges
- Define a 4D (3D + t) function
  \[ a(p, t): \mathbb{R}^4 \to \mathbb{R}^+ \]
- Authoring such a function is complex
- Chaotic natural phenomena (thermodynamics, fluid physics)
Related work: atmosphere animation

Cloud
- Miyazaki2001
- Dobashi2008
- Ebert1997
- Harris2003
- Dobashi2010
- Neyret1997
- Miyazaki2002
- Withers2008

Cloudscape
- Our method

Planet atmosphere
- Dobashi2006
- Dobashi1998

Modeling
- 10^{-2}
- 10^3
- 10^5
- 10^8 Scale (m)

Animation
- Simulation
  - Miyazaki2001
  - Dobashi2006
  - Goswami2016
  - Neyret1997
  - Miyazaki2002
  - Dobashi2010

- Procedural
  - Ebert1997
  - Liu2006
  - Chung-Min2011

- Morphing

Procedural cloudscapes, Antoine Webanck et al.
Pipeline preview

Morphing computation
Anisotropic Shortest Path
Optimal transport

Environment
Wind field $w(p, t)$
Terrain $z(p)$

Keyframes
$c(p, t_A)$, $c(p, t_B)$

Generic control function $c(p, t)$

Procedural clouds details
$\phi(p, t)$
$\delta(p, t)$

Cloudscape generation

Atmosphere density field $a(p, t)$

Feedback control loop
$c(p, t_C)$ with $t_C \in ]t_A, t_B[$

Procedural cloudscapes, Antoine Webanck et al.
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

Different clouds at different altitudes

<table>
<thead>
<tr>
<th>Type</th>
<th>Range (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cirrus</td>
<td>1 to 2</td>
</tr>
<tr>
<td>Stratus</td>
<td>0.2 to 0.8</td>
</tr>
<tr>
<td>Cumulus</td>
<td>0.1 to 5</td>
</tr>
<tr>
<td>Cumulonimbus</td>
<td>7 to 12</td>
</tr>
</tbody>
</table>
Global cloudscape model

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

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

- Cloud density function of layer \(i\)
  \( a_i(p, t): \mathbb{R}^4 \rightarrow \mathbb{R}^+ \)

- Blend of all layers
  \( a(p, t) = \sum_i a_i(p, t) \)
Control function

- **Large-scale control function** for the cloud cover $c(p, t): \mathbb{R}^4 \rightarrow \mathbb{R}^+$
- Defined manually for special effects

- Defined with primitives for morphing $c(p, t) = \sum_j c_j(p, t)$

- Primitives parameters
  - Maximum density
  - Position
  - Radius

Procedural cloudscapes, Antoine Webanck *et al.*
Clouds shape and details

- Control function and procedural noise
  \[ a_i(p, t) = f(c(p, t), \phi(p, t), \delta(p, t)) \]

- Smooth control function
  \[ c(p, t) : \mathbb{R}^4 \to \mathbb{R}^+ \]

- Large shape
  \[ \phi(p, t) : \mathbb{R}^4 \to \mathbb{R}^+ \]

- Smaller details
  \[ \delta(p, t) : \mathbb{R}^4 \to \mathbb{R}^+ \]
Introduction

1. Cloudscape model

2. Cloudscape morphing algorithm

Results

Conclusion
Cloudscape morphing

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

1. Trajectories computation
2. Trajectories selection
3. Ghosting
4. Interpolation

Procedural cloudscapes, Antoine Webanck et al.
Step 1: trajectories computation

- Compute all **anisotropic shortest paths** $\rho_{ij}^*$
  - Sample the XY domain
  - Discretize time between keyframes $\Delta t = [t_A, t_B]$ in $n$ time steps $t_k$
  - Define anisotropic cost function $\kappa$ between subsequent samples
  - Anisotropic shortest path between subsequent samples
Step 1: anisotropic cost function

- Anisotropic cost function
  - \( \kappa(p_0, t_0, p_1, t_1) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^+ \)
  - \( \kappa(p_0, t_0, p_1, t_1) = \lambda_w \kappa_w + \lambda_s \kappa_s + \lambda_e \kappa_e \)

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

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

\( \kappa_w(p_0, t_0, p_1, t_1) = \|p_1 - \delta t \vec{w}\| \cdot (2 - \cos(\theta)) \)

Drag distance

Angle

Procedural cloudscapes, Antoine Webanck et al.
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 $\rho_{ij}^*$
- Extrapolate trajectories following the wind
  - Backward (fade in) $\rho_{ij}^-$
  - Forward (fade out) $\rho_{ij}^+$
- 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

Procedural cloudscapes, Antoine Webanck et al.
Results – Good weather

Clouds: Stratus and Cumulus
Statistics: 460 primitives, morphing in 68s
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\textsuperscript{1}, Yann Cortial\textsuperscript{2}, Eric Guérin\textsuperscript{2}, Eric Galin\textsuperscript{1}

\textsuperscript{1} Université Lyon 1, CNRS, LIRIS, France
\textsuperscript{2} INSA-Lyon, CNRS, LIRIS, France