Welcome to the slides!

Cloud Study by Luke Howard, 1802

Clouds have fascinated us for millennia.

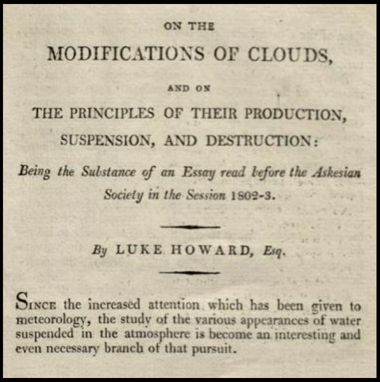
Renee Descartes said of them “We think of them as the throne of god. That makes me hope that if I can explain their nature, one will easily believe that it is possible to find the causes of everything wonderful about earth.”

For centuries, soothsayers and prophets applied this idea quite literally. Clouds and storms were used to foretell drought or political upheaval. But, all of this was done without a scientific understanding of the nature of clouds.

By as late as 1802 the sentiment was that clouds were just clouds and understanding or even classifying them seemed as unreachable as they were to observers from the ground. Think about that.

Out Of Darkness

Luke Howard, Meteorologist, 1802



ON THE

MODIFICATIONS OF CLOUDS,

AND ON

THE PRINCIPLES OF THEIR PRODUCTION,

SUSPENSION, AND DESTRUCTION:

Being the Substance of an Essay read before the Askesian Society in the Session 1802-3.

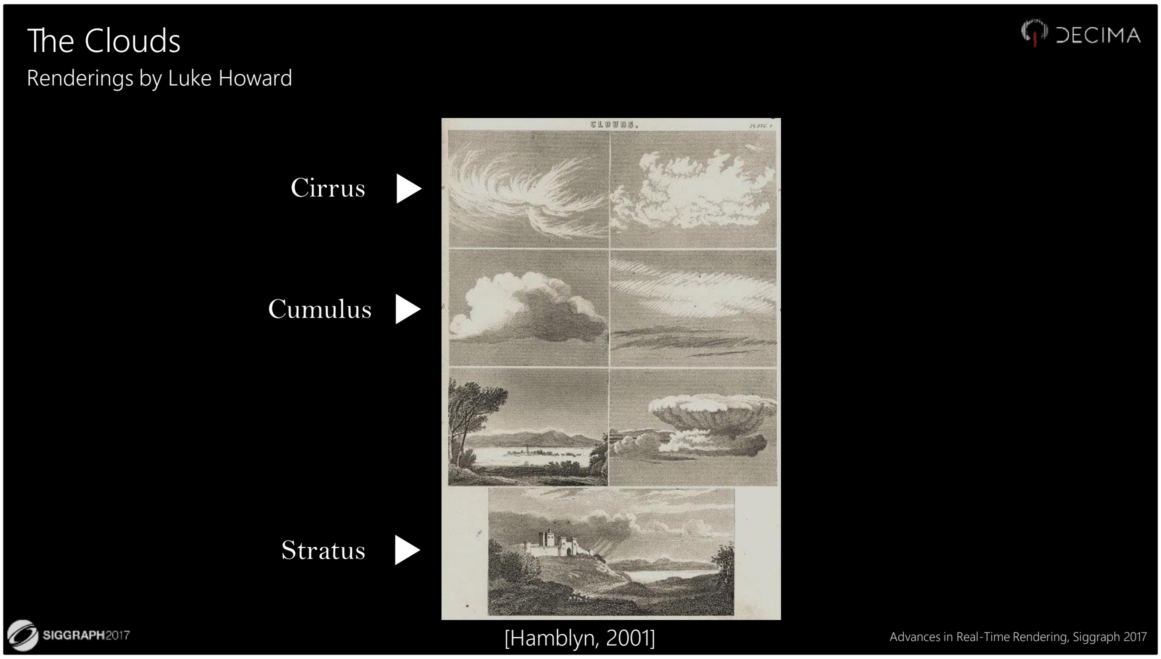
By LUKE HOWARD, *Esq.*

SINCE the increased attention which has been given to meteorology, the study of the various appearances of water suspended in the atmosphere is become an interesting and even necessary branch of that pursuit.

Portrait / Art from the archive of the National Meteorological Library And Archive [UK]

[Hamblyn, 2001]

But one night in that year, a young man named Luke Howard gave a lecture called "On the modifications of clouds" In this lecture, he classified clouds by their altitude, basic physical characteristics and the conditions that gave rise to their development using latin nouns and adjectives.



The Clouds

Renderings by Luke Howard

Cirrus

Cumulus

Stratus

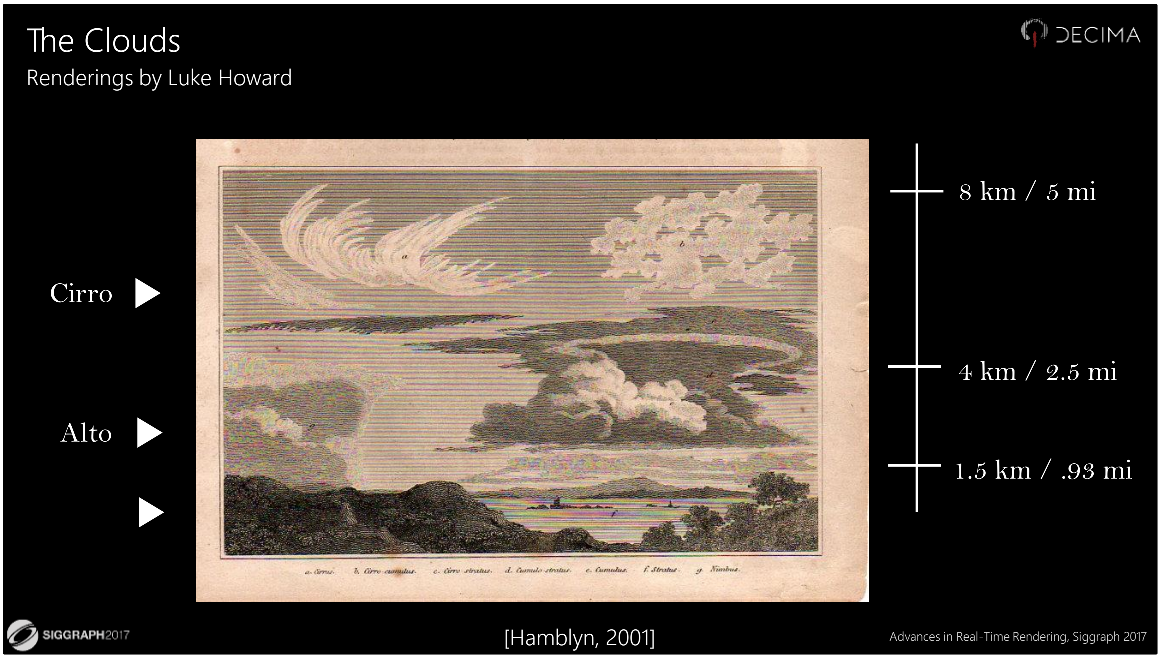
[Hamblyn, 2001]

Advances in Real-Time Rendering, Siggraph 2017

Here are some Renderings that Howard made to illustrate his points.

Howard dubbed the

- stacked round clouds Cumulus, which means a heap or pile in Latin.
- The Long flat clouds became Stratus, which means layer or sheet in latin.
- The wispy stretched clouds that exist high in the atmosphere became Cirrus, which means hair or fibre in latin.



The Clouds

Renderings by Luke Howard

[Hamblyn, 2001]

Howard noticed that clouds formed in clusters at different altitudes below 10km. So altitude also became a classifying factor. (I converted these to miles for all of the Yankees like me in the room.)

- While clouds below 1.5km retained their names like Stratus and Cumulus…
- Clouds above 1.5km but below 4km in altitude earned the prefix “Alto” Giving us names like altocumulus and altostratus.
- Clouds above 4km, in the top of the cloud zone earned the prefix Cirro.

Howard was the first to propose, with evidence, that clouds formed from water vapor that had condensed on dust particles in the atmosphere. He referred to this process as ‘Nubification’ The phrase never really caught on..

But for Luke Howard, the rest was history.
His system persists to this day, and the reason is because it squares an ancient circle in a way that was intuitive and accessible to everyone.

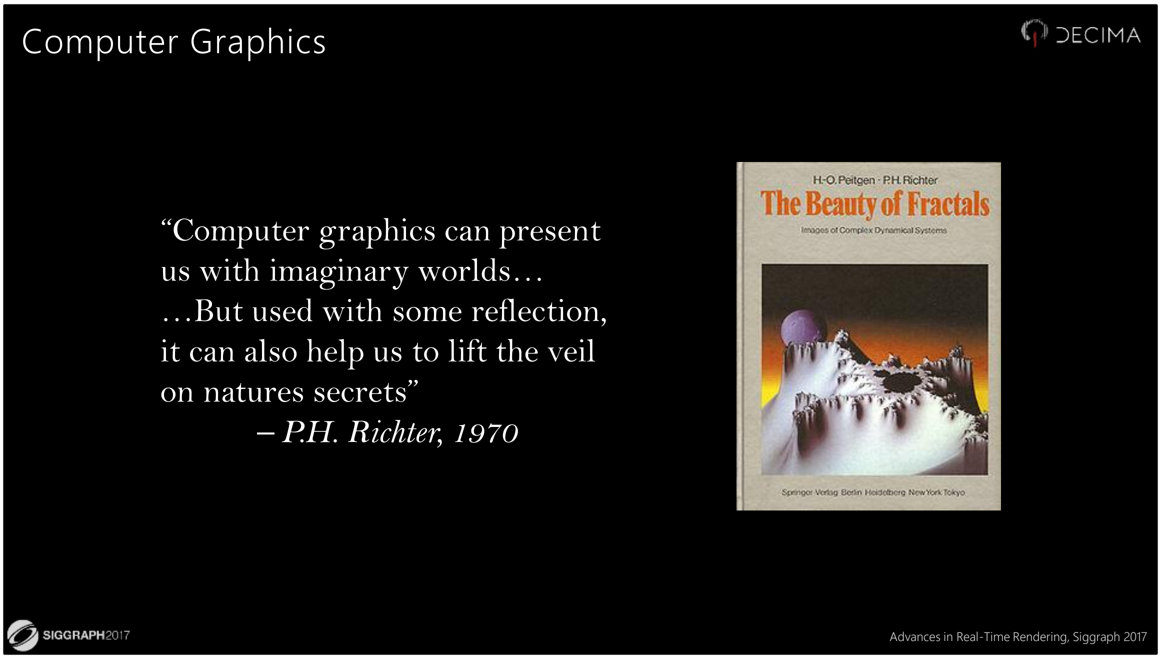
Cloud Rendering

Painting by Claude Joseph Vernet, 1772

Advances in Real-Time Rendering, Siggraph 2017

Scientists were not the only people developing an understanding of clouds. Before the invention of the camera, landscape painters needed to develop a system for remembering what the constantly changing cloudscapes looked like in order to compose their paintings. Maybe they were the first to understand the underlying physics involved as far as it served their purposes of recreation. These techniques were handed down within the confines of the fine art community for centuries.

Within the last half century, the computer evolved into a medium where science and art catalyzed the development of a new discipline called computer graphics.



Computer Graphics

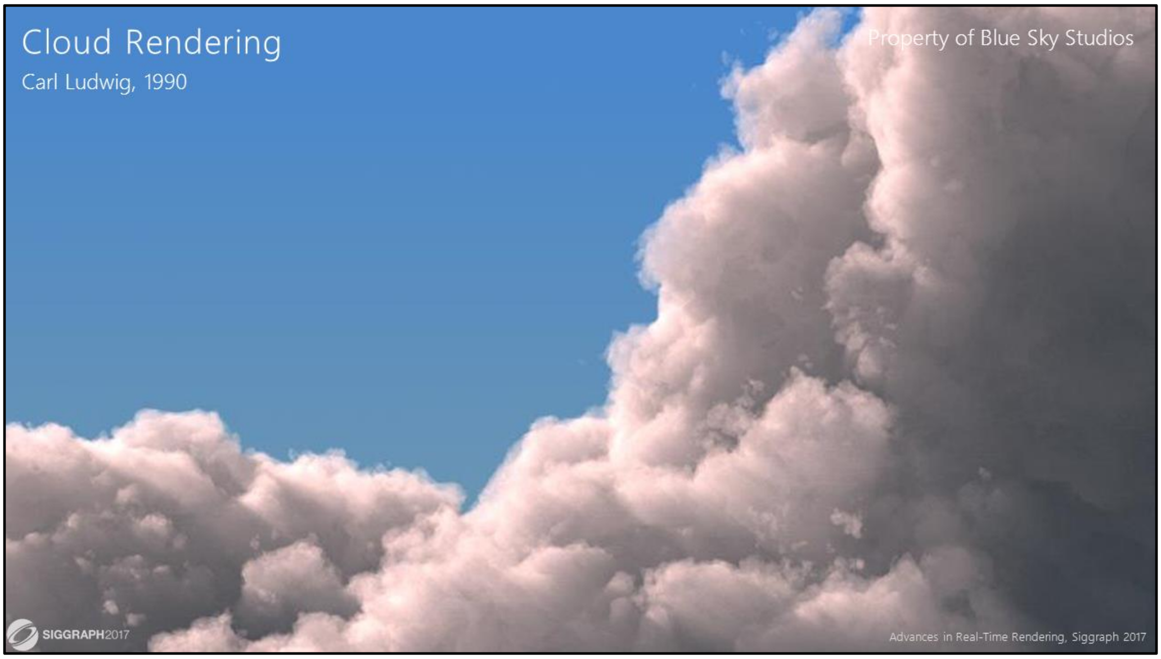
“Computer graphics can present
us with imaginary worlds…
…But used with some reflection,
it can also help us to lift the veil
on natures secrets”
– *P.H. Richter, 1970*

H.-O. Peitgen · P.H. Richter
The Beauty of Fractals
Images of Complex Dynamical Systems
Springer Verlag Berlin Heidelberg New York Tokyo

In his book, the Beauty of Fractals, P.H. Richter said “Computer graphics can present us with imaginary worlds … But used with some reflection, it can also help us to lift the veil on natures secrets”

Well, I think that As artists and developers, this is one of the most exciting frontiers that we get to explore in computer graphics.

Soon enough, early pioneers of computer art and science turned their attention to clouds..



Cloud Rendering

Carl Ludwig, 1990

Property of Blue Sky Studios

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

In 1990, Carl Ludwig, one of the pioneers of ray-tracing at Blue Sky Studios designed a program which rendered clouds using a ray-march and implicit surfaces constructed from noise. I bring this example up instead of the numerous film and vfx examples of cg clouds from this decade because what he did back in the 90's was elegant, physically accurate and convincingly real. This is actually the first time that this image has been shown outside of the circle of people who know Carl, so thanks for letting us use it today for educational reference, Carl.

Before the recent advances in hardware and theory, we didn't have the tools or the language required to render something like this in real-time.

Cloud Rendering

Early Real-Time Volumetric Clouds

- TrueSky [Simul, 2013]
- Reset Engine [Reset, 2012]

Since then, processing power has improved and the pioneers working on

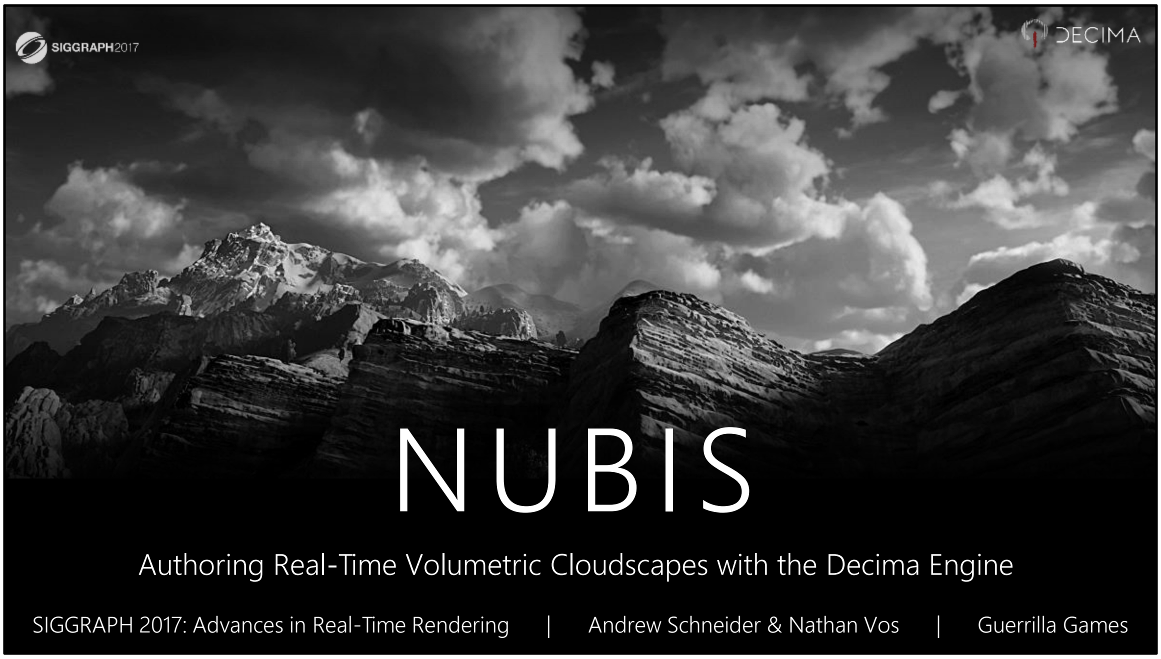
- True Sky and the The Reset Engine started developing methods for rendering volumetric clouds in real-time for use inactual games. These recent advances gave my co-developer Nathan Vos and I and the rest of the team at Guerrilla the courage to try this ourselves.

Cloud Rendering

in-game render, 2015

Advances in Real-Time Rendering, Siggraph 2017

In 2014, we decided to develop our own approach to rendering clouds for use in our game Horizon: Zero Dawn and in our engine, Decima, which is being used by Kojima Productions. In 2015, we shared our prototype as part of this course and offered a high level overview of how we model, light and render our cloudscapes in under 2 milliseconds. This prototype, while successful and able to run on its own using a procedural weather simulation, still lacked the level of reproducibility and control for modeling, animating and lighting that would be required to support a heavily art directed game like Horizon, where the goal was for nature to appear hyper-real in gameplay and cutscenes.

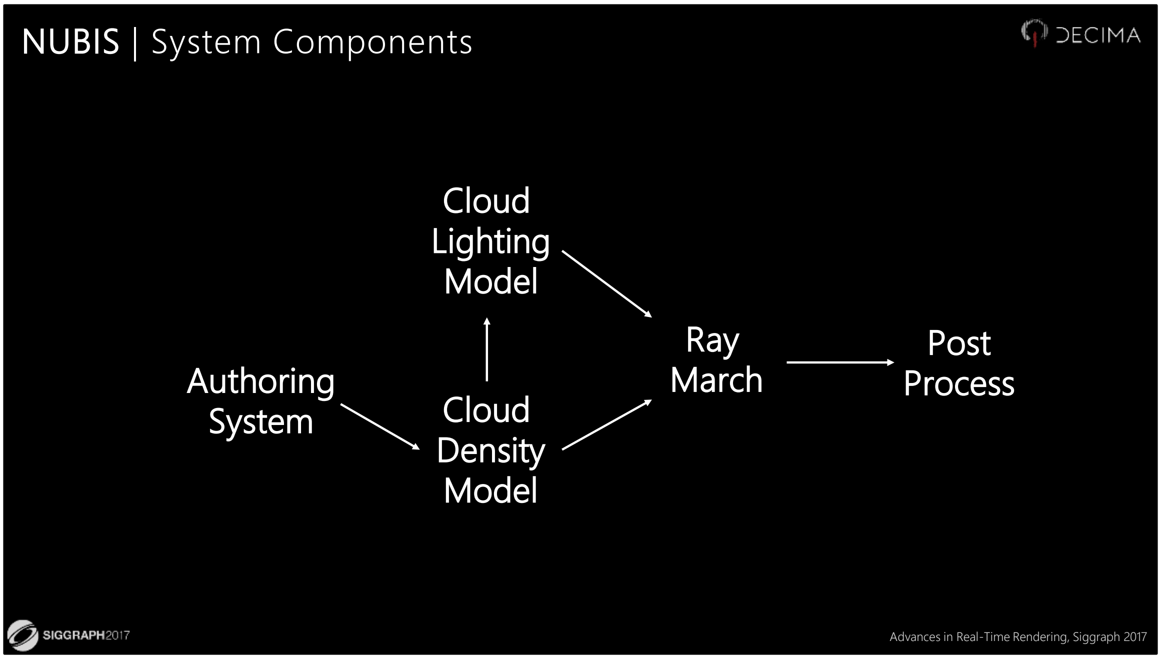


NUBIS

Authoring Real-Time Volumetric Cloudscapes with the Decima Engine

SIGGRAPH 2017: Advances in Real-Time Rendering | Andrew Schneider & Nathan Vos | Guerrilla Games

Our solution to these challenges and to simulating the the important natural phenomena of clouds is what we are going to discuss today.
Nubis is our approach to artistically authoring real-time volumetric cloudscapes for games. In addition to including an authoring component, we improved the performance beyond our 2 ms budget.
Nubis uses no assets, only sets of instructions from our authoring system that evoke behaviors that are defined by probability. This holds true across the entire system.
There are 5 main components: #



NUBIS | System Components

DECIMA

Cloud Lighting Model

Authoring System

Cloud Density Model

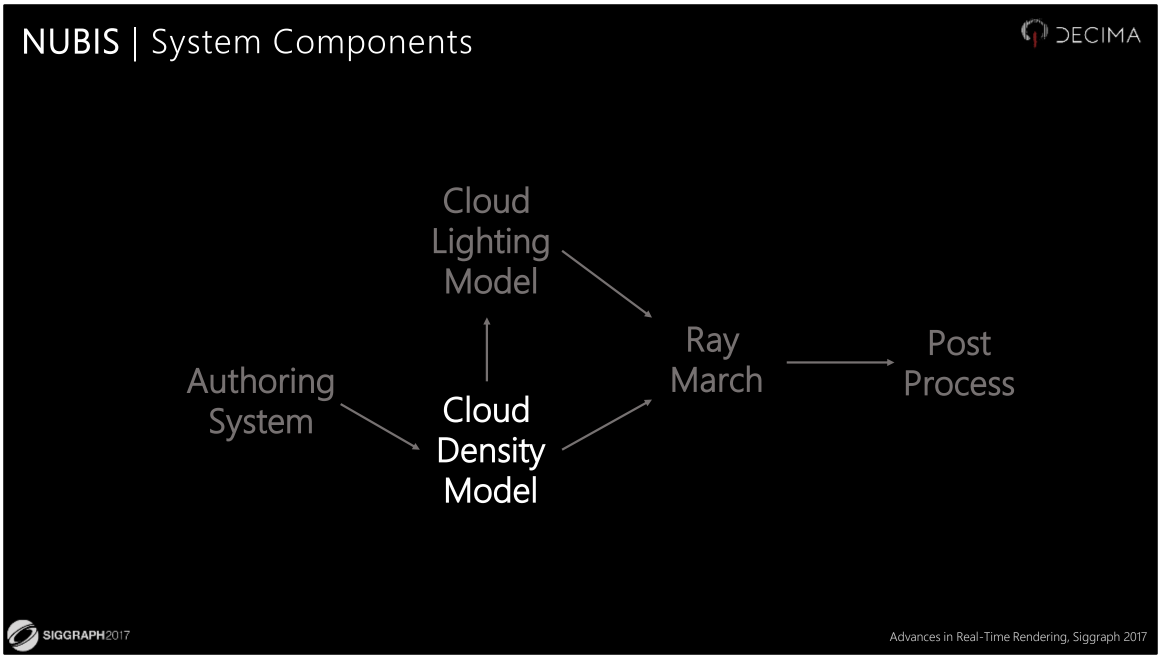
Ray March

Post Process

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

- The Authoring system, which defines where to draw clouds and what type of clouds to draw as well as transitions and detail characteristics;
- Then there is the density model, which generates the physical forms of our clouds including the wispy and billowy shapes as well as the deformations. #
- Next is The Lighting model which simulates the scattering and absorption effects associated with clouds. The lighting and density models are both a part of the
- ray-march operation which is used to render the density and lighting data in slices away from the camera in an optimized way
- Finally there is the post process shader which integrates our cloudscapes into each frame



NUBIS | System Components

DECIMA

Authoring System

Cloud Density Model

Cloud Lighting Model

Ray March

Post Process

SIGGRAPH2017

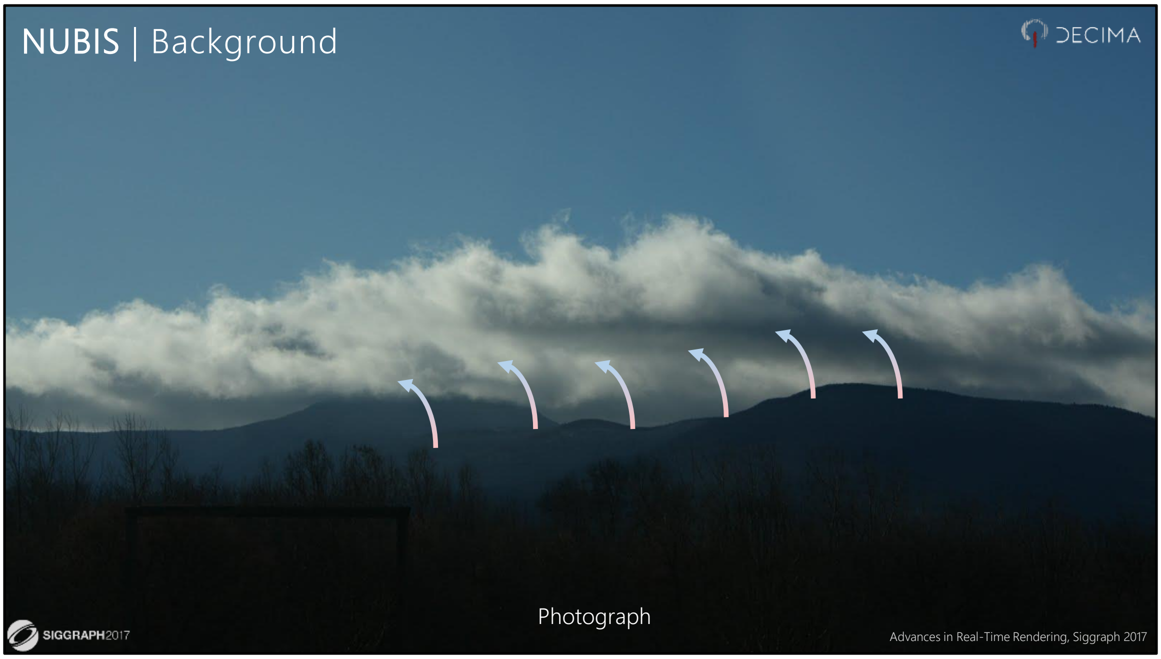
Advances in Real-Time Rendering, Siggraph 2017

It is necessary to understand the fundamentals of how and on what basis we model a single cloud before we can explain how we build entire cloudscapes. So first, we are going to start with the density model. Our cloud modeling approach is inspired by Howards 'Nubification', but rather than jumping right into our implementation, lets first go on a brief thought experiment. To truly understand the physics involved in cloud formation we need to experience it up close.

So, we are going to imagine ourselves as balloons that have been released into the air.

I know it’s early and a lot of us are jetlagged, but lets try.

Our upward journey starts in the morning as the sun begins to heat the earth and forces water vapor to rise into the higher colder layers of the atmosphere.



NUBIS | Background

Photograph

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

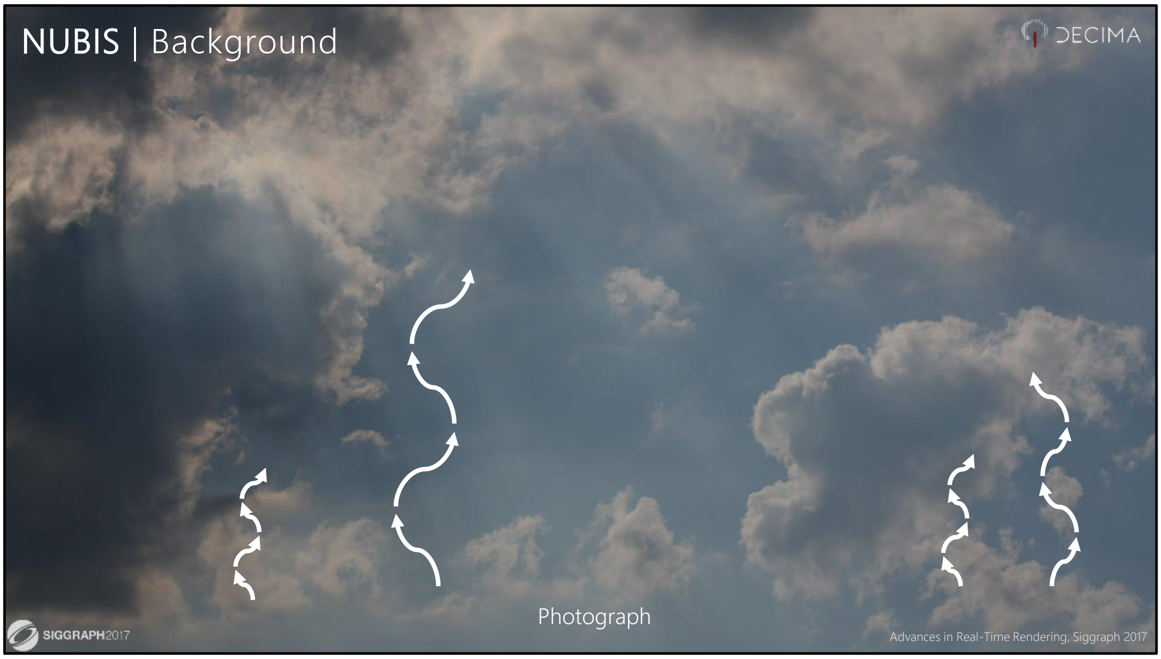
In the distance this vapor starts to slowly seep upward from the earth and enter a cool layer of morning air.

- This slow emission of vapor into cool air produces the sheet-like shapes that Howard referred to as stratus clouds.

NUBIS | Background

We notice that elsewhere, large pulses of warm vapor

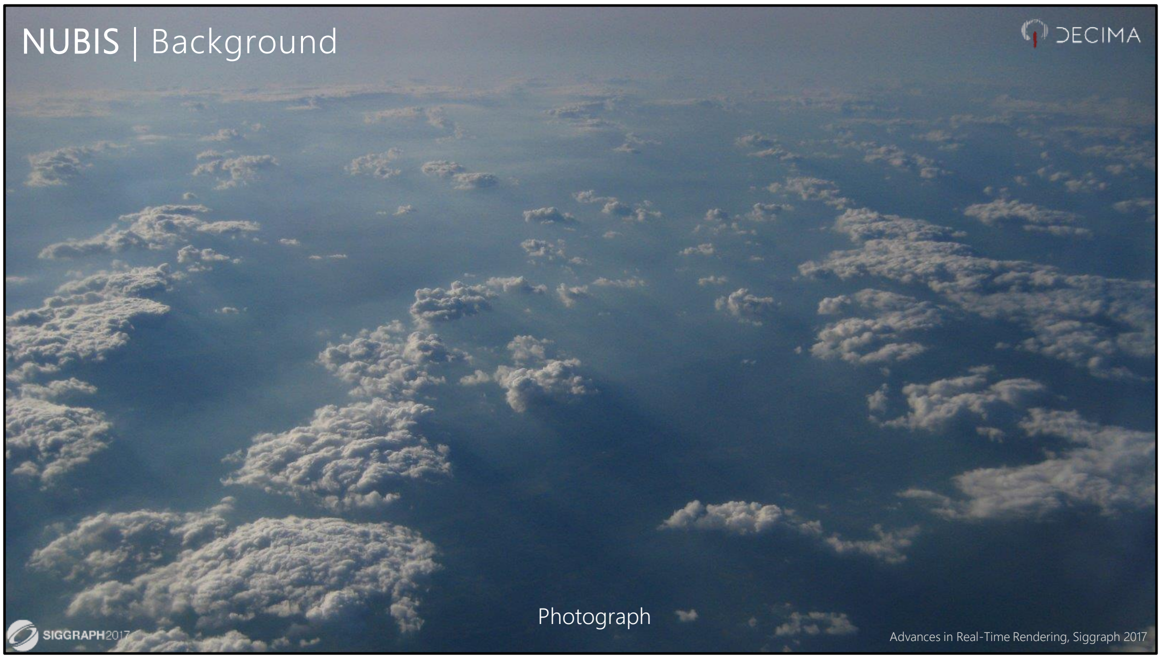
- appear to create the stacked, round shapes that Howard describes as Cumulus clouds. They tower around us as a warm updraft of air lifts us into the atmosphere. This updraft is part of what is known as a convection current.



We feel a force pushing us to the side and then dropping us. We have entered a pocket instability in the convection current called turbulence.

- This disturbing force shreds clouds and tears them apart all around us, which produces wispy and curling shapes.

We see that the light from the sun creates shafts of light in the hazy pockets of water vapor all around us.



NUBIS | Background

DECIMA

Photograph

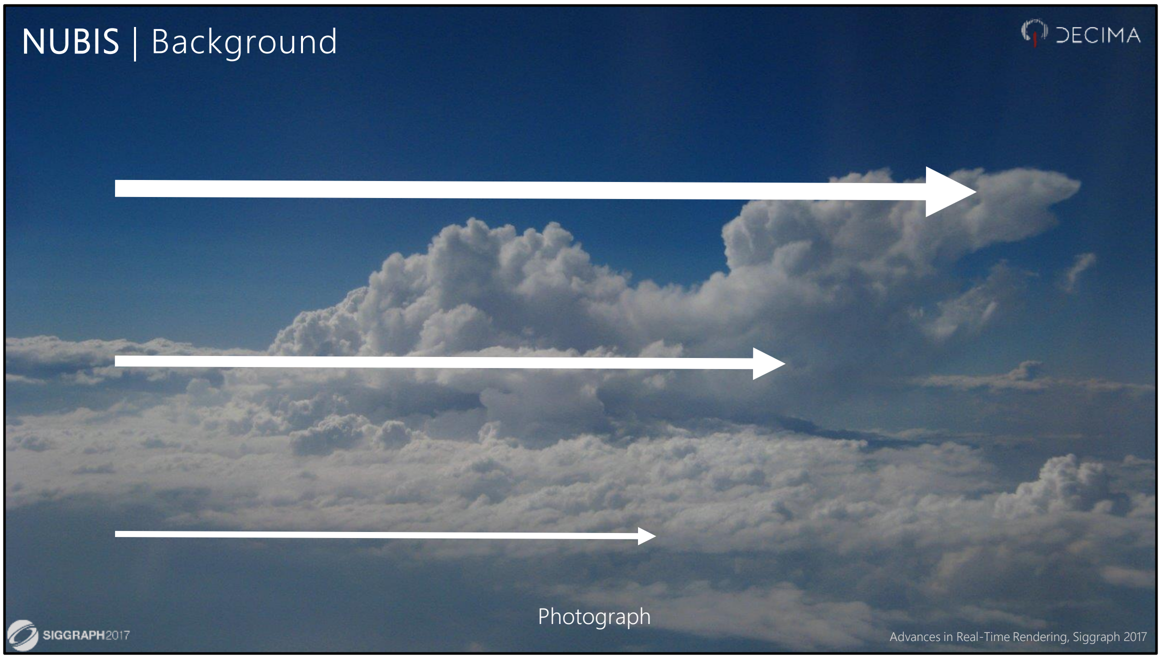
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

As we rise higher, past the point where this vapor cools and condenses on dust and salt particles, we see a clear line of demarcation where local pockets of condensed vapor become dense enough that light rays can’t avoid hitting the vapor and scatter instead of continuing to the ground.

We also notice that some cumulus clouds have begin to overlap and cluster together into sheets of their own. This is what Howard referred to as stratocumulus clouds because they are both a sheet and a stack of round shapes.

Also, The temperature is dropping by 6.5 degrees Celsius for every kilometer that we ascend. (3) This shouldn’t be hard to imagine in this ballroom.



NUBIS | Background

Photograph

Advances in Real-Time Rendering, Siggraph 2017

The wind up here is quite strong

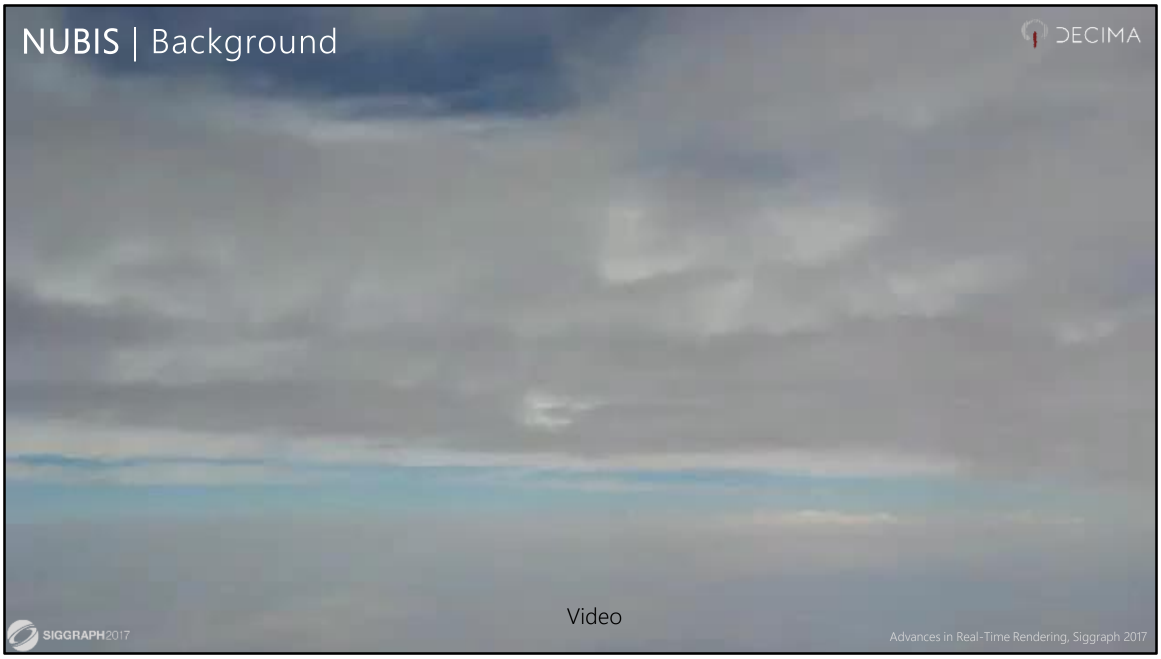
- and it begins to push us and other clouds to the side as we rise.

NUBIS | Background

Video

Advances in Real-Time Rendering, Siggraph 2017

A large billowing cumulus congestus cloud passes by us on our journey. There is a peculiar effect on the clouds that we did not see from the ground. The edges appear dark. Curious.



NUBIS | Background

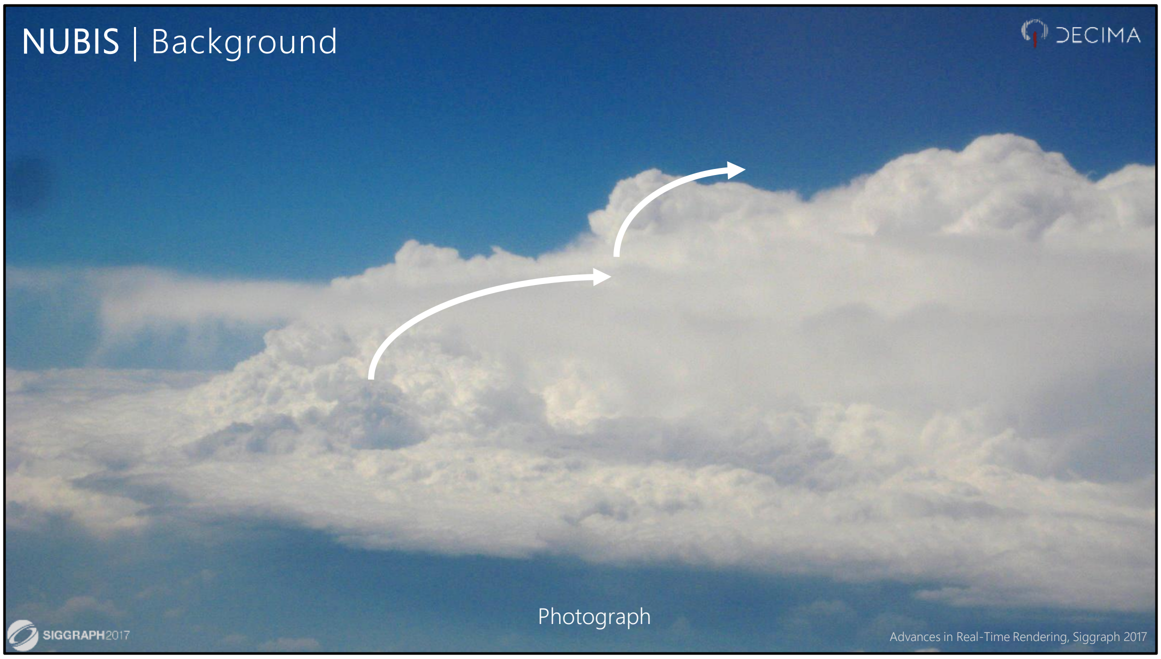
DECIMA

Video

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Suddenly the temperature drops sharply. We look back to the horizon and see that we are entering a new layer of clouds, the Alto layer. We are passing through a sheet that Howard described as the Altostratus clouds.



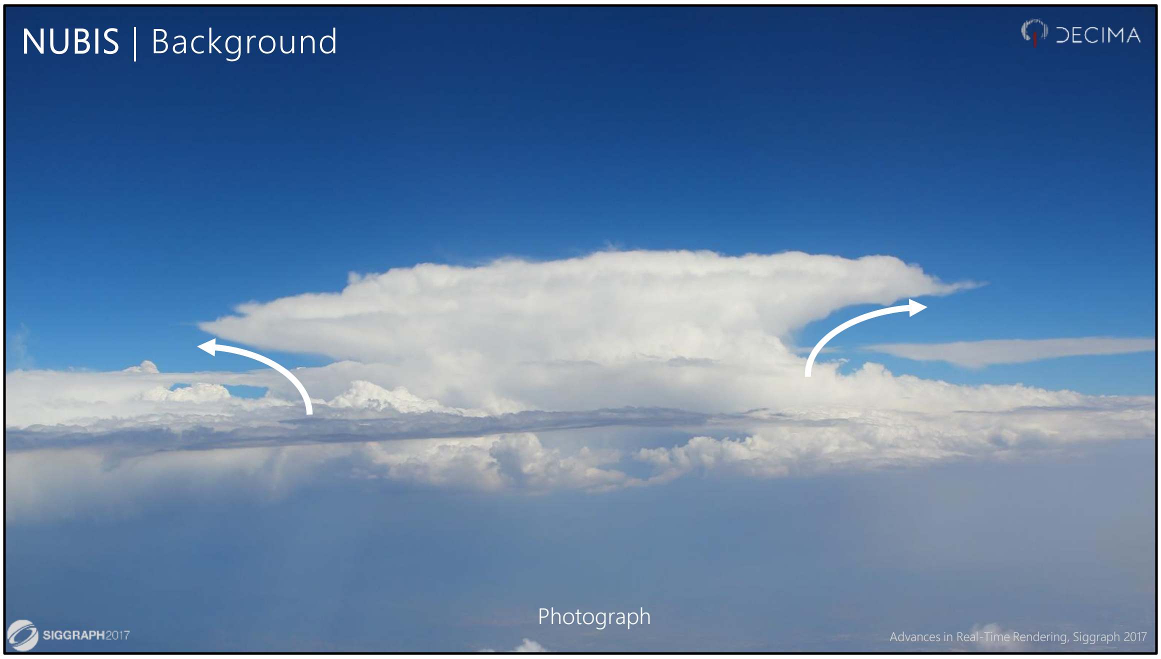
NUBIS | Background

Photograph

Advances in Real-Time Rendering, Siggraph 2017

Behind us we see that a cumulus cloud has

- punched through the alto layer.
- It is becoming the grandest of the cloud forms, the Cumulonimbus cloud.



NUBIS | Background

DECIMA

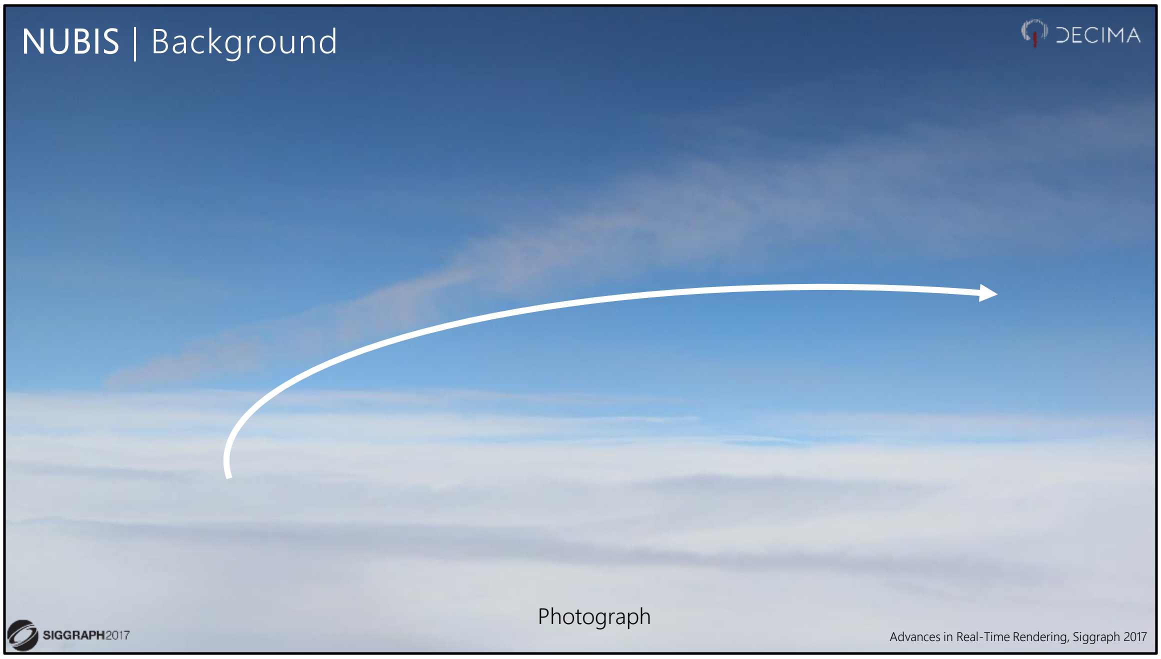
Photograph

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

With the alto clouds now below us we see the top of the cumulus cloud continues to rise until it hits another colder layer of air in what is called the cirro layer.

- It spreads out in the classic anvil shape.



NUBIS | Background

DECIMA

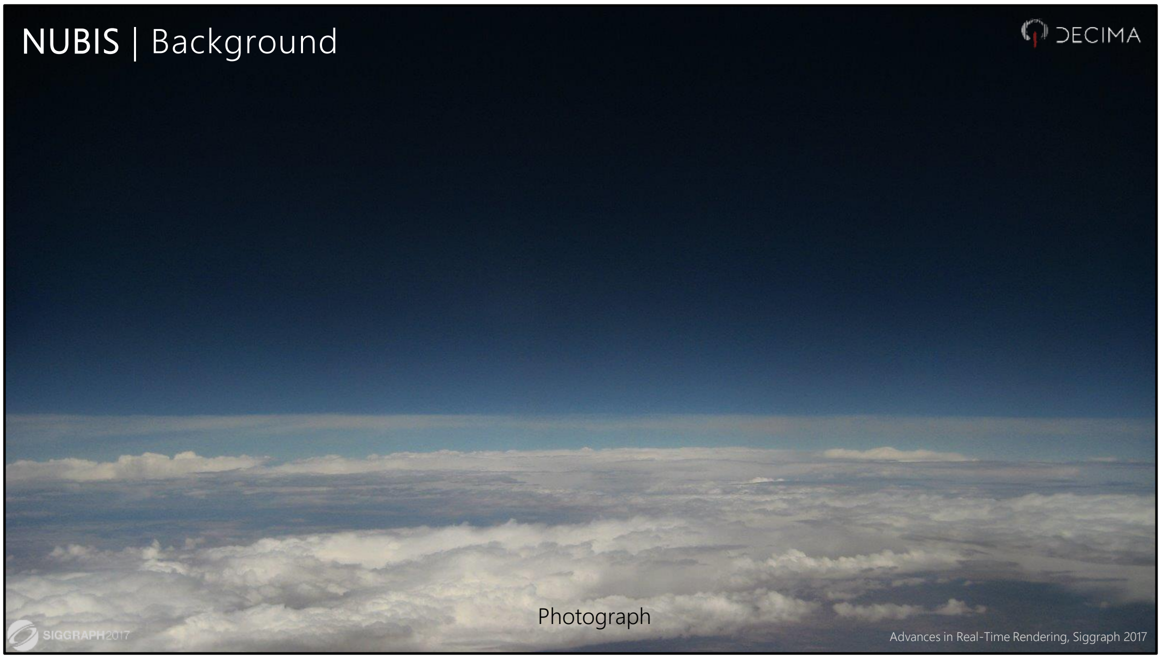
Photograph

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Behind us we see the stretched, fibrous shapes of the cirrus clouds. Up here it is below freezing and all water has crystalized into ice.

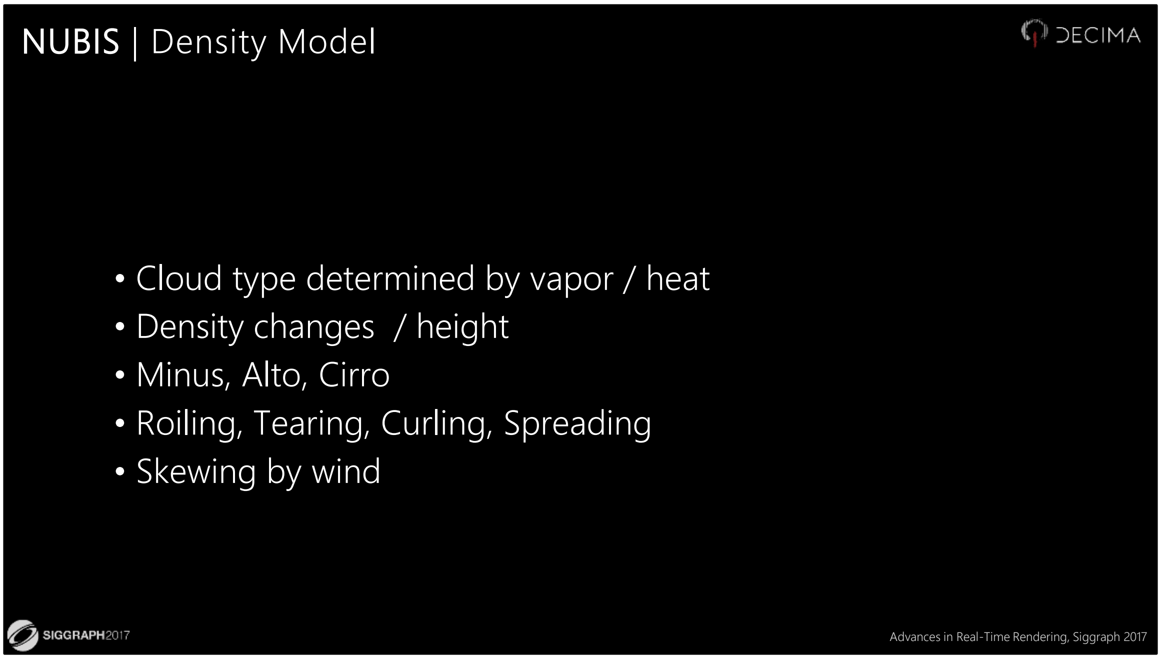
- which has compressed the clouds into sheets of wispy striations.



NUBIS | Background

Photograph

Eventually the low pressure stresses our elastic bodies so much that we pop……. And fall back into our chairs in this convention center ballroom.



NUBIS | Density Model

DECIMA

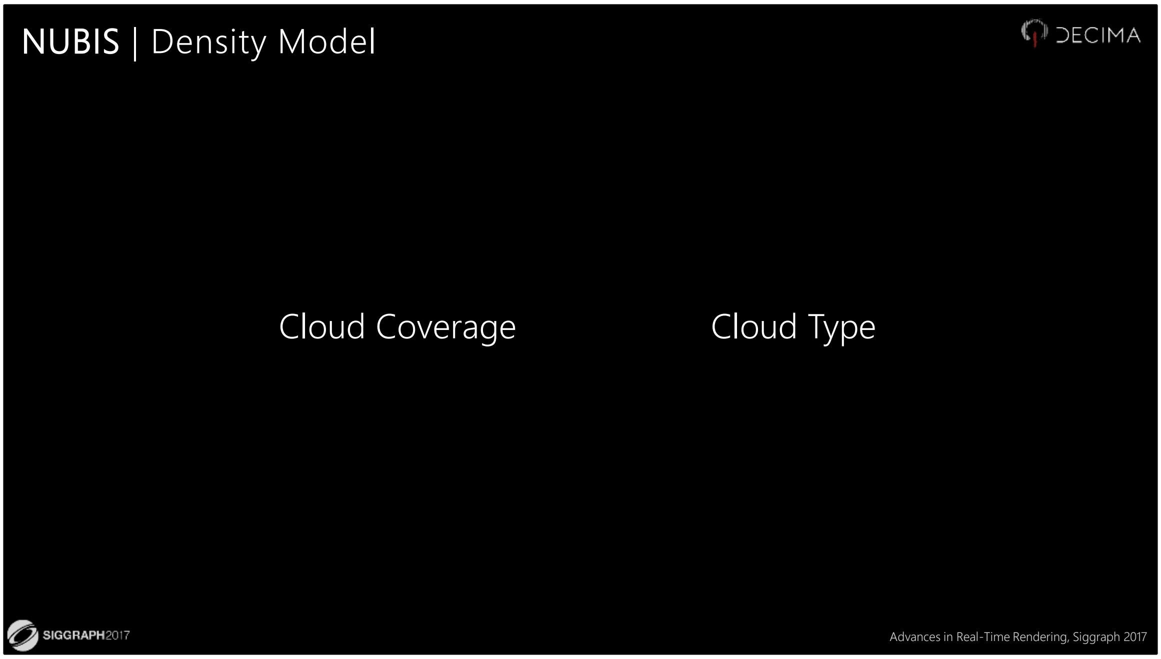
- Cloud type determined by vapor / heat
- Density changes / height
- Minus, Alto, Cirro
- Roiling, Tearing, Curling, Spreading
- Skewing by wind

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Ok, Lets catalog what we observed on our journey.

- We saw how vapor and heat rise to form clouds
- And that over this vertical journey, temperature decreases causing the density of water molecules to increase
- We also noticed 3 distinct regions of clouds: The the lower or minus clouds, The alto clouds and the cirro clouds.
- Additionally, we saw how instability can shape clouds by and producing tearing, spreading, and roiling shapes.
- We saw that Wind also skews clouds as they rise.



NUBIS | Density Model

Cloud Coverage

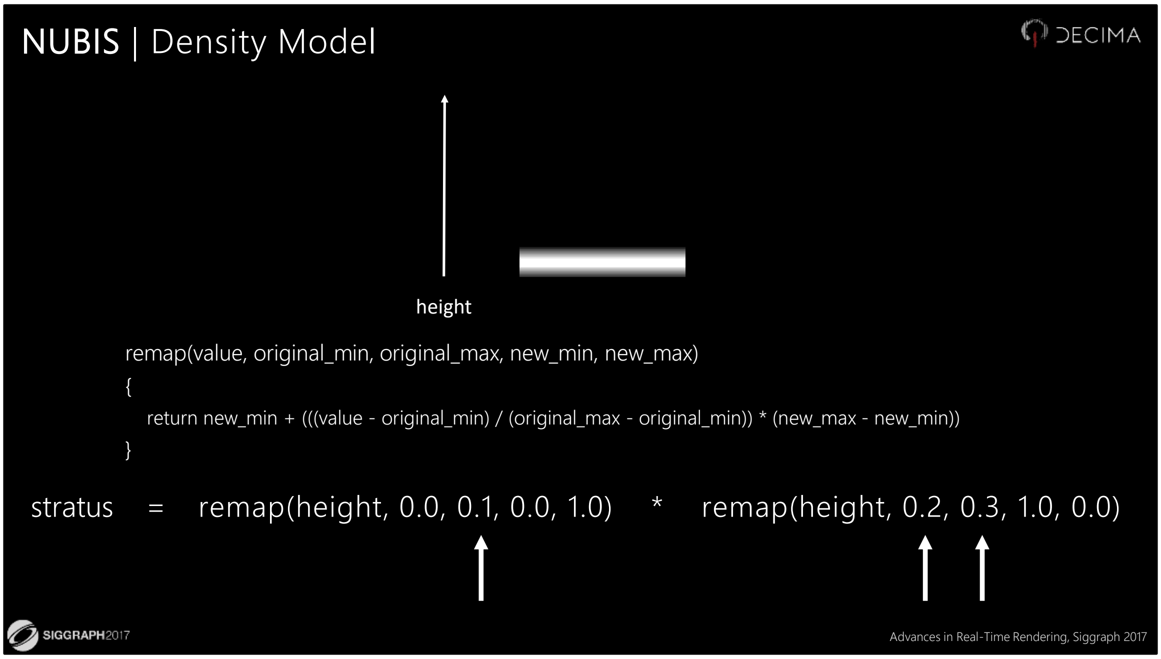
Cloud Type

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Now, lets form some abstractions using these observations.
Real-time is all about compression, so If we had to reduce the experience we just had down to two pieces of information we could say

- that we saw variations in cloud coverage and in cloud type over space. I mentioned early on that Nubis is comprised of a set of predefined behaviors. Most of the behaviors in the density model are dictated by these variables.



NUBIS | Density Model

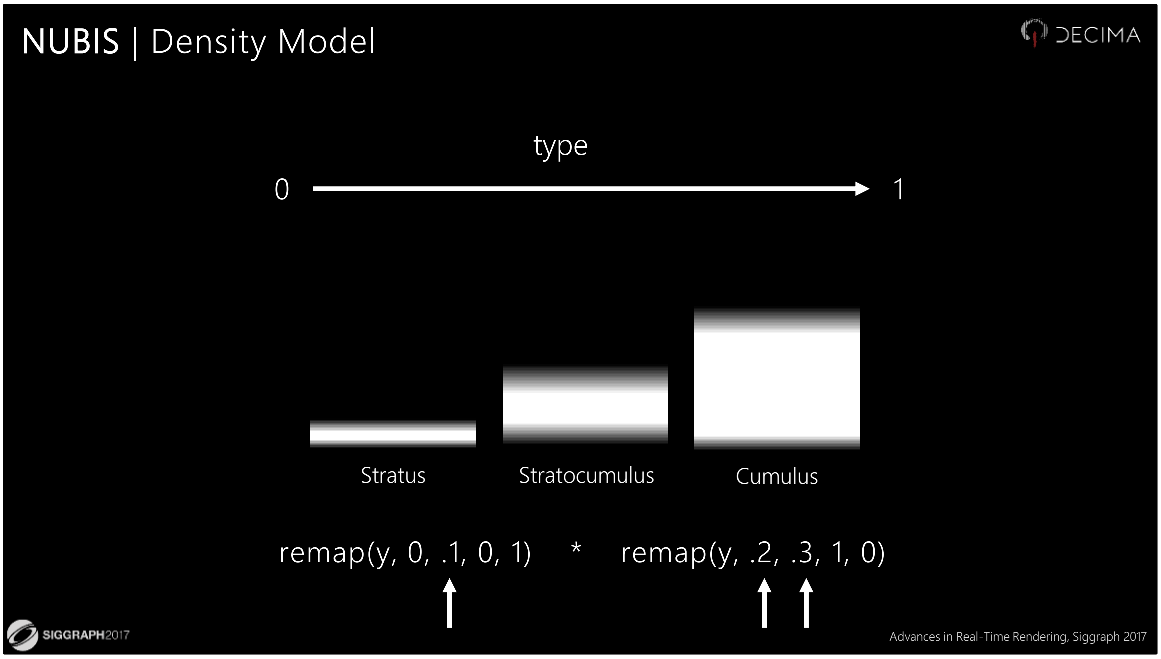
```
remap(value, original_min, original_max, new_min, new_max)
{
    return new_min + (((value - original_min) / (original_max - original_min)) * (new_max - new_min))
}
```

```
stratus   =   remap(height, 0.0, 0.1, 0.0, 1.0)   *   remap(height, 0.2, 0.3, 1.0, 0.0)
```

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

- If we were to describe the probability of change in density over height for a cloud we might envision a gradient like this
- We can implement this mathematically using a remapping function. For the non-coders in the room this is like the setRange in Maya or the Fit in Houdini.
- We combine two of these gradients to create an area of high density probability as in the gradient above.
- If we wish to describe all of the stages of a cloud in the minus layer, we can simply adjust the in and out points of our remap function.



NUBIS | Density Model

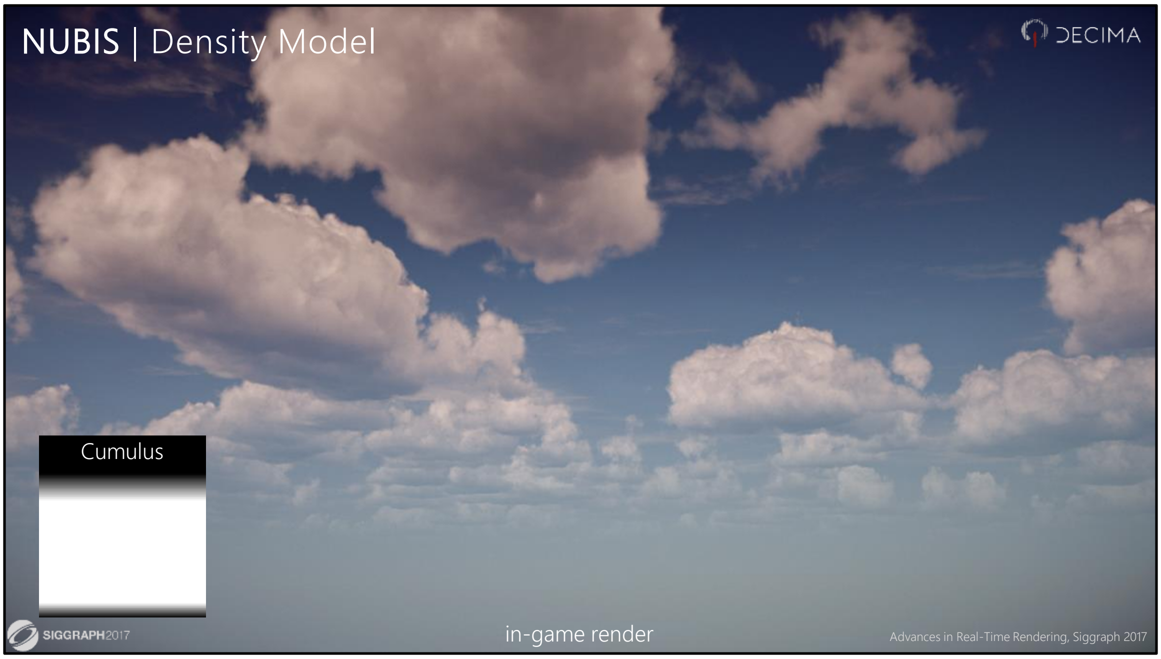
remap(y, 0, .1, 0, 1) * remap(y, .2, .3, 1, 0)

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

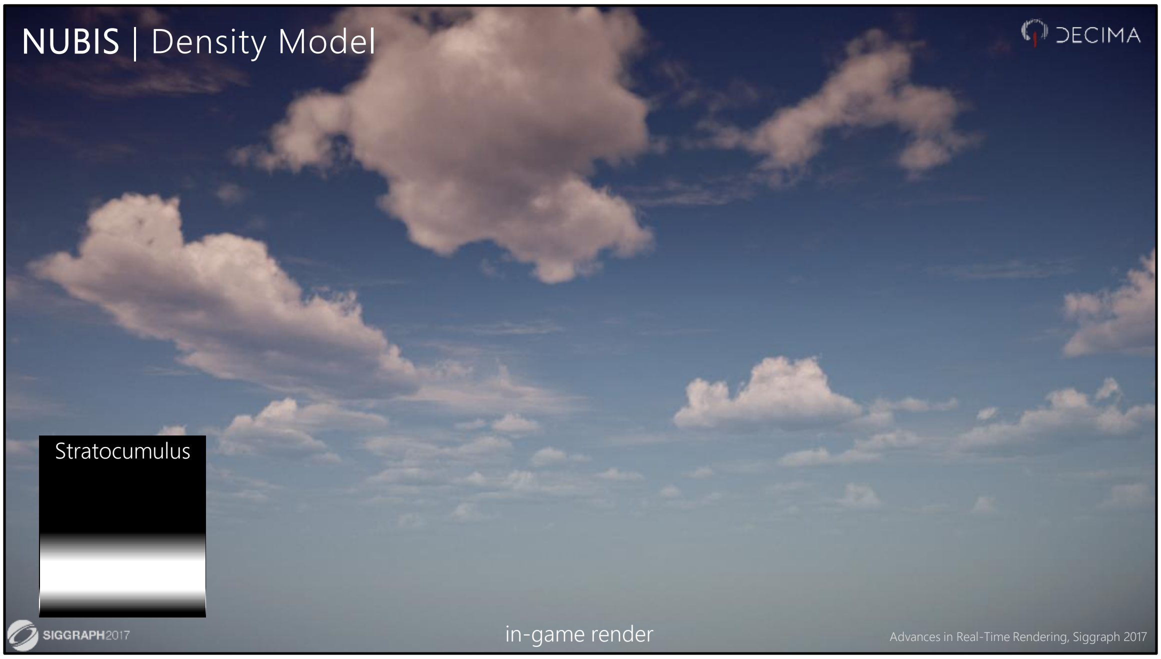
- Since we know that cloud type is determined by rate of emission and temperature, and that each cloud type has a different height range
- we can order a set of these adjustments according to cloud type in order to represent the height probabilities for each type of cloud.
- Type, can then serve as the value which modifies the in and out points of our remap functions.

Lets look at this in practice.



NUBIS | Density Model

This is our cumulus Gradient



NUBIS | Density Model

DECIMA

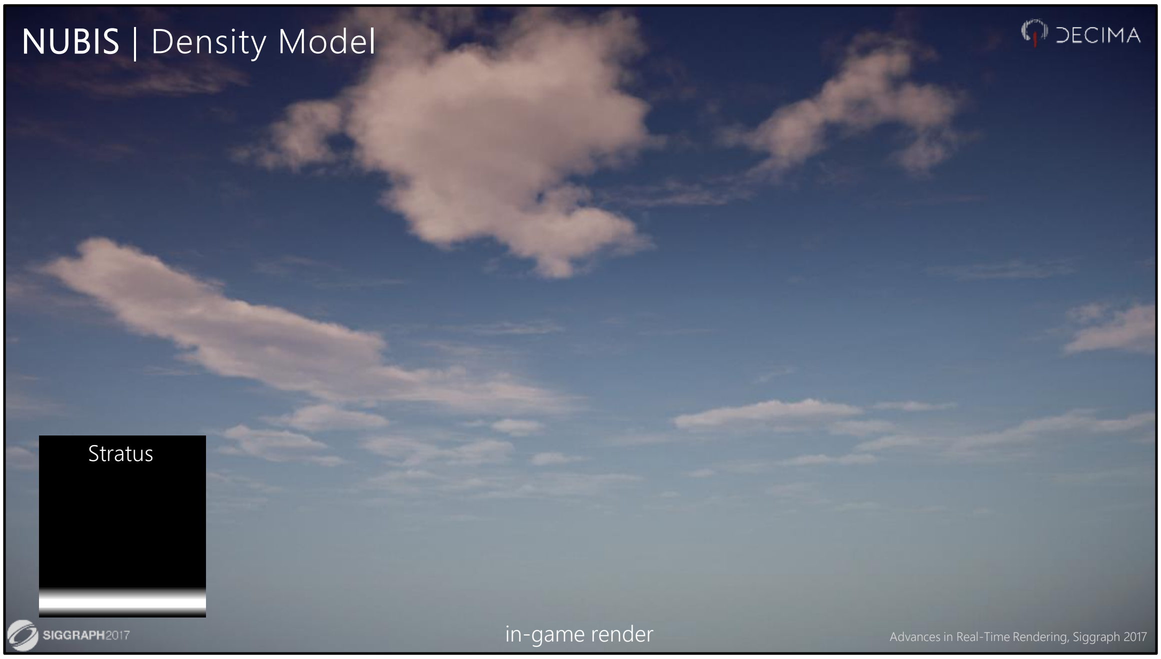
Stratocumulus

in-game render

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Our Stratocumulus gradient



NUBIS | Density Model

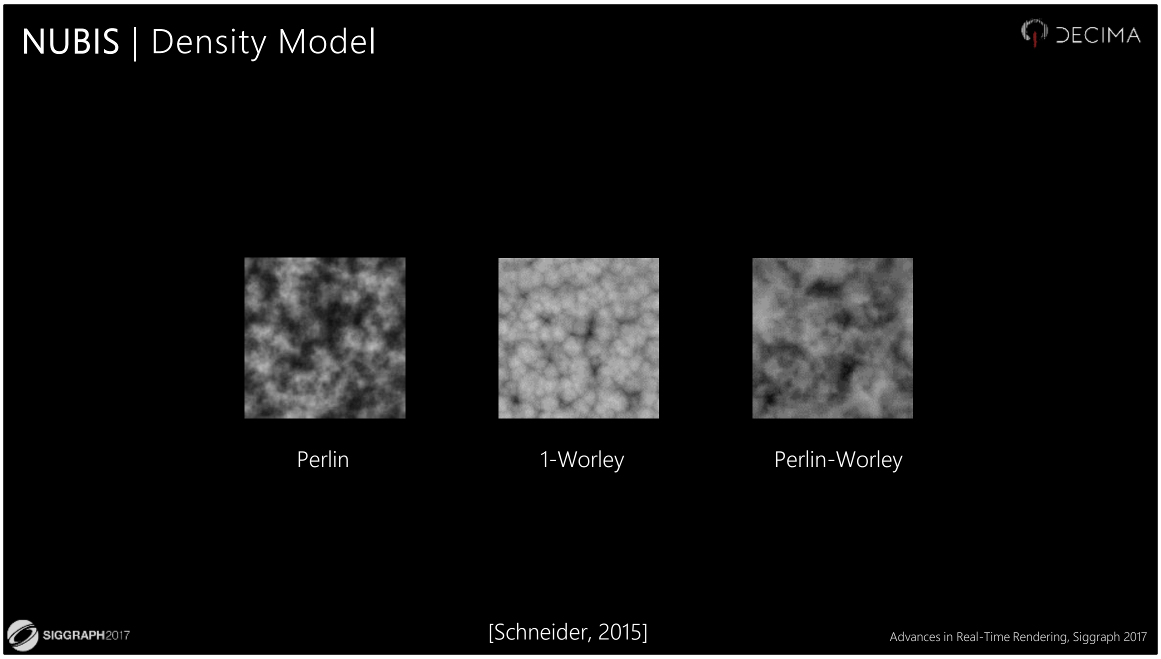
DECIMA

SIGGRAPH2017

in-game render

Advances in Real-Time Rendering, Siggraph 2017

And our stratus gradient.



NUBIS | Density Model

DECIMA

Perlin

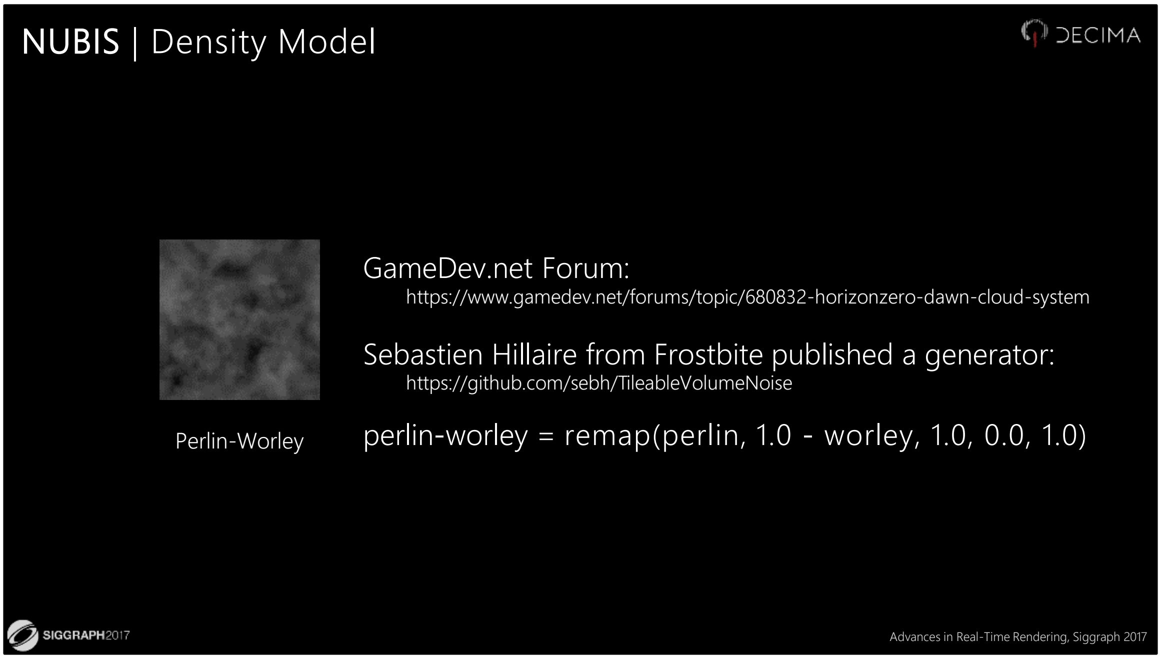
1-Worley

Perlin-Worley

[Schneider, 2015]

SIGGRAPH2017

If we were to describe the curling and roiling shapes over 3 dimensional space we might use 3d textures like Perlin noise, inverted Worley noise or, as we proposed in 2015, a combination of both depending on cloud type or height.
Our Perlin-Worley composite noise has been the subject of discussion in the community.



NUBIS | Density Model

Perlin-Worley

GameDev.net Forum:
https://www.gamedev.net/forums/topic/680832-horizonzero-dawn-cloud-system

Sebastien Hillaire from Frostbite published a generator:
https://github.com/sebh/TileableVolumeNoise

perlin-worley = remap(perlin, 1.0 - worley, 1.0, 0.0, 1.0)

SIGGRAPH2017

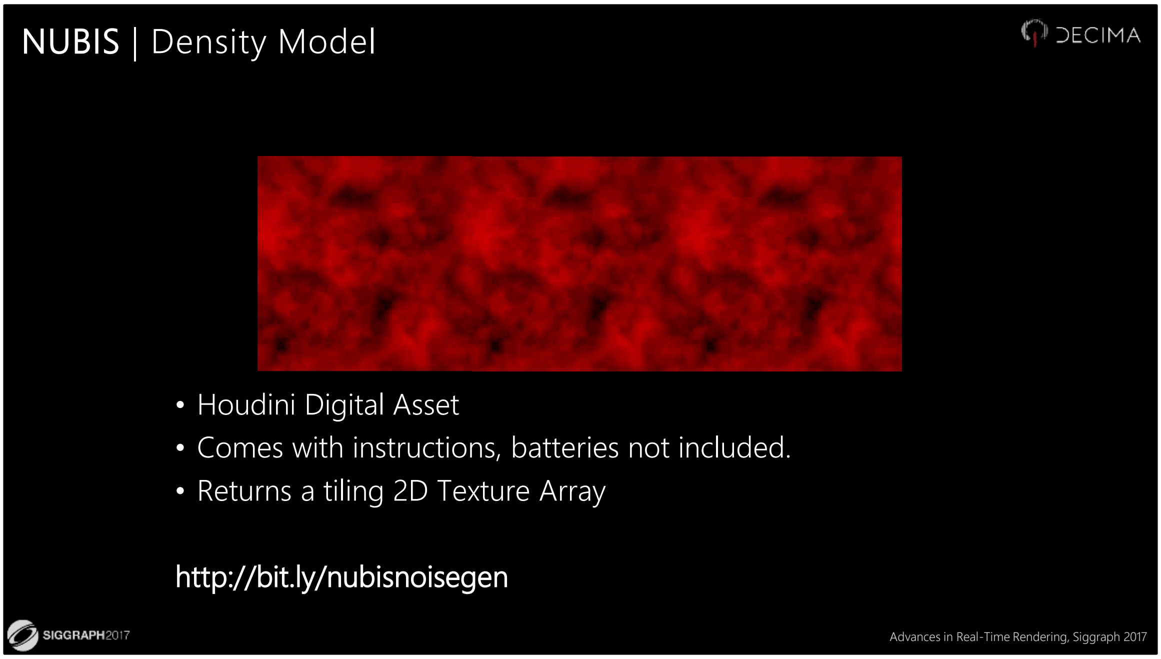
Advances in Real-Time Rendering, Siggraph 2017

- Some developers in the gamedev.net forum are doing great work with it, you should really check it out.

- Sebastien Hillaire from frostbite also put out some code last year to generate this noise. I spoke to him about this and in the near future, you can expect a contribution from us to this codebase.

- Our approach, In principle, was to subtract the web like shapes of Worley noise from the low density regions of perlin noise in order to introduce round shapes there.

But, Rather than hashing out the entire procedure for this here (sorry for the bad pun) I’m going to provide something that can get you started right away…



NUBIS | Density Model

- Houdini Digital Asset
- Comes with instructions, batteries not included.
- Returns a tiling 2D Texture Array

http://bit.ly/nubisnoisegen

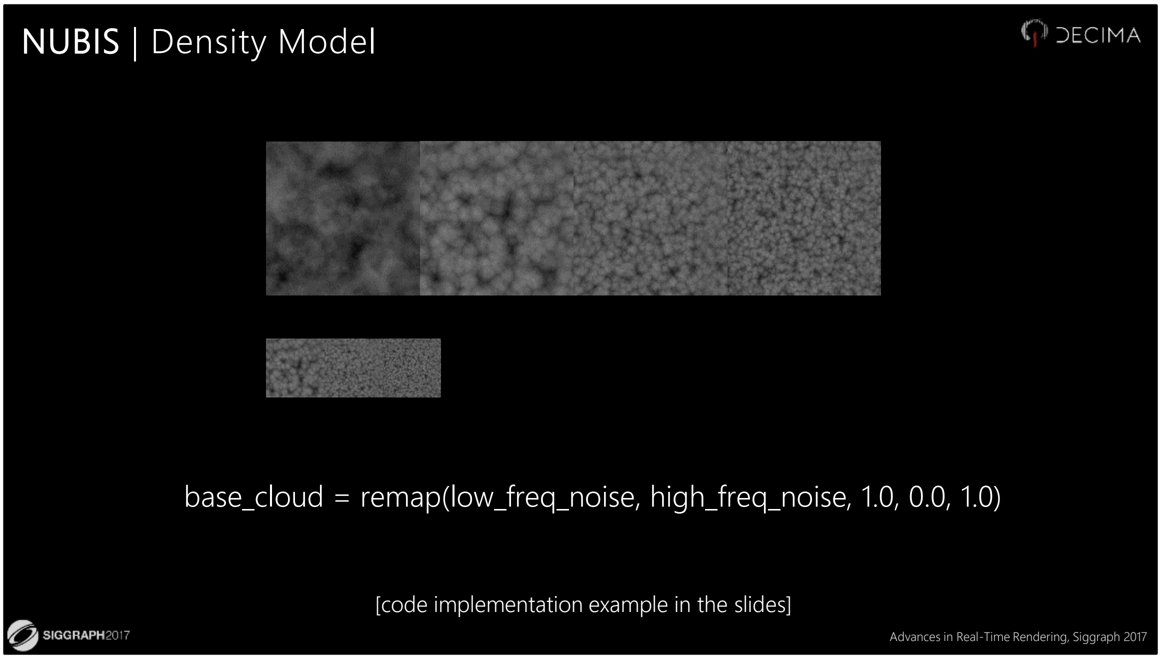
Advances in Real-Time Rendering, Siggraph 2017

We are going to give you the noise generator we used.

- It is a Houdini Digital asset That operates in the cops context.
- There are usage instructions in the download
- It produces a tiling array of 3d texture slices which you can convert into a 3d texture using your engine of choice.

You are free to use this but please share your results with the community, especially when you inevitably improve it! The slides will be online today so you can also grab this link then.

[shaderbits.com blog]



NUBIS | Density Model

base_cloud = remap(low_freq_noise, high_freq_noise, 1.0, 0.0, 1.0)

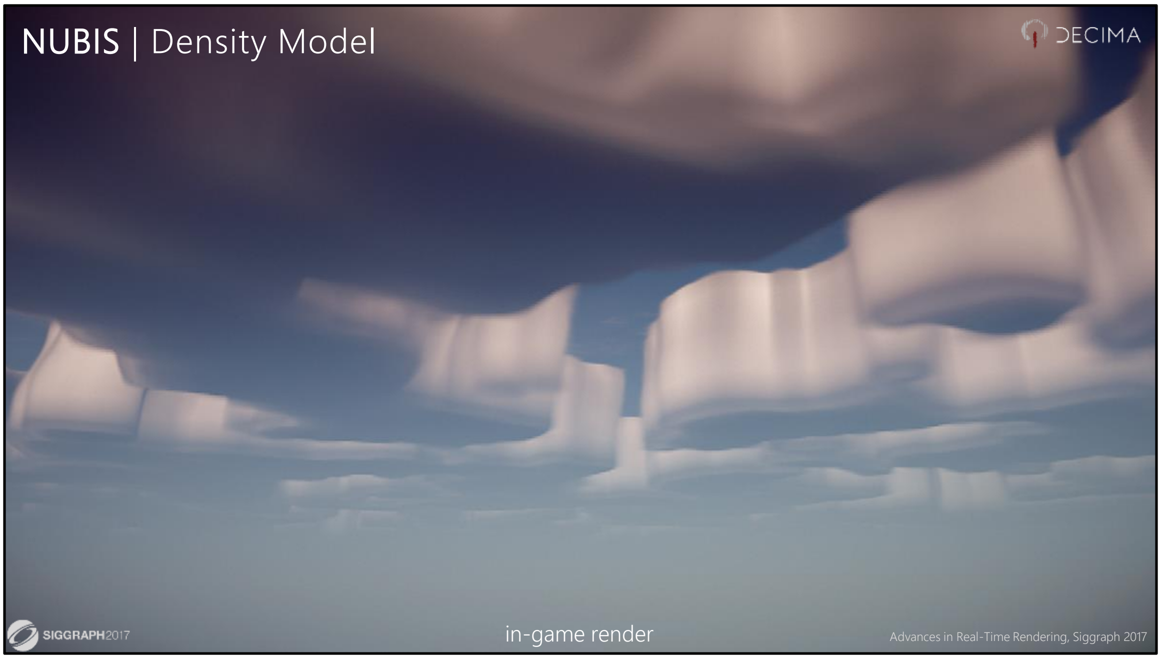
[code implementation example in the slides]

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

As detailed in in the 2015 course, we use a set of

- low frequency Perlin Worley and Worley noises to form the basis of the potential cloud surface and a set of
- high frequency Worley noises to add detail by eroding the base cloud shape with
- a remapping function. We decided to use remapping functions in our noise fBms because of a really useful behavior: as opposed to multiplying the noises together, Remapping prevents a loss of too much density at the core of the base cloud shape.



NUBIS | Density Model

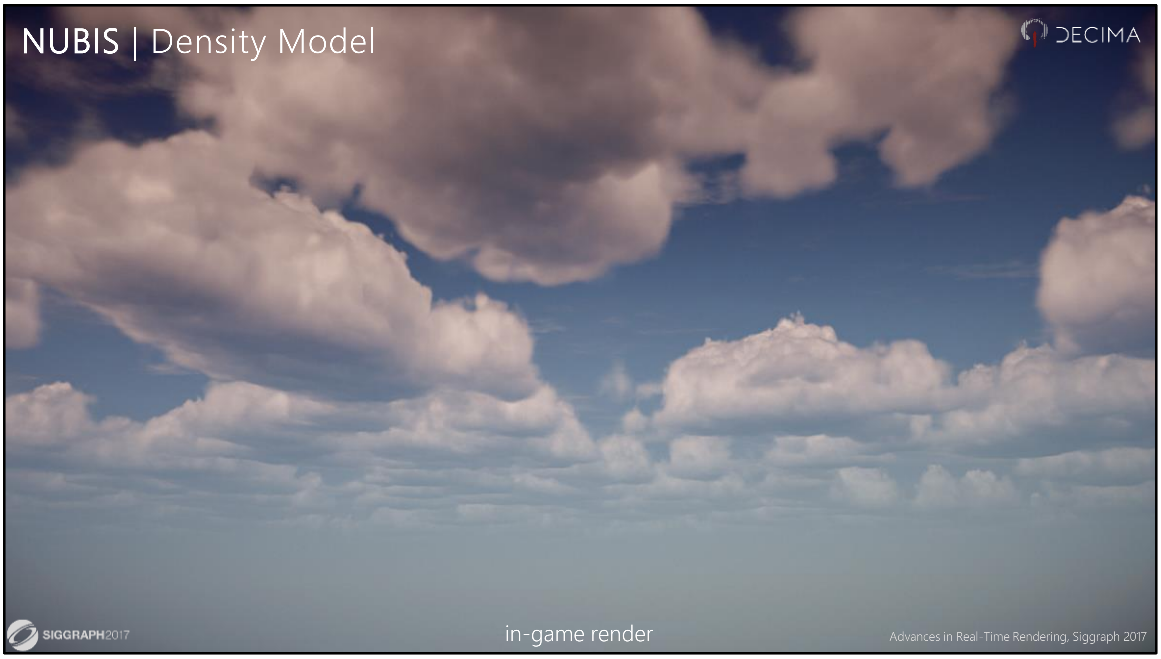
DECIMA

SIGGRAPH2017

in-game render

Advances in Real-Time Rendering, Siggraph 2017

Here's what our clouds look like WITHOUT using noise as a base probability for density.

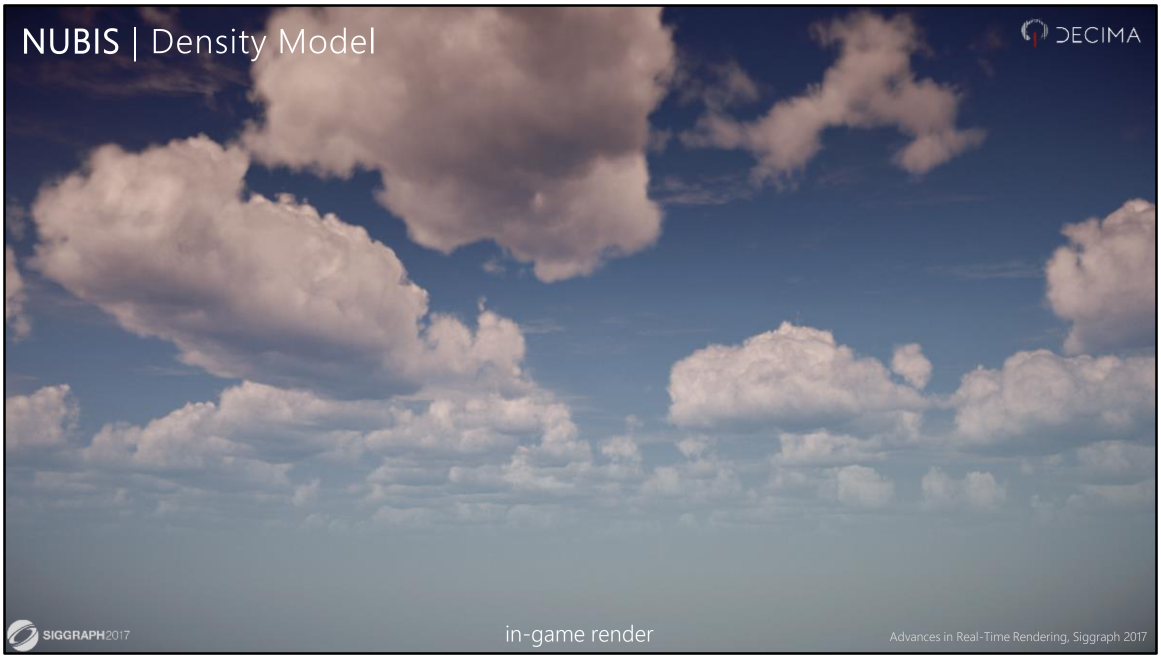


NUBIS | Density Model

in-game render

Advances in Real-Time Rendering, Siggraph 2017

Our Low frequency noises really form the foundation of our clouds, as you can see.



NUBIS | Density Model

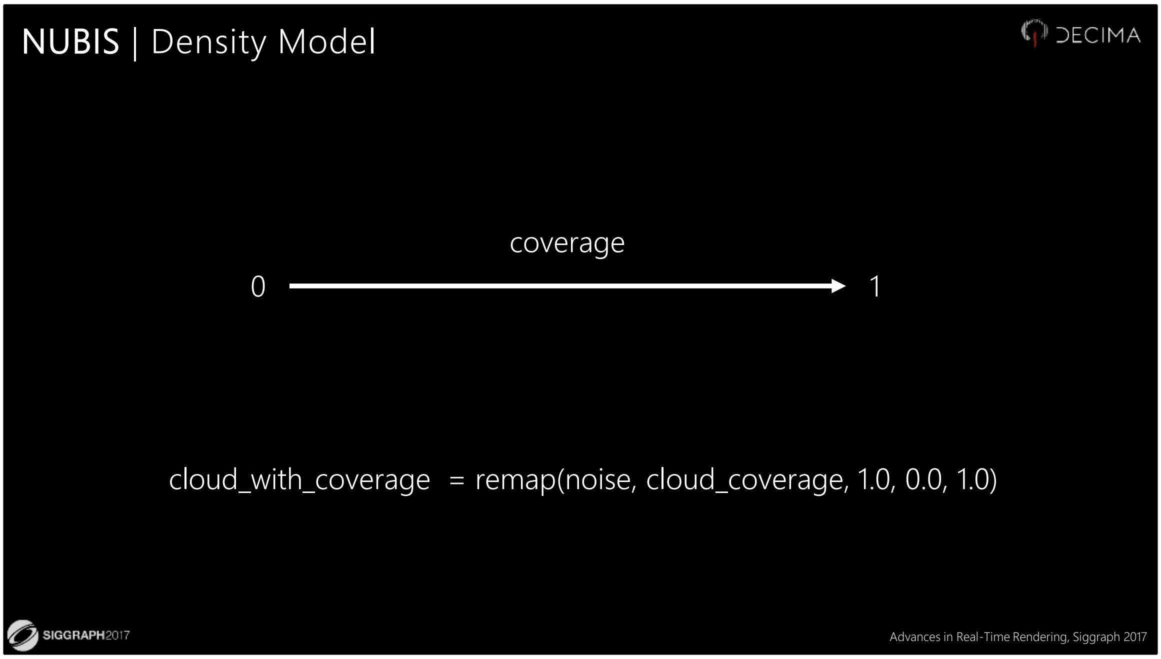
in-game render

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

And the high frequency noises add detail without taking anything away at the center of the cloud.

Its kind of like carving out a block of clay, except that all of the details are already stored in the clay and you are just revealing them as you carve. But in that analogy what dictates how deep to carve?



NUBIS | Density Model

coverage
0
1

cloud_with_coverage = remap(noise, cloud_coverage, 1.0, 0.0, 1.0)

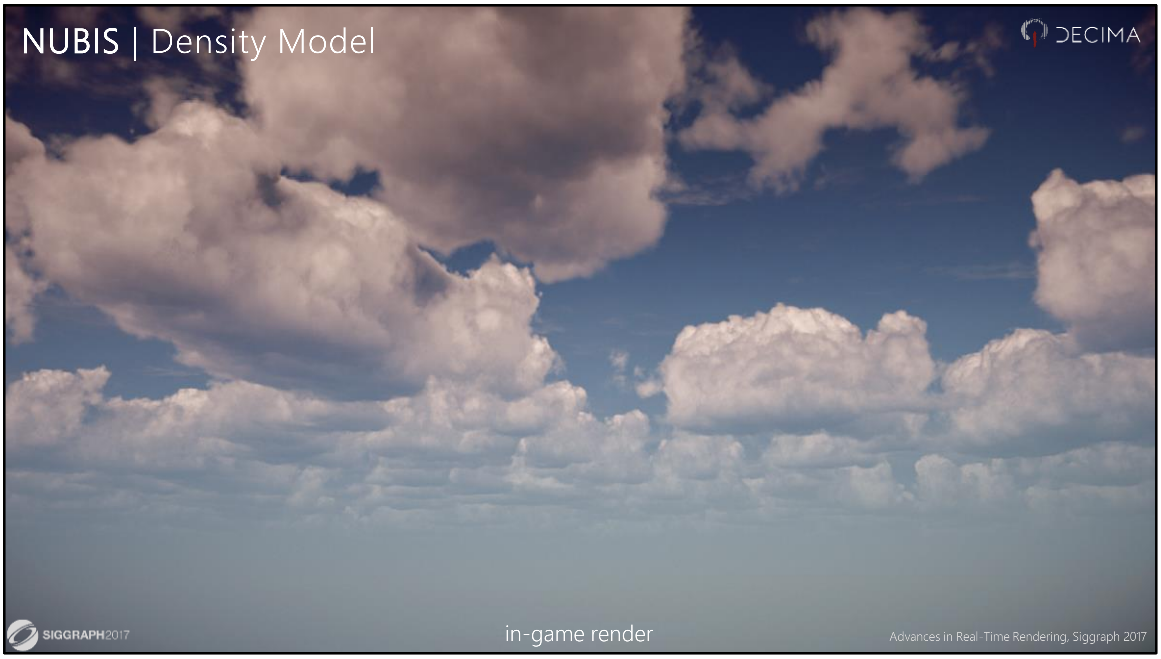
If we think back to our thought experiment we also recall that there were large scale variations in emission of moisture which created individual clouds.

- We can express this by defining a cloud by its coverage of the sky at a given point.

Also, because we are using our noise as a base probability for cloud density in a sample,

- we can apply the coverage value as an erosion in another remapping function. This allows the cloud appear to expand or contract over a gradient of coverage values in space.

Here is an example.

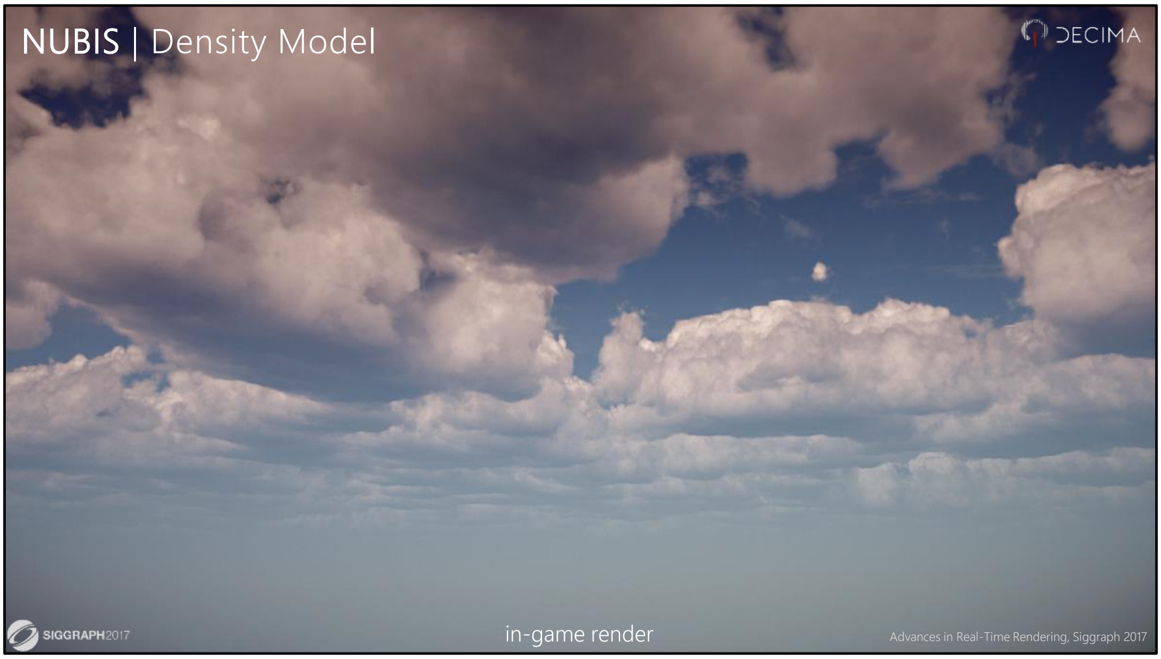


NUBIS | Density Model

in-game render

Advances in Real-Time Rendering, Siggraph 2017

As the coverage value increases across the sky, the clouds in this image begin to inflate. This is useful when animating cloud coverage in transitions.

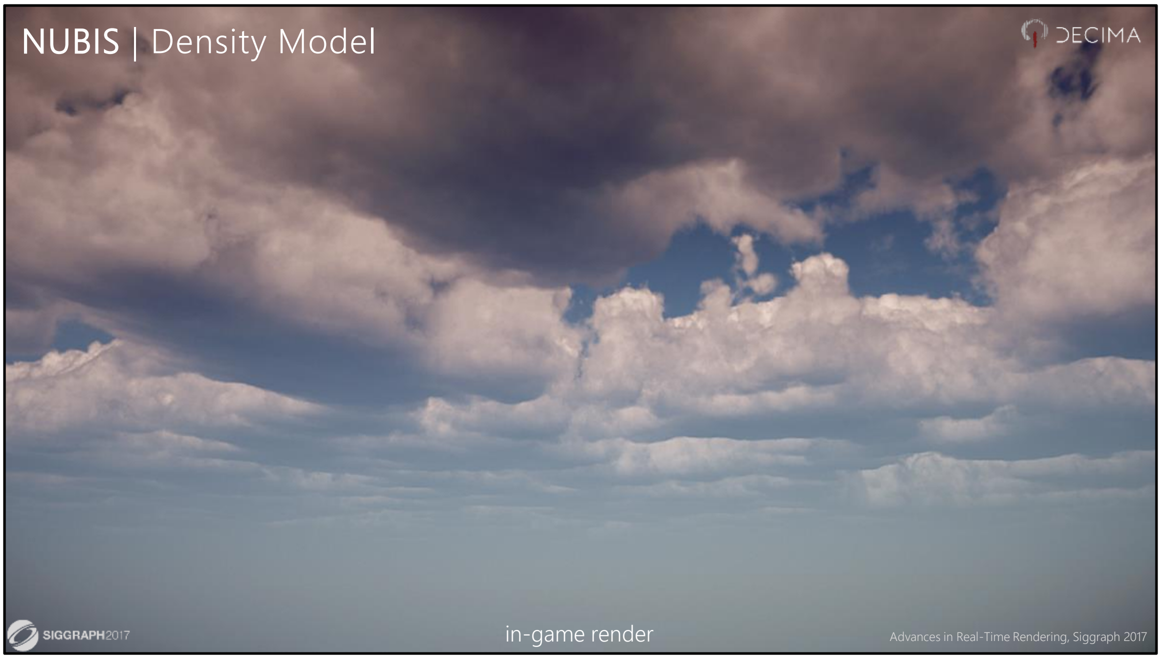


NUBIS | Density Model

in-game render

Advances in Real-Time Rendering, Siggraph 2017

As the coverage value increases across the sky, the clouds in this image begin to inflate. This is useful when animating cloud coverage in transitions.

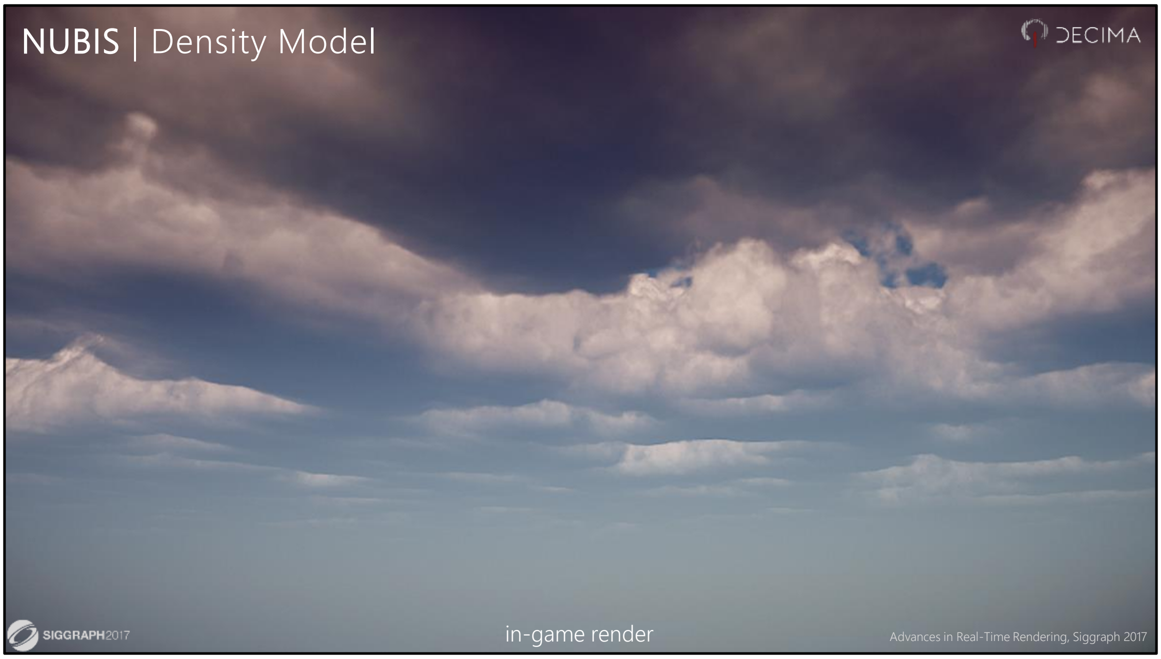


NUBIS | Density Model

in-game render

Advances in Real-Time Rendering, Siggraph 2017

As the coverage value increases across the sky, the clouds in this image begin to inflate. This is useful when animating cloud coverage in transitions.



NUBIS | Density Model

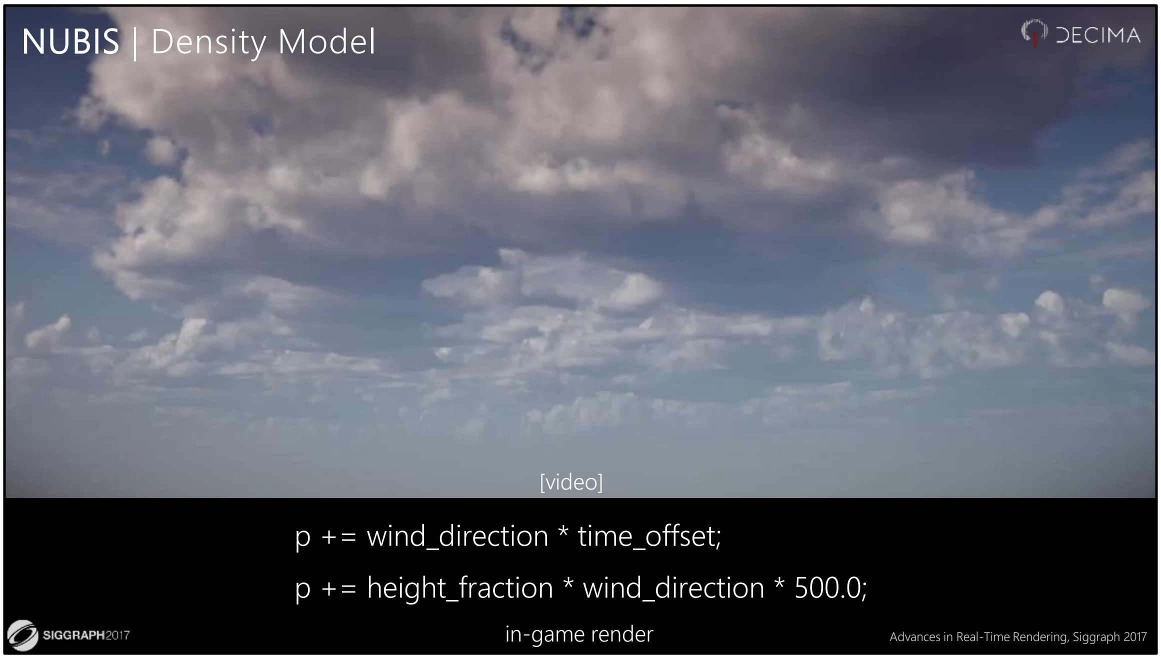
DECIMA

SIGGRAPH2017

in-game render

Advances in Real-Time Rendering, Siggraph 2017

As the coverage value increases across the sky, the clouds in this image begin to inflate. This is useful when animating cloud coverage in transitions.



NUBIS | Density Model

DECIMA

[video]

```
p += wind_direction * time_offset;
p += height_fraction * wind_direction * 500.0;
```

SIGGRAPH2017

in-game render

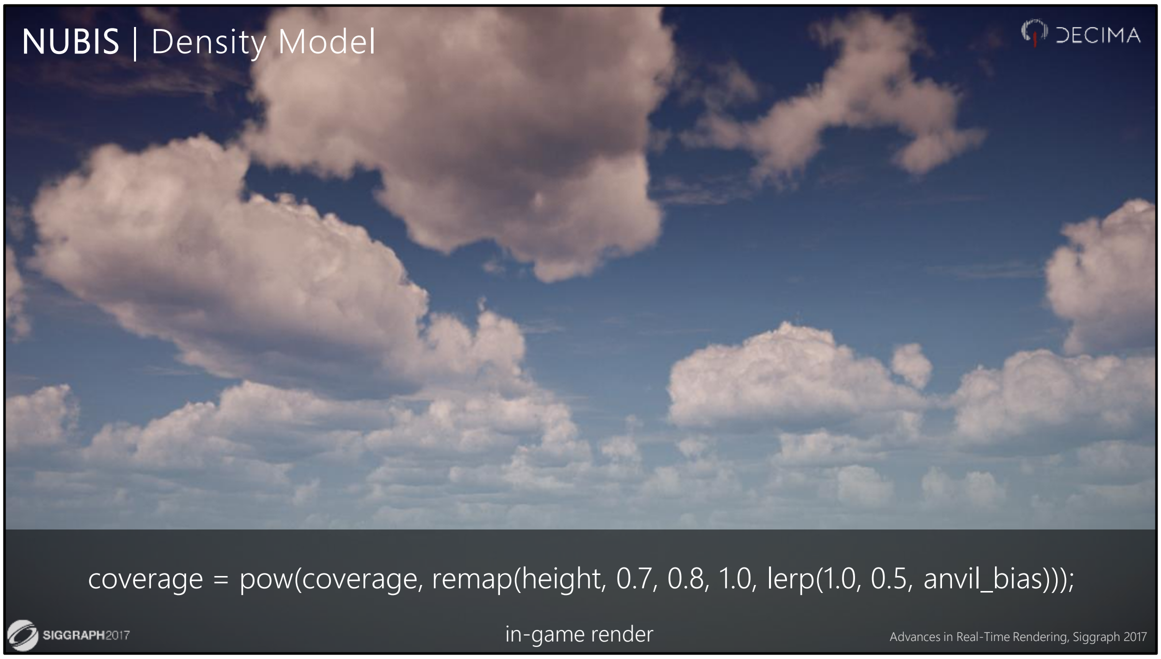
Advances in Real-Time Rendering, Siggraph 2017

In reality, clouds are not free floating objects but ephemeral results in a changing volume of probabilities.

- We can simulate their movement across their region of probability by adding an offset in the wind direction which is incremented in time.
- We can also use the direction of the wind to apply a skew to the clouds in this direction over height.

Here is a sped up example. Notice that the general shapes do not change, but the details do. This is because while the noise in the density model is animated, the coverage signal is not. For us this is an important deviation from our prototype system because when we are constructing art-directed cloudscapes as opposed to a simulated sky, we only want them to appear to be changing without altering the larger structure of the cloudscape.

We are so glad that we didn’t have to figure out how to do this with a skybox.



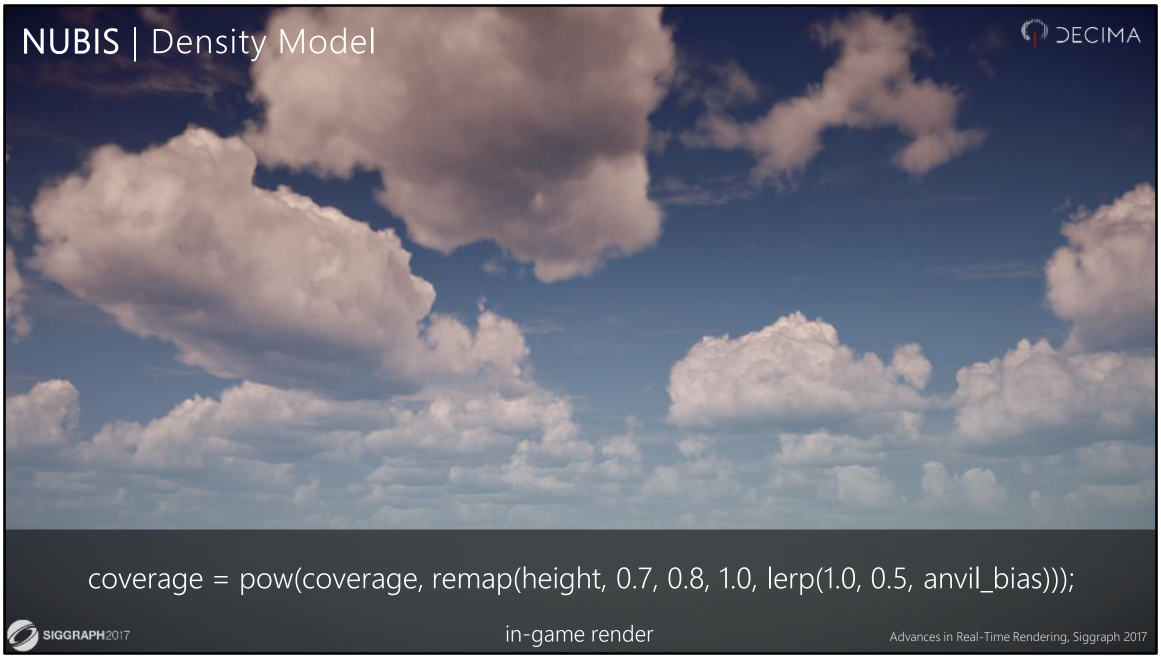
NUBIS | Density Model

```
coverage = pow(coverage, remap(height, 0.7, 0.8, 1.0, lerp(1.0, 0.5, anvil_bias)));
```

in-game render

Advances in Real-Time Rendering, Siggraph 2017

Finally, recall that when we were in the alto and cirro layers, the cumulus cloud started spreading out to form an anvil. To mimic this we treat these anvil shapes as a variation on the cumulus form by inflating the …



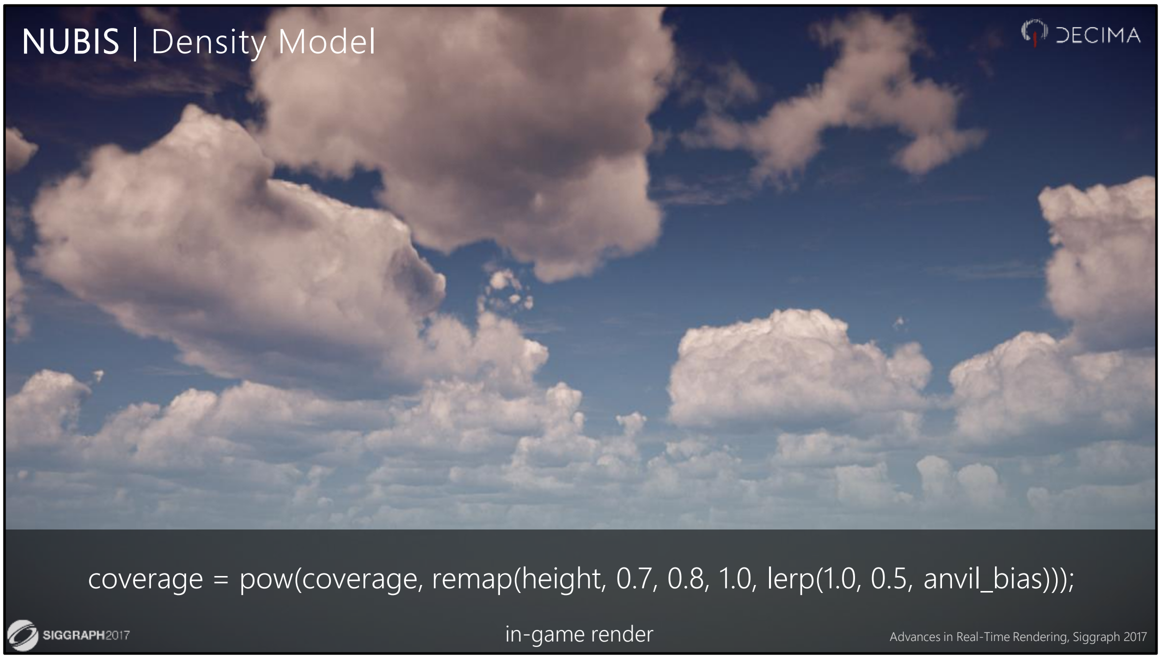
NUBIS | Density Model

```
coverage = pow(coverage, remap(height, 0.7, 0.8, 1.0, lerp(1.0, 0.5, anvil_bias)));
```

in-game render

Advances in Real-Time Rendering, Siggraph 2017

coverage signal where it approaches



NUBIS | Density Model

DECIMA

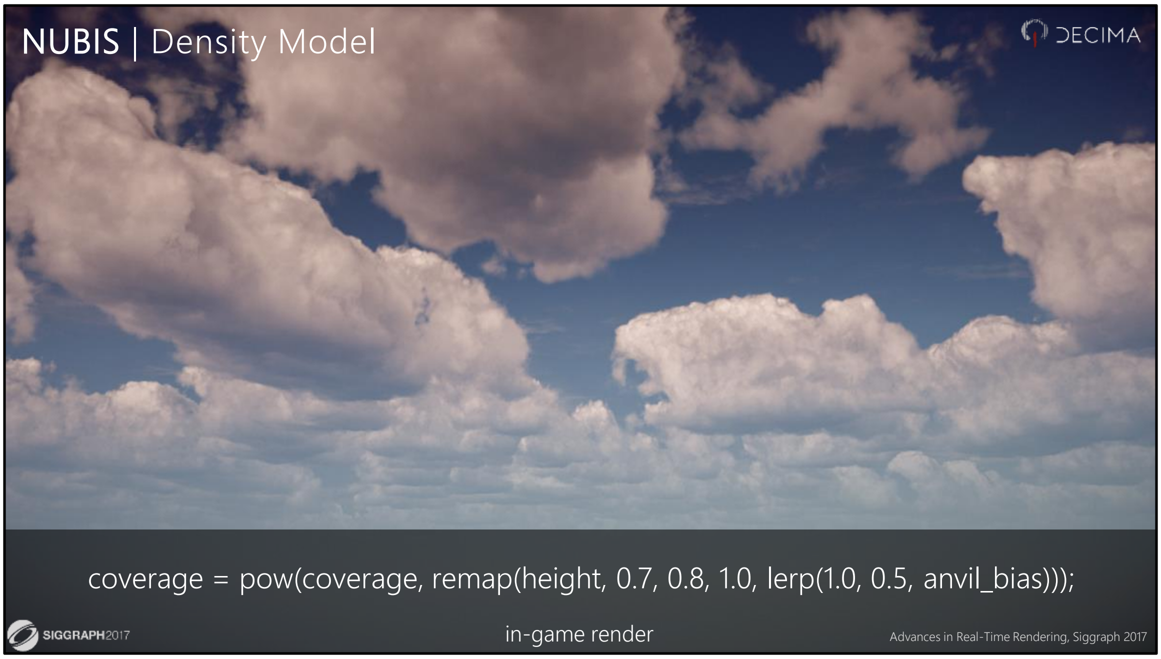
```
coverage = pow(coverage, remap(height, 0.7, 0.8, 1.0, lerp(1.0, 0.5, anvil_bias)));
```

SIGGRAPH2017

in-game render

Advances in Real-Time Rendering, Siggraph 2017

the top of our cloud layer.



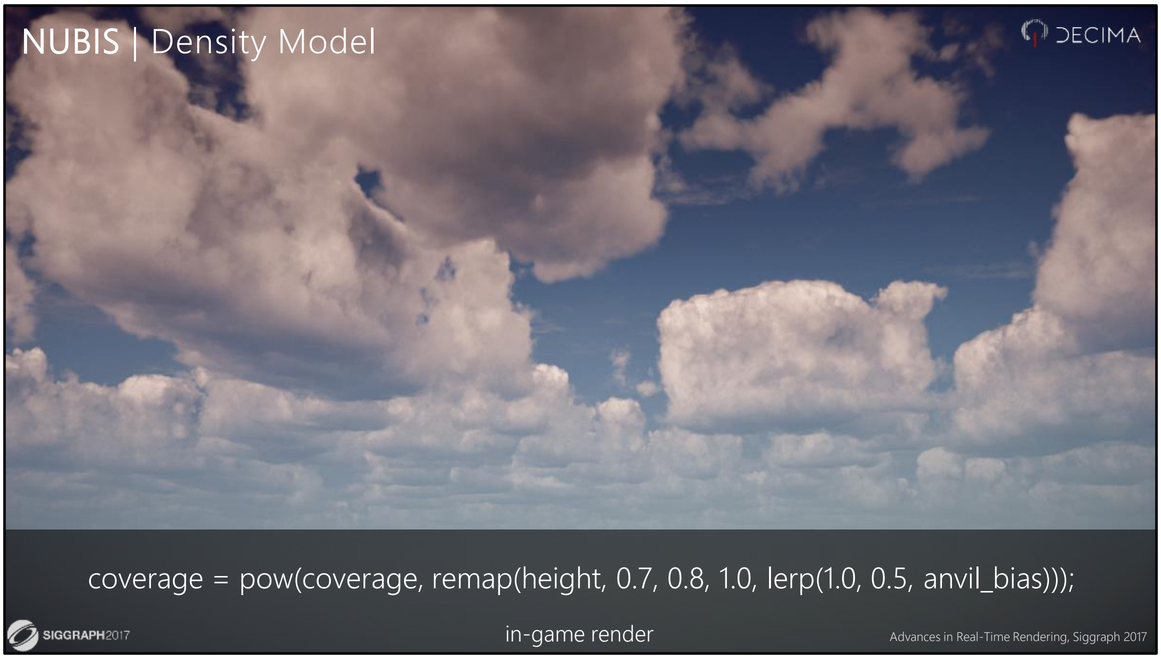
NUBIS | Density Model

```
coverage = pow(coverage, remap(height, 0.7, 0.8, 1.0, lerp(1.0, 0.5, anvil_bias)));
```

in-game render

Advances in Real-Time Rendering, Siggraph 2017

For Dramatic effect, we add a variation to the wind vector that allows …



NUBIS | Density Model

```
coverage = pow(coverage, remap(height, 0.7, 0.8, 1.0, lerp(1.0, 0.5, anvil_bias)));
```

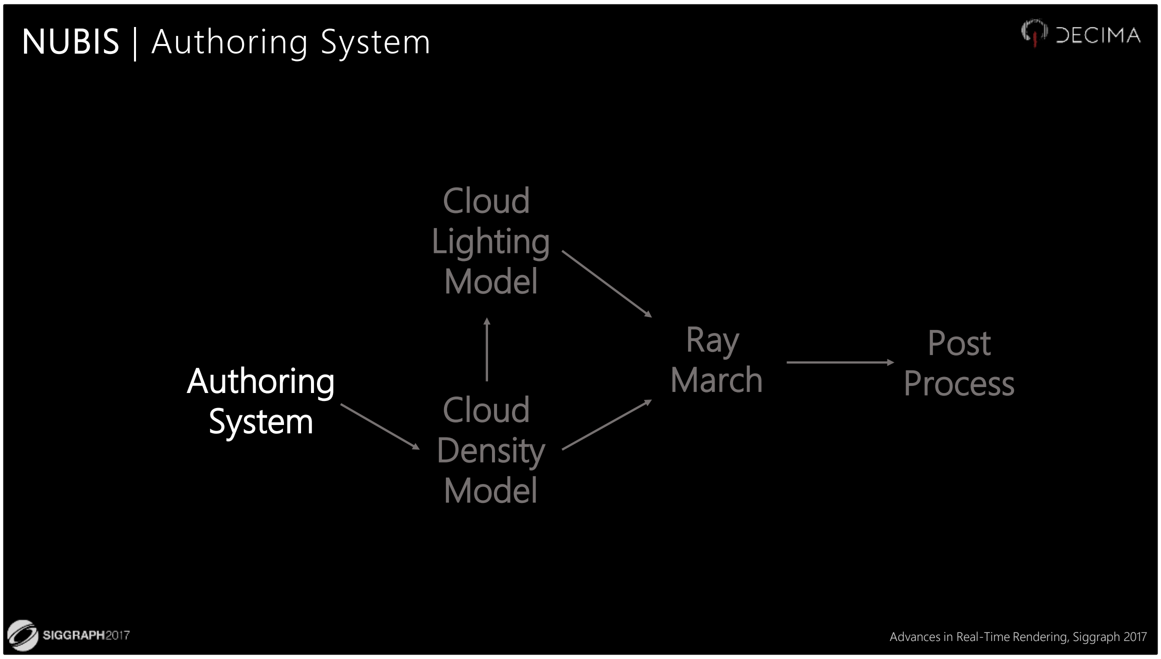
in-game render

us to skew the anvils in a direction that is not natural but looks cool.

So we have modeled some of the characteristics of the clouds of the lower atmosphere, but what about the alto and cirrus clouds?

The version that we shipped with Horizon only models the low clouds and uses a 2D texture lookup for the cirrus clouds, but I will show you some work we have been doing on this a bit later in the talk.

In the slides there will be a code example of how all of these parts are used together in the density sampler.



NUBIS | Authoring System

DECIMA

Cloud Lighting Model

Ray March

Post Process

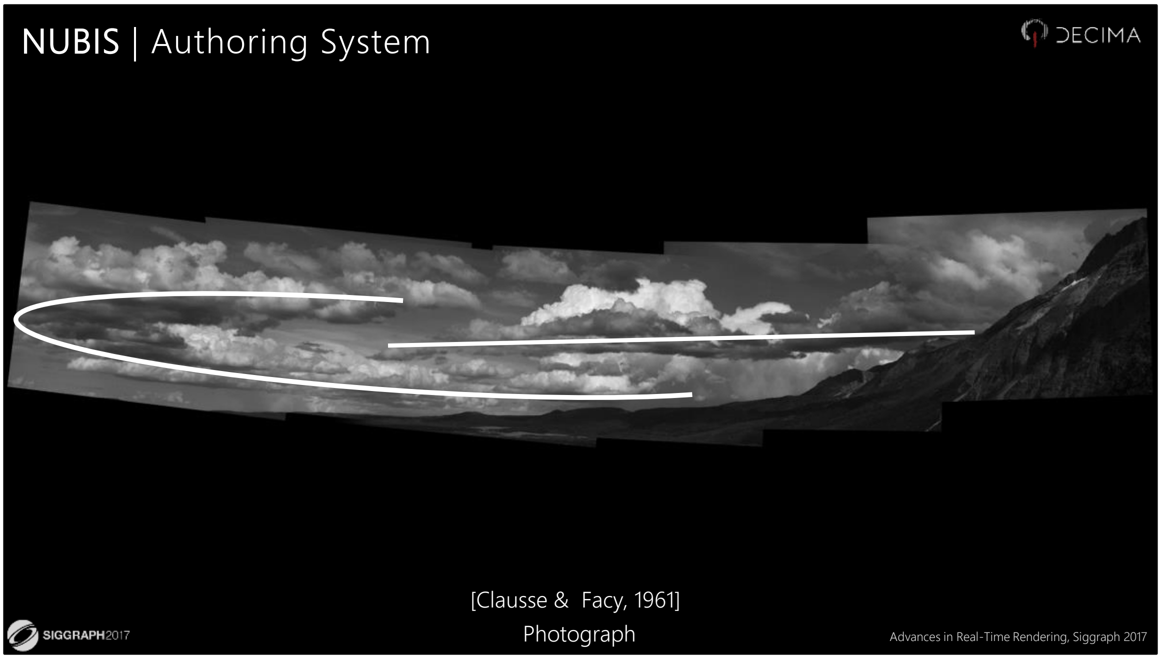
Authoring System

Cloud Density Model

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Now that we have an idea of how to model a single cloud, we can define a procedural system, which generates the coverage and type values in order to construct entire cloud formations.
But what is a cloud formation and how do they form?



NUBIS | Authoring System

DECIMA

[Clausse & Facy, 1961]
Photograph

SIGGRAPH2017

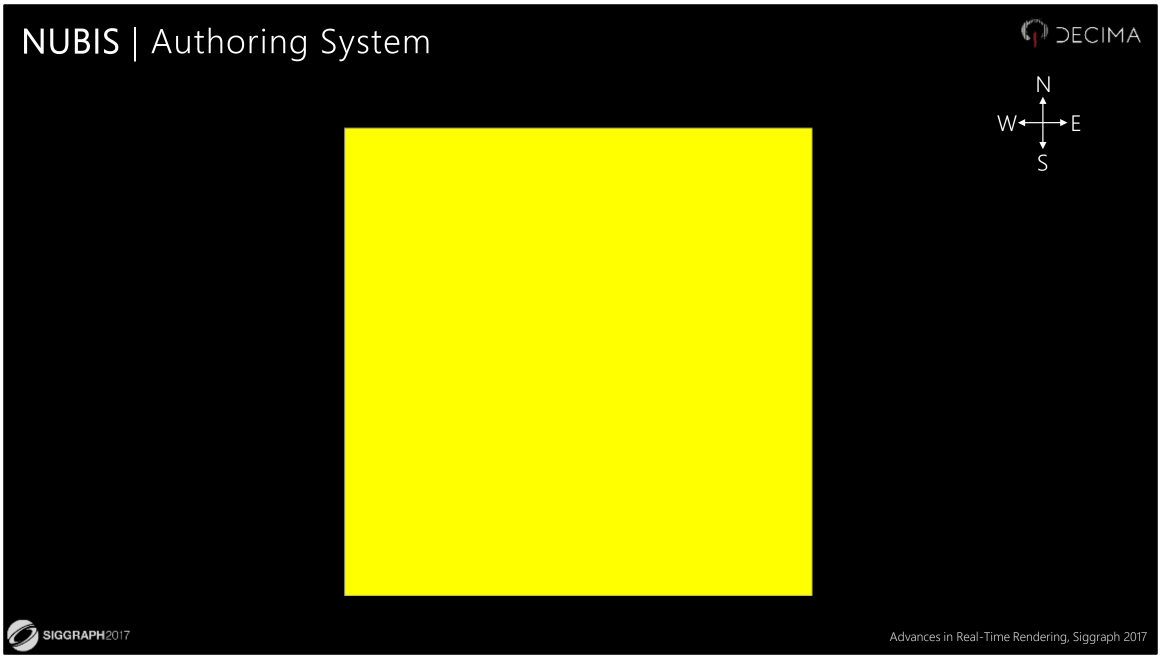
Advances in Real-Time Rendering, Siggraph 2017

According to Clausse and Facy in their seminal book, The Clouds, cloud formations form where two or more masses of air overlap. In these overlap regions, the vapor in the warmer air mass condenses when it contacts the colder air mass.

- If you look closely at this photograph, you can see the places where these air masses meet and allow clouds to form. You can also see that there are gaps in these areas where no clouds form or where their type changes.

Recall that our density model defines clouds according to Type and Coverage values.

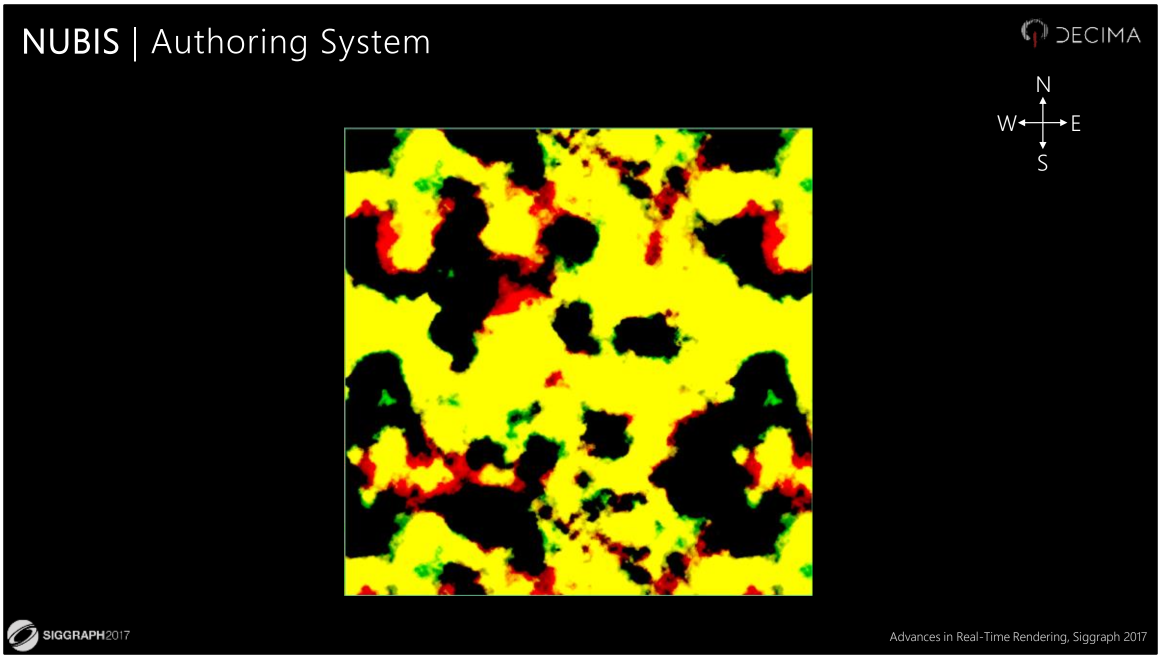
As mentioned in our 2015 course, we generate these variations over space using several octaves of noise, and produce what we call a cloud map.



NUBIS | Authoring System

Advances in Real-Time Rendering, Siggraph 2017

The cloud map is an RGB buffer which represents a coverage area of about 100 square kilometers. The red and green channels represent cloud coverage. In this example cloud coverage is at 100% for the whole map.



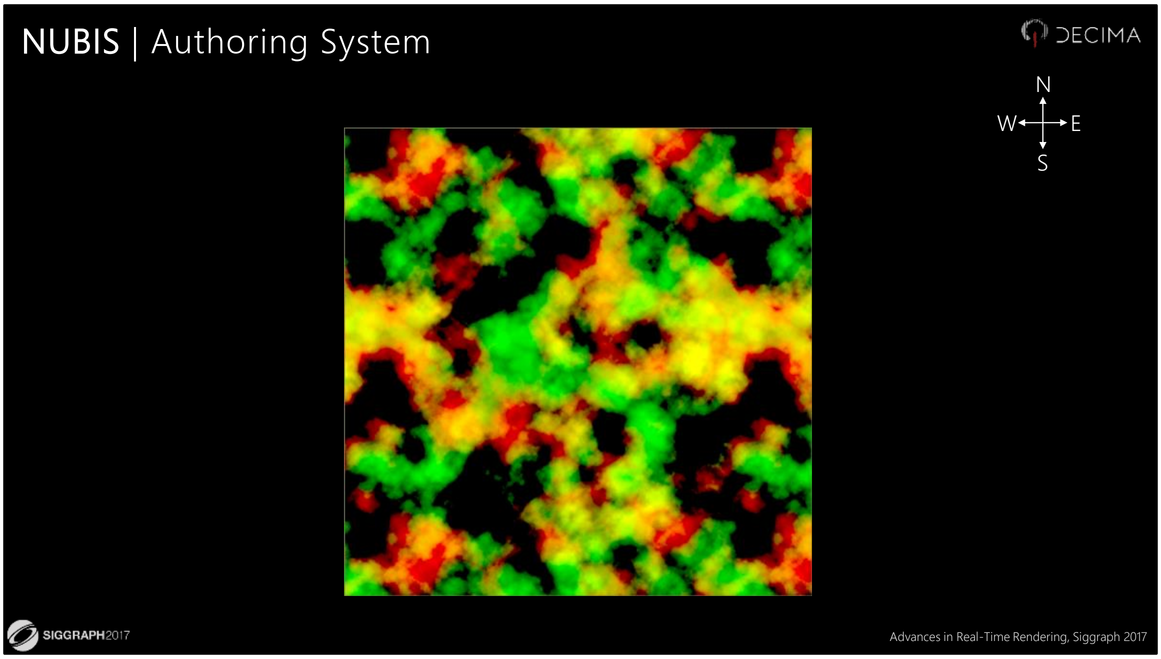
NUBIS | Authoring System

DECIMA

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

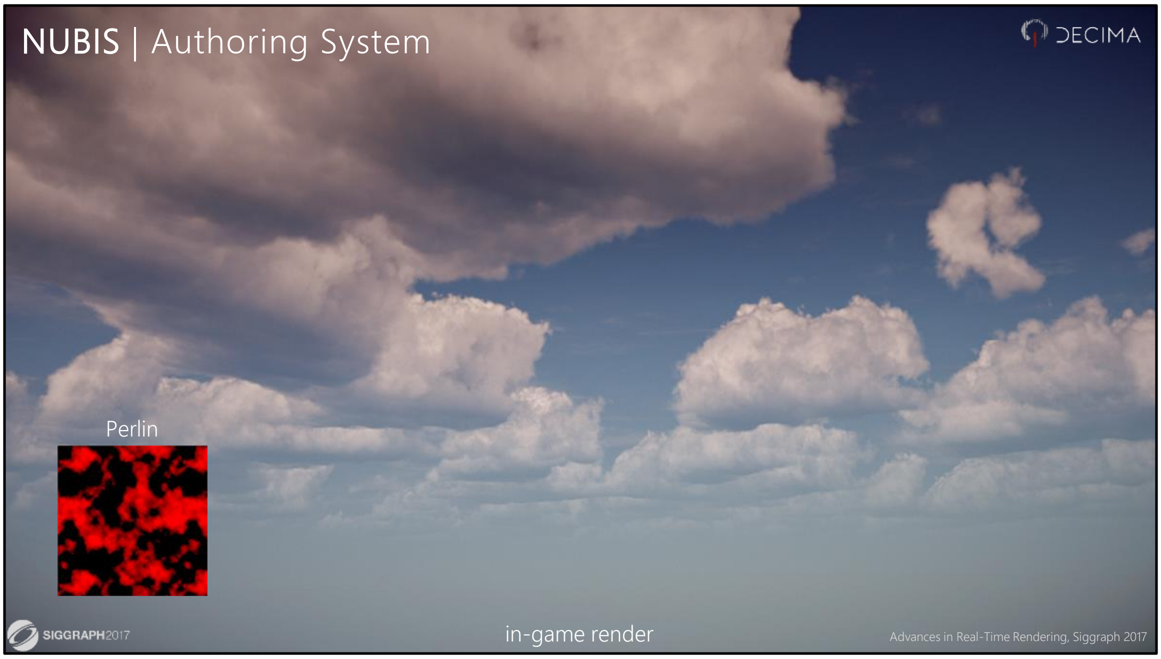
A low frequency noise modulates cloud coverage to simulate those large regions where air masses overlap and clouds are allowed to form.



NUBIS | Authoring System

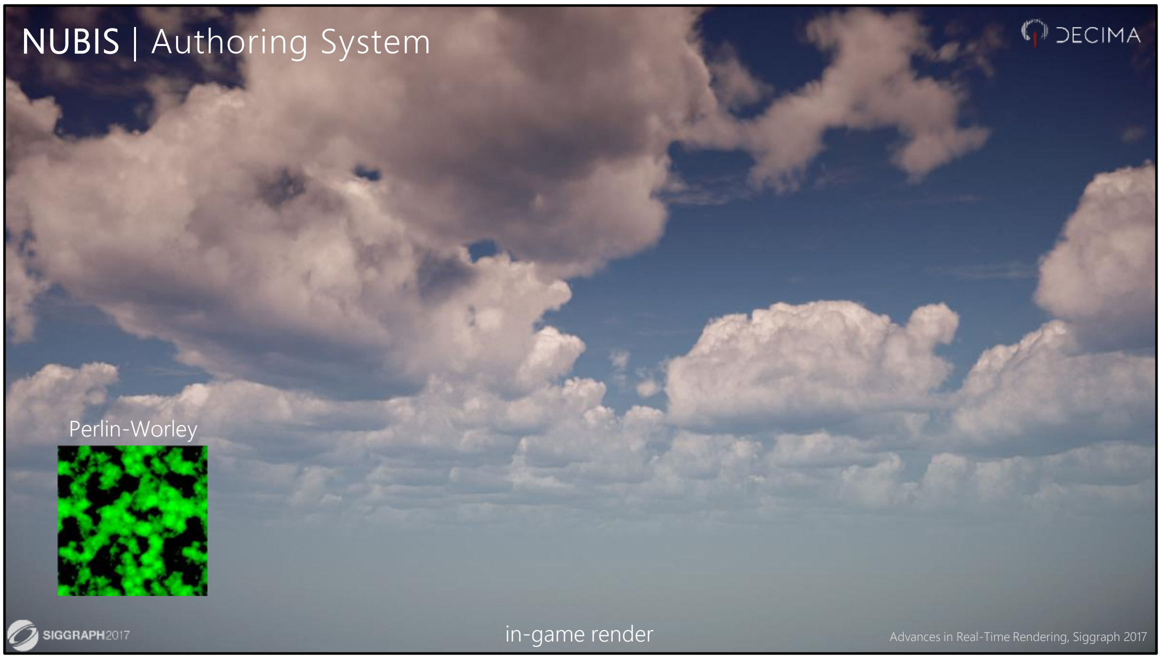
Advances in Real-Time Rendering, Siggraph 2017

Several higher frequency noises are composited with this to simulate the variations in vapor emission in these regions which form the footprints of our clouds. Perlin noise is used in the red channel , while a Perlin-Worley noise is used in the green channel.



NUBIS | Authoring System

Either noise is used in the density sampler depending on if the artist wants connected cloud formations like this from Perlin noise…

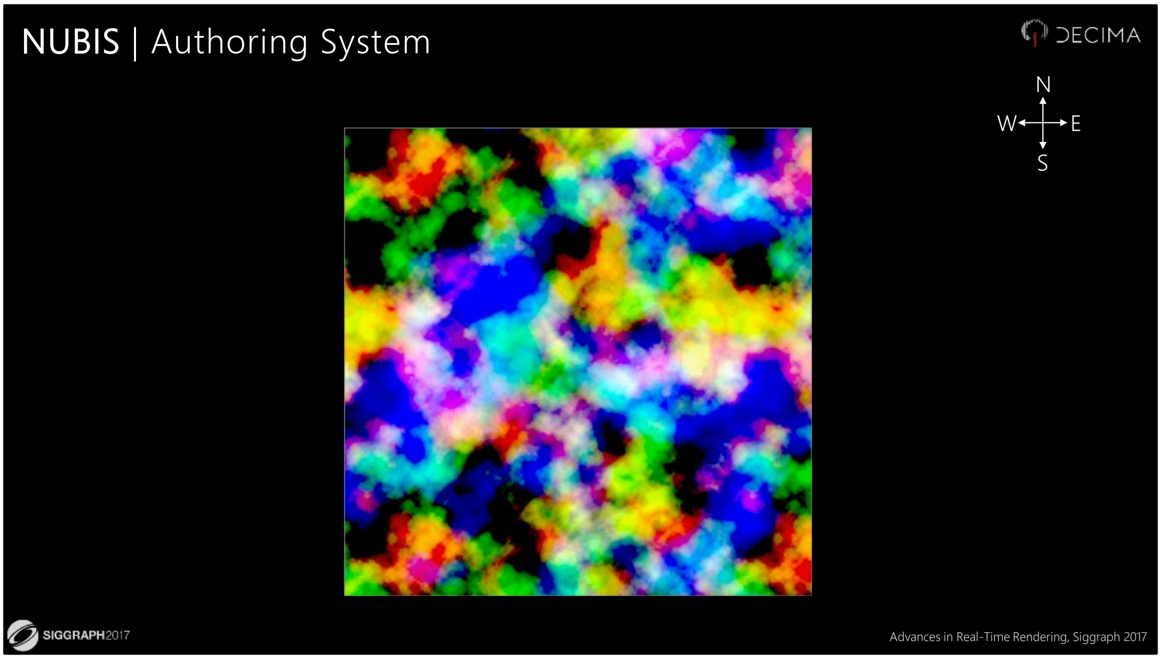


NUBIS | Authoring System

in-game render

Advances in Real-Time Rendering, Siggraph 2017

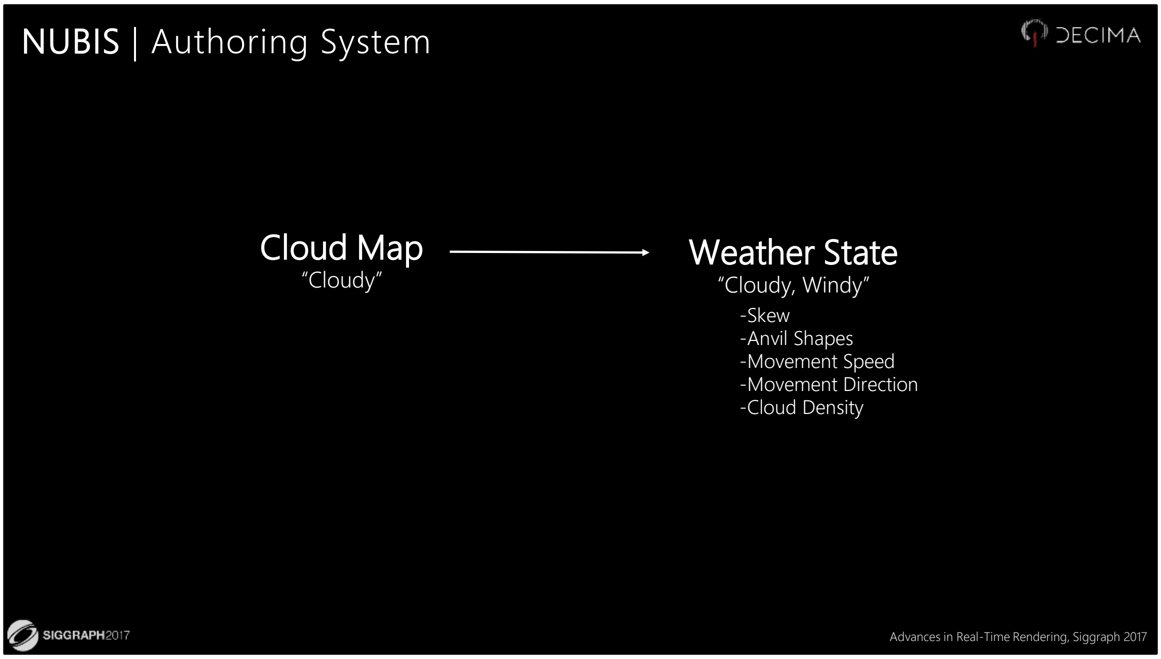
…or more isolated island like shapes with connective tissue from Perlin-Worley noise.



NUBIS | Authoring System

Advances in Real-Time Rendering, Siggraph 2017

Finally, Another low frequency noise is used to define the modulation in cloud type. This goes into the blue channel, with a value of Zero being stratus, , 1 being cumulus or value of .5 indicates stratocumulus clouds.
Recall that there are other things going on in our density model…



NUBIS | Authoring System

DECIMA

Cloud Map
"Cloudy"

Weather State
"Cloudy, Windy"

-Skew
-Anvil Shapes
-Movement Speed
-Movement Direction
-Cloud Density

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

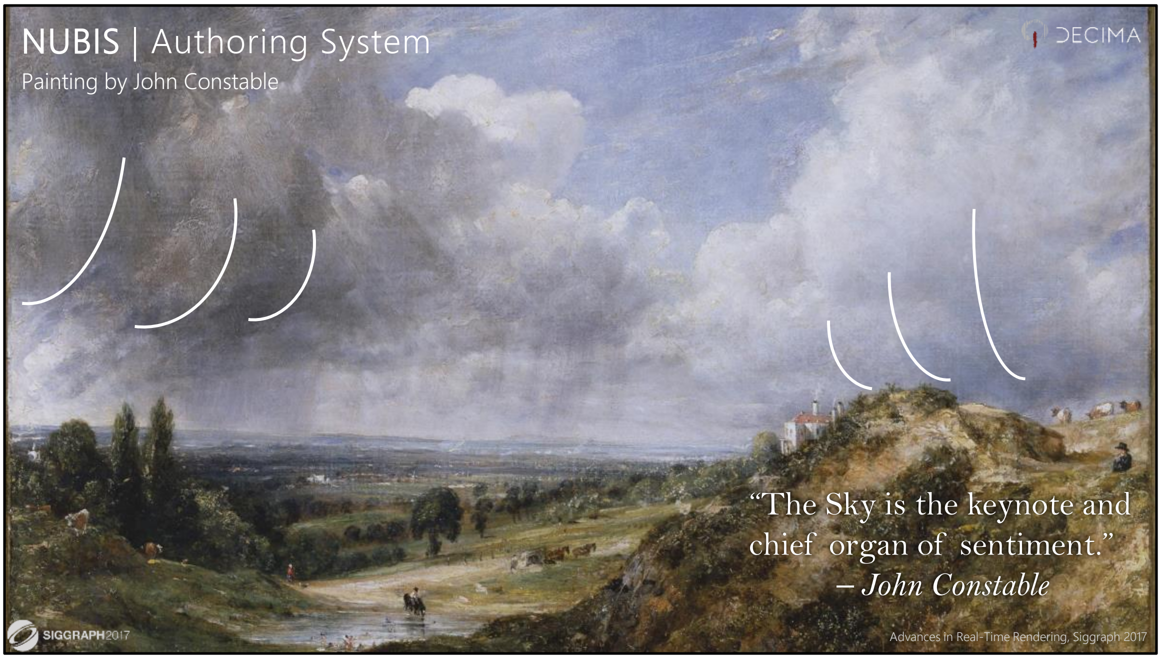
The cloud map is just a part of a larger set of instructions that tell our density model what to produce.

- The map is incorporated into what we call a weather state. The weather state adds more characteristics used in modeling clouds.

These include
- Skew,
- Anvil inflation,
- Movement
- and direction controls and
- density controls.

These are treated same at every sample, so they didn't need to be stored in the Cloud Map.

Before we look at how to use all of these controls to build various cloudscapes, Lets briefly look at how cloudscapes are used in art and photography as this informs our approach for Horizon.

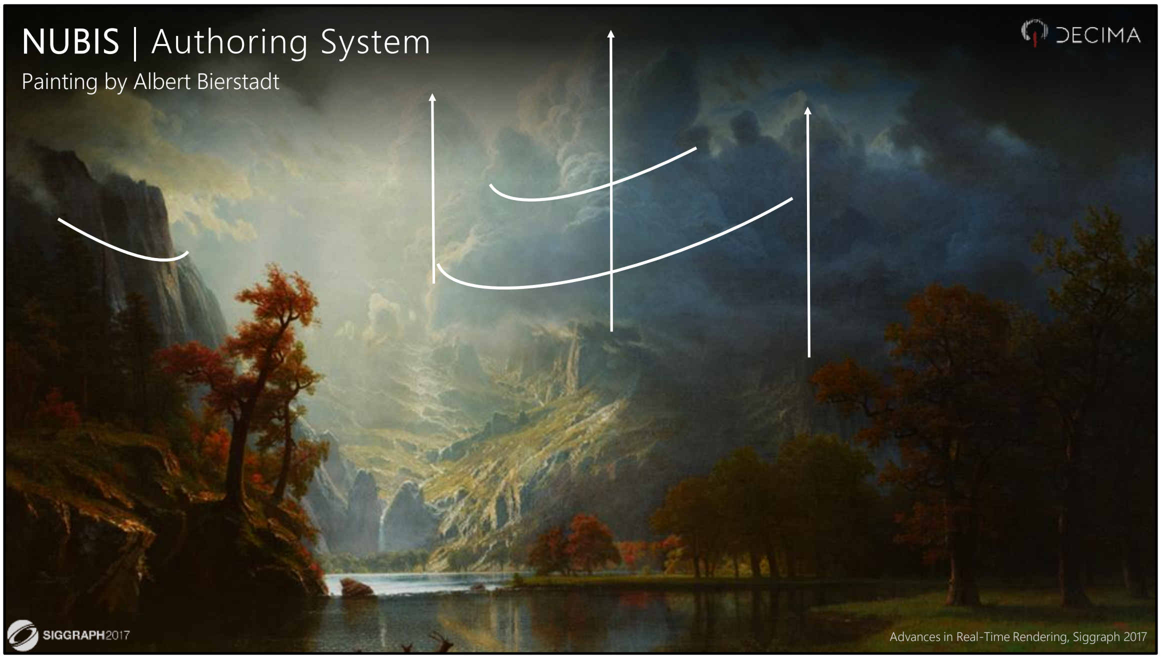

NUBIS | Authoring System

Painting by John Constable

"The Sky is the keynote and chief organ of sentiment."
– *John Constable*

We should always start with the British landscape painter, John Constable.

- Look at the subtle shapes produced by the clouds. [His arrangement of clouds in this work adds depth layers in the top of the frame in a way that is similar to cascading curtains that are pulled back at the beginning of a play.]
- Constable said that the "Sky is the keynote and the chief organ of sentiment." What he meant was that clouds are an important emotional element of any landscape artwork regardless of how visible they are. When we think back to how early man feared the storms that brought destruction, or the clear skies that could bring drought, it is easy to understand why this emotional response is so engrained. We don't even realize how images like this affect us emotionally until we actually think about it. In games, we want to elicit emotional responses like this, so for Nubis, the clouds, as viewed from each camera angle, would need to evoke a similar emotional response.

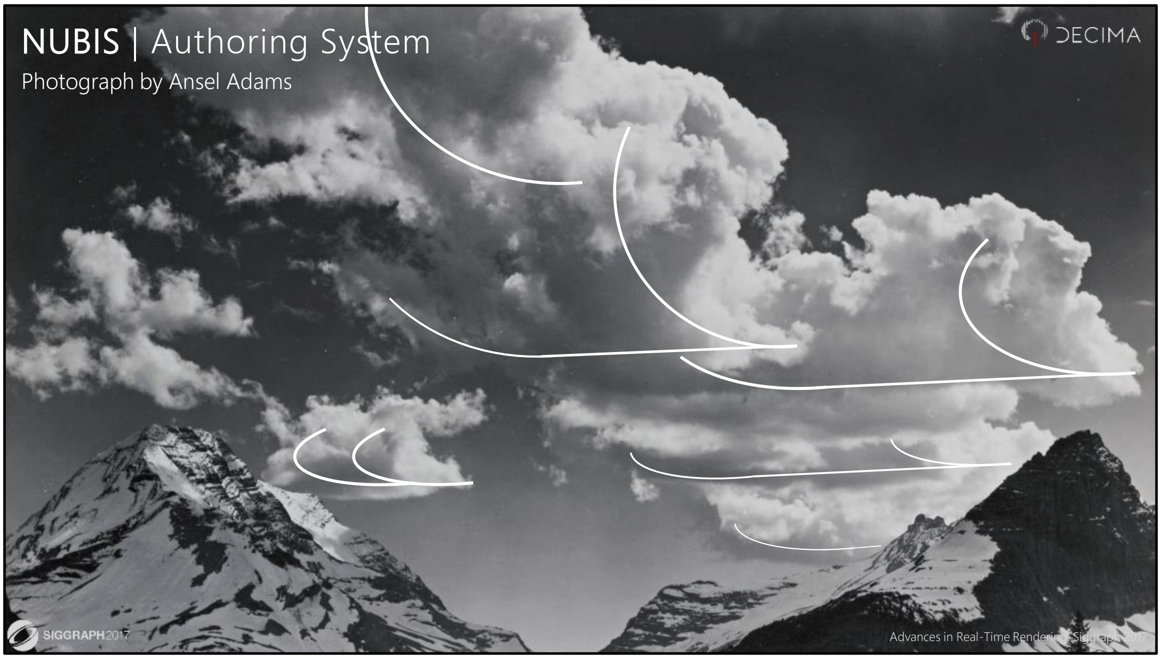


NUBIS | Authoring System

Painting by Albert Bierstadt

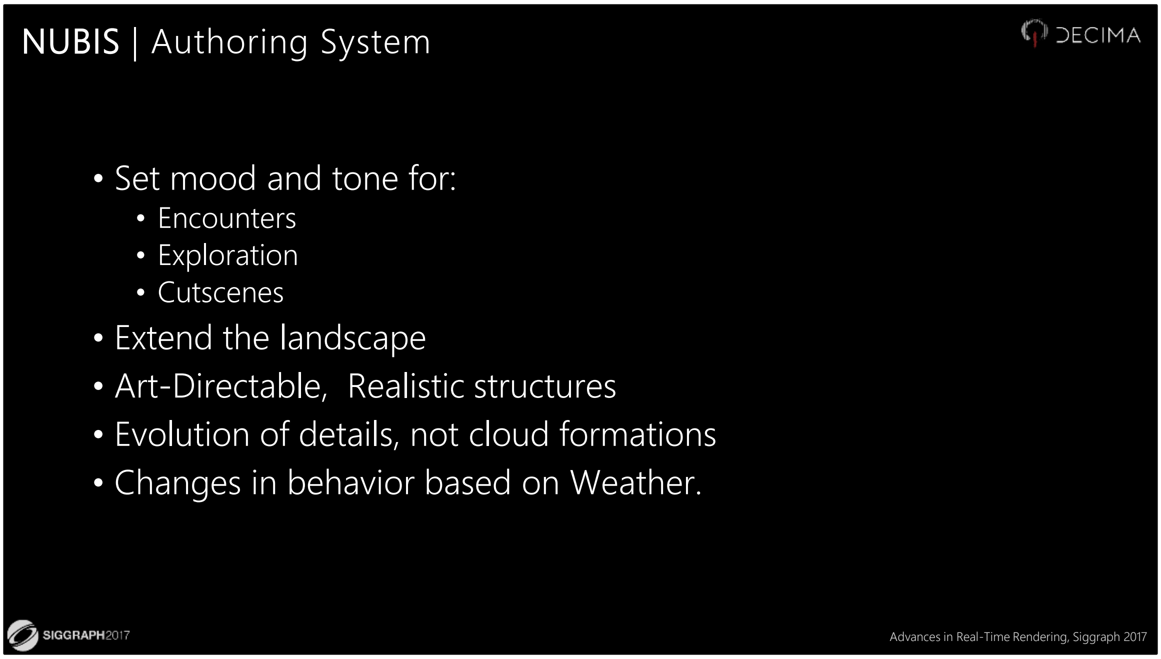
Artists like Albert Bierstadt, here, were part of a natural romanticism movement, which really captured several aspects of what we were trying to achieve for Horizon.

- This image promotes a feeling of mystic awe as the eye is drawn to the dramatic bright area in the mountains where pillars of cumuloform clouds appear to
- grow out of the mountains, which has the effect of extending the landscape vertically.


NUBIS | Authoring System
Photograph by Ansel Adams
DECIMA
SIGGRAPH2017
Advances in Real-Time Rendering | Siggraph 2017

Other artists, such as the Photographer, Ansel Adams, were also very important for us because of the detail, tonal contrasts and composition of clouds

- which lend dimensionality to the sky. His images are Iconic representations of the American West, as well.



NUBIS | Authoring System

- Set mood and tone for:
  - Encounters
  - Exploration
  - Cutscenes
- Extend the landscape
- Art-Directable, Realistic structures
- Evolution of details, not cloud formations
- Changes in behavior based on Weather.

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

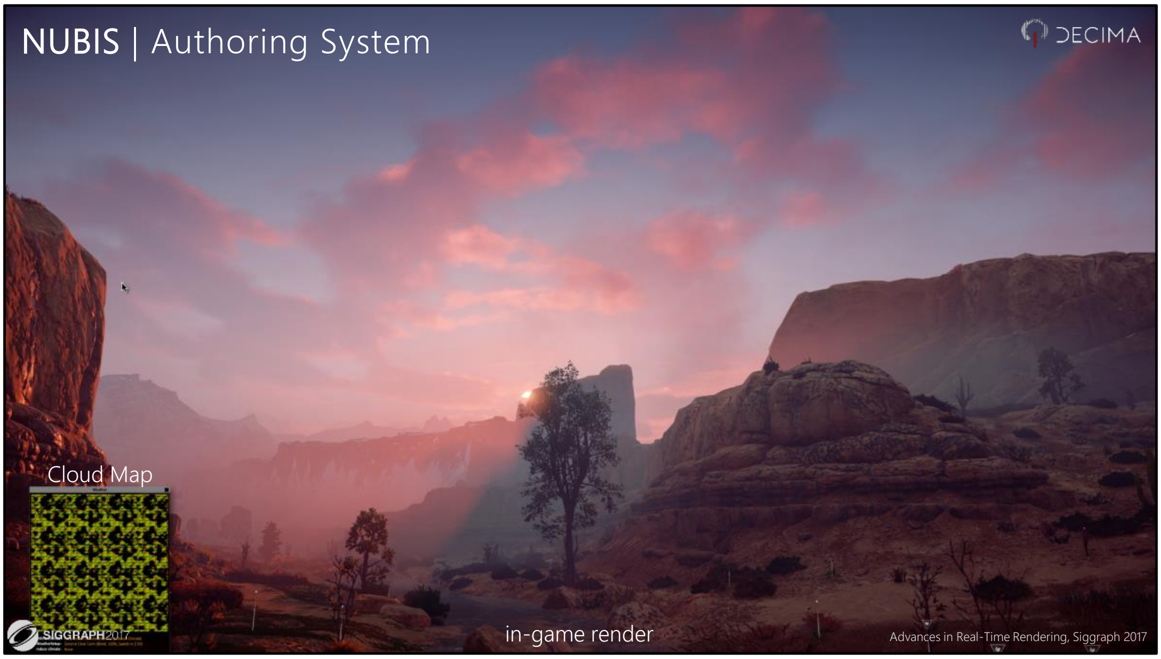
So the precedents set by these artists were clear.

- We would need to help set the tone for Encounters, cut scenes and exploration of the world of Horizon.
- The cloudscapes should also extend the landscape of the world and

- Design wise, we also wanted to be able to generate realistic structures, not purely random noise patterns.
- Additionally, we knew that we wanted the clouds to evolve over time, but this evolution should not break the general arrangement of our cloudscapes, and
- Finally, That the cloud formations themselves would need to change as the weather conditions changed.

- Now that I have covered our methods and goals for cloudscapes, lets look at some examples from horizon.

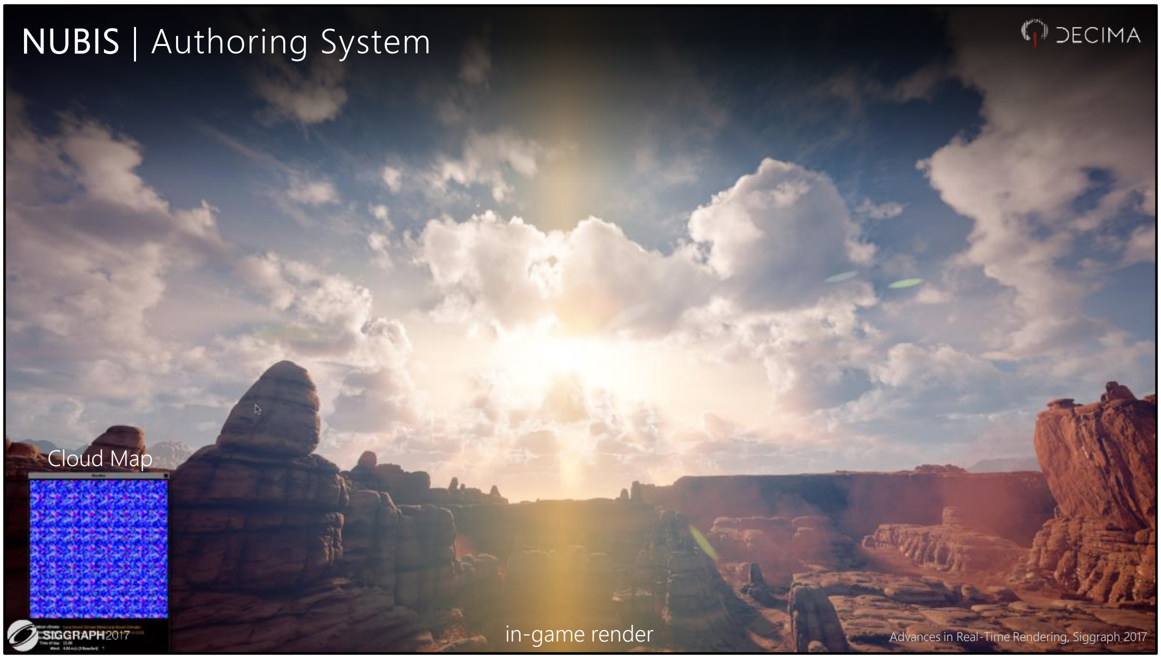
NUBIS | Authoring System



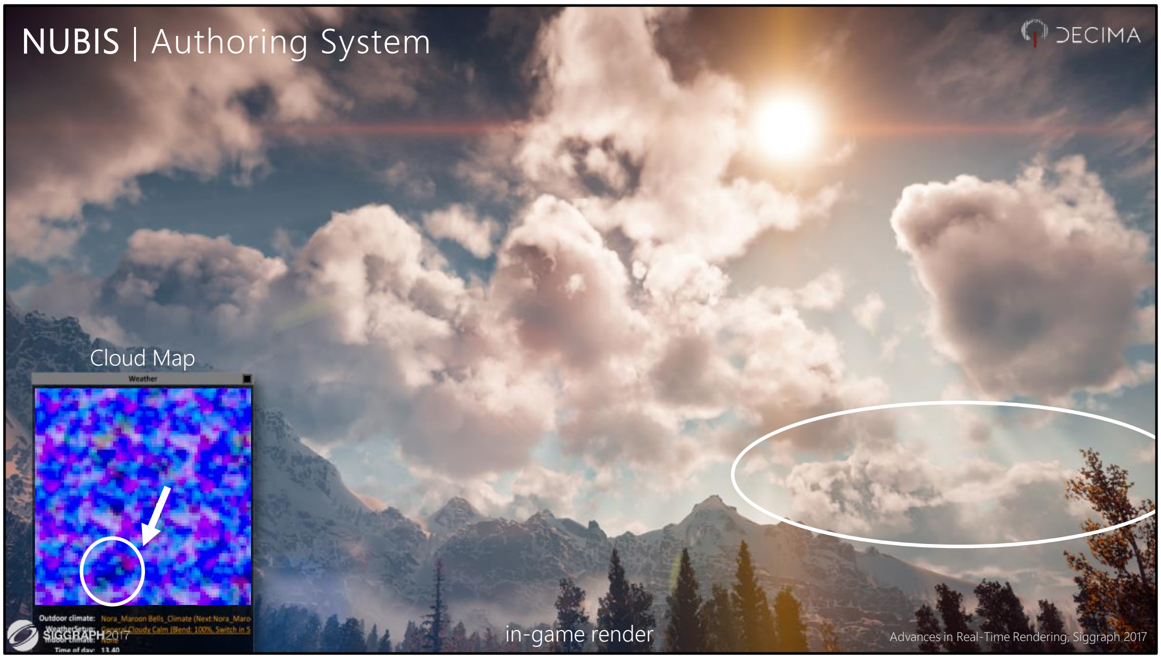
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Mornings in the desert can be quite cold. If there is any vapor emission, it will probably create stratus clouds. So Cloud type was set to zero for this cloud map as you can see by the lack of Blue. We also wanted our mornings to be calm and peaceful, so In situations like this the clouds were meant to be felt more than seen, such as in the Constable painting.


NUBIS | Authoring System

In contrast, the probability of cumulus clouds can increase in the afternoons, So this cloud map uses a higher cloud type value. We modeled the desert cumulus clouds after the Ansel Adams photographs.



NUBIS | Authoring System

Outdoor climate: Nora_Maroon Bells_Climate (Next:Nora_Maro

WeatherSetup: General Cloudy Calm (Blend: 100%, Switch in 5

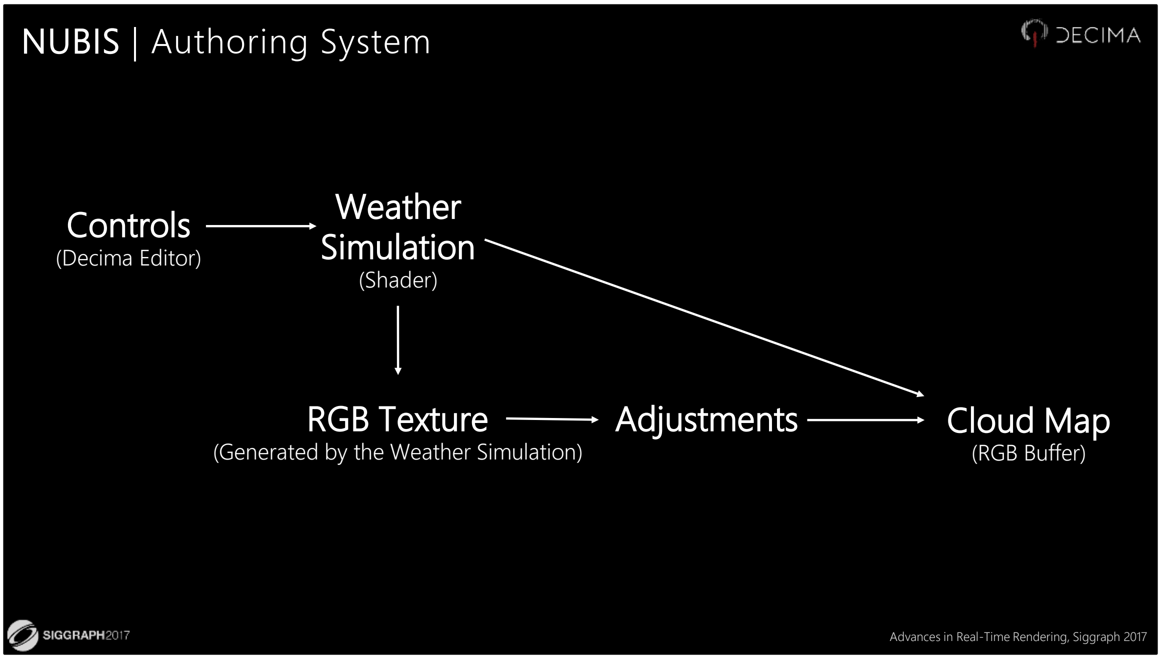
Indoor climate: None

Time of day: 13.40

Advances in Real-Time Rendering, Siggraph 2017

In the alpine regions, we tried extending the landscape …Bierstadt style… by modeling isolated cumulus congestus columns.
The big white arrow in the weather map indicates the camera direction.
If you look closely, you can see where

- the reduction in the blue channel corresponds
- to a stratus cloud band.



NUBIS | Authoring System

DECIMA

Controls
(Decima Editor)

Weather Simulation
(Shader)

RGB Texture
(Generated by the Weather Simulation)

Adjustments

Cloud Map
(RGB Buffer)

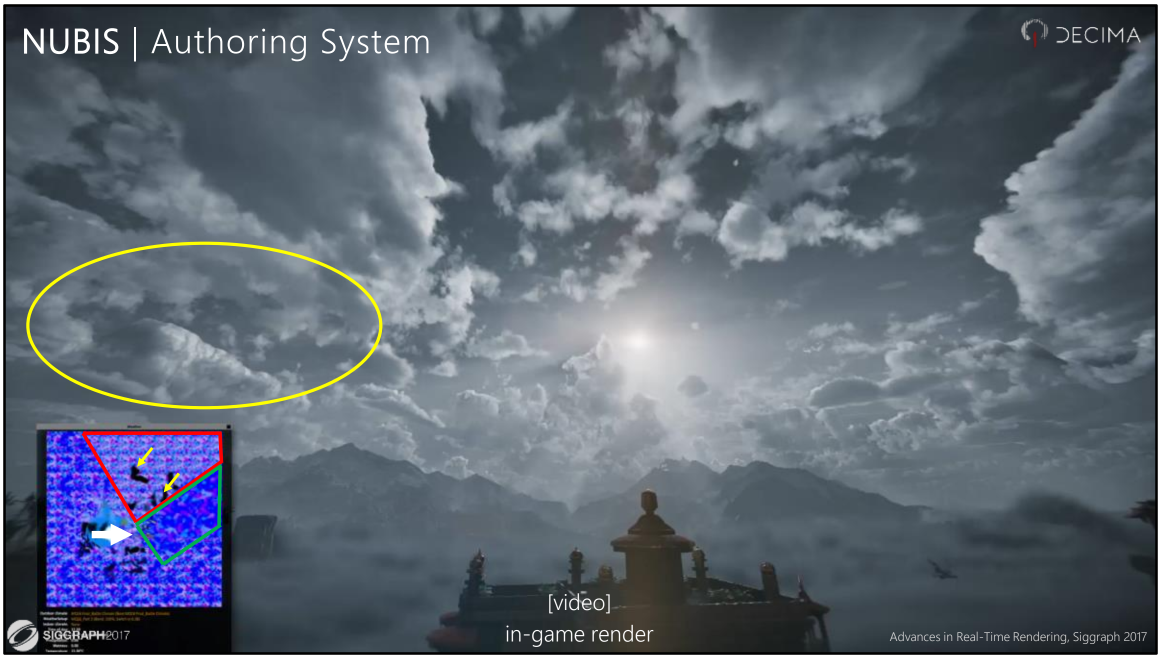
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

We used the weather simulation to produce procedural cloud maps like these for most of the game, but there were some situations, like cutscenes and important events, where the cloudscape needed a bit more control.

- We can override the result of the weather system using any texture. So, What we do to in cases like these is
- adjust the results of the weather simulation in an image editor by increasing or decreasing coverage and type in specific areas.

Here’s a dramatic example.



NUBIS | Authoring System

[video]
in-game render

Advances in Real-Time Rendering, Siggraph 2017

This is a custom cloud map that I made for the Big boss fight at the end of the game.

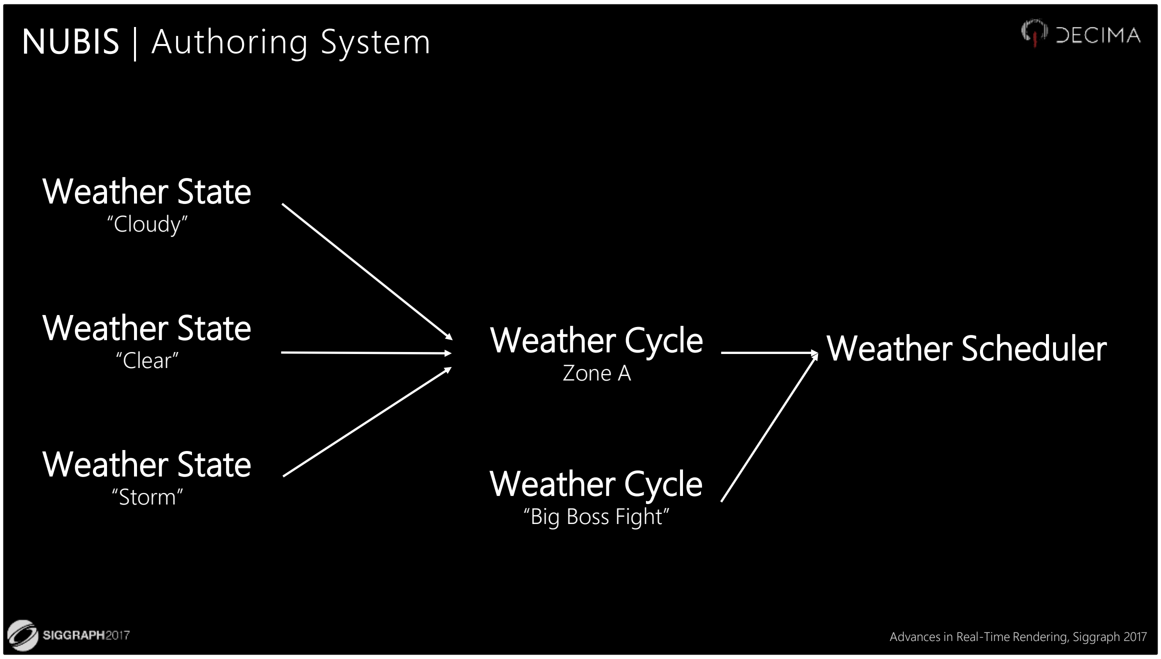
In the weather map in the corner of the screen, the white arrow indicates the camera direction. The point of the arrow indicates the camera position. You can see that the big blue area with low coverage corresponds to the opening around the sun in the image.

- By combining several weather maps I made a kind of Frankencloudscape that incorporated the characteristics of several different systems at once.

- Also, when the the art director came by and said “I want a hole right here” it was as simple as
- painting black on to the cloud map in the area that he indicated.

This is an area that really interests us. We are beginning to look at ways to make this more intuitive and procedural in the future.

So I have described how we generate procedural or custom cloud maps for a given location, but we also Have to account for the changing weather conditions throughout the day.



NUBIS | Authoring System

DECIMA

Weather State
"Cloudy"

Weather State
"Clear"

Weather State
"Storm"

Weather Cycle
Zone A

Weather Cycle
"Big Boss Fight"

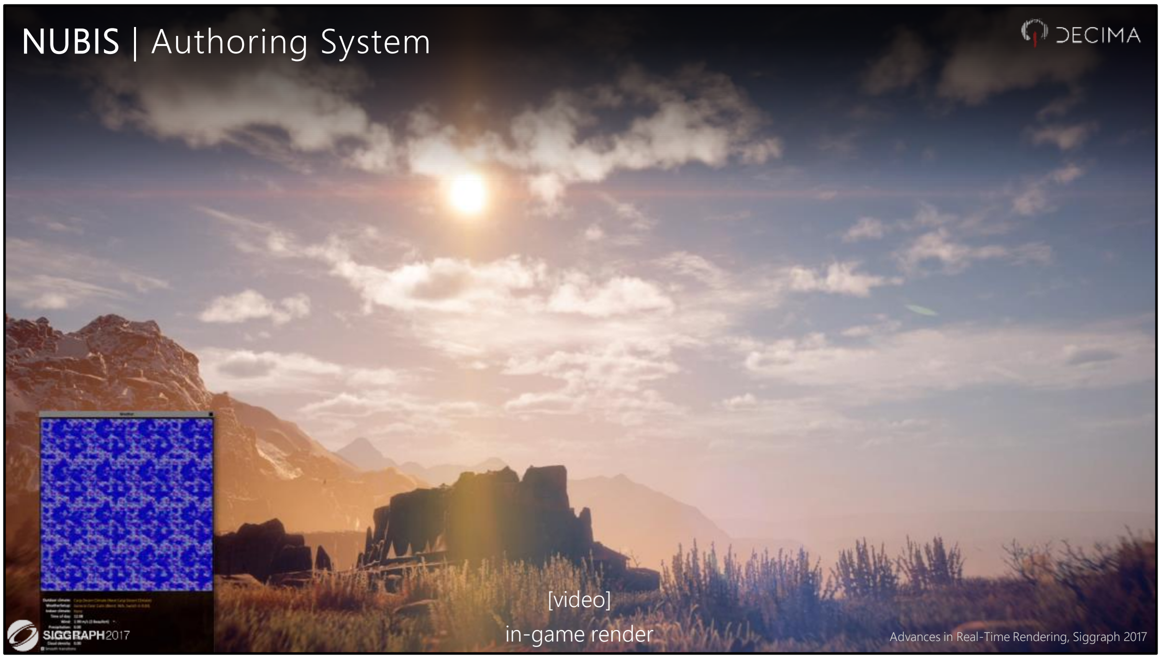
Weather Scheduler

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Based on the climate and Art direction for a given region or Zone in the world map,

- we made a collection of weather states to cycle between. In this cycle, we blend between the 3 basic presets in the game. The States in Zone A, for example,
- were used in the weather scheduler, Which decides what weather state to transition to next. The weather scheduler makes its decisions based on chance, but we have control over which weather states that it can use based on the time of day. Recall that Cumulus clouds usually don't appear in the morning.
- For Encounters like a boss fight, we defined specific cycles that overrode the regional cycles.



NUBIS | Authoring System

DECIMA

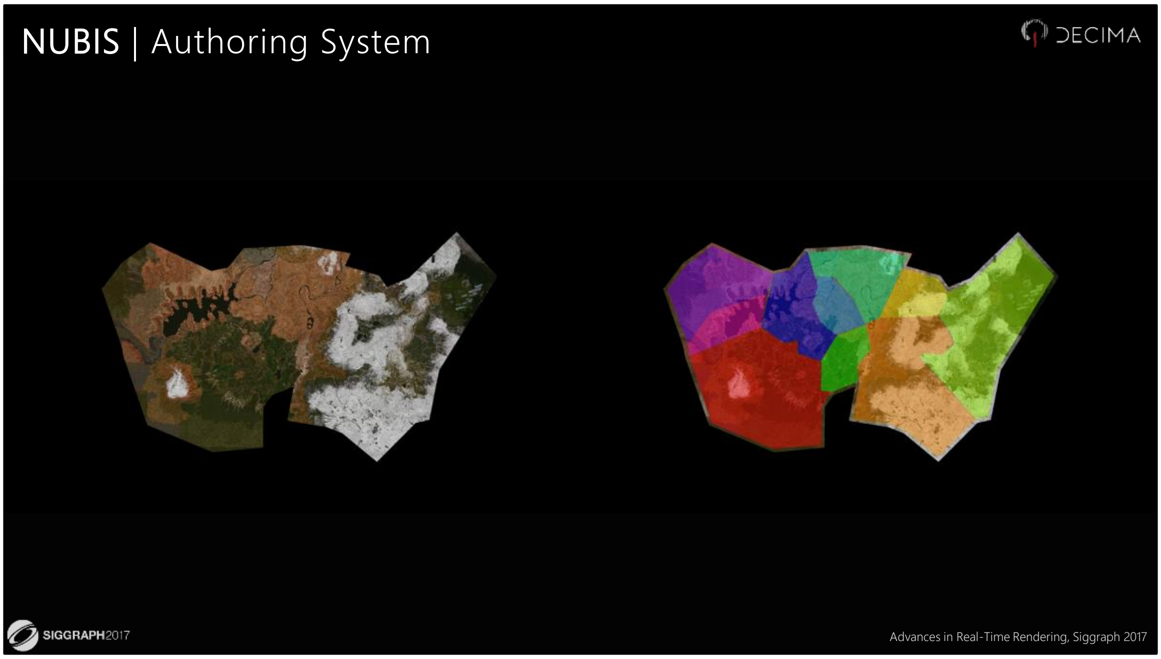
[video]
in-game render

SIGGRAPH 2017

Advances in Real-Time Rendering, Siggraph 2017

The switch from one state to another is handled as a blend that takes place over about 10-30 seconds depending on the situation. This example has been sped up, otherwise you wouldn’t really notice the change. But you can check it out yourself in the game.

It’s a simple lerp between the current state and the next one in the schedule



NUBIS | Authoring System

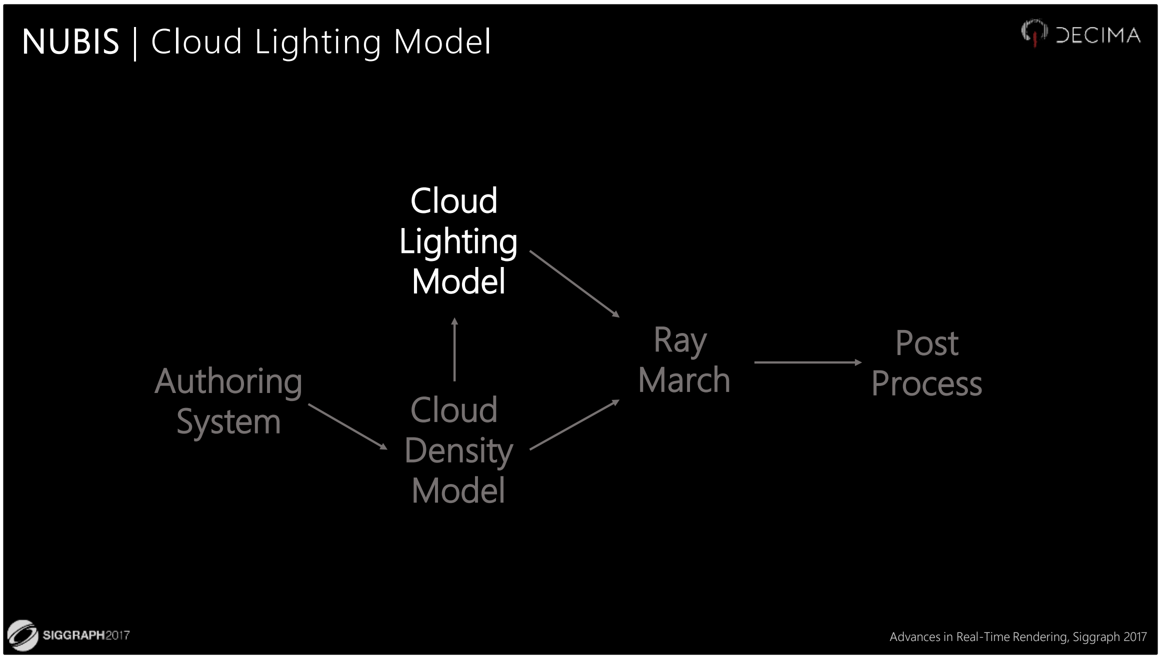
DECIMA

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

And what about regional differences in climate and art direction? We had 10 regional zones in Horizon and 70 zones for things like Bandit camps , where the art director always wanted foreboding dark clouds .

As you move between these zones, The weather Scheduler replaces the next weather state in the schedule with the first randomly selected weather state in the new zone, and then starts the transition immediately.



NUBIS | Cloud Lighting Model

Cloud Lighting Model

Authoring System

Cloud Density Model

Ray March

Post Process

SIGGRAPH2017

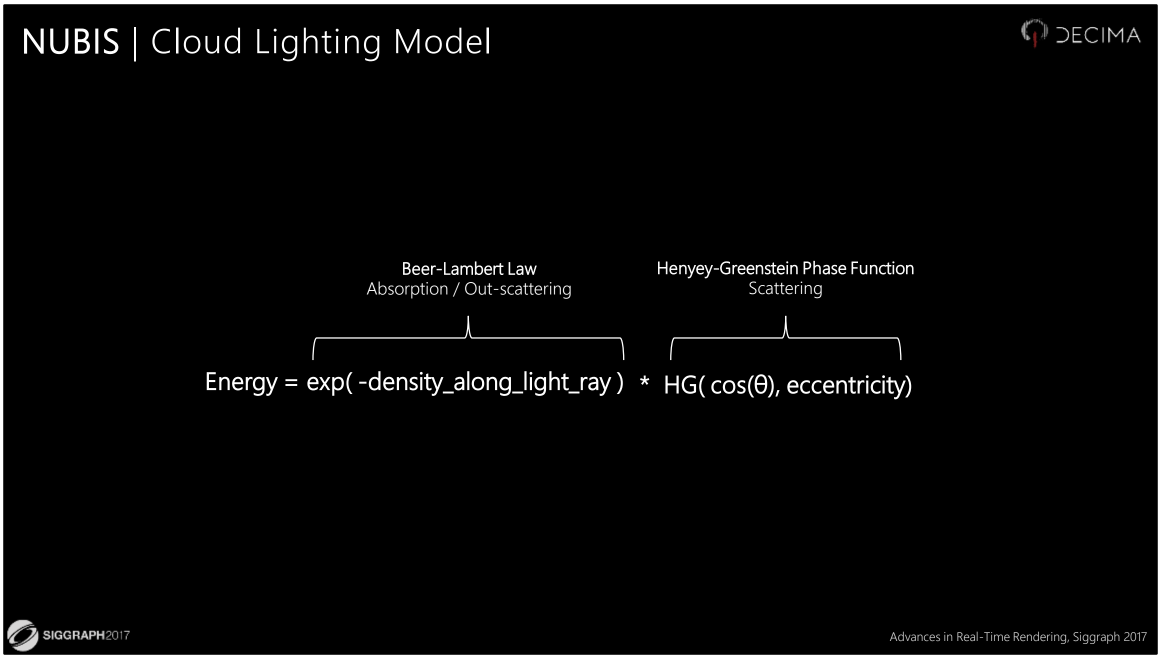
Advances in Real-Time Rendering, Siggraph 2017

Our Authoring system and density model both directly inform our lighting calculations when we ray-march our cloudscapes.

NUBIS | Cloud Lighting Model

DECIMA
“Clouds are bodies without surface.”
– Leonardo da Vinci
Photograph
SIGGRAPH2017
Advances in Real-Time Rendering, Siggraph 2017

Leonardo, the master of deduction and abstraction, called clouds “Bodies without surface”. This is a fact. When you look at a cloud you are peering into it’s interior. This makes shading volumetrics like clouds more complicated than shading a surface.



NUBIS | Cloud Lighting Model

DECIMA

$$\text{Energy} = \overbrace{\text{exp(-density\_along\_light\_ray)}}^{\substack{\text{Beer-Lambert Law} \\ \text{Absorption / Out-scattering}}} \; * \; \overbrace{\text{HG(cos(}\theta\text{), eccentricity)}}^{\substack{\text{Henyey-Greenstein Phase Function} \\ \text{Scattering}}}$$

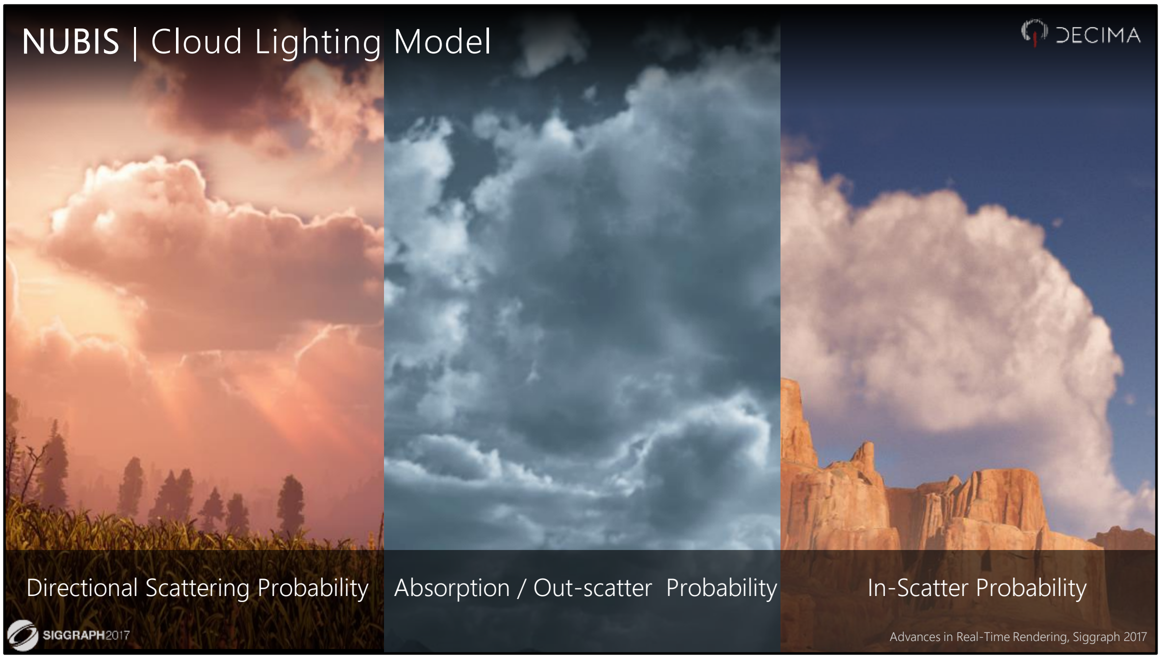
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

For performance reasons, the way you normally calculate Radiance or the light energy at a point in a volume is to combine the

- Beer-Lambert law, which simulates Transmittance and accounts for out-scattering and absorption with the
- Henyey-Greenstein Phase Function, which simulates the directional effects of light scattering. This is what's known as a single scattering lighting model.

While this works for optically thin media like fog, it doesn't work so well for thick clouds because it fails to account for light that has scattered in, or in-scattered, to the sample point after bouncing off of hypothetical water molecules in the cloud. So, we propose a new model specifically for cheaply lighting thick volumetric clouds in a way that is more realistic.



NUBIS | Cloud Lighting Model

Directional Scattering Probability

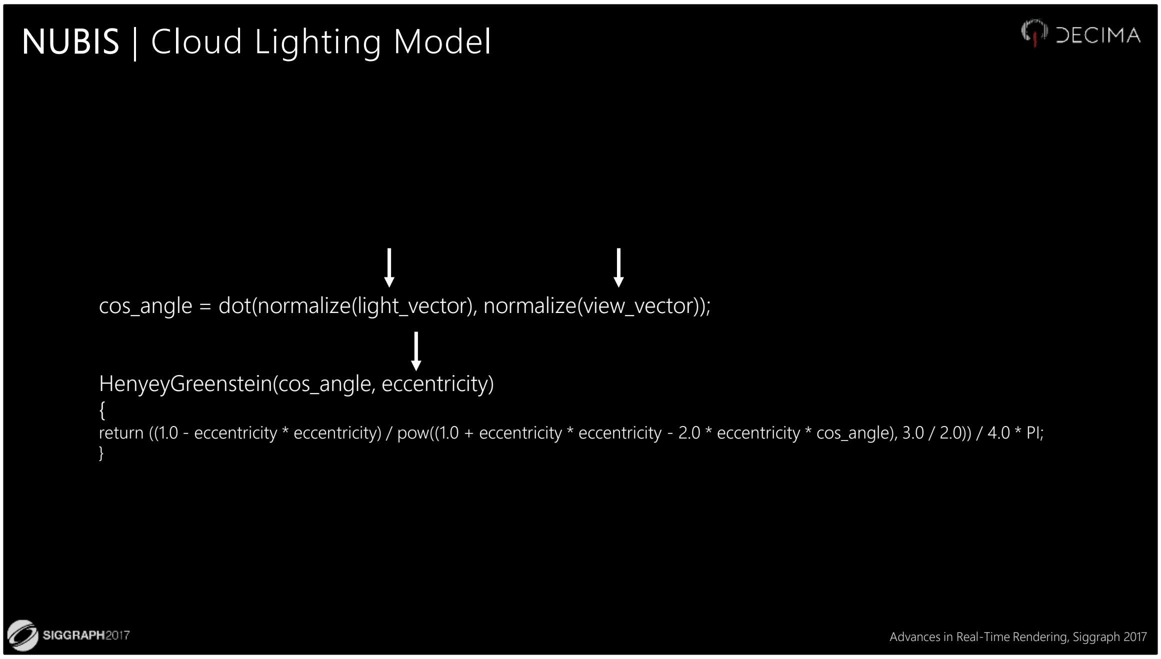
Absorption / Out-scatter Probability

In-Scatter Probability

Our Lighting model is an attenuation model, meaning that we start with the full light intensity and only do work to reduce it. It is a combination of 3 probabilities:

Directional scattering probability, which gives us finer control over the silver lining effect in clouds.
Absorption / out-scatter probability which also accounts for in-scattering.
And in-scattering probability, which accounts for the dark edges and bases to clouds.

First, lets look at directional scattering.



NUBIS | Cloud Lighting Model

DECIMA

```
cos_angle = dot(normalize(light_vector), normalize(view_vector));

HenyeyGreenstein(cos_angle, eccentricity)
{
return ((1.0 - eccentricity * eccentricity) / pow((1.0 + eccentricity * eccentricity - 2.0 * eccentricity * cos_angle), 3.0 / 2.0)) / 4.0 * PI;
}
```

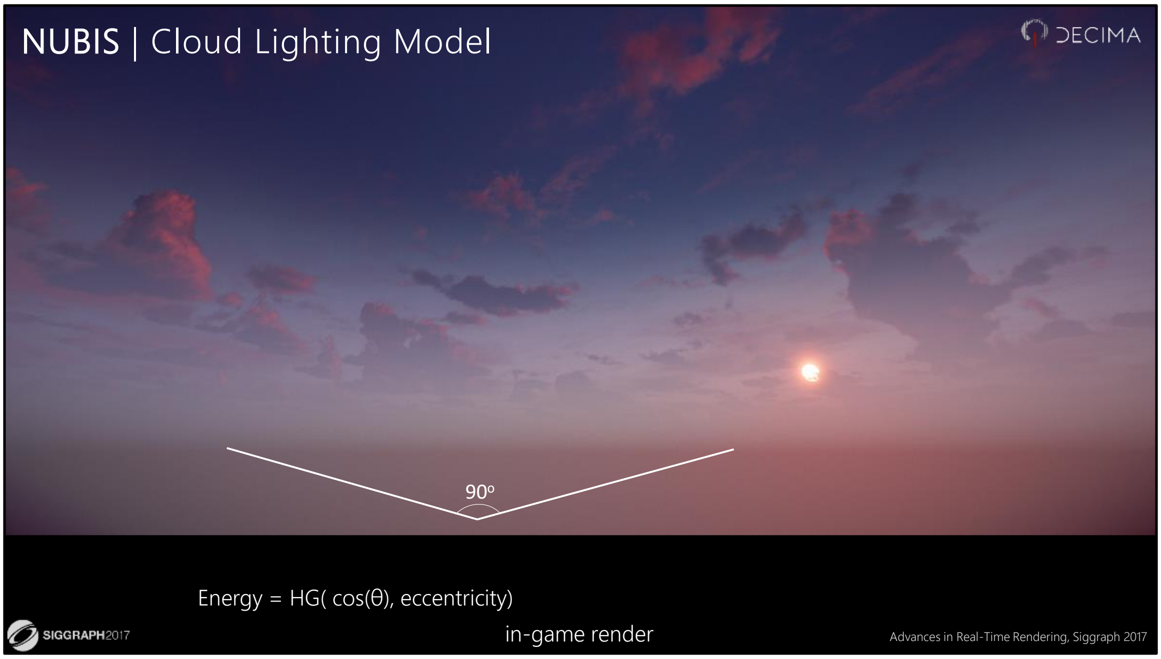
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

This is the standard implementation of the Henyey-Greenstein function. You calculate the cosine of an angle using the

- light vector and the
- view vector and supply it along with a
- directional scattering bias to the function, which makes the light scatter forward or backward. You then apply the result wherever you calculate the radiance of your sample.

There is one limitation with this though.

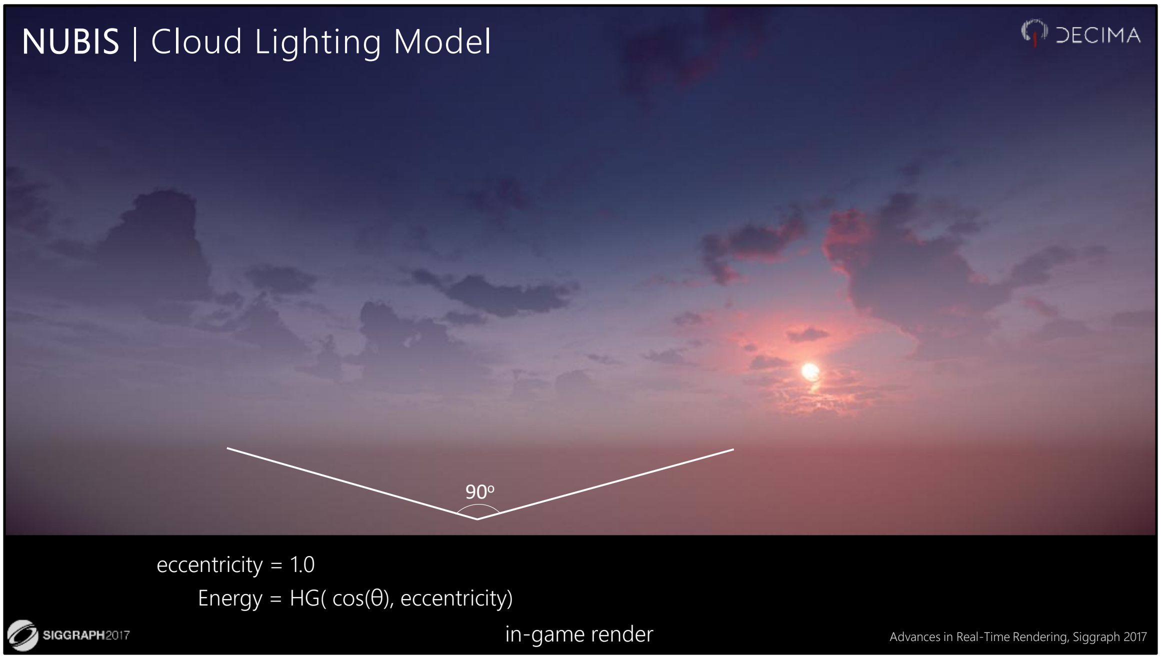


NUBIS | Cloud Lighting Model

Energy = HG(cos(θ), eccentricity)

in-game render

The Eccentricity value that worked well for mid-day failed to provide the bright highlights around the sun that we needed at sunset. If we stepped into the wilderness of artistic license and changed the eccentricity so that it was …



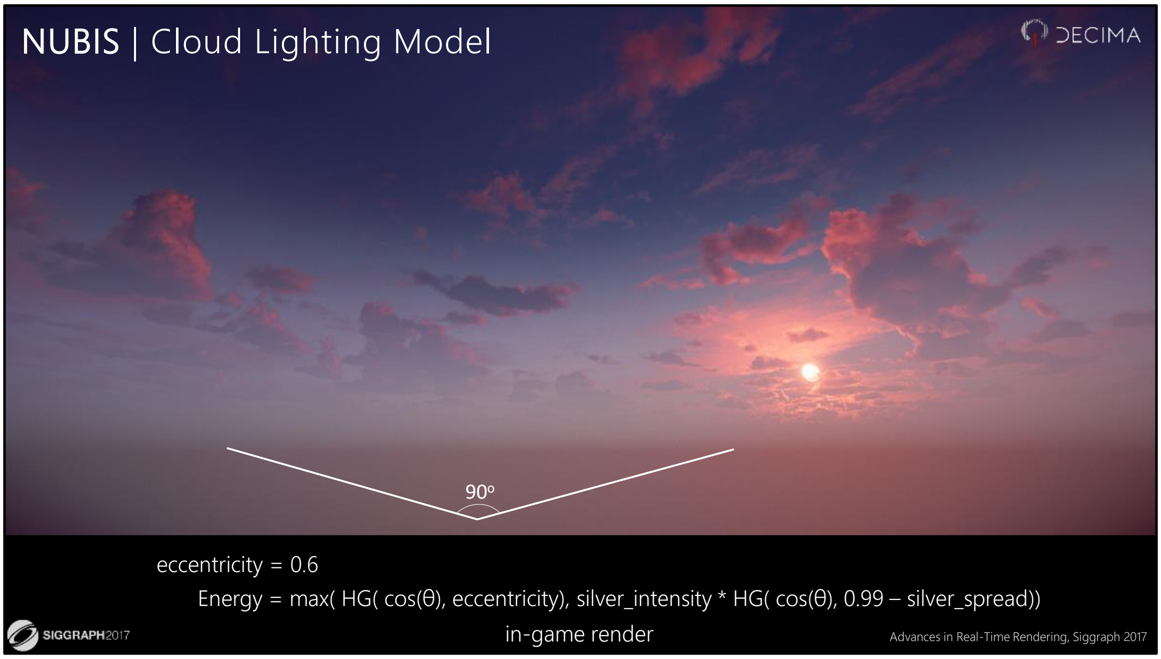
NUBIS | Cloud Lighting Model

eccentricity = 1.0

Energy = HG(cos(θ), eccentricity)

in-game render

more forward scattering, we got the highlights that we needed near the sun, but the clouds 90 degrees away from the sun became too dark. In order to retain the baseline forward scattering behavior and get the silver lining highlights that we needed,



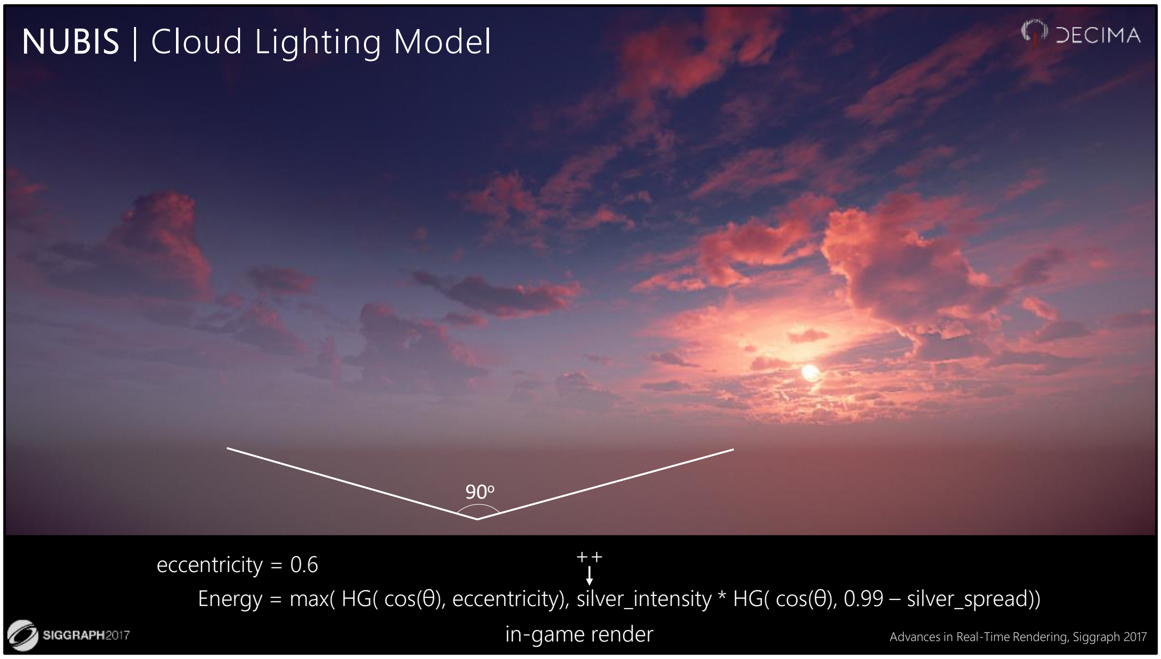
NUBIS | Cloud Lighting Model

eccentricity = 0.6

Energy = max(HG(cos(θ), eccentricity), silver_intensity * HG(cos(θ), 0.99 – silver_spread))

in-game render

we combined two Henyey-Greenstein phase functions using a max() operation. Now, since we had already taken artistic license, why stop there? We added an intensity control for this second phase function call that allowed us to



NUBIS | Cloud Lighting Model

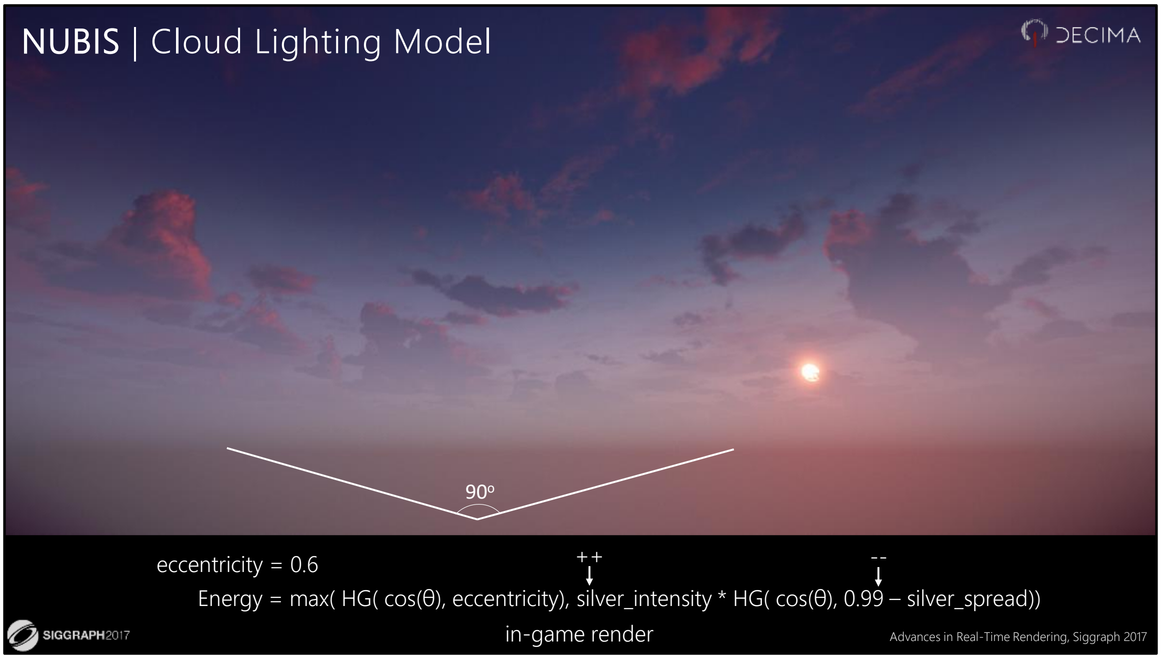
eccentricity = 0.6

++

Energy = max(HG(cos(θ), eccentricity), silver_intensity * HG(cos(θ), 0.99 – silver_spread))

in-game render

independently control the intensity of this effect



NUBIS | Cloud Lighting Model

eccentricity = 0.6

++ --

Energy = max(HG(cos(θ), eccentricity), silver_intensity * HG(cos(θ), 0.99 – silver_spread))

in-game render

and its spread away from the sun.
It may not be particularly Kosher, in fact I’m sure someone in the audience is pulling their hair out right now. To that I would say first, enjoy the fact that you have hair!
Second, We don’t like bandaids like this so this is also an area that we are continuing to investigate.

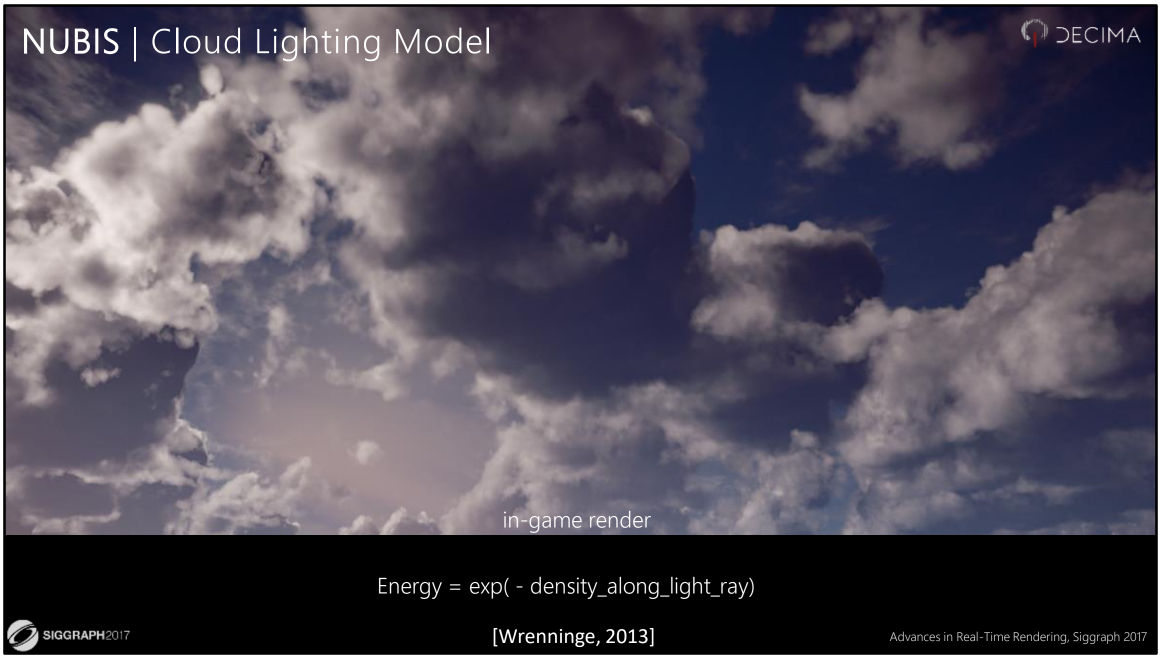
NUBIS | Cloud Lighting Model

[video]

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

Here’s an example of how this looked during the time of day transition.



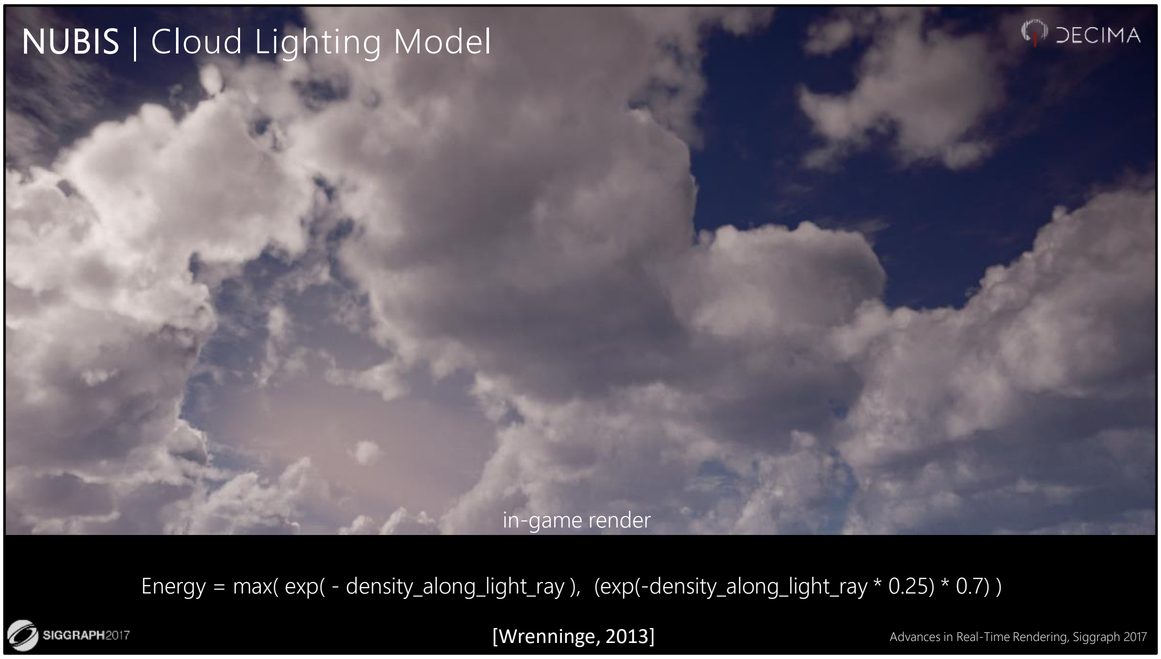
NUBIS | Cloud Lighting Model

in-game render

Energy = exp(- density_along_light_ray)

[Wrenninge, 2013]

As mentioned before, the beer-lambert law only accounts for attenuation of light and not emission from light that has in-scattered to the sample point. This makes clouds too dark as in the image above. Our solution was similar to Magnus Wrenninge's Multiple Scattering approximation.



NUBIS | Cloud Lighting Model

in-game render

Energy = max(exp(- density_along_light_ray), (exp(-density_along_light_ray * 0.25) * 0.7))

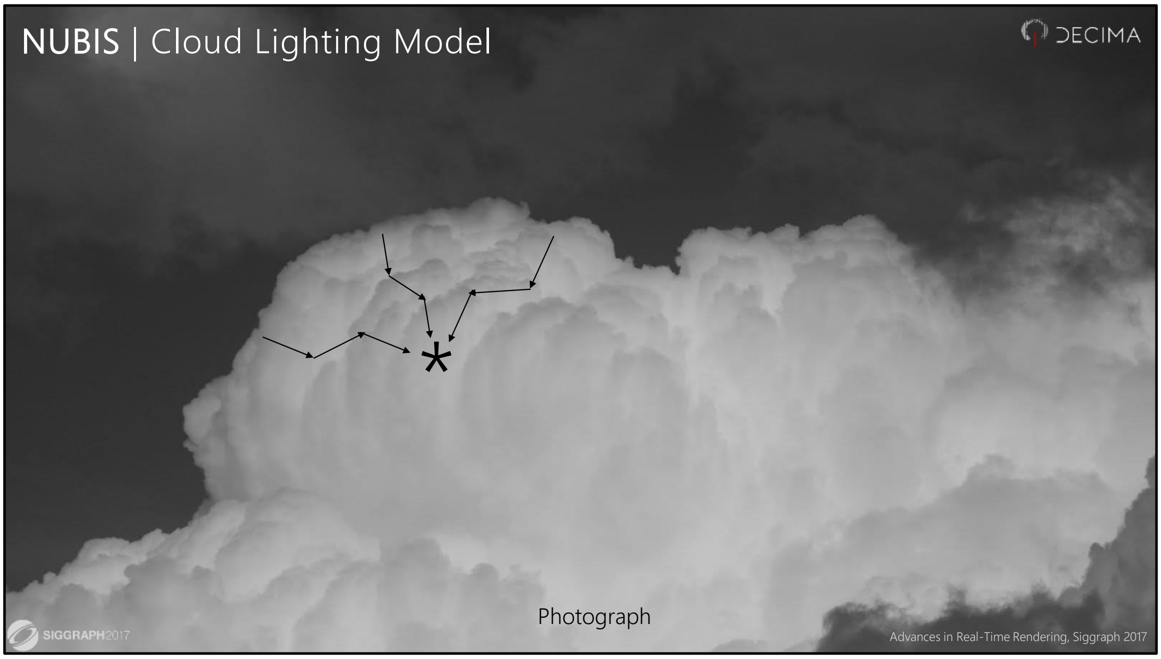
[Wrenninge, 2013]

SIGGRAPH2017

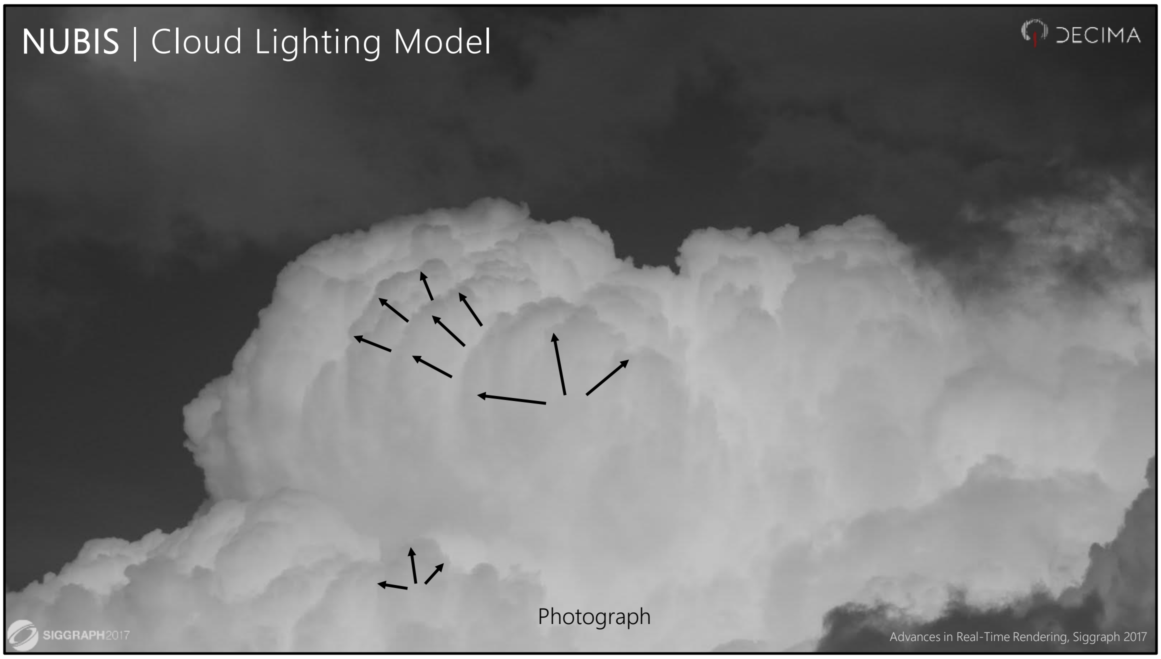
Advances in Real-Time Rendering, Siggraph 2017

We combine two beer-lambert functions using a max operation. The attenuation value for the second function was reduced to push light further into the cloud. However, we reduced its influence so that we didn’t overpower the result. Additionally, to make sure this only happens when we look away from the sun, we ramp down this affect as the angle between the view ray and the light ray decrease. You can find a code example of that in the slides.

There is a link at the end of this presentation to the solution that Seb Hillaire offered in 2016, which is worth considering as well.


NUBIS | Cloud Lighting Model
DECIMA
*
Photograph
SIGGRAPH2017
Advances in Real-Time Rendering, Siggraph 2017

In-scattering or rather, the lack of it, also creates another lighting effect associated with clouds. In-scattering is when a light ray that has scattered in a cloud is combined with others on its way to your eye, effectively brightening the region of the cloud you are looking at.
In order for this to happen, you need to be looking at an area that has lots of rays scattering into it. Scattering only occurs where there is cloud material. So, it follows that the deeper you are in a cloud, the more scattering contributors you will have. Recall the Leonardo quote. We can actually see into the regions where a lot of scattering happens.



NUBIS | Cloud Lighting Model

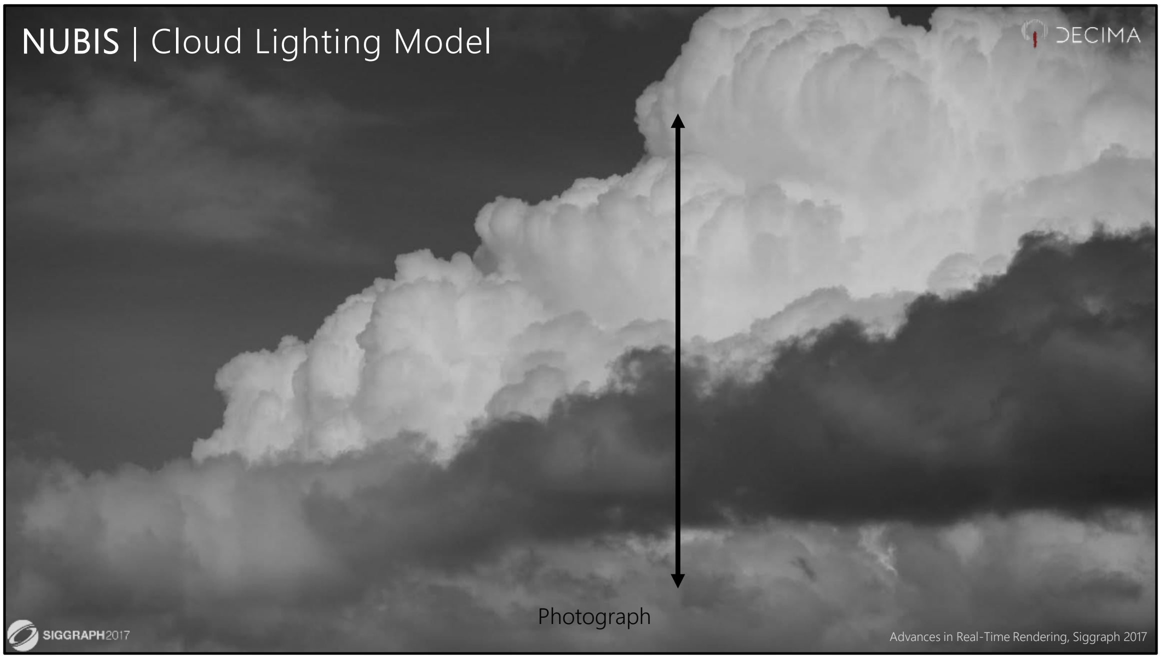
DECIMA

Photograph

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

So the amount of in-scattering on the edges of clouds is lower, which makes them appear dark. This is especially noticeable when looking at the sun facing sides of clouds and this has to do with the directional scattering effect I mentioned earlier. In 2015 we made a function to approximate this effect and called it the powder sugar function. We have since improved this to account for the fact that the effect is not purely directional.

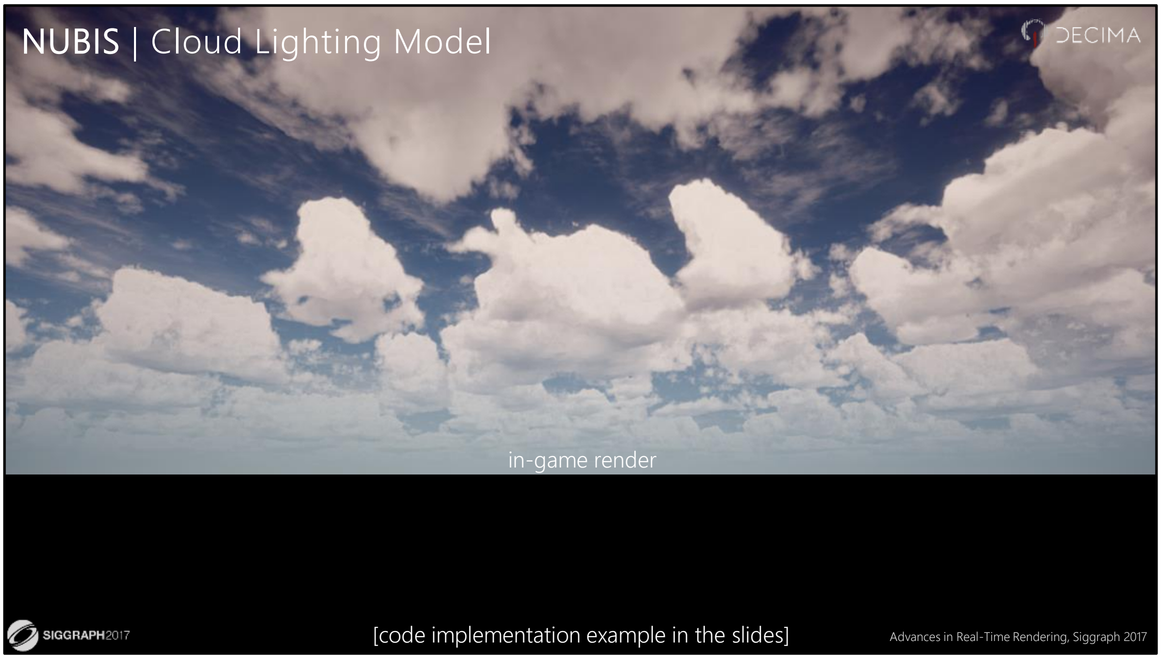


NUBIS | Cloud Lighting Model

Photograph

Advances in Real-Time Rendering, Siggraph 2017

Additionally, we can assume, that because there are no strong scattering sources below clouds, the bottoms will have fewer occurrences of in-scattering as well,.
We represent both of the probabilities in a two term function called the In-Scatter Probability Function.
Lets take a look.



NUBIS | Cloud Lighting Model

DECIMA

in-game render

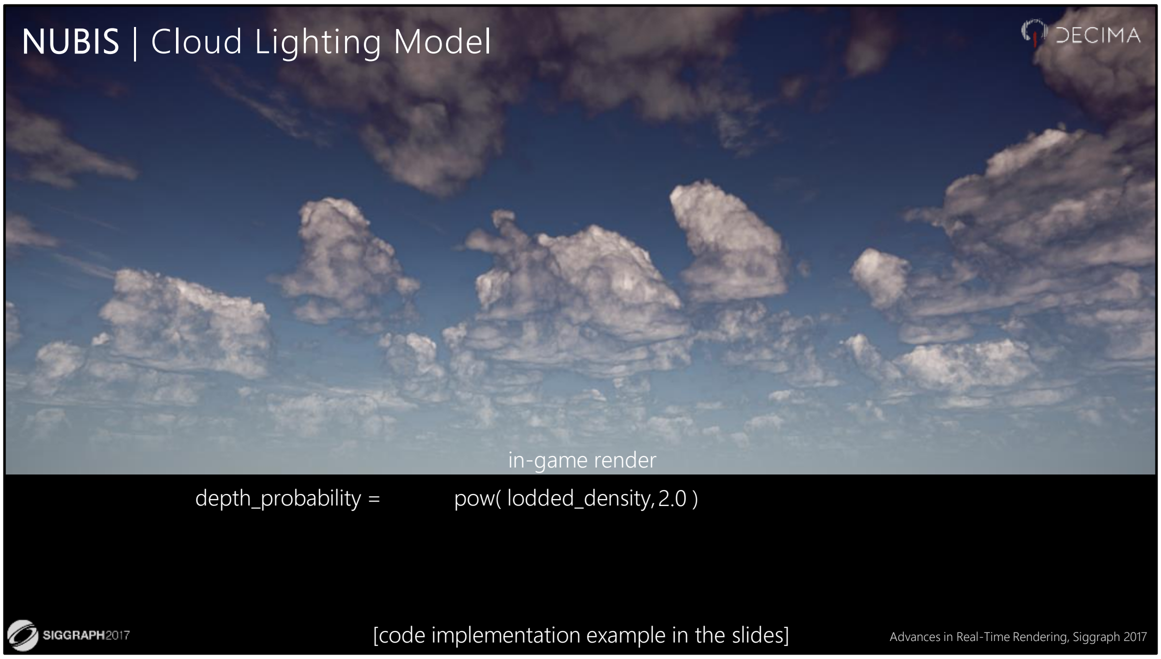
SIGGRAPH2017

[code implementation example in the slides]

Advances in Real-Time Rendering, Siggraph 2017

What you see here is the result of our lighting model with JUST the attenuation and phase components.

Now lets account for in-scattering probability.. First, we need to know how much density exists around our sample position to have an idea of how much light could potentially scatter into the point.



NUBIS | Cloud Lighting Model

in-game render

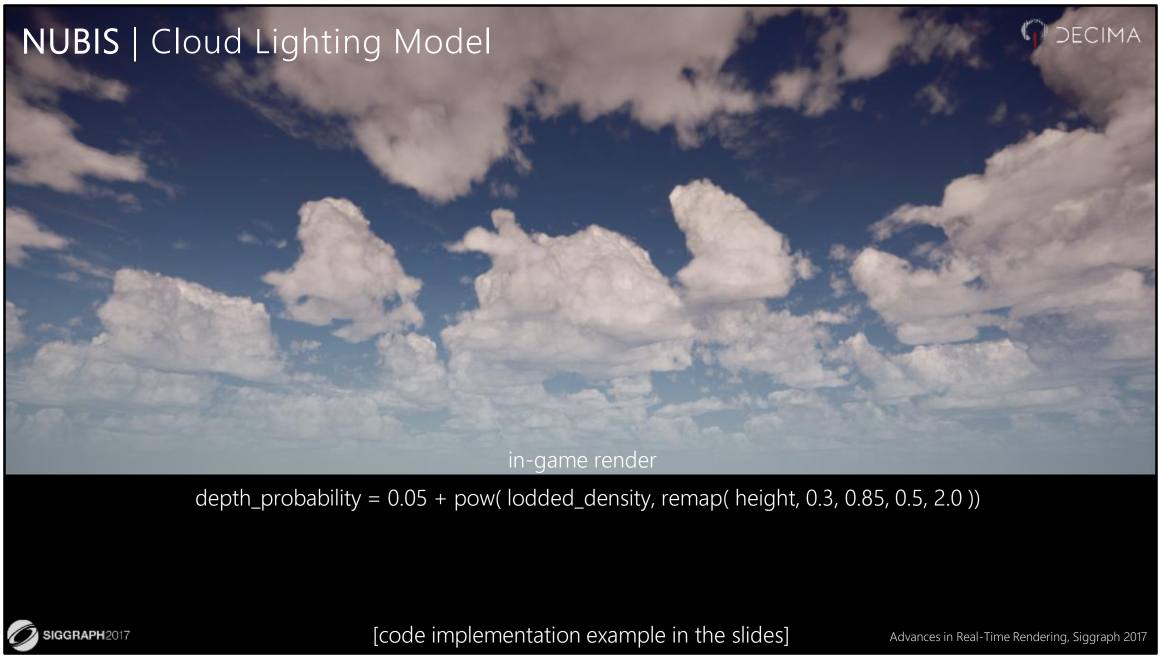
```
depth_probability =           pow( lodded_density, 2.0 )
```

SIGGRAPH2017

[code implementation example in the slides]

Advances in Real-Time Rendering, Siggraph 2017

We do this by sampling our cloud at a low LOD level and then squaring the result to account for attenuation along the in-scatter path. However, you can see that it is too dark. This is because we are now saying that there is little to no in-scattering on the edges of clouds. Which of course, is incorrect.



NUBIS | Cloud Lighting Model

DECIMA

in-game render

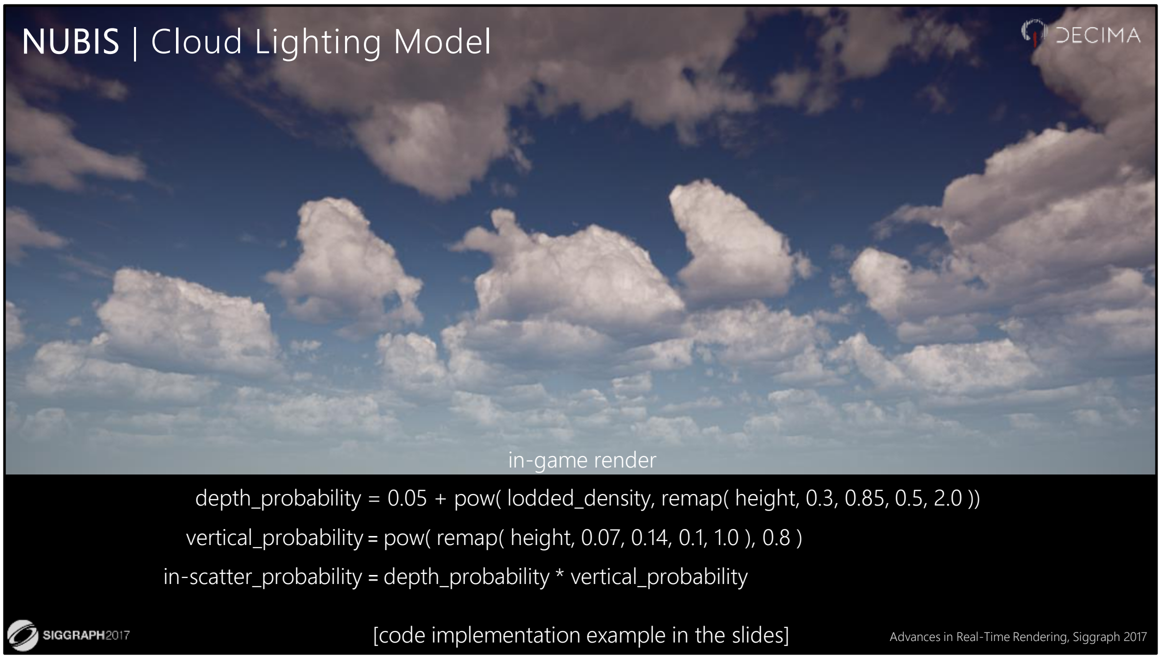
depth_probability = 0.05 + pow(lodded_density, remap(height, 0.3, 0.85, 0.5, 2.0))

[code implementation example in the slides]

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

So, we also relax this effect over altitude and apply a small bias to compensate for this.
In the slides you will see an example of how to ensure that this effect retains some directionality. This is an improvement over the purely directional result from the Powder Sugar function we offered in 2015.



NUBIS | Cloud Lighting Model

in-game render

```
    depth_probability = 0.05 + pow( lodded_density, remap( height, 0.3, 0.85, 0.5, 2.0 ))
  vertical_probability = pow( remap( height, 0.07, 0.14, 0.1, 1.0 ), 0.8 )
in-scatter_probability = depth_probability * vertical_probability
```

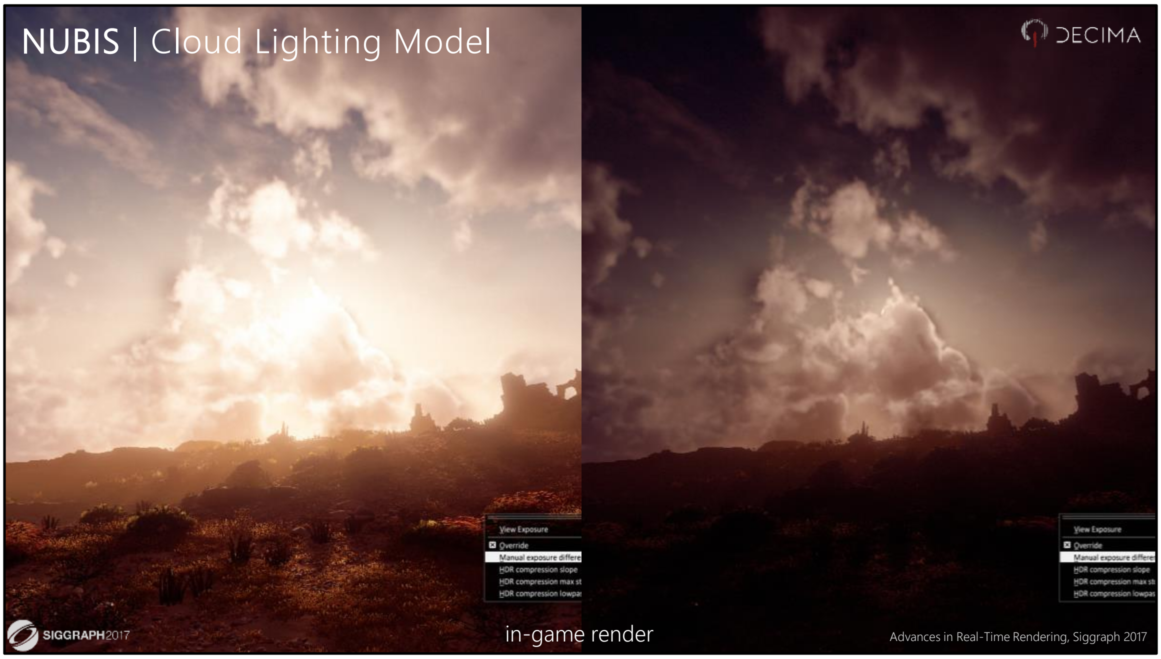
[code implementation example in the slides]

SIGGRAPH2017 Advances in Real-Time Rendering, Siggraph 2017

The second component accounts for The decrease in in-scattering over height.
We represent this with a gradient. The range of this gradient depends, of course, on how you define your height gradients in the density model.

We multiply both components to represent in-scattering probability.

Again, there will be full code examples for the lighting model in the slides.

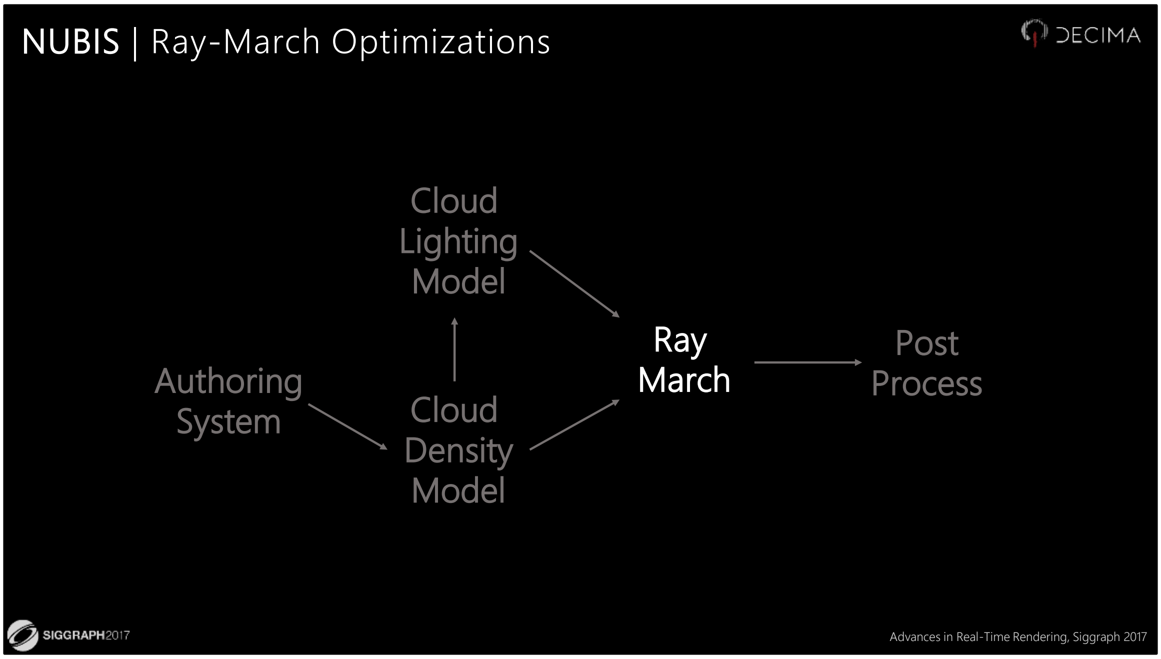


NUBIS | Cloud Lighting Model

in-game render

Advances in Real-Time Rendering, Siggraph 2017

Additionally, one side note, we do not clamp any values in our lighting model. This allows support for HDR, which is a big thing for the PS4.



NUBIS | Ray-March Optimizations

DECIMA

Cloud Lighting Model

Authoring System

Cloud Density Model

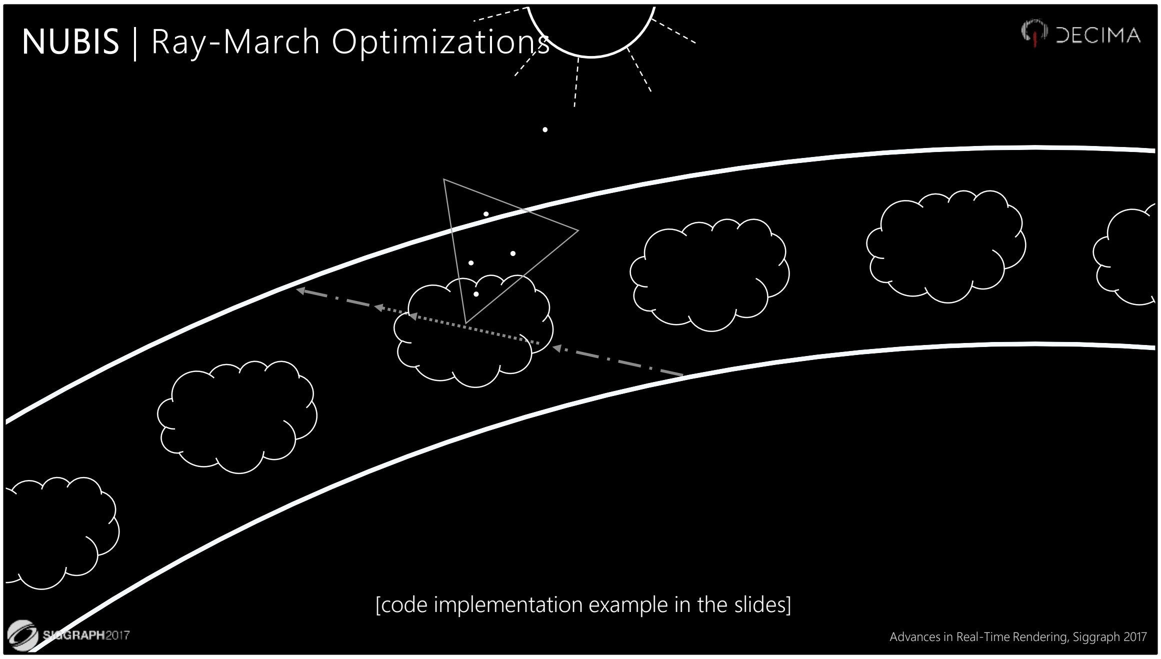
Ray March

Post Process

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

The Lighting model and density model are actually functions which get called inside of the ray-march. I'm going to explain our Ray march at a high level and offer a code examples in the slides so that we can focus on how the optimizations that we made over a standard ray-march improved performance.



NUBIS | Ray-March Optimizations

[code implementation example in the slides]

Our ray march takes place in a spherical layer of atmosphere. The step count ranges between 54 and 96 samples depending on if the view ray is pointing up or to the horizon.

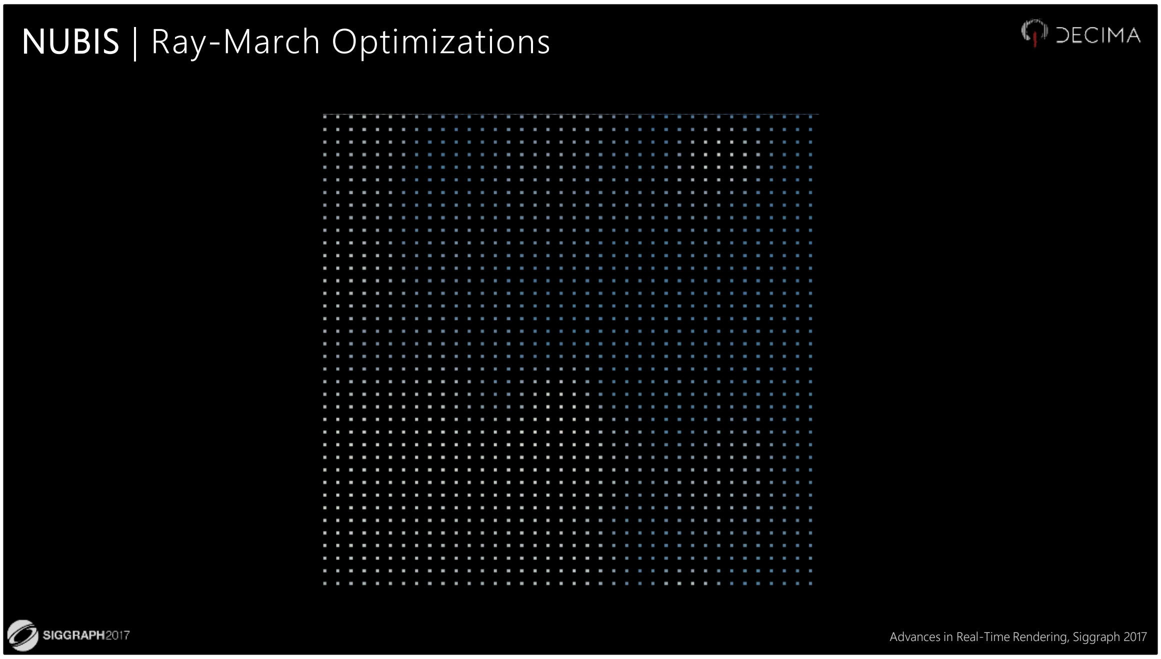
In the beginning of our ray-march we take large steps through the volume and sample only the low frequency noises at a low LOD level with our density sampler. We can get away with this because the high frequency cloud noises are applied as a subtraction to the edges of the low frequency noise.

We then take short steps and sample using all of the frequencies of noise and related instructions.

Once we have taken 10 samples in this manner which return zero density,

we switch back to large steps and cheap samples until we reach an alpha value of 1 along the view ray or we reach the end of our spherical ray-march volume.

At each expensive sample, when the density is nonzero, we take 5 density samples in a light aligned cone to use in our lighting calculation, decreasing the LOD level for each sample. The cone sample and the decreasing LOD level have the effect of smoothing out the artifacts that you would normally get from taking only 5 light samples to light a dense volume.



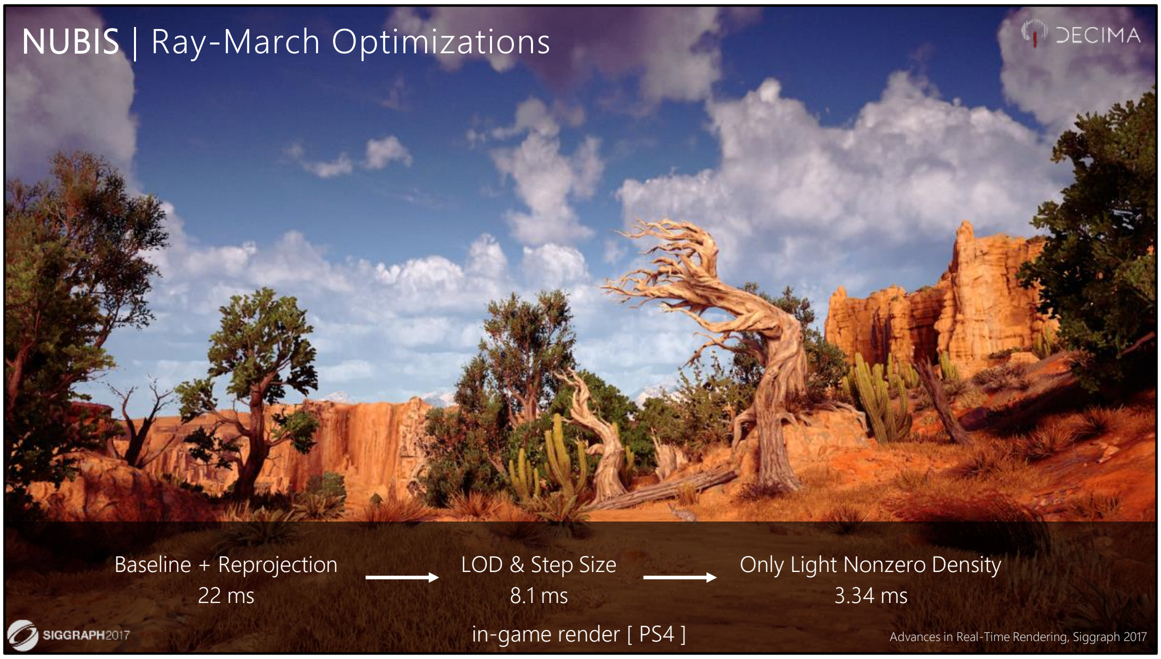
NUBIS | Ray-March Optimizations

DECIMA

SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

The reason we are able to render our cloudscapes in 2ms is re-projection. You can look at the 2015 course for more information on this. One thing to keep in mind is that your cloud ray march must be fast enough so that all of the threads can produce pixels fast enough to prevent artifacts.



NUBIS | Ray-March Optimizations

DECIMA

Baseline + Reprojection
22 ms

→

LOD & Step Size
8.1 ms

→

Only Light Nonzero Density
3.34 ms

SIGGRAPH2017

in-game render [PS4]

Advances in Real-Time Rendering, Siggraph 2017

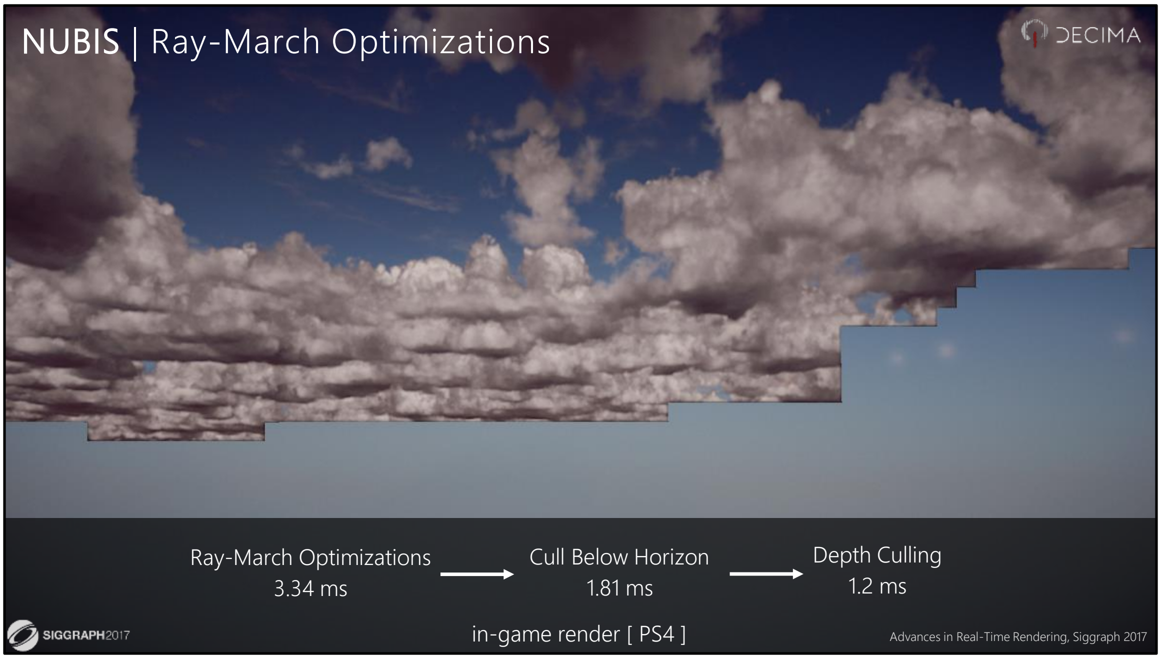
As for the ray march optimizations, lets look at a typical scenario in our game. Clouds are partially obscured by the landscape and take up roughly half of the screen. Without any optimizations except temporal reprojection, Our shader takes 22ms to draw on the Standard PS4 Hardware.

As I add optimizations, the image will update. But as you will notice, there’s no apparent difference in the clouds with each optimization.

When we introduce our Adaptive step size and LODing algorithm that number drops to 8.1

When we are careful to only take light samples when we are inside of a cloud, that number drops to 3.34 ms
In addition, we also made some optimizations before the ray-march even started.

Lets exclude the geometry so that we can see the entire cloud layer.



NUBIS | Ray-March Optimizations

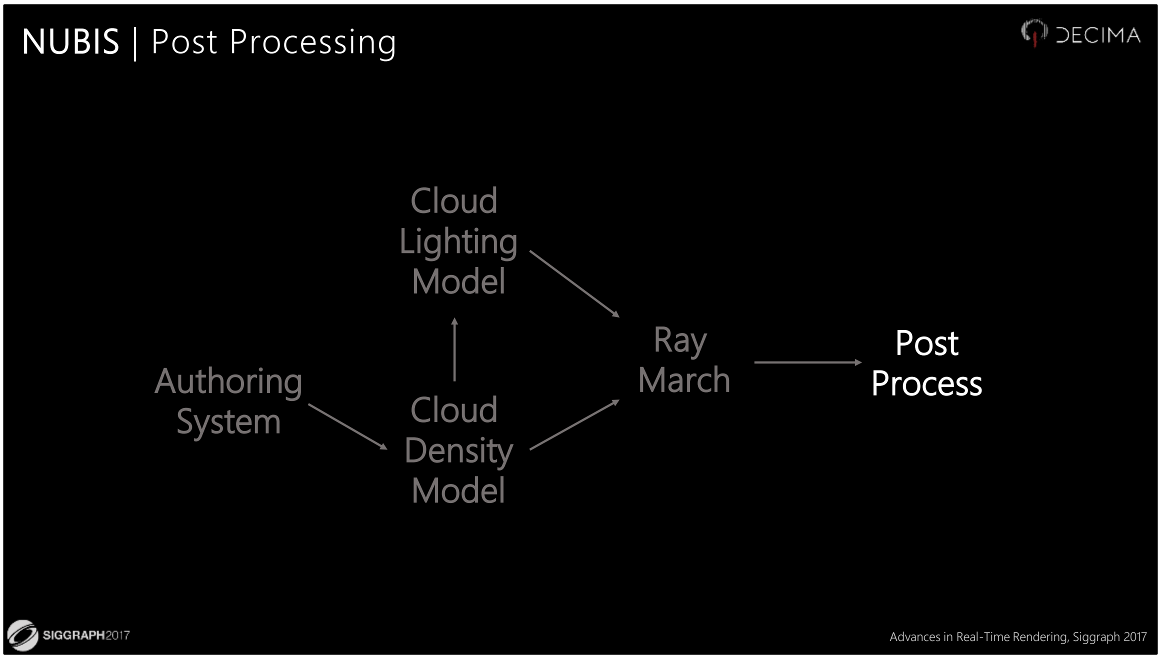
Ray-March Optimizations 3.34 ms → Cull Below Horizon 1.81 ms → Depth Culling 1.2 ms

in-game render [PS4]

What we are looking at now is the raw result of the ray march and re-projection. You can see that the cloud sphere drops below the horizon and begins to tile noticeably.

Since we don’t see anything below the vanishing point of our spherical cloud volume we can exclude the section in that area. This brings the render time down to 1.81 ms.

Finally, we only start the ray march in areas of the frame where the sky is or has the potential to be un-occluded in the next frame.
However, thin occluders like trees could move across frame quickly enough to introduce reprojection artifacts. So, to prevent this we take a max() of a low lod sample of our depth channel. This ensures that we are conservative enough to prevent artifacts while reducing the number of ray-march steps. This brings the render time down to 1.2 ms



NUBIS | Post Processing

Cloud Lighting Model

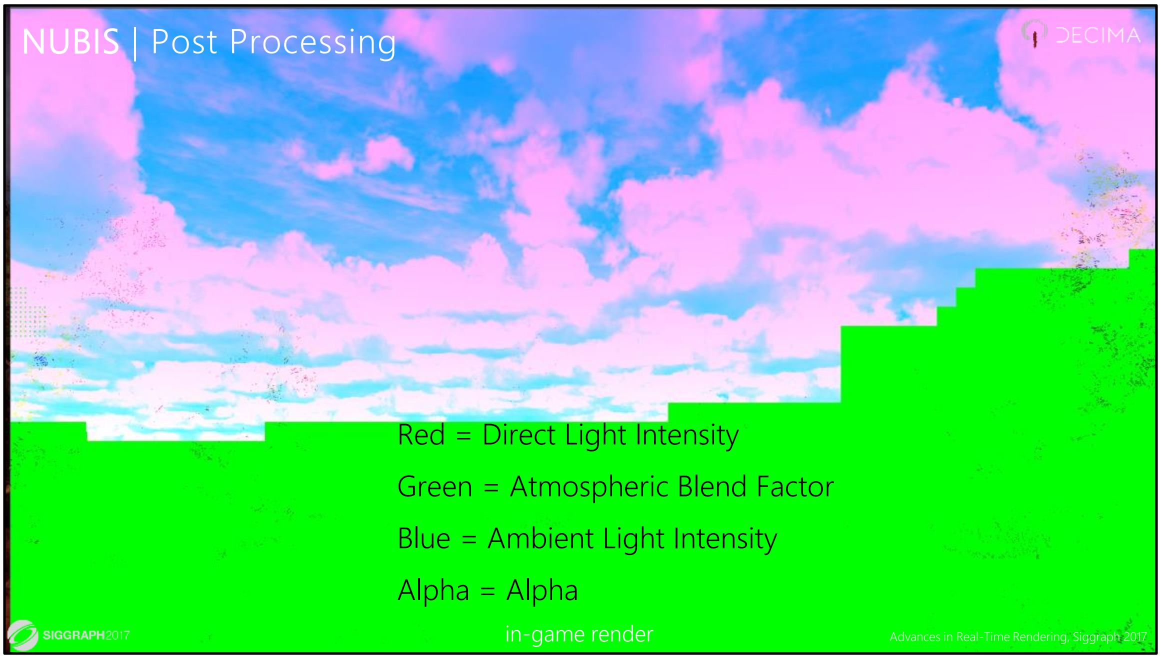
Authoring System

Cloud Density Model

Ray March

Post Process

After we complete the ray march, we integrate its result into the frame using a post process shader.



NUBIS | Post Processing

Red = Direct Light Intensity

Green = Atmospheric Blend Factor

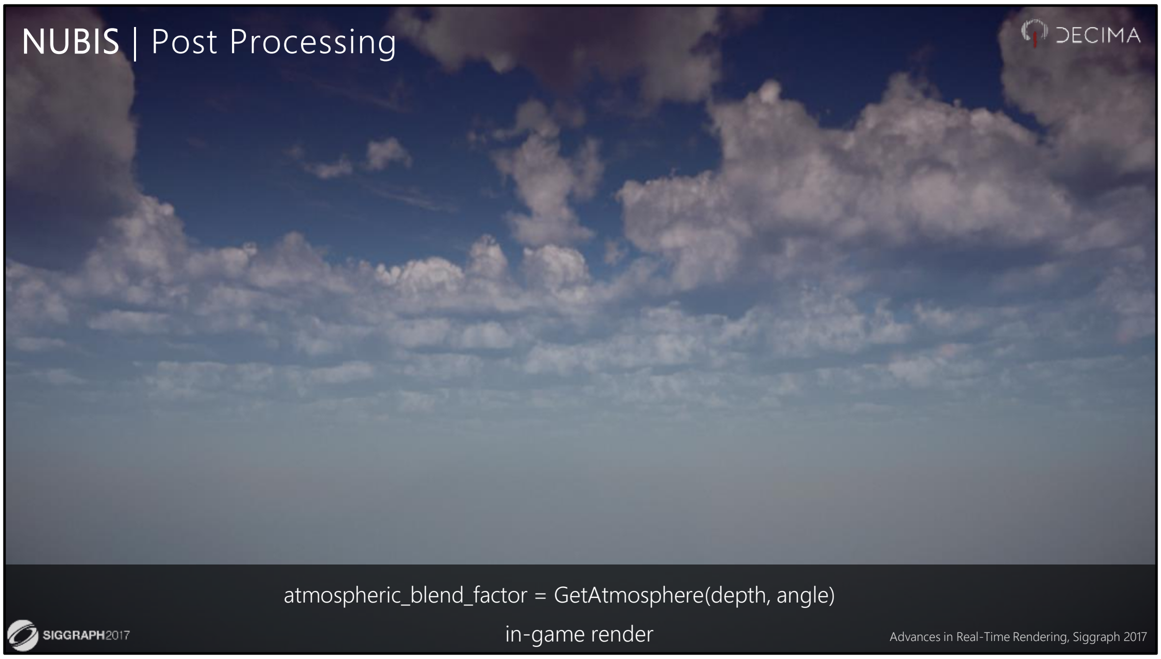
Blue = Ambient Light Intensity

Alpha = Alpha

in-game render

Back to our typical case…

- This is what comes out of our ray-march in the form of an RGBA buffer. Light intensity, Atmospheric haze blend factor, ambient light contribution and the alpha channel.
- We have already discussed how we calculate our direct light intensity, but the atmospheric blend factor…



NUBIS | Post Processing

atmospheric_blend_factor = GetAtmosphere(depth, angle)

in-game render

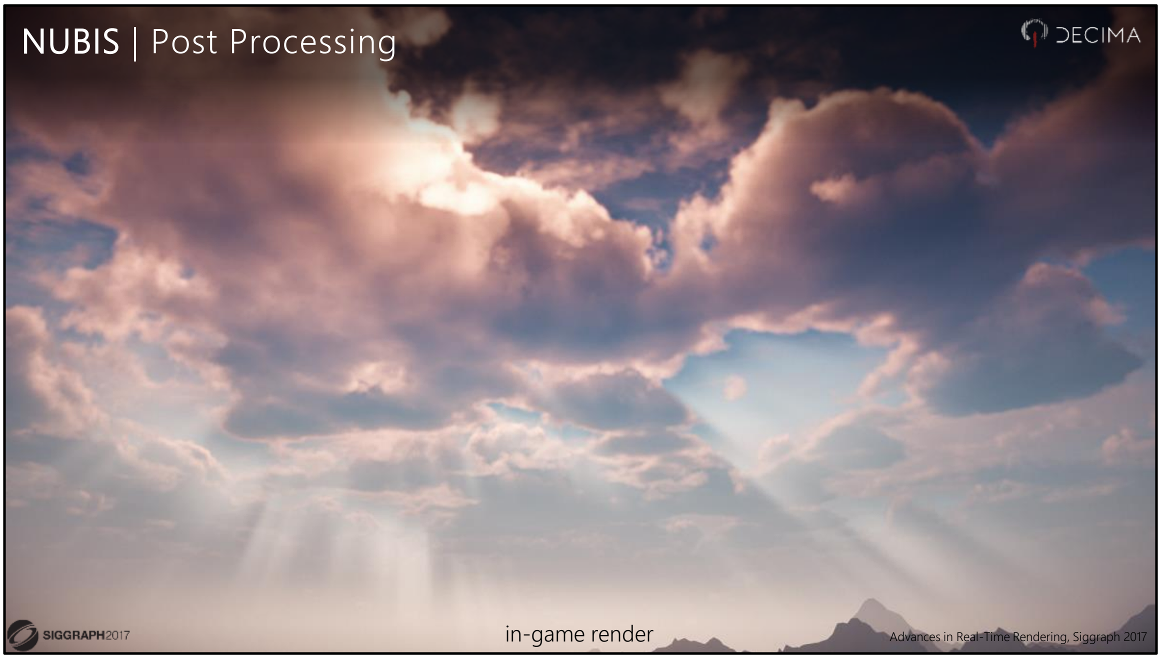
Is calculated in two steps.

- First, during the ray march, we create a depth value when the alpha channel has reached a value of .5.

Next, after the ray-march we sample the accumulated atmospheric scattering contribution at this depth.

We then use this value, which is stored in the Green channel of the cloud buffer,

- as a blend factor between the color
- of the clouds and the color of the sky.



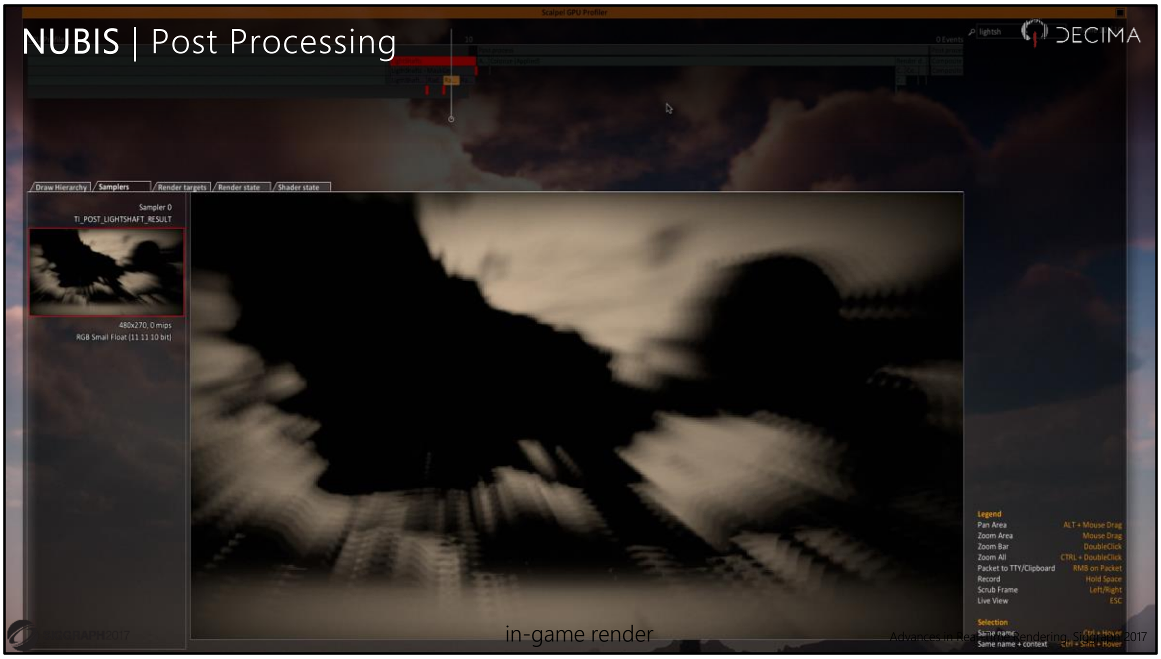
NUBIS | Post Processing

in-game render

Advances in Real-Time Rendering, Siggraph 2017

To create the shafts of light that break through the clouds so dramatically in Bierstadt paintings, we referred to the work of Kenny Mitchell entitled, “Volumetric light scattering as a post process.”

NUBIS | Post Processing



Sampler 0
TI_POST_LIGHTSHAFT_RESULT
480x270, 0 mips
RGB Small Float (11 11 10 bit)

Legend

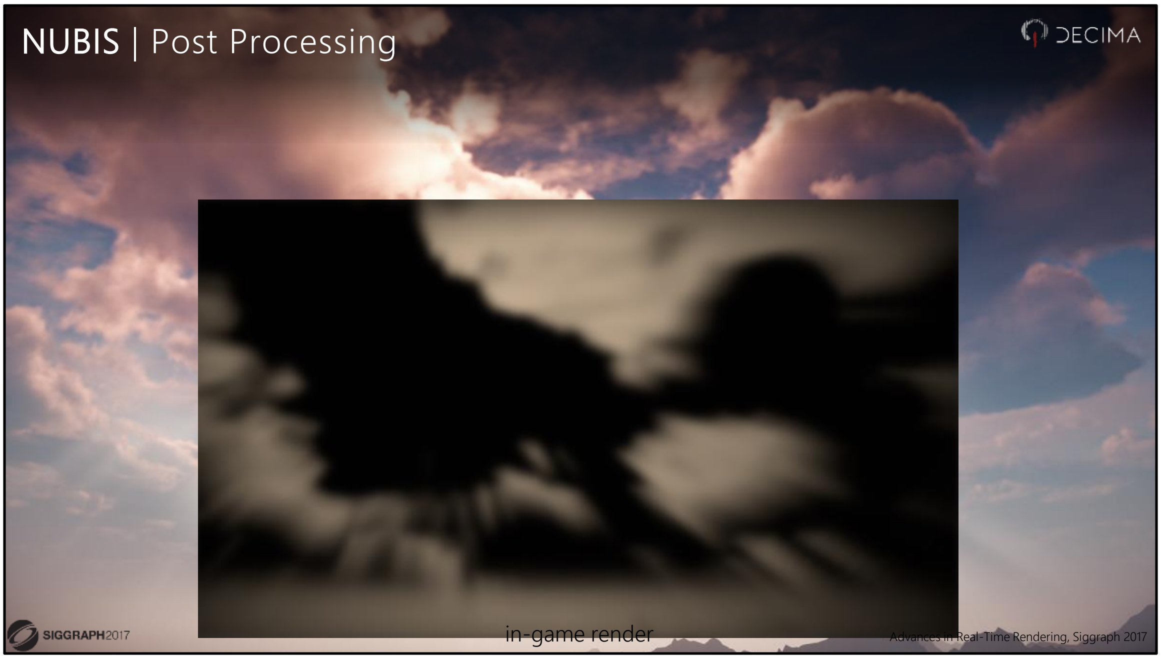
Pan Area	ALT + Mouse Drag
Zoom Area	Mouse Drag
Zoom Bar	DoubleClick
Zoom All	CTRL + DoubleClick
Packet to TTY/Clipboard	RMB on Packet
Record	Hold Space
Scrub Frame	Left/Right
Live View	ESC

Selection

Same name	Ctrl + Hover
Same name + context	Ctrl + Shift + Hover

in-game render

The fist step is to render a bright highlight around the sun where it is not occluded by clouds. Then we offset this highlight radially away from the sun. Next we apply a radial blur on this image to create the light shafts intensity buffer.



NUBIS | Post Processing

in-game render

Instead of adding this image directly to the scene like the original article suggests we use this intensity buffer as a mask when we apply the Mie scattering in our atmospheric scattering shader which is only applied a larger distances. This way the color and intensity of the light shafts will match with what you would expect from Mie scattering and the light shafts will not been drawn over close by scenery. Additionally, the light shafts connect perfectly to our other volumetrics.

NUBIS | The Future

in-game render

DECIMA
SIGGRAPH2017

Advances in Real-Time Rendering, Siggraph 2017

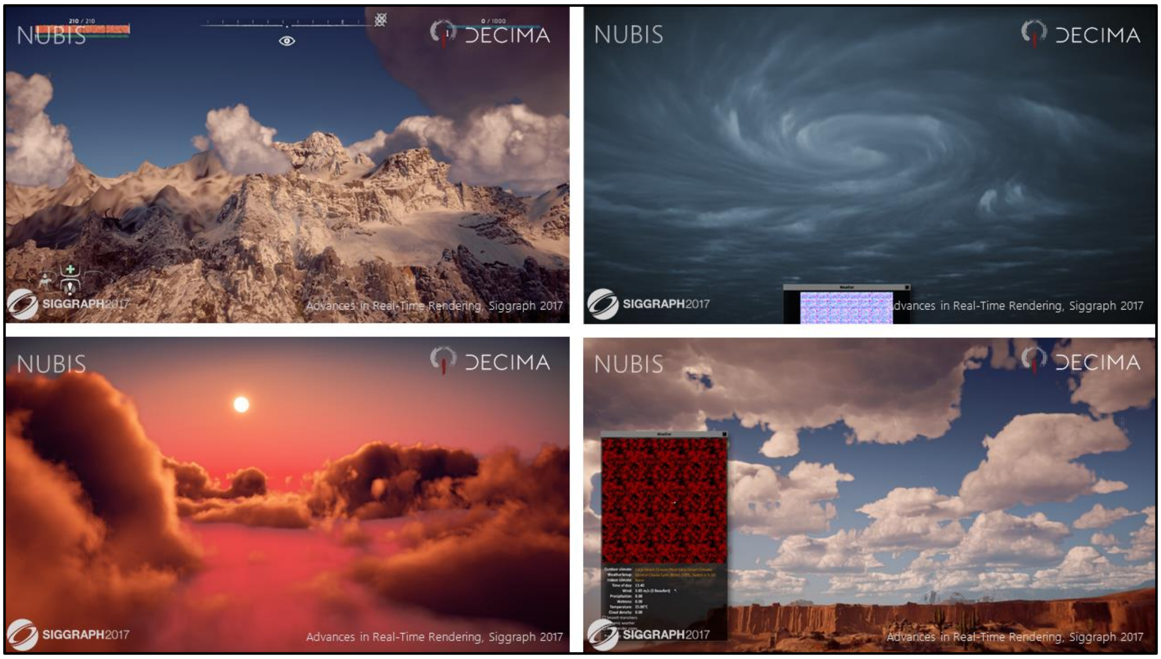
Hopefully what I have shared with you today has piqued your interest in rendering cloudscapes in real-time.

When developing complex systems that mimic reality, we have to be bold like Howard was with his naming system and remember that eventually, on a long enough timeline, all obstacles will become features.

I had a saying while working on all of this:

Without care, you can make a procedural system that is excellent at generating ugly. The first time around, our goal should be to get to 80% not ugly. We feel like we got pretty close to this but there is still much to do.

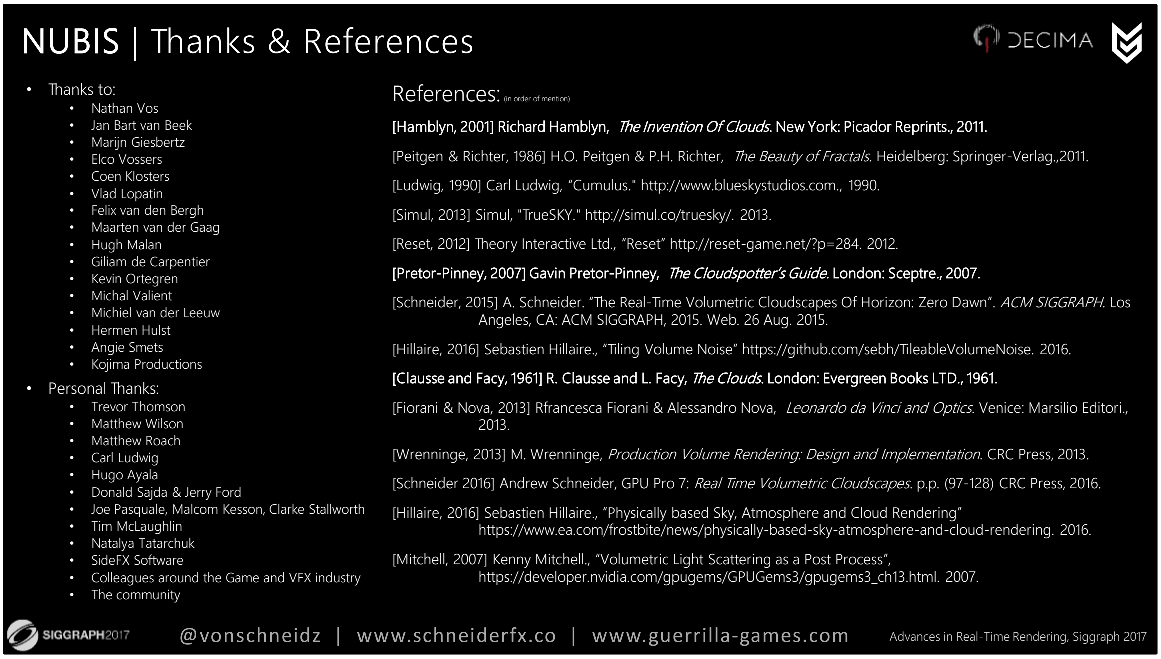
I feel good enough about where we are with a few of our tests to share some early experiments with you. Please remember that these do not reflect our idea of final quality and they ARE JUST experiments.


NUBIS
DECIMA
SIGGRAPH2017
Advances in Real-Time Rendering, Siggraph 2017
NUBIS
DECIMA
SIGGRAPH2017
dvances in Real-Time Rendering, Siggraph 2017
NUBIS
DECIMA
SIGGRAPH2017
Advances in Real-Time Rendering, Siggraph 2017
NUBIS
DECIMA
SIGGRAPH2017
Advances in Real-Time Rendering, Siggraph 2017

So rendering clouds in front of objects, close to the player with more detail, adding more distortion behaviors, and adding the remaining cloud layers… these are the things that really represent the next frontier for our work with Nubis and and we plan to present some of this to you next year.

Rosa &
our kids &
and family.

I want to thank my wife and kids and my family we could be having dinner and a thought would pop into my head while I was looking at the mashed potatoes and I would bolt out of the room to dictate something into my phone. Thank you for tolerating me. Ik hou van jou.



NUBIS | Thanks & References

DECIMA

- Thanks to:
  - Nathan Vos
  - Jan Bart van Beek
  - Marijn Giesbertz
  - Elco Vossers
  - Coen Klosters
  - Vlad Lopatin
  - Felix van den Bergh
  - Maarten van der Gaag
  - Hugh Malan
  - Giliam de Carpentier
  - Kevin Ortegren
  - Michal Valient
  - Michiel van der Leeuw
  - Hermen Hulst
  - Angie Smets
  - Kojima Productions
- Personal Thanks:
  - Trevor Thomson
  - Matthew Wilson
  - Matthew Roach
  - Carl Ludwig
  - Hugo Ayala
  - Donald Sajda & Jerry Ford
  - Joe Pasquale, Malcom Kesson, Clarke Stallworth
  - Tim McLaughlin
  - Natalya Tatarchuk
  - SideFX Software
  - Colleagues around the Game and VFX industry
  - The community

References: (in order of mention)

[Hamblyn, 2001] Richard Hamblyn, *The Invention Of Clouds*. New York: Picador Reprints., 2011.

[Peitgen & Richter, 1986] H.O. Peitgen & P.H. Richter, *The Beauty of Fractals*. Heidelberg: Springer-Verlag.,2011.

[Ludwig, 1990] Carl Ludwig, "Cumulus." http://www.blueskystudios.com., 1990.

[Simul, 2013] Simul, "TrueSKY." http://simul.co/truesky/. 2013.

[Reset, 2012] Theory Interactive Ltd., "Reset" http://reset-game.net/?p=284. 2012.

[Pretor-Pinney, 2007] Gavin Pretor-Pinney, *The Cloudspotter's Guide*. London: Sceptre., 2007.

[Schneider, 2015] A. Schneider. "The Real-Time Volumetric Cloudscapes Of Horizon: Zero Dawn". *ACM SIGGRAPH*. Los Angeles, CA: ACM SIGGRAPH, 2015. Web. 26 Aug. 2015.

[Hillaire, 2016] Sebastien Hillaire., "Tiling Volume Noise" https://github.com/sebh/TileableVolumeNoise. 2016.

[Clausse and Facy, 1961] R. Clausse and L. Facy, *The Clouds*. London: Evergreen Books LTD., 1961.

[Fiorani & Nova, 2013] Rfrancesca Fiorani & Alessandro Nova, *Leonardo da Vinci and Optics*. Venice: Marsilio Editori., 2013.

[Wrenninge, 2013] M. Wrenninge, *Production Volume Rendering: Design and Implementation*. CRC Press, 2013.

[Schneider 2016] Andrew Schneider, GPU Pro 7: *Real Time Volumetric Cloudscapes*. p.p. (97-128) CRC Press, 2016.

[Hillaire, 2016] Sebastien Hillaire., "Physically based Sky, Atmosphere and Cloud Rendering" https://www.ea.com/frostbite/news/physically-based-sky-atmosphere-and-cloud-rendering. 2016.

[Mitchell, 2007] Kenny Mitchell., "Volumetric Light Scattering as a Post Process", https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch13.html. 2007.

SIGGRAPH2017 @vonschneidz | www.schneiderfx.co | www.guerrilla-games.com Advances in Real-Time Rendering, Siggraph 2017

- Thanks to my coworkers at Guerrilla and Kojima Productions,
- As well as others who have helped me over the years.
- Thanks to Natasha for allowing us to be part of the course again this year.

Finally, thank you to members of the community. While we are pursuing a specific track with Nubis, the challenge of mastering clouds in general is a continuation of a very old human effort.

- Here are some of the references that I mentioned.
- If you are interested reading more about clouds from a non-graphics perspective I recommend the references in bold.
- If you are interested in the early ray-tracing work done by Carl Ludwig and others at Blue Sky, There is a panel happening down the hall this afternoon.

The slides for this should be online as soon as Natasha can upload them.

Thank you for your time.

I can take questions now.