

Creating a Planet and Clouds Lightyears Away

Laura Murphy
lmurphy@pixar.com
Pixar Animation Studios
Emeryville, CA, USA

Joshua Jenny
joshj@pixar.com
Pixar Animation Studios
Emeryville, CA, USA

Michael O'Brien
mobrien@pixar.com
Pixar Animation Studios
Emeryville, CA, USA

Colin Thompson
colin@pixar.com
Pixar Animation Studios
Emeryville, CA, USA



Figure 1: Planet T'Kani Prime in *Lightyear*, featuring procedurally generated crater terrain and a shell of volumetric clouds.
©Disney/Pixar.

ABSTRACT

For *Lightyear*, Space Ranger Buzz Lightyear's exploits take him on a journey around the planet, T'Kani Prime and its neighboring star. In order to create realistic planets as seen from his star ship, we built a new workflow for creating procedural planet terrains and volumetric clouds as seen from space. These techniques needed to produce realistic results but also be highly art directable to help the audience believe Buzz could get to Infinity and Beyond.

ACM Reference Format:

Laura Murphy, Joshua Jenny, Michael O'Brien, and Colin Thompson. 2022. Creating a Planet and Clouds Lightyears Away. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '22 Talks)*, August 07-11, 2022. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3532836.3536268>

1 VOLUMETRIC PLANET CLOUDS

Our standard clouds pipeline [Murphy et al. 2018] [Wrenninge 2015] is designed to dress volumetric clouds into a sky scape while providing highly artistic controls. At planet scale, we wanted a method for the same quality and art directability as clouds we create in a skyline, but at a much larger scale. We created a RenderMan plugin that would texture map a 2D image into a repeating spherical, volumetric patch with thickness. Overview in Figure 2.

We use the traditional spherical coordinate projection to map an image into a spherical shape. We chose spherical as it was the

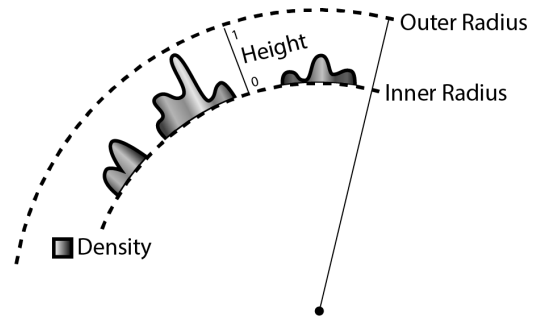


Figure 2: The RenderMan volume plugin uses an inner and outer radius to determine a candidate region for the clouds. Then for each (θ, ϕ) , the height above the inner radius and the density are sampled from an image. ©Pixar.

easiest for art to paint and the clearest to show the relationship between sections of the image and how those appear in the final render. In the mapping from the image to the sphere, we provide controls to scale the mapping in both θ, ϕ so that we can ensure the image covers the most screen space in the resulting camera.

The plugin allows for both density and height of the resulting atmosphere to be specified by textures. The combination of those two signals allowed us to work with art to produce clouds that look like familiar images from satellites, but with an alien feel. We originally intended to use two different images (one for density and one for height). It turned out we got better visual results and had an easier time working with art by using the same image. We worked with art so they understood that white in the image would be the full cloud density and full cloud height. This bred a rough guideline that clumps of white would look like cumulonimbus clouds (and should be painted as clumps) and gray streaks would look like cirrus clouds (and painted as such).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH '22 Talks, August 07-11, 2022, Vancouver, BC, Canada

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9371-3/22/08...\$15.00

<https://doi.org/10.1145/3532836.3536268>

Submission ID: ID. 2022-05-26 16:27. Page 1 of 1-2.

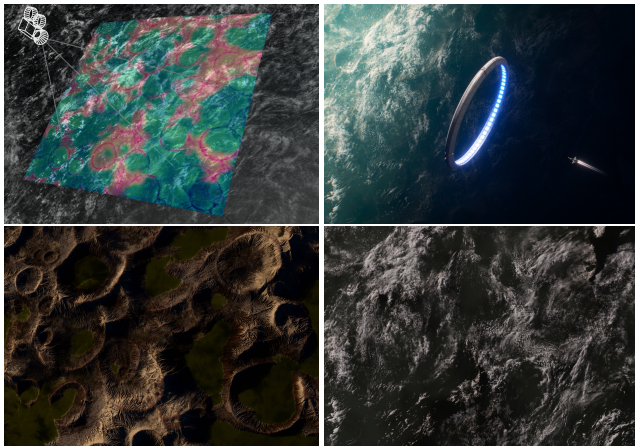


Figure 3: Comparing clouds and terrain in houdini and rendered frames (houdini viewport, final comp, terrain render, and cloud render). ©Disney/Pixar.

The spherical volume boundaries are fast to compute, making rendering in RenderMan very efficient. This allowed us to use multiple layers of spherical volumes to achieve different looks. For some cases rather than using a painted texture, we used a constant presence, with settings for dialing the falloff along the height to thin the atmosphere away from the planet’s surface.

2 VOLUMETRIC SHOT WORKFLOW

Early in the show, we established a default planet cloud rig and we painted a default 8k tiling cloud presence texture. These were used as the starting point for all shots. We were able to use the default texture to create the final composition for many shots using variations in frequency, rotation, and framing. Some shots also used options from the rig to adjust cloud height and density.

We had the option to visualize the planet clouds in Houdini to make framing decisions. This was particularly useful when comparing to the planet’s crater terrain, and for tackling shots where both planet clouds and clouds from our standard cloud pipeline are needed. Fig. 3 shows a comparison of houdini and rendered images.

Shot specific updates to the textures are painted in photoshop and delivered with our mattepaint export pipeline. One particularly useful upgrade to the mattepaint export process was adding Slack (chat system) support to automatically notify the technical artist over Slack when a painting export is finished and ready to use.

To promote tight collaboration between the technical artist and painting, we stored the painting name in the shot database, facilitating painting creation, version management, and installation. We also kept a standard set of commands for daytime rendering and viewing frames. This reduced the friction of data making the round trip from photoshop to seeing the results in render.

3 PROCEDURAL PLANET TERRAIN

The procedural terrain pipeline was designed to make it easy to generate high fidelity planet terrain (as seen from space) for a specific shot. See Fig. 4.

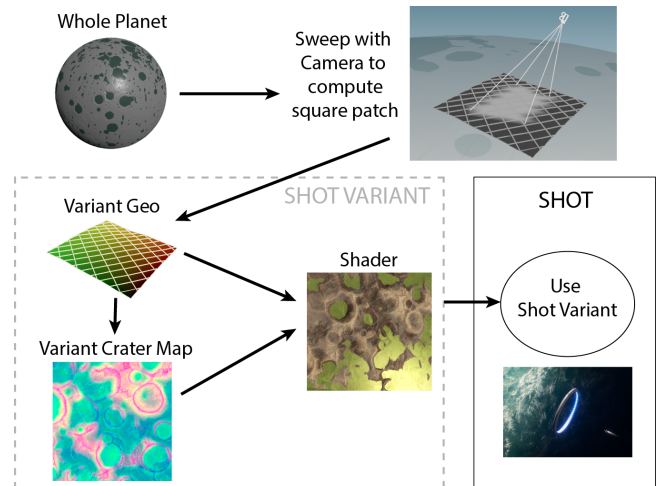


Figure 4: Starting with the whole planet model, we sweep the camera over the terrain to find a square patch, and export the patch as the variant geometry. Then we generate the variant crater map, which is used by the shader. Finally, we enable the shot specific variant in the shot. ©Disney/Pixar.

Before we begin, the Layout team (camera department) uses a low resolution model of the whole planet to compose shots. Then we can upgrade and replace the roughly spherical planet geometry with the roughly square shot specific variant.

Specifically, we create a variant per shot in houdini that includes a square geometry patch and corresponding high resolution texture map. Once the variant is created and installed, we enable the upgraded terrain in a shot by adding the shot to a shot enabled group, which uses expressions to select the correct variant for the shot.

To create each shot variant in houdini, first we sweep the camera frustum over the base planet model to find a padded square area of the planet that is visible in the shot. This region is exported as the geometry for the shot variant. Then we use houdini heightfield tools to generate crater displacement and erosion maps. The network was packaged into a Houdini Digital Asset so we can easily set up a new shot and tweak a few parameters. The variants are added to the model with Houdini’s Solaris. The terrain shader is created and applied in our proprietary shading software, Flow, where we add additional details, including small craters and small lakes.

4 ACKNOWLEDGEMENTS

Thanks to all those who helped support this workflow, including Max Bickley, Chia-Chi Hu, Paul Topolos, Martin Senn, Alec Bartsch, Keith Kohn, Bill Watral, Sudeep Rangaswamy, and Jane Yen.

REFERENCES

- Laura Murphy, Martin Sebastian Senn, and Matthew Webb. 2018. Efficient Hybrid Volume and Texture Based Clouds. In *ACM SIGGRAPH 2018 Talks* (Vancouver, British Columbia, Canada) (*SIGGRAPH '18*). Association for Computing Machinery, New York, NY, USA, Article 39, 2 pages. <https://doi.org/10.1145/3214745.3214797>
- Magnus Wrenninge. 2015. Art-Directable Multiple Volumetric Scattering. In *ACM SIGGRAPH 2015 Talks* (Los Angeles, California) (*SIGGRAPH '15*). Association for Computing Machinery, New York, NY, USA, Article 24, 1 pages. <https://doi.org/10.1145/2775280.2792512>