Building detailed fractal sets for "Guardians of the Galaxy Vol. 2"

Matt Ebb
Animal Logic
Matt.Ebb@al.com.au

Richard Sutherland
Animal Logic
Richard.Sutherland@al.com.au

Daniel Heckenberg
Animal Logic
Daniel.Heckenberg@al.com.au

Miles Green
Animal Logic
Miles.Green@al.com.au

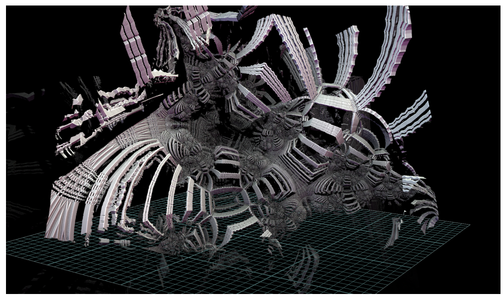

Figure 1: Fractal modelling

ABSTRACT

Building the digital sets for *Guardians of the Galaxy Vol. 2* presented a unique challenge for *Animal Logic's* asset and FX teams. The creative brief involved two separate alien environments made from complex mathematical shapes, with an unprecedented amount of detail. As an additional challenge the environments had to match the look and feel of specifically styled concept art with a grand, monumental design. To meet the artistic requirements required a high level of creative control and manipulation of set elements that were to be rendered alongside many other highly detailed objects, and propagated quickly through the pipeline for fast feedback iterations. The team used a novel approach to modelling fractal objects using point clouds, taking advantage of pipeline capabilities to integrate FX objects with environment set-pieces. Additionally the team, leveraged instancing and high levels of geometric complexity using the in-house renderer *Glimpse*.

CCS CONCEPTS

•**Computing methodologies** →**Rendering;** *Computer graphics; Animation;*

KEYWORDS

fractals, ray marching, distance fields, instancing, modelling, point clouds

DOI: http://dx.doi.org/10.1145/3084363.3085060

ACM Reference format:
Matt Ebb, Richard Sutherland, Daniel Heckenberg, and Miles Green. 2017. Building detailed fractal sets for "Guardians of the Galaxy Vol. 2". In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085060

1 FRACTAL MODELLING

From an early stage it was clear that the project's creative design process would require an smarter alternative to a brute force modelling approach. Various techniques for modelling fractals as 3D objects were considered, such as exporting static geometry from external applications, building high-resolution volumes, preprocessing depth and shadow maps, or tracing the fractal function directly at render-time. The team had used some of these techniques previously on *Avengers: Age of Ultron*, and were aware of the limitations. During the design process, to quickly visualise a variety of potential fractal styles that could be used, a modular, interactive fractal visualiser was built in *Houdini* by translating fractal formulae to VEX code. Inspired by distance field rendering (eg. shadertoy.com), the tool used sphere-tracing to efficiently intersect the fractal surface from the camera, but rather than generating pixels it created a dense point cloud as Houdini geometry, with data such as a normal, colour, occlusion, and other fractal specific attributes stored per-point. This enabled near real-time 'scouting' through fractal structures in the Houdini viewport and had a major advantage of being very easy to freely explore deep inside the fractal space at various scales, opening up a vast range of design possibilities that were previously not easily accessible. It soon became clear that this technique could be used not just for visualisation, but also for generating final models. Once an interesting area of fractal was discovered, artists could model bounding geometry to clip and extract the selection. They would then generate a final and extremely dense point cloud by tracing points inwards from the bounding surface. Most final objects ranged between between 10 to 100 million points, but because only externally visible surfaces were evaluated, this allowed for a very high level of detail with a low speed and memory budget. The final generated point clouds could then still be manipulated with a high degree of creative control using any of *Houdini's* native tools, allowing them to be post carved, warped, duplicated and re-arranged. Some final set pieces were created from a combination of several different fractal point clouds, after editing to fit the desired architectural volumes.

2 PIPELINE

Putting this into practice to generate entire sets with a sense of design continuity involved multiple artists and departments working together. Modellers and layout artists would rough out the basic architecture, generating low-resolution bounding objects for 'fractalisation', while also modelling high resolution framework geometry to be used in conjunction with the fractal objects. Fractal point clouds were then generated, that fit inside and were sympathetic to the shapes of the bounding geometry, and delivered in a renderable *Glimpse* archive. A recent upgrade in the *Animal Logic* environment pipeline, allowed any arbitrary FX renderable data to be attached to set piece assets that didn't necessarily need to be loaded into the viewport. With this functionality, fractal objects could be treated as top-level assets that could be instanced in sets and shots, using the custom '*Scenery* environment publishing tools, without needing to re-publish any of the fractal geometry when layout changes occured.

3 RENDERING

Rather than converting to polygon geometry, the fractal point clouds were rendered directly in *Glimpse* as millions of efficient tightly packed sphere primitives, using the stored fractal normal to override the sphere's shading normal to give the impression of a solid. This was surprisingly effective, even for smooth metallic surfaces, and led to much faster turnarounds than converting to and cleaning up polygon geometry. Surfacing artists used fractal data stored in the point clouds and others such as density, occlusion, or curvature to control shader parameters, using *Glimpse's* node-based shading system *Ash*. Instancing of the fractal point cloud objects in *Glimpse* was heavily used in order to contain memory usage - one of the final sets contained 33 set pieces of total 250 million points, but 2.97 trillion points if not instanced.