Exploring Fields of Tension in VR

Benjamin Storch, Artist, PhD
Melbourne, Australia
www.benjaminstorch.com
e-mail: benstorchstudio@gmail.com

Abstract

Strange attractors occur in the mathematical plotting of dynamical systems and display curious behaviours when the system evolves over time. The author originally explored their visual qualities as a source of inspiration for metal sculptures at the start of the millennium. At the time, some basic animations in MATLAB with a changing parameter displayed patterns reminiscent of colliding galaxies and subatomic forces. Recent advances in VR hardware and software promised an avenue to bring these fascinating 'fields of tension' to life in an immersive and inter- active medium, and some of the methods and outcomes are discussed here.

1 Background

I was always fascinated by the movement of flowing water and other patterns found in nature that speak of forces and self-organising systems. In the arts, I admired both the fluid lines of Art Nouveau and non-orthogonal angular geometries. These two, the fluid and the geometrical, seemed to merge in a series of diagrams called 'Harmonograms' that are drawn by a system of coupled pendulums. Besides acting like a trajectory under the influence of centrifugal and centripetal forces, differences of amplitude and phase between the pendulums create dynamic angular changes in the 'compound orbit', and the repeating yet offset curve patterns suggest three-dimensional, twisting surfaces.

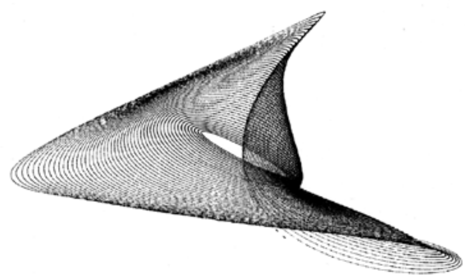


Fig. 1: 'Harmonogram' drawn with home-built 'Harmonograph'.

Fig. 2: Resonance, stainless steel, 2018

With a background in silversmithing, my research was primarily focused on the forming processes required to stretch metal into the compound, saddle-shaped curvatures needed to realise these 'fluid' qualities.

In the years following the completion of my research I was too absorbed in the making process to devote time to 'generative mathematics', but I had always wanted to revisit the potential of animating strange attractors.

At the Bridges Conference 2006 [1] an encounter with mathematicians Bernd Krauskopf and Hinke Osinga led to a collaboration on sculpting a version of the 'Lorenz manifold', a not often seen aspect of the Lorenz attractor, which the two mathematicians had been investigating.

Fig. 3: Manifold, stainless steel, 2008, displaying a section of the Lorenz manifold.

The Lorenz attractor is the most widely publicised visual depiction of a strange attractor, plotted with equations that seek to model the deterministic but chaotic behaviour occurring in the convection of fluids, with the aim of better understanding weather patterns.

Rather than depicting the flow of air currents, the pattern of the attractor reveals how different starting values and parameters (viscosity, turbulence) affect its time evolution and demonstrates states of (in)stability on an abstracted level. Stable states act as attractors, and overall, the evolution depicts various stages of flux and probability encircling the attractors.

Every day, we rely on a vast number of systems to be in equilibrium, such as our planetary orbit, our heartbeat, and gut bacteria, to name some obvious ones. At the same time, we enjoy playing with the boundaries of other systems, such as teasing our partner, increasing the centrifugal forces on our bicycle or balancing a spoon on the rim of a cup.

Saying that I was purely interested in the visual qualities of the patterns is probably not entirely correct, since I saw them as a language of fundamental forces, which could possibly help us better understand not just external forces, but our emotional states as well. To me they represent a desire to engage with the world fluidly, which in general is probably better fulfilled through music and dance, but as a visual artist I was seeking visual representations, whether static or animated - and while we can find plenty of visual examples of dynamical systems in the real world, strange attractors seemed to embody fundamental principles in a clear visual language, that contains enough unpredictability and surprise to hold our curiosity and make us wonder.

2. Revisiting digital media

Three years ago, advances in VR hardware and software encouraged me to explore the possibility of animating strange attractors in an immersive, interactive medium. On investigating the game engine ‘Unity’ as a suitable platform to build a VR app, it turned out that several people had already coded some strange attractors for it. One author [2] applied the recently developed Entity Component and DOTS systems in Unity to be able to render several hundred thousand spheres in real time. Others [3] used Compute shaders for similar results, and further adapted this to the VFX Graph that offers a node-based system as a visual scripting interface for particle systems, which was more accessible to coding novices like myself, and also highly optimised for GPU rendering.

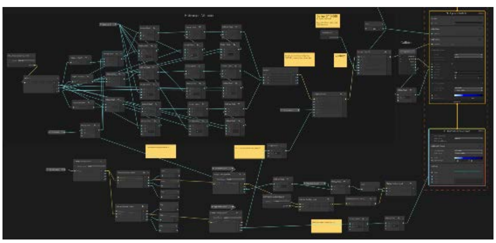


Fig. 4: VFX Graph, a node-based, visual ‘scripting’ system in Unity

In all of the above methods, the ‘particle’ positions are generally spawned from multiple start positions over a period of time to generate a ‘cloud’ of points that then evolve together over time. Particles can be given a limited lifetime to allow for new ones to be spawned within the overall quantity limit.

VFX Graph allows greater control of changing the colour and/or size over lifetime, and, more importantly, the time step, whereas the Entity system and Compute shaders seemed less flexible in terms of the graphics processing, communication between CPU and GPU.

One workaround was to have several systems running in parallel, where each system has slightly different settings.

With a good graphics card, these systems could render several hundred thousand spheres on an Oculus Rift in real time, and the Quest 2 managed to render around twenty thousand spheres, or one hundred thousand ‘points’ with an acceptable frame rate.

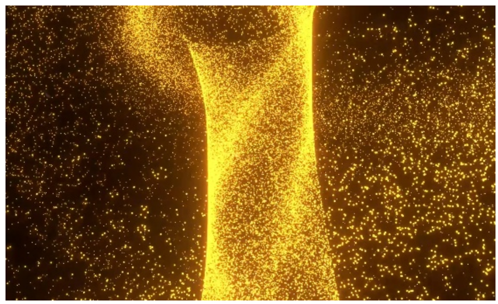


Fig. 5: Aizawa attractor animated in real time with several hundred thousand spheres using ECS & DOTS

In the above methods, a parameter change affects the entire time evolution of the positions/trajectories, and as such was different from the original method tried in MATLAB, where a full iteration from a single starting position is rendered at once, and then all the positions are updated with a parameter change. Two date the only equivalent method developed in Unity was a simple monobehaviour script that updates the full set of positions of one iteration with subsequent full sets generated with one or more parameter changes. This was not very optimised in terms of graphics processing, allowing for only around ten thousand spheres to be updated in real time.

The change in positions can be very erratic and the rate of change of the behaviour is not proportional to the amount of change of the parameter value. Sometimes a change in the first decimal place results in little change of the positions, and at other values, minimal changes in the seventh decimal place can cause rapid position changes that disrupt the sense of flow.

Nevertheless, this original method produced some fascinating results, as was to be expected, judging by the early trials in MATLAB. My main interest was to reveal the fundamental forces of attraction at play, which result in toroids, torus knots and vortices that condense, split and dissipate. Sometimes a spinning planar vortex, reminiscent of a galaxy, evolves into a torus, that can portray a great variety of wave formations through minute parameter changes that can be hard to control. Then the torus brakes up into a ring of smaller vortices, which further merge in pairs, like minute galactic collisions.

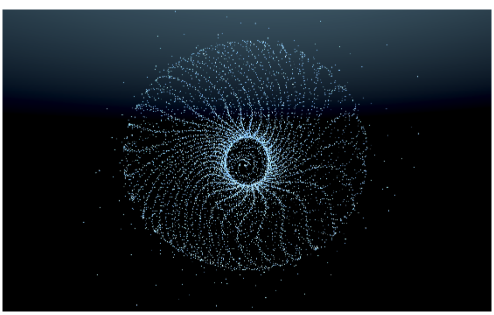
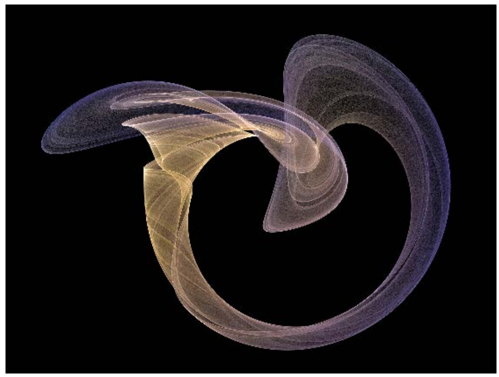


Fig. 6: Torus depicting wave-like patterns, which change in mesmerising ways when animated.

Renders created in Chaoscope [4] were sometimes used as reference, to then try and recreate similar animated plots in Unity. Equations called ‘Chaotic Flow’ and ‘Lorenz-84’ were successfully implemented in Unity, and in my view offer a fascinating range of patterns, even though elusive and hard to control at times.



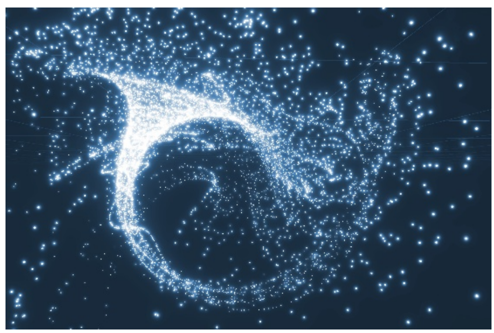


Fig.s 7&8: Lorenz-84 attractor rendered in Chaoscope, and its ‘Unity’ adaptation

To summarise, challenges are given by:

a. The way a system is initialised, spawned and whether the position updates and render parameters allow for real-time adjustments.

b. The scale of interesting behaviour varies, sometimes requiring the scaling up of the attractor by a factor of ten thousand or more to make it visible.

c. The time step of the iteration affects the pattern, and with the first three methods it was difficult to control independently from game time, ie. the animation speed.

d. With the last method, parameter changes can have nonlinear effects that are highly sensitive and as such not suitable for an interactive control.

Each method has its advantages and dis-advantages, and I still apply them in different ways to bring out particular qualities of the systems.

Unity offers the ability to separately animate the behaviour of the parent game object containing the strange attractor computation, to counteract some of these idiosyncrasies. Sometimes Unity's built-in animation tools are used to adjust one or several parameters on a timeline, similar to keyframing in video-editing.

In the context of VR, the next step was to tie the parameters of the particular algorithm/iteration to the XYZ coordinates of the hand/controller positions, to allow the user to influence the behaviour. In one example, the left controller influences the time step, which not only affects the iteration/animation speed, but also the pattern. This also caused a problem, since a visually pleasing evolution does not necessarily generate the most interesting pattern.

The right controller's vertical position was linked to one of the attractor parameters, and its horizontal position was linked to the colour values. As the animation intensified as a result of the time step, the colours would go warmer or brighter, which could be further intensified with post-processing bloom effects.

Chance results are an important part of exploration. For example, implementing the Pickover attractor [5] in VFX Graph resulted in some strange repetitions, leading to the entire space being filled with varying galaxy-like vortices connected by transversal flows.

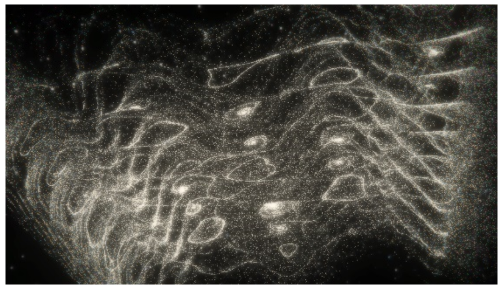


Fig. 9: Chance result from implementing the Pickover attractor in Unity's VFX Graph

3. Conclusions

Several protype versions of VR apps were tried out in an installation in May '21, with positive feedback from the audience. Sadly, a COVID lockdown ended the exhibition after just two days.

Both Oculus Rift S and Oculus Quest 2 headsets were used to test the capability of the PC CPU & GPU versus the built-in mobile processor on the Quest 2, and, while more limited, the latter performed better than expected.

Virtual Reality is still improving every year as the display technology and graphics computing power allows for a wider field of view, higher image resolution and more realistic rendering. The risk of nausea still limits movement in the virtual space to position jumps and snap turns, which is not as smooth as one would hope.

In some ways large immersive, fluid, dynamic non-VR works, such as Refik Anadol's *Quantum memories*, are visually superior and don't come with the limitations of VR, but being surrounded by particles that feel alive through their curious, self-organising behaviour that responds to your gestures is a very special experience.

High quality particle effects have become widespread in advertising, film and games, and what I started exploring twenty years ago has lost novelty, but I feel there is enough room for originality left to develop it further.

Levete kindly granted me permission to use their soundtracks, to add to the immersive ambience [6]. In the future, the aim is to make the sound responsive to the strange attractor behaviour, or have the different frequency bands of the soundtrack influence the mathematical parameters, but this is still work in progress.

A screen-based animation features in this year's conference exhibition.

Please contact the author if you would like access to one of the VR apps running on either Oculus Quest 2 or Rift (S).

Acknowledgements

Aside from the people mentioned under the references, I would like to thank Creative Victoria for part-funding the VR development through their *Sustaining Creative Workers Initiative*, and Tin Nguyen from Liminal VR (Melbourne) for his Unity coding expertise, and my PhD supervisor Keith Osman (Birmingham City University), who helped me code the animations in MATLAB that led to this investigation. I would also like to thank Unity for offering their software free of charge to emerging developers (www.unity3d.com).

References

[1] Annual Bridges Conferences on mathematical connections in art, music, architecture and culture.
https://www.bridgesmathart.org/
A joint paper on the creation of the sculpture Manifold can be found in the archive of the 2008 conference.
[2] Unity assets by NBody Physics (Peter Musgrave).
https://assetstore.unity.com/packages/tools/physics/numerical-tools-192894
[3] Tasuku Takahashi adapted a Compute shader for strange attractors written by Tokyo based 'Indie Visual Lab' (https://github.com/IndieVisualLab) to the VFX Graph format, to use with a North Star AR headset.
https://github.com/supertask/VFXNorthStar
[4] 'Chaoscope' programme by Nicolas Desprez, http://www.chaoscope.org/
[5] Pickover, C. A.(1991), "Computers, Patterns, Chaos and Beauty". London: St. Martin's Press
[6] Jim and Jillian Graham aka Levete, https://levete.bandcamp.com/album/aether