UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Houdini de SideFx: creación de efectos visuales y desarrollo de scripts con Python y VEX para la elaboración de herramientas de ayuda a equipos de postproducción

Trabajo Fin de Grado Integrado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e Imagen

Grado en Comunicación Audiovisual

AUTOR/A: **Esteban Carmona, Marta**

Tutor/a: **Flores Asenjo, Santiago José**

Tutor/a: **Pérez Esteban, José Antonio**

CURSO ACADÉMICO: ***2023/2024***

Dedicado a las salas de cine, por enseñarnos historias que inspiran las nuestras.

AGRADECIMIENTOS

A Tania por su apoyo y ayuda incondicional. Y por guiarme, sin ti este trabajo no habría sido posible. A Miguel y a mis tutores por prestarme consejo siempre que lo necesitaba.

A la Ñita por darme la seguridad que necesitaba cuando empezó este viaje. A la Nana, a mis amigos, a mis perros, gatos, caballos y loro. En resumen, a mi familia.

A mi padre por enseñarme a valorar hasta el detalle más pequeño y a mi madre por animarme a no tener miedo de soñar a lo grande.

Y a mí, por estos cinco años.

Esto es posible gracias a vosotros.

RESUMEN

Los efectos visuales (VFX) se encuentran cada vez más presentes en la mayoría de los sectores audiovisuales, en ocasiones de forma prácticamente imperceptible.

Este proyecto tiene como objetivo explorar y desarrollar herramientas personalizadas, *scripts* y *plugins* para el software Houdini, de la empresa SideFX, con un enfoque en la programación en Python y VEX (un lenguaje de *scripting* de alto rendimiento que está diseñado para funcionar con la arquitectura basada en nodos de Houdini). Houdini es conocido por su capacidad de personalización y extensibilidad, y este TFG aprovecha esas características para crear soluciones prácticas para la industria de efectos visuales (VFX) y animación.

Dada la cantidad de departamentos y tareas que caracterizan el flujo de trabajo de los VFX, surge la necesidad de la figura del Pipeline TD (Technical Director), responsable de que los artistas cuenten con las herramientas de software necesarias para crear un flujo de trabajo lo más eficiente posible y elevar así el nivel de funcionalidad a través de la creación de herramientas de automatización basadas principalmente en Python.

En este TFG se desarrollan ejemplos de herramientas de automatización con Python y de modelado procedural con VEX que ayudan en la labor de un Pipeline TD (Technical Director) con el objetivo de proporcionar asistencia a equipos de postproducción.

De esta forma es posible adquirir habilidades sólidas en programación en Python y VEX, así como comprender en profundidad el entorno de Houdini y las necesidades de la industria de VFX y animación.

PALABRAS CLAVE

Houdini; VFX; animación; Pipeline TD; scripting; modelado procedural; Python; VEX

ABSTRACT

Visual effects (VFX) are increasingly present in most audiovisual sectors, sometimes almost imperceptibly.

This project aims to explore and develop custom tools, scripts, and plugins for Houdini software from SideFX, focusing on programming in Python and VEX (a high-performance scripting language designed to work with Houdini's node-based architecture). Houdini is known for its customization and extensibility capabilities, and this final degree project leverages these features to create practical solutions for the visual effects (VFX) and animation industry.

Given the number of departments and tasks that characterize the VFX workflow, the need arises for the role of the Pipeline TD (Technical Director), responsible for ensuring that artists have the necessary software tools to create the most efficient workflow possible and thus enhance the level of functionality through the creation of automation tools primarily based on Python.

In this final degree project, examples of automation tools with Python and procedural modeling with VEX are developed to assist the work of a Pipeline TD (Technical Director) with the aim of providing support to post-production teams.

This way, it is possible to acquire solid skills in Python and VEX programming, as well as to gain a deep understanding of the Houdini environment and the needs of the VFX and animation industry.

KEYWORDS

Houdini; VFX; animation; Pipeline TD; scripting; procedural modeling; Python; VEX

ÍNDICE

1. INTRODUCCIÓN

a. JUSTIFICACIÓN

El cine lleva años logrando que confundir ficción con realidad resulte muy sencillo. Los efectos visuales se han convertido en una parte integral del sector audiovisual, permitiendo ampliar los límites de la narración desde hace tiempo.

Pero esto no habría sido posible sin los avances tecnológicos y softwares especializados que se han ido incorporando a este departamento. Pues llevar a cabo una tarea de tal envergadura requiere una gran cantidad de profesionales, equipos y tiempo, exigiendo así una coordinación y comunicación considerables por parte de los diferentes departamentos implicados.

Es por esto que Houdini, conocido por su capacidad de personalización y extensibilidad a través de lenguajes de programación como Python y VEX, ha alcanzado una posición tan remarcable en la producción de los VFX. Sus prestaciones permiten crear un flujo de trabajo más eficiente y reducir tanto temporal como económicamente los costes de una producción, dando lugar así a la figura del "Pipeline TD" ó "Technical Artist/Director".

Dado que los pilares en los que se apoya este sector, la combinación entre creatividad y tecnología, han sido la base de mi formación universitaria, este trabajo supone una oportunidad para estudiar el papel de este rol y conocer todas las posibilidades que Houdini ofrece, como primer paso hacia el que será mi campo de estudio en el futuro.

Así mismo, el estudio del recorrido histórico y la realización de proyectos basados en Python y VEX me permitirá adquirir una base sólida de conocimientos y dominar estos lenguajes de programación. De esta forma será posible mejorar mis capacidades técnicas en el desarrollo de *scripts* para la creación de herramientas y *setups* propios de un estudio de VFX real.

b. OBJETIVOS

El objetivo principal de este trabajo es crear efectos visuales de forma procedural y desarrollar *scripts* a través de Python y VEX para la elaboración de herramientas de ayuda a equipos de postproducción, demostrando de esta forma la versatilidad de Houdini y así experimentar el papel de un Pipeline TD en el flujo de los VFX.

Para llevar a cabo esta idea, se plantean varios objetivos secundarios. Como primer objetivo secundario se pretende estudiar y profundizar en la raíz y la evolución de los VFX para conocer así la transición analógico-digital marcada por la aparición de la tecnología y los softwares especializados, además de reflexionar sobre los avances en el futuro de la industria de los efectos visuales. Adicionalmente, comprender las características que diferencian los efectos especiales, de los efectos visuales y del CGI (Computer Generated Imagery), términos a menudo utilizados de forma errónea y distinguir la finalidad de cada una de estas técnicas.

El siguiente objetivo secundario trata de explicar las funciones de un "Pipeline TD" y su trascendencia en el flujo de los VFX, así como la importancia de los diferentes equipos dentro de este departamento. Será imprescindible comprender las dificultades a las que se enfrenta cada uno de ellos, para desarrollar las soluciones adecuadas.

Para poder alcanzar el objetivo principal, otro de los objetivos secundarios importantes es comprender el concepto de modelado procedural, estudiar la terminología y lógica dentro de Houdini, así como su interfaz y forma de trabajo, consiguiendo de este modo tomar contacto con el entorno en el que se lleva a cabo el primer objetivo. De igual forma, será indispensable estudiar los lenguajes de programación de Python y VEX, además de plantear las preguntas adecuadas y desarrollar algoritmos con el fin de realizar los proyectos que permitirán demostrar el potencial de Houdini.

Por último, establecer una red de contactos para obtener apoyo y consejo por parte de profesionales del sector será de suma utilidad para la realización de este trabajo y así dar un paso más hacia mi futuro laboral.

c. METODOLOGÍA

Para alcanzar los objetivos planteados, se definieron diferentes pautas como parte de la metodología a seguir. Como análisis previo, se estableció contacto a través de redes sociales y portales de empleo con profesionales del sector, con la intención de obtener una visión objetiva de la amplitud del proyecto y poder así acotar el tema para determinar el mejor procedimiento a seguir. Para comenzar, se realizaron cursos indicados por los profesionales consultados, tanto en Python como especializados en VEX para Houdini, a fin de construir una base sólida en el manejo del *software*.

A continuación, como primer paso de la investigación teórica, se consultaron otros proyectos y trabajos de investigación tales como revistas y artículos académicos, siguiendo una metodología basada en la investigación bibliográfica, para obtener información sobre el origen y evolución de los efectos visuales aprendiendo a diferenciar entre estos, los

efectos especiales y el CGI. Por otro lado, se revisaron materiales de carácter audiovisual: películas, documentales, videos, etc. con vistas a analizar el flujo de los VFX y los departamentos que lo componen, que permite comprender la labor del "Pipeline TD" en una producción audiovisual.

De igual manera, se estudió la dinámica de Houdini para comprender el funcionamiento de un flujo de trabajo basado en nodos y el concepto de modelado procedural por medio de la investigación en páginas webs, foros y manuales de usuario.

Por último, se realizaron dos proyectos diferentes en el propio Houdini con la intención de mostrar las habilidades adquiridas a través de la realización de este trabajo.

d. ESTRUCTURA

El presente trabajo se ha estructurado en diez puntos. Primeramente, se introduce la justificación o motivación de la elección del tema como parte del contexto, seguida de los objetivos, la metodología y la estructura establecida.

En segundo lugar se encuentra el apartado "Historia y evolución de los efectos visuales, desde el origen hasta la actualidad", en el que se realiza una síntesis que recoge los inicios de los efectos visuales hasta la fecha, haciendo hincapié en el estudio de la transición analógico-digital y su evolución hasta nuestros días. Este sirve como punto de partida para conocer los conceptos básicos sobre los efectos digitales y su presencia en la industria audiovisual.

A continuación, en el tercer capítulo "El rol del Pipeline en los VFX", se analiza de forma detallada el flujo de trabajo de los VFX y los departamentos que forman parte de este para poder entender la relevancia del director técnico y comprender así su aportación a cada uno de ellos.

En cuarto lugar, el punto "Houdini de SideFX" se enfoca en el *software* especializado escogido para la realización de este trabajo. Con base en la investigación teórica, se describen sus prestaciones y repercusión en el sector audiovisual, comparando estas características con las de programas similares, lo que sirve de justificación para la elección de Houdini como herramienta de trabajo.

A partir del conocimiento adquirido sobre Houdini, en el quinto capítulo "Modelado procedural" se profundiza en este concepto para ir más allá y conocer sus ventajas frente al modelado tradicional, analizando sus características y creciente presencia en la industria.

De igual forma que con los diferentes tipos de efectos, el sexto apartado "Python y VEX: diferencias y funciones dentro de Houdini" se centra en estudiar los distintos lenguajes de programación que soporta Houdini y en qué se emplea cada uno de ellos. Para ello, se exponen las principales particularidades que caracterizan su sintaxis y se reflexiona acerca de sus utilidades.

Así pues, habiendo estudiado el significado de esta idea, se pone en práctica en el punto "Desarrollo de proyectos con Houdini" en el que se analiza de forma metódica la realización de dos proyectos en Houdini y sus respectivos códigos y algoritmos en los que basan su funcionamiento. Partiendo de los *scripts* desarrollados será posible conocer de manera experimental la versatilidad de Houdini y las herramientas que ofrece.

Para finalizar, en el apartado de "Conclusiones" se recogen los resultados e ideas adquiridas y se evalúa el curso de los objetivos planteados así como la superación de los problemas surgidos en la realización de este trabajo.

Posteriormente se dispone de la bibliografía, filmografía y un índice de anexos, que contienen la información que ha sido relevante en la elaboración de esta memoria. También se incluye una bibliografía adicional para profundizar más en ciertos conceptos mencionados a lo largo del trabajo.

2. HISTORIA Y EVOLUCIÓN DE LOS EFECTOS VISUALES, DESDE EL ORIGEN HASTA LA ACTUALIDAD

Desde el *Viaje a la luna (Méliès, 1902)* de George Méliès y los primeros "trucos" de principio del siglo XX hasta las imágenes generadas por ordenador que protagonizan las pantallas de hoy en día, el crecimiento de los efectos visuales está marcado por la unión entre la creatividad humana y el avance tecnológico.

Este arte impulsado por el ingenio, ha evolucionado a través de películas pioneras en la introducción de innovadoras técnicas y herramientas las cuales se han desarrollado hasta convertirse en los modernos VFX que predominan en la industria actual. Para conocer mejor la historia de los efectos visuales y comprender el contexto en el que surgieron, es esencial repasar y analizar algunos de los principales filmes que dieron paso a este tipo de efectos.

Pero antes, es importante diferenciar entre "efectos especiales", "efectos visuales" y "CGI", términos que a menudo producen confusión y que, aunque guarden relación, no son comparables.

a. EFECTOS ESPECIALES, EFECTOS VISUALES Y CGI: DEFINICIÓN Y DIFERENCIAS

A lo largo de la historia el cine ha llenado las pantallas con cantidad de trucos para engañar o sorprender al espectador mediante ilusiones ópticas, creando realidades imposibles de imaginar sin estas técnicas. Desde decorados y cambios de perspectiva, los efectos especiales han existido casi desde la creación del cine y, gracias a los avances tecnológicos, ahora se ayudan de efectos visuales o digitales. Considerando la evolución de este campo presente en las fases de producción y postproducción, cada vez es más frecuente confundir ficción con realidad. Por ello, aunque se desarrollan desde perspectivas diferentes, estos conceptos suelen dar pie a confusión. Es necesario definir qué tipo de efectos engloban estos términos y las diferentes herramientas de las que se sirve cada uno para referirse a ellos en el siguiente apartado.

Por un lado, los efectos especiales, prácticos o físicos, conocidos por su abreviatura FX (en inglés *special effects*), pertenecen a la fase de producción, es decir, se realizan durante el

rodaje. Desde el nacimiento de la cinematografía se han utilizado para crear escenarios, decorados y personajes mediante mecanismos físicos, sin el uso de ordenadores, aunque hoy en día suelen ser retocados en postproducción. Dentro de esta clasificación, se diferencian varios tipos de efectos especiales según su finalidad:

- Efectos especiales ópticos: inspirados en el ilusionismo, basan su funcionamiento en la perspectiva y los juegos de luces para confundir al ojo humano. Entre estos se encuentran la rotoscopia, el *matte painting*, el *stop motion* o la retroproyección, conceptos que se comentarán con más detalle en adelante.

- Efectos especiales mecánicos: en esta categoría se incluyen desde explosiones controladas, caídas y fenómenos atmosféricos hasta la caracterización o creación de personajes y escenarios mediante el uso de maquetas, maquillaje, prótesis, y animatrónicos[1]. Un ejemplo de este tipo de técnicas puede verse en *Lo imposible* (Bayona, 2012), para la que se construyó un tanque de agua real en el que se recreó un tsunami.

Por otro lado, los efectos visuales o digitales (más conocidos por su abreviatura VFX, en inglés *visual effects*) se llevan a cabo, normalmente, durante la postproducción. Con el uso de ordenadores y softwares especializados[2], como Houdini o Nuke, permiten modificar, componer y crear imágenes de forma digital.

Actualmente la mayoría de películas, series y videojuegos incluye este tipo de efectos dada su versatilidad e infinidad de posibilidades que este tipo de técnicas digitales permiten realizar. Con el paso del tiempo, cada vez son más realistas e imperceptibles, convirtiendo su principal objetivo en el de que ningún espectador pueda imaginar que las imágenes que está viendo han necesitado de efectos digitales.

Algunos de sus usos y funciones más características son el modelado de escenarios, a los que luego se añaden materiales o texturas (técnica que se conoce como *shading*), y criaturas o personajes, posteriormente dotados de una estructura deformable para poder ser animados (*rigging*). También permiten realizar simulaciones de derrumbamientos (RBD o Rigid Body Dynamics) o explosiones; fluidos, por ejemplo, lluvia o nieve; y fuego o humo, a través de la manipulación de diferentes parámetros físicos como son la gravedad, el peso o la velocidad, ayudándose de lenguajes de programación y otras ciencias aplicadas.

Además, dentro de esta categoría se incluyen otras técnicas como:

[1] La animatrónica es la rama de la ingeniería que combina el uso de la mecánica y la robótica para dar movimiento a maquetas o modelos físicos con el fin de simular en estos el comportamiento de un ser vivo.

[2] Programa informático enfocado en la ejecución de operaciones y automatización de tareas comunes relativas a un sector o industria o concretos.

- ***Morphing***: basado en la interpolación[3]. Este proceso de animación digital permite realizar una transición gradual entre dos objetos o elementos diferentes obteniendo un efecto de transformación. La técnica original utilizaba mallas superpuestas sobre las imágenes para delinear las características clave de cada una e intercalarlas sucesivamente hasta aplicar la malla final para distorsionar la imagen y disolverla con la otra. Con la llegada del CGI y el 3D, esto pasó a realizarse con puntos clave o *key points*. Su primer uso se dio en 1988, con el estreno de la película *Willow* (Howard, 1988), en la que un animal se transforma en otro hasta convertirse en una hechicera.

- **Captura de movimiento o *motion capture***: este sistema consiste en la creación y animación de modelos digitales a partir de las acciones y expresiones de los actores, que normalmente son captadas por sensores de movimiento y trajes especiales. Una de las películas que se sirvió de este método fue *Dawn of the Planet of the Apes* (Reeves, 2014).

- **Simulación de multitudes o *crowds***: mediante este tipo de animación basada en ciclos y replicación, es posible crear pequeños o grandes grupos de personajes para casos en los que no es posible reunir una cantidad elevada de los mismos en el set de rodaje debido a limitaciones de logística u otro tipo de restricciones. Resulta muy útil en escenas de guerras o batallas, por ejemplo.

- ***Full CG***: en los planos o imágenes generados con esta técnica todo es digital, es decir, no hay parte "real" o "plate " ya que nada ha sido rodado de forma física. Una de las series que hizo uso de esta fue *Game of Thrones* (Benioff, 2011-2019).

- ***Match moving***: basándose en el rastreo de movimiento, posibilita la inserción de un elemento en una secuencia para integrar datos 3D o digitales en imágenes reales. Esta técnica se utilizó para acoplar el traje de superhéroe en *Iron Man* (Favreau, 2008).

También existen otros métodos basados en el uso de *keyframes*[4], la realidad virtual o la combinación de ambas, como fue el caso de *The Lion King* (Favreau, 2019), para el que se empleó una tecnología más similar a la de los videojuegos.

Por último, el término CGI (Computer Generated Imagery) hace referencia a las imágenes generadas por ordenador, utilizando *software* especializado como Autodesk Maya o Blender. En realidad, no es tanto otra categoría de efectos, sino que se trata de una técnica específica dentro de los efectos visuales. El CGI se centra en la creación de elementos completamente digitales, mientras que los VFX incluyen la integración de estos elementos

[3] Método de estimación o aproximación para obtener los puntos no muestreados de una función a partir de los datos de sus puntos conocidos más cercanos.

[4] Animación basada en la introducción de claves (o *keys*) en los fotogramas (o *frames*) para ajustar la posición, rotación y escalado de un objeto o personaje que actuará en función de estos parámetros.

en la producción audiovisual, junto a otros efectos prácticos (UT-HUB, 2024). De hecho, dada su cantidad de prestaciones, actualmente se emplea en diversos sectores como el del entretenimiento, la publicidad, la arquitectura o la ingeniería.

Sin embargo, a pesar de sus diferencias, los efectos visuales y especiales se combinan simultáneamente en multitud de ocasiones. Es el caso de los fondos verdes, por ejemplo, presentes de forma física durante el rodaje para después cambiar su aspecto en la postproducción.

Como concluye Favreau (Vega, 2019), con el paso del tiempo surgen nuevas herramientas para contar historias, se hallan sus limitaciones y se desarrollan otras. Desde la perspectiva forzada o el *stop motion*, aparecen inventos de forma repentina que cambian la forma de hacer efectos inspirando a una nueva generación de artistas. A continuación, se analizará el surgimiento de algunas de estas técnicas, a través de filmes de referencia, hasta llegar a la actualidad.

b. LE VOYAGE DANS LA LUNE (MÉLIÈS, 1902)

En *L'Arrivée d'un train à La Ciotat* (Lumière, 1896) los hermanos Lumière consiguieron que parte de los espectadores abandonaran la sala al ver como una locomotora se aproximaba hacía ellos a través de la pantalla. Sin embargo, fue con el estreno de su invento, el cinematógrafo[5], en 1895 cuando un mago remplazó lo que había sido su teatro, el Teatro Robert-Houdin, por un estudio.

Este hombre era Georges Méliès, un ilusionista y cineasta francés al que también se le conoce como el "padre de los efectos especiales", dado que es considerado el precursor del ilusionismo en la gran pantalla. En 1902 estrenó *Le Voyage dans la Lune* (Méliès, 1902), hoy aclamada por ser la primera película de ciencia ficción de la historia[6], pionera en ofrecer los trucos visuales que sentaron las bases de los efectos especiales.

Esta obra fue la responsable de que multitud de cineastas de aquel entonces se atrevieran a explorar nuevas formas de contar historias. Al carecer de la compleja tecnología actual, Méliès se basó en el ingenio y la fabricación artesanal para crear imágenes inauditas en

5 Aparato formado por una caja de madera con un objetivo y una película perforada que, mediante el movimiento de una manivela, permitía capturar fotografías posteriormente superpuestas para crear una secuencia la cual podría reproducirse o proyectarse más adelante en una pantalla con la propia máquina. Se estrenó en *La Sortie de l'usine Lumière à Lyon* (Lumière, 1895)

6 Es importante hacer mención de la cineasta francesa Alice Guy, que años antes estrenó *La Fée aux Choux* (Guy, 1886). Aunque este es considerado el primer largometraje de ficción, no es tan reconocido dado que la época en la que se estrenó estaba caracterizada por una sociedad que no solía otorgar créditos a los méritos de las mujeres. Sin embargo, Guy utilizó este medio para contar una historia a través de las imágenes, al contrario del uso científico que se le pretendía dar a este arte. Escribió, dirigió y produjo sus propias películas, convirtiéndose en la primera persona capaz de mantenerse económicamente gracias al negocio cinematográfico.

aquella época. Algunas de las innovadoras técnicas que Méliès empleó fueron las siguientes:

- **Truco de la parada o *stop trick***: el primer efecto que Méliès realizó no fue intencionado, sino que, al igual que muchos otros descubrimientos, surgió por accidente. Era 1897 cuando su cámara quedó atascada mientras filmaba en la Ópera de París y al ver el resultado de las imágenes reveladas, detectó que lo que antes era un carro de caballos se había convertido en un carruaje fúnebre. Se trata del truco de sustitución o de la parada, el cual consigue un efecto de desaparición o reemplazo. Al percatarse de la cantidad de oportunidades que le brindaba este recurso, empezó a utilizar este truco de manera consciente, mediante el corte y la unión de diferentes tomas, comenzando por *Escamotage d'une dame au théâtre Robert Houdin* (Méliès, 1896), considerada la primera película con efectos especiales.

- **Animación fotograma a fotograma o s*top motion***: este método consiste en la animación de objetos inanimados a través del movimiento físico de estos entre fotogramas. Como resultado, se obtiene una sensación de movimiento cuando las imágenes capturadas se reproducen a una velocidad normal. De esta forma fue posible grabar la simbólica escena del cohete incrustado en el ojo de la luna (véase figura 1). El español Segundo de Chomón, también considerado pionero en este ámbito, profundizó en este sistema con *Hôtel électrique* (Segundo de Chomón, 1905). Esta técnica sería perfeccionada posteriormente con entregas como *Star Wars. Episode V: The Empire Strikes Back* (Kershner, 1980).

Figura 1. Cohete incrustado en el ojo de la luna en *Le Voyage dans la Lune* (Méliès, 1902) usando la técnica de *stop motion*. (Rawpixel, 2014).

- **Exposición múltiple**: la doble exposición o exposición múltiple permite captar dos fotografías en un mismo fotograma, mediante la exposición sobre la misma imagen de dos instantes diferentes que se funden creando un efecto único. Méliès, entre otros cineastas, para lograr este tipo de transiciones probaba a superponer tomas con el fin de obtener distintos efectos, por ejemplo, hacer aparecer y desaparecer elementos como personajes o partes del decorado. Este recurso le permitió crear las escenas en las que los viajeros que se encuentran con extraños "extraterrestres".

- **Fundidos y disoluciones**: mediante esta técnica fue posible hacer cambios entre escenas, dando una sensación de continuidad y dinamismo a la narrativa. Aunque este tipo de transiciones es muy común en la actualidad, supuso una novedosa revelación para aquel entonces.

- **Perspectiva forzada**: mediante la manipulación del tamaño y el posicionamiento de los objetos dentro del cuadro, las imágenes adquirían ilusión de profundidad y escala. Este recurso permitía dotar a los objetos de gran tamaño y percibir los escenarios como inmensos paisajes. Como añadido, en los decorados se combinaban los elementos tridimensionales con los dibujos, tratando de imitar la profundidad en algunos de ellos.

- **Sobreimpresión**: otro de los trucos que Méliès inventó, el cual más tarde se convertiría en el predecesor de lo que hoy se conoce como *chroma key*[7]. Se basaba en la superposición repetitiva de los mismos fotogramas para conseguir que las imágenes se solapasen. Tras filmar, se rebobinaba y se rodaba una nueva acción. El fondo negro no quedaba registrado, de modo que se reservaba un área sensible que se podía llenar o sustituir más adelante. Este es el caso de *L'homme orchestre* (Méliès, 1900), en el que se utilizó para multiplicar un personaje siete veces de forma simultánea.

Aunque Méliès no obtuvo el reconocimiento que merecía en aquel entonces, su trabajo y aportaciones a la cinematografía sirvieron como referencia para sentar los valiosos pilares que sostienen la base de los efectos visuales de hoy en día. A través de su ingenio y conocimientos en mecánica construyó un legado que permanece presente inspirando a generaciones de artistas, demostrando que hacer lo imposible es posible con un poco de imaginación.

[7] En el siglo XIX nace el *chroma key* o llave de color, que consiste en la extracción de un color de la imagen para reemplazar el área que ocupaba por otra imagen. Esto se logra al grabar el elemento protagonista sobre el fondo a sustituir, el cual debe ser de color uniforme (azul o verde, colores con menos presencia en los tonos de la piel humana) y mediante técnicas digitales se cambia por otra imagen o escenario.

c. METROPOLIS (LANG, 1927)

Este clásico dirigido por Fritz Lang se desarrolla en un futuro distópico en el que la tecnología y la opresión social se han vuelto dominantes. Su estética visual, su escenografía y sus admirables efectos especiales entre otras particularidades hacen que su influencia continue muy presente hoy en día, como, por ejemplo, en la última entrega del director Giórgos Lánthimos *Poor Things* (Lánthimos, 2023).

De estos efectos se encargó el director de fotografía Eugen Schüfftan, empleando una combinación de revolucionarias técnicas, entre los que se encuentra el "Proceso Schüfftan". Schüfftan inventó este sistema basado en espejos y el uso de la iluminación para crear ilusiones ópticas. De este modo, era capaz de conceder gran tamaño a las maquetas para posteriormente combinar estas imágenes con las de los actores, que habían sido rodadas por separado, haciendo que estos guardasen un tamaño minúsculo en comparación con los grandes escenarios. Esta método tan innovador permitió crear inmensos entornos urbanos de gran detalle.

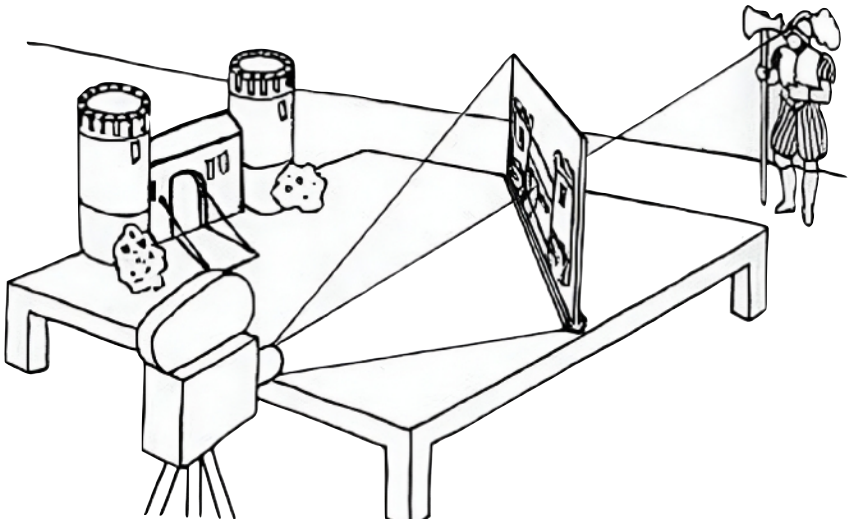


Figura 2. Diagrama explicativo del Proceso Schüfftan. "Proceso Schüfftan". (Fernández, 2013).

El procedimiento, como se muestra en la figura 2, consistía en un espejo inclinado con un ángulo de 45 grados entre la cámara y el escenario, en el que la maqueta o decorado quedaban reflejados. Se realizaba una abertura en el espejo de forma que los actores, que estaban detrás, eran registrados por la cámara, integrándolos así en la maqueta. Después se procedía a ajustar el tamaño del actor alejando o acercando su posición con respecto a la cámara.

Al igual que el cineasta Schüfftan seguramente tomó el truco del *Pepper's ghost*[8] como inspiración para crear su proceso, sucesores como el español Emilio Ruiz del Río tomaron su técnica como referencia. Ruiz mejoró el sistema pintando directamente sobre un cristal a través del cual se grababa la escena, logrando una perfecta adaptación del trazado con el escenario real. Indudablemente, cuando un efecto nace del ingenio la tecnología lo puede suplantar[9], pero siempre conservará su esencia.

Otra de las novedades técnicas que introdujo la película fue el uso, también por vez primera en el cine, de una cámara giroscópica[10], la cual permitía filmar panorámicas en 360 grados, es decir, en todas las direcciones. (Méndiz, 2012).

Estas son algunas de las muchas razones por las que se puede afirmar que *Metropolis* (Lang, 1927) fue más allá de la innovación técnica, demostrando el inmenso potencial de los efectos especiales para enriquecer la narración de historias y transportar a las audiencias a mundos nunca antes concebidos.

d. 2001: A SPACE ODYSSEY (KUBRICK, 1968)

Los efectos especiales se fueron perfeccionando hasta llegar a 1968, cuando se estrenó *2001: A Space Odyssey* (Kubrick, 1968), la cual no solo es considerada una película de referencia en el ámbito de la ciencia ficción, sino que supuso un punto de inflexión en la evolución de los efectos especiales.

Y debe varias de sus escenas más emblemáticas a algunas hazañas de la ingeniería, como, por ejemplo, los escenarios giratorios, para lo que Kubrick encargó la construcción de una centrifugadora de unos diez metros de diámetro.

Otra de las características con más trascendencia de la película es el uso del control de movimiento o *motion control*. *2001: A Space Odyssey* (Kubrick, 1968) fue pionera en este sistema, el cual se empleó para realizar las complejas escenas de las naves espaciales flotando en el espacio. La técnica permite un control del movimiento de cámara muy preciso, de manera que las tomas pueden repetirse las veces que sean necesarias imitando el mismo desplazamiento o ajustando el ángulo de la cámara según su objetivo. Hoy en

[8] Este truco creaba una ilusión de aparición y desaparición de un fantasma. A través de un cristal entre la escena y el espectador se reflejaba una habitación oculta para este. Cuando el lado izquierdo de la escena donde se encontraba el espíritu se oscurecía, no se reflejaba el fantasma, mientras que cuando se encendían las luces, este empezaba a aparecer gradualmente.

[9] Con la aparición del *chroma key*, esta técnica fue reemplazada, aunque directores nostálgicos como Peter Jackson la rescatarían años más tarde en películas como *The Lord of the Rings: The Return of the King* (Jackson, 2003).

[10] Cámara conectada a un sensor giroestabilizador que permite estabilizar la imagen al reducir los movimientos de balanceo no deseados durante la grabación.

día continua siendo una herramienta muy utilizada para tomas con efectos visuales, pero también en la industria de la televisión y otros sectores.

Esta puede ser una de las razones que explique la confusión cuando se dice que *2001: A Space Odyssey* (Kubrick, 1968) hizo uso de CGI, cuando en realidad no utilizó ningún gráfico por ordenador. El CGI tal y como se conoce en la actualidad no llegaría hasta años más tarde. Sin embargo, teóricamente, la primera vez que se presenció esta técnica en la gran pantalla fue en *Vertigo* (Hitchcock, 1958). En esta se llevó a cabo la primera animación hecha por ordenador realizada mediante la modificación de una computadora mecánica, llamada M5 Gun Director, presente en un sistema antiaéreo de la Segunda Guerra Mundial que originalmente servía para detectar y apuntar objetivos en movimiento. La máquina se adaptó mediante un péndulo y una plataforma giratoria para producir las famosas espirales de los créditos de apertura de la película. Este hito fue obra de John Whitney a quién se le atribuye la creación de los *motion graphics*.

Por otro lado, la proyección frontal (ver figura 3) es otra de las técnicas de las que también se considera pionera esta película. Este método permitió a Kubrick filmar la secuencia inicial que protagonizan los homínidos.

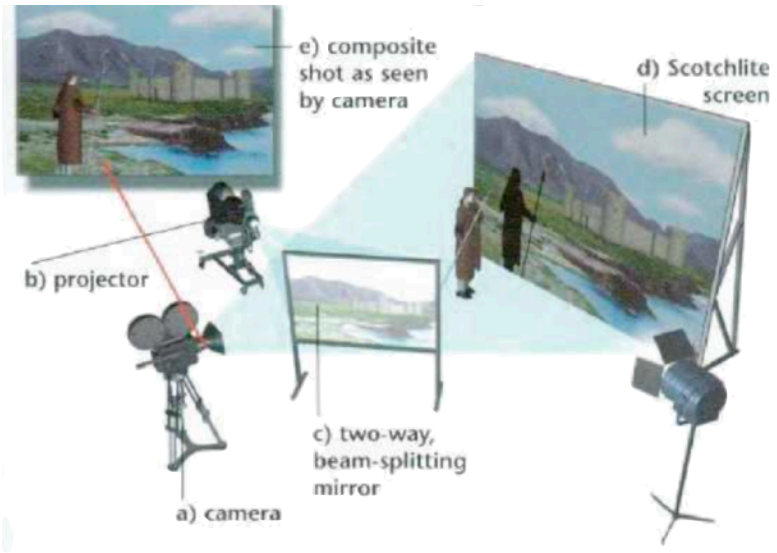


Figura 3. Esquema ilustrativo de la técnica de proyección frontal. (Rickitt, 2000).

El equipo de producción envió una segunda unidad de grabación a África a rodar los fondos, mientras que la escena final se grabó en los estudios de Inglaterra, convirtiéndose así en el primer filme de alto presupuesto en emplear este sistema. Consiste en la proyección de una imagen en un espejo bidireccional y semitransparente, con unas cualidades reflectantes específicas que permiten discriminar determinadas longitudes de onda, situado a 45º del proyector. Este tipo de espejos permiten reflejar un alto porcentaje

de la luz, que recibe una pantalla hecha de un material reflectante conocido como ScotchLite, de modo que la imagen queda proyectada en el decorado. Así, era posible utilizar tomas o secuencias previamente filmadas como fondo de la acción.

También se utilizó la proyección trasera o retroproyección como técnica de composición. Mediante una pantalla traslúcida y proyectores sincronizados con la cámara, el obturador se abría únicamente cuando el del proyector se cerraba, evitando así la "desaparición de la imagen" en la pantalla de retroproyección. De esta forma, la cámara registraba el conjunto de manera que pareciera que los actores estaban posicionados en el fondo proyectado. Este efecto se comenzó a usar con frecuencia para falsear el exterior por medio de proyecciones, como se hizo en *Terminator 2: Judgment Day* (Cameron, 1991) y *Aliens* (Cameron, 1986). El efecto evolucionaría en la retroproyección envolvente mediante el uso de pantallas LED, sistema del que se hablará más adelante.

Estos hechos ponen de manifiesto el requerimiento de especialización tecnológica por parte del departamento de producción, esencial para valorar la conveniencia y adaptación de este tipo de técnicas que permiten abordar los retos visuales que incluyen los guiones cinematográficos (García, 2013).

Por lo que, a pesar de no usar gráficos generados por ordenador, esta película supuso un punto crucial en la evolución de los efectos especiales y visuales, allanando el camino hacia un futuro en el que la tecnología se convertiría en una herramienta esencial en el ámbito cinematográfico. La atención al detalle de Kubrick tuvo un gran impacto en los cineastas que lo siguieron y las técnicas que inventaron aún se utilizan hoy en día, o al menos se imitan en formato digital. De hecho, es por el CGI y otros avances que cada vez menos películas necesitan hacer uso de ellas, pero si hubo un claro beneficiario de estos experimentos iniciales, ese fue George Lucas.

e. STAR WARS (LUCAS, 1977)

Los años 80 fueron la época dorada de los efectos prácticos, previa a la llegada del CGI. Sin embargo, diez años antes George Lucas concibió el universo de *Star Wars: Episode IV - A New Hope* (Lucas, 1977), lleno de extravagantes personajes y escenarios que, a partir de su imaginación, quiso plasmar en el guion. No obstante, crear estos elementos y adaptarlos para crear este universo fantástico no era tarea fácil, pues la mayoría de los departamentos de efectos especiales de los grandes estudios habían cerrado y los que aún estaban en funcionamiento no contaban con los materiales técnicos necesarios.

Ante esta ausencia, buscó un equipo de personas interesadas en hacer efectos especiales, un equipo que, ajeno a las limitaciones convencionales, explorase territorios desconocidos

en este ámbito. No importaban tanto sus áreas de conocimientos como la forma en la que se las ingeniaban para llevar acabo el trabajo, por lo que empezó a contratar mecánicos, artistas, ingenieros industriales, cámaras, físicos, maquetistas, fotógrafos…

Dada la complejidad requerida para abordar los efectos ópticos que el guion exigía, Lucas contactó con John Dykstra, que había participado en el diseño de las maquetas de *2001: A Space Odyssey* (Kubrick, 1968), para que se encargara de supervisar los efectos especiales de la película.

Figura 4. George Lucas operando una cámara Panavision PSR-200 en el rodaje de la primera entrega de *Star Wars*. (ILM, s.d.).

Una de las dificultades de este reto fueron las batallas espaciales, pues considerando la cantidad de elementos que se combinaban en cada una de las tomas, se necesitaba registrar siempre la misma perspectiva. Hasta entonces, este tipo de secuencias se habían realizado con modelos que requerían que la cámara se mantuviera fija y fueran estos los que se movieran hacia ella. Pero a este tipo de tomas, le faltaba el dinamismo que George buscaba, por lo que Dykstra, aplicando sus conocimientos de diseño industrial, creó el primer sistema de control de movimiento de cámara controlado digitalmente por un ordenador, al que denominaron: Dykstraflex (Espín, 2022).

Para ello, modificó una grúa de rodaje y la equipó con una cámara VistaVision[11]. Al conjunto se le conectó un conjunto de circuitos y una memoria RAM[12] (Random Access Memory). El objetivo era definir sus movimientos mediante un programa informático para poder repetirlos con exactitud y de manera indefinida. Múltiples cables conectaban

[11] Formato similar al *full frame* de la fotografía fija en negativo pero con ocho perforaciones por fotograma en una ubicación contraria a la del resto de formatos cinematográficos. Dysktra propuso rodar los planos con efectos en este formato dado que cuando se positivaban los negativos, la calidad de la imagen final se asemejaba más al material sin efectos, rodado en formato 35mm. (Aguilar, 2015).

[12] Memoria de acceso aleatorio destinada al almacenaje de datos temporales.

circuitos TTL[13] (Transistor-Transistor Logic) con servomotores[14] y un DEC PDP-11/45 (Programmed Data Processor), un ordenador con 8 registros de 16 bits que se encargaba de mandar las órdenes individualmente a cada uno de los motores.

De esta forma, era posible controlar digitalmente el movimiento de la grúa en siete ejes distintos, además de variar distintos aspectos de la configuración de la cámara, como el enfoque del objetivo o el control de los obturadores (López, 2017).

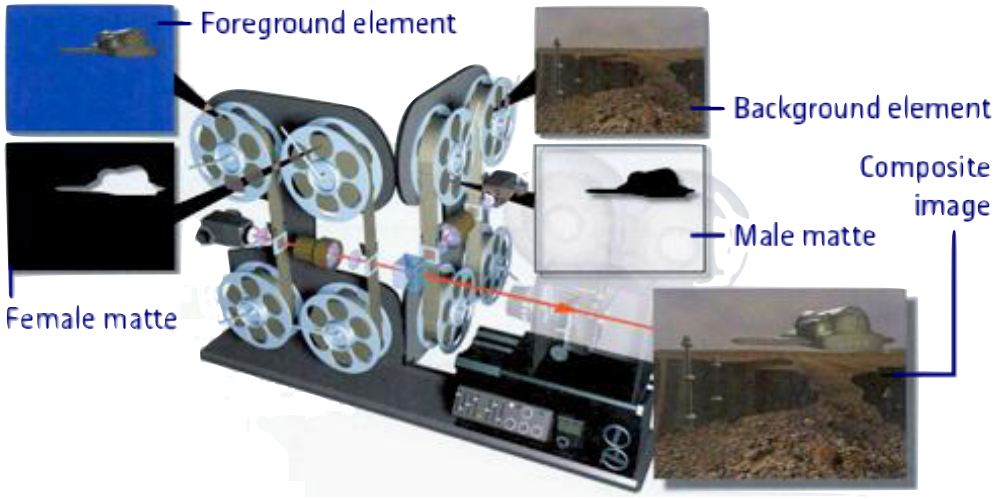


Figura 5. Ilustración explicativa del funcionamiento de una impresora óptica. (Ràfols, s.d.).

Esto facilitó la grabación precisa de movimientos para escenas que exigían mucha precisión, como es el caso de la escena de la batalla de la Estrella de la Muerte[15]. Posteriormente estas tomas se combinaban mediante una impresora óptica[16] que superponía los planos para simular un montaje en el que todas las tomas habían sido realizadas simultáneamente, es decir, en el mismo momento, componiendo así el cuadro final (véase figura 5).

[13] Se traduce como "Lógica Transistor a Transistor". En los componentes fabricados con esta tecnología los elementos de entrada y salida del dispositivo son transistores bipolares, realizan a la vez función lógica y de amplificación.

[14] Motor de corriente continua que permite controlar el movimiento de su eje para la gestión de su posición angular, aceleración y velocidad. Se emplean en aplicaciones que exigen variaciones rápidas de velocidad como la robótica industrial o sistemas de automatización.

[15] En el universo ficticio de Star Wars, se trataba de una estación espacial imperial utilizada para destruir de planetas.

[16] Dispositivo que permitía combinar diversas tomas para crear una composición de imágenes. Su funcionamiento se basaba en un proyector y una cámara enfrentados entre los que se disponía una o más capas de cintas de celuloide sobre las que se proyectaban las imágenes del proyector que, con el uso de máscaras, se imprimían únicamente sobre la parte de la cinta que quedaba al descubierto.

Sin embargo, no todos los efectos especiales de la película fueron innovadores, sino que se combinaron técnicas más modernas con otras más tradicionales, como es el caso de la rotoscopia, que resultó muy útil para integrar las "espadas láser" en la película.

La rotoscopia se basa en el uso de máscaras cambiantes para otorgar movimiento a un sujeto concreto y lograr la superposición de elementos digitales sobre imágenes reales. Originalmente se realizaba un trazado fotograma a fotograma hasta conseguir el efecto deseado, pero en la actualidad existen programas y softwares que permiten separar este elemento del fondo haciendo de este un proceso más sencillo que se conoce con el nombre de *motion capture* o *motion tracking*.

Por otro lado, en 1972 se habrían dado los primeros indicios de la incursión digital con la mano tridimensional que Ed Catmull[17] modeló por ordenador, mostrando todo su potencial a partir de los noventa. Basándose en el mismo proceso, Catmull creó maquetas 3D con el fin de simular la estrategia que seguirían los rebeldes para atacar la Estrella de la Muerte. (Navarro, 2020).

El responsable de diseñar los planos de esta nave fue Larry Cuba, uno de los pioneros en la animación computacional. Para llevar a cabo esta tarea, usó un lenguaje de desarrollo de gráficos vectoriales denominado GRASS (GRAphics Symbiosis System) que ofrecía herramientas específicas para la animación 2D, como escalado, rotación, movimiento, etc. De esta manera fue posible crear las líneas que formaban la malla del esquema de la Estrella de la Muerte en la sesión informativa dentro del cuartel de la Alianza Rebelde[18].

Para ello, primero se registraron las formas a replicar a partir de fotos de las maquetas, paso conocido como digitalización. Hecho esto, se almacenaron las estructuras en el equipo, para posteriormente, combinarlas y crear bloques o unidades. Una vez se almacenaron todos los vectores y puntos en la memoria interna, se dispusieron de manera consecutiva frente a una cámara para poder aportarles movimiento (López, 2017).

Otro de los métodos tradicionales que contribuyó a crear la estética y las ilusiones ópticas que caracterizan los efectos especiales de la película fue el *matte painting* (pintura "mate", "máscara" o "con enmascaramiento"). El *matte painting* se trata de una técnica de montaje utilizada para crear escenarios, principalmente, fantásticos. Deriva de otra técnica más antigua conocida como *glass shot* (plano con cristal) la cual consistía en colocar un cristal de grandes dimensiones frente a la cámara y pintar en este los elementos que se querían añadir a la escena, normalmente de compleja elaboración de forma física. Aunque esta

[17] Cofundador de Pixar y ex-presidente de Walt Disney Studios, realizó diversos descubrimientos en el desarrollo de los gráficos por ordenador. Fue contratado por Lucasfilm para liderar una división que trasladase los gráficos por ordenador a la industria cinematográfica.

[18] En el universo ficticio de Star Wars, fue un movimiento de resistencia formado por Bail Organa y Mon Mothma para oponerse a la supremacía del Imperio Galáctico.

técnica se popularizó mucho en el cine mudo, no resultaba sencillo trasladar el cristal hasta los exteriores o escenarios requeridos, por lo que se buscó una forma de fusionar el *glass shot* con el *matte shot*, un truco de exposición múltiple en el que se bloqueaba parte de lo filmado con una cartulina negra para impedir la exposición de la película en esa zona. Esta zona, enmascarada en negro, se cubría con la pintura del elemento a añadir en la escena y, después, el trazado y el escenario se combinaban mediante una impresora óptica (Lizcano, 2020).

Nació así Industrial Light and Magic (ILM), respondiendo a la carencia de otras compañías gracias a su capacidad técnica y creativa. El nombre representaba muy bien la dualidad del proceso creativo fusionado con el rigor técnico industrial y la "magia" que se obtenía como resultado. La producción de la película supuso un desafío caracterizado por la resolución continua de problemas técnicos, pero el proyecto consiguió salir adelante y le siguieron otras entregas como *Raiders of the Lost Ark* (1981, Steven Spielberg), *E.T.: The Extra-Terrestrial* (1982, Steven Spielberg) o *Back to the Future* (1985, Robert Zemeckis). Dykstra recibió el Óscar Científico y Técnico por el desarrollo del sistema de control de movimiento y sus múltiples contribuciones a la película.

Y es que a pesar de que los primeros gráficos digitales de ILM no ofrecían buenos resultados, con el paso del tiempo, la mejora fue notable, alcanzando un punto en el que fue posible introducir imágenes y gráficos digitales en las películas, hacer composiciones, pintar y hacer retoques desde el propio ordenador. A esta tecnología la denominaron Pixar, porque trabajaba con píxeles, y crearon un ordenador llamado Pixar Image Computer para las películas de ILM.

La repercusión que *Star Wars: Episode IV - A New Hope* (Lucas, 1977) tuvo en la integración del CGI supuso un gran adelanto en la historia de los efectos visuales impulsando la manipulación digital de imágenes. Así, estas técnicas continuaron desarrollándose hasta 1982, cuando se presentó *Tron* (Lisberger, 1982), considerado el primer largometraje en contener secuencias enteras de imágenes generadas por ordenador, para en 1989, potenciar la simulación de fluidos en *Terminator 2: Judgment Day* (Cameron, 1991).

f. JURASSIC PARK (SPIELBERG, 1993)

Era 1995 cuando Steven Spielberg en *The Making of 'Jurassic Park'* (Schultz, 1995) predijo que la tecnología del futuro sería "increíble" y en el ámbito cinematográfico, es su creación, *Jurassic Park* (Spielberg, 1993), la principal responsable del modo en el que se hacen las películas hoy en día.

La historia trata sobre un empresario que tras lograr crear dinosaurios reales con ingeniería genética, decide abrir un parque temático al que lleva, entre otros personajes, a una pareja de paleontólogos para contar con su aprobación antes de la apertura.

Aunque no era una tarea tan compleja como crear dinosaurios a partir de materia fósil hallada en mosquitos prehistóricos, llevar estas criaturas a la gran pantalla tampoco suponía un reto sencillo en aquella época. Al principio, el equipo encargado de dar vida a estas criaturas no tenía claro cómo iban a crear seres que el ojo humano nunca había presenciado. La esencia de la película era que los dinosaurios fueran realistas, por lo que Spielberg se planteó hacer casi todos mediante robótica, es decir, con maquetas mecánicas. Decidió poner esta tarea en manos de John Rosengrant y Shane Mahan, de Stan Winston Studios, autores de criaturas como el alíen de *Aliens* (Cameron, 1986), que se muestra en la figura 6, o los *terminators* de *Terminator 2: Judgment Day* (Cameron, 1991).

Figura 6. Ajuste de la maqueta del alien durante el rodaje de *Aliens*. (Stan Winston School, s.d.)

Lo cierto es que, como previó Steve 'Spaz' Williams, la idea de un tiranosaurio hidráulico a tamaño real era muy ambiciosa, pero sería imposible hacer correr un mecanismo de esas dimensiones (Volk-Weiss, 2019, min. 8-9). Dado que, como explica Mark A.Z Dippé, esta técnica solo es eficaz cuando se emplea en primeros planos (Volk-Weiss, 2019, min. 9-10). En los planos más abiertos o generales se planteó hacer uso de *stop motion*, para lo que Spielberg contactó con Phil Tippett, modelista y animador con experiencia en esta técnica. Tippett realizó unas animáticas previas a la película para medir tiempos y tener una idea más clara de los planos, proceso que hoy se conoce como *previs*[19] (previo). Sin embargo, uno de los defectos que tiene el *stop motion* es que no reproduce el desenfoque de

[19] Etapa en la preproducción de los VFX en la que se crean modelos de una calidad inferior y representaciones de los lugares donde se desarrollan las escenas en colaboración con el equipo de producción para configurar los ángulos de la cámara y preparar así las escenas más complejas con anticipación.

movimiento, dando resultados menos realistas para las escenas de acción que incluyen actividades como carreras o saltos, puesto que en la animación fotograma a fotograma, cada una de las imágenes es nítida y está bien definida.

Pero Tippett ya había hecho frente a este problema antes cuando inventó el *go motion* para algunas escenas de *Star Wars. Episode V: The Empire Strikes Back* (Kershner, 1980). Esta técnica llevaba el *stop motion* un paso más allá aplicando un ligero movimiento a los objetos que componen la animación con el objetivo de obtener un ligero desenfoque o *motion blur* y así obtener resultados más creíbles.

Por lo que Spielberg decidió acudir a Steve Williams que, junto a Dippé, pertenecía al elenco de Industrial Light and Magic, para encargarse de esta labor. No obstante, la idea no convencía a estos informáticos que creían poder aportar una forma más realista de dar vida a estas criaturas mediante gráficos por ordenador.

Williams empezó a modelar un tiranosaurio digital por su cuenta. Primero, modeló los huesos del forma ortogonal para conseguir un bloque de datos tridimensionales y, aunque surgieron contratiempos en el momento de imitar el movimiento del animal, consiguió crear la primera animación del esqueleto de un dinosaurio digital, consiguiendo una proeza técnica jamás realizada en la historia del cine (ver figura 7). Así que, en un visionado rutinario al que acudieron el director Steven Spielberg y la productora Kathleen Kennedy con la intención de ver una prueba del desenfoque de movimiento, aprovechó para mostrarles su animación del esqueleto en movimiento, ante lo que estos quedaron deslumbrados.

Figura 7. Steve 'Spaz' Williams modeló la primera animación del esqueleto digital de un tiranosaurio para *Jurassic Park* (Spielberg, 1993). (ILM 1991).

Poco después se le añadió piel a la criatura, a lo que, tras ver el resultado, Spielberg expresó "That's the future. That's the way it's gonna be from now on." [Ese es el futuro. Así será a partir de ahora.] (Volk-Weiss, 2019, min. 20-21). Y así el departamento de efectos especiales presenció como el destino de la animación *stop motion* se acercaba a su "extinción".

Por lo que una vez se decidió optar por la vía digital, los animadores de ILM trataron de adivinar cómo sería el comportamiento de los dinosaurios con el fin de hacer que se movieran de forma orgánica, para lo que Tippett, director de criaturas, fue de gran ayuda. El proceso para crear estas imágenes comenzaba con las referencias en papel, a partir de las cuales se construía un modelo tridimensional en el ordenador. El uso del software SoftImage 3D[20] fue imprescindible para ubicar las articulaciones, ajustar la perspectiva de cada fondo y establecer puntos referenciales entre los elementos digitales y el espacio real. A partir de estos datos, era posible construir *wireframes*, una especie de esqueleto o diagrama digital que actúa como guía para poder visualizar mejor la estructura de un elemento. Una vez definido el movimiento, se hacía uso del *software* Viewpaint para dotar de realismo a la piel a través de un mapa de textura y operaciones matemáticas. Posteriormente, se realizaba el proceso de renderizado[21] mediante el *software* Renderman, para lo que hicieron falta ordenadores de gran capacidad que pudieran soportar la potencia de cálculo que requería cada imagen.

Por último, se combinaban los elementos creados con las tomas reales por medio de un sistema de *tracking* (rastreo), que permitía a los animadores realizar el seguimiento de un elemento en una escena para registrar su trayectoria y posición en cada fotograma, técnica de la cual se sigue haciendo uso hoy en día. En total se crearon más de cincuenta efectos con CGI entre los que no solo se incluían dinosaurios, sino que, por ejemplo, para la escena de la lluvia se modificó la profundidad de campo para simular el reflejo del dinosaurio en los cristales de los coches (López, 2017).

Mientras, su versión animatrónica estaba casi terminada para el rodaje. Con seis metros de altura y casi tres toneladas de arcilla, su mecanismo se basaba en el chasis de un simulador de vuelo que se controlaba de forma hidráulica con un aparato de telemetría[22] al que denominaron "Wally". De modo que al interactuar con Wally, se recreaba el mismo movimiento en su versión más grande (ver figura 8).

[20] Software especializado desarrollado por Softimage Co. que más adelante se convertiría en Avid y la empresa adquirida por Autodesk.

[21] Generación de una imagen 2D a partir de modelos tridimensionales para convertir los datos virtuales en representaciones visuales realistas aplicando texturas y simulando la interacción de la iluminación con el entorno.

[22] La telemetría estudia la medición de diferentes magnitudes físicas de forma remota. A través de sensores permite recopilar datos para su posterior envío hasta el monitor del sistema.

Figura 8. Animatrónico del tiranosaurio durante el rodaje de *Jurassic Park* (Spielberg, 1993). (Stan Winston Studios, 1993).

Para esto se creó el DID (Dinosaur Input Device), una herramienta basada en una armadura cuadrúpeda con la forma del dinosaurio y cubierta de sensores de movimiento que permitía trasladar la animación *stop motion* a un modelo digital (Espín, 2022).

Todo fue según lo previsto y el sistema funcionó muy bien, hasta el momento de rodar la escena con lluvia. La piel del dinosaurio estaba fabricada con espuma de látex, un material muy absorbente que al humedecerse multiplicaría su peso hasta tres veces su carga inicial y, aunque todo el equipo fue advertido de que no debía mojarse, fue imposible evitarlo. De hecho, se hinchó tanto que también fue necesario ensanchar su versión digital para evitar fallos de *raccord* o continuidad, pues en esta escena se combinaron planos reales con planos digitales. Cuando se ve el cuerpo entero del tiranosaurio, es una imagen realizada con CGI, mientras que en los primeros planos, se trata del animatrónico. (Volk-Weiss, 2019).

Así, tras tres años de trabajo, la película se estrenó en todo el mundo, convirtiéndose en el mayor éxito de taquilla de la historia hasta ese momento. El público enloqueció ante las imágenes a ordenador y la película fue galardonada con el Óscar a los mejores efectos visuales. Pero no solo fue una buena película sino que cambió el modo de hacer cine para siempre. Pues antes de Parque Jurásico, crear un dinosaurio por ordenador parecía tan posible como hacer uno de verdad. (Volk-Weiss, 2019, min. 42-43).

Y es que la verdadera revolución del CGI llegó con los dinosaurios. Por primera vez los gráficos generados por ordenador compartieron pantalla con los actores, marcando un

antes y un después en la historia, que dejó atrás los efectos ópticos para dar lugar a la nueva era digital.

g. ACTUALIDAD

A este hito le siguieron otros como la técnica del *freeze time*, de la que se hizo uso en la película *The Matrix* (The Wachowski, 1999) para dar la impresión de que el tiempo se congelaba, el *motion capture* en *The Lord of the Rings: The Fellowship of the Ring* (Jackson, 2001) y su revolución con *Avatar* (Cameron, 2009), que consiguió capturar los matices de expresión y reacciones emocionales de los actores con una precisión sin precedentes; el desarrollo del sistema IMoCap (Image Based Motion Capture Technology), que implementó técnicas de seguimiento visual para la tripulación de *Pirates of the Caribbean: Dead Man's Chest* (Verbinski, 2006) o el uso del renderizado en tiempo real con Unreal Engine[23] y pantallas LED para crear escenarios virtuales.

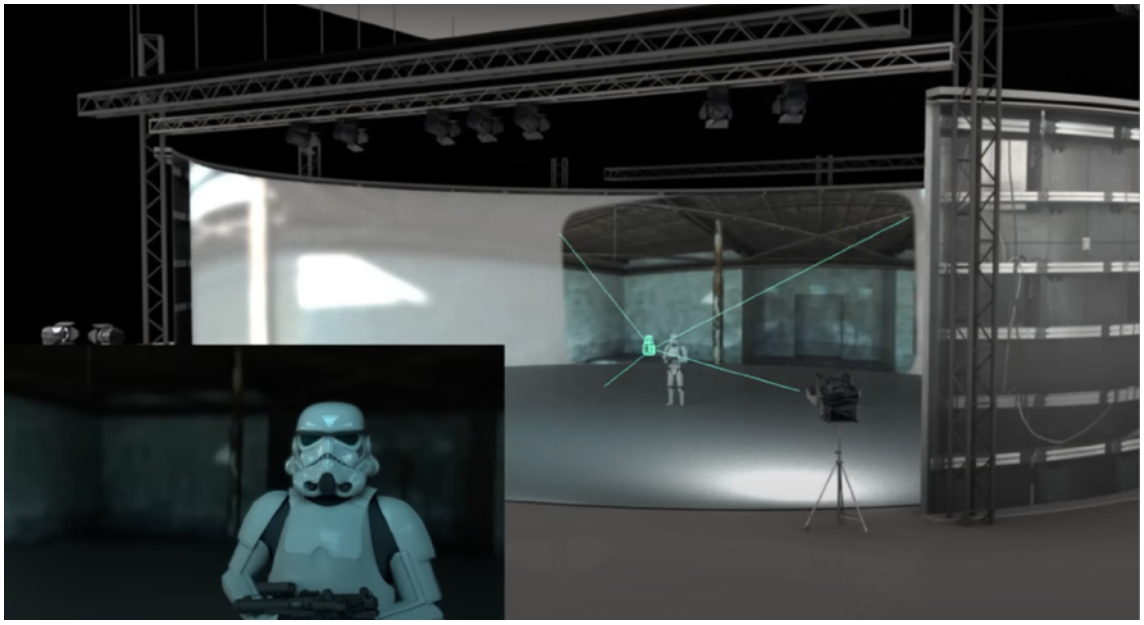


Figura 9. Imagen explicativa del cuadro renderizado en la pantalla según posición de la cámara. (ILM, s.d.).

En el presente ILM se posiciona como la empresa líder en efectos especiales y digitales del mercado audiovisual actual y encabeza los avances técnicos abriendo paso a sistemas cada vez más avanzados como es el caso del escenario virtual StageCraft. Inspirado en la retroproyección y el *chroma key*, Stagecraft ha proporcionado un enfoque revolucionario para la creación de entornos virtuales inmersivos. Su funcionamiento se basa en el uso de

[23] Motor desarrollado por la compañía Epic Games para la producción de videojuegos que permite a los desarrolladores de todos los sectores crear contenido a tiempo real de forma muy versátil.

pantallas LED de alta resolución y motores gráficos de renderización en tiempo real para crear fondos digitales que se proyectan durante el rodaje, mientras que un sistema de seguimiento en tiempo real ajusta la perspectiva del fondo según los movimientos de la cámara. La implementación de Stagecraft en el universo de Star Wars se estrenó con *The Mandalorian* (Favreau, 2019), permitiendo la creación de diferentes escenarios, desde desiertos hasta planetas imaginarios, con una eficiencia y flexibilidad creativa nunca antes vista.

Otro de los elementos destacados de esta tecnología son los *render nodes*[24] (nodos de renderización), los cuales reaccionan de forma natural al movimiento de la cámara en el set, reduciendo el riesgo de descuidos (ver figura 9). Además, permiten realizar modificaciones del set digital en tiempo real, como, por ejemplo, cambios en la iluminación. (Cazallas, 2019).

Este sistema no solo facilitó la creación de entornos preconfigurados con una calidad fotorrealista, sino que, junto a otras técnicas innovadoras, extendió los límites de la creatividad y narración visual en la industria cinematográfica. Su potencial es una muestra más de los avances tecnológicos que han contribuido a la evolución de los efectos visuales, cada vez más realistas y complejos. Y es que actualmente todos los productos audiovisuales, ya sean películas, series, videojuegos o publicidad, incluyen algún tipo de edición digital, haciendo de los ordenadores y softwares especializados herramientas imprescindibles en todas las fases de la producción cinematográfica.

En un mundo cada vez más inmerso en lo digital, los efectos visuales han experimentado un constante desarrollo que avanza a medida que surgen nuevas tecnologías, convirtiéndose en un elemento clave para contar historias a través de elementos visuales. (Calvo, 2023). Como expresa Navarro (2020): "El cine se ha convertido en el arte que avanza de la mano de la tecnología." Es por esto que los profesionales de este sector necesitan mantenerse actualizados para adaptarse a los softwares especializados que se vuelven más sofisticados y potentes con cada nueva versión.

Por otro lado, la creciente demanda de contenido visual ha hecho que este sector se encuentre cada vez más presente en variedad de aspectos de la vida cotidiana. Según Calvo (2023), basándose en datos extraídos del Libro Blanco "La industria española de la animación y de los efectos visuales" de DIBOOS (Federación Española de Asociaciones de Productoras de Animación) "actualmente esta disciplina cuenta con más de 250 empresas, genera el 20% del empleo total del sector audiovisual y su facturación nacional llega a los 900 millones de euros". Esto no sorprende, dado que además de sus ventajas, los VFX

[24] Componentes individuales que se interconectan como bloques para producir los efectos de renderizado. Los nodos de textura, de ubicación, etc. y sus conexiones (atributos) de entrada y salida definen todos los aspectos de las imágenes renderizadas. (Autodesk).

permiten abaratar costes al evitar la construcción de decorados o traslados a localizaciones remotas.

En este contexto nace la figura del Pipeline TD (Technical Director) para dar respuesta a la necesidad de un flujo de trabajo eficiente en la sucesión de tareas y labores necesarias en la realización de este tipo de efectos. A continuación, se analizarán algunas de sus funciones y rasgos principales.

3. EL ROL DEL PIPELINE TD EN LOS VFX

Cuando un efecto visual está bien hecho es invisible para el ojo humano, por lo que a veces puede parecer fácil de realizar, pero nada más lejos de la realidad. Lo cierto es que hacer un producto audiovisual es un proceso largo y complejo, basta con ver la suma de nombres que incluyen los créditos finales. Los efectos visuales conforman un campo muy amplio formado por cantidad de profesionales y equipos especializados en realizar diferentes materias y ramas que permiten elaborar este tipo de trabajos tan meticulosos. Se ubican dentro de la fase de postproducción[25], en la cual existe otra serie de divisiones organizadas en grupos, pero también se tienen en cuenta en las de preproducción y producción. Tanto para los efectos más fantásticos como para los más elementales, este conjunto de personas y tareas sigue un proceso metódico y ordenado en diferentes pasos que se conoce con el nombre de "pipeline".

El *pipeline* consiste en una especie de organigrama (véase figura 10) que representa el engranaje de departamentos y la sucesión de sus respectivas acciones. La coordinación y armonización entre estos es clave para obtener buenos resultados de forma eficaz. Este término, traducido como "tubería", asemeja metafóricamente el flujo de trabajo con el agua que fluye por esta. Para analizarlo se estudiarán las principales funciones de cada división en las etapas de la producción de los VFX, profundizando más en la de postproducción por ser esta en la que más presentes se hallan.

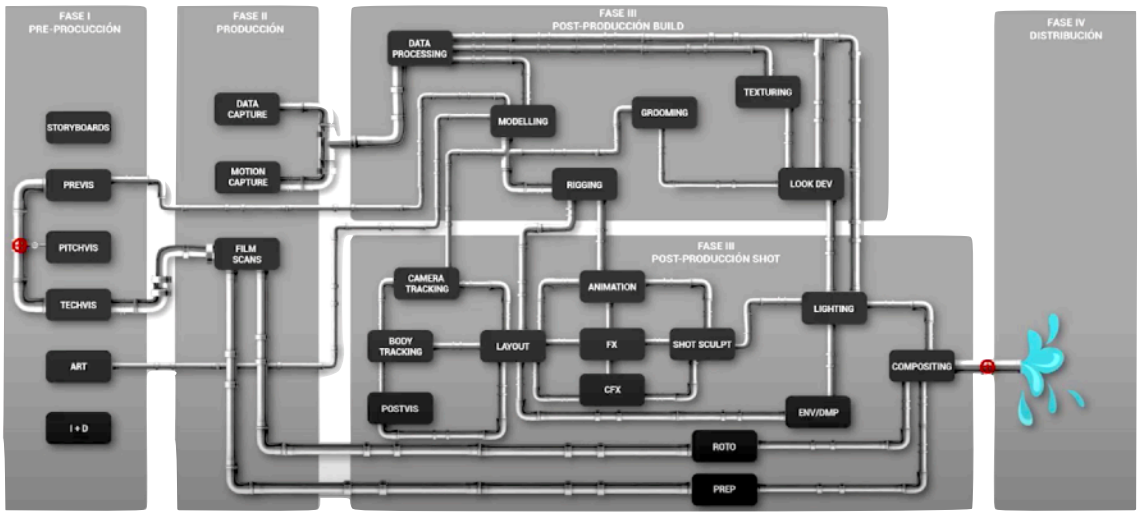


Figura 10. Ejemplo del esquema de un Pipeline. (Domínguez, 2022).

[25] En la creación de una producto audiovisual existen tres grandes etapas principales: la preproducción, la producción y la postproducción.

a. PREPRODUCCIÓN DE LOS VFX

La producción de los VFX comienza con los *storyboards*, viñetas que proporcionan la primera idea visual del producto plasmando el guion en imágenes conceptuales. Esto desemboca en el *previs* o previsualización, en el cual se crean versiones animadas de calidad inferior para configurar los ángulos de la cámara, organizar tiempos y preparar las escenas con anticipación.

En esta etapa se encuentra también, entre otros, el departamento de I+D, que incluye figuras más técnicas: físicos, matemáticos, ingenieros, programadores, etc. Este grupo multidisciplinar de profesionales estudia qué herramientas y medios serán necesarios para lograr ciertas tomas, adaptando los recursos existentes y desarrollando los que sean precisos.

b. PRODUCCIÓN DE LOS VFX

Aunque los efectos visuales se desarrollan principalmente en la postproducción, es imprescindible que parte del equipo acuda al rodaje para asegurarse de que la realización tiene en cuenta las necesidades referentes a las posteriores ediciones. Por ejemplo, los *data captures* capturan información útil para crear mapas 3D de iluminación y recogen parámetros de la cámara, como la distancia focal o el tipo de lente, a través de fotografías y escáneres.

c. POSTPRODUCCIÓN DE LOS VFX

El primer paso que se lleva a cabo en esta fase viene de la mano de los responsables del *data processing* que procesan, ordenan y limpian la información que llega del rodaje. A partir de aquí, tomando como referencia la clasificación de Ollé (2020), se pueden diferenciar dos grandes grupos: 3D y 2D. Ambos trabajan con imágenes a ordenador, pero los primeros de forma tridimensional y los segundos con imágenes planas.

3D

Al contar con percepción de profundidad, el ojo humano es capaz de procesar y percibir tres dimensiones a partir de una combinación de imágenes 2D, lo que se conoce como visión estereoscópica. La tecnología 3D permite crear este tipo de imágenes tridimensionales que pretenden representar o simular la realidad de la forma más fiel posible por medio de volúmenes, texturas, partículas y otros muchos recursos, reduciendo

el límite entre lo material y lo digital. Dado que este proceso conlleva una gran variedad de habilidades, se desglosa en diferentes ocupaciones o labores. Entre ellas se definen las siguientes subdivisiones y conceptos:

- **Modelado**: con el apoyo del equipo de arte, las *previs* y el material escaneado que reciben del *data processing*, estos especialistas se encargan de esculpir cada personaje, elemento o escenario que requiera la película, creando desde estructuras rígidas hasta formas más orgánicas a partir de geometrías o primitivas. La tecnología 3D permite crear este tipo de imágenes tridimensionales mediante algoritmos que realizan una representación matemática basada en vértices (*vertex*) con coordenadas espaciales conectadas por líneas, conocidas como aristas (*edges*), para formar caras (*faces*). Las superficies forman polígonos que a su vez conforman una malla (*mesh*) que permite visualizar el modelo tridimensional y a la que se puede aplicar todo tipo de operaciones y transformaciones (véase figura 11).

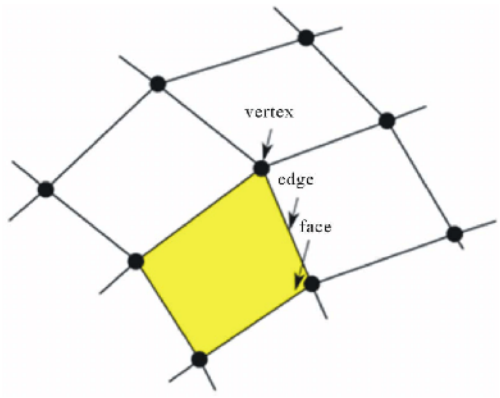


Figura 11. Elementos constructivos de un elemento 3D. (Kulovec, 2012).

Figura 12. Ejemplo de modelado con el personaje de Nala para *The Lion King* (Favreau, 2019). (MPC, 2020).

- ***Rigging***: los profesionales de este ámbito emplean esta técnica para crear estructuras deformables o esqueletos con controladores móviles que permiten a los animadores dar vida a la criatura en cuestión según su anatomía y morfología. Como explica Soler (2021), este esqueleto se compone de huesos (*joints*) conectados de acuerdo a una jerarquía basada en una relación padre-hijo en la que el movimiento de los primeros influye sobre los segundos.

- ***Grooming***: crean el pelaje y fibras capilares de los personajes o *assets*[26], desde el diseño de su apariencia hasta la simulación física cuando se interactúa con el mismo.

[26] Término que hace referencia a los recursos de un proyecto, ya sean un modelo 3D, una imagen u otro tipo de archivos.

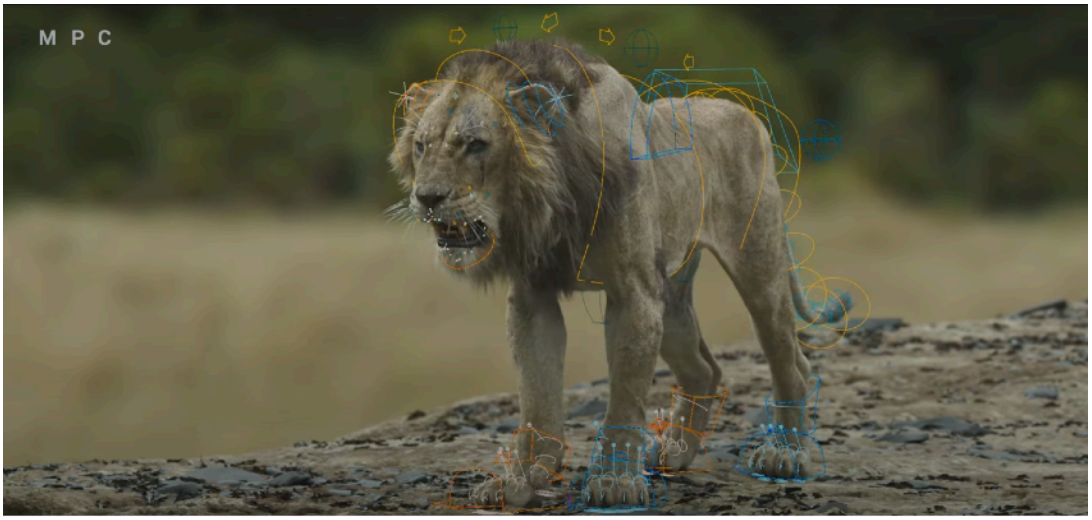


Figura 13. Ejemplo del *rigging* elaborado para el personaje de Scar para *The Lion King* (Favreau, 2019). (MPC, 2020).

• ***Shading* o texturizado**: desarrollan mapas de coordenadas UV[27] y mapas de textura con unas propiedades determinadas (reflexión, refracción, translucidez) que definen el comportamiento de la superficie donde se apliquen en función de la luz y sus parámetros físicos (amplitud, longitud de onda). Pueden imitar materiales ya existentes, como la madera o el cristal, o inventar nuevos a partir de bocetos a papel para crear texturas como la piel de un dragón.

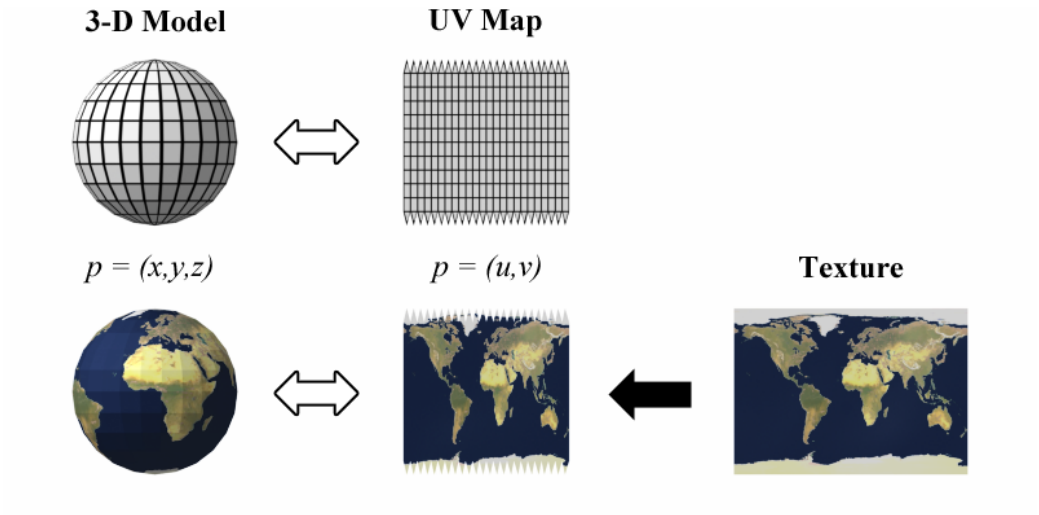


Figura 14. Imagen explicativa del proceso de aplicación de una textura UV. (Wikipedia, s.d.)

[27] El mapeado UV o *UV unwrapping* consiste en el proceso por cual se proyecta o desenvuelve (*unwrap*) la superficie de un objeto 3D para ser visualizada en un espacio bidimensional adaptando sus coordenadas (x, y, z) en solo dos ejes: U (eje x) y V (el eje Y).

• **Animación**: como su nombre indica, animan a los personajes o *assets* a partir del modelo con el *rigging*. Se trata de una tarea lenta y minuciosa que en ocasiones se realiza fotograma a fotograma. Su objetivo es dar vida a los seres modelados mediante acciones precisas, incluyendo gestos faciales y movimientos musculares casi imperceptibles.

• ***Lighting* o iluminación**: sin luz ninguno de los anteriores trabajos podría apreciarse. De esto se ocupan los *lighters*, actúan como directores de fotografía iluminando los diferentes planos de forma digital para lograr el mayor realismo y coherencia visual posibles (ver figura 15).

Figura 15. Ejemplo de la influencia del *lighting* en el personaje de Scar para *The Lion King* (Favreau, 2019). (MPC, 2020).

• ***Camera tracking***: como se mencionó con *Jurassic Park* (Spielberg, 1993), el rastreo de movimiento permite combinar los elementos creados por ordenador con las tomas reales. Con este fin, se genera una versión digital de la cámara del rodaje con sus mismas características para poder aplicarla en los efectos, de la misma manera sucede en *body tracking* o *match move*. Esto permite sincronizar los elementos virtuales con la cámara para registrar su trayectoria y posición en cada fotograma y así ajustar su movimiento en la escena.

• **FX o dinámica**s: fuego, líquidos o fluidos, partículas, humo, rayos, explosiones, multitudes, textiles… Este departamento crea simulaciones de todo tipo de fenómenos naturales mediante cálculos y parámetros físicos, trasladando esta tarea, que sería muy tediosa de realizar para los animadores, a los ordenadores. Por ejemplo, para animar una tormenta de arena, no es necesario configurar el movimiento de cada granito.

• ***Rendering***: una vez se obtiene el resultado deseado, se generan *renders* con la ayuda de motores de renderizado basados en algoritmos que permiten obtener una imagen plana a partir de un modelo tridimensional. Cada uno de estos *renders* formará parte de los *frames* que componen la película, calculados para cada instante en función de su iluminación, textura y demás características asignadas en las fases anteriores.

2D

Una vez renderizada la escena en 3D esta pasará a ser una capa bidimensional más que debe combinarse con el resto de imágenes y *renders*. De integrar estos dos tipos de contenidos es responsable el departamento de composición. Es necesario tener en cuenta que las imágenes 3D se generan por ordenador y, por tanto, en condiciones ideales, mientras que las reales están influidas por factores externos muchas veces imposibles de controlar. Por este motivo lo que se pretende con la composición de la imagen final es obtener un acabado homogéneo y equilibrado que engañe al ojo humano para que procese ambos tipos de datos como una única información: correcciones de color, distorsiones y desenfoques de cámara, etc.

Figura 16. Ejemplo de la influencia de la composición en un fotograma en *The Lion King* (Favreau, 2019). (MPC, 2020).

Además, se pueden diferenciar otros subdepartamentos como el de *roto*, que realiza el proceso de rotoscopia de forma digital mediante la creación de máscaras con el objetivo de separar elementos del fondo, o el de *prep*, que limpia y elimina todo lo innecesario o defectuoso del plano: cables, micros, personas, etc.

d. TECHNICAL DIRECTOR

Una vez comprendidas las fases y funciones de los VFX en las etapas de producción, se presenta el papel del Technical Director o TD. Tanto para artistas de Houdini que trabajan con VEX o Python, o de 3DS Max con MaxScript, Maya con MEL Script, y Nuke o Blender con *scripts*[28] de Python, esta figura se aplica.

Como explica McKay (2021), el ámbito de los VFX es un sector relativamente nuevo. Hasta hace unos años no existían cargos oficiales sino que a los responsables se les conocía como "generalistas 3D" y realizaban las diversas tareas anteriores - modelar, texturizar,

[28] Término empleado en programación para referirse a los fragmentos de código que constituyen el total de una aplicación o función.

animar, iluminar… - pero ninguna en específico. En torno al año 2007, Comet[29] publicó un artículo en el que desglosaba los diferentes roles esenciales en la producción de contenido 3D, haciendo mención de uno en concreto: el *technical director* o director técnico, que describió como un sujeto artístico familiarizado con el aspecto técnico y que englobaba, por tanto, lo mejor de los dos mundos. En aquel entonces, por un lado estaba el personal de I+D, que codificaba y programaba *scripts*, y por otro lado los artistas, de modo que hablar sobre un papel en un punto intermedio entre ambos suponía un gran avance (McKay, 2021).

En la actualidad existen estudios de diferentes dimensiones. Si bien uno de tamaño pequeño o medio puede contar con un único director técnico, uno grande como ILM, WetaFX[30] o Framestore[31] puede disponer de una serie específica de directores técnicos o TD especializados, como por ejemplo: un Character TD, en criaturas y personajes, un Lighting TD, en iluminación, un Cloth TD, en ropajes y textiles, un FX TD, en efectos, etc.

Los conocimientos técnicos de un TD le permiten crear *scripts* para automatizar tareas y desarrollar herramientas que optimizan el flujo de trabajo, proporcionando apoyo al resto del equipo ante la aparición de un posible problema o contratiempo. De modo que, básicamente, el *technical director* trabaja en sus tomas asignadas y cuando es necesario respalda al resto del departamento. Así nace el rol del Pipeline TD, centrado en mejorar la eficiencia mediante la resolución de problemas con la ayuda de códigos para crear nuevas funciones. Estas funciones o botones personalizados se añaden a la interfaz del programa y llaman a una secuencia de comandos que ejecuta el algoritmo respectivo.

De hecho, muchas veces se crean *tools*[32] específicas y únicas según las necesidades del proyecto y estudio en concreto. Primero se desarrolla el efecto a copiar y a partir de este se automatiza el proceso. Entonces se comparte esta nueva herramienta entre el resto de artistas júnior o de nivel medio para que puedan aprovechar las ventajas de esta sin necesidad de poseer conocimientos técnicos avanzados. Como resultado los estudios se ahorran el tener que contratar artistas muy experimentados y costosos, pudiendo optar por artistas de un nivel inferior y un director técnico que los asista (McKay, 2021).

---

[29] Michael Comet es un ingeniero de software y artista 3D que actualmente trabaja como Character Supervisor en los estudios de Pixar. El desarrollo del sistema cMuscleSystem (ahora conocido como MayaMuscle) y otros *plugins* para el software de Maya son solo algunas de sus aportaciones al sector de la animación.

[30] Estudio neozelandés de VFX fundado por Peter Jackson en 1993, hasta 2021 cuando fue adquirido por el desarrollador de videojuegos Unity. Está detrás de de producciones como *Avengers: Endgame* (2019, Russo) o las trilogía de *The Lord of the Rings* (Jackson, 2001-2003), con la que fue galardonado con el Oscar por sus efectos visuales y el desarrollo de software propio como el programa MASSIVE para la generación de masas.

[31] Estudio de animación y VFX con sede en Londres que fue fundado en 1986. Algunas de las grandes producciones en las que ha trabajado son, entre otras muchas, *The Dark Knight* (2009, Nolan) y *Harry Potter and the Deathly Hallows: Part I* (2010, Yates), ambas nominadas al Oscar por los mejores efectos visuales.

[32] Término inglés para referirse a las herramientas desarrolladas en VFX.

En resumen, la función de un Pipeline TD tiene como objetivo ahorrar tiempos y recursos haciendo que todos los departamentos trabajen en una misma dirección. Se aseguran de que cada departamento cuente con las herramientas de software necesarias para llevar a cabo su parte del proyecto, por lo que es fundamental que exista comunicación con los artistas de VFX a fin de conocer sus necesidades para poder abordarlas, tanto las más técnicas como aquellas relacionadas con la productividad. Por consiguiente, entender cada uno de los papeles en la tubería y comprender las dificultades a las que se enfrentan resulta una requisito imprescindible para ayudar a resolver problemas sobre la marcha o proporcionar apoyo técnico específico.

Esta labor conlleva tener conocimientos en lenguajes de programación como son Python, y VEX, además de saber manejar hábilmente los softwares de VFX, como Houdini, del cual se hablará a continuación.

4. HOUDINI DE SIDEFX

Cumplir con el *pipeline* sería imposible sin los softwares especializados que permiten hacer posible lo imposible. Cada uno de los departamentos vistos requiere una variedad de habilidades y, por lo tanto, diferentes prestaciones por parte de estos programas. En este contexto, Houdini se presenta como una solución para la creación efectos visuales en un entorno personalizable y sin necesidad de *plugins*, posicionándose como una de las aplicaciones más valoradas de su ámbito en la actualidad.

El nombre de Houdini proviene del ilusionista Erik Weisz. Nacido en Budapest, se mudó a Estados Unidos y allí adoptó el nombre artístico de Harry Houdini en honor a Jean-Eugène Robert-Houdini. Este mago francés era el dueño de un teatro de magia al que George Méliès solía acudir con frecuencia, y que finalmente acabó comprando. Allí combinó la cámara de cine con la magia del teatro, logrando una fusión única de ambos mundos. La filosofía detrás de este software está estrechamente relacionada con esta historia. Al igual que Méliès, Houdini busca aportar esa creatividad a la industria audiovisual, alcanzando, como expresa Gallego (2020), una conexión casi poética que resalta la esencia del programa: conectar tecnología con la magia del cine.

Houdini es un software avanzado de animación 3D especializado en la generación de gráficos procedurales. Fue desarrollado por Side Effects Software (SideFX), una empresa con sede en Toronto que inicialmente lanzó PRISMS (Production of Realistic Image Scene Mathematical Simulation), un programa pionero en la creación de imágenes mediante simulaciones matemáticas. Contaba con un conjunto de herramientas gráficas 3D que sentaron las bases para el desarrollo de Houdini y se diferenció del resto por su arquitectura. Houdini empezó a comercializarse en 1996 como una extensión de este y desde entonces se actualiza regularmente para mantenerse a la vanguardia y poder dar servicio a prácticamente todos los departamentos de una producción.

Aunque Houdini es famoso por su aplicación en la simulación de partículas y entornos dinámicos altamente realistas, también cuenta con un conjunto completo de herramientas para modelado, animación y renderizado. Se trata de un software muy potente con una estructura y características que le permiten soportar la cantidad de prestaciones que ofrece.

El rasgo más valorado y distintivo de Houdini reside en su naturaleza procedural y no destructiva. Esta cualidad, basada en el uso de nodos (ver figura 17), facilita la creación de activos de forma no lineal, lo que le aporta una flexibilidad y control superior en comparación a la de sus competidores. De este modo es posible realizar cambios de forma

rápida y crear modelos dinámicos que se remodelan según las necesidades del usuario, simplificando así el flujo de trabajo y mejorando la productividad.

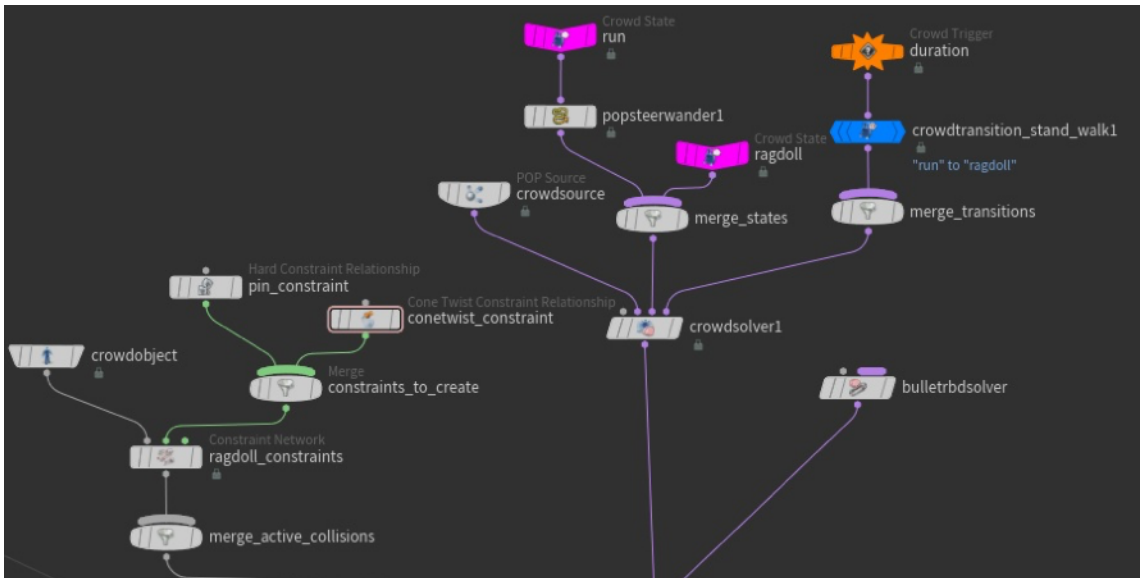


Figura 17. Ejemplo de una red de nodos en el *node view* o *network view* (espacio de trabajo de los nodos) de Houdini. (SideFX, s.d.).

Este enfoque procedural se fundamenta en la creación de un sistema para la generación de elementos, en lugar de únicamente la creación del mismo. En otras palabras, es posible diseñar un modelo para posteriormente controlar sus parámetros a través del software, lo que resulta muy práctico a la hora de modificar aspectos específicos sin la necesidad de empezar de nuevo. A este tipo de modelado se le conoce como "modelado procedural" y se estudia con más detalle en el siguiente punto.

Por otro lado, Houdini ofrece un entorno abierto para la programación a través de diversas API[33] (Application Programming Interface), con Python como el lenguaje predominante en la mayoría de los paquetes. Esto permite la automatización de tareas y la creación de *plugins*[34], haciéndolo extremadamente versátil (Petty, s.d).

Asimismo es posible declarar estas herramientas personalizadas como HDA (Houdini Digital Assets) o archivos digitales, para extender su uso a softwares externos u otros entornos como Unity o Unreal. Por ejemplo, en lugar de exportar manualmente, Houdini permite crear instancias, aplicar materiales e incluso introducir luces directamente en el

[33] Interfaz de programación de aplicaciones que define el lenguaje y los estándares para que los componentes de un software puedan comunicarse entre sí. Actúa como intermediario facilitando que las diferentes herramientas soliciten datos y acciones a sistemas independientes.

[34] Complementos que amplían las funciones de los softwares en los que se añaden. Permiten la personalización y adición de nuevas características en el programa original.

motor de juego, pudiendo ver los diseños renderizados prácticamente en tiempo real. (Bragagna, 2023).

En cuanto al renderizado, por efecto utiliza el motor Mantra por su capacidad para procesar grandes cantidades de datos, aunque también es compatible con motores de terceros como Renderman. Además, si bien Houdini es especialmente útil para multitud de efectos, también permite componer escenas completas que integren estos efectos con animaciones realizadas en otros programas como, por ejemplo, Maya.

Considerando la multitud de servicios que ofrece, cabe destacar que su curva de aprendizaje es bastante pronunciada, puesto que para el análisis y simulación de fenómenos físicos es necesaria una base técnica en matemáticas, física y álgebra vectorial, entre otros, además de un buen manejo de los controles e interfaz del programa.

Sin embargo, Houdini no es el único programa que se utiliza en la industria. Entre los principales programas empleados destacan algunos como Nuke (Foundry). Nuke es un software especializado en composición que Phil Beffrey desarrolló en 1993 para la empresa de efectos visuales Digital Domain. Inicialmente se diseñó para uso interno como una herramienta para la ejecución de comandos, pero también permitía realizar transformaciones, difuminados (*blur*), aplicar filtros, y correcciones de color. A día de hoy, la eliminación de *cromas*, rotoscopia, *color grading* (corrección de color) y *camera tracking* son solo algunas de las posibilidades que ofrece. Establecido en los grandes estudios desde hace varios años, se ha convertido en una herramienta esencial para las producciones actuales, donde la composición digital es un proceso indispensable.

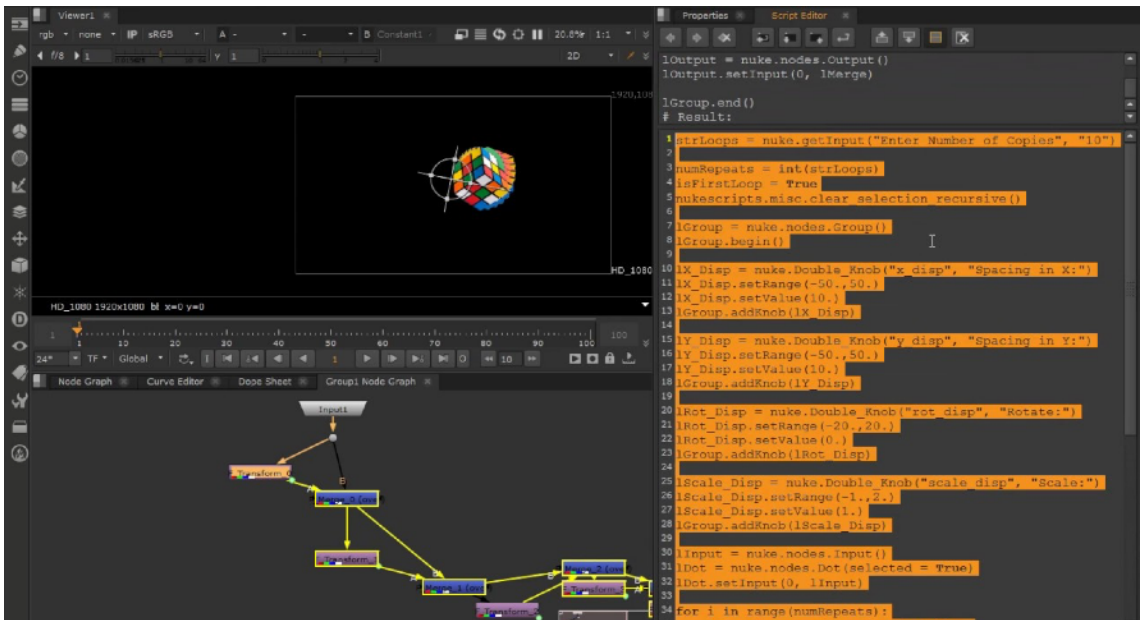


Figura 18. Ejemplo del uso de nodos y Python en la interfaz de Nuke. (Mathew, 2017).

Uno de sus competidores es After Effects (Adobe Creative Cloud). Sin embargo, una de las principales diferencias entre ambos es que este último funciona por capas mientras que Nuke trabaja con nodos. Este método nodal se ha establecido como el favorito de la industria por las ventajas que ofrece en la optimización del tiempo. De este modo, el espacio de trabajo se percibe de una forma más limpia, facilitando la lectura a nivel visual del historial de modificaciones. En consecuencia la gestión del trabajo en equipo se simplifica, lo que supone una gran ventaja dada la frecuencia de rotación de las escenas entre el personal.

Otra de las virtudes de la composición es el control artístico que proporciona en el renderizado. Pues aquellos planos con elementos CGI o tomas *full CG* no se renderizan simultáneamente, sino por separado, lo que permite modificar individualmente cada uno de los componentes de la escena para darles un tratamiento específico antes de reunirlos en la imagen final. Esto posibilita, como ejemplifica Gallego (2023), cambiar el color del pelo de un personaje o reducir la densidad de una nube de humo ahorrando tiempos.

Por su parte, Adobe After Effects se distingue como un software especializado en gráficos en movimiento, muy presente en televisión, publicidad, diseño, redes sociales y webs. Sus usos habituales incluyen el diseño de títulos, personajes y *motion graphics*, una técnica basada en la aplicación de movimiento a elementos tradicionalmente estáticos como logotipos, texto y otros componentes gráficos (Smith, 2024).

En cuanto al modelado, Maya (Autodesk) se presenta como el software por excelencia de su área para crear animaciones y modelados 3D. Tanto en videojuegos como en películas, con Maya es posible dar forma y vida a todo tipo de elementos: personajes, paisajes, mundos u objetos, desde formas simples hasta estructuras orgánicas más complejas.

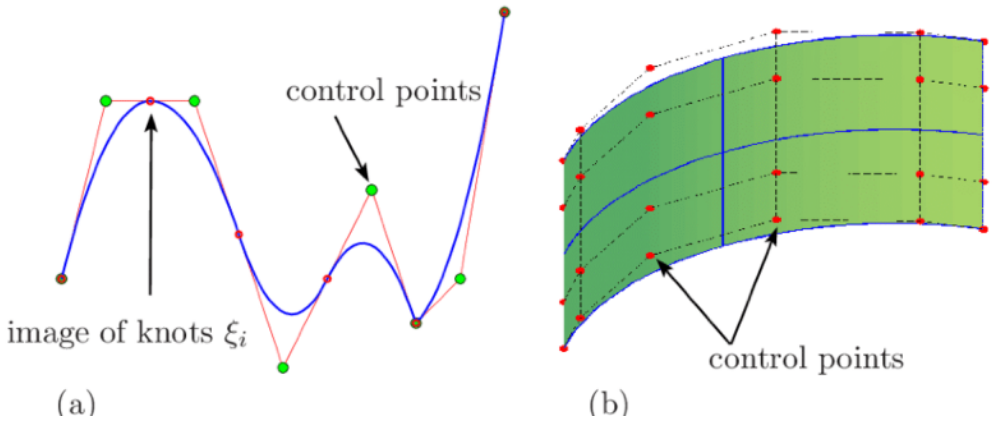


Figura 19. Esquema ilustrativo del papel de los puntos de control en una superficie basada en NURBS. (Nguyen, 2013).

El desarrollo de Maya llegó a finales de los 80 de la mano de Disney con el propósito de crear un programa con una interfaz de usuario amigable pero suficientemente potente para la película *Dinosaur* (Zondag, R & Leighton, E, 2000) (Soler, 2021). Desde entonces se ha convertido en uno de los softwares 3D más empleado en el sector del entretenimiento, la arquitectura y diseño de productos.

Una sus características más destacadas es el sistema de NURBS (Non-Uniform Rational B-Splines) que incorpora para el modelado. NURBS optimiza la creación de formas complejas gracias a la flexibilidad que ofrece para las curvas. A través de los manejadores y puntos de ancla y de control facilita la manipulación de superficies (ver figura 19).

NURBS fue desarrollado en 1950 por ingenieros que necesitaban un método de representación matemática para superficies de forma libre, con el fin de diseñar el revestimiento de coches, exteriores aeroespaciales y cascos navales, y reproducirlos con precisión en cualquier momento. Estos pioneros fueron Pierre Bézier, ingeniero en Renault, y Paul de Casteljau, ingeniero en Citroën. Bézier y Casteljau trabajaron en paralelo, pero de manera independiente. Dado que el trabajo de Bézier fue publicado primero, su apellido se asoció a las *splines*[35], como *Bézier splines* o curvas de Bézier, mientras que Casteljau es más conocido por los algoritmos que desarrolló para la evaluación de superficies paramétricas (EcuRed, s.d.).

En la actualidad resulta un elemento básico en el diseño por ordenador (Computer-Aided Design o CAD), fabricación (Computer-Aided Manufacturing o CAM) e ingeniería (Computer-Aided Engineering o CAE). Sus herramientas también se integran en otros softwares de animación además de Maya, como Blender, 3D Max y Rhino3D.

También proporciona variedad de herramientas para el texturizado, el sombreado y la iluminación, que posibilitan el desarrollo de materiales y reflejos de alta calidad y realismo (ver figura 20). Para ello, integra Arnold, un software de renderización de iluminación global, como motor de renderizado. Además, su sistema de *rigging* simplifica la animación de personajes, abarcando desde la animación mediante fotogramas clave hasta la captura de movimiento. Es totalmente adaptable, lo que permite crear soluciones de *rigging* personalizadas según las necesidades específicas (Vanas, s.d.).

Por otro lado, el flujo de trabajo procedural de Houdini permite reducir costes y tiempos en la producción, por lo que no es de extrañar que grandes estudios como Disney, en películas como *Frozen* (Buck & Lee, 2013) o *Spider-Man: Homecoming* (Watts, 2017); HBO, para *Game of Thrones* (Benioff, 2011-2019); o Sony Pictures sean solo una parte del gran número de usuarios de eligen este software.

[35] Representación lineal definida mediante puntos de control que describen la misma curva y polinomios que interpolan su forma entre puntos.

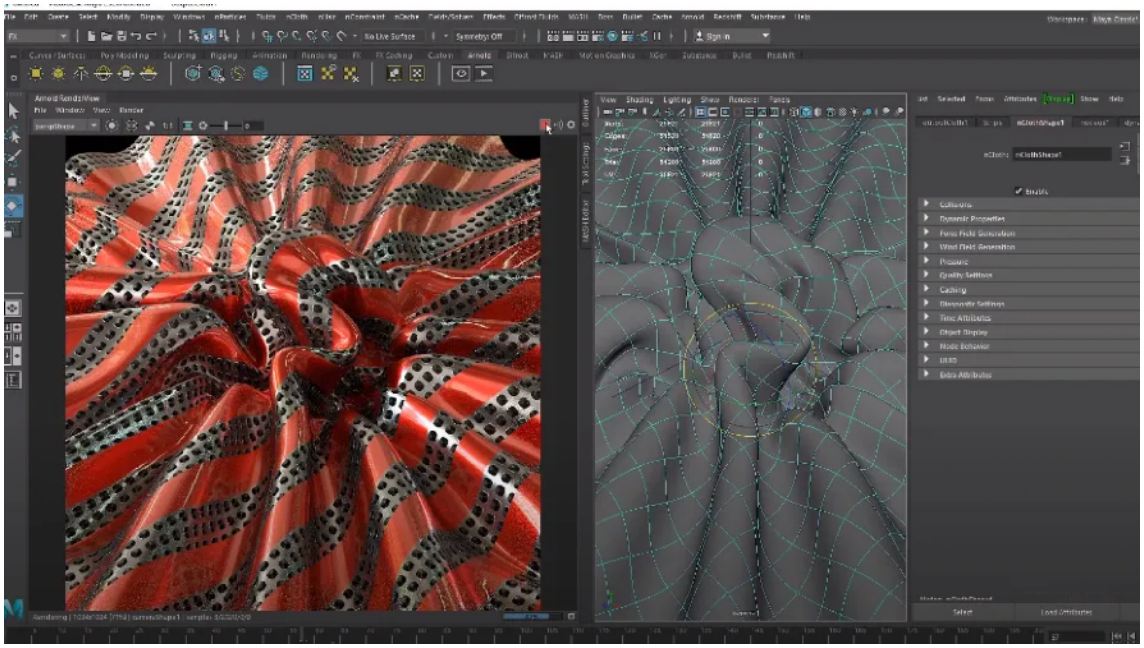


Figura 20. Ejemplo de texturizado e iluminación aplicados en Maya y renderizado con Arnold. (Serra, 2020).

Paralelamente, Houdini está muy presente en la industria de los videojuegos para la realización de cinematográficas en sagas como *Call of Duty* (Activision), *FarCry* (Ubisoft), *Uncharted* (Naughty Dog) o *League of Legends* (Riot Games). También en el ámbito publicitario, marcas como Kia, Lexus, Nissan o Nike eligen este software para las simulaciones complejas, demostrando así su versatilidad y potencial en los diferentes soportes y medios.

En definitiva, Houdini se ha actualizado a lo largo de los años para responder a las necesidades de sus consumidores hasta consolidarse como el software más relevante de la industria audiovisual actual. Houdini más que una herramienta se considera un conjunto de estas que ha conseguido transformar el *pipeline* y la forma de abordar los proyectos de hoy en día. Estas aplicaciones especializadas abren un amplio abanico de posibilidades que las vuelve casi tan esenciales como la propia cámara.

5. MODELADO PROCEDURAL

El modelado procedural se trata de un técnica digital basada en algoritmos y reglas predefinidas para la generación automática de contenido visual en lugar del modelado tradicional, basado en métodos manuales. Este sistema se fundamenta en la lógica y las matemáticas para, a partir de un conjunto de datos, generar contenido digital de forma eficiente y flexible mediante el ajuste de parámetros.

Aunque este método se comenzó a indagar en el 1960, no fue hasta 1990 cuando, debido a la expansión de los videojuegos y el avance de la tecnología, la industria del entretenimiento empezó a adoptar y desarrollar estos sistemas para la generación de gráficos. Sin embargo, fue en 2010 cuando realmente se popularizó con la llegada de softwares específicos que facilitaron la evolución de esta técnica y su inserción en multitud de sectores donde la repetición de patrones es común. (Torre, 2023). El modelado procedural permite crear contenido "aleatorio" para reproducir cantidad de elementos similares con variaciones definidas por una serie de pautas determinadas. Es decir, lo que hasta entonces se realizaba manual e individualmente, se automatiza con ayuda de algoritmos y principios específicos que describen cómo generar cada tipo de contenido, a diferencia del modelado tradicional.

Sin embargo, el modelado procedural no sustituye al clásico. El proceso de modelado, como se explicó en el tercer apartado, se lleva a cabo mediante operaciones - por ejemplo, extrusión de polígonos, divisiones o suavizados - que se aplican sobre una geometría base para reproducir cualquier objeto real en un entorno 3D. El modelado procedural utiliza este proceso como base inicial en su flujo de trabajo y posteriormente lo sistematiza con el fin de ahorrar tiempos y recursos. (Tejedor, E. & Martínez, J., 2019).

En Houdini, uno de los campos que se relaciona especialmente con este enfoque procedural es la simulación de físicas. Para ello, se trabaja con módulos especializados en diferentes dinámicas, como por ejemplo dinámicas de cuerpos rígidos, de cuerpos blandos, fluidos o textiles. (Peña, 2024). Es por esto que, como expresa Torre (2023), el modelado procedural: "está a caballo entre la programación y el diseño, ya que mezcla los procesos creativos tradicionales con procesos iterativos informáticos y secuencias matemáticas para generar contenido mediante algoritmos."

En cuanto a usos y aplicaciones, entre sus más habituales se encuentra la creación de entornos. Principalmente en videojuegos supone una herramienta de suma utilidad dado que la generación aleatoria de terrenos y escenarios permite ofrecer al usuario una nueva experiencia en cada partida o sesión. También se emplea en la producción de texturas y

shaders, mediante la creación de máscaras a partir de un conjunto de datos, como ruido o imágenes, que se mezclan matemáticamente para obtener texturas personalizadas según las exigencias del producto. (Tejedor, E. & Martínez, J., 2019).

Para comprender mejor el funcionamiento del modelado procedural con un ejemplo práctico, Tejedor, E. & Martínez, J. (2019) asemejan el modelado tradicional con la creación de una galleta, mientras que con la técnica procedural se obtendría la fábrica de estas, a las cuales se les asigna una serie de parámetros con cualidades variables. De esta forma, algunas galletas tendrán mayor o menor radio, mayor o menor grosor, más o menos pepitas de chocolate, un color más claro u oscuro, etc.

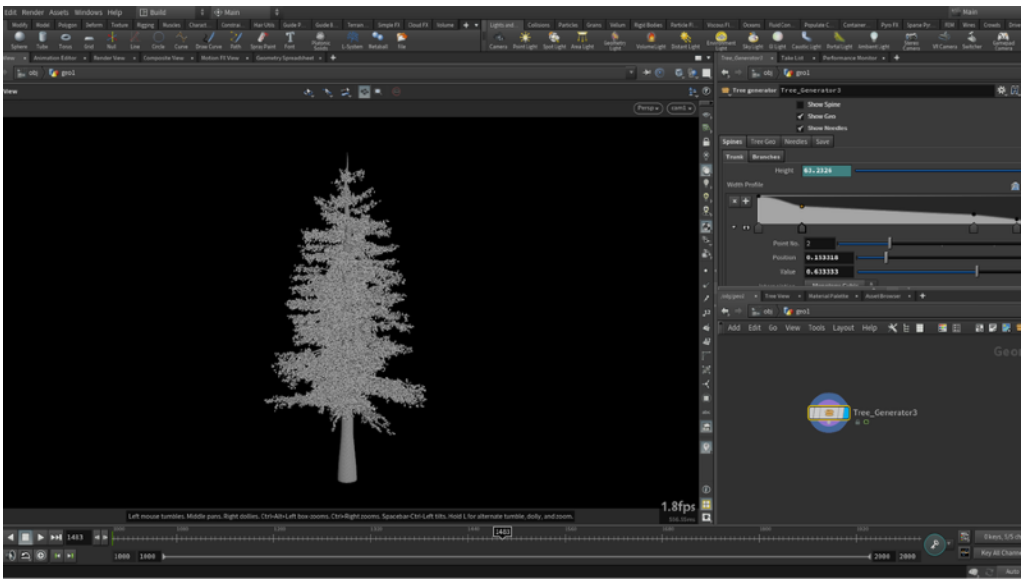


Figura 21. Ejemplo del *viewport* de un generador procedural de árboles en Houdini. (Laszcz, 2020).

Dicho de otra manera, Torre (2023) ilustra este concepto de la siguiente forma: un bosque de árboles con diferentes ramas, alturas u hojas, cada criterio con valores dentro del rango asignado (ver figuras 21 y 22). En términos de programación orientada a objetos, un árbol creado manualmente sería considerado un "objeto", mientras que el modelado procedural del bosque sería la "clase" que define esos objetos. No obstante, en el séptimo apartado se expone con detalle el desarrollo de un modelado procedural con VEX.

Como se puede observar las ventajas de este tipo de modelado son indefinidas. Entre ellas destaca su naturaleza “viva”, en contraste con el carácter destructivo del modelado clásico. Este último no permite recuperar versiones anteriores a los estados almacenados como *backup*, mientras que el método procedural permite modificar las instrucciones dadas cronológicamente a través de los nodos, visualizando su aspecto en tiempo real.

Figura 22. Ejemplo de la variación entre diferentes árboles generados mediante modelado procedural en Houdini. (Laszcz, 2020).

Otra de las aplicaciones del modelado procedural con un futuro prometedor, es la generación de imágenes para el entrenamiento de las IA (Inteligencia Artificial). Aunque esta técnica lleva tiempo presente en el sector de los videojuegos y los efectos visuales también se incluye en otros ámbitos como la arquitectura y, últimamente, la generación de conjuntos de datos. Este método permite ampliar los recursos para IAs que precisan de grandes cantidades de información de una forma más sistemática. Es muy útil en aquellas basadas en redes neuronales y aprendizaje a base de patrones, como el reconocimiento de imágenes, que necesita numerosas imágenes etiquetadas para poder clasificar correctamente el objeto de estudio. El modelado procedural permite reproducir multitud de datos para mejorar la producción de estos modelos de forma rápida y cómoda. Cabe mencionar que, aunque las IAs generativas han mejorado notablemente su capacidad para crear contenido visual, el modelado procedural se mantiene como el líder en situaciones que requieren un alto grado de control y precisión para elementos específicos. En el futuro es posible que ambas tecnologías se integren, beneficiándose mutuamente. (Peña, 2024).

En resumen, la idea principal del modelado procedural reside en su versatilidad y escalabilidad para generar contenido variado de manera eficiente, ahorrando tiempo y recursos. Permite crear numerosos elementos a partir de un primer modelo, manteniendo siempre el control de los parámetros. Por ello, ha evolucionado hasta convertirse en una tecnología imprescindible dentro de la industria del entretenimiento.

6. PYTHON Y VEX: DIFERENCIAS Y FUNCIONES DENTRO DE HOUDINI

Houdini incluye la posibilidad de interacción a través de programación, herramienta que se ha convertido en un potente aliado para los artistas de VFX permitiéndoles trabajar desde una nueva perspectiva.

Ya en sus inicios Houdini incorporaba HScript, el lenguaje que utilizaba como "expresion language" por defecto. Previamente a la inclusión de Python y VEX, este era el lenguaje principal de *scripting*[36] en sus versiones anteriores. Aunque permite realizar variedad de funciones, la flexibilidad y potencia de estos nuevos lenguajes han terminado por sustituirlo en la mayoría de tareas. El mismo SideFX recomienda a sus usuarios que utilicen Python dada su mayor versatilidad y comodidad. A continuación se realiza una descripción de estos dos lenguajes, con el fin de definir para qué está indicado cada uno.

a. PYTHON

Python es un lenguaje de programación informático empleado en la creación de software, la automatización de tareas o procesos y el análisis de datos. Aunque Guido van Rossume empezó a desarrollarlo como un proyecto, se ha convertido en uno de los lenguajes de programación más relevantes en el mundo. Su simplicidad le ha permitido integrarse muy bien en variedad de sectores, entre ellos, el de los efectos visuales. La mayoría de softwares de esta industria, como Nuke, Maya, Blender, soportan la aplicación e interacción a través de este lenguaje.

Python se caracteriza por ser un lenguaje intuitivo y eficiente. Sus cualidades lo convierten en un lenguaje ideal para *scripting* y desarrollo de aplicaciones en cantidad de soportes y plataformas. Algunos de los rasgos más característicos que justifican su creciente expansión son los siguientes:

- Sintaxis sencilla: similar al pseudocódigo[37] y, por lo tanto, fácil de aprender y redactar.
- Lenguaje interpretado: no compilado, por lo que es más flexible e independiente.

[36] Los lenguajes de *scripting* están diseñados para ser interpretados mediante un programa o intérprete que procesa las instrucciones o comandos.

[37] El pseudocódigo describe los pasos que realizará el algoritmo a programar. Actúa como paso intermedio posterior al diagrama de flujos y anterior al código con la sintaxis del lenguaje de programación.

• Tipado dinámico: no es necesario especificar el tipo de variable al declararla, el tipo está ligado al valor. De modo que el tipo de variable puede variar según el valor que se le asigne.

• Tipado fuerte: lo que significa que para que una variable cambie su tipo una vez declarada se debe realizar una conversión explícita. Al contrario que con el tipado débil, el lenguaje es más estricto al operar entre distintos tipos de datos.

• Orientado a objetos: basa su estructura en clases (que actúan como "plantillas") y métodos (que definen el "comportamiento" de estas) entre otros elementos reutilizables para crear instancias de objetos con diferentes atributos y funciones.

• Multiplataforma y de código abierto: por lo que es totalmente gratuito y portátil o compatible con cualquier plataforma (Windows, macOS, Linux).

• Recursos: cuenta con gran cantidad de librerías con diferentes propósitos y funciones a la que cada día se suman nuevos módulos y bibliotecas que la comunidad aporta a este lenguaje.

En Houdini es posible desarrollar expresiones en Python para interactuar con los nodos y manipularlos, aunque su uso está centrado principalmente en la automatización procesos. Para esto, Houdini integra una API conocida como HOM (Houdini Object Model), que permite trabajar con Python en lugar de HScript. Incluye el paquete "hou" que contiene los módulos, funciones y clases que definen la interfaz. Este módulo se importa automáticamente al interactuar con el editor de parámetros y en la *shell*[38] de comandos Python. (SideFX). Existen varias formas de trabajar con Python en Houdini:

- **Python Shell**: ventana que sirve como intérprete de comandos para ejecutar pequeñas órdenes o scripts. Se puede acceder a ella como: Window > Python Shell.

- **Python Source Editor**: editor de código de Python que permite escribir código para guardarlo junto al archivo de la escena en la que se está trabajando (*.hip) para que al abrirlo se ejecute automáticamente.

- **Nodos**: aunque estos elementos se estudiarán con más detalle en adelante, otra forma de utilizar Python en Houdini es a través de sus parámetros. Para ello, se configura Python como lenguaje de trabajo en lugar de HScript.

- **Digital Assets**: como se ha mencionado, es posible acceder al editor de Python cambiando del lenguaje de un nodo a Python y configurar herramientas personalizadas

[38] Intérprete que permite al usuario interactuar con el sistema operativo a través de órdenes y comandos.

declarando un nodo, como *subnet*[39] para posteriormente: Click derecho en el nodo > Digital Asset (ver figura 23) > Create New. De esta forma, Houdini permite convertir el código en nodos reutilizables conocidos como activos digitales.

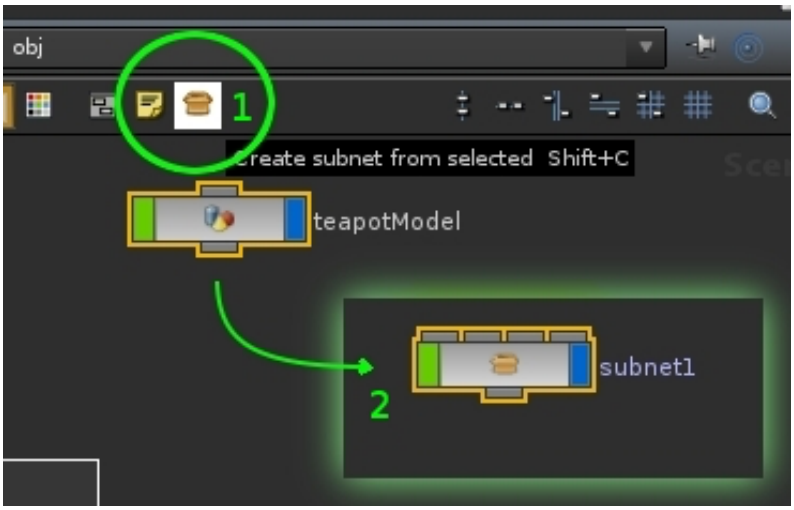


Figura 23. Imagen ilustrativa de los pasos a seguir para crear una *subnetwork* a partir de un nodo en Houdini. (Orbolt, s.d.).

- **PySide**: Houdini incorpora este módulo que permite la creación y personalización de interfaces dedicadas a la implementación de herramientas. De esta forma es posible desarrollar tools útiles y accesibles para más usuarios.

Como se puede comprobar, Python es un lenguaje muy presente en Houdini que se integra de forma óptima en el flujo de trabajo de los efectos visuales dada su versatilidad y sencillez. Para estudiar su funcionalidad de forma práctica, en el séptimo apartado de este trabajo se desarrolla una herramienta basada en este lenguaje.

b. VEX

VEX (Vector Expression Language) se lanzó en 1999, cuando Side Effects Software lo introdujo por primera vez en la versión 4.0 de Houdini. Desde entonces, se ha convertido en una parte integral de este software y se ha ido actualizando a lo largo de los años. Con las palabras de SideFX, se trata de un lenguaje de expresión de alto rendimiento y propósito general. Está orientado a la manipulación de geometrías, creación de *shaders* y otras operaciones que resultan más eficientes de realizar con líneas de código que con nodos.

[39] En Houdini, una *subnet* o *subnetwork* actúa como contenedor para simplificar visualmente redes complejas. Almacena un grupo de operaciones en un solo nodo en el *Network Editor*.

Aunque su estructura es similar a la de lenguajes basados en C, es específico de Houdini por lo que su arquitectura está ideada para trabajar sobre nodos. Es un lenguaje multihilo o *multithreading*[40] y, al contrario que Python, no necesita una API sino que existe una ventana llamada "VEXpression" en la misma interfaz de Houdini.

Por lo general, VEX consigue mejor rendimiento que si se realiza la misma tarea mediante un conjunto de nodos. (Jove, 2022). Sin embargo, muchos artistas de VFX desconocen la cantidad de posibilidades que otorga este lenguaje, cuando, como expresa Serebrennikova (s.d.), si no estás usando VEX, no estás utilizando Houdini en todo su potencial. Según la documentación de SideFX, algunas de los campos en los que VEX es aplicable son los siguientes:

- **Renderizado**: Mantra, el motor de renderizado de Houdini, utiliza VEX para los cálculos de *shading* y *lighting*, incluyendo efectos como niebla.

- **Composición**: es posible crear potentes nodos de composición personalizados (COPs) mediante VEX a una velocidad cercana a la de C/C++ y mucho más rápida que con los COPs estándar.

- **Partículas**: este lenguaje es muy útil en la manipulación de partículas, pues una sola función - unas líneas de código - en VEX puede realizar la tarea de muchos operadores de partículas (POPs). Dado que el código VEX se localiza en un solo operador, generalmente funcionará más rápido que una red de POPs, lo que se traduce en aumento significativo del rendimiento.

- **Modelado**: VEX también se utiliza para el modelado, especialmente cuando se trata de formas complejas con geometría atípica. Los VEX SOP permiten desarrollar herramientas para manipular los atributos de los puntos del modelo en concreto, de forma que es posible mover puntos, ajustar velocidades, cambiar colores, agrupar puntos y otras muchas tareas en menos tiempo que con los SOPs estándar.

En la web de SideFx se encuentra publicado un manual completo con información detallada para el uso de este lenguaje. No obstante, para el séptimo apartado, en el que se desarrolla un proyecto con este lenguaje, se incluye una breve explicación de algunos de los conceptos relevantes para su comprensión.

En conclusión, aunque la principal diferencia entre estos dos lenguajes reside en su función, el papel de estos dos lenguajes no tienen nada que ver. Python está más indicado

[40] La programación multihilo o *multithreading* permite ejecutar de forma simultánea e independiente varias tareas o flujos de control, logrando optimizar así la velocidad de cálculo y el rendimiento de la CPU.

para labores de *pipeline,* en las que se automatizan flujos de trabajo o procesos repetitivos (organizar escenas, conexiones con otros softwares, etc.). VEX se utiliza en tareas cotidianas, en el modelado 3D, para trabajar con partículas, *shaders* y los diferentes elementos de las geometría entre otros. VEX permite realizar una cantidad elevada de operaciones de una forma más limpia que con nodos o VOPs (Vector Operator, concepto que se comentará en adelante).

De hecho, también es posible manipular geometría con Python pero no es lo más recomendado dado que se volvería un proceso muy lento. Por lo que la combinación de ambos resulta ideal en el proceso de creación de efectos visuales.

7. DESARROLLO DE PROYECTOS CON HOUDINI

En el presente capítulo se ha realizado un desglose de los pasos seguidos en la realización de dos proyectos en Houdini a partir de los conocimientos adquiridos de la mano del maestro Miguel Ángel Arribas, uno basado en VEX y otro en Python, con el objetivo de mostrar de forma práctica la versatilidad y potencia de estos lenguajes de programación en un contexto audiovisual. Ahora bien, para comprender la terminología y conceptos aplicados en el desarrollo de este apartado, es necesario realizar una descripción previa de los fundamentos básicos que suponen la base de la metodología de trabajo en Houdini.

a. FUNDAMENTOS BÁSICOS

La unidad básica de geometría se conoce como "primitiva" y suponen la base de toda geometría o modelo en Houdini. Pero no son el único componente, se pueden diferenciar tres niveles principales:

- **Objetos**: Los objetos suponen el nivel superior en la representación de los elementos de la escena y están compuestos por todas las primitivas. Los modelos se construyen a partir de *Geometry objects* (objetos de geometría) que actúan como contenedores para los nodos que definen la geometría y acciones del objeto.

- **Primitivas**: Las primitivas representan una unidad de geometría. Se encuentran en un nivel inferior a un objeto y superior a los puntos. Existen variedad de tipos, entre ellos: caras de polígono, volúmenes, tetraedros, superficies Bézier/NURBS y sus curvas, etc. También es posible empaquetar una geometría como una *packed primitive* para referencias una geometría almacenada en otro lugar, lo cual es muy práctico para trabajar con más geometrías complejas. Los archivos *alembic*, por ejemplo, se representan en Houdini como primitivas empaquetados. Este tipo de archivos se estudiarán con más detalle en el desarrollo de una herramienta para su gestión.

- **Puntos y vértices**: Mientras que el concepto de "punto" comprende las coordenadas que definen un punto en el espacio (X, Y, Z, W), los vértices hacen referencia a estos. Es decir, las primitivas usan vértices para llamar a ciertos puntos. Las primitivas pueden compartir puntos, pero los vértices pertenecen exclusiva y respectivamente a cada primitiva. Por ejemplo, dos primitivas pueden tener vértices que compartan puntos en el espacio, como las esquinas de un polígono, pero no compartirán vértices.

Esta información se condensa en los **nodos**, la base del funcionamiento de Houdini. Los nodos son operadores que registran los parámetros introducidos en el programa, por lo que permiten deshacer modificaciones o versiones modelos anteriores. Almacenan cada una de las acciones que se realizan y las relacionan mediante conexiones. Estas conexiones forman redes de nodos organizadas en jerarquías, las cuales contienen el historial de decisiones que compone el resultado.

En la parte superior de estos árboles de nodos se encuentran las primitivas originales de las que se parte. Estas se conectan con las entradas de los siguientes nodos que modifican las primitivas, puntos, vértices o atributos que reciben del nodo anterior para generar una salida diferente. En el ejemplo de la figura 24, se parte de una malla o *grid* como primitiva inicial, a la que se le aplica un nodo *mountain*, para finalmente suavizar el resultado con un nodo *smooth*.

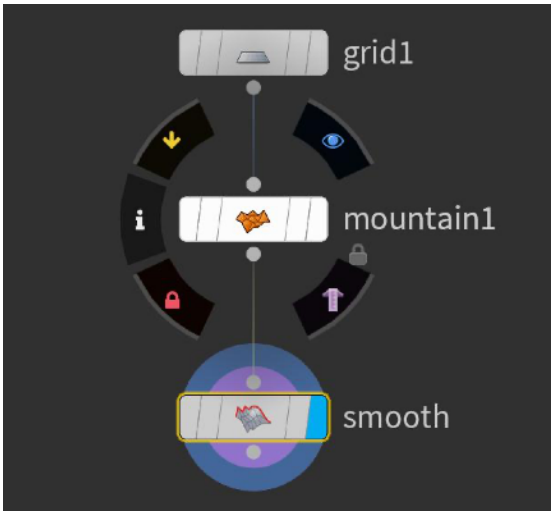


Figura 24. Ejemplo de una sencilla red de nodos para generar una montaña a partir de una malla. (SideFX, s.d.).

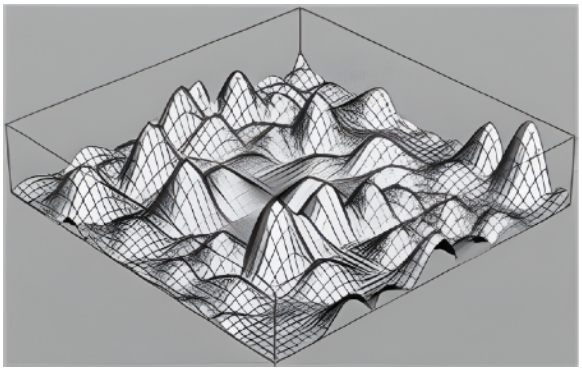


Figura 25. Ilustración explicativa del efecto de aplicar ruido a las normales de una malla. (Elaboración propia).

El comportamiento de estos nodos dependerá de los parámetros asignados por el usuario de forma manual, mediante referencias a otros parámetros o con funciones de VEX (Jove, 2022).

Por otro lado, los nodos tienen *flags*, una especie de etiquetas que, entre otras utilidades, permiten configurar qué clase de información se representa en el área visible o *viewport*. Existen diferentes tipos, como el de selección, para interactuar con el objeto desde el *viewport*; el de *display*, para mostrar el objeto en el *viewport*; el de *template*, visibiliza la estructura de la geometría aunque el display no esté activado; o el de *lock*, para congelar o paralizar el estado de ese nodo.

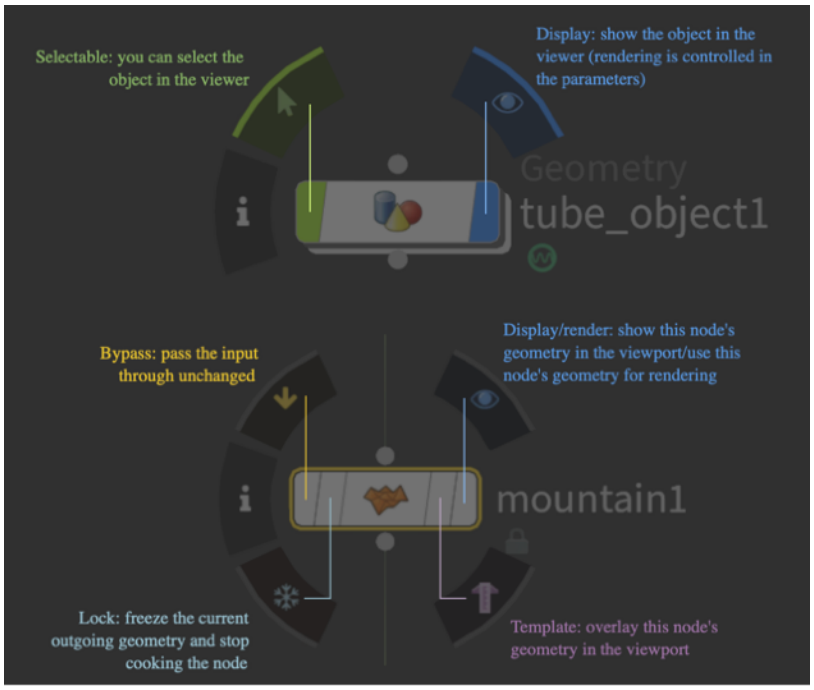


Figura 26. Flags de los nodos y sus funciones. (SideFX, s.d.).

De esta forma se consigue un flujo de trabajo no destructivo en el que los cambios no son permanentes, al contrario que ocurre con los destructivos y lineales. La ventaja es que estos historiales basados en parámetros pueden ser manipulados mediante *scripting*. También, es posible agrupar un conjunto de nodos en un único nodo conocido como HDA (Activo Digital de Houdini), que crea una interfaz desde la que es posible modificar la datos introducidos.

Además, se diferencian multitud de clases de nodos, entre ellos:

- **SOP (Surface OPerators)**: nodos de geometría que se emplean para crear y modificar geometrías. Cualquier tipo de geometría, desde polígonos hasta volúmenes.

- **DOP (Dynamic OPerators)**: estos operadores dinámicos o nodos de simulación se utilizan para llevar a cabo las simulaciones, se lee la información de los SOP y se traslada a los *solvers* de los DOP.

- **VOP (Vector OPerators)**: los operadores vectoriales permiten definir un programa, desde *shaders* hasta volúmenes, a través de la conexión de nodos dentro de la red. Posteriormente, Houdini compila esta red en código VEX ejecutable.

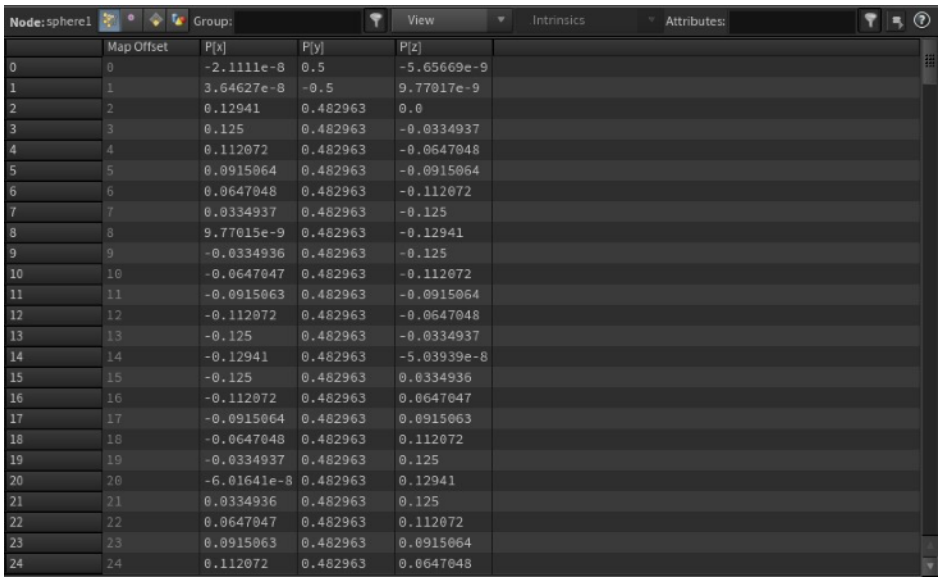
Los VOPs con útiles para visualizar en su interior un desglose de las variables y atributos que contiene el nodo en cuestión (ver figura 27). Es decir, con VOPs se pueden realizar las mismas o muy similares operaciones que con código en VEX, pero de una forma más gráfica. Aunque para cantidades más elevadas de operaciones o elementos, VEX se trata de un método más limpio y eficiente.

Como explica SideFX, la capacidad de guardar nodos y compartir información en forma de atributos a lo largo del flujo de trabajo es lo que le da a Houdini su naturaleza procedural. Los **atributos** son parámetros con un valor determinado que se almacenan en vértices, puntos, primitivas y objetos. Pueden ser de distinto tipo - vectores, *integers*, *floats* o *strings* - y es posible interactuar con ellos mediante expresiones de código. Algunos atributos son el color (Cd), la posición (P) o la normal (N), por ejemplo, aunque Houdini también permite crear atributos personalizados para los scripts.

Atributo	Nombre	Traducción	Tipo dato
P	Position	Posición	vector
v	Velocity	Velocidad	vector
force	Force	Fuerza	vector
age	Age	Edad	flotante
life	Life	Vida	flotante
id	Id	Id	entero
Cd	Color Difussion	Color	vector
uv	UV	uv	vector
N	Normal	Normal	vector
Time	Time	Tiempo	flotante
TimeInc	Time Increment	Incremento Tiempo	flotante
Frame	Frame	Frame	flotante
ptnum	Point Number	Número de punto	entero
primnum	Primitive Number	Número de primitiva	entero
vtxnum	Vertex Number	Número de vértice	entero
numpt	Points Number	Número de Puntos	entero
numprim	Primitives Number	Número de Primitivas	entero
numvtx	Vertices Number	Número de Vértices	entero
OpInput1	Operator Input 1	Operador de entrada 1	string
OpInput2	Operator Input 2	Operador de entrada 2	string
OpInput3	Operator Input 3	Operador de entrada 3	string
OpInput4	Operator Input 4	Operador de entrada 4	string

Figura 27. Atributos, nombres, traducción y tipos de datos de un VOP. (U21, 2022).

También existe un panel denominado Geometry Spreadsheet que muestra información detallada sobre los puntos, vértices y primitivas de la geometría seleccionada, como los valores de sus atributos y variables asociados. Permite consultar, en formato tabla, datos que no se pueden analizar de forma visual.



	Map Offset	P[x]	P[y]	P[z]
0	0	-2.1111e-8	0.5	-5.65669e-9
1	1	3.64627e-8	-0.5	9.77017e-9
2	2	0.12941	0.482963	0.0
3	3	0.125	0.482963	-0.0334937
4	4	0.112072	0.482963	-0.0647048
5	5	0.0915064	0.482963	-0.0915064
6	6	0.0647048	0.482963	-0.112072
7	7	0.0334937	0.482963	-0.125
8	8	9.77015e-9	0.482963	-0.12941
9	9	-0.0334936	0.482963	-0.125
10	10	-0.0647047	0.482963	-0.112072
11	11	-0.0915063	0.482963	-0.0915064
12	12	-0.112072	0.482963	-0.0647048
13	13	-0.125	0.482963	-0.0334937
14	14	-0.12941	0.482963	-5.03939e-8
15	15	-0.125	0.482963	0.0334936
16	16	-0.112072	0.482963	0.0647047
17	17	-0.0915064	0.482963	0.0915063
18	18	-0.0647048	0.482963	0.112072
19	19	-0.0334937	0.482963	0.125
20	20	-6.01641e-8	0.482963	0.12941
21	21	0.0334936	0.482963	0.125
22	22	0.0647047	0.482963	0.112072
23	23	0.0915063	0.482963	0.0915064
24	24	0.112072	0.482963	0.0647048

Figura 28. Ventana del panel Geometry Spreadsheet que muestra información sobre el atributo de posición. (SideFX, s.d.).

b. MODELADO DE UN SISTEMA DE GENERACIÓN PROCEDURAL CON VEX

La idea de este proyecto se basa en la simulación de una fractura en una pared formada por bloques de ladrillos. Aunque se trata de una sencilla representación sobre un muro, la estructura de este sistema se podría aplicar a la recreación del derrumbamiento de un edificio, por ejemplo. El carácter procedural de este modelo permite variar desde el número de ladrillos hasta la cantidad de fracturas a producir, entre otros parámetros. De esta forma, este *setup* puede adaptarse a los requerimientos de la producción audiovisual en la que se aplique.

Los pasos a seguir en el procedimiento quedan reflejados en el diagrama de flujo mostrado en la figura 29. El código completo para esta herramienta se encuentra recogido en el anexo A.

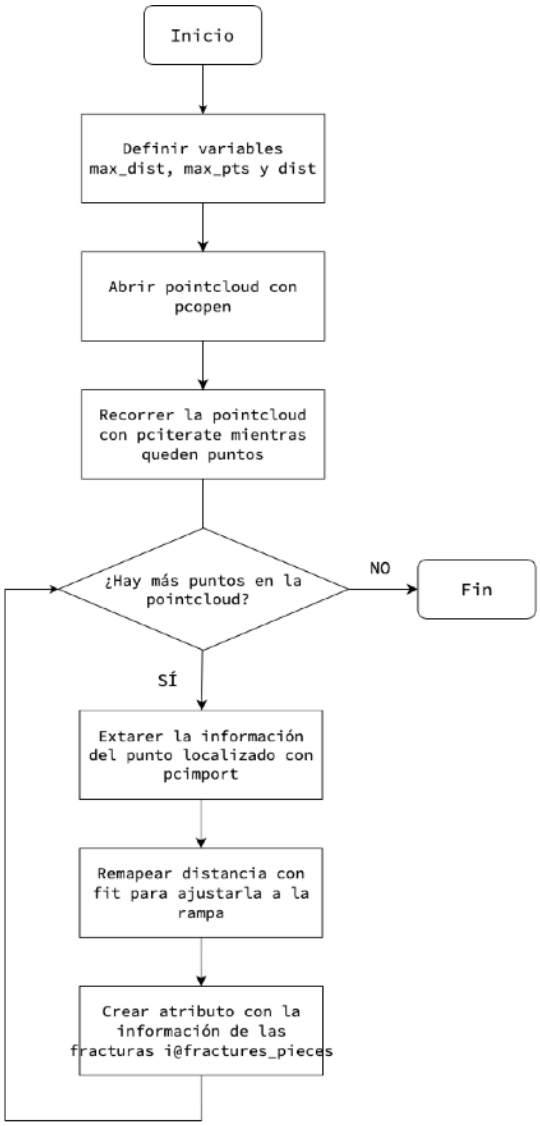


Figura 29. Diagrama de flujo del sistema procedural con VEX. (Elaboración propia).

Para empezar se debe crear la pared a derruir, para la que se parte de un primer bloque o ladrillo. Pero antes es necesario crear un contexto **geo** (ver figura 30, número 1) que actúe como contenedor para los nodos que definirán la geometría y acciones del sistema. Dentro de este, se añade un nodo **box** como primitiva raíz (ver figura 30, número 2) al que se le da forma de ladrillo modificando sus parámetros con un nodo **transform**, que permite realizar operaciones de translación, rotación y escala utilizando una matriz de transformación.

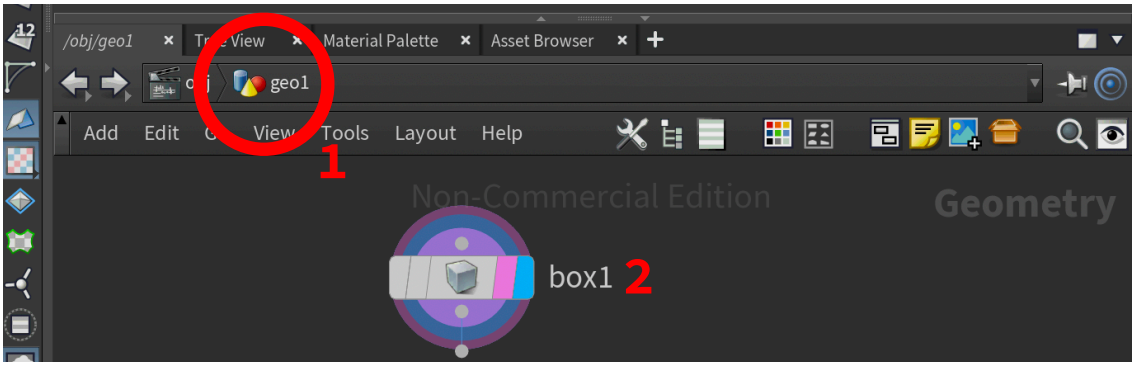


Figura 30. Nodo *box* dentro de un contexto *geo*. Elaboración propia a partir de (Arribas, 2019).

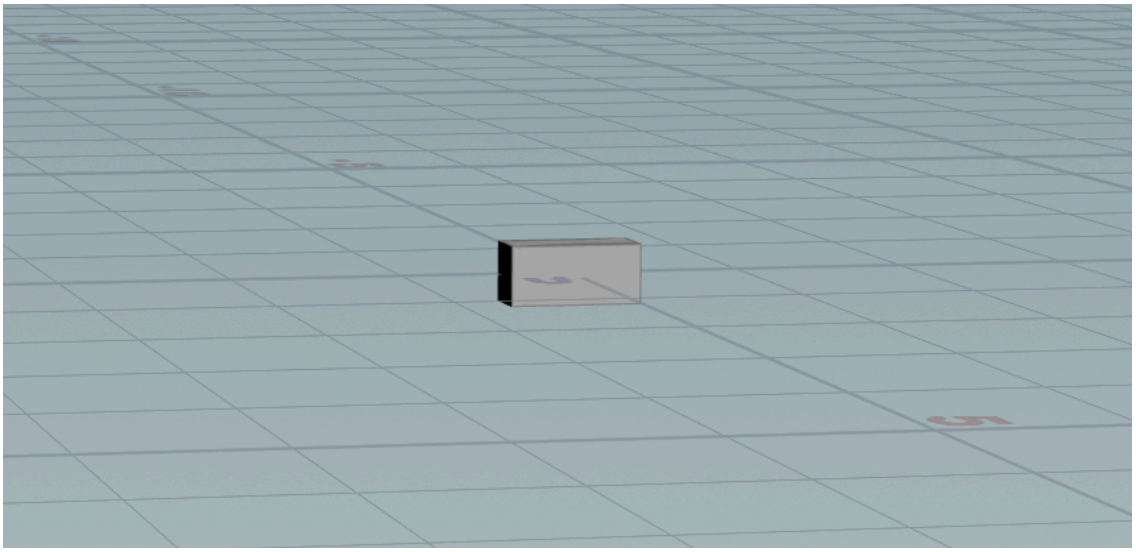


Figura 31. Resultado de la transformación para dar forma de ladrillo al primer bloque. Elaboración propia a partir de (Arribas, 2019).

Este primer bloque sirve de base para el resto, que serán copias sucesivas del mismo. Mediante un nodo **copyandtransform** se lleva a cabo la primera fila formada por 11 ladrillos (ver figura 32). Este nodo además de copiar las geometrías tiene en cuenta las transformaciones a las que han sido sometidas. Para evitar que las copias se solapen entre ellas, se tiene en cuenta un *offset* ligeramente superior a su tamaño en el eje Z.

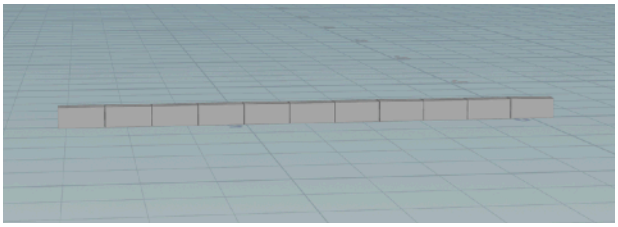


Figura 32. La primera fila de ladrillos resultado del nodo *copyandtransform* se muestra en la ventana del *viewport*. Elaboración propia a partir de (Arribas, 2019).

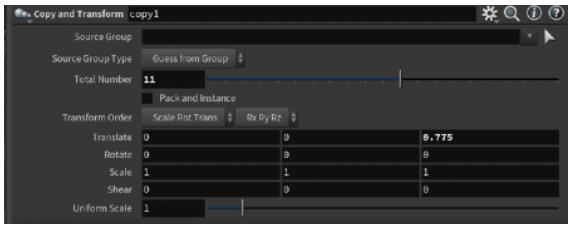


Figura 33. *Operator parameters* o parámetros de operación del nodo activo. Elaboración propia a partir de (Arribas, 2019).

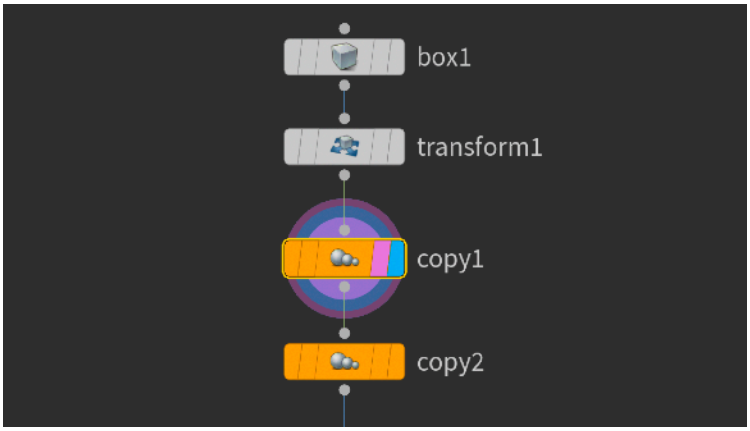


Figura 34. *Node/network view* para la gestión de nodos. Elaboración propia a partir de (Arribas, 2019).

Del mismo modo, pero en el eje Y, ocurre con el resto de filas, se multiplica esta primera un total de 11 veces (ver figura 35).

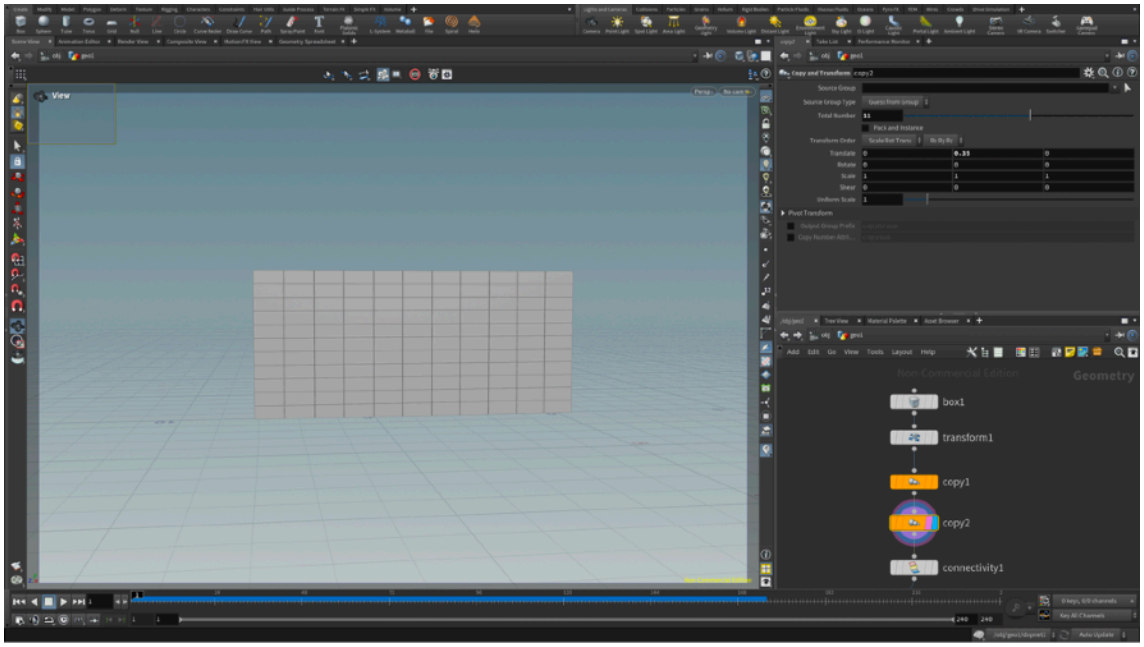


Figura 35. Resultado del segundo *copyandtransform* sobre la primera fila de ladrillos. Elaboración propia a partir de (Arribas, 2019).

El siguiente paso será añadir un nodo **foreach** para iterar por cada uno de los ladrillos. Este nodo actúa como un bucle *for* en programación y permite repetir un conjunto de operaciones un número de veces determinado, además de la posibilidad de modificar los parámetros durante cada iteración.

Por otro lado se crea un plano con un nodo **grid** (ver figura 36) que determinará el rango donde se producirán las fracturas. De esta forma, cuando más cerca se encuentren los ladrillos de esta, más roturas se producirán y viceversa, cuanto más lejos, menos o ninguna.

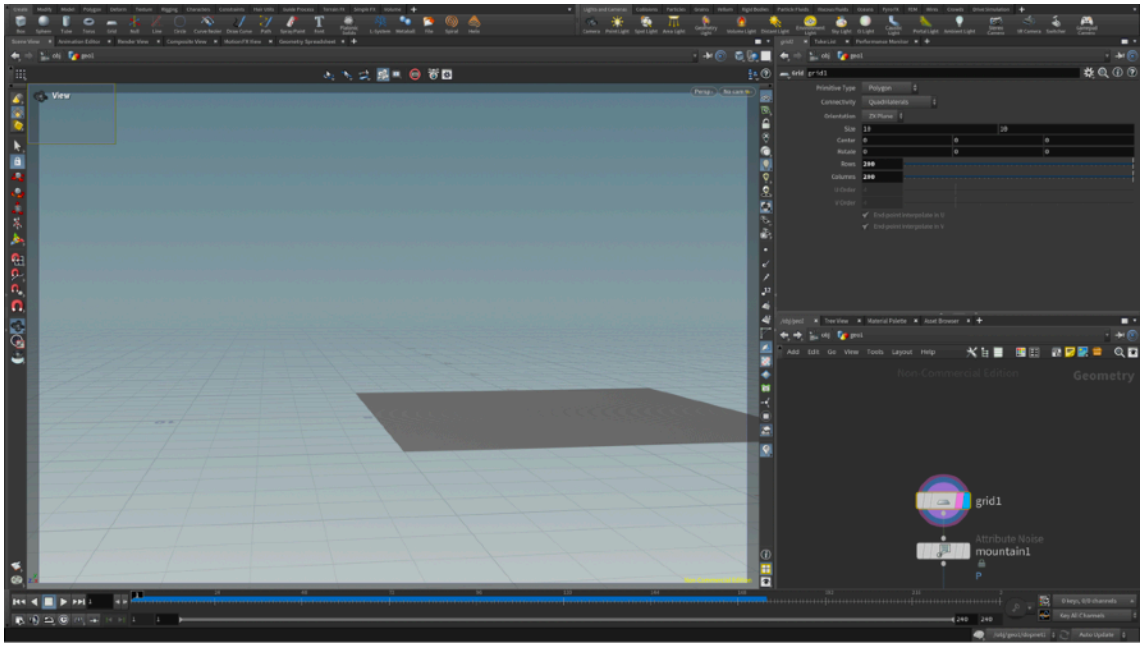


Figura 36. El nodo *grid* crea una geometría plana. Elaboración propia a partir de (Arribas, 2019).

A esta se le aplica un nodo **mountain**. Este nodo desplaza los puntos de la geometría a lo largo de sus normales produciendo ruido en la superficie (véase figura 37).

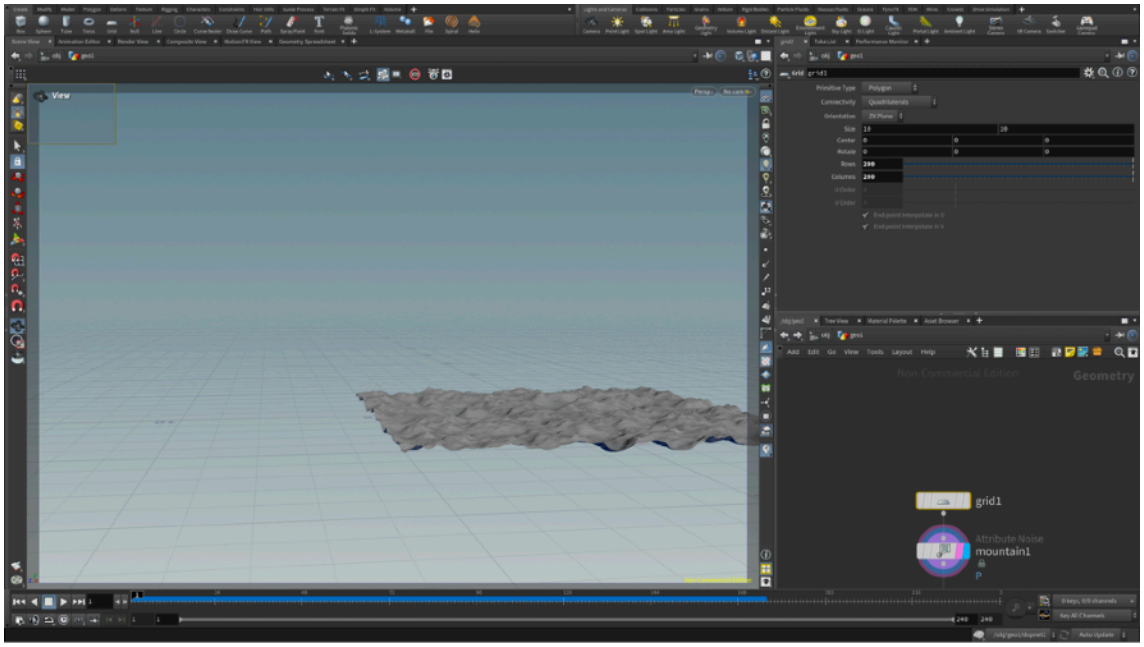


Figura 37. El nodo *mountain* introduce ruido sobre la *grid* provocando ondulaciones y distorsiones sobre la superficie. Elaboración propia a partir de (Arribas, 2019).

Una vez ubicada esta malla en la zona del muro donde se quiere dirigir la fractura principal, se le aplica un nodo **scatter** (ver figura 38) para distribuir puntos de forma aleatoria pero uniforma sobre esta.

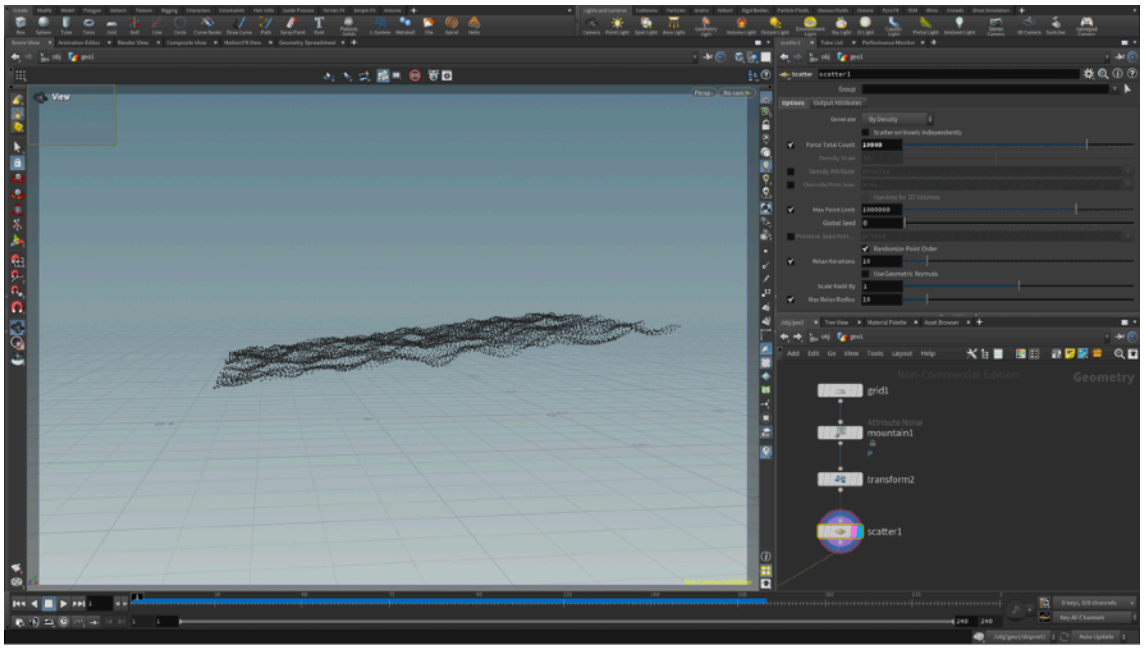


Figura 38. Resultado de aplicar el nodo *scatter* sobre la *grid* creando una nube de puntos. Elaboración propia a partir de (Arribas, 2019).

Volviendo a los ladrillos, se añade un nodo **wrangle** (ver figura 39, número 1). Es uno de los nodos más útiles de Houdini ya que permite recorrer y ejecutar código VEX sobre cada primitiva y sus elementos, como si se tratase de un bucle *for*. También posibilita la creación de atributos que faciliten la clasificación y filtrado de elementos escribiendo su nombre con @.

En la ventana "VEXpression" (ver figura 39, número 2) se introduce el código VEX que se ejecutará sobre la geometría entrante, en este caso, los ladrillos por la entrada o *input* 0 y la nube de puntos de la **grid** por el *input* 1. Primeramente se declaran las siguientes variables:

- **`max_dist:`** esta variable almacenará el valor de la distancia o radio máximo en el que se buscarán puntos de interacción con la **grid**. Se declara como canal (*channel*) de tipo *float*: "chf". Este tipo de funciones permite obtener el valor de un parámetro añadiendo un nuevo campo en la interfaz del programa. De esta forma, mediante el uso de canales de referencia, Houdini permite crear configuraciones personalizadas para controlar nodos dentro de la subred a través de la interfaz en el nivel superior (véase figura 40).

- **`max_pts:`** esta variable limitará el valor del máximo número de puntos encontrados a almacenar.

- **`dist:`** esta variable almacenará el valor de la distancia a la nube de puntos según la iteración en la que se encuentre el bucle.

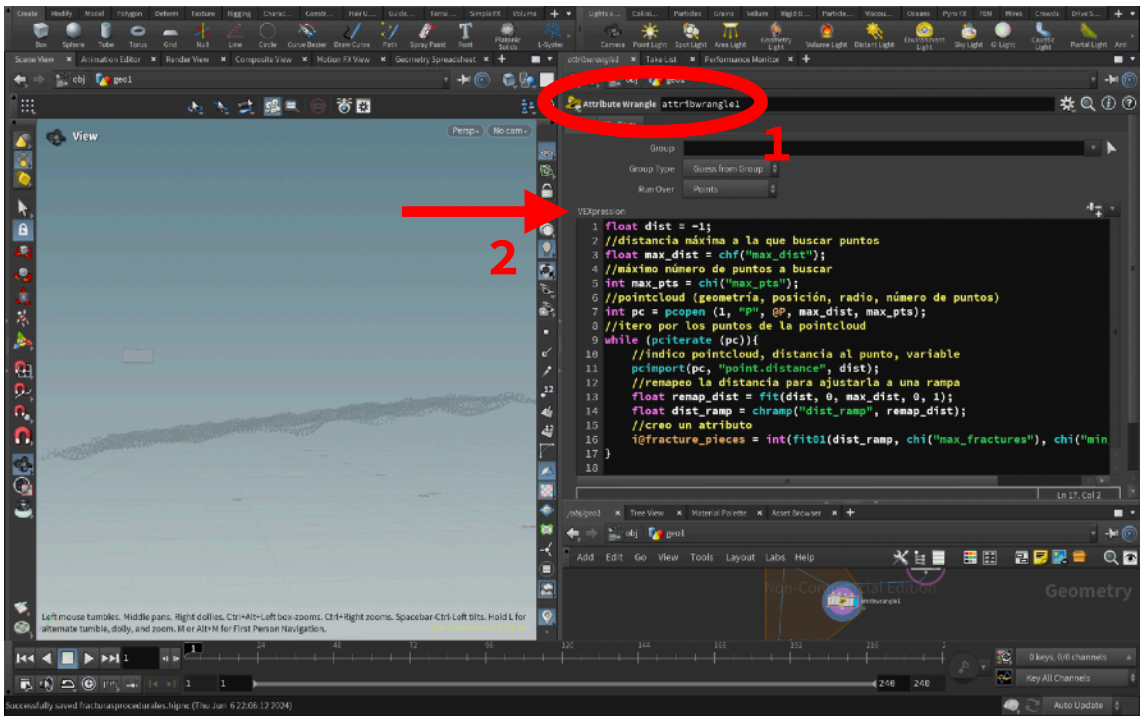


Figura 39. Nodo *wrangle* y el cuadro VEXpression para introducir código VEX. Elaboración propia a partir de (Arribas, 2019).

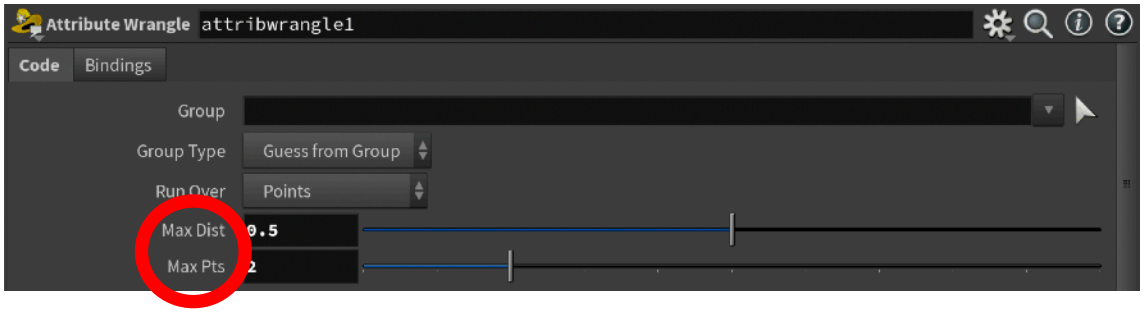


Figura 40. Canales de referencia añadidos a la parte de la interfaz que contiene los parámetros de operación del *wrangle* para facilitar la configuración de la distancia máxima y el máximo número de puntos. Elaboración propia a partir de (Arribas, 2019).

A continuación se hace uso de una de las funciones propias de Houdini, la función **`pointclouds`** que permite buscar puntos en función de unos criterios establecidos. Con esta se pretende encontrar puntos (un máximo de **`max_pts`**) dentro de un radio determinado (**`max_dist`**). Así, cada uno de los ladrillos que entra al **wrangle** calculará su distancia a la **grid** para determinar la intensidad de fragmentación que recibe.

VEXpression

```
1  //distancia máxima a la que buscar puntos
2  float max_dist = chf("max_dist");
3  //máximo número de puntos a buscar
4  int max_pts = chi("max_pts");
5  float dist = -1;
6  //pointcloud (geometría, posición, radio, número de puntos)
7  int pc = pcopen (1, "P", @P, max_dist, max_pts);
8  //itero por los puntos de la pointcloud
9  while (pciterate (pc)){
10     //indico pointcloud, distancia al punto, variable
11     pcimport(pc, "point.distance", dist);
12     //remapeo la distancia para ajustarla a una rampa
13     float remap_dist = fit(dist, 0, max_dist, 0, 1);
14     float dist_ramp = chramp("dist_ramp", remap_dist);
15     //creo un atributo
16     i@fracture_pieces = int(fit01(dist_ramp, chi("max_fractures"), chi("min_fractures")));
17 }
```

Figura 41. Código VEX que ejecutará el nodo *wrangle* por cada iteración. Elaboración propia a partir de (Arribas, 2019).

Con la expresión **pcpen** (ver figura 41, línea 7) se accede a la nube de puntos proporcionando el número de la entrada por la que se recibe (1), el nombre del parámetro que se valora (PChannel), el vector de la posición desde la que se buscarán los puntos (valor del atributo de posición @P, el cual representará la posición de cada uno de los puntos de los ladrillos recorridos en cada iteración), el radio de búsqueda y el número máximo de puntos a almacenar.

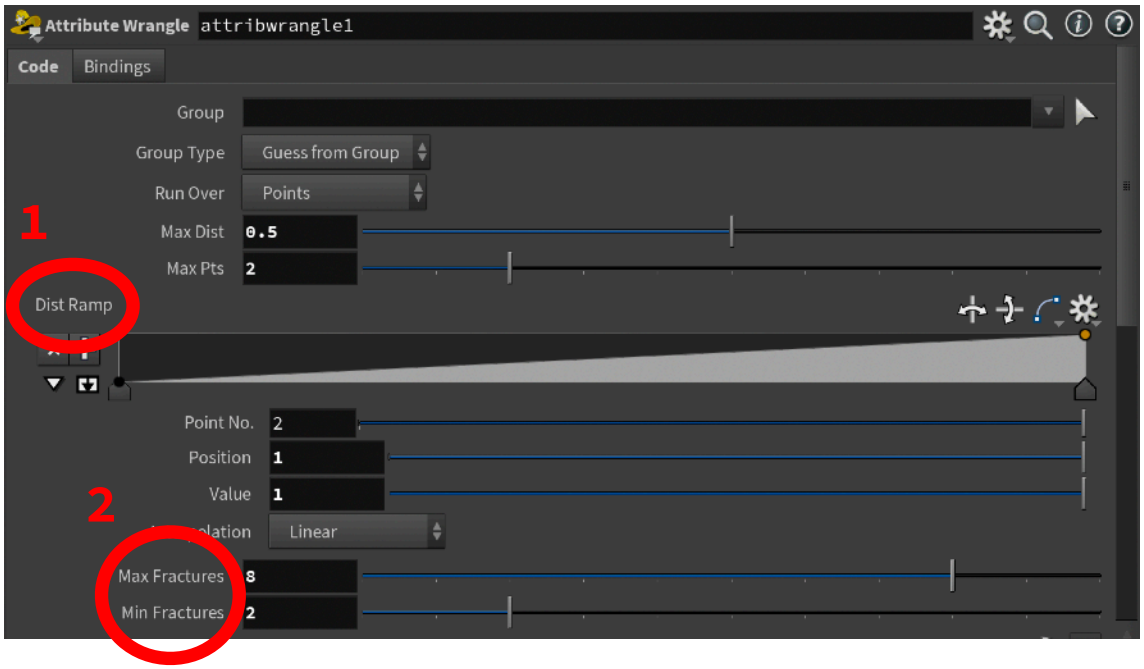


Figura 42. Canales y rampa de referencia añadidos a la parte de la interfaz que contiene los parámetros de operación del *wrangle* para facilitar la configuración de la frecuencia de fracturas. Elaboración propia a partir de (Arribas, 2019).

Hecho esto, se recorre la nube de puntos mediante un bucle *while* y la función **`pciterate`**, que devolverá un *booleano* con valor TRUE siempre que existan puntos nuevos en la nube y parará (FALSE) cuando ya no queden y, por lo tanto, no se cumpla la condición. Por otro lado, **`pcimport`** extrae la información del punto localizado y **`point.distance`** calcula la distancia hasta este desde la geometría (ladrillo).

Llegado a este punto se crea una "rampa" (véase figura 42, número 1) para que el usuario pueda configurar cuantas fracturas crear según el radio de incidencia, para lo que es necesario ajustar los valores a un rango de (0,1) con la función **`fit01`** (véase figura 41, líneas 13 y 16).

Finalmente se crea un atributo para poder diferenciar los ladrillos en función de su lejanía o cercanía a la **grid** (fractura). Se asigna el valor mínimo y máximo de piezas a generar según la distancia a la **grid**: cuanto menor sea (valor mínimo de distancia) se asigna el mayor número de fracturas, mientras que cuanto mayor sea (valor máximo de distancia) menor número de roturas (ver figura 41, línea 16, y figura 42, número 2).

Así pues se empaqueta la geometría con un nodo **assemble** y se añade otro **scatter** para controlar el número de fracturas a generar. Este último permite forzar el número de puntos mediante HScript (ver figura 43), indicando en qué nodo se encuentra el atributo (**`fracture_pieces`**) a considerar para cada elemento.

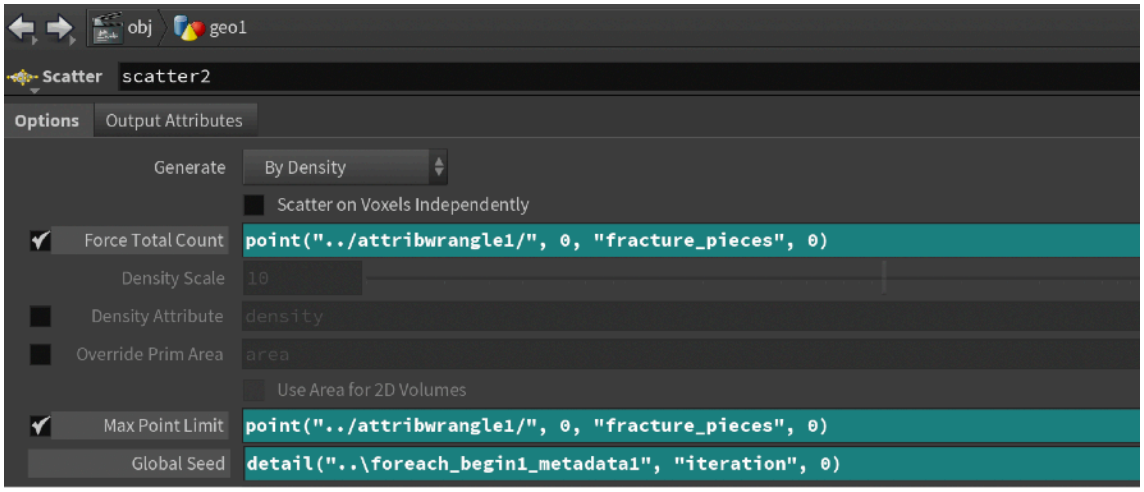


Figura 43. Configuración de los parámetros del nodo *scatter* empleando HScript. Elaboración propia a partir de (Arribas, 2019).

A esto se le aplica un nodo **voronoifracture** el cual fractura la geometría de entrada (en este caso, por el *input* 0) realizando una descomposición de Voronoi alrededor de los puntos de entrada (*input* 1).

El diagrama de Voronoi se basa en la proximidad. Dado un conjunto de puntos en el plano (**grid**):

$$P = \{p_1,...,p_n\} \text{ donde } n \geq 2$$

Se asigna a cada p_i un punto del plano en función de su cercanía, formando conjuntos que recubren a éste. Cada punto abarca una región que engloba el espacio más cercano a ese punto que a otro cualquiera. De esta forma el plano queda dividido en tantas áreas como puntos tiene el conjunto (ver figura 44). Esto supone que aquellas zonas que solo reciban un punto constituirán un único área, por lo que ese ladrillo no se fragmentará.

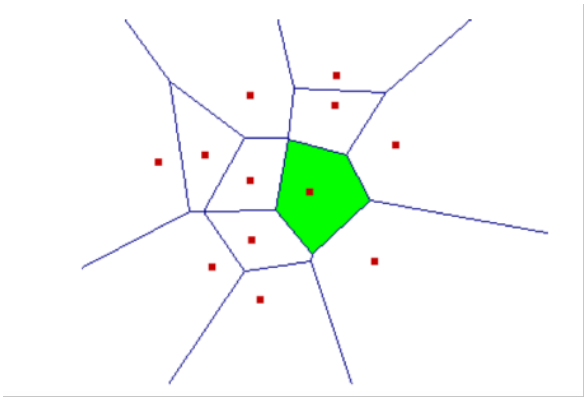


Figura 44. Ilustración explicativa del resultado de una fractura basada en el diagrama de Voronoi. (US, 2021).

Para previsualizar cómo se generarán estas fracturas en cada ladrillo se añade un nodo **explodedview**, que descompone pieza por pieza la geometría en cuestión (ver figura 45).

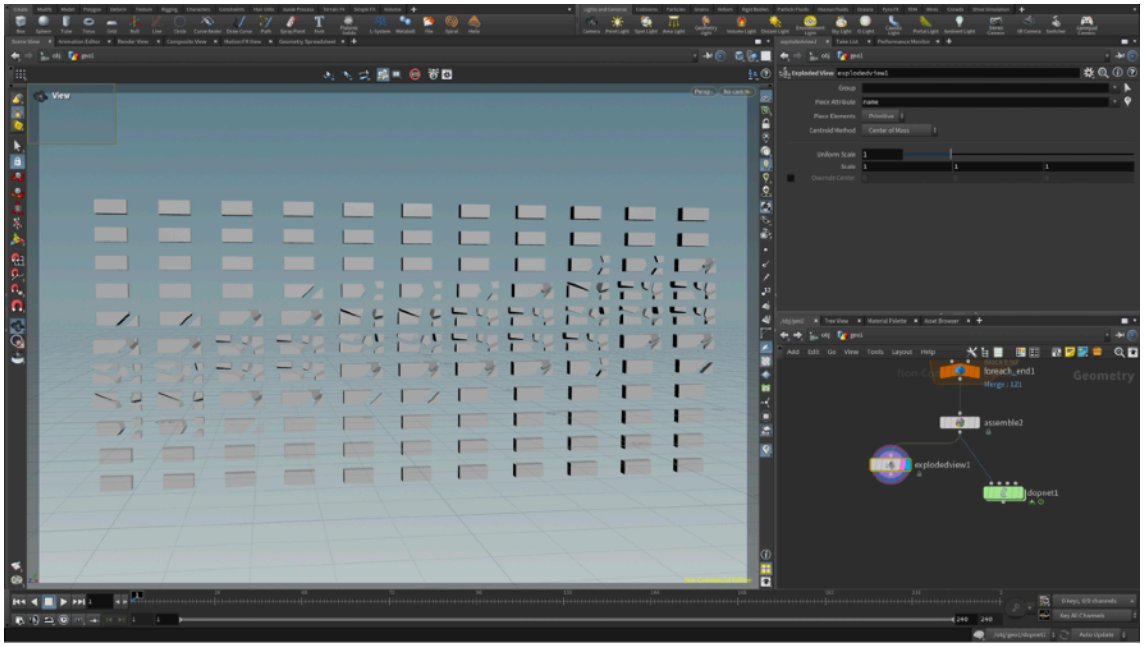


Figura 45. Efecto de la descomposición consecuencia del nodo *explodedview* aplicado en los ladrillos. Elaboración propia a partir de (Arribas, 2019).

Para terminar, se generan las dinámicas, es decir, el sistema de fuerzas que empujará a la geometría a su deterioro. Para ello, se crea un nodo **dopnetwork** que contendrá los nodos responsables de realizar la simulación. Dentro de este se añade, primero, un **rbdpackedobject** para crear un único DOP a partir de la geometría SOP que recibe.

De este modo, emplea cada primitiva para representar las dinámicas provocadas al actuar sobre este modelo rígido, un objeto RBD. Se toma un único punto por primitiva y sus atributos para almacenar información como la orientación, la masa y la velocidad del movimiento. Hecha esta conversión es posible aplicar un **rigidbodysolver** como *solver* para la configuración de RBD, que une dos motores diferentes: el motor RBD y el motor Bullet. El primero es útil para geometría más compleja y deformable, mientras que el segundo es adecuado para simulaciones rápidas y a gran escala.

Por último, se añade un nodo **gravity** para aplicar una fuerza similar a la de un campo gravitacional sobre la geometría. La magnitud de la fuerza que experimentará cada objeto será, por tanto, proporcional a su masa. Aunque mediante el parámetro "Gravity Scale" es posible ajustar la intensidad de esta fuerza, se configura por defecto para imitar la de Tierra, a unos 9.8 m/s². Lo mismo ocurre con la dirección del vector, predefinido en el eje vertical Y con valor negativo. Esto se lleva a cabo sobre un **groundplane** que actúa como suelo infinito sobre el que que las fracturas puedan precipitarse sin importar la distancia que se desplacen.

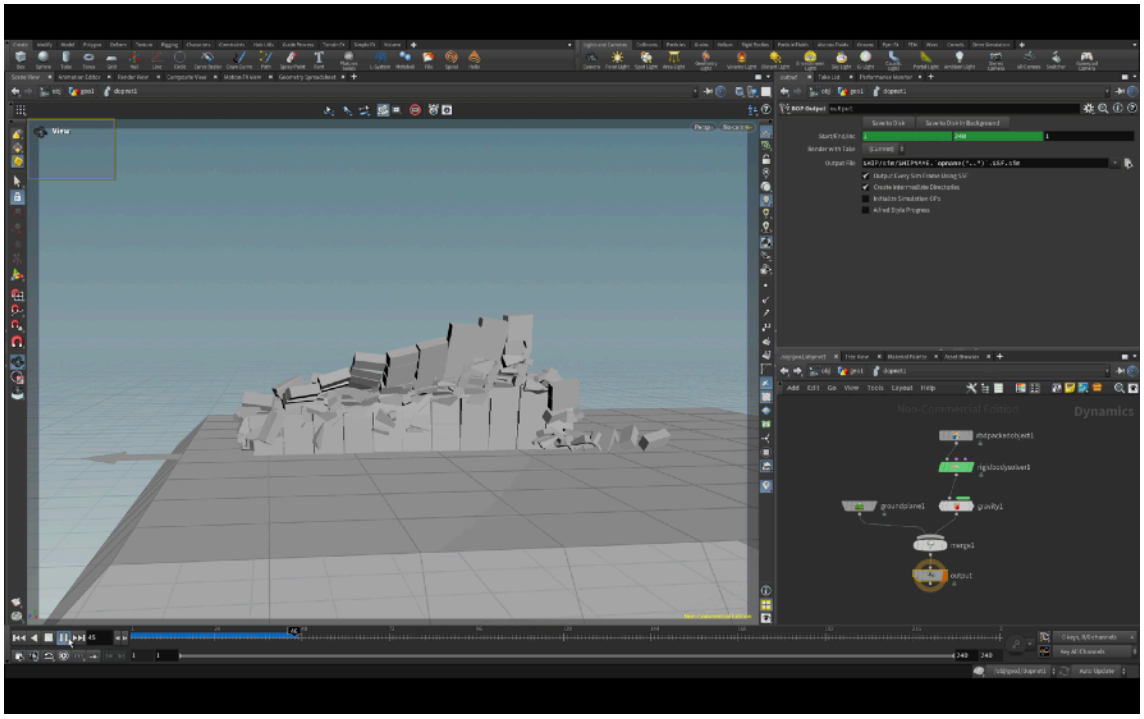


Figura 46. Frame 45 que muestra el resultado de la simulación. Elaboración propia a partir de (Arribas, 2019).

Finalmente se comprueba el satisfactorio funcionamiento de la simulación (ver figura 46), que se comporta de la forma propuesta como objetivo en el inicio del proyecto. Este resultado puede apreciarse en el anexo E.

c. HERRAMIENTA O TOOL CON PYTHON

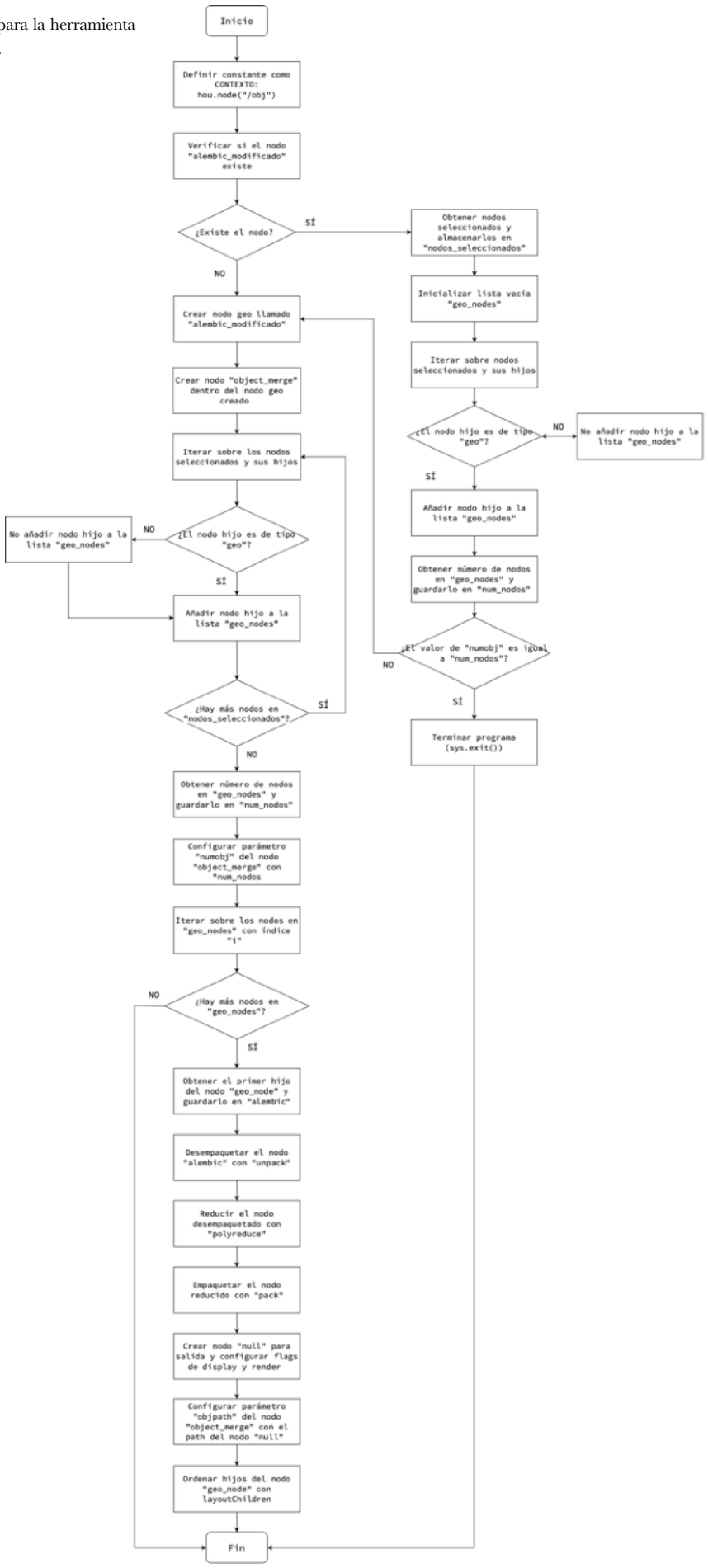
Los *alembic* (.abc) son un tipo de formato que se emplea para el intercambio de datos información de geometrías entre diferentes softwares 3D. Permite importar y exportar archivos, lo que es especialmente útil en flujos de trabajo colaborativos entre diferentes artistas. En este tipo de archivos es frecuente encontrar grandes cantidades de nodos que forman el modelo principal. En ocasiones es necesario automatizar o tratar de forma individual cada una de estas partes, lo que puede resultar un proceso tedioso.

La siguiente herramienta trata de realizar este proceso manual de una forma más eficiente con ayuda de Python, recorriendo cada uno de los nodos para reducir su contenido de forma que sea posible acceder a una versión más simplificada del *alembic* en cuestión.

Dado que la versión gratuita de Houdini, Apprentice, no soporta la exportación de este tipo de archivos, se recurrió a la artista de efectos visuales Tania Navarro Villena, que nos proporcionó un modelo *alembic* sobre el que aplicar esta herramienta.

Los pasos a seguir en el procedimiento quedan reflejados en el diagrama de flujo mostrado en la figura 47. El código completo para esta herramienta se encuentra recogido en el anexo B.

Figura 47. Diagrama de flujo para la herramienta con Python. Elaboración propia.



Para empezar, el modo de escritorio "Technical" de Houdini permite crear herramientas con Python desde la ventana "TD Tools". Esta vista incorpora la "Python Shell" (ver figura 48), una consola interactiva para la ejecución de comandos en tiempo real.

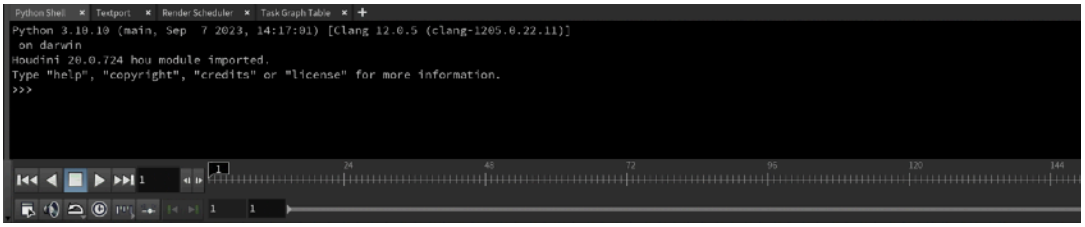


Figura 48. Parte de la interfaz que muestra la consola Python Shell para la introducción de comandos en el modo Technical de Houdini. Elaboración propia a partir de (Arribas, 2019).

Una vez analizado el itinerario a seguir, lo primero a realizar en el *script* de la nueva herramienta es declarar un contexto o raíz que sirva dirección para el restos de elementos relevantes en la elaboración de esta *tool*.

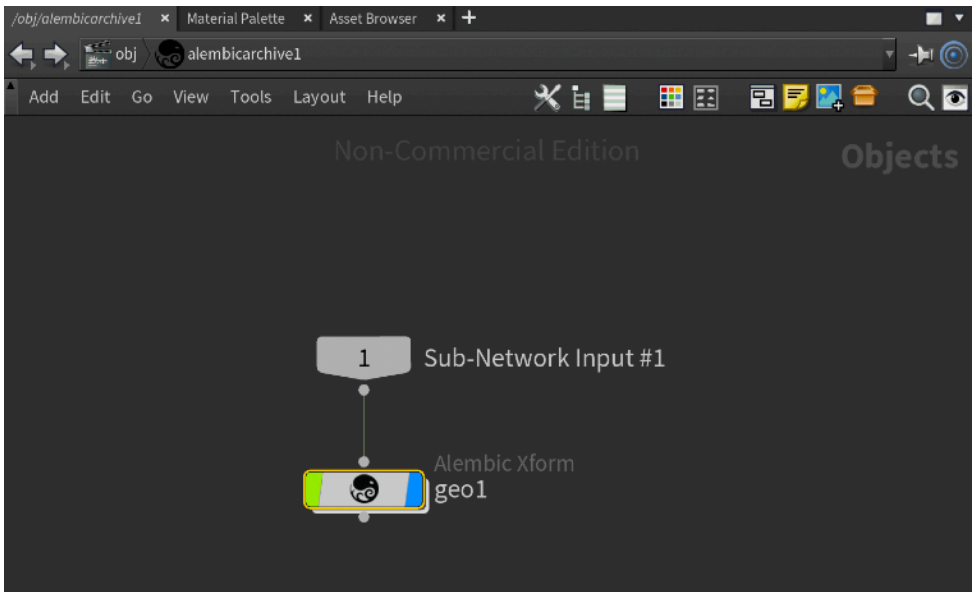


Figura 49. Contenido del archivo *alembic* tomado con ejemplo en la elaboración de esta herramienta. Elaboración propia a partir de (Arribas, 2019).

Dentro de un archivo *alembic* se halla una *subnet* (véase figura 49) conectada a otro nodo tipo **geo** que contiene todas sus partes, por lo que para acceder a estas es necesario navegar por la jerarquía de nodos hasta alcanzar el elemento final sobre el que aplicar las modificaciones. Para ello, como en el modelado procedural, es conveniente realizar primero estas operaciones de forma manual para contemplar la metodología a seguir de la forma más óptima (véase figura 50).

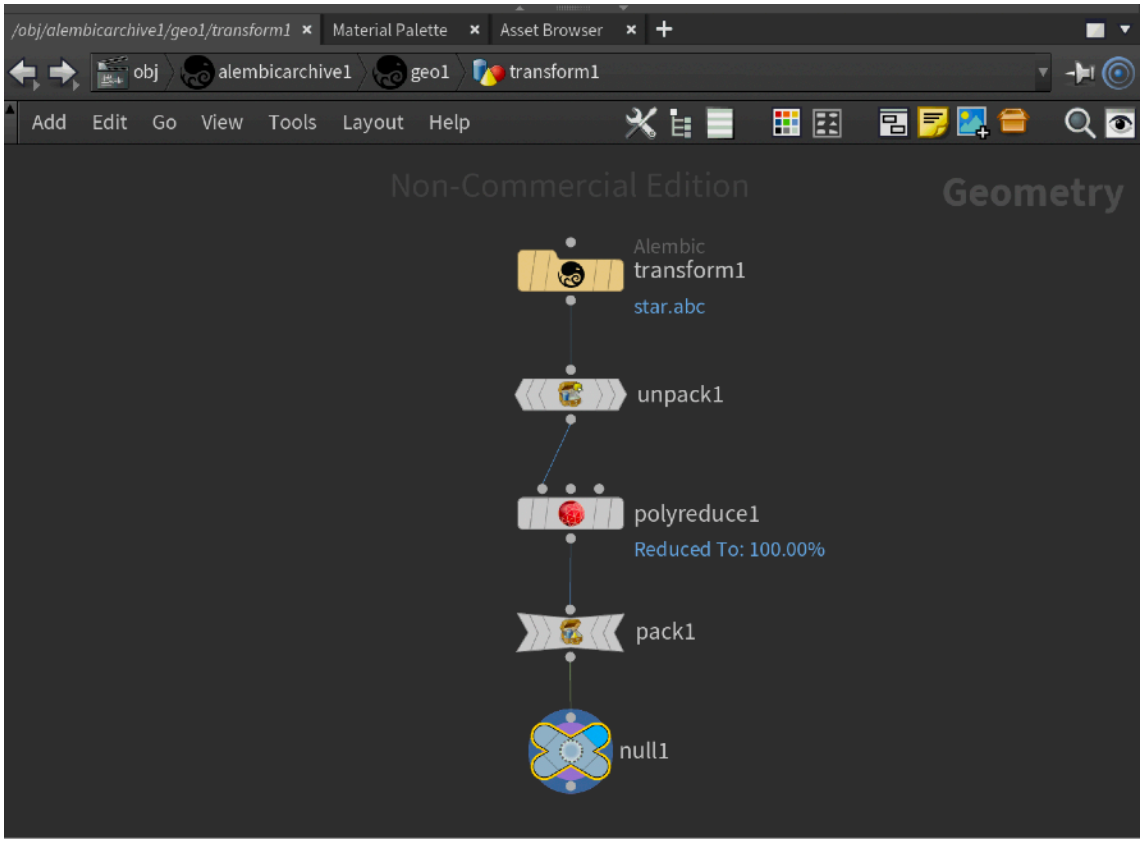


Figura 50. Primero se realizan las operaciones de forma manual, en este caso, se desempaqueta, reduce y vuelve a empaquetar el nodo original para finalmente volcar el contenido en un *null*. Elaboración propia a partir de (Arribas, 2019).

Para combinar todas estas partes del *alembic* en un nodo que permita gestionar y visualizar su ubicación de una forma más cómoda se utiliza un **merge**. Previamente se crea un nodo **geo** que actúe como contenedor para este.

El siguiente paso es el de indicar al sistema los nodos *alembic* sobre los que se aplicará la herramienta, para lo que se hace uso de la función `selectedNodes()`, incluida en el módulo `hou` que Houdini incorpora de serie, por lo que no es necesario importarlo. Esta función devuelve una tupla de objetos que representan los nodos actualmente seleccionados.

Como se ha explicado, para acceder a la información que se desea "comprimir", hay que profundizar en la red de nodos. Con este propósito, dado que el último elemento que contiene el *alembic* es un **transform** (nodo de tipo **geo**, ver figura 51), se emplea un bucle `for` y la función `allSubChildren()` para recorrer todos los nodos "hijo" de este clase y almacenarlos en una nueva lista (ver figura 52, línea 42).

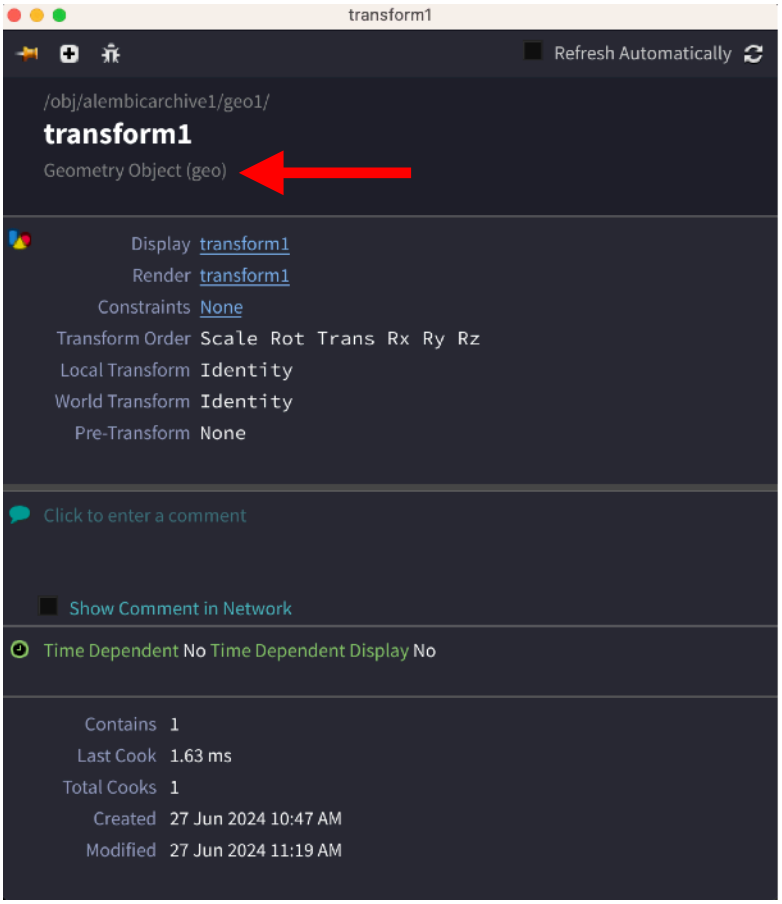


Figura 51. Ventana con la información del nodo *transform* de tipo *geo*. Elaboración propia a partir de (Arribas, 2019).

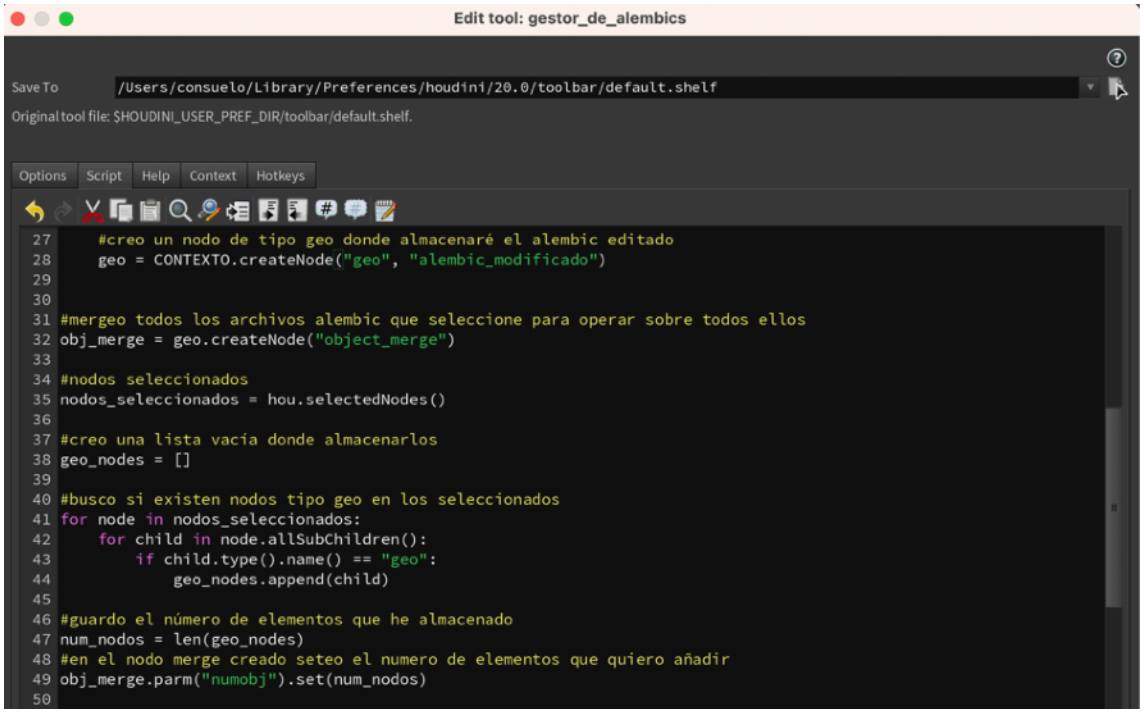


Figura 52. Parte del código Python para la herramienta de gestión de *alembics*. Elaboración propia a partir de (Arribas, 2019).

A continuación, se configura el número de partes encontradas en el **merge**, que coincidirá con el del tamaño de la lista implementada en el **`for`**. Para llevar a cabo esta modificación se debe comprobar el nombre de la variable o parámetro respectiva a este valor, que se conoce como **`numobj`** (ver figuras 52, línea 49, y figura 53).

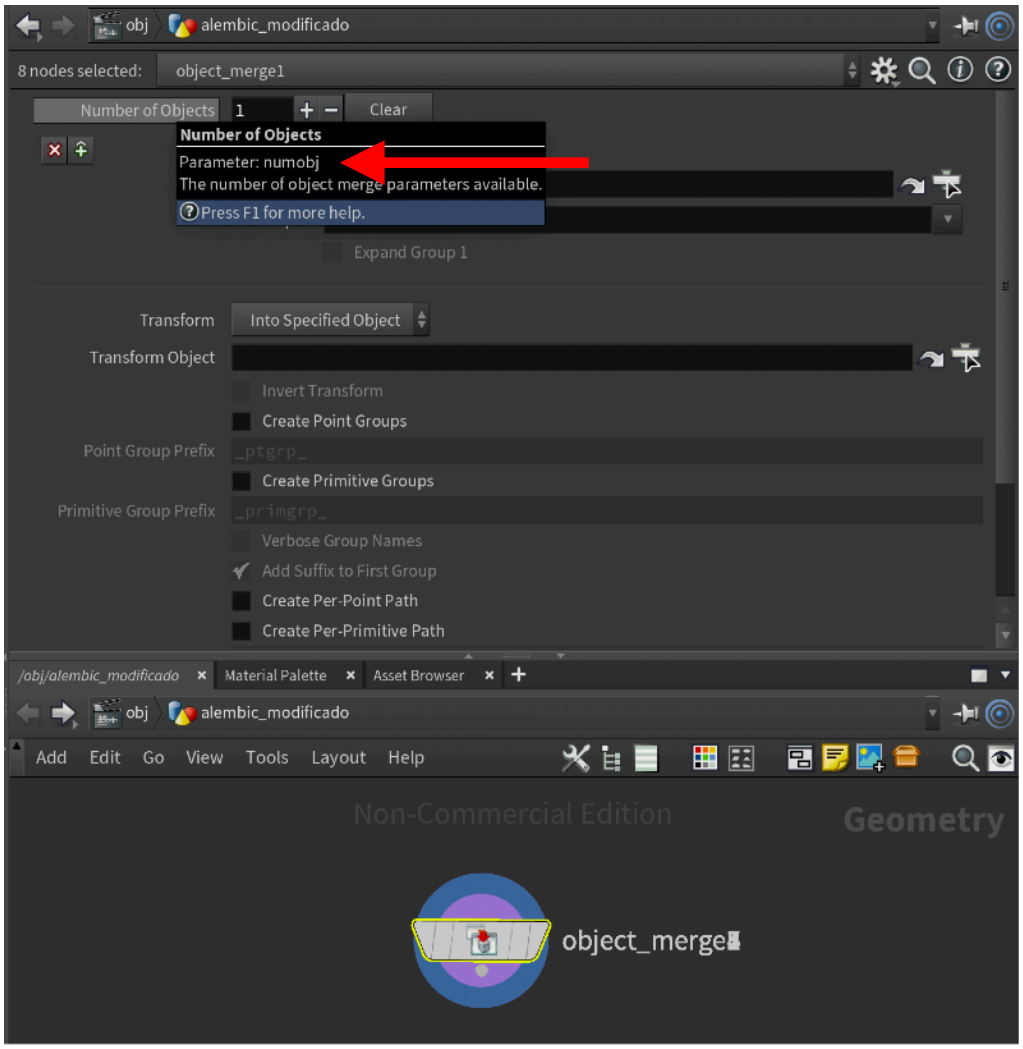


Figura 53. Para comprobar el parámetro que contiene el número de objetos del nodo *merge* basta con sostener el ratón encima del nombre. Elaboración propia a partir de (Arribas, 2019).

Para actuar sobre cada uno de los elementos de la lista se crea un índice o *index* mediante la función **`enumerate`** con el fin de identificar y gestionar cada uno de ellos en el **merge** (véase figura 55, línea 52). Una vez hecho esto, como se muestra en la figura 50, se aplican los siguientes operadores: primero se desempaqueta la geometría con un nodo **unpack** y después se reduce con un **polyreduce** para finalmente volcar el resultado nuevamente empaquetado en un **null**. De esta forma el resultado se habrá reducido un 90% respecto a la geometría original (véase figura 54).

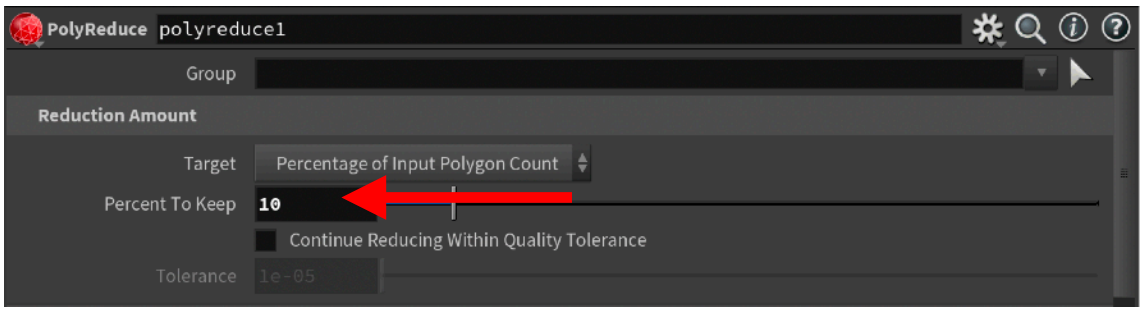


Figura 54. Verificación de cómo el código configura automáticamente el porcentaje de la geometría que se mantiene sin reducir. Elaboración propia a partir de (Arribas, 2019).

Finalmente se configura la dirección o *path* de cada objeto reducido en el parámetro **`objpath`** del **merge** (ver figura 55, línea 69). Como el nombre del parámetro variará con cada nodo que recorra el bucle, se aprovecha el *index* para diferenciarlos. Para ello, como en Python no es posible concatenar una cadena de caracteres con valores numéricos, se realiza una conversión con la función **`str`**.

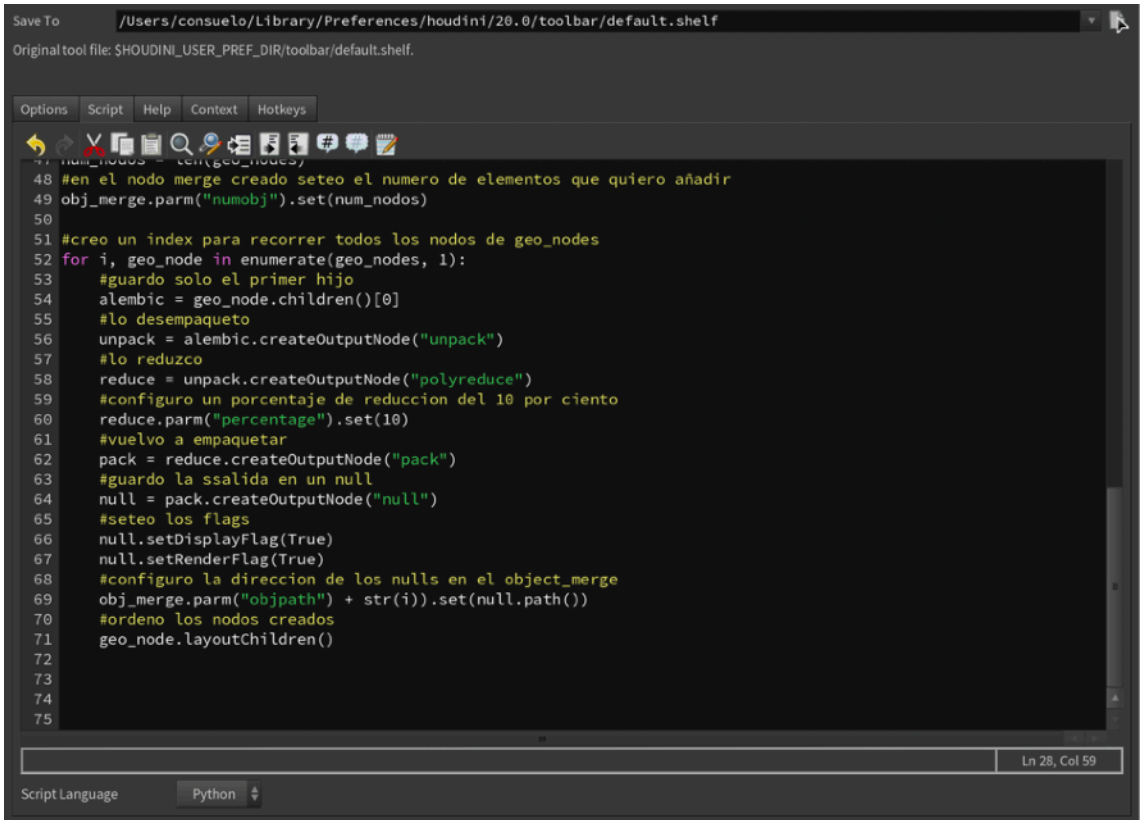


Figura 55. Bucle *for* con función *enumerate*. Elaboración propia a partir de (Arribas, 2019).

De este modo se ha logrado obtener una versión "low poly" o simplificada de la geometría y clasificar cada una de sus piezas en un nodo **merge** que permite tener acceso a su nombre y ubicación para una gestión más eficiente de los recursos. Para probar la

herramienta sobre un mayor número de partes, dado que el archivo *alembic* empleado solo contiene un único elemento, en el ejemplo de las figuras 56 y 57 se ha triplicado el mismo nodo con el archivo *alembic*.

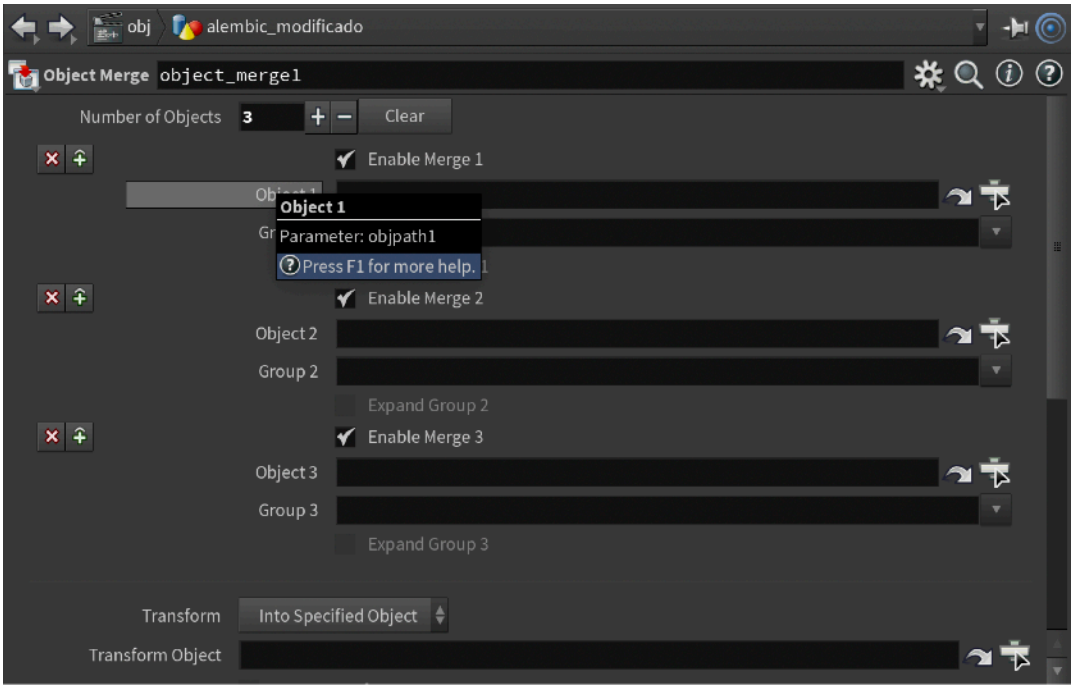


Figura 56. Información del nodo *merge* antes de configurar las direcciones de los nodos que contienen las partes del *alembic*. Elaboración propia a partir de (Arribas, 2019).

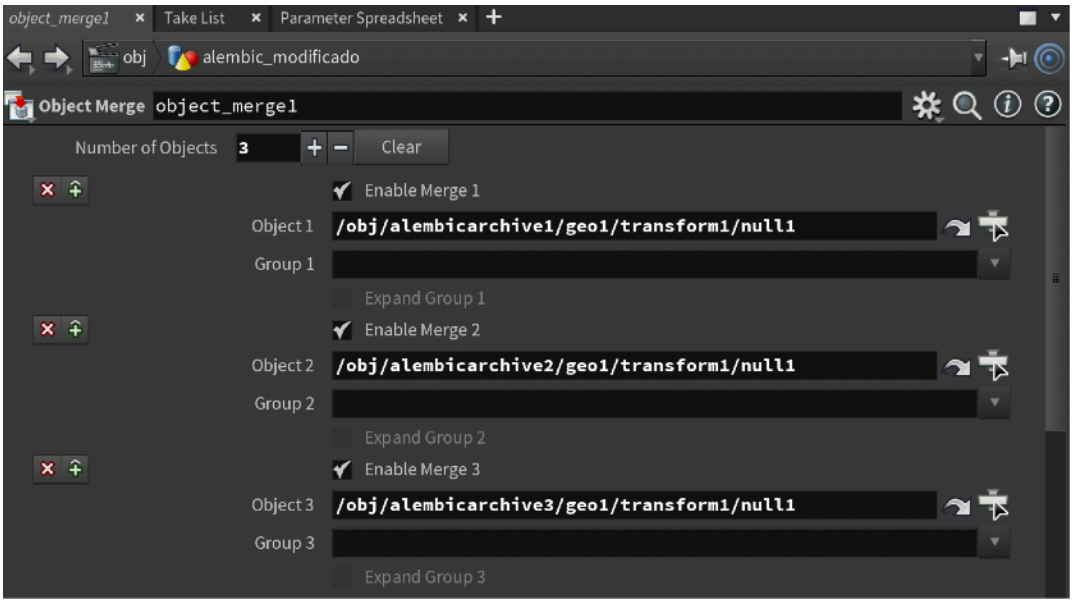


Figura 57. Ejemplo de la información del nodo *merge* después de configurar las direcciones de los nodos que contienen las partes del *alembic*. Elaboración propia a partir de (Arribas, 2019).

8. CONCLUSIONES

En la realización de esta memoria se ha analizado el desarrollo de herramientas en Houdini, con un enfoque particular en la programación en Python y VEX. Para ello, ha sido necesario profundizar en el extenso mundo de los efectos visuales, su historia y cada uno de los departamentos que lo componen. De esta forma ha sido posible comprender el papel que los softwares de 3D y técnicas como el modelado procedural juegan en la industria audiovisual, además de sus múltiples aplicaciones. Asimismo, conocer la labor de un director técnico y las funciones que le atañen, ha facilitado la adquisición de habilidades técnicas avanzadas, así como una mejor comprensión de las herramientas necesarias para manejar un software tan especializado.

Sin embargo, cabe destacar la aparición de problemas y obstáculos que surgieron en la elaboración de este trabajo. Por un lado, la pronunciada curva de aprendizaje que presenta Houdini ha supuesto un gran reto a la hora de entender el funcionamiento del mismo. El adaptarse a su interfaz y metodología representaron un desafío que costó meses de estudio y preparación previa a la redacción de este proyecto.

Hay que mencionar que este estudio se llevó a cabo de forma autodidacta. No obstante, este esfuerzo lo ha convertido en una experiencia más enriquecedora que forzó a la búsqueda de expertos en el sector, quienes prestaron asesoramiento a lo largo de la investigación del trabajo. Su amabilidad, apoyo y profesionalidad han supuesto un pilar imprescindible en el desarrollo del trabajo, el cual no habría sido posible realizar sin su ayuda y colaboración. Este enfoque ha potenciado la autonomía e iniciativa personal, además de la obtención de valiosos contactos para el futuro laboral.

Por otro lado, el aprendizaje constante que implica la programación ha sido un reto permanente que se tuvo que aprender a incorporar al flujo de trabajo. En este contexto, la continua visualización y consulta de numerosos tutoriales en plataformas como YouTube y Vimeo, así como de cursos específicos de Houdini, han sido esenciales para superar obstáculos técnicos y conceptuales encontrados en el camino.

También hay que mencionar que el *render* del sistema de fracturas procedurales no se llevó a cabo dado que su realización implicaba un estudio adicional de este área que no se encontraba entre los objetivos de estudio.

A pesar de esta complejidades se ha podido avanzar en el manejo del software, pudiendo cumplir los objetivos planteados con éxito. Pues profundizar en la personalización y automatización de procesos mediante *scripting* en Python y VEX ha permitido comprender

cómo pueden integrarse estas técnicas en un flujo de trabajo profesional para optimizar su uso en contextos específicos de producción. Además, los conocimientos y habilidades adquiridos constituyen una base sólida tanto para mi desarrollo académico, como para mi carrera profesional y futuros estudios.

En conclusión, el desarrollo de este proyecto y el compromiso para reflejar en la memoria el carácter artístico y técnico de ambos grados han supuesto una experiencia que, además de mejorar mis competencias técnicas, me ha permitido conocer más a fondo la relevancia de los efectos visuales. Enfrentando la complejidad de Houdini me ha sido posible comprender la importancia de la perseverancia en este área. Asimismo ocurre con la programación, en la que "lo importante no es tanto cuánto sepas sino la habilidad que tengas para buscar las respuestas" (Martínez, 2022). Aprecio ahora tantas semejanzas entre estos campos, y cómo existe la posibilidad de que vayan de la mano.

Los efectos visuales, como la tecnología, se encuentran en constante evolución. Una evolución que avanza con cada nueva historia que se proyecta en las salas de cine. Que no sería posible sin la combinación de ambas materias, sin la intención de apostar por un arte tan vivo como digital. Porque los mundos a los que las pantallas nos transportan y las realidades que nos permiten imaginar, no serían posibles sin la existencia de estas dos disciplinas. El arte necesita la ciencia para impedir que vuele demasiado cerca del sol, y la ciencia… pues sin el arte podría no despegar jamás.

BIBLIOGRAFÍA

Aguilar, I. (2015, 8 de abril). Formatos cinematográficos (III) [Entrada de blog]. HarmonicaCinema. http://www.harmonicacinema.com/formatos-cinematograficos-iii/

Arribas, M. (2019,1 de julio). Curso Online: VEX en Houdini. 1 - Introducción al Curso [Vídeo]. Youtube. https://www.youtube.com/watch?v=pQpzJQZgKUI

Arribas, M. (2024). *HoudiniTD. Python + VEX* [Video]. Gumroad. https://stupidbrain.gumroad.com/l/houdinitd

Bragagna, J. (2023). What Can You Do With Houdini? [Entrada de blog]. Artstation. https://www.artstation.com/blogs/julianbragagna/Know/what-can-you-do-with-houdini

Calvo, R. (2023). Animación 3D, VFX y diseño gráfico: La revolución visual demanda más profesionales. *Renderout*, 39, 28-32. https://renderout.es/revista/

Cazallas, J. (2019). *Qué es Stagecraft, la tecnología detrás de The Mandalorian*. Hobbyconsolas. https://www.hobbyconsolas.com/reportajes/stagecraft-tecnologia-detras-mandalorian-543837

Comet, M. (2009). *About Me: (the quick version)*. Comet-cartoons. http://www.comet-cartoons.com/aboutme.html

DIBOOS. (2018). *Libro Blanco. La industria española de la animación y de los efectos visuales*. SpainAudiovisualHub. https://spainaudiovisualhub.mineco.gob.es/content/dam/seteleco-hub-audiovisual/resources/pdf/DIBOOS_LIBRO BLANCO_Sep2018.pdf

Domínguez, M. (2022, 13 de septiembre). *A Pipeline Story | ¿Cómo funciona el PIPELINE de los VFX?* [Vídeo]. Youtube. https://www.youtube.com/watch?v=jI8RZPblKv0&t=103s

Dykstra, J. (2023). *Star Wars: Miniature and Mechanical Special Effects*. TheAsc. https://theasc.com/articles/star-wars-miniature-effects-dykstra

Ecured. (s.d.). *NURBS*. Ecured. https://www.ecured.cu/NURBS

Espín, J. (2022). *Las innovaciones de ILM en la industria cinematográfica durante el siglo XX. La digitalización de los efectos visuales.* (Trabajo de Fin de Grado, Universidad Politécnica de Valencia). https://riunet.upv.es/handle/10251/189162?show=full

Fernández, E. (2014). El cine y la tecnología: El proceso Schüfftan [Entrada de blog]. NEOTEO. https://www.neoteo.com/el-cine-y-la-tecnologia-el-proceso-schufftan/

Gallego, H. (2020, 15 de febrero). *LA MAGIA de HOUDINI en el CINE | EFECTOS Y CINEMATOGRAFIA* [Vídeo]. Youtube. https://youtu.be/PDjV16vpRuQ?si=G2Fyt_KTgPATQlan

Gallego, H. (2023, 29 de junio). *¿Por qué DEBERÍAS APRENDER NUKE? @FoundryTeam* [Vídeo]. Youtube. https://www.youtube.com/watch?v=lAL5m78U-JQ

García, O. (2013). Composición Digital: Perspectiva Histórica de una Evolución Tecnológica. *Orbis: Revista Científica Electrónica de Ciencias Humanas, 26, 9-27*. ResvistaOrbis. https://dialnet.unirioja.es/servlet/articulo?codigo=7208977

Jove, D. (2022). *Generación procedural de edificios 3D utilizando los lenguajes VEX/Python (Houdini) a partir de datos georreferenciados por Sistemas de Información Geográfica (SIG).* (Trabajo de Fin de Grado, Universidad de Coruña). https://ruc.udc.es/dspace/handle/2183/31214

La pausa del render. (2017, 27 de noviembre). Diferencias entre efectos especiales y visuales [Entrada de blog]. Lapausadelrender. https://www.lapausadelrender.com/educacion-audiovisual/efectos-especiales-vs-efectos-visuales/

López, P. (2017). *Historia de los efectos especiales en el siglo XX* (Trabajo de Fin de Grado). Universidad Politécnica de Madrid. https://oa.upm.es/48961/

Lucasfilm. (Prod.), y Kasdan, L. (Dir.). (2022). *Light & Magic: Un sueño no tan lejano* [Documental]. Lucasfilm, Disney. https://www.disneyplus.com/

Lizcano, D. (2020, 11 de noviembre). Historia del "matte painting" en España. [Entrada de blog]. Efectos Especiales en el Cine Español. http://efectosespecialescinespaniol.blogspot.com/2020/11/historia-del-matte-painting-en-espana.html

Magee, R. (2007, 6 de julio). Intro to Houdini's Node-based Workflow [Entrada de blog]. SideFX. https://www.sidefx.com/tutorials/intro-to-houdinis-node-based-workflow/

Martínez, J. (2022, 26 de julio). *MASTERCLASS - Introducción a Python para Houdini | Tus primeros pasos* [Vídeo]. Youtube. https://www.youtube.com/live/rrosrkdCP0Y?si=5hGdFZxkWkXVOk68

McKay, A. (2021, 29 de mayo). *Technical Artist + Technical Director Job for VFX Explained (Allan McKay)* [Vídeo]. Youtube. https://youtu.be/vsEtkMZ4emY?si=SnuPNbT9bvuH10lf

Méndiz, A. (2012). Metrópolis (1927), de Fritz Lang (parte 2). *FilaSiete: críticas de cine y series. FilaSiete*. https://filasiete.com/articulos/making-of/metropolis-1927-fritz-lang-2/

Navarro, T. (2020). *Creación de efectos visuales mediante la generación de gráficos procedurales, partículas y fluidos con Houdini de SideFx*. (Trabajo de Fin de Grado, Universidad Politécnica de Valencia). https://riunet.upv.es/handle/10251/150117

Norén, A. (2018, 14 de octubre). Node Based Procedural Workflow [Entrada de blog]. Flyro. https://med3017m-1819-14565893level3digital.coursework.lincoln.ac.uk/2018/10/14/node-based-procedural-workflow/

Ollé, A. (2020, 13 de mayo). *La industria de los efectos especiales (para dummies)* | *AriO* [Vídeo]. Youtube. https://www.youtube.com/watch?v=ogIA33iOFbs

Ozturk, S. (2023, 15 de marzo). Vicious Circle: John Whitney and the Military Origins of Early CGI [Entrada de blog]. Brightlightsfilm. https://brightlightsfilm.com/vicious-circle-john-whitney-and-the-military-origins-of-early-cgi/

Petty, J. (s.d.). *What is Houdini & What Does It Do?*. Conceptartempire. https://conceptartempire.com/what-is-houdini-software/

Python. (2024). *El tutorial de Python*. Python.org. https://docs.python.org/es/3/tutorial/

Rebelway (2024). *The Evolution of VFX: From "Metropolis" to 2024*. Rebelway. https://www.rebelway.net/the-evolution-of-vfx/

Rickitt, R. (2000). *Special effects the history and techniques.* Hollywood, USA: Billboard Books.

Salom, L. (2019). *UPValenciaX: Python: aprender a programar* [Video]. Universidad Politécnica de Valencia. edX. https://www.edx.org/es/learn/python/universitat-politecnica-de-valencia-python-aprender-a-programar

Serebrennikova, K. (s.d.). *What is VEX in Houdini?*. Rebelway. https://www.rebelway.net/what-is-houdini-vex/

SideFx. Manual de houdini. (s. d.). Recuperado de https://www.sidefx.com/docs/

SideFX. (s.d.). *VEX*. SideFX. https://www.sidefx.com/docs/houdini/vex/index.html

Soler, C. (2021). *Desarrollo de animales antropomórficos como personajes de animación 3D* (Trabajo de Fin de Grado). Universidad de Alicante. https://rua.ua.es/dspace/handle/10045/116960

Smith, C. (2024, 23 de mayo). What is After Effects [Entrada de blog]. Agitraining. https://www.agitraining.com/adobe/after-effects/classes/what-is-after-effects

Tejedor, E. & Martínez, J. (2019). Modelado procedural. *Renderout*, 22, 107-113. Renderout. https://renderout.es/revista/

Universidad de Sevilla. (s.d.). *Geometría Computacional*. Universidad de Sevilla. https://asignatura.us.es/fgcitig/contenidos/gctem3ma.htm

UT-HUB. (2024, 7 de marzo). ¿Cuál es la diferencia entre CGI y VFX? [Entrada de blog]. Ut-hub. https://www.ut-hub.com/cual-es-la-diferencia-entre-cgi-y-vfx/

Vanas. (s.d.). What is Autodesk Maya [Entrada de blog]. Vanas. https://www.vanas.ca/en/blog/what-is-autodesk-maya

BIBLIOGRAFÍA ADICIONAL

Blaine, M. (2021, 11 de noviembre). *El proceso Schüfftan es una antigua técnica de efectos especiales.* [Vídeo]. Youtube. https://youtu.be/cEA_UhcTG54?si=j6yng1dVM8cImNHB

Brannon, A. (2024, 5 de abril). *Behind the scenes: the evolution of special effects in movies*. [Entrada de blog]. Moviemetropolis. https://moviemetropolis.net/2024/04/05/the-evolution-of-special-effects-in-movies/#The_age_of_digital_mastery_VFX_in_film

Chuidiang. (2015, 6 de junio). Tipado fuerte y débil. Tipado estático y dinámico. [Entrada de blog]. Diario de Programación. https://blog.chuidiang.org/2015/06/06/tipado-fuerte-y-debil-dinamico-y-estatico/

Cinema Katharsis. (2021, 10 de marzo). *Cinediccionario Episodio 1: FX, VFX, CGI y SFX*. [Vídeo]. Youtube. https://youtu.be/JI6Y-9eyj5A?si=9pi_HAQOneBNkper

Cine Retro. (2023, 6 de febrero). *La historia detrás de los primeros efectos especiales: George Méliès y su legado*. [Vídeo]. Youtube. https://youtu.be/x3ugaUjKHG0?si=HYZ232M-lVLiX4mC

ComputerHoy.com. (2022, 25 de marzo). ¿Sabes qué es PYTHON y por qué es un lenguaje de programación tan importante? [Vídeo]. Youtube. https://youtu.be/Rv910T1BJUw?si=t4ond86-8k-Dphie

Cortés, J. (2019, 16 de agosto). Los VFX de Houdini | Reel & Novedades [Entrada de blog]. Notodoanimacion. https://www.notodoanimacion.es/los-vfx-de-houdini/

Delgado, H. (2020). *Modelos de animación*. Researchgate. https://www.researchgate.net/profile/Heider-Delgado/publication/350447652_Modelos_de_animacion/links/60601a68a6fdccbfea0f6291/Modelos-de-animacion.pdf

Dweb3d. (2017, 23 de septiembre). ¿Qué es 3D? [Entrada de blog]. Dweb3d. https://www.dweb3d.com/blog/que-es-3d/

Ebac. (2023, 7 de agosto). ¿Qué es el modelado 3D y cómo funciona? [Entrada de blog]. Ebac. https://ebac.mx/blog/que-es-el-modelado-3d

EduCaixa. (2014). *De la sobreimpresión al croma*. EduCaixa. https://educaixa.org/documents/10180/0/de_la_sobreimpresion_al_croma.pdf/98d24da3-9066-3281-0459-480c3621e61f?t=1646819648328

El Libro de Python. (2024). *¿Qué es Python? Introducción*. Ellibrodepython. https://ellibrodepython.com/que-es-python

Failes, I. (2020). *A visual history of Nuke*. Beforesandafters. https://beforesandafters.com/2020/02/20/a-visual-history-of-nuke/

Fandom. (s.d.). *Tyrannosaurus rex animatronics (Jurassic Park)*. Fandom. https://jurassicpark.fandom.com/wiki/Tyrannosaurus_rex_animatronics_(Jurassic_Park)

Ftrack. (2021, 24 de septiembre). What is a Pipeline TD and how can I become one? [Entrada de blog]. ftrack. https://www.ftrack.com/en/2021/09/what-is-a-pipeline-td.html#what-does-a-pipeline-td-do

García, P. (2024). los hermanos Lumière y el nacimiento del cine. *National Geographic*. National Geographic. https://historia.nationalgeographic.com.es/a/hermanos-lumiere-y-nacimiento-cine_12264

Jiménez, C. (2017, 30 de enero). *¿Cómo funciona el Pipeline de VFX?* [Vídeo]. Youtube. https://youtu.be/_zn_1cf87vc?si=AgpeF54B0C8wXwdx

Kiryha. (2021, 25 de agosto). Python for artists [Entrada de blog]. Github. https://github.com/kiryha/Houdini/wiki/python-for-artists

La pausa del render. (2016, 7 de marzo). ¿Qué es un Croma Key? [Entrada de blog] Lapausadelrender. https://www.lapausadelrender.com/educacion-audiovisual/que-es-un-croma-key/

Lightbox. (s. d.). *Diferencia entre efectos visuales y efectos especiales*. Lightbox. https://lboxacademy.es/blog/diferencia-entre-efectos-visuales-y-efectos-especiales-vfx-sfx/

Meza, G. (2015, 6 de julio). *Historia de los efectos especiales en el Cine* [Vídeo]. Youtube. https://youtu.be/px_lL41pnNQ?si=RTVbNJ_0Tu4szXkj

Pérez, E. (2021). *PDP-11/45: la computadora de 16 bits que permitió crear el CGI de la Estrella de la Muerte en 1976*. Xataka. https://www.xataka.com/historia-tecnologica/pdp-11-45-computadora-16-bits-que-permitio-crear-cgi-estrella-muerte-1976

Prada, X. (2014, 27 de abril). Posiciones o roles en un estudio de efectos visuales [Entrada de blog]. Elephantvfx. http://www.elephantvfx.com/blog/2014/4/27/posiciones-o-roles-en-un-estudio-de-efectos-visuales

Ràfols, R. (2018). *Historia de los motion graphic: La impresora óptica*. Universidad Oberta de Catalunya. https://disseny.recursos.uoc.edu/materials/motion-graphics/es/3-la-impresora-optica/

Ricaerredois. (2022, 28 de junio). SOP, POP, DOP, for begginers [Entrada de blog]. Reddit. https://www.reddit.com/r/Houdini/comments/t4agvh/sop_pop_dop_for_begginers/

Ricaño, C et al. (s. d.). *TTL (Transistor- Transistor Logic) o "Lógica Transistor a Transistor"*. Tutorialcid. https://tutorialcid.es.tl/Familia-TTL.htm

SpanishRevolution. (2021, 8 de junio). *Alice Guy, la madre del cine.* [Video]. Youtube. https://youtu.be/fiCRlcjivSM?si=64pxtuLdSxy8WMU7

VFXAcademy. (2024, 27 de febrero). Difference Between CGI vs VFX [Entrada de blog]. Vfxacademy. https://vfxacademy.in/difference-between-cgi-vs-vfx/

Vulture. (2018, 3 de abril). *4 Ways 2001: A Space Odyssey Was a Visual-Effects Pioneer*. [Vídeo]. Youtube. https://youtu.be/5ch5WC54egU?si=lQpGmGgcntWhWW0g

Visconti, L. (2021, 10 de marzo). *Efectos Especiales o Efectos Visuales - FX - SFX - SPFX - VFX - CGI - IN CAMERA*. [Vídeo]. Youtube. https://youtu.be/7RluYs4p8Nk?si=uYQVZf2al15-oUOy

Wirtz, B. (2023, 28 de julio). *Visual Effects (VFX): Past, Present, Future (Transforming The Ordinary To Extraordinary)*. GameDesigning. https://www.gamedesigning.org/animation/vfx/

Yobo. (s.d.). Tipado débil vs tipado fuerte [Entrada de blog]. Yobo. https://www.estudioyobo.com/tipado-debil-vs-tipado-fuerte/

FILMOGRAFÍA

Bayona, A. (Dir.). (2021). *Lo imposible* [Película]. Telecinco Cinema. https://www.filmaffinity.com/es/film329915.html

Benioff, D. (Prod.), y Benioff, D. et al. (Dir.). (2011). *Game of Thrones* [Serie de televisión]. Estudio, nombre de la productora o distribuidora. https://www.filmaffinity.com/es/film874956.html

Del Vecho, P. (Prod.), y Buck, C. & Lee, J. (Dir.). (2013). *Frozen* [Película]. Walt Disney Pictures. https://www.filmaffinity.com/es/film926588.html

Anne, G. (Prod.), y Cameron, J. (Dir.). (1986). *Aliens* [Película]. Brandywine Productions. https://www.filmaffinity.com/es/film682537.html

Cameron, J. (Dir.). (1991). *Terminator 2: Judgment Day* [Película]. Carolco Pictures. https://www.filmaffinity.com/es/film576352.html

Cameron, J. (Dir.). (2009). *Avatar.* [Película]. 20th Century Fox. https://www.filmaffinity.com/es/film495280.html

Chomón, S (Dir.). (1908). *Hôtel électrique* [Película]. Pathé. https://www.filmaffinity.com/es/film125740.html

Arad, A. (Prod.), y Favreau, J. (Dir.). (2008). *Iron Man* [Película]. Marvel Studios. https://www.filmaffinity.com/es/film201496.html

Favreau, J. (Dir.). (2019). *The Lion King* [Película]. Walt Disney Pictures. https://www.filmaffinity.com/es/film316246.html

Bartnicki, J. (Prod.), y Favreau et al. (Dir.). (2019). *The Mandalorian.* [Serie de televisión]. Lucasfilm. https://www.filmaffinity.com/es/film544948.html

Guy, A. (Dir.). (1896). *La Fée aux Choux* [Película]. https://www.filmaffinity.com/es/film372015.html

Rattray, E. (Prod.), y Henson, J. (Dir.). (1986). *Labyrinth* [Película]. Henson Associates, Inc. https://www.filmaffinity.com/es/film937270.html

Hitchcock, A (Dir.). (1958). *Vertigo* [Película]. Alfred J. Hitchcock Productions. https://www.filmaffinity.com/es/film647094.html

Johnston, J. (Prod.), y Howard, R. (Dir.). (1988). *Willow* [Película]. Lucasfilm. https://www.filmaffinity.com/es/film350632.html

Jackson, P. (Dir.). (2001). *The Lord of the Rings: The Fellowship of the Ring.* [Película]. New Line Cinema. https://www.filmaffinity.com/es/film750283.html

Jackson, P (Dir.). (2003). *The Lord of the Rings: The Return of the King* [Película]. New Line Cinema. https://www.filmaffinity.com/es/film226427.html

Gilbar, M. (Prod.), y Kasdan, L. (Dir.). (2022). *Light & Magic* [Serie documental]. Lucasfilm Ltd. https://www.filmaffinity.com/es/film147187.html

Kurtz, G. (Prod.), y Kershner, I (Dir.). (1980). *Star Wars. Episode V: The Empire Strikes Back* [Película]. Lucasfilm Ltd. https://www.filmaffinity.com/es/film605090.html

Kubrick, S. (Dir.). (1968). *2001: A Space Odyssey* [Película]. Metro-Goldwyn-Mayer. https://www.filmaffinity.com/es/film171099.html

Lang, F. (Dir.). (1927). *Metropolis* [Película]. UFA. https://www.filmaffinity.com/es/film282386.html

Lánthimos, G. (Prod.), y Lánthimos, G. (Dir.). (2023). *Poor things* [Película]. Film4 Productions https://www.filmaffinity.com/es/film270023.html

Arnold, B. (Prod.), y Lasseter, J. (Dir.). (1995). *Toy Story* [Película]. Pixar Animation Studios. https://www.filmaffinity.com/es/film459936.html

Johnson, M. (Prod.), y Levinson, B. (Dir.). (1985). *Young Sherlock Holmes* [Película]. Amblin Entertainment. https://www.filmaffinity.com/es/film662675.html

Kushner, D. (Prod.), y Lisberger, S. (Dir.). (1982). *Tron* [Película]. Walt Disney Productions. https://www.filmaffinity.com/es/film488334.html

Kurtz, G. (Prod.), y Lucas, G (Dir.). (1977). *Star Wars: Episode IV - A New Hope* [Película]. Lucasfilm. https://www.filmaffinity.com/es/film605090.html

Lumière, L. (Dir.). (1895). *La Sortie de l'usine Lumière à Lyon* [Película]. https://www.filmaffinity.com/es/film333390.html

Lumière, L. (Dir.). (1896). *L'Arrivée d'un train à La Ciotat* [Película]. https://www.filmaffinity.com/es/film753379.html

Méliès, G. (Dir.). (1896). *Escamotage d'une dame au théâtre Robert Houdin* [Película]. https://www.filmaffinity.com/es/film694896.html

Méliès, G. (Dir.). (1900). *L'homme orchestre* [Película]. https://www.filmaffinity.com/es/film959541.html

Méliès, G. (Dir.). (1902). *Le Voyage dans la Lune* [Película]. Star Film. https://www.filmaffinity.com/es/film363136.html

Bennett, H. (Prod.), y Nimoy, L. (Dir.). (1986). *Star Trek IV. The Voyage Home* [Película]. Paramount Pictures. https://www.filmaffinity.com/es/film691196.html

Chernin, P. (Prod.), y Reeves, M. (Dir.). (2014). *Dawn of the Planet of the Apes* [Película]. Chernin Entertainment. https://www.filmaffinity.com/es/film613575.html

Schultz, J. (Dir.). (1995). *The Making of 'Jurassic Park'* [Documental]. Amblin Entertainment. https://www.filmaffinity.com/es/film310618.html

Kennedy, K. (Prod.), y Spielberg, S. (Dir.). (1982). *E.T.: The Extra-Terrestrial* [Película]. Universal Pictures. https://www.filmaffinity.com/es/film627362.html

Marshall, F. (Prod.), y Spielberg, S. (Dir.). (1985). *Raiders of the Lost Ark* [Película]. Lucasfilm. https://www.filmaffinity.com/es/film727297.html

Kennedy, K. (Prod.), y Spielberg, S. (Dir.). (1993). *Jurassic Park* [Película]. Amblin Entertainment. https://www.filmaffinity.com/es/film152490.html

Feige, K. (Prod.), y The Russo. (Dir.). (2019). *Avengers: Endgame* [Película]. Marvel Studios. https://www.filmaffinity.com/es/film993884.html

Bruckheimer, J. (Prod.), y Verbinski, G. (Dir.). (2006). *Pirates of the Caribbean: Dead Man's Chest* [Película]. Walt Disney Pictures. https://www.filmaffinity.com/es/film616895.html

Verbinski, G. (Dir.). (2011). *Rango* [Película]. Nickelodeon Movies. https://www.filmaffinity.com/es/film126636.html

Frost, B. (Guion.), y Volk-Weiss, B. (Dir.). (2019). Jurassic Park [Episodio de serie]. En C. Henson (Productor ejecutivo), *The Movies that Made Us*. Netflix. https://www.filmaffinity.com/es/film550656.html

Silver, J. (Prod.), y Wachowski, L. (Dir.). (1999). *The Matrix.* [Película]. Warner Bros. Pictures. https://www.filmaffinity.com/es/film932476.html

Feige, K. (Prod.), y Watts, J. (Dir.). (2017). *Spider-Man: Homecoming* [Película]. Columbia Pictures. https://www.filmaffinity.com/es/film478990.html

Finerman, W. (Prod.), y Zemeckis, R. (Dir.). (1994). *Forrest Gump* [Película]. Paramount Pictures. https://www.filmaffinity.com/es/film444796.html

Marsden, M. (Prod.), y Zondag, R & Leighton, E. (Dir.). (año). *Dinosaur* [Película]. Walt Disney Pictures. https://www.filmaffinity.com/es/film946072.html

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

LISTA DE ABREVIATURAS Y SIGLAS

FORMA	EQUIVALENCIA
VEX	Vector EXpressions
VFX	Visual Effects
TD	Technical Director
CGI	Computer Generated Images
FLIPS	Fluid Implicit Particles
RBD	Rigid Body Dynamics
RAM	Random Access Memory
TTL	Transistor-Transistor Logic
PDP	Programmed Data Processor
ILM	Industrial Light and Magic
API	Application Programming Interface
HDA	Houdini Digital Assets
NURBS	Non-Uniform Rational B-Splines
HOM	Houdini Object Model
SOP	Surface OPerators
DOP	Dynamic OPerators
VOP	Vector OPerators