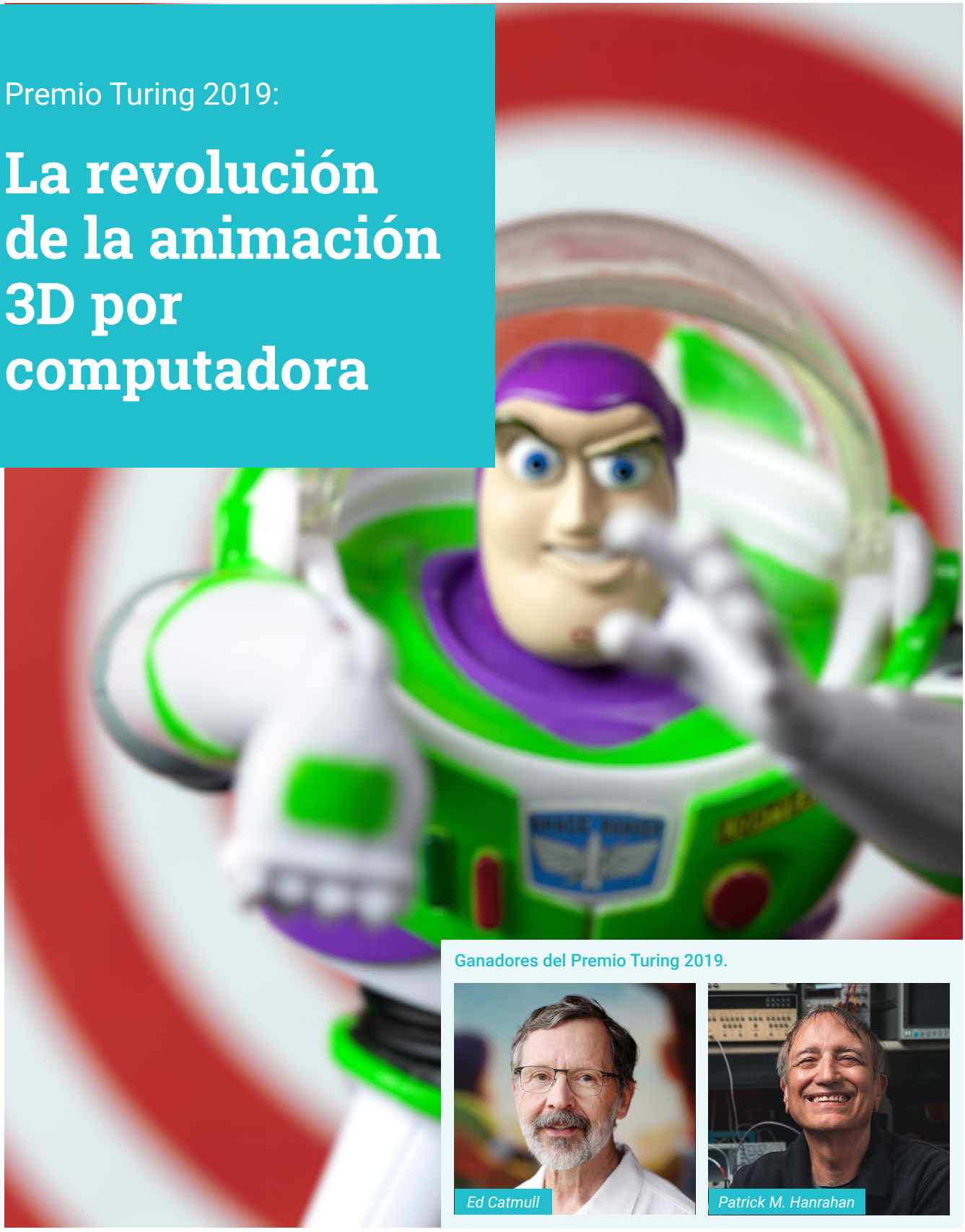
Premio Turing 2019:

La revolución de la animación 3D por computadora

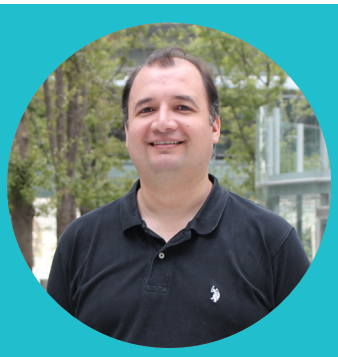


Ganadores del Premio Turing 2019.

Ed Catmull

Patrick M. Hanrahan

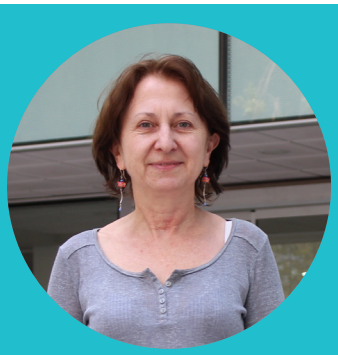
Fuente: Ed Catmull https://awards.acm.org/award-winners/catmull_1244219 | Patrick Hanrahan: https://news.stanford.edu/2020/03/18/pat-hanrahan-wins-turing-award/



BENJAMÍN BUSTOS

Profesor Titular del Departamento de Ciencias de la Computación, Universidad de Chile. Investigador Asociado del Instituto Milenio Fundamentos de los Datos. Doctor en Ciencias Naturales por la Universidad de Konstanz, Alemania. Líneas de investigación: recuperación de información multimedia basada en contenido, búsqueda por similitud y bases de datos multimedia.

bebustos@dcc.uchile.cl



NANCY HITSCHFELD KAHLER

Profesora Asociada del Departamento de Ciencias de la Computación, Universidad de Chile. Miembro del CEnter for Modern Computational ENgineering (CEMCEN). Doctora en Technischen Wissenschaften por la ETH-Zurich, Suiza. Líneas de investigación: mallas de polígonos y poliedros, algoritmos paralelos (computación en GPU), algoritmos en ciencia e ingeniería computacional y educación en computación. Participa de comisiones y actividades para atraer mujeres a STEM.

nancy@dcc.uchile.cl

Este reconocimiento fue otorgado en el año 2019 a Edwin E. Catmull y Patrick M. Hanrahan, por sus contribuciones en el área de *Computer-Generated Imagery* (imágenes generadas por computadora o CGI). Estas contribuciones han liderado los avances en el área de la computación gráfica, innovando en sus conceptos fundamentales, algoritmos, hardware y software. Es interesante destacar que este Premio Turing 2019 lo comparten dos personas con perfiles bastante distintos. Por una parte, Catmull es un emprendedor e innovador que tuvo un gran impacto en la industria del cine, pero su producción científica tiene relativamente pocas citas. En cambio, Hanrahan es un académico más tradicional, con alto impacto en investigación medido en citas. El Premio Turing 2019 reconoce tanto los aportes de Catmull como de Hanrahan desde sus respectivas perspectivas, dado su innegable impacto en el desarrollo de la computación gráfica en 3D y en CGI.

En las siguientes secciones de este artículo introducimos primero en qué consiste *Computer-Generated Imagery*, luego describimos quién es cada uno de los galardonados y sus contribuciones más importantes, y finalmente concluimos con el impacto que han tenido sus contribuciones no sólo en el área de la computación gráfica, sino también en el cine y en otras áreas de la ciencia e ingeniería.

Imágenes generadas por computadora

El concepto de imágenes generadas por computadora o CGI (por la sigla en inglés de *Computer-Generated Imagery*) se refiere a la generación de imágenes y gráficos 3D aplicadas al arte, cine, televisión, etc., mediante el uso de computadores utilizando algoritmos y técnicas de computación gráfica. La ventaja de utilizar CGI es que permite a los creadores tener una libertad virtualmente ilimitada para generar escenas, incluso "rompiendo" las leyes de la física, permitiéndoles crear escenas que serían muy difíciles o muy costosas de recrear en un escenario real.

Los inicios de la CGI se remontan a las primeras décadas de desarrollo de la computación como ciencia. Ya a finales de la década de los cincuenta, en el film *Vértigo* de Alfred Hitchcock se utilizó CGI para generar una animación en 2D que corresponde a la secuencia de apertura de esta película. En la década de los setenta se empezó a utilizar CGI para generar, en forma rudimentaria aún, pequeñas escenas de acción 2D en películas, y también se empezó a utilizar animación en 3D. Luego, en los años ochenta la película *TRON* fue una de las primeras en hacer uso intensivo de CGI para generar secuencias largas de animación en 3D, de varios minutos de duración. Finalmente, la primera película que fue 100% generada utilizando CGI fue *Toy Story* en 1995 (ver Figura 1). De ahí en adelante, los avances en las técnicas de CGI y en la capacidad de cómputo de los computadores actuales, han permitido el ir generando efectos cada vez más espectaculares y realistas.

El propósito de la animación en 3D es generar escenarios en tres dimensiones con el uso del computador. Para lograr esto, primero es necesario definir cómo se representará la información 3D en el computador. Una de las principales formas para hacer esto es representando los objetos como mallas de polígonos (triángulos o cuadriláteros), en donde cada polígono de la malla se define por las coordenadas en el espacio cartesiano tridimensional de sus vértices. Al colocar polígonos en forma adyacente se pueden formar las superficies de los objetos que se están modelando. Adicionalmente, se puede representar la orientación de la superficie (hacia el interior o hacia el exterior) usando el orden en que se almacenan los vértices de cada polígono. La ventaja que tiene

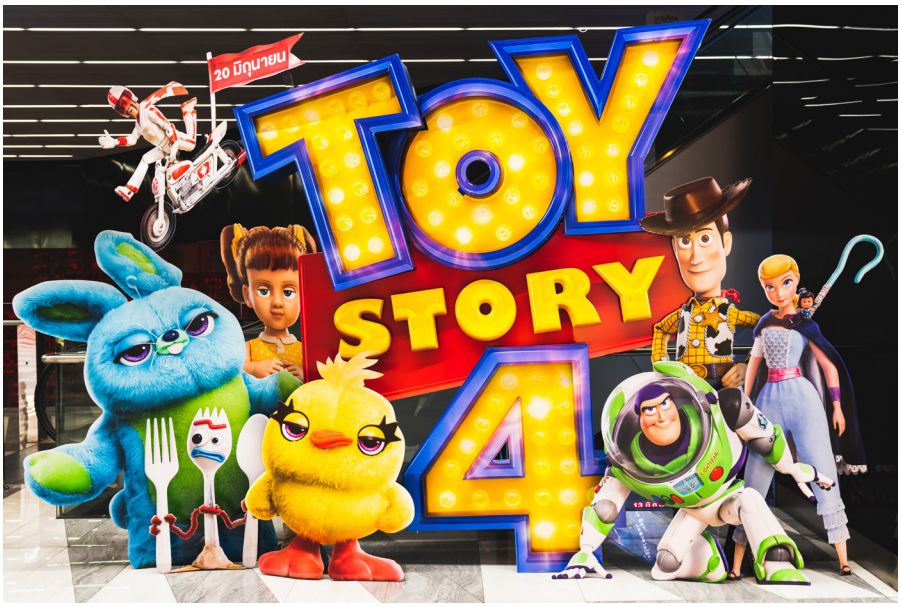


Figura 1. Cuarta parte de la saga *Toy Story*, cuyas películas de animación 3D fueron totalmente producidas por computadora (CGI).

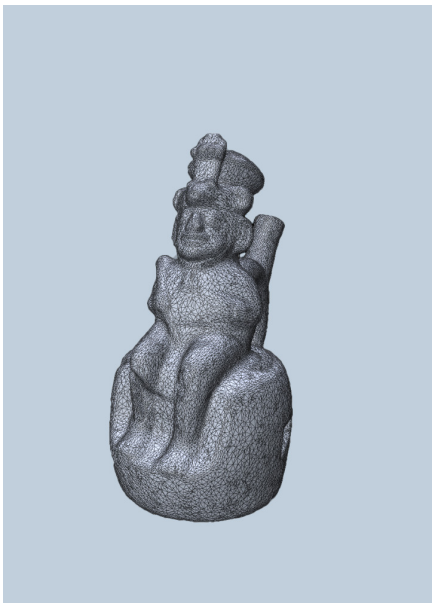


Figura 2. Ejemplo de objeto 3D representado como malla de polígonos.

el usar mallas de polígonos es que incluso con pocos polígonos se pueden modelar objetos complejos, lo que permite tener una representación simple del objeto y que no ocupa mucha memoria en el computador. En caso que se requiera mayor precisión o nivel de resolución, siempre es posible refinar las mallas y agregar más polígonos para definir la superficie del objeto con mayor nivel de detalle. La Figura 2 muestra un ejemplo de una malla de polígonos que representa un objeto 3D.

Las primeras animaciones generadas usando CGI utilizaron mallas de triángulos para representar objetos simples. Un ejemplo notable es una animación computarizada en 3D de una mano creada por Edwin E. Catmull y Fred Parke en 1972 en la Universidad de Utah. Para este proyecto, Catmull creó un modelo de yeso de su propia mano. Luego, junto a Parke midieron y calcularon manualmente una triangulación 3D del modelo de yeso, usando un par de cientos de triángulos. Finalmente, introdujeron toda esta información en un computador, con lo que produjeron una visualización en 3D de su mano. En el video titulado "A Computer Animated Hand" se observa el modelo 3D de la mano, que puede rotar y flexionar los dedos.[1] Este video ha sido descrito como "revolucionario" para su época, y fundó las bases para todo el desarrollo posterior de la CGI [1].

Edwin E. Catmull

Edwin Catmull fue cofundador de Pixar Animation Studios y presidente de Pixar y Walt Disney Animation Studios. Obtuvo el grado de Bachelor of Science en Física y Ciencias de la Computación (1970) y el PhD en Ciencias de la Computación (1974) en la Universidad de Utah. Durante su carrera fue vicepresidente de la División de Computación en Lucasfilm Ltd., donde dirigió el desarrollo en áreas de computación gráfica, edición de videos, videojuegos y audio digital.

Su motivación por crear películas nació desde muy pequeño inspirado por las películas de Walt Disney como *Peter Pan* y *Pinocho*. Él creó animaciones armando cuadernillos de imágenes, en que página a página contenían dibujos que varían gradualmente. Al mostrar rápidamente las páginas consecutivas, las imágenes parecían animarse simulando un movimiento.

Durante su años en la universidad, realizó dos aportes fundamentales a la

1 | https://vimeo.com/59434349.

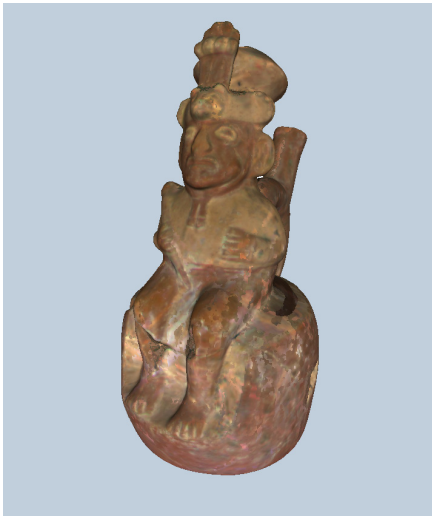


Figura 3. La misma malla del objeto 3D de la Figura 2, pero con texturas aplicadas sobre la malla.

computación gráfica: (i) mapeo de texturas (*texture mapping*) y (ii) parches bicúbicos (*bicubic patches*).

Mapeo de texturas. Una textura es una imagen que contiene la forma en que queremos pintar una parte de un objeto, difícil de especificar geométricamente, como por ejemplo la rugosidad de la piel de la mano mencionada en la sección anterior. Aplicado a nuestro ejemplo, el mapeo de texturas permite hacer calzar una imagen real de la piel de una mano, en dos dimensiones, a cada triángulo de la malla que la representa, una vez que el modelo de la mano ha sido proyectado a dos dimensiones para generar su imagen en el computador. El mapeo se realiza desde la textura al triángulo proyectado, en donde los colores de la textura son usados para pintar los píxeles que están contenidos en el triángulo a pintar (ver Figura 3). El aporte de Edmund Catmull fue diseñar un nuevo algoritmo, para hacer calzar una textura con la proyección de un triángulo, todo como parte del mismo proceso de generar la imagen del objeto a pintar [2].

Este Premio Turing 2019 lo comparten dos personas con perfiles bastante distintos [...]: Catmull es un [...] innovador que tuvo un gran impacto en la industria del cine [...] En cambio, Hanrahan es un académico [...] con alto impacto en investigación.

Hasta ese momento, el proceso de mapear texturas era impreciso y lento pues consistía en el proceso inverso: dado un píxel a pintar, se buscaba en el espacio en tres dimensiones, qué parte de la textura asociada a un triángulo proyectado, le correspondía.

Parches bicúbicos. Los *bicubic patches* pueden ser vistos como polígonos de cuatro lados, en que cada lado está representado por un polinomio de grado 3 (curva cúbica). Cada lado necesita dos puntos de control adicionales y en el interior del polígono se requieren cuatro puntos adicionales, para permitir representar superficies curvas de forma más realista que polígonos planares. Catmull introdujo técnicas innovadoras para crear y pintar de manera realista *bicubic patches* en vez de polígonos planos, de las cuales surgió, junto al mapeo de texturas, la técnica del *z-buffer* (descrito al mismo tiempo por Wolfgang Strasser). Para determinar qué partes de la escena modelada (por ejemplo, qué triángulos de la malla que representa la mano mencionada más arriba) serán visibles e influyen en cómo se pinta en la imagen generada, la técnica del *z-buffer* permite seleccionar, al estar pintando la imagen, el color asociado al triángulo más cercano a la cámara (punto desde donde se mira la escena). A cada triángulo de la escena, se le aplican las transformaciones de movimiento y proyección para llevarlo a una representación normalizada, representada por un cubo (-1,-1,-1) y (1,1,1), en donde la imagen a generar coincide con el rectángulo (-1,-1,-1) y (1,1,-1). Este rectángulo se discretiza en píxeles, considerando la resolución que se desea de la imagen a generar. Cada triángulo que queda incluido dentro de este cubo se recorre en función de los píxeles (*scanline*) que cubre, y el color o textura asociada a este triángulo define como pintar estos píxeles. Cada vez que se pinta un píxel se almacena la profundidad en el eje *z* (dentro de este cubo) del triángulo que definió su color. Si aparece otro triángulo más cercano a la cara del cubo que representa la imagen a generar, se usa el color/textura de este nuevo triángulo y se recuerda esta profundidad. Esta técnica almacena, en todo momento, el color y la profundidad del triángulo que está definiendo el color actual. Cuando se terminan de recorrer los triángulos de la escena, se tiene la imagen calculada.

El algoritmo del *z-buffer* fue más tarde generalizado al *A-buffer* para permitir el manejo de transparencias, y complementado con técnicas que simplifican los modelos tridimensionales que tienen una enorme cantidad de polígonos a rasterizar, el *z-buffer* puede aplicarse a los polígonos que tendrán un impacto en la imagen final.

Cabe destacar que bajo su conducción por más de treinta años, Pixar realizó una serie de películas muy exitosas usando el software *RenderMan*. Este software ha sido usado en 44 de las últimas 47 películas nominadas por la Academy Award en la categoría de efectos visuales. Entre estas películas se encuentran *Avatar*, *Titanic*, *La Bella y la Bestia*, y *El Señor de los Anillos*. En sus laboratorios, fueron inventadas y publicadas una serie de tecnologías

El impacto de los algoritmos, fundamentos teóricos y software que desarrollaron Catmull y Hanrahan [...] no sólo se mide en citas o cantidad de artículos [...], sino también [...] en Premios Óscar.

fundacionales entre las cuales están composición de imágenes, *motion blur*, y simulación de ropa, entre otras.

Patrick M. Hanrahan

Pat Hanrahan es actualmente Profesor de Ciencias de la Computación e Ingeniería Eléctrica en el Computer Graphics Laboratory de la Universidad de Stanford. Obtuvo su grado de Bachelor of Science en Ingeniería Nuclear (1977) y un PhD en Biofísica (1985) de la Universidad de Wisconsin-Madison. Él fue una de las primeras personas contratadas en Pixar por Edwin Catmull. Como científico senior permaneció allí desde el año 1986 hasta el año 1989. Entre los años 1991 y 1994 fue Profesor Asociado en la Universidad de Princeton y desde el año 1994 hasta ahora está en la Universidad de Stanford.

Durante su estadía en Pixar, Hanrahan lideró el desarrollo del nuevo sistema gráfico *RenderMan*, software que permite que formas curvas puedan ser pintadas de manera realista considerando iluminación y las propiedades de los materiales (*shaders*). La idea clave fue separar el comportamiento de reflexión de la luz de la geometría del objeto y calcular el color, transparencia, y textura sobre puntos de la superficie del objeto [3]. *RenderMan* también incluyó el concepto de *z-buffering* y los algoritmos sobre los parches bicúbicos introducidos por Edwin Catmull. *RenderMan* es considerado el modelo estándar para generar efectos visuales en CGI.

La contribuciones de Hanrahan son casi innumerables; ha creado nuevos conceptos, modelos, algoritmos tanto secuenciales como paralelos, lenguajes de programación gráficos y para las GPU's, y software para *rendering* realístico de objetos, entre otras. Es difícil decidir cuales son sus aportes más importantes, pero sin duda entre estos se encuentran: (i) la creación de un nuevo método, *light field rendering,* que da al usuario la sensación de volar a través de las escenas, generando nuevas vistas desde puntos de visión arbitrarios sin información de profundidad ni geométrica, sino muestreando pedazos (*slices*) en grandes arreglos de imágenes previamente digitalizadas o pintadas [4]; (ii) técnicas para representar la piel y el pelo usando *subsurface scattering* [5]; (iii) algoritmos para modelar efectos complejos de la interacción entre distintas fuentes de luz y los objetos de la escena (iluminación global) usando *Monte Carlo ray tracing* [6]; y (iv) lenguajes para programar GPU's.

Los lenguajes para programar las GPU's (unidades de procesamiento gráfico) han sido un aporte revolucionario pues permitieron que animaciones y videojuegos tridimensionales complejos se puedan realizar en tiempo real. En este ámbito, apenas aparecieron las GPU's en los años noventa, Hanrahan y sus estudiantes extendieron el lenguaje de *shading* incluido en *RenderMan* para usar la GPU, motivando más tarde el desarrollo de versiones comerciales y el lenguaje de shading GLSL de OpenGL, la librería gráfica abierta más usada en el mundo. Más aún, en los años 2000, nuevamente junto a sus estudiantes, desarrollaron el lenguaje Brook [7], un lenguaje que permitió comenzar a usar las GPU's como poder de cálculo de propósito general y no sólo para aplicaciones gráficas. Brook motivó y condujo al desarrollo de Cuda, un lenguaje de programación de propósito general para las tarjetas gráficas NVidia.

Epílogo

El Premio Turing 2019 fue otorgado a Edwin Catmull y Patrick Hanrahan por sus contribuciones en CGI y en animación computarizada 3D. El impacto de los algoritmos, fundamentos teóricos y software que desarrollaron Catmull y Hanrahan durante sus carreras no sólo se mide en citas o en cantidad de artículos científicos destacados, sino que también se mide en Premios Óscar. Todos en nuestra vida cotidiana podemos ver ejemplos en donde sus contribuciones fueron fundamentales, por ejemplo al sentarnos al ver una película con animaciones o al jugar a nuestro videojuego favorito. Pero, sus contribuciones no sólo se limitan a la industria del entretenimiento. El desarrollo del lenguaje Brook permitió que los procesadores gráficos conocidos como GPU's, actualmente con miles de procesadores y disponibles a un precio razonable en notebooks y computadores de escritorio, pudieran ser usados como unidades de cálculo multipropósito y no sólo en el proceso de *rendering* gráfico. Es así como hoy en día se usan para para correr algoritmos incluidos en aplicaciones computacionales de alto desempeño, tales como simulaciones numéricas, análisis de imágenes en biología y medicina, entrenamiento de algoritmos de *machine learning* sobre datos masivos para aplicaciones de inteligencia artificial, entre otras. Muchos descubrimientos y avances aún por venir en el futuro se los debemos en parte al trabajo de Catmull y Hanrahan. ▪

Agradecimientos

Nuestros agradecimientos a Iván Sipirán, quien amablemente contribuyó con las imágenes para las Figuras 2 y 3 de este artículo.

REFERENCIAS

[1] Andrew Utterson. A Computer Generated Hand. Ensayo para el National Film Registry. https://www.loc.gov/static/programs/national-film-preservation-board/documents/computer_hand2.pdf (último acceso: 14 de abril de 2021).

[2] E. Catmull, AR Smith. 3-D transformations of images in scanline order. ACM SIGGRAPH Computer Graphics 14 (3):279-285. 1980.

[3] RA Drebin, L Carpenter, P Hanrahan. Volume rendering. ACM SIGGRAPH Computer Graphics, 22 (4):65-74. 1988.

[4] Marc Levoy, Pat Hanrahan. Light field rendering. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pages 31-42. 1996.

[5] SR Marschner, HW Jensen, M Cammarano, S Worley, P Hanrahan. Light scattering from human hair fibers. ACM Transactions on Graphics (TOG) 22 (3):780-791. 2003.

[6] State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis. SIGGRAPH 2001 Course 29. 2001. Available from: https://www.researchgate.net/publication/2872516_State_of_the_Art_in_Monte_Carlo_Ray_Tracing_for_Realistic_Image_Synthesis#fullTextFileContent (último acceso: 20 de mayo de 2021).

[7] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, Pat Hanrahan. Brook for GPUs: stream computing on graphics hardware. ACM Transactions on Graphics (TOG) 23 (3):777-786. 2004.