Engenharia de

Sistemas e

Computação

PESC

50 Anos

1970 - 2020

Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

UFRJ faz 100 ANOS 1920 | 2020
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Engenharia de Sistemas e Computação
PESC

Prêmio Turing 2019
Pat Hanrahan & Ed Catmull

Claudio Esperança e Ricardo Marroquim

“ ACM named Patrick M. (Pat) Hanrahan and Edwin E. (Ed) Catmull recipients of the 2019 ACM A.M. Turing Award for fundamental contributions to 3-D computer graphics, and the revolutionary impact of these techniques on computer-generated imagery (CGI) in filmmaking and other applications. ”

Ed

Pat

CGI - Computer Graphics Imagery

Uso de computadores para gerar imagens

- Imagens estáticas
- Animações
  - 2D
  - 3D

4 conceitos fundamentais de CG

- Imagem
- Modelo
- Renderização
- Animação

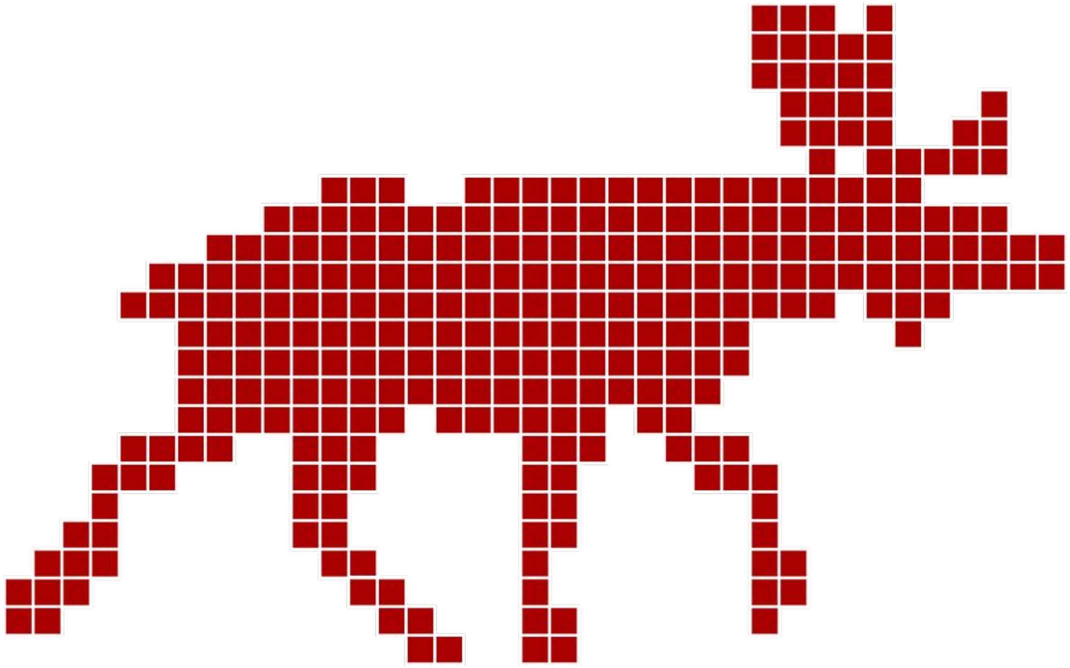
4 conceitos - Imagem

Hoje em dia ⇒ Imagens matriciais (matriz de *pixels*)

Pixel ⇒ *Picture Element* (amostra de *cor*)

Cor

- 0/1 ⇒ branco/preto
- 0...m ⇒ tons de cinza
- $[C_0, C_1, \dots C_n]$
  - n ⇒ 3 (visão humana)
  - tipicamente [R,G,B]

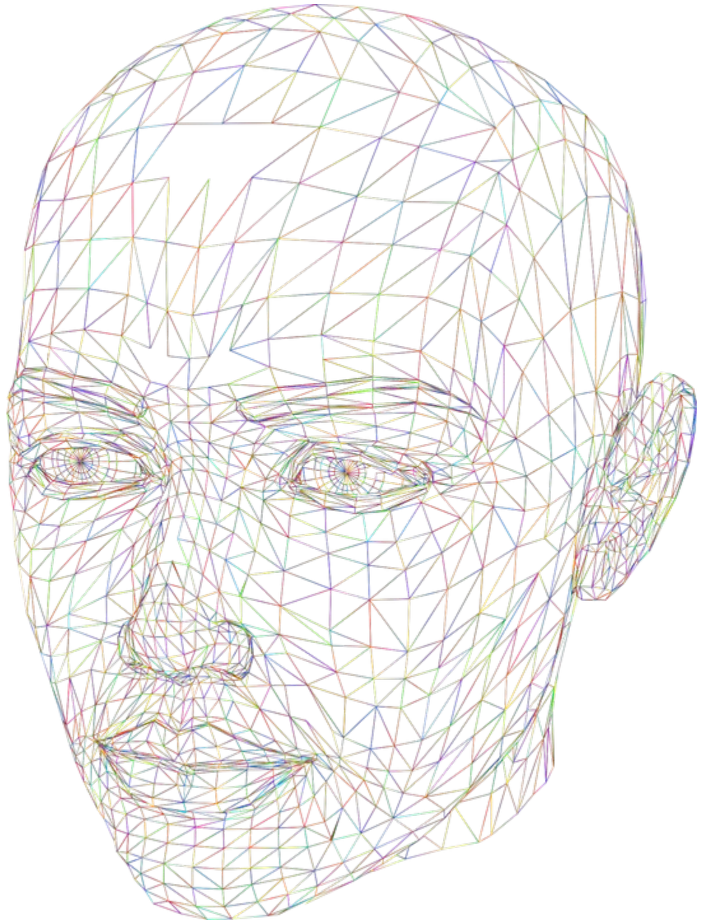


4 conceitos - Modelo

Representação do que se quer desenhar

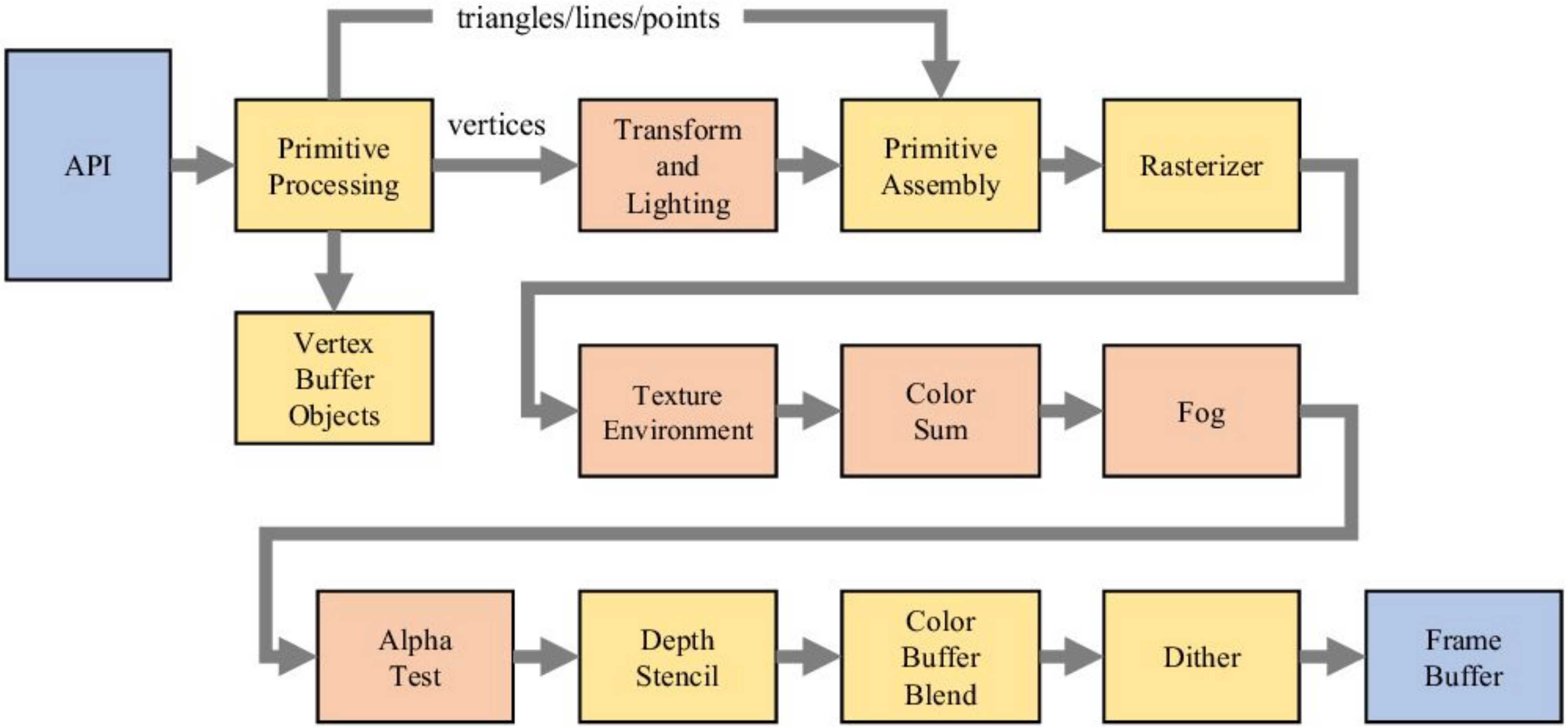
Estruturas de dados

- Geometria
- Atributos visuais
  - iluminação / cor / materiais / textura
- Tempo (para animação)



4 conceitos - Renderização

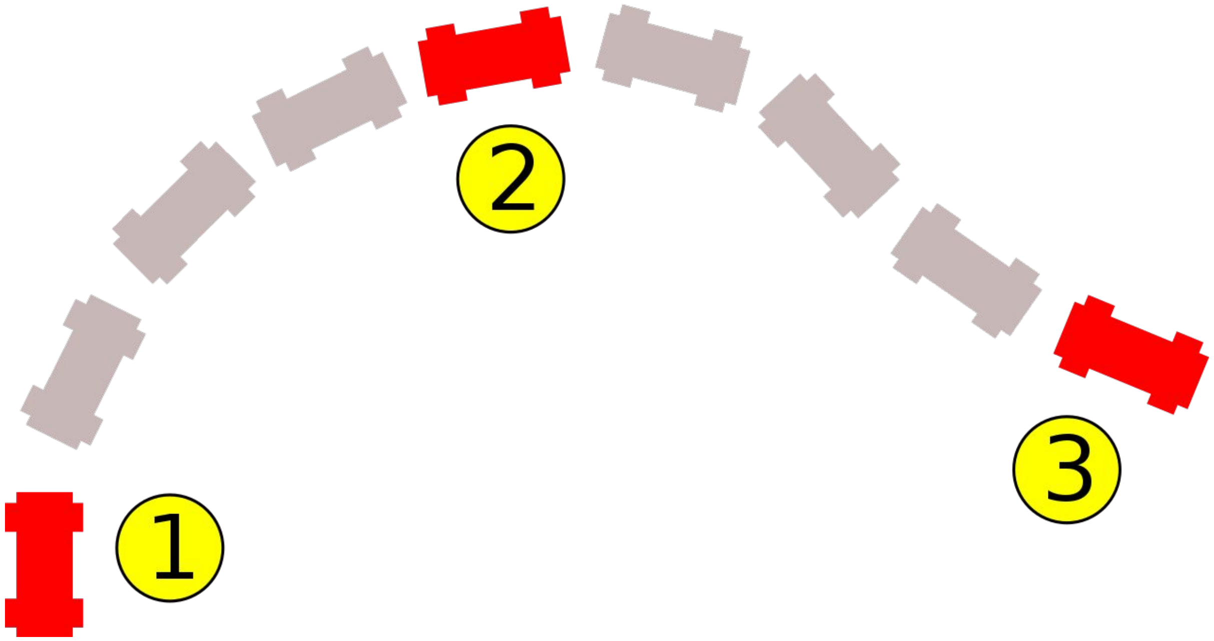
Processo de gerar ***imagens*** a partir de ***modelos***



4 conceitos - Animação

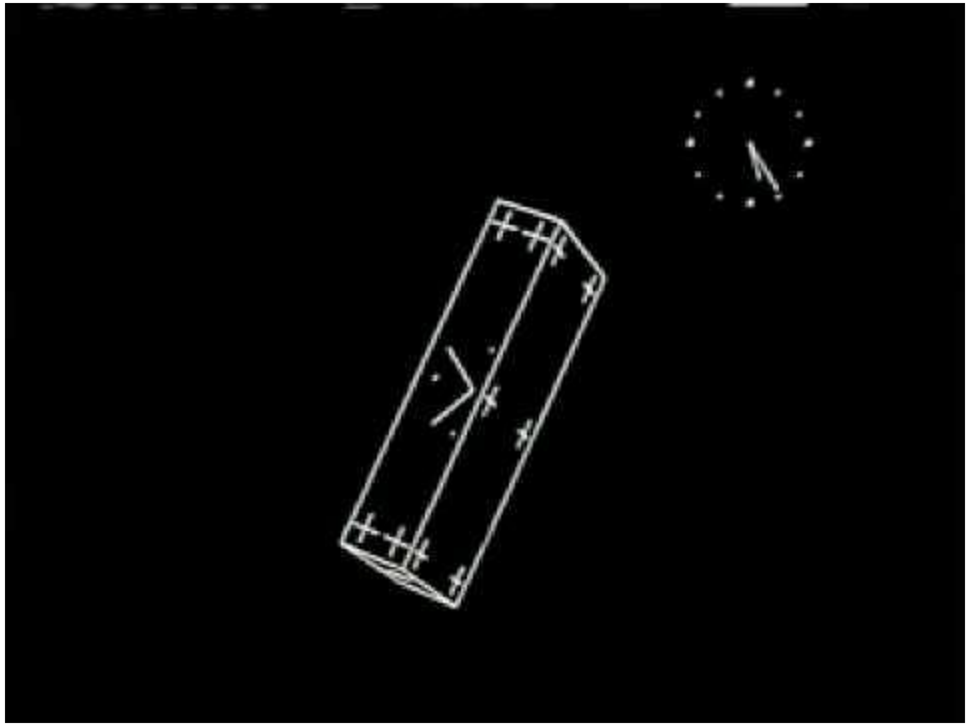
Processo de gerar imagens em rápida sucessão

- Tipicamente ≥ 24 fps
- Modelos ⇒ funções de t
  - *Keyframing*
    - Quadros chave
    - Funções de interpolação



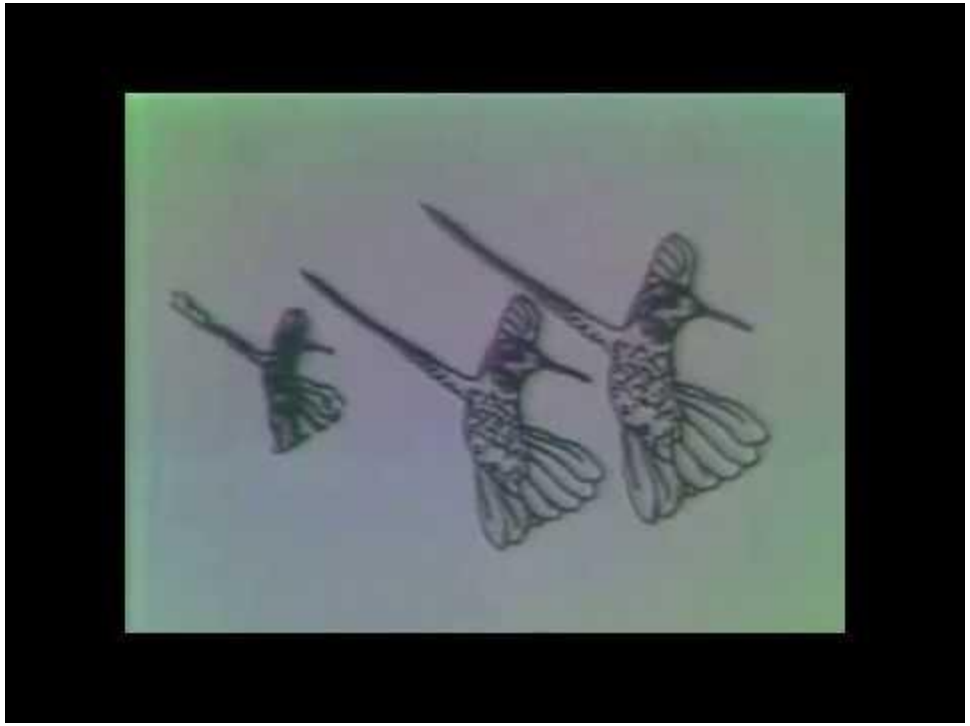
História da animação por computador

Edward Zajac *A two gyro gravity gradient altitude control system (Bell Labs 1963)*



História da animação por computador

Charles Csuri and James Shaffer “Hummingbird” (1967) - primeiro exemplo de *morphing* digital



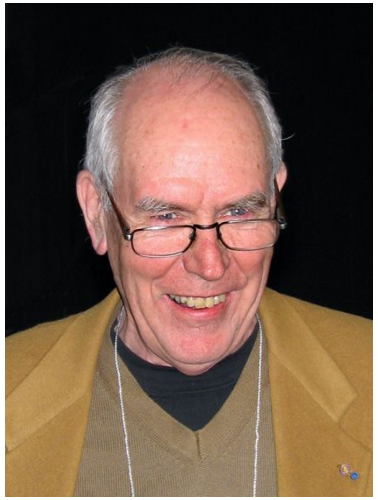
História da animação por computador

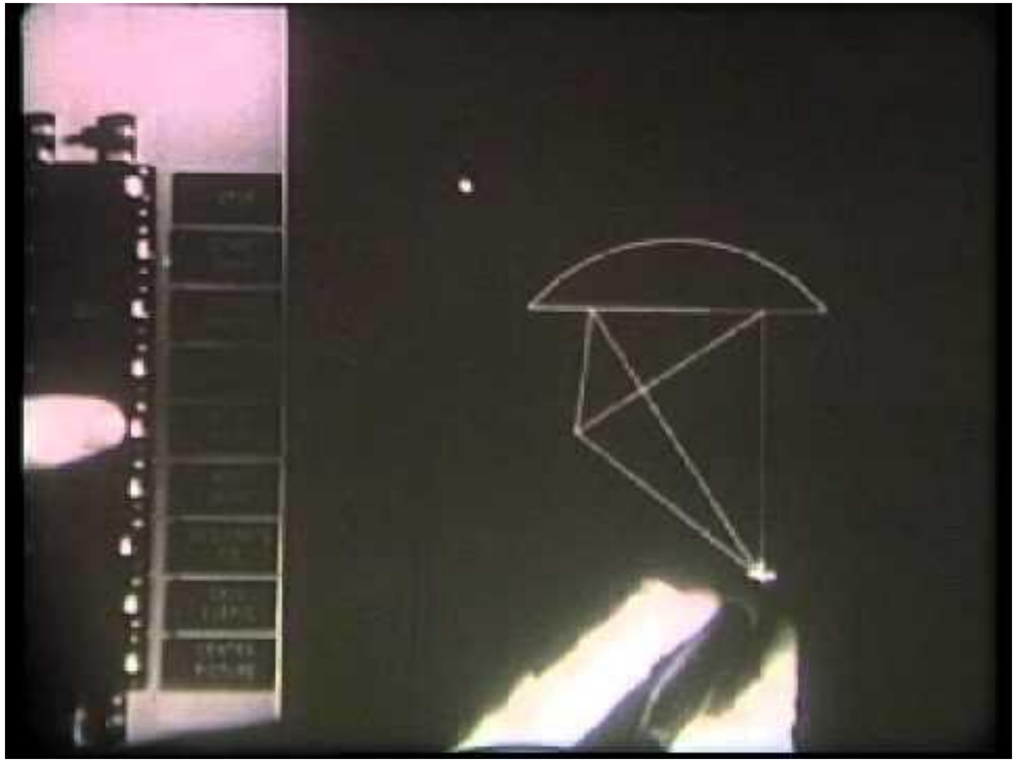
Nikolai Konstantinov - “Kitty”, criado no computador russo BESM-4 (1968)

Ed inicia o doutorado (1969)

Universidade de Utah

Orientador: Ivan Sutherland - *Turing award 1988*

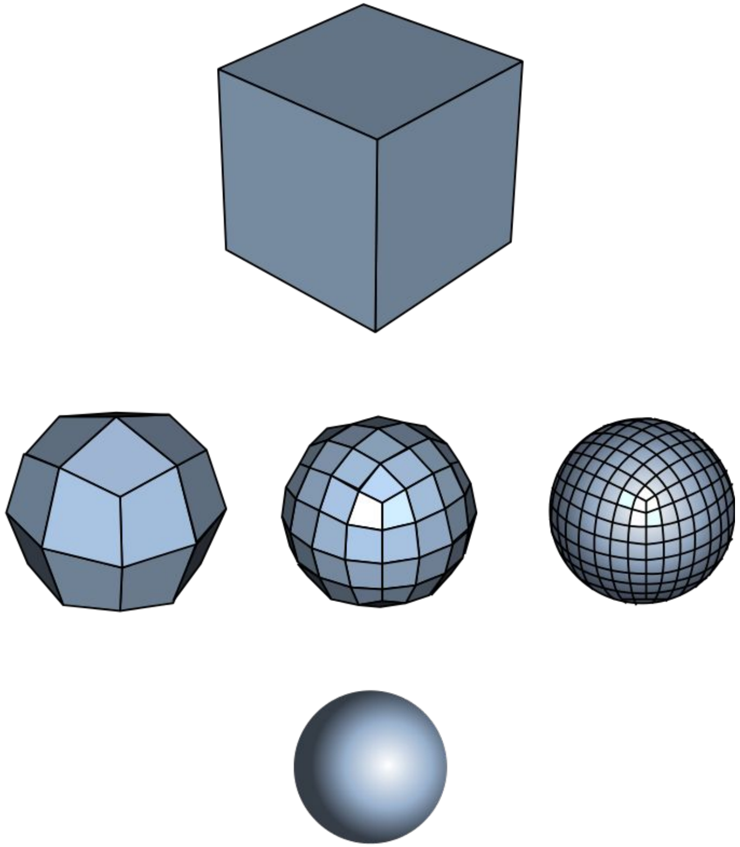




Sketchpad - 1963

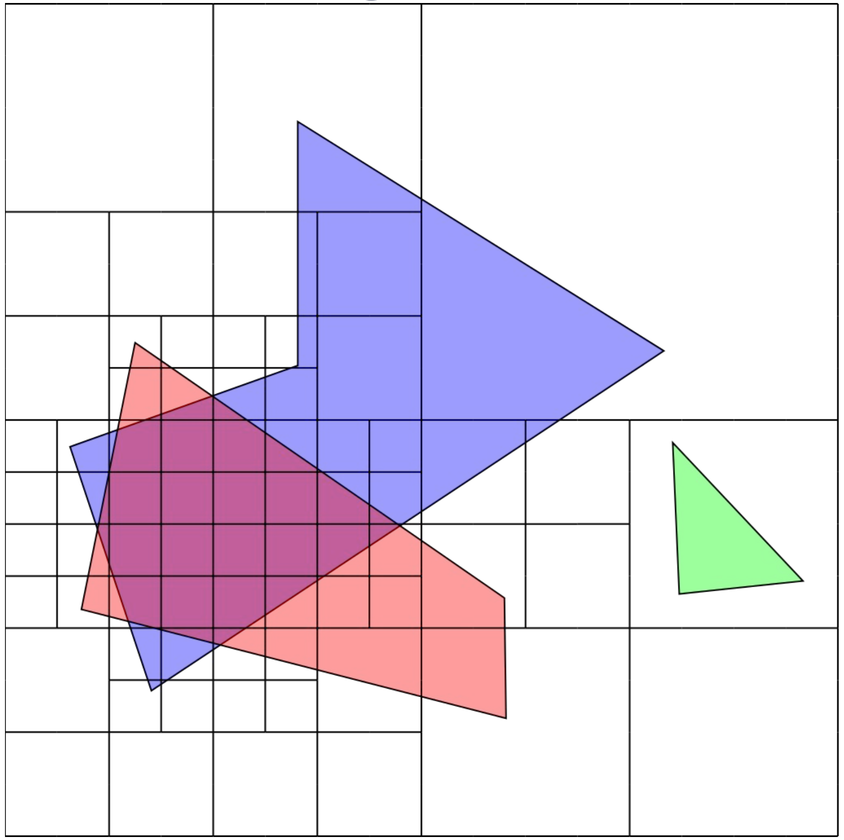
Ed é colega de classe de

- James H Clark
  - Subdivisão de Catmull-Clark
  - Fundador da Netscape e Silicon Graphics



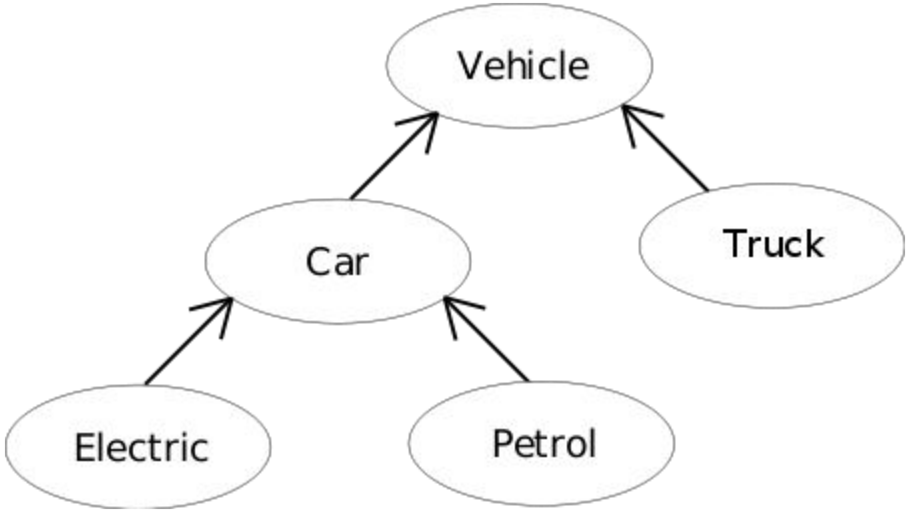
Ed é colega de classe de

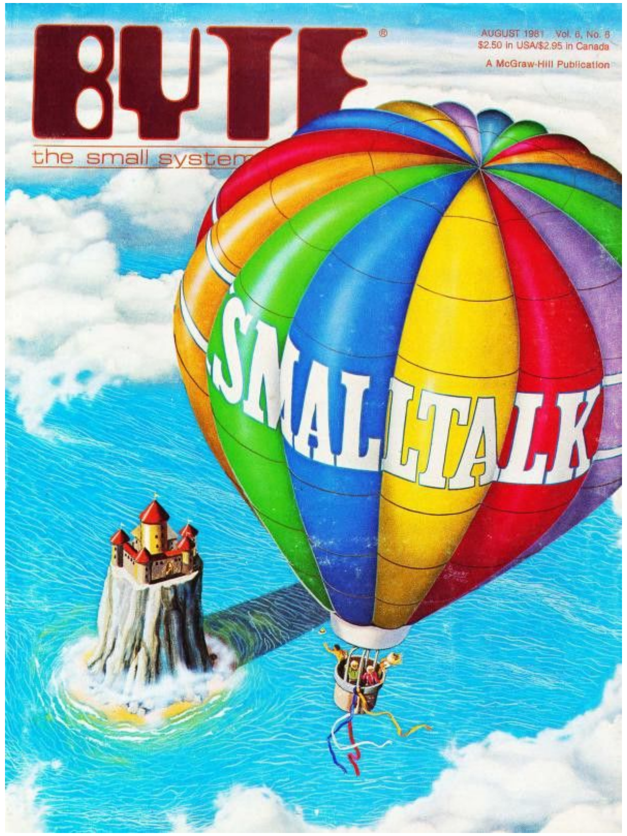
- John Warnock
  - Co-fundador da Adobe
  - Algoritmo de Warnock (visibilidade)



Ed é colega de classe de

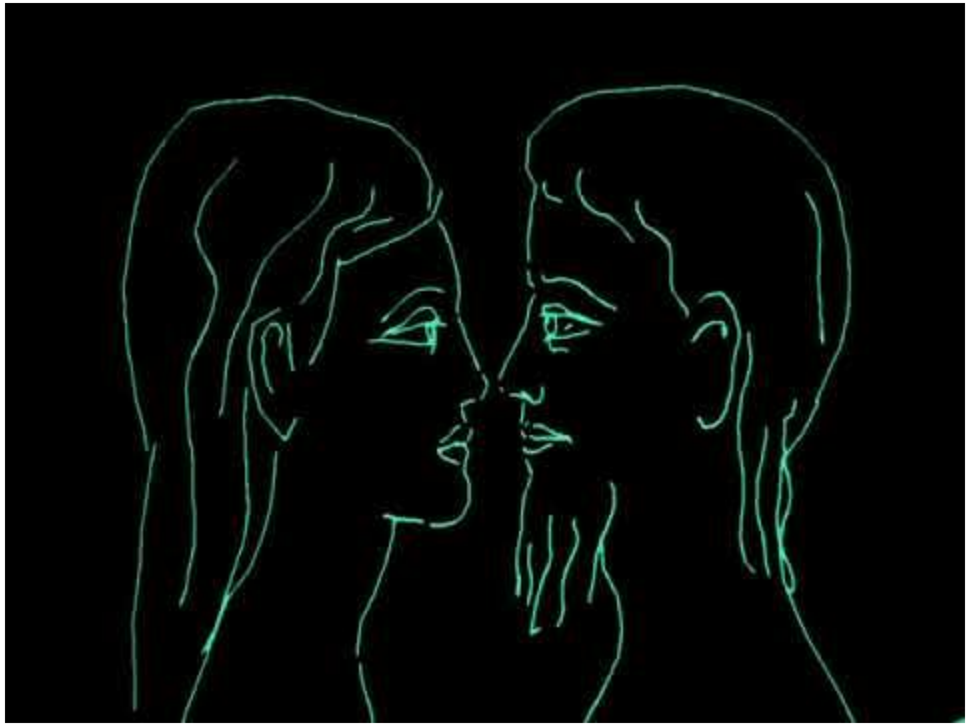
- Alan Kay
  - Pai da orientação a objetos
  - Linguagem Smalltalk





História da animação por computador

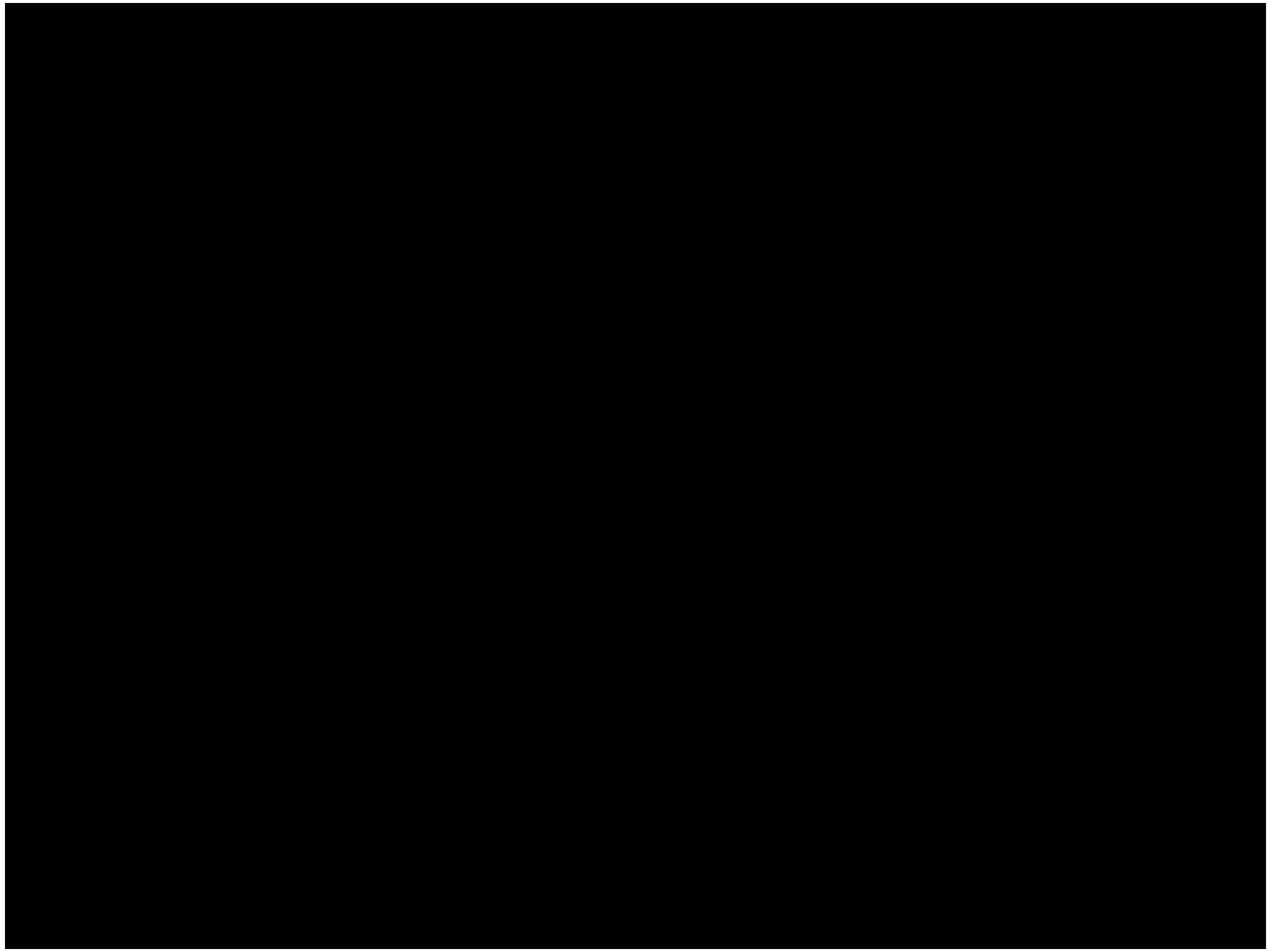
"Metadata", Peter Foldes (1971). Primeiro exemplo do emprego de quadros-chave interpolados por computador.



História da animação por computador

“A computer animated hand”, Ed Catmull (1972).

Clip usado no thriller Futureworld (1976).



Ed termina o doutorado (1974)

(1) Subdivisão de retalhos bicúbicos

CHAPTER FOUR

EXTENSION OF CUBIC SUBDIVISION TO SURFACES

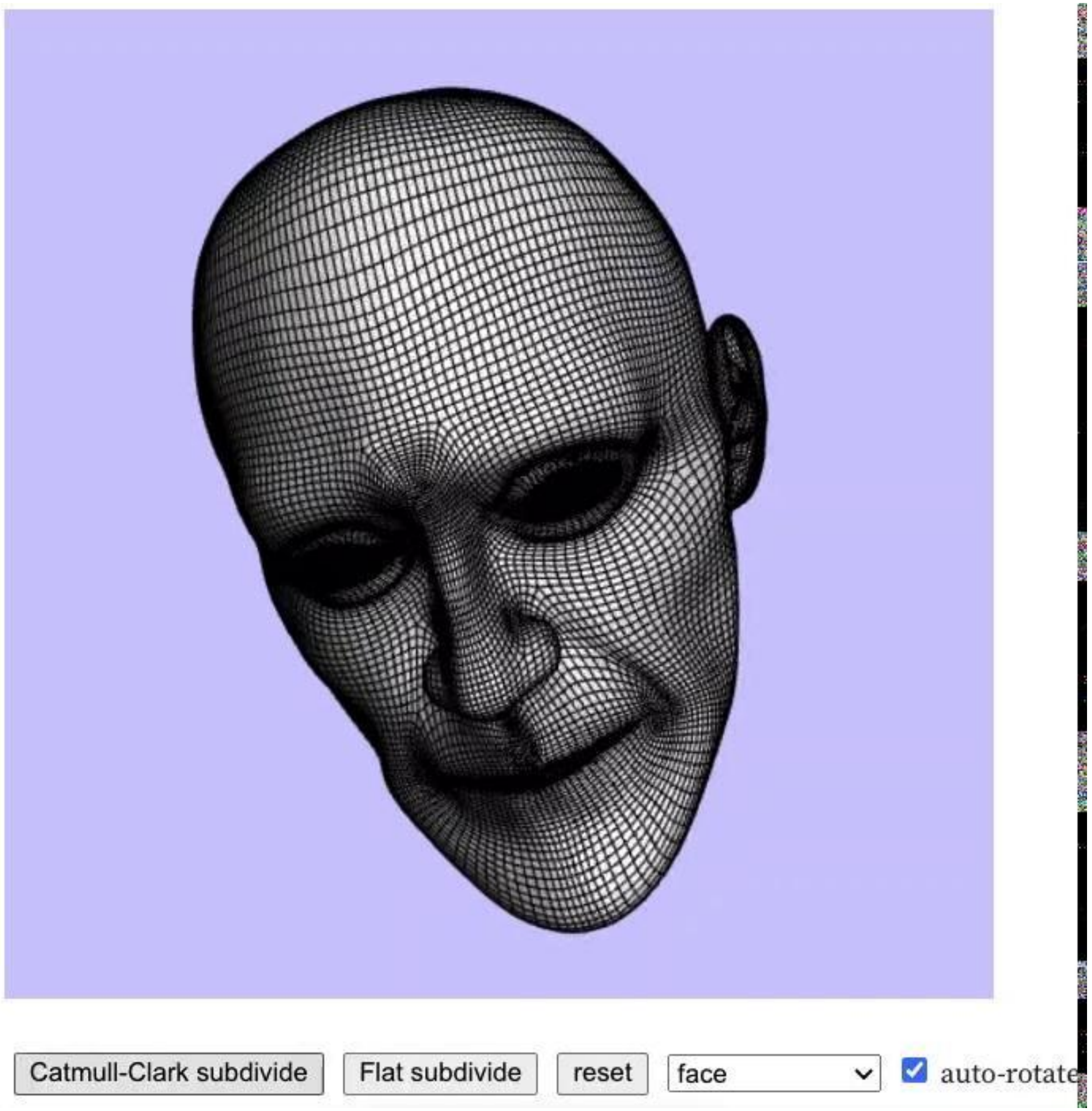
The method of subdividing cubic curves can be extended to bicubic surfaces. With a cubic curve there is a value and a correction term at each end; with a bicubic patch there is a value and three correction terms at each corner. Subdivision of the patch into four pieces means finding the midpoint of each of the sides and the midpoint of the patch.

(Subdivisão de malhas)

Permite aproximar uma superfície através de uma malha grosseira

Método de Catmull-Clark (1978)

- Modelagem 3D
- Malhas quadrangulares



https://observablehq.com/@esperanc/catmull-clark-subdivision

Engenharia de Sistemas e Computação PESC

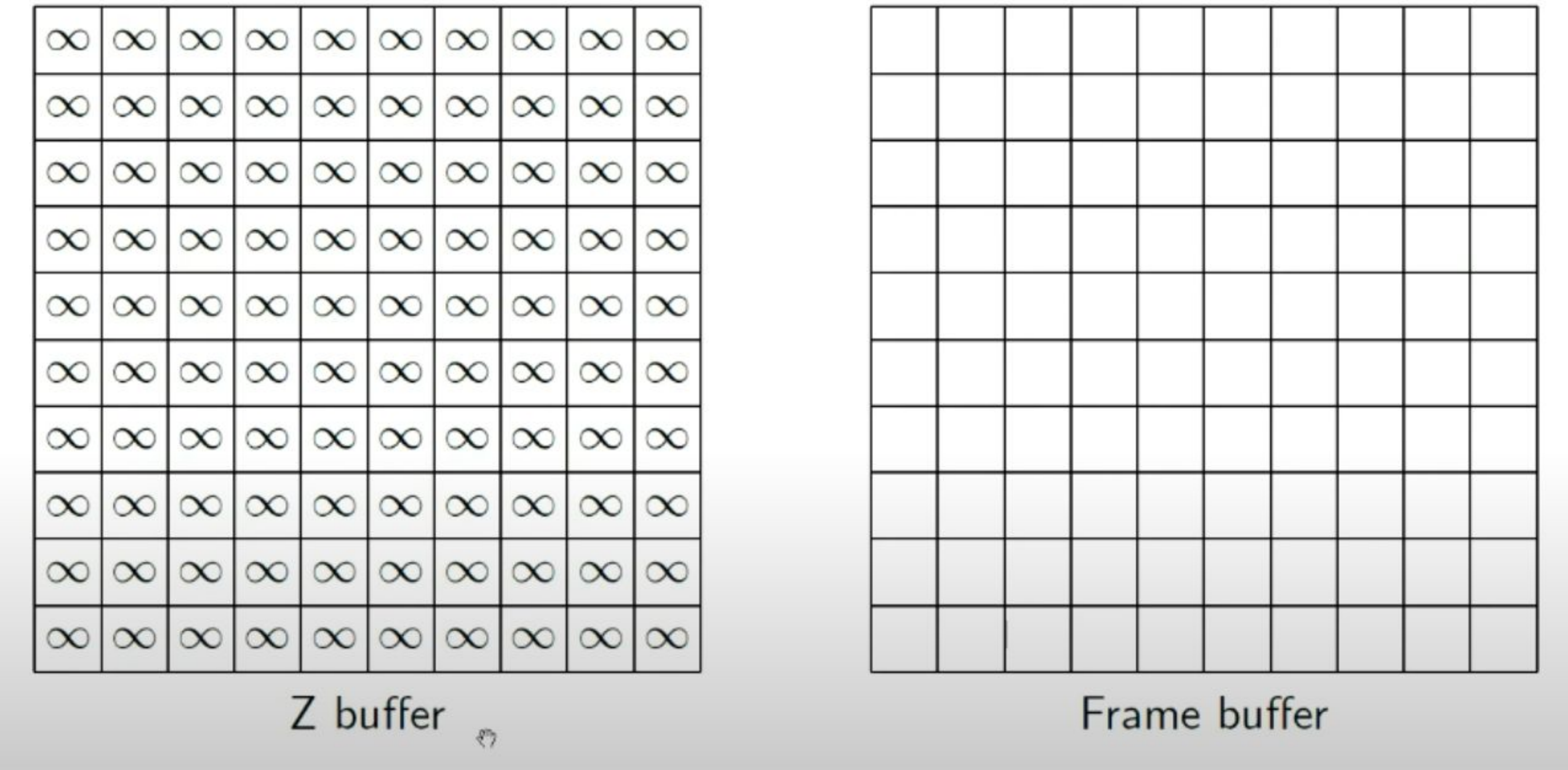
Ed termina o doutorado (1974)

(2) Descoberta independente do Z-buffer (8 meses após a tese de Wolfgang Straßer)

THE Z-BUFFER

The z-buffer is an extension of the frame-buffer idea in that the z value from the image-space of the visible object is stored at every raster-element as well as the intensity. The z value of any new point to be written into the buffer is compared with the z value of the point already there. If the new point is behind, it is discarded. If it is in front it replaces the old value.

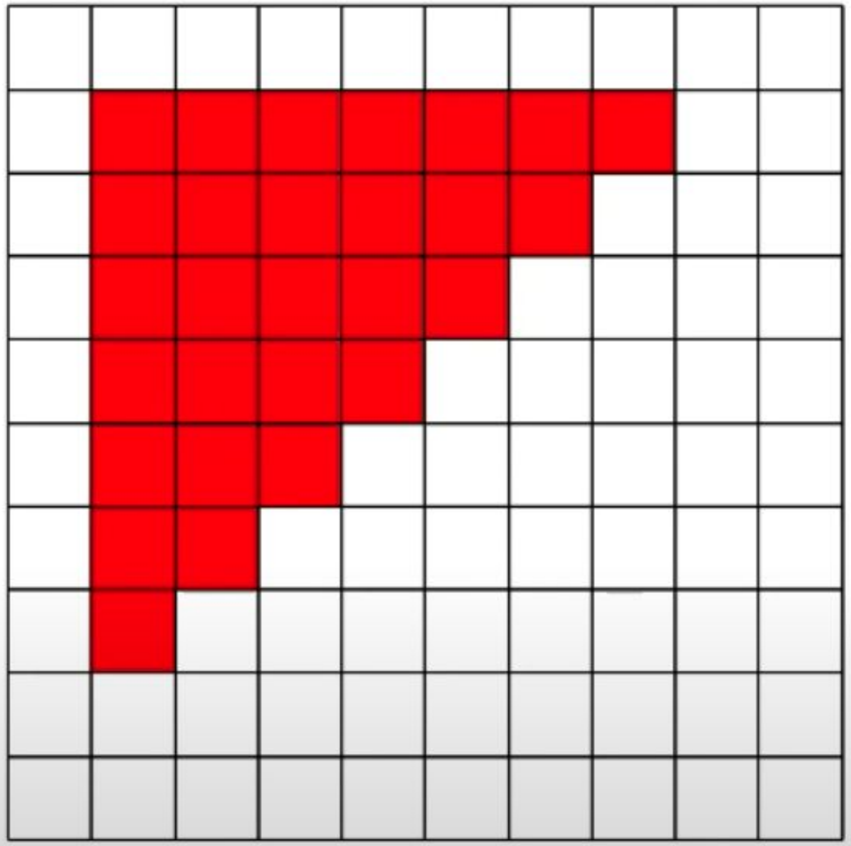
(Z-buffer)



(Z-buffer)

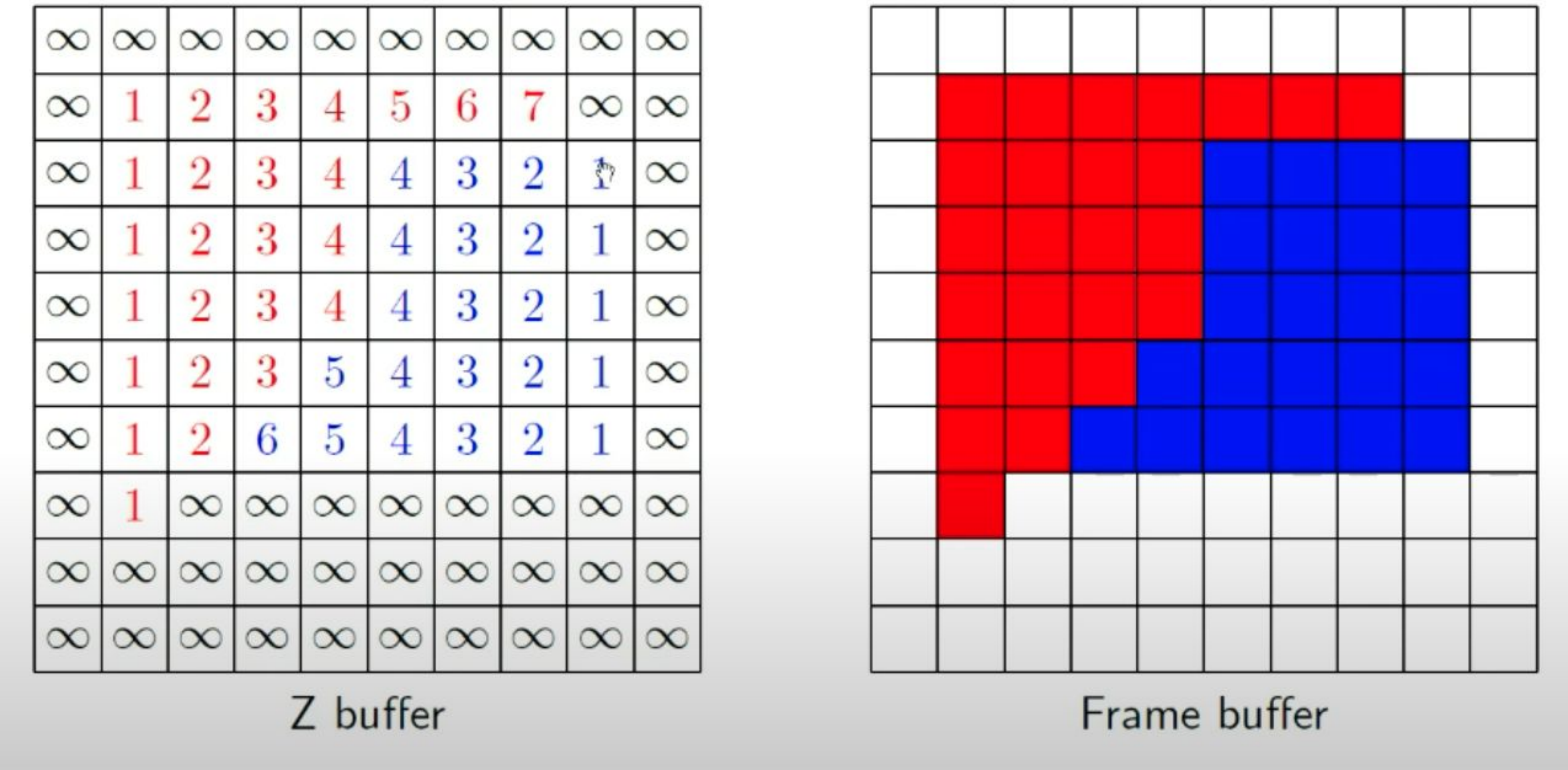
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	1	2	3	4	5	6	7	∞	∞
∞	1	2	3	4	5	6	∞	∞	∞
∞	1	2	3	4	5	∞	∞	∞	∞
∞	1	2	3	4	∞	∞	∞	∞	∞
∞	1	2	3	∞	∞	∞	∞	∞	∞
∞	1	2	∞	∞	∞	∞	∞	∞	∞
∞	1	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Z buffer



Frame buffer

(Z-buffer)



Créditos: Jon Shiach (https://www.youtube.com/watch?v=jDIsICV6axg)

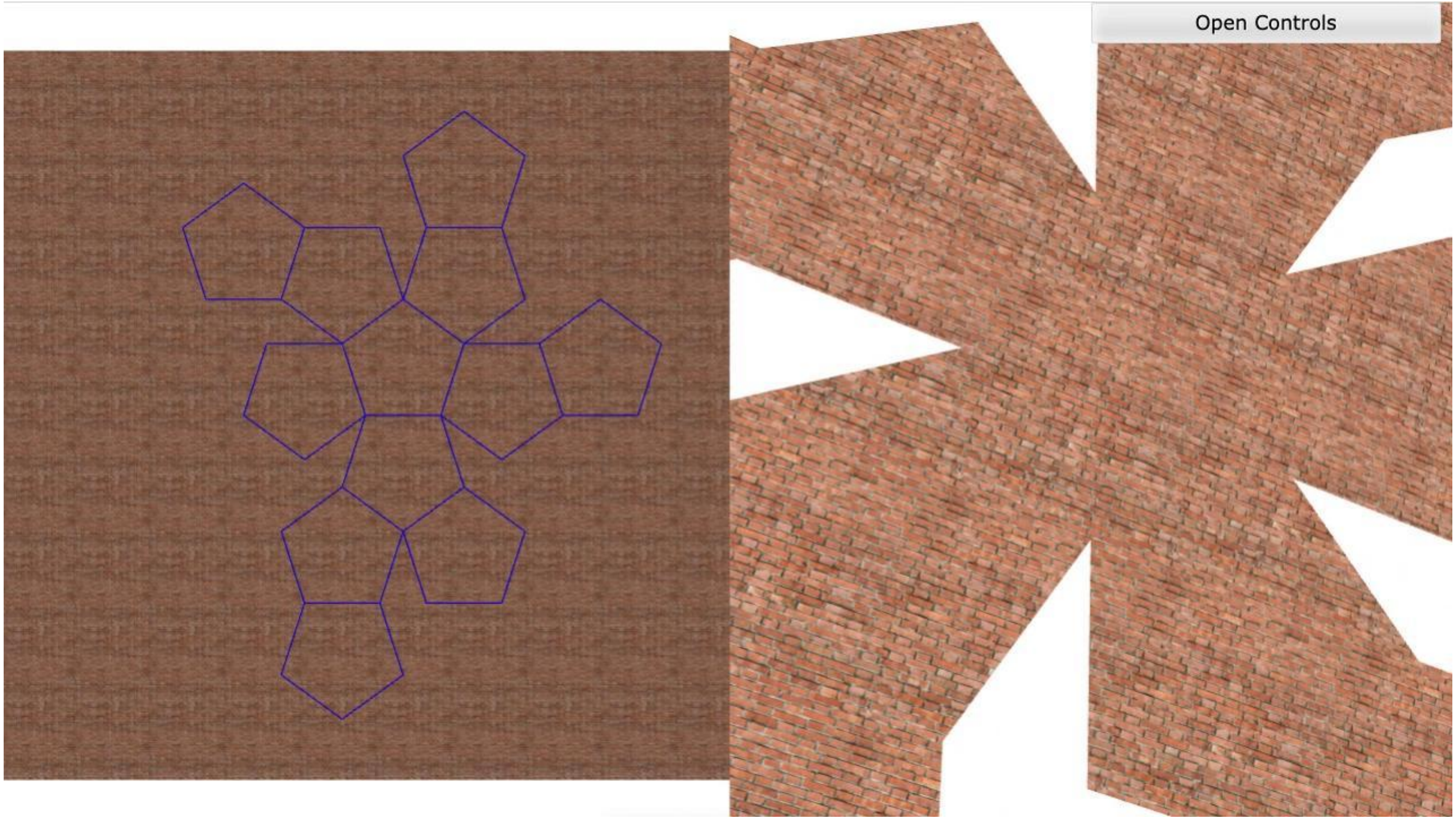
Ed termina o doutorado (1974)

(3) Mapeamento de texturas

MAPPING

Photographs, drawings, or any picture can be mapped onto bivariate patches. This is one of the most interesting consequences of the patch splitting algorithm. It gives a method for putting texture, drawings, or photographs onto surfaces. It also allows one to have reflections in pictures, as in flat or curved mirrors.

(Mapeamento de texturas)



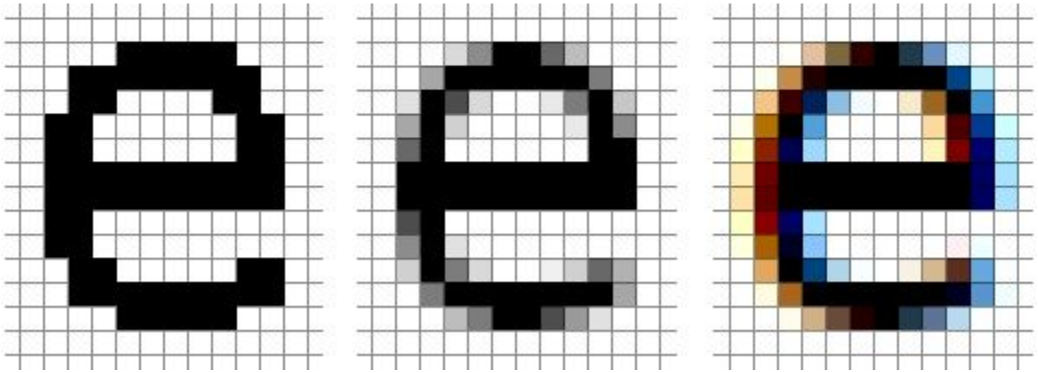
Ed termina o doutorado (1974)

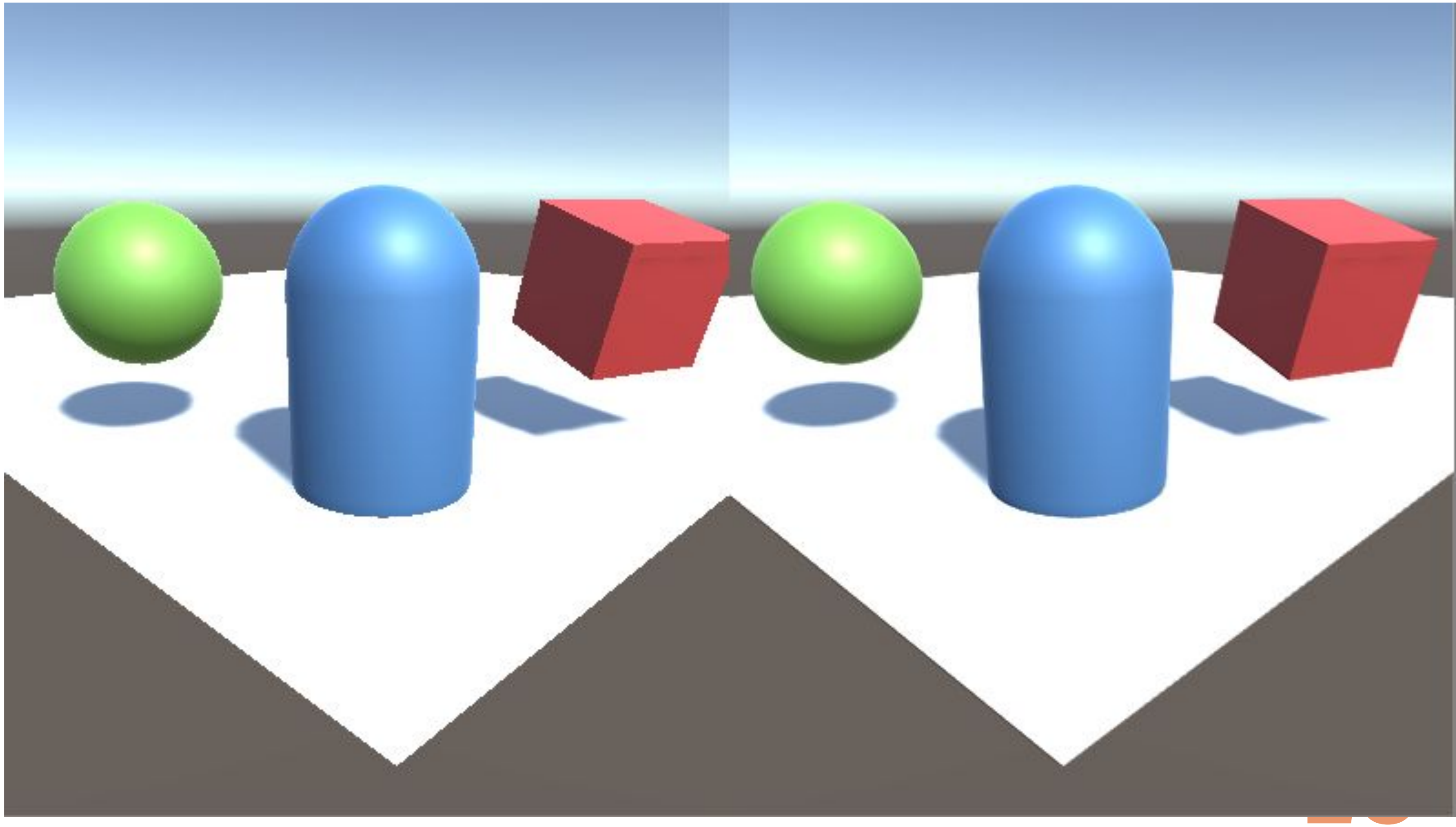
(4) Rasterização com *antialiasing*

ANTI-ALIASING FOR THE SUBDIVISION ALGORITHM

The subdivision algorithm can be modified to allow for area-sampling. Such a modification requires techniques for determining what is visible in each raster-element square and some method for storing and combining intensity values at each square to get the average. The modified algorithm has some drawbacks which will be discussed at the end of the chapter. Before presenting the modification, some groundwork needs to be laid and an "area-averaging algorithm" must be described.

(Antialising)





Ed começa a carreira

- NYIT (1974)
  - Trabalho com Alvy Ray Smith (Paint, HSV)
  - 1977 → sistema Tween (animação por quadros-chave)
- Lucasfilm (1979)
  - Vice-presidente da Industrial Light and Magic
    - Efeitos especiais para Star Wars
  - Objetivo: criar animação comercial 100% por computador
    - Problema: $$$

Ed e Pat se encontram

New York Institute of Technology (NYIT)

Pat ainda “apenas” engenheiro nuclear

- Doutorado em biofísica pela Univ. de Madison em 1985

Outros “pesos pesados” da CG

- Jim Blinn (Blinn-Phong shader model)
- James H. Clark
- Alvy Ray Smith (Paint3, HSV)

Pat Hanrahan - percurso inicial

- 1977 - B.S. em Engenharia Nuclear - Univ. Wisconsin-Madison
  - primeiro da turma
  - Departamento de Zoologia: "developed computer models of the motornervous system of the nematode Ascaris"
  - auto-didata em Computação
- 1980s NY Inst. of Technology Comp. Graphics Lab
- 1985 - PhD em Biophysics - Univ. Wisconsin-Madison
- 1986 - Pixar
- 1989 - Princeton University
- 1995 - Stanford University

Adventures of André and Wally B. (1984)

Pixar

- Steve Jobs compra a divisão digital da LucasFilm
- Com Ed e John Lasseter, Pixar é fundada

From “The Pixar Story” documentary (2007)

Pat se junta à Pixar (1986)

Responsável pela especificação da API do **RenderMan**

- Sistema de renderização foto-realística da PIXAR
- Versão não comercial disponível desde 2015
- Rendeu a Pat e ao time de desenvolvedores da Pixar 3 prêmios da academia (Technical achievement)

Especificação da Interface RenderMan (RISpec)

- Define elementos como câmeras, materiais e geometrias
- Usa a arquitetura de software *Reyes Rendering*
- Desenvolvida em colaboração com 19 companhias
- Ainda sem implementação quando apresentada no SIGGRAPH de 1993
- Finalmente implementada em 1995

Arquitetura Reyes

Robert L. Cook., Loren Carpenter, and Edwin Catmull. "The Reyes Image Rendering Architecture." *Computer Graphics (SIGGRAPH '87 Proceedings)*, pp. 95–102.

Figure 6. 1986 Pixar Christmas Card by John Lasseter and Eben Ostby.

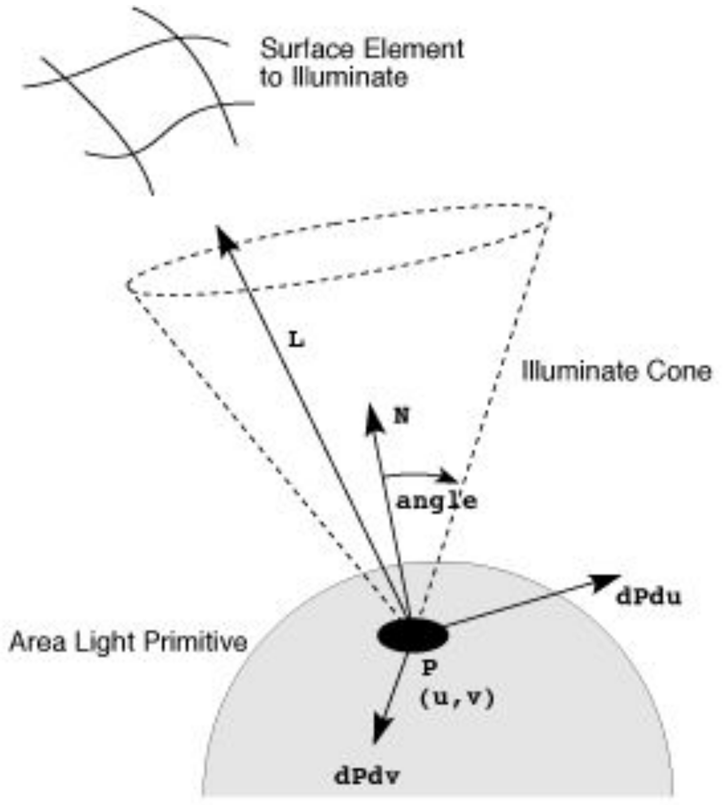
Arquitetura Reyes

- Flexibilidade
  - Modelagem → estruturas de dados diversas / modelagem procedural
  - Iluminação → múltiplos modelos / *shaders*
- Rapidez
  - Ray-tracing mínimo
  - Rendering baseado em subdivisão → *Antialising* e visibilidade
  - Paralelismo
- Qualidade
  - Superfícies curvas suaves, texturas
  - Efeitos como *motion blur* e profundidade de campo

Toy Story - 1995

Pat - visionário

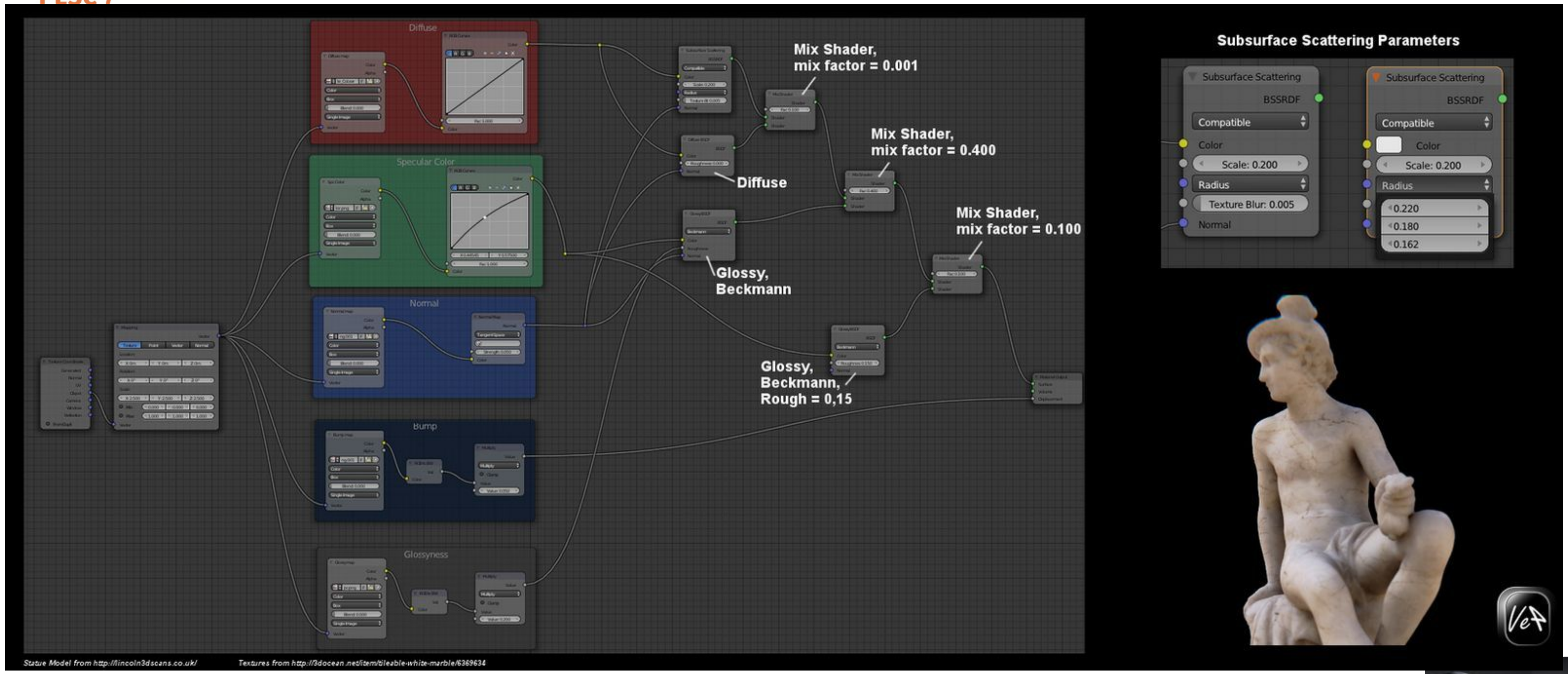
- Shaders
  - Conceito introduzido pela *RISpec*
  - Componente de algoritmos de renderização
    - Surface Shaders
    - Light Source Shaders
    - Volume Shaders
    - Displacement Shaders
    - Imager Shaders
- descrição dos materiais dos modelos e da iluminação



Light Source Shader

(crédito: RISpec)

Shaders (atual) - criação artística (OSL)



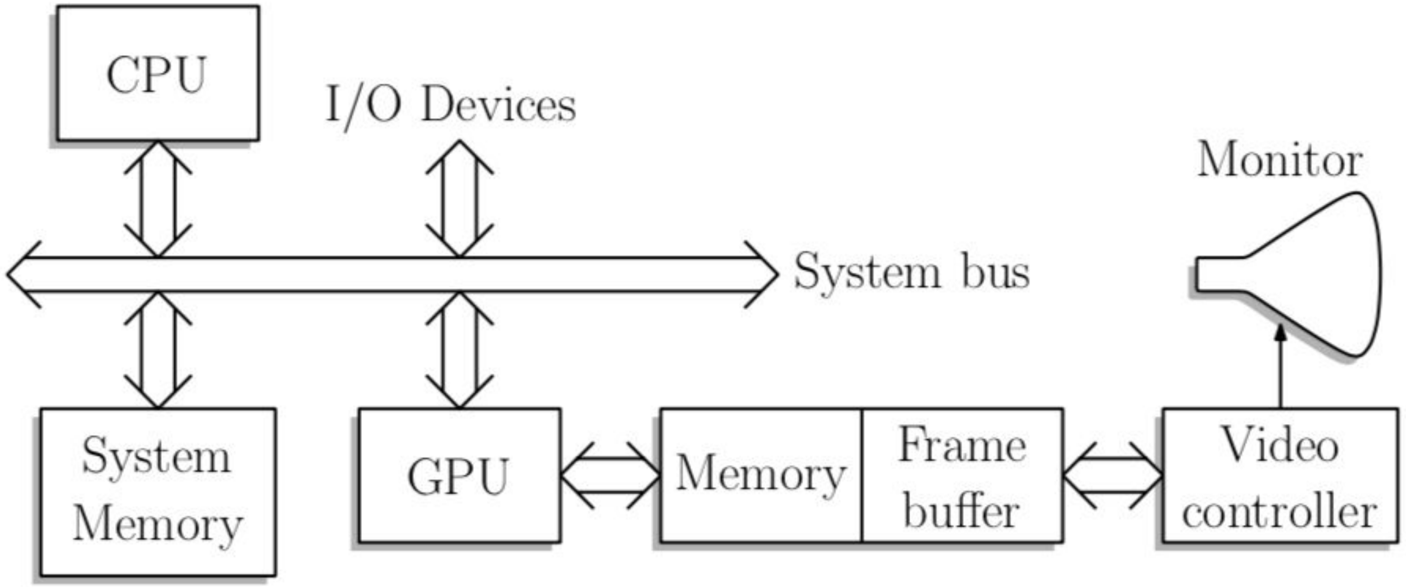
https://www.deviantart.com/4vector/art/SSS-Marble-Shader-Node
-Setup-for-Cycles-Render-426235456

https://www.blendernation.com/2015/09/29/150-free-cycles-shaders/

GPU - Graphical Processing Unit

“Circuito eletrônico especializado projetado para manipular e alterar rapidamente a memória e acelerar a criação de imagens em um frame buffer” (wikipedia)

- Componente de todas as placas gráficas desde os anos 70
- GeForce 256 (NVidia) → “Primeira GPU”

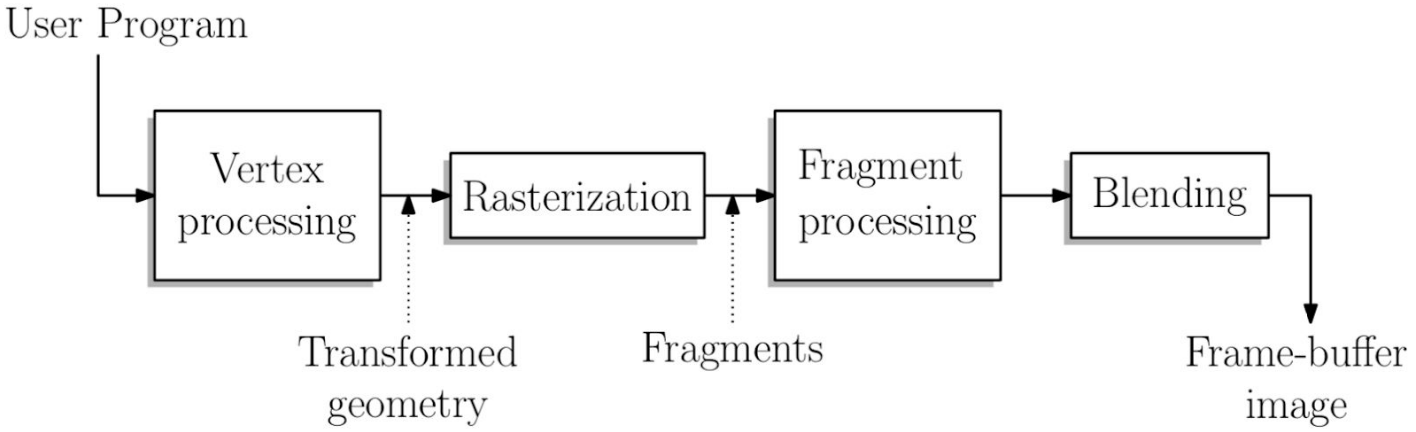


Programação de GPUs

- Funcionalidade fixa
  - Pipeline gráfico configurável através de variáveis de estado
- GPU programável (2001)
  - NVidia GeForce 3 ⇒ Vertex Shader
  - ATI Radeon 8500 ⇒ Fragment Shader

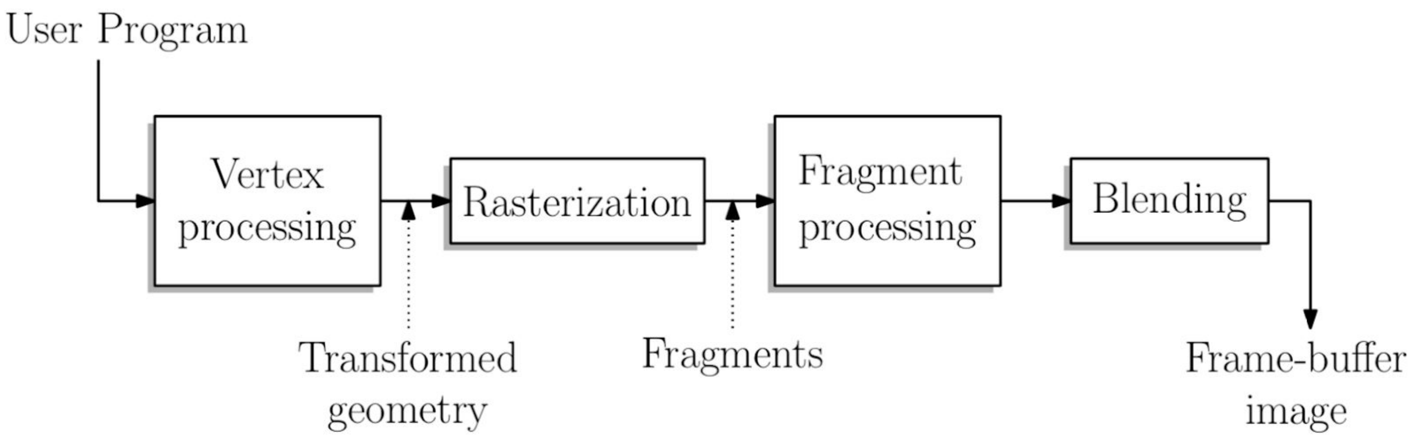
GPU Shaders

- flexibilizar a programação de GPUs
- linguagens Shaders de alto nível
  - 2002 - DirectX 9
  - 2004 - GLSL 1.1

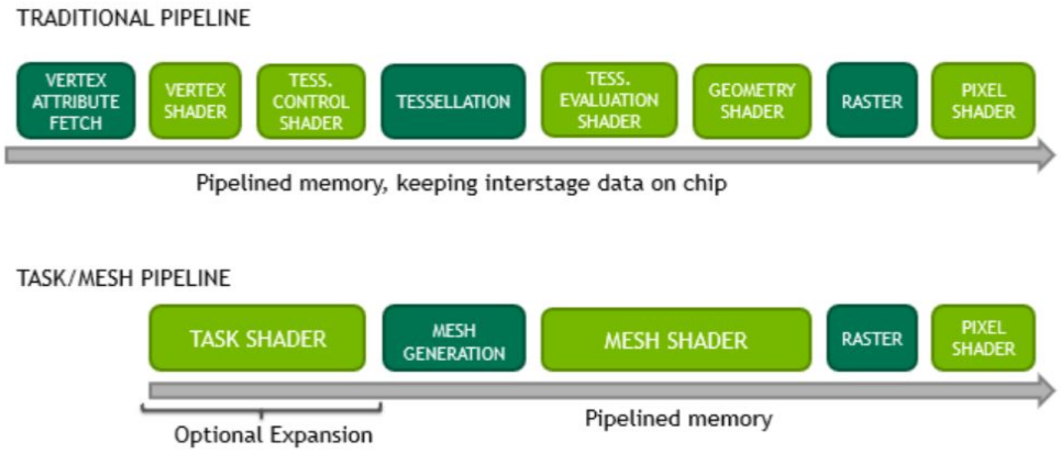


GPU Shaders

- flexibilizar a programação de GPUs
- linguagens Shaders de alto nível
  - 2002 - DirectX 9
  - 2004 - GLSL 1.1



2018 - Nvidia Turing (GLSL 4.6)



GP-GPU e *Brook*

- GP-GPU → General Purpose GPU programming
  - Usar o hardware gráfico para tarefas não gráficas
- Inicialmente, Shaders
  - Dados ⇔ pixels e vértices
  - tortura dos dados (e mental)
- Um modelo de programação mais apropriado
  - Stream programming

⇒ *Brook*

Pat - Visionário

BrookGPU

SIGGRAPH 2004

Brook for GPUs: Stream Computing on Graphics Hardware

Ian Buck Tim Foley Daniel Horn Jeremy Sugerman Kayvon Fatahalian Mike Houston Pat Hanrahan

Stanford University

Abstract

In this paper, we present Brook for GPUs, a system for general-purpose computation on programmable graphics hardware. Brook extends C to include simple data-parallel constructs, enabling the use of the GPU as a streaming coprocessor. We present a compiler and runtime system that abstracts and virtualizes many aspects of graphics hardware. In addition, we present an analysis of the effectiveness of the GPU as a compute engine compared to the CPU, to determine when the GPU can outperform the CPU for a particular algorithm. We evaluate our system with five applications,

modern hardware. In addition, the user is forced to express their algorithm in terms of graphics primitives, such as textures and triangles. As a result, general-purpose GPU computing is limited to only the most advanced graphics developers.

This paper presents *Brook*, a programming environment that provides developers with a view of the GPU as a streaming coprocessor. The main contributions of this paper are:

- The presentation of the Brook stream programming model for general-purpose GPU computing. Through the use of streams, kernels and reduction operators,

PEAKSTREAM

Pat - Visionário

BrookGPU

SIGGRAPH 2004

Brook for GPUs: Stream Computing on Graphics Hardware

Ian Buck Tim Foley Daniel Horn Jeremy Sugerman Kayvon Fatahalian Mike Houston Pat Hanrahan
Stanford University

Abstract

In this paper, we present Brook for GPUs, a system for general-purpose computation on programmable graphics hardware. Brook extends C to include simple data-parallel constructs, enabling the use of the GPU as a streaming coprocessor. We present a compiler and runtime system that abstracts and virtualizes many aspects of graphics hardware. In addition, we present an analysis of the effectiveness of the GPU as a compute engine compared to the CPU, to determine when the GPU can outperform the CPU for a particular algorithm. We evaluate our system with five applications,

modern hardware. In addition, the user is forced to express their algorithm in terms of graphics primitives, such as textures and triangles. As a result, general-purpose GPU computing is limited to only the most advanced graphics developers.

This paper presents *Brook*, a programming environment that provides developers with a view of the GPU as a streaming coprocessor. The main contributions of this paper are:

- The presentation of the Brook stream programming model for general-purpose GPU computing. Through the use of streams, kernels and reduction operators,

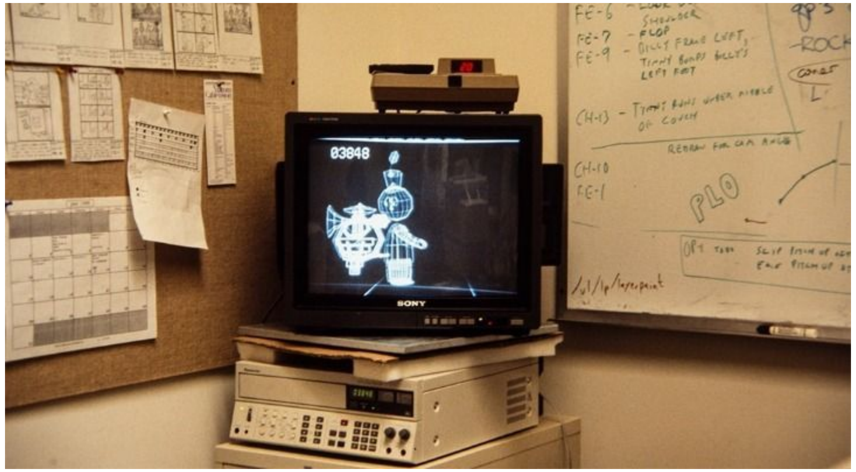
PEAKSTREAM

NVIDIA CUDA

Pat - Academy Awards

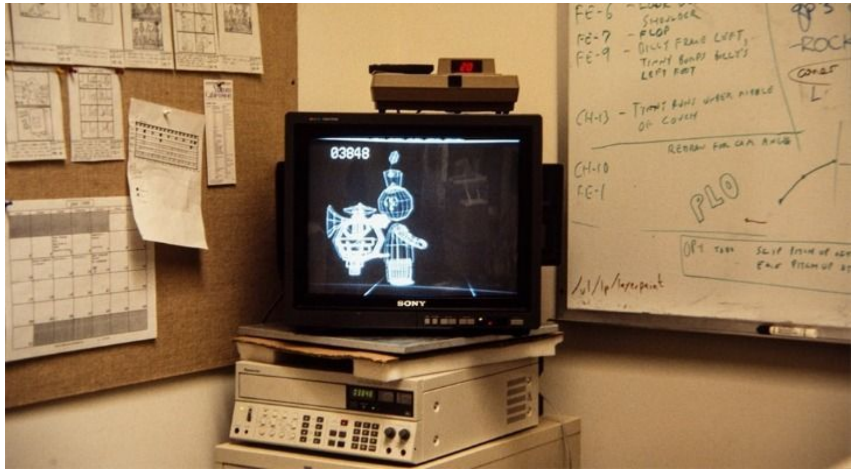
Pat - Academy Awards

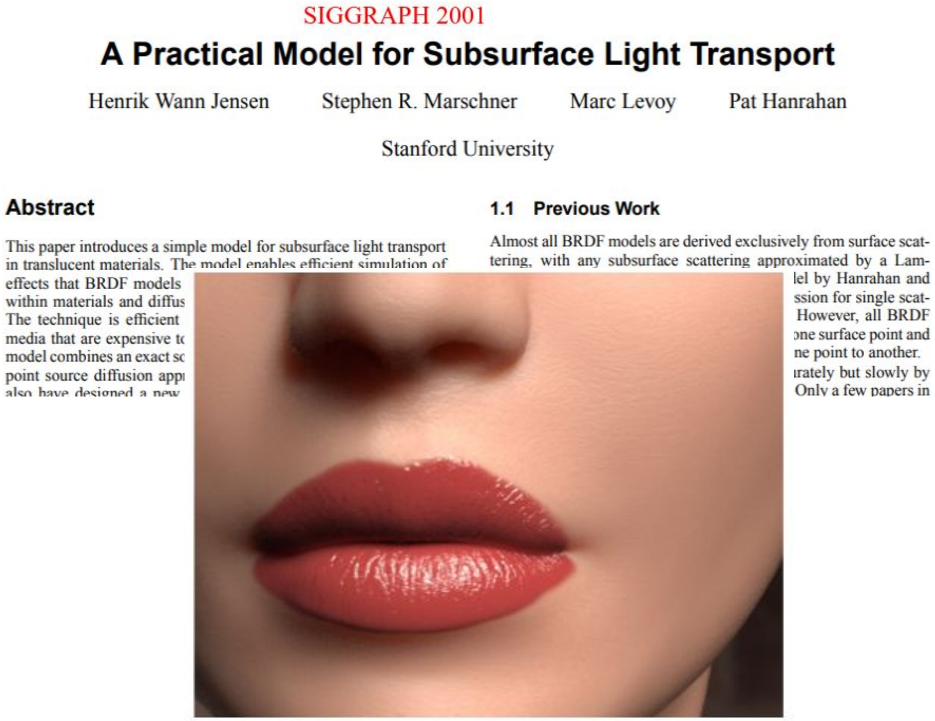
- 1993 - **scientific and engineering** - Renderman



Pat - Academy Awards


- 1993 - **scientific and engineering** - Renderman
- 2004 - **technical achievement** - "subsurface scattering"





SIGGRAPH 2001

A Practical Model for Subsurface Light Transport

Henrik Wann Jensen Stephen R. Marschner Marc Levoy Pat Hanrahan

Stanford University

Abstract

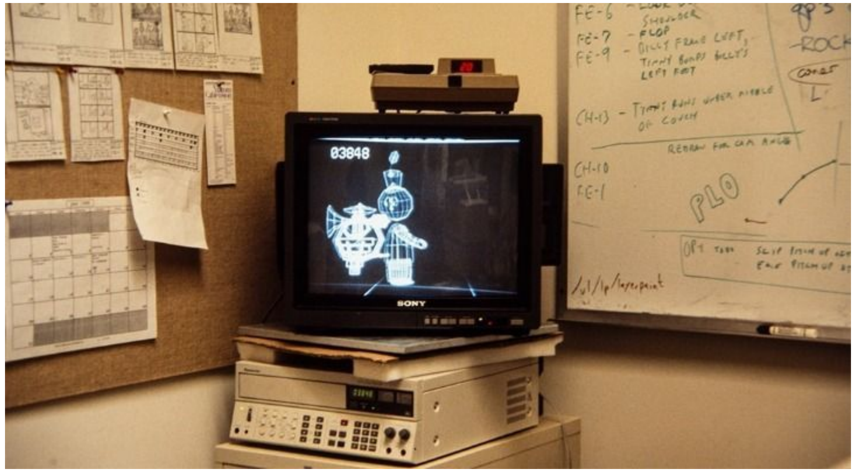
This paper introduces a simple model for subsurface light transport in translucent materials. The model enables efficient simulation of effects that BRDF models within materials and diffus The technique is efficient media that are expensive t model combines an exact sc point source diffusion appr also have designed a new

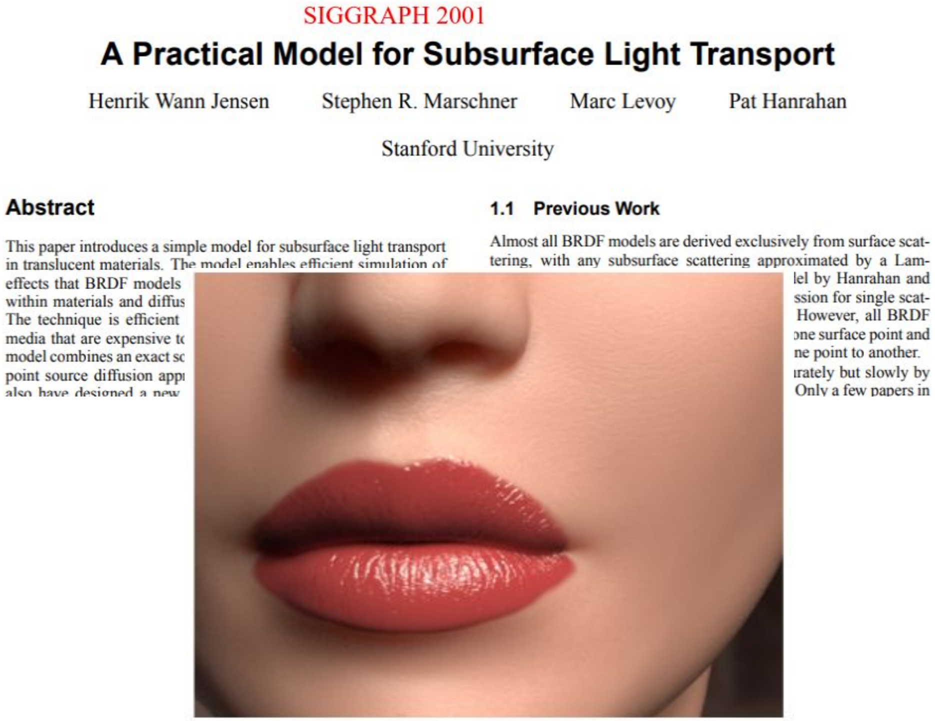
1.1 Previous Work

Almost all BRDF models are derived exclusively from surface scattering, with any subsurface scattering approximated by a Lam- lel by Hanrahan and ssion for single scat- However, all BRDF one surface point and ne point to another. rately but slowly by Only a few papers in

Pat - Academy Awards

- 1993 - **scientific and engineering** - Renderman
- 2004 - **technical achievement** - "subsurface scattering"
- 2013 - **technical achievement** - livro PBRT





SIGGRAPH 2001

A Practical Model for Subsurface Light Transport

Henrik Wann Jensen Stephen R. Marschner Marc Levoy Pat Hanrahan

Stanford University

Abstract

This paper introduces a simple model for subsurface light transport in translucent materials. The model enables efficient simulation of effects that BRDF models ... within materials and diffus... The technique is efficient ... media that are expensive t... model combines an exact sc... point source diffusion appr... also have designed a new

1.1 Previous Work

Almost all BRDF models are derived exclusively from surface scattering, with any subsurface scattering approximated by a Lam... ...lel by Hanrahan and ...ssion for single scat... However, all BRDF ...one surface point and ...ne point to another. ...rately but slowly by Only a few papers in

Pat - colaborações (5 papers Siggraph'93)

On the Form Factor between Two Polygons

Peter Schröder   Pat Hanrahan

Department of Computer Science
Princeton University

act

actors are used in radiosity to describe the fraction of dif-
reflected light leaving one surface and arriving at another.
re a fundamental geometric property used for computation.
special configurations admit closed form solutions. How-
le important case of the form factor between two polygons
lution. We give
ivex or concave,

nputer Graphics]:
J.2 [Physical Sci-

In thi
between
is quite
a detaile
result to

2 Cl

The forn
gral by

A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection

Larry Aupperle   Pat Hanrahan

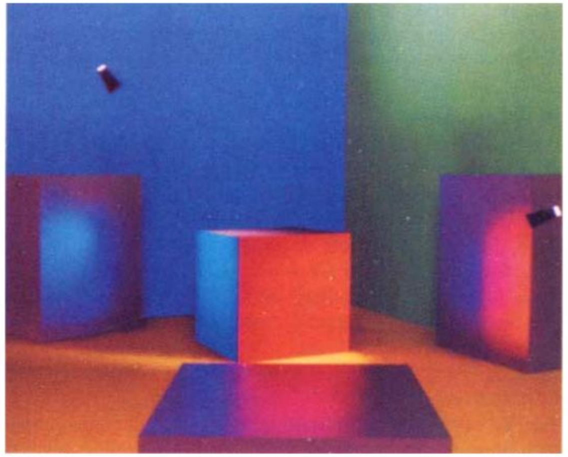
Department of Computer Science
Princeton University

Abstract

We develop a radiance formulation for discrete three point transport, and a new measure and description of reflectance: *area reflectance*. This formulation and associated reflectance allow an estimate of error in the computation of radiance across triples of surface elements, and lead directly to a hierarchical refinement algorithm for global illumination.

We have implemented and analyzed this algorithm over surfaces

There are many ways to pa
each leads to a different choi
theory community two techn
division (the method of disc
harmonics (P_N). These two
methods of Immel et al. and
ing variations are possible. O
based on Kajiya's original for
Under this formulation, the re



Global Visibility Algorithms for Illumination Computations

Seth Teller
Institute of Computer Science
Hebrew University of Jerusalem

Pat Hanrahan
Department of Computer Science
Princeton University

Abstract

The most expensive geometric operation in image synthesis is visibility determination. Classically this is solved with hidden surface removal algorithms that render only the parts of the scene visible from a point. Global illumination calculations, however, may require information between any two points in the scene. This paper describes global visibility algorithms that preprocess polygon databases in order to accelerate visibility determination during illumination calculations. These algorithms are sensitive to the output complexity in visibility space; that is, how many pairs of objects are mutually visible. Furthermore, the algorithms are

Visibility preprocessing. To compute what is visible from all points on the surfaces of the objects being shaded, we preprocess the scene to speed future visibility tests. For the purposes of global illumination we need only consider all pairwise interactions between objects. Preprocessing removes totally invisible pairs from consideration, and accelerates later queries regarding visibility between points on partially visible pairs.

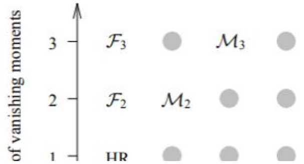
Incremental visibility maintenance. The most efficient global illumination algorithms operate iteratively based on error criteria. Examples are hierarchical radiosity, where surfaces are subdivided with respect to each other according to potential light transfers

Wavelet Radiosity

Steven J. Gortler   Peter Schröder   Michael F. Cohen   Pat Hanrahan

Department of Computer Science
Princeton University

be an effective means
in Lambertian diffuse
te the radiosity integral
osity function into a set
resulting in a set of n



Reflection from Layered Surfaces due to Subsurface Scattering

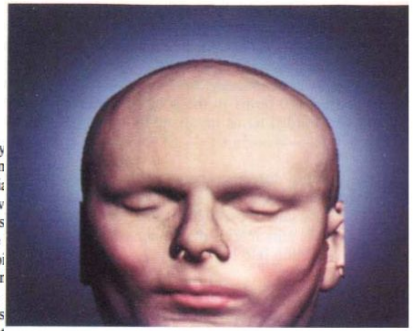
Pat Hanrahan
Department of Computer Science
Princeton University

Wolfgang Krueger
Department of Scientific Visualization
German National Research Center
for Computer Science

Abstract

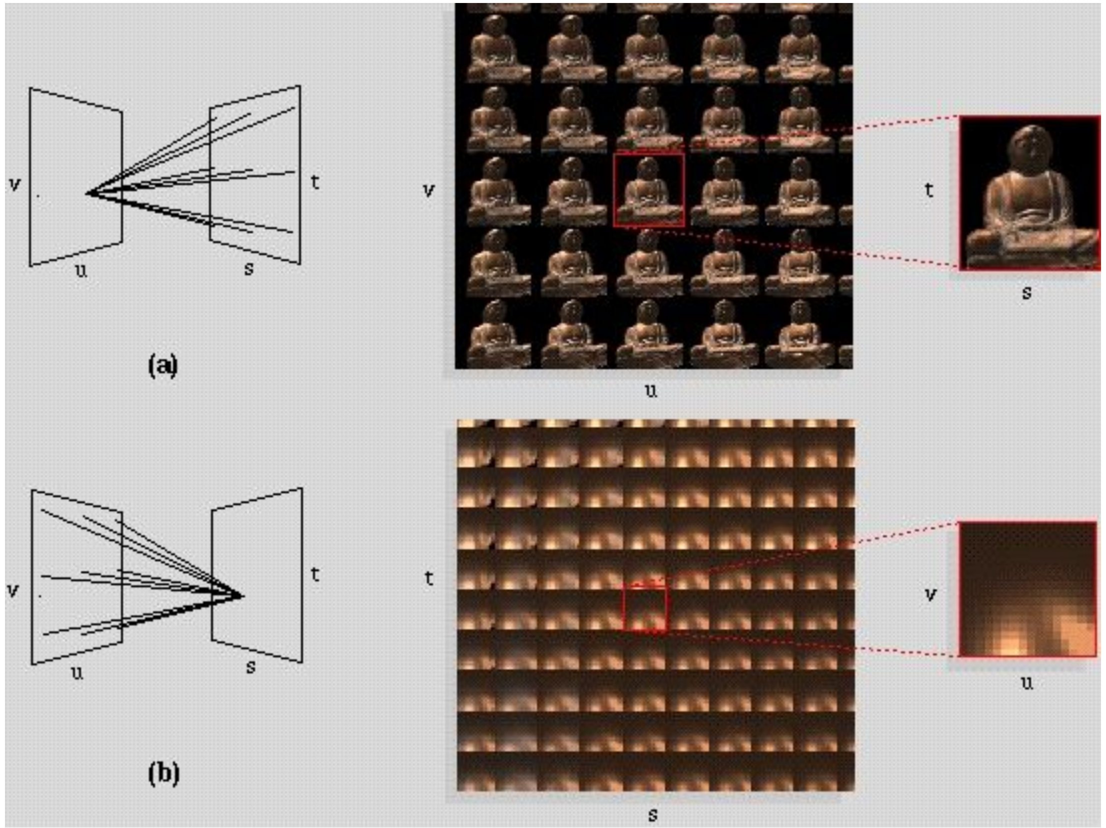
The reflection of light from most materials consists of two major terms: the specular and the diffuse. Specular reflection may be modeled from first principles by considering a rough surface consisting of perfect reflectors, or micro-facets. Diffuse reflection is generally considered to result from multiple scattering either from a rough surface or from within a layer near the surface. Accounting for diffuse reflection by Lambert's Cosine Law, as is universally done in computer graphics, is not a physical theory based on first principles.

of incidence. Diffuse reflection is qualitatively
subsurface scattering [18]: Light enters the n
and scattered, and eventually exits the materi
this subsurface interaction, light at different w
entially absorbed and scattered, and hence is
for the color of the material. Moreover, in the
is scattered multiple times, it becomes isotropi
rection in which it leaves the material is esse
qualitative explanation accounts for both the
ometric properties of diffuse materials. This
motivated by an early proof that there canno



Pat - Visionário

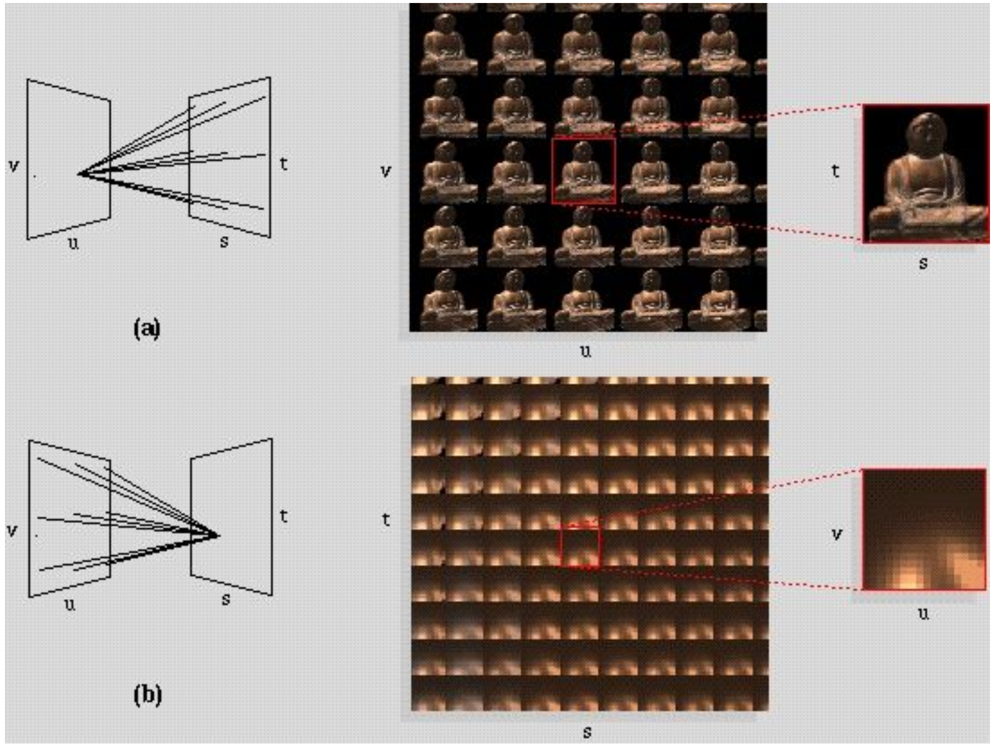
Light Field Rendering (1996) - 4098 citações



Engenharia de Sistemas e Computação PESC

Pat - Visionário

Light Field Rendering (1996) - 4098 citações



Immersive Light Field Video with a Layered Mesh Representation

MICHAEL BROXTON*, JOHN FLYNN*, RYAN OVERBECK*, DANIEL ERICKSON*, PETER HEDMAN, MATTHEW DUVALL, JASON DOURGARIAN, JAY BUSCH, MATT WHALEN, and PAUL DEBEVEC, Google

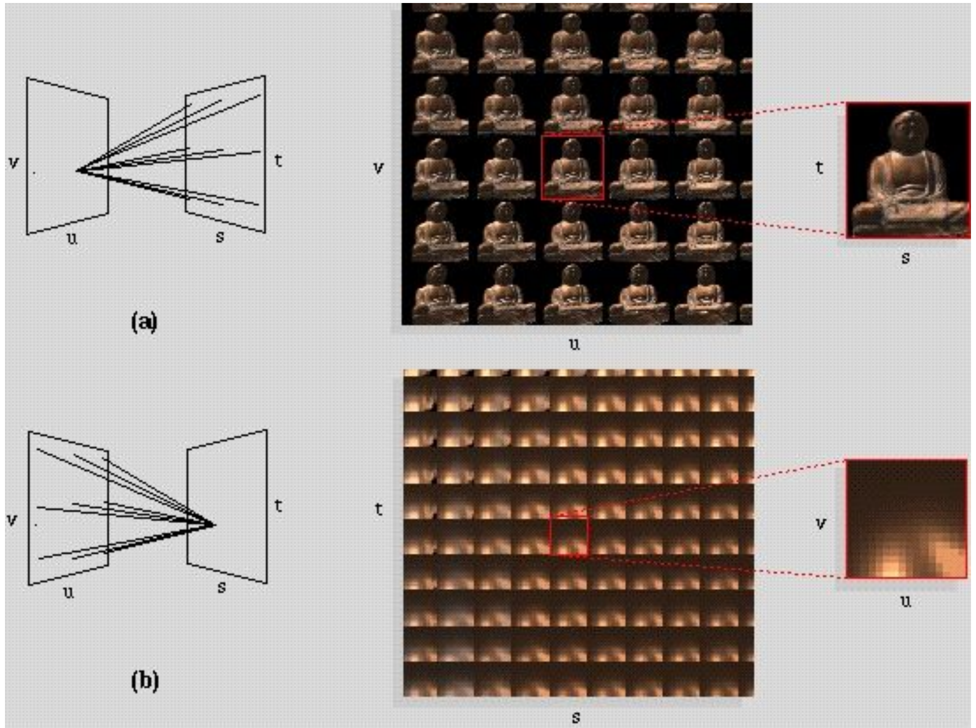
2 BACKGROUND

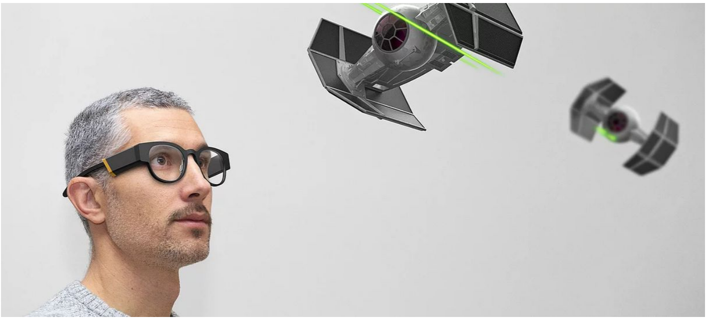
We build on significant bodies of work in multi-view stereo [Furukawa and Hernandez 2015], image based rendering [Shum et al. 2007], light field capture [Gortler et al. 1996; Levoy and Hanrahan 1996], and particularly recent advances in learning based view synthesis [Flynn et al. 2016]. Many of these efforts have focused

https://augmentedperception.github.io/deepviewvideo/

Pat - Visionário

Light Field Rendering (1996) - 4098 citações

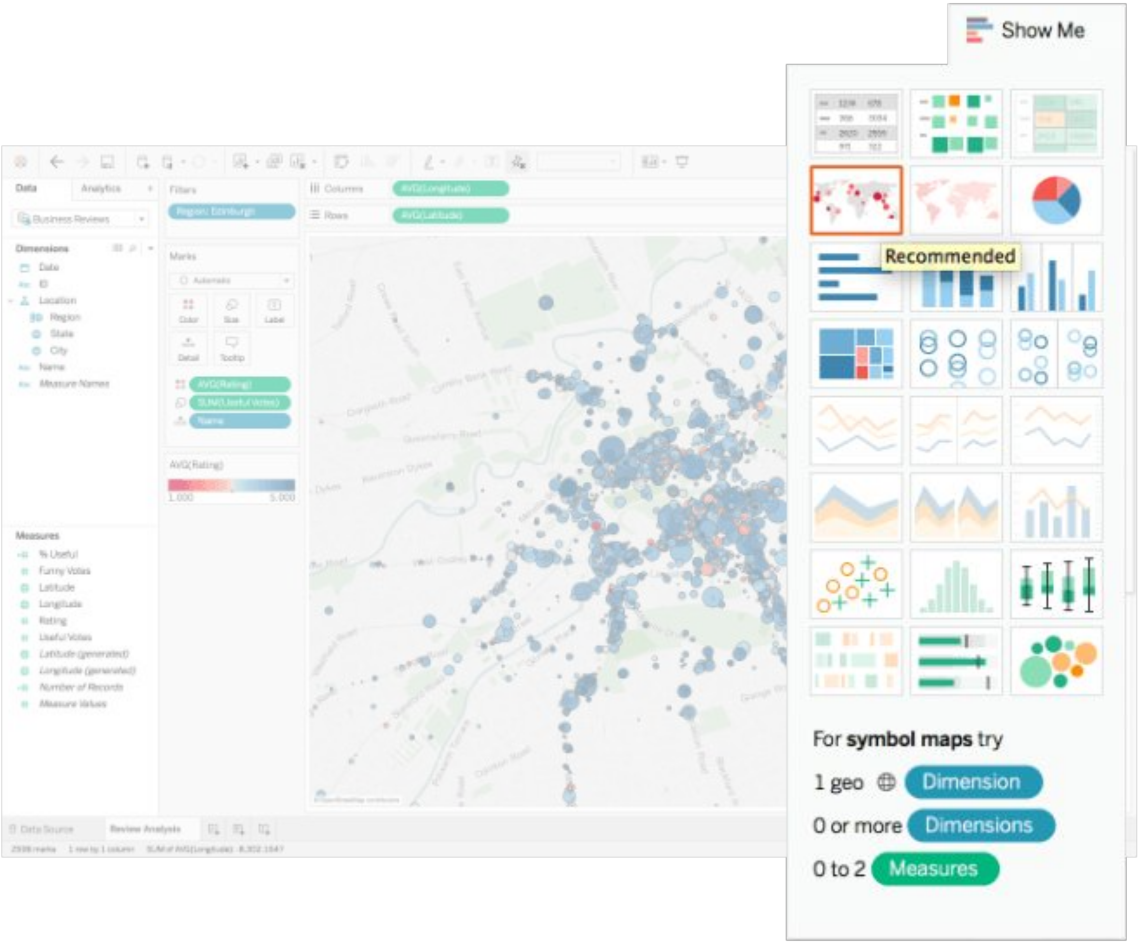




GOOGLEVR; AYE3D; MOPIC; CREAL; WINTUAL ...

Pat - Visionário

Tableau



Tamara Munzner

Visualization Career Award, 2006

Pat

em 2004

Sibgrapi 2004 - Pat invited speaker

About
News
Committees
Invited
Submission
Site
Program
Registration
Sponsors
Accomodation

Program

SIBGRAPI
XVII Brazilian Symposium on Computer Graphics and Image Processing
Curitiba-PR-Brazil / 17-20 October 2004
SIACG
II Ibero-American Symposium on Computer Graphics

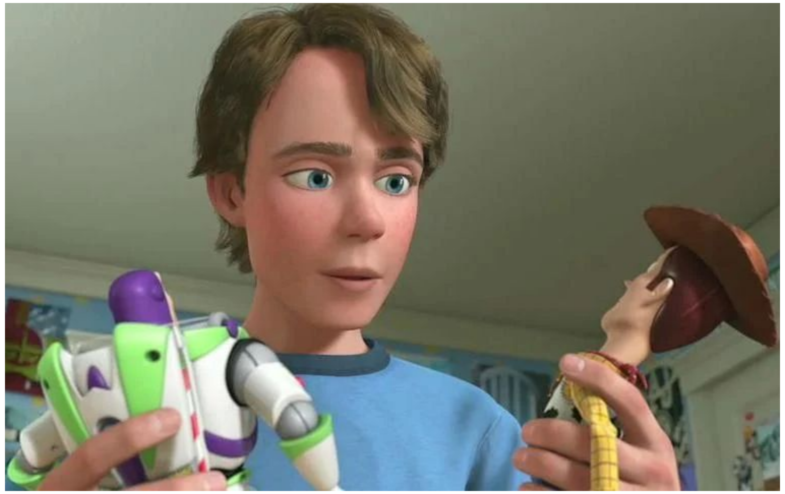
PROGRAM

- Invited Talks
- Tutorials
- Technical Sessions
- Poster Sessions
- Workshop of Theses and Dissertations (WTDCGPI)
- Workshop Of Undergraduate Work (WICCGIP)
- Lab Presentations
- Video Festival (Tuesday 19:00)
- Plenary Meeting (Tuesday 16:00-18:00)
- Welcome Cocktail (Monday 17:30-21:00)
- Banquet (Tuesday 20:30-24:00)

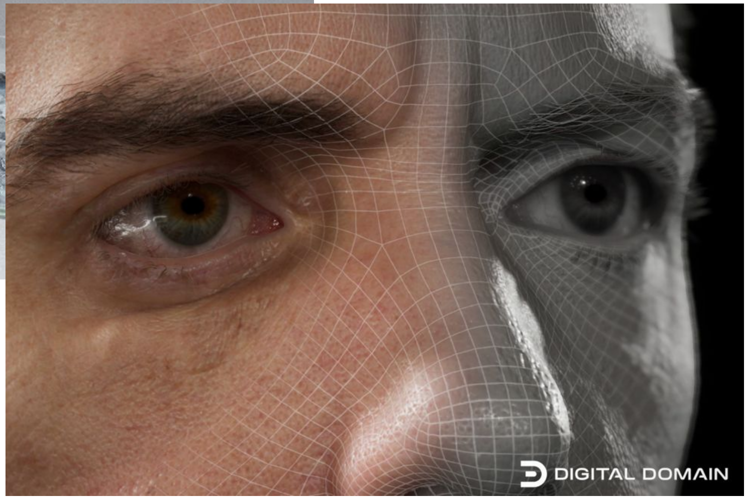
INVITED TALKS

Kim Boyer (Monday 11:00-12:00)
Patrick Hanrahan (Monday 16:00-17:00)
Xavier Pueyo (Tuesday 11:00-12:00)
Jorge Stolfi (Wednesday 11:00-12:00)

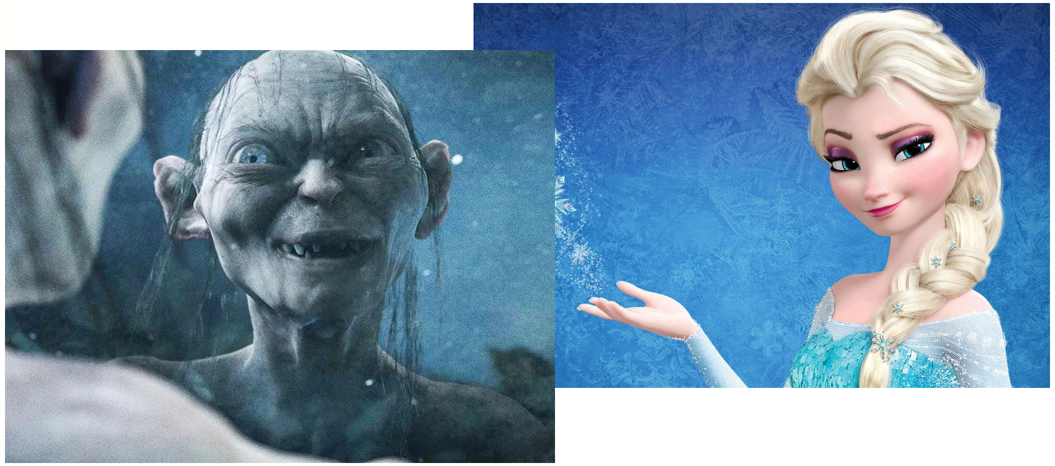
CG - últimos 30 anos

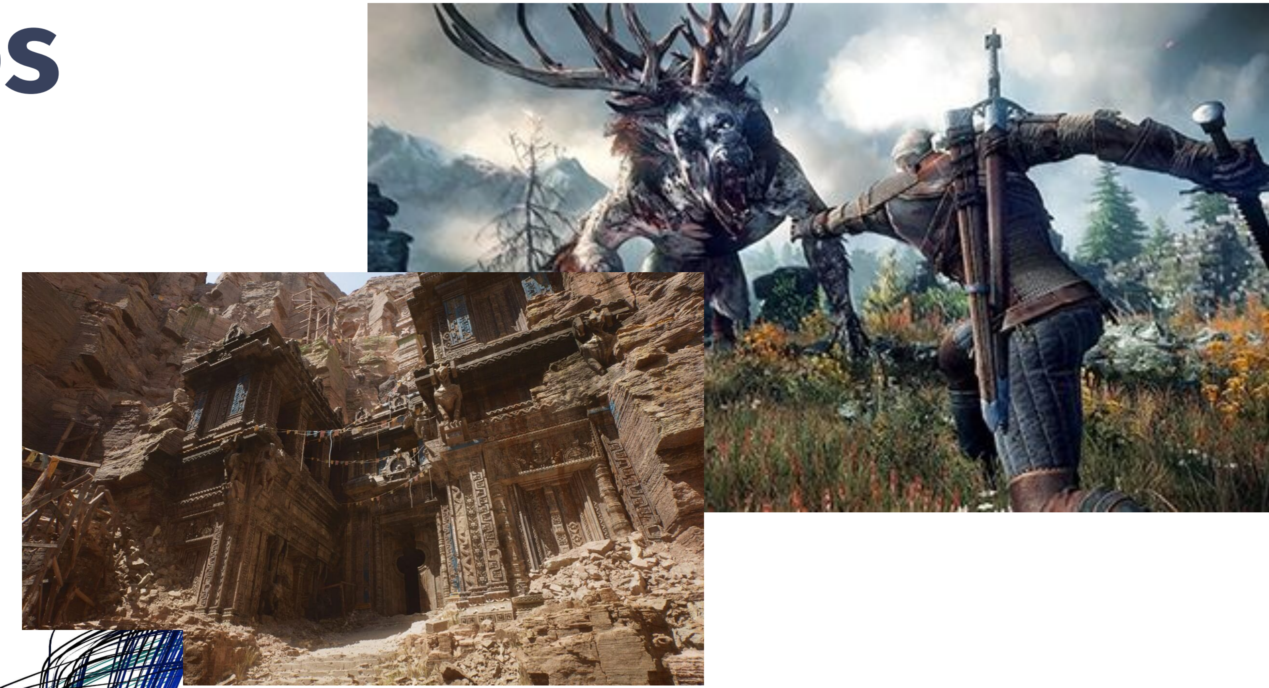


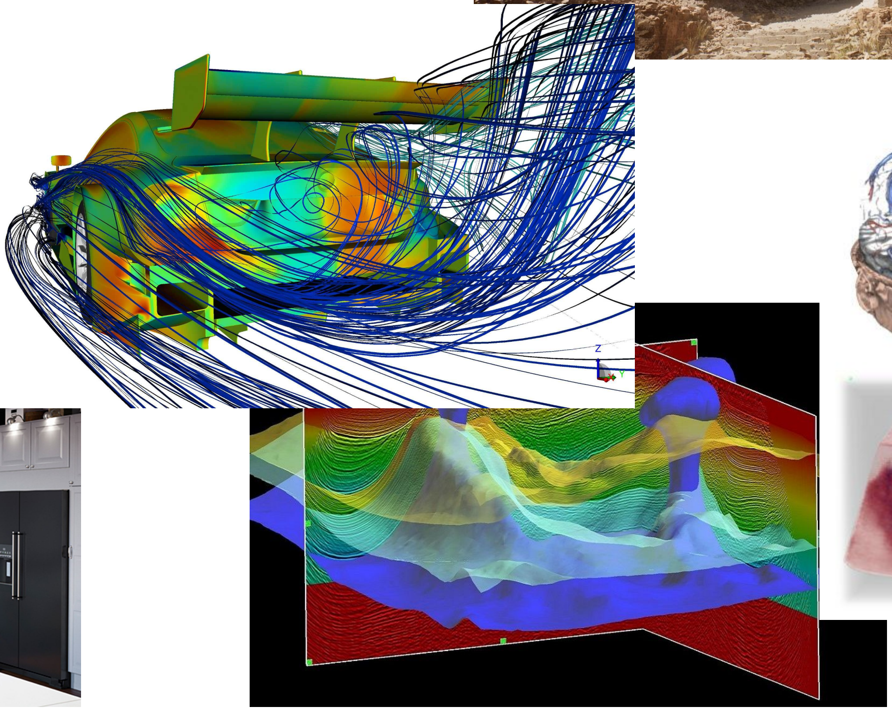
CG - últimos 30 anos

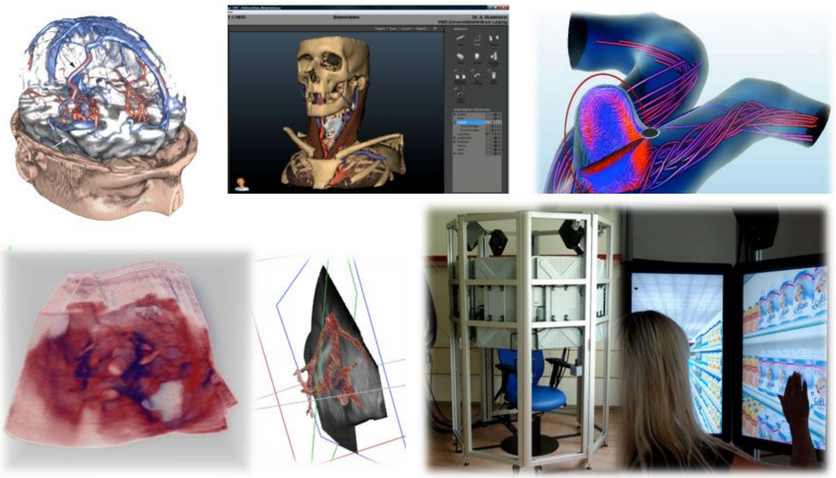


CG - últimos 30 anos

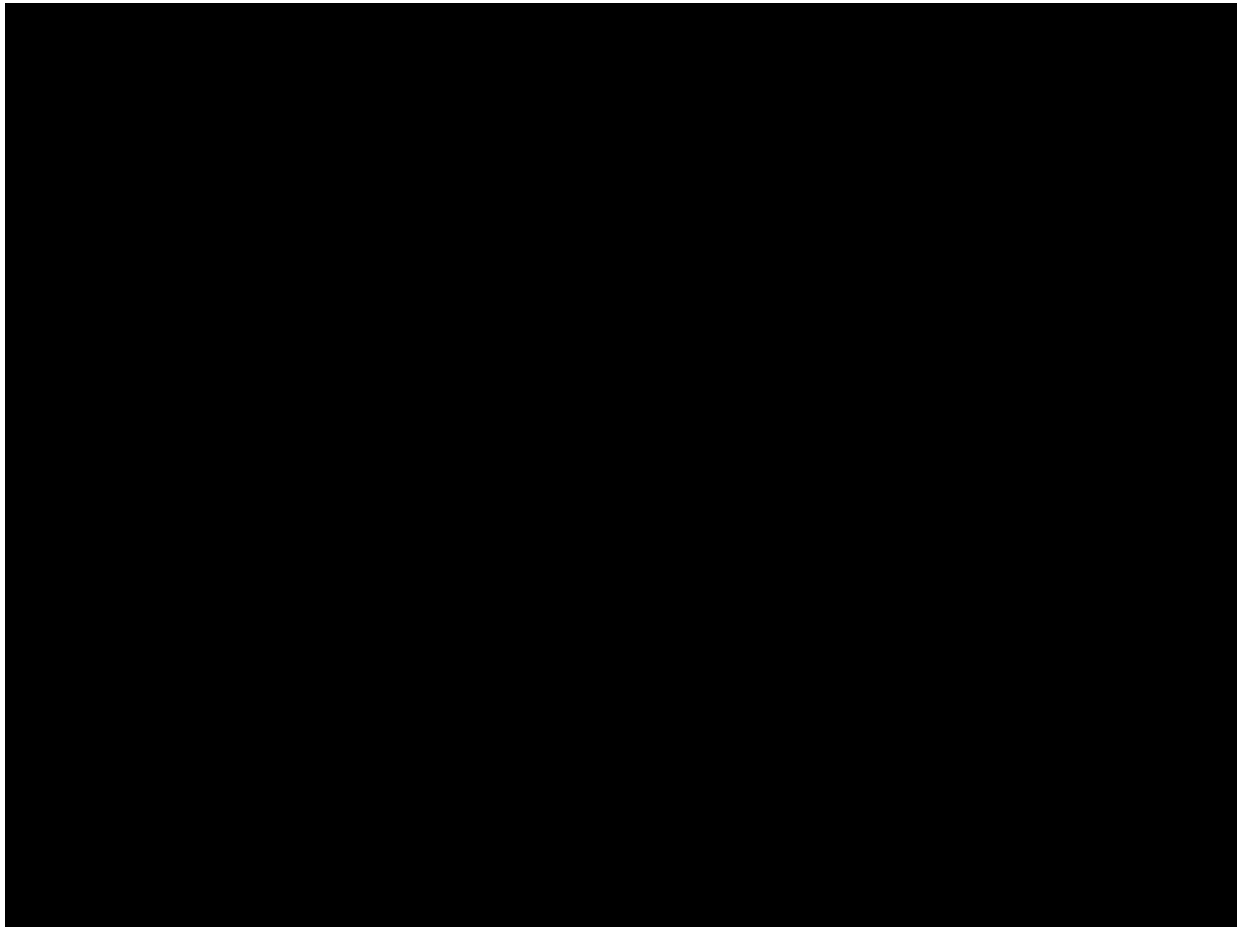








Prêmio Turing 2019
Pat Hanrahan & Ed Catmull



Engenharia de **Sistemas** e **Computação**
PESC

Prêmio Turing 2019
Pat Hanrahan & Ed Catmull