



Computer Graphics

Adam Finkelstein
Guest lecture in COS 111
Princeton University
Spring 2005

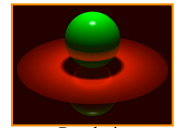


Overview

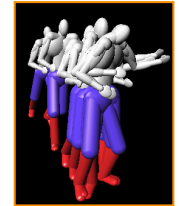
- I. Applications
- II. Images
- III. Rendering
- IV. Animation



Image Processing
(Rusty Coleman, CS426, Fall99)



Rendering
(Michael Bostock, CS426, Fall99)



Animation
(Jon Beyer, CS426, Spring04)



Part I: Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art



Applications

➔ Entertainment

- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art



Geri's Game
(Pixar Animation Studios)

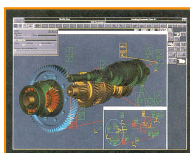


Quake
(Id Software)



Applications

- Entertainment
- ➔ **Computer-aided design**
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art



Gear Shaft Design
(Intergraph Corporation)



Los Angeles Airport
(Bill Japson, UCLA)

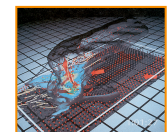


Boeing 777 Airplane
(Boeing Corporation)

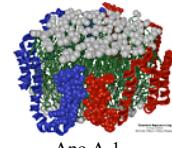


Applications

- Entertainment
- Computer-aided design
- ➔ **Scientific visualization**
- Training
- Education
- E-commerce
- Computer art



Airflow Inside a Thunderstorm
(Bob Wilhelmson, University of Illinois at Urbana-Champaign)



Apo A-1
(Theoretical Biophysics Group, University of Illinois at Urbana-Champaign)



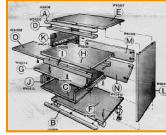
Visible Human
(National Library of Medicine)

Applications

- Entertainment
- Computer-aided design
- Scientific visualization

→ Training

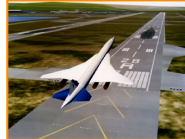
- Education
- E-commerce
- Computer art



Desk Assembly
(Silicon Graphics, Inc.)



Driving Simulation
(Evans & Sutherland)



Flight Simulation
(NASA)



Applications

- Entertainment
- Computer-aided design
- Scientific visualization

→ Training

- Education
- E-commerce
- Computer art



Forum of Trajan
(Bill Japson, UCLA)



Human Skeleton
(SGI)



Applications

- Entertainment
- Computer-aided design
- Scientific visualization

- Training
- Education

→ E-commerce

- Computer art



Interactive Kitchen Planner
(Matsushita)



Virtual Phone Store
(Lucent Technologies)

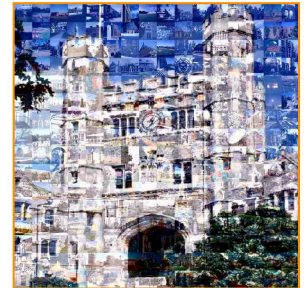


Applications

- Entertainment
- Computer-aided design
- Scientific visualization

- Training
- Education

→ Computer art



Blair Arch
(Marissa Range '98)



Part II: Images

- What is an image
- How to display images
- Photoshop operations
 - Blur / sharpen
 - Lighten / darken
 - Composition
- Morphing...



Image Composition
(Michael Bostock, CS426, Fall99)

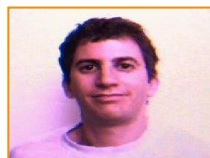


Image Morphing
(All students in CS 426, Fall98)



What is an Image?

- An image is a 2D rectilinear array of pixels



Continuous image



Digital image



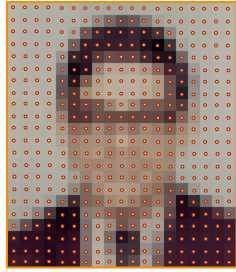
What is an Image?



- An image is a 2D rectilinear array of pixels



Continuous image



Digital image

A pixel is a sample, not a little square!

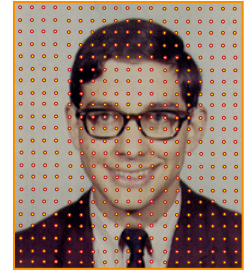
What is an Image?



- An image is a 2D rectilinear array of pixels



Continuous image



Digital image

A pixel is a sample, not a little square!

Image Display



- Re-create continuous function from samples
 - Example: cathode ray tube

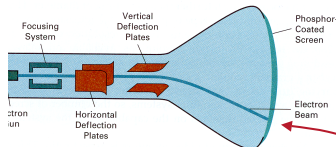


Image is reconstructed by displaying pixels with finite area (Gaussian)

Color CRT

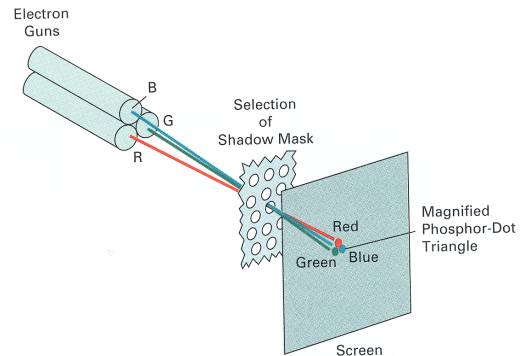
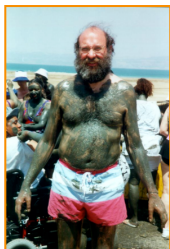


Figure 2.8 from H&B

Image Warping



- Move pixels of image
 - Mapping
 - Resampling



Source image

Warp

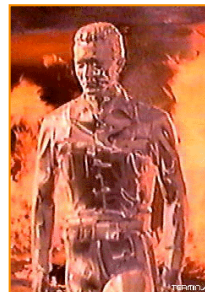


Destination image

Image Morphing



- Animate transition between two images



Terminator 2

Image Morphing

- Animate transition between two images

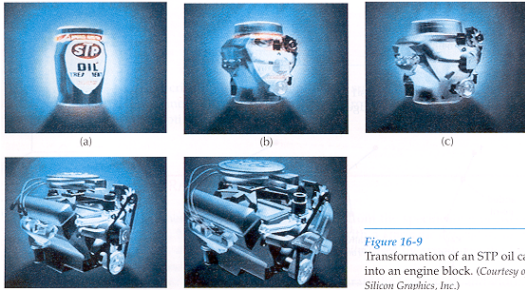


Figure 16-9
Transformation of an STP oil can
into an engine block. (Courtesy of
Silicon Graphics, Inc.)

H&B Figure 16.9

Cross-Dissolving

- Blend images with “over” operator
 - alpha of bottom image is 1.0
 - alpha of top image varies from 0.0 to 1.0

$$\text{blend}(i,j) = (1-t) \text{src}(i,j) + t \text{dst}(i,j) \quad (0 \leq t \leq 1)$$

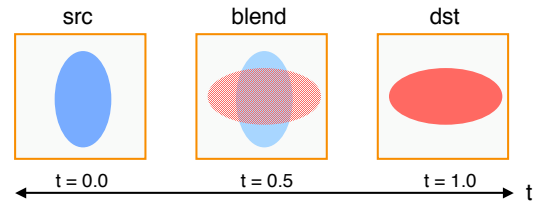


Image Morphing

- Combines warping and cross-dissolving

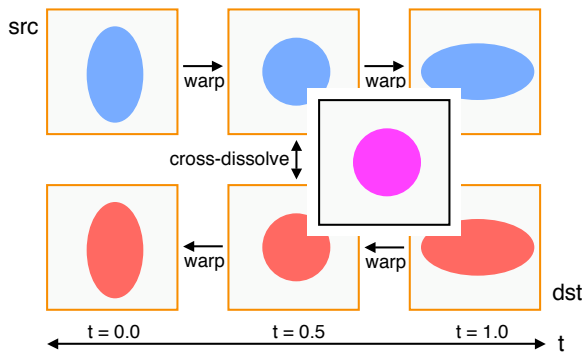
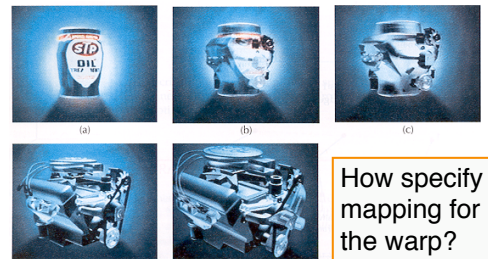


Image Morphing

- The warping step is the hard one
 - Aim to align features in images

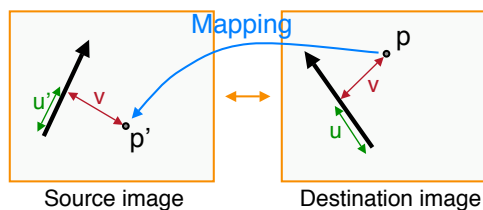


How specify
mapping for
the warp?

H&B Figure 16.9

Feature-Based Warping

- Beier & Neeley use pairs of lines to specify warp
 - Given p in dst image, where is p' in source image?

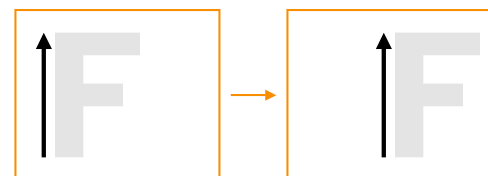


u is a fraction
 v is a length (in pixels)

Beier & Neeley
SIGGRAPH 92

Warping with One Line Pair

- What happens to the “F”?

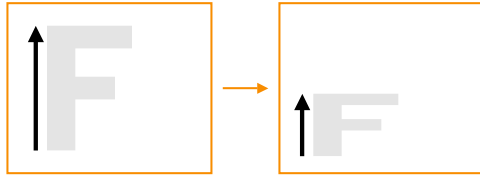


Translation!

Warping with One Line Pair



- What happens to the "F"?

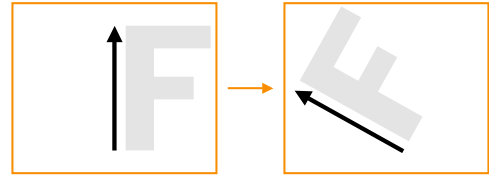


Scale!

Warping with One Line Pair



- What happens to the "F"?

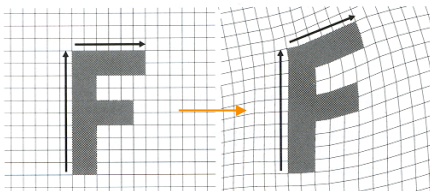


Rotation!

Warping with Multiple Line Pairs



- Use weighted combination of points defined by each pair of corresponding lines

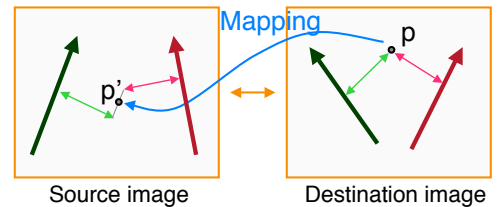


Beier & Neeley, Figure 4

Warping with Multiple Line Pairs

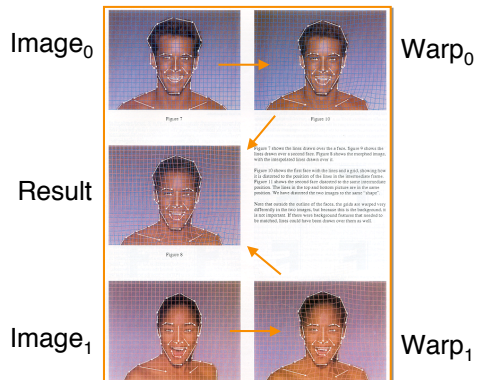


- Use weighted combination of points defined by each pair of corresponding lines

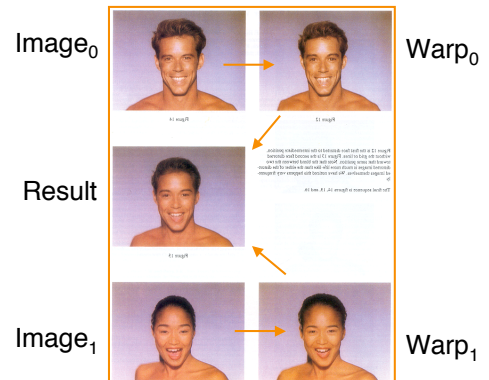


p' is a weighted average

Beier & Neeley Example



Beier & Neeley Example



Part III: Rendering



- Direct illumination
 - One bounce from light to eye
 - Implemented in graphics cards
 - OpenGL, DirectX, ...



OpenGL
(Chi Zhang, CS 426, Fall99)

- Global illumination
 - Many bounces
 - Ray tracing

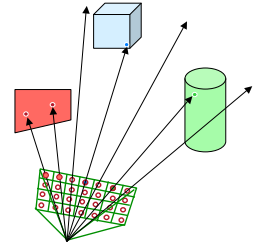


Ray Tracing
(James Percy, CS 426, Fall99)

Ray Casting



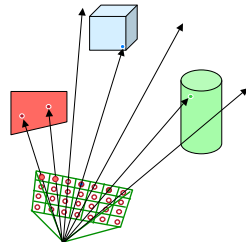
- For each sample ...
 - Construct ray from eye position through image plane
 - Find first surface intersected by ray through pixel
 - Compute color of sample based on surface properties



Ray Casting



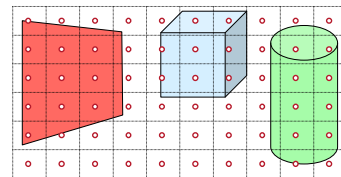
- For each sample ...
 - Construct ray from eye position through image plane
 - Find first surface intersected by ray through pixel
 - Compute color of sample based on surface properties



Ray Casting



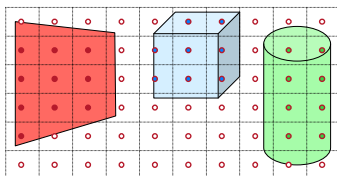
- For each sample ...
 - Construct ray from eye position through image plane
 - Find first surface intersected by ray through pixel
 - Compute color of sample based on surface properties



Ray Casting



- For each sample ...
 - Construct ray from eye position through view plane
 - Find first surface intersected by ray through pixel
 - Compute color of sample based on surface properties

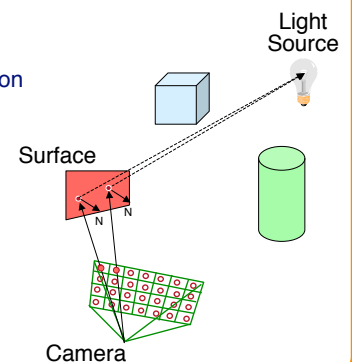


More efficient algorithms, such as those in graphics cards, utilize spatial coherence!

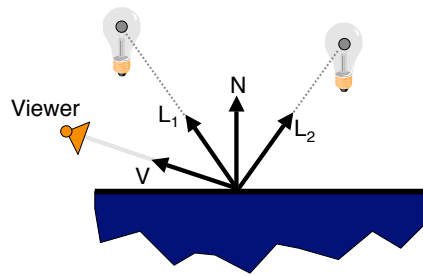
Lighting Simulation



- Lighting parameters
 - Light source emission
 - Surface reflectance
 - Atmospheric attenuation
 - Camera response



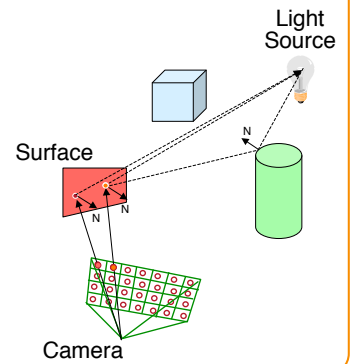
Lighting Simulation



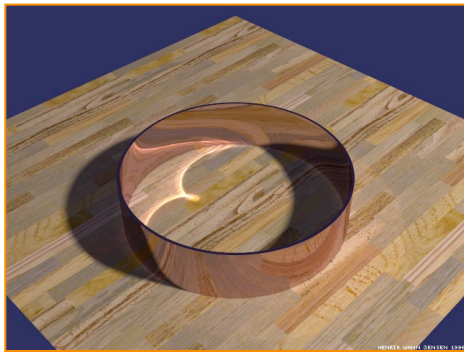
Lighting Simulation



- Direct illumination
 - Ray casting
- Global illumination
 - Ray tracing



Ray Tracing



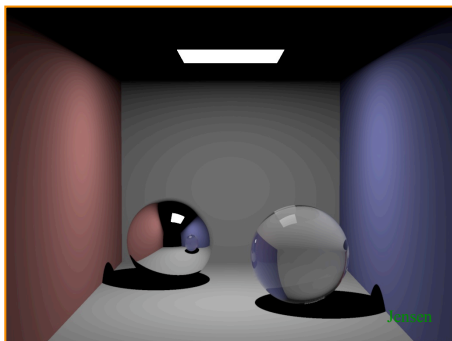
Henrik Wann Jensen

Ray Tracing



RenderPark

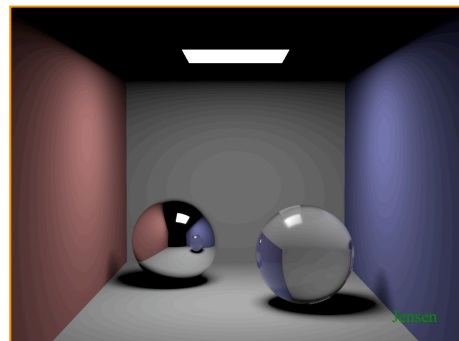
Ray Tracing



Basic ray tracing

Henrik Wann Jensen

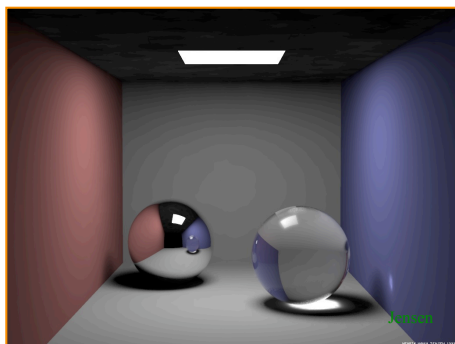
Ray Tracing



+ soft shadows

Henrik Wann Jensen

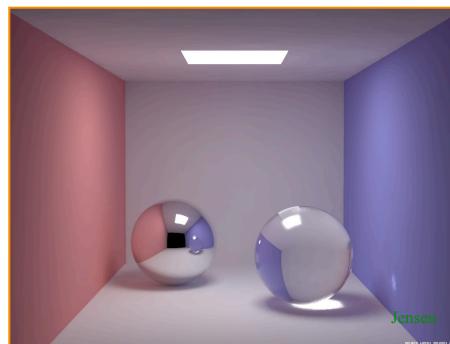
Ray Tracing



+ caustics

Henrik Wann Jensen

Ray Tracing



+ indirect diffuse illumination

Henrik Wann Jensen

Ray Tracing

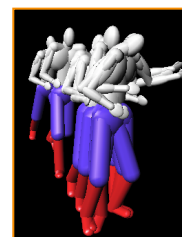


Terminator 2

Part IV: Animation



- Keyframe animation
 - Articulated figures



Animation
(Jon Beyer,
CS426, Spring04)

- Simulation
 - Particle systems



Simulation

Keyframe Animation: Luxo Jr.

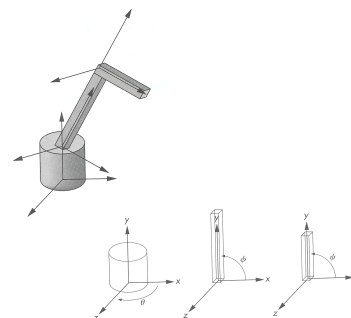
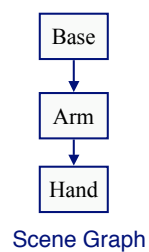


Pixar

Articulated Figures



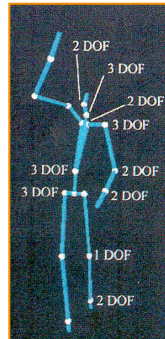
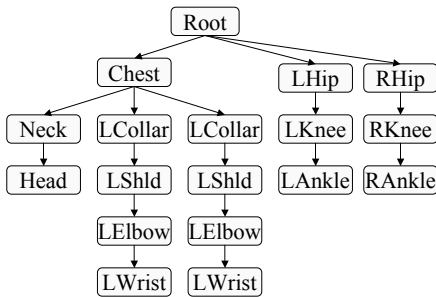
- Character poses described by set of rigid bodies connected by "joints"



Angel Figures 8.8 & 8.9

Articulated Figures

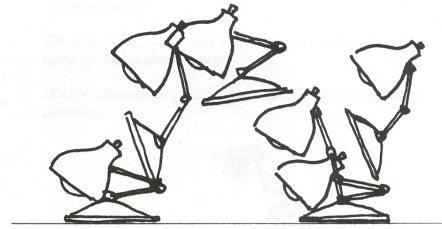
- Well-suited for humanoid characters



Rose et al. '96

Keyframe Animation

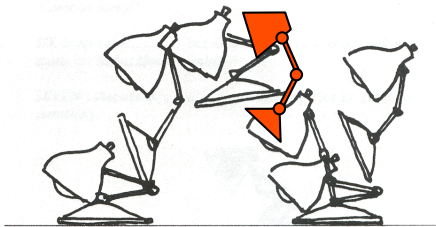
- Define character poses at specific time steps called "keyframes"



Lasseter '87

Keyframe Animation

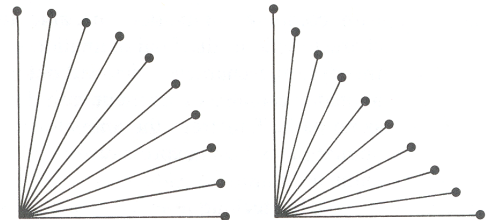
- Interpolate variables describing keyframes to determine poses for character in between



Lasseter '87

Articulated Figures

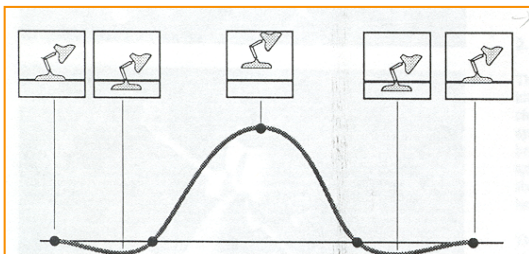
- Inbetweening
 - Compute joint angles between keyframes



Watt & Watt

Keyframe Animation

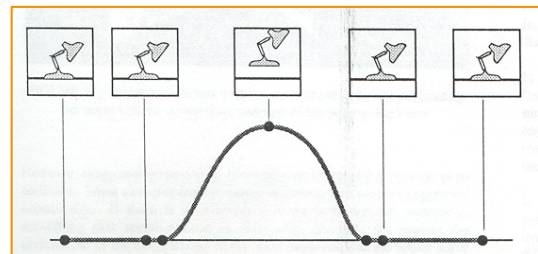
- Inbetweening:
 - Cubic spline interpolation - maybe good enough
 - » May not follow physical laws



Lasseter '87

Keyframe Animation

- Inbetweening:
 - Cubic spline interpolation - maybe good enough
 - » May not follow physical laws

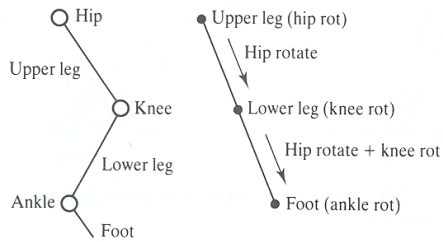


Lasseter '87

Example: Walk Cycle



- Articulated figure:

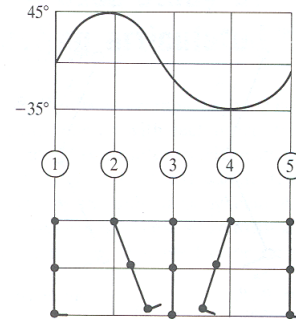


Watt & Watt

Example: Walk Cycle



- Hip joint orientation:

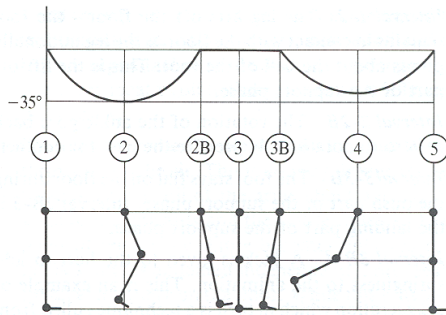


Watt & Watt

Example: Walk Cycle



- Knee joint orientation:

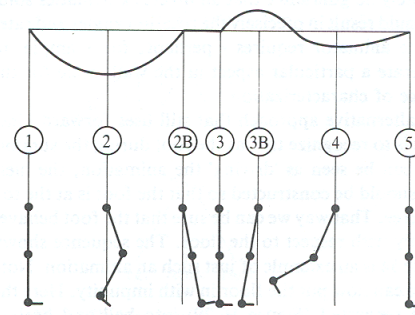


Watt & Watt

Example: Walk Cycle



- Ankle joint orientation:

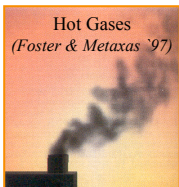


Watt & Watt

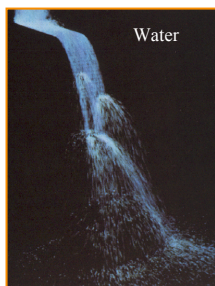
Simulation



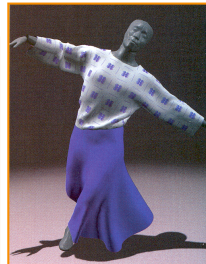
- Animator would not want to specify motion for:
 - Smoke
 - Water
 - Cloth
 - Hair
 - Fire



Hot Gases
(Foster & Metaxas '97)



Water

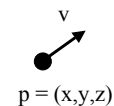


Cloth
(Baraff & Witkin '98)

Particle Systems



- A particle is a point mass
 - Mass
 - Position
 - Velocity
 - Acceleration
 - Color
 - Lifetime
- Use lots of particles to model complex phenomena
 - Keep array of particles



Particle Systems



- For each frame:
 - Create new particles and assign attributes
 - Delete any expired particles
 - Update particles based on attributes and physics
Newton's Law: $f=ma$
 - Render particles

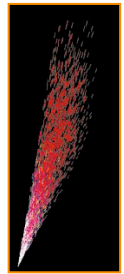


Creating/Deleting Particles



- Where to create particles?
 - Around some center
 - Along some path
 - Surface of shape
 - Where particle density is low
- When to delete particles?
 - Where particle density is high
 - Life span
 - Random

This is where person controls animation



Example: Wrath of Khan

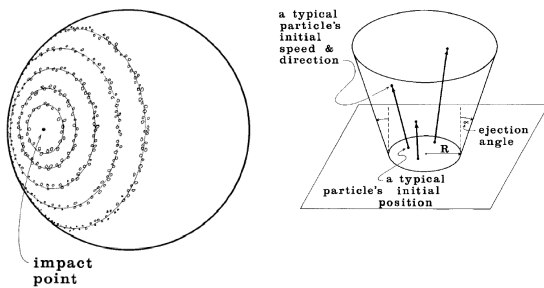
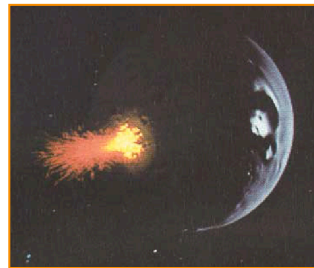


Fig. 2. Distribution of particle systems on the planet's surface.

Reeves

Example: Wrath of Khan



Reeves

Example: Wrath of Khan



Fig. 7. Wall of fire about to engulf camera.

Reeves

Summary



- Applications
- Images
- Rendering
- Animation

Underlying much of computer graphics are the fun parts from math.