Computer Graphics & Geometry

고려대학교 컴퓨터 그래픽스 연구실

Contents

CGVR

- What is Computer Graphics?
  - Definition
  - Main themes
  - Applications
- Geometry in 3D Computer Graphics
  - Modeling
  - Rendering
  - Animation

Inspirational Video

CGVR

- Entirely Computer-Generated

"Geri's Game." (c) Pixar 1997

Exercise: Video Analysis

CGVR

- What is Modeled or Simulated to Produce this Video?

Exercise: Video Analysis

CGVR

- What is Modeled or Simulated to Produce this Video?
  - Representing of objects
  - Simulate light
  - Shadows
  - Textures
  - Camera focus (depth of field)
  - Opacity
  - Refraction
  - Animation
  - Physics
  - Music
  - Story ideas

What is Computer Graphics?

CGVR

- Producing Pictures or Images using a Computer
- Algorithms for Visual Simulations

Main Themes

CGVR

- Imaging
  - Representing 2D images
- Modeling
  - Representing 3D objects
- Rendering
  - Constructing 2D images from 3D models
- Animation
  - Simulating changes over time

Imaging

CGVR

- Warping
- Metamorphosis
- Non-Photorealistic Rendering

Image Warping

CGVR

- Move pixels of image
  - Mapping
  - Resampling

Source Image

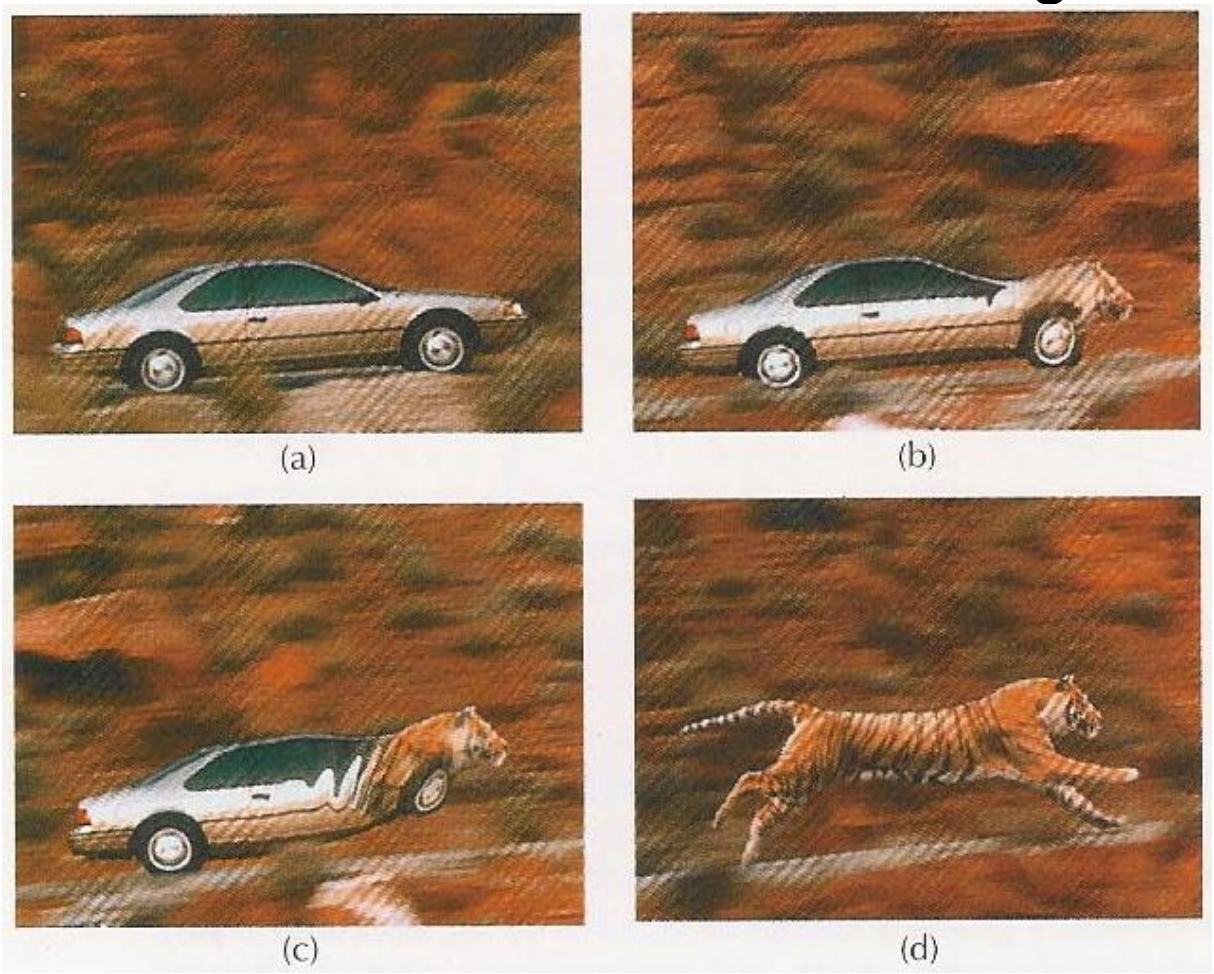
Destination Image

Courtesy of Princeton Univ.

Image Morphing

CGVR

- Animate transition between two images



Non-Photorealistic Rendering

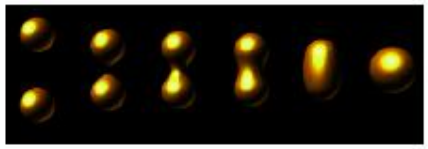
CGVR

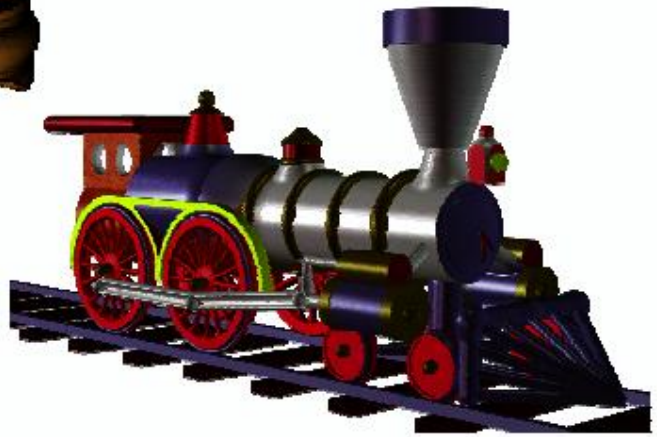
Modeling

CGVR

- Shape Description & Design



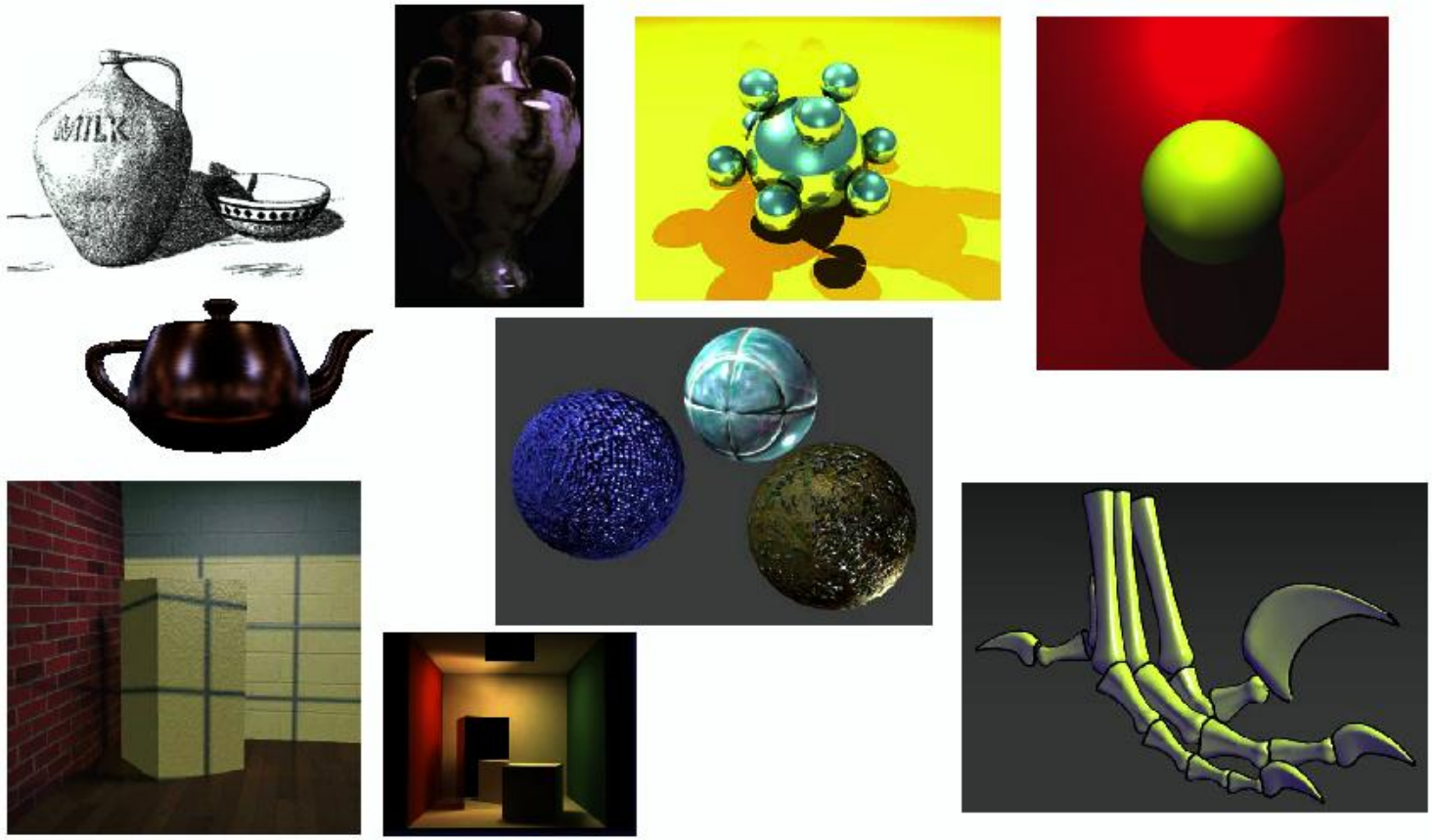






Rendering

CGVR

- Simulating Behavior of Lights & Image Formation



Animation

CGVR

- Motion Representation & Control

Applications

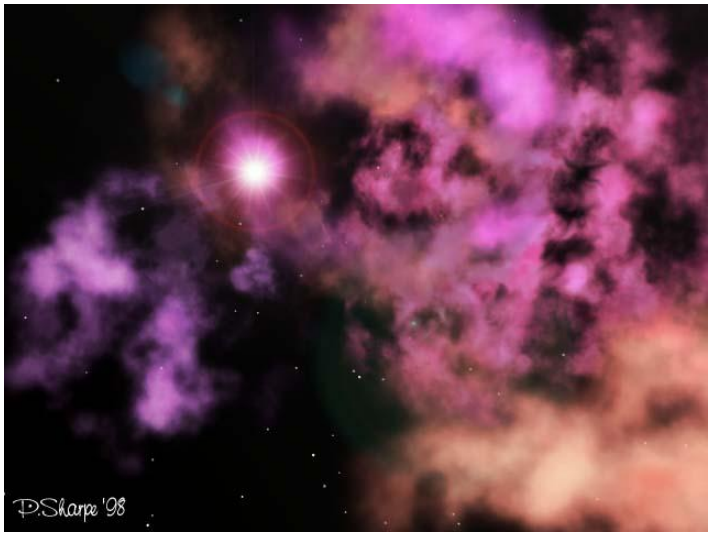
CGVR

- Display of Information
- Design
- Simulation
- Computer Art
- Entertainment

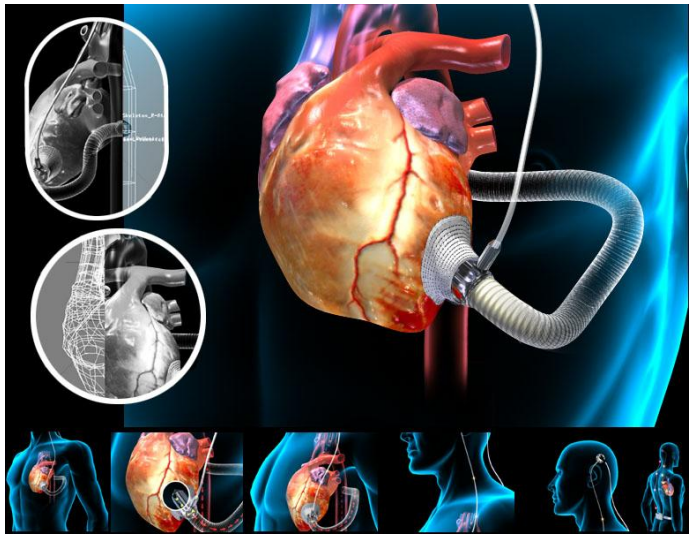
Display of Information

CGVR

- Graphics for Scientific, Engineering, and Medical Data



Nebula

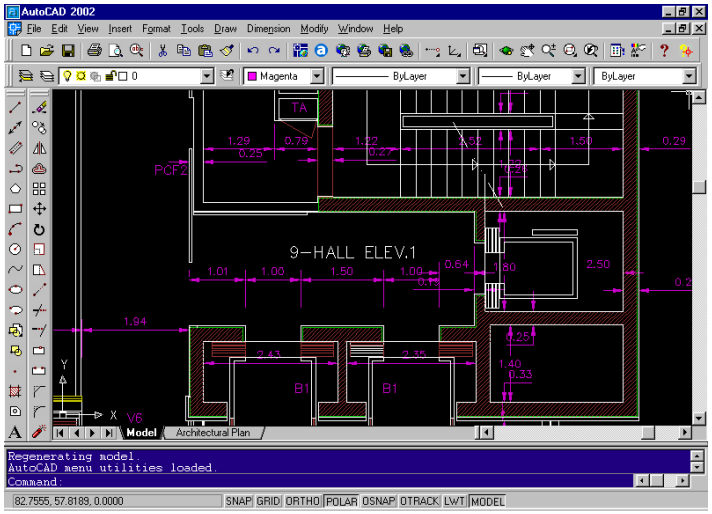


Medical Image

Design

CGVR

- Graphics for Engineering and Architectural System
- Design of Building, Automobile, Aircraft, Machine etc.



AutoCAD 2002

Interior Design

Simulation

CGVR

- Computer-Generated Models of Physical, Financial and Economic Systems for Educational Aids

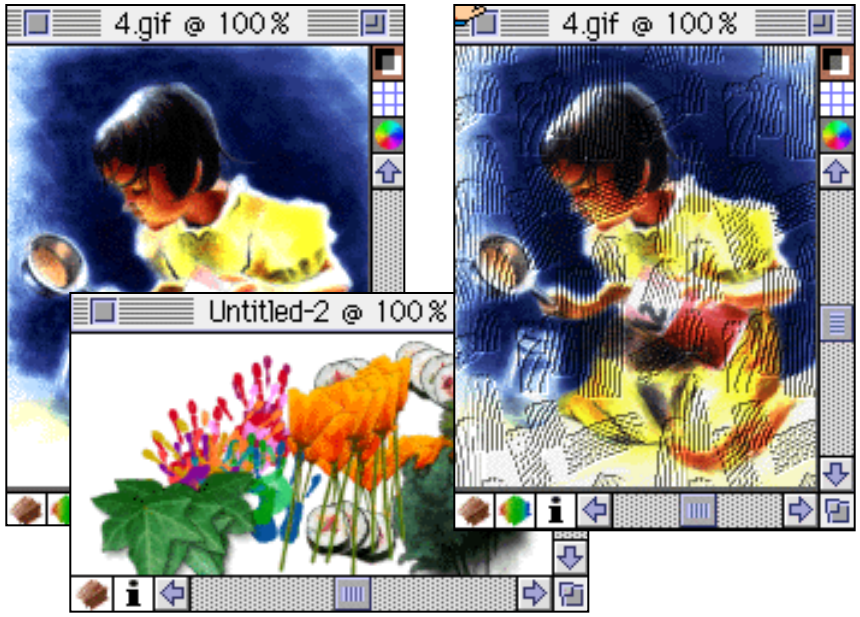
Flight Simulator

Mars Rover Simulator

Computer Art

CGVR

- Graphics for Artist



Metacreation Painter

Entertainment

CGVR

- Graphics for Movie, Game, VR etc.

Final Fantasy

Online Game

Contents

CGVR

- What is Computer Graphics?
  - Definition
  - Main themes
  - Applications
- Geometry in 3D Computer Graphics
  - Modeling
  - Rendering
  - Animation

Modeling

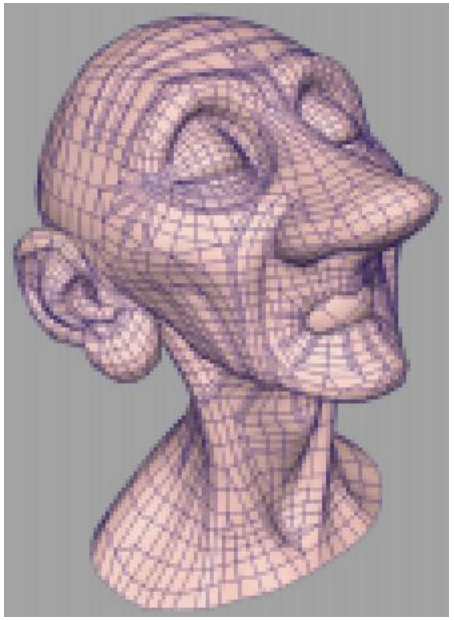
CGVR

- Reconstruction from a Point Set
- Simplification
- Subdivision

Modeling

CGVR

- Geri



Director Jan Pinkava with a model used in the development of *Geri's Game*.

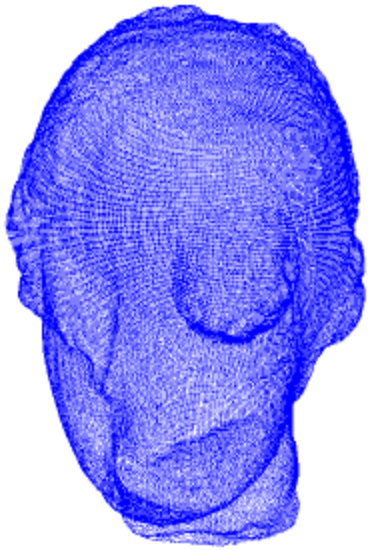
Subdivision Surface from a Point Set

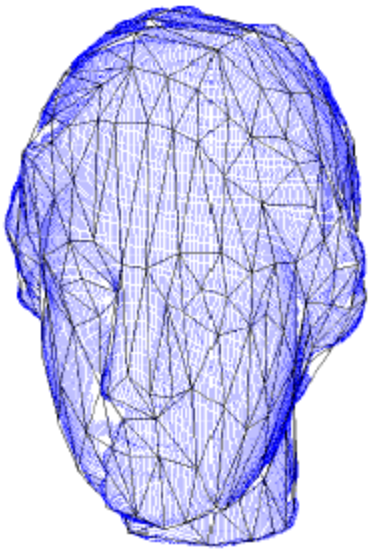
CGVR

Parametrization

Wavelet Analysis

Point Cloud → Base Mesh → Displace-ment Sampling → Subdivision Mesh





Subdivision Surface from a Point Set

CGVR

Parametrization

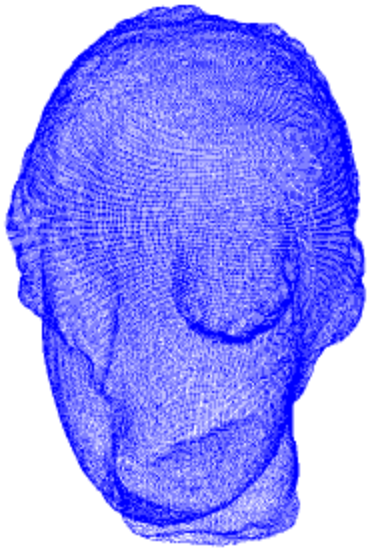
Wavelet Analysis

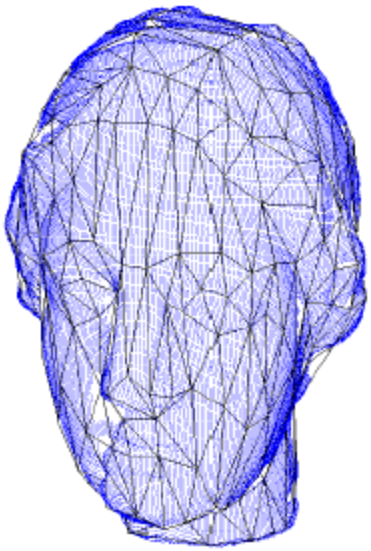
Point Cloud

Base Mesh

Displace-ment Sampling

Subdivision Mesh

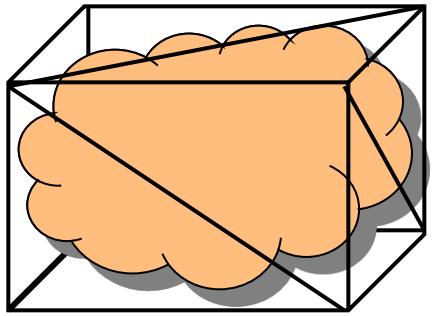




Reconstruction (1/2)

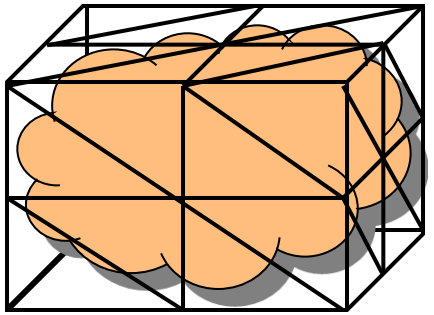
CGVR

- Step 1.
  - Make a **Bounding Cube** of Given Point Cloud



Bounding Cube

- Step 2.
  - **Subdivide the Cube** Several Times Until It Reaches Proper Resolution



Subdivided Cube

Reconstruction (2/2)

CGVR

- Step 3.
  - Apply Two Basic Operation Repeatedly
    - **Attraction Force**

$$f_{v_i} = x_i - v_i$$

$$M_b' = \{v_i + \mu f_{v_i} \mid v_i \in M_b\}$$

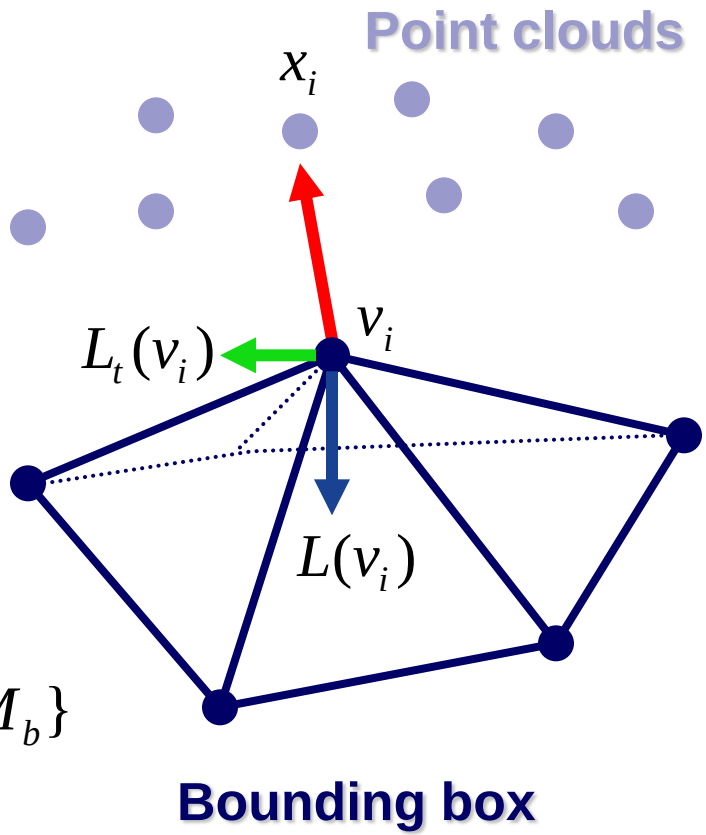
    - **Relaxation Force**

$$L(v_i) = \frac{1}{m} \sum_{v_j \in nbhd(v_i)} (v_j - v_i)$$

$$L_t(v_i) = L(v_i) - (L(v_i) \cdot n)n$$

$$M_b'' = \{v_i + \mu f_{v_i} + \lambda L_t(v_i) \mid v_i \in M_b\}$$

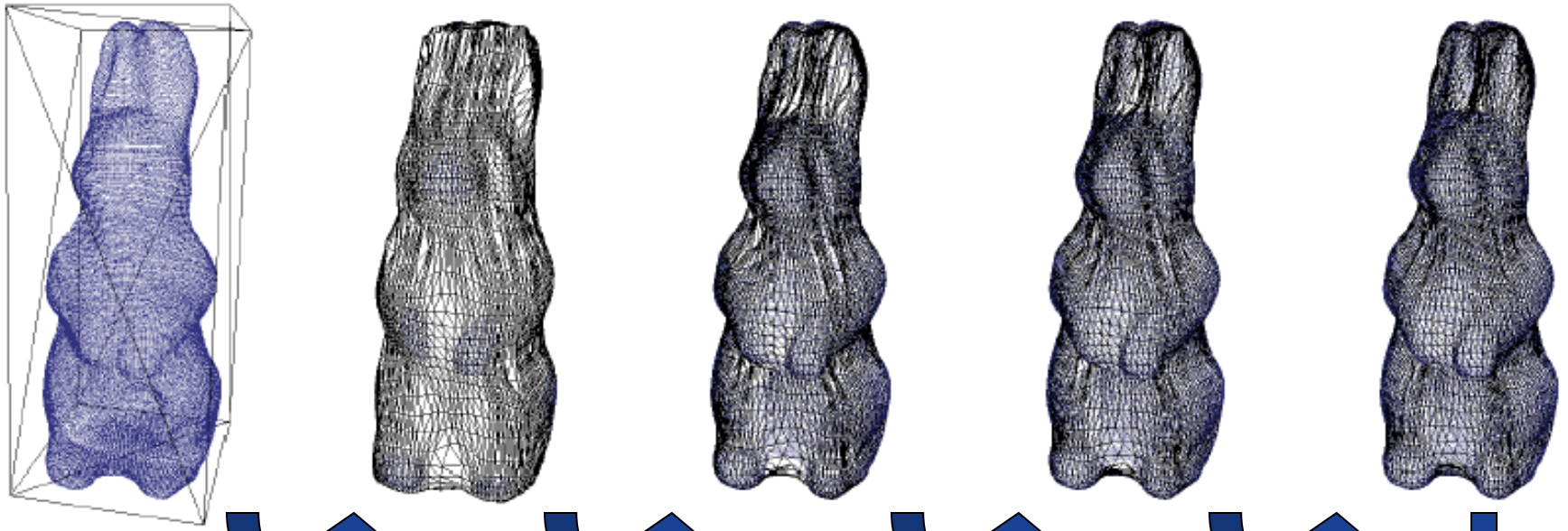
- Step 4.
  - Simplify It Using **Point-Based Simplification**



Example

CGVR

Step 1.



Step 2.

→ Subdivide 5 times

Step 3.

Step 4.

→ Simplify

Modeling

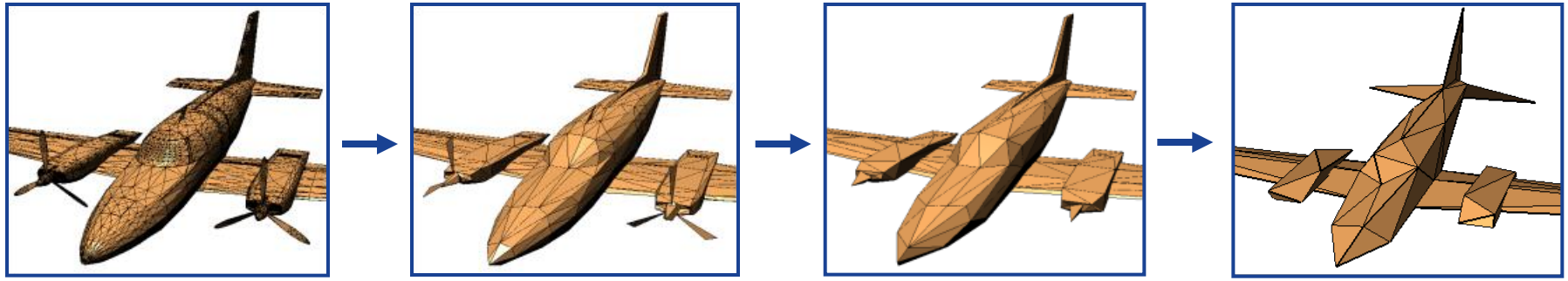
CGVR

- Reconstruction from a Point Set
- Simplification
- Subdivision

Surface Simplification

CGVR

- Reduce the number of polygons
  - try to retain a good approximation to the original shape and appearance
- Example by QEM [Garland 97]



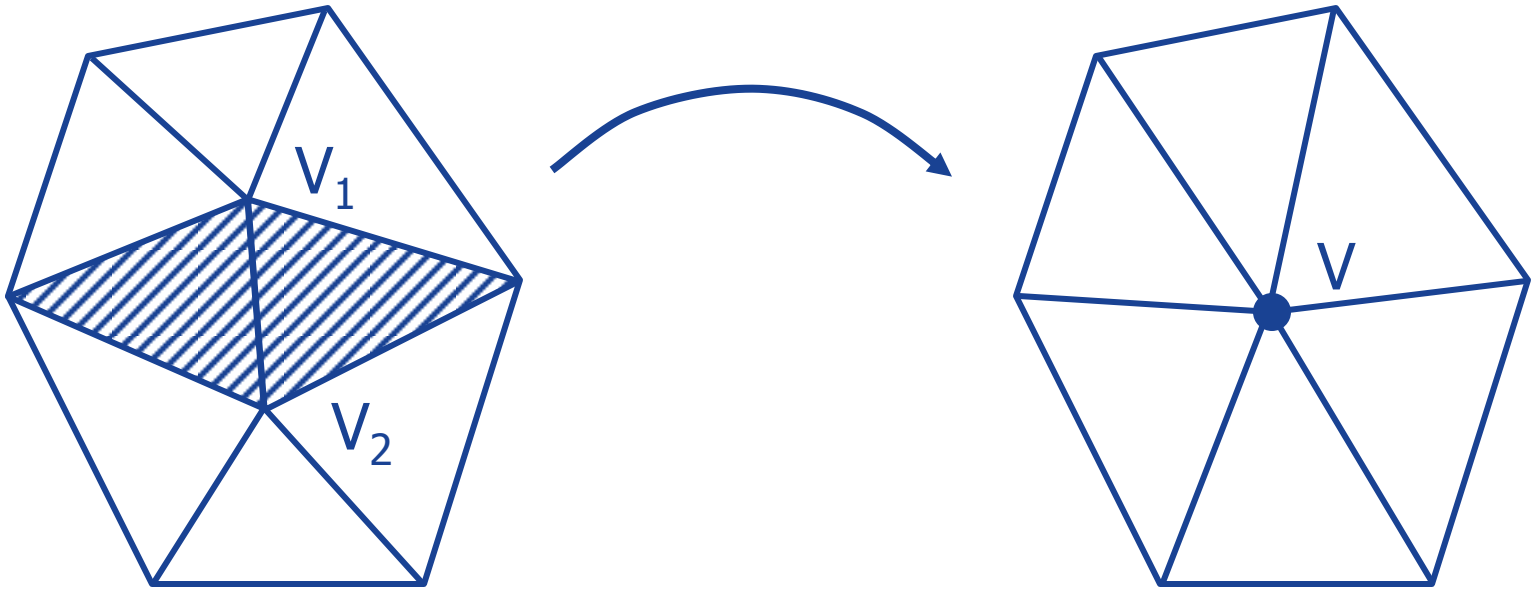
13,546 faces → 700 → 325 → 100 faces

➔ aim at fast and more accurate scheme !!

Edge Collapse

CGVR

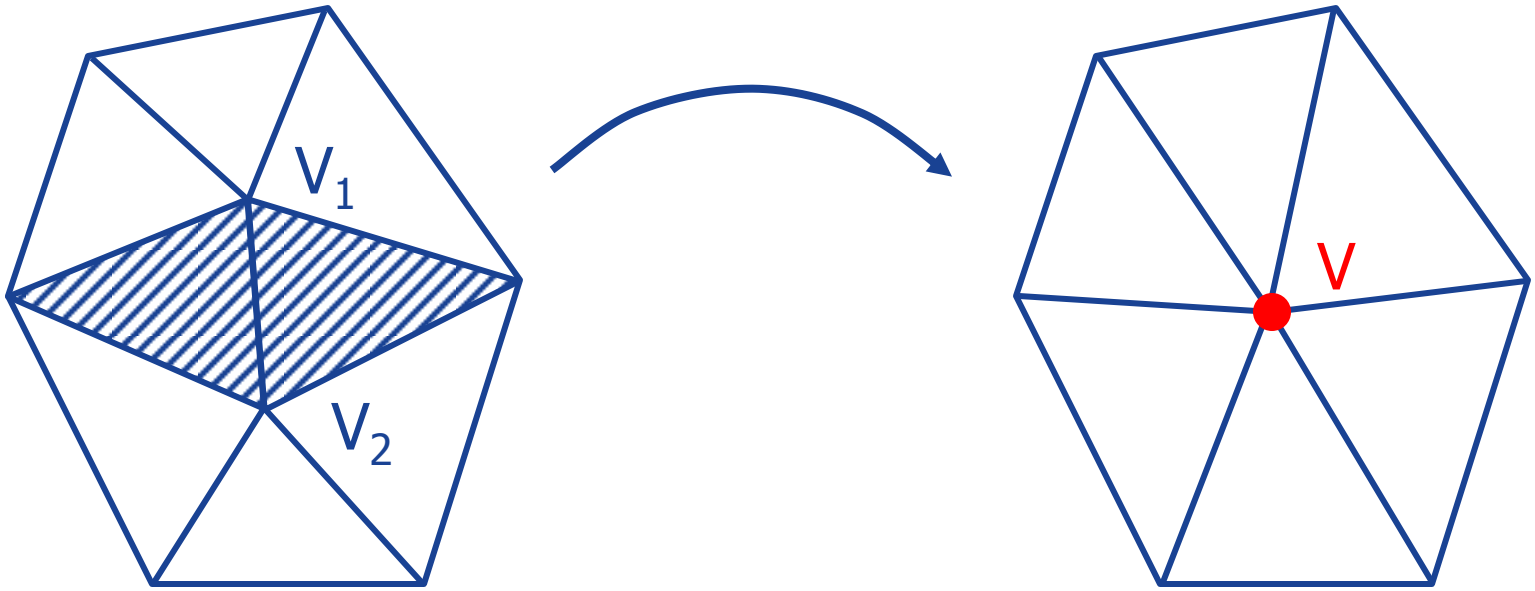
- Contract an edge into a single point



Edge Collapse

CGVR

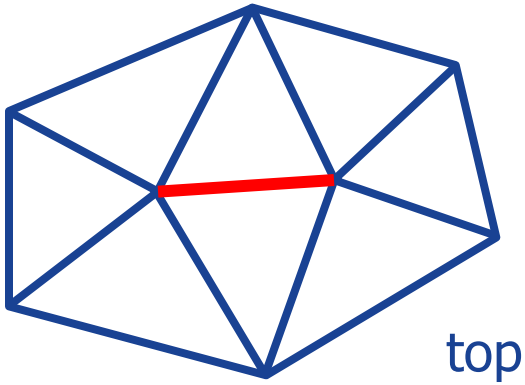
- Contract an edge into a single point
  - how to choose the *target position*?
  - how to choose the *edge to be collapsed*?



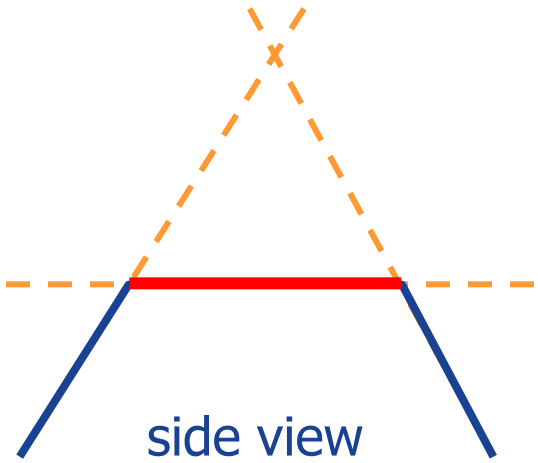
Quadric Error Metric

CGVR

- Choose the target position
  - to *minimize the sum of squared distances* to planes of neighbor triangles



top view

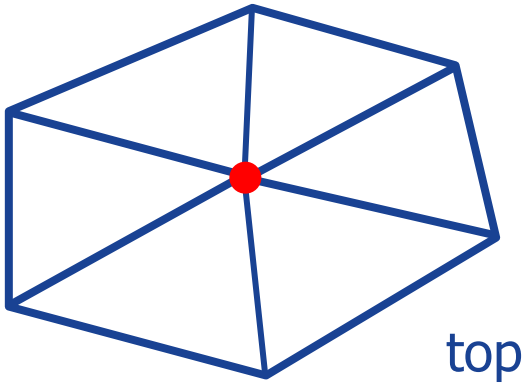


side view

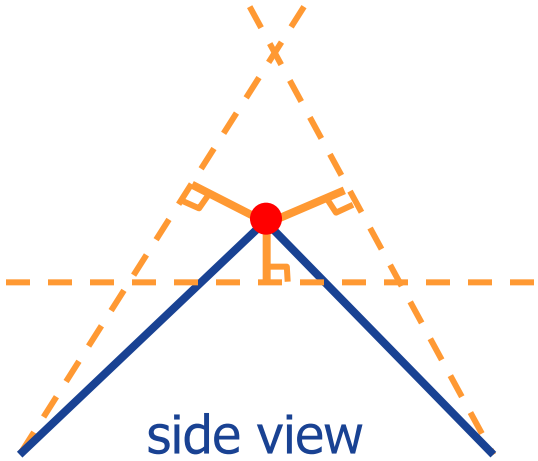
Quadric Error Metric

CGVR

- Choose the target position
  - to *minimize the sum of squared distances* to planes of neighbor triangles



top view

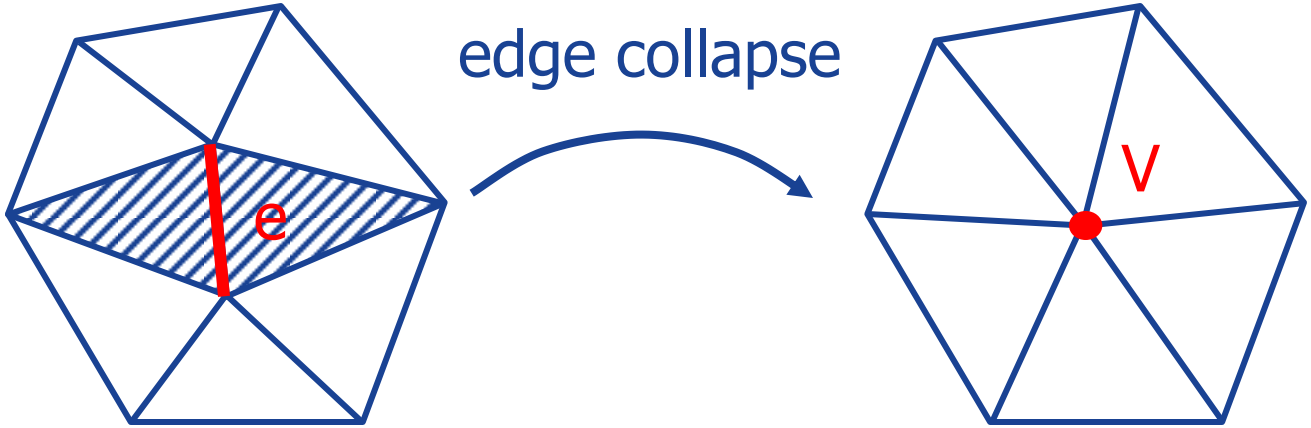


side view

- Choose the edge to be collapsed
  - the edge with the *least cost function* is collapsed first

→ the sum of squared distances

Method of QEM



- Solve the target position to minimize QEM

$$\mathbf{v} = -\mathbf{A}^{-1}\mathbf{b}$$

- Assign its cost function to each edge

$$QEM^{e}(\mathbf{v}) = \mathbf{v}^{T}\mathbf{A}\mathbf{v} + 2\mathbf{b}^{T}\mathbf{v} + c$$

- Construct a priority queue for the edges and collapse the one with minimum cost function

Discrete Differential Error Metric

CGVR

- A new method for edge cost function

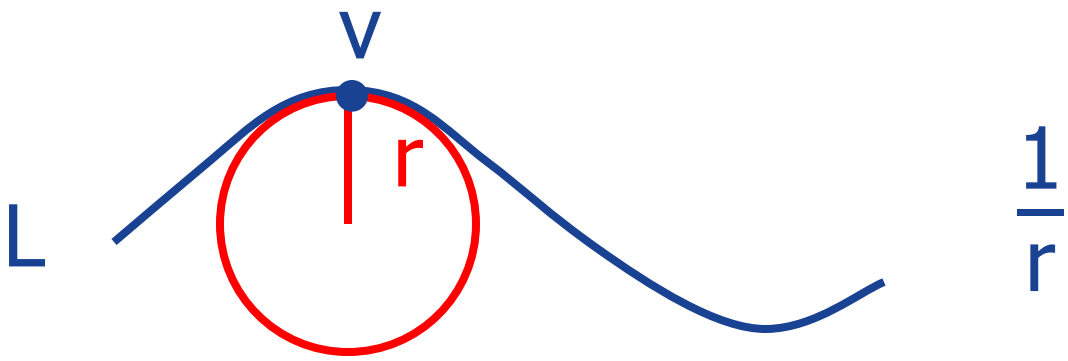
$$\text{DDEM} = \text{distance error metric (QEM)} + \text{tangential error metric} + \text{discrete curvature error metric}$$

- Properties
  - *relatively fast* – not increase the dimension of the matrix of QEM
  - *more accurate* – take into account of tangential and discrete curvature error metrics

Curvature in 2D

CGVR

- Measurement of bending
- Two criteria
  - the curvature of a *straight* line is *zero*
  - the curvature of a *circle* is the *same* at each point.

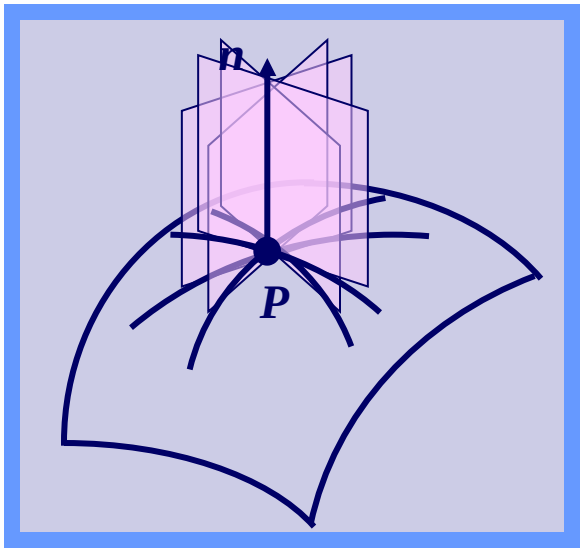


the derivative has constant length 1/r

Geometric Curvatures

CGVR

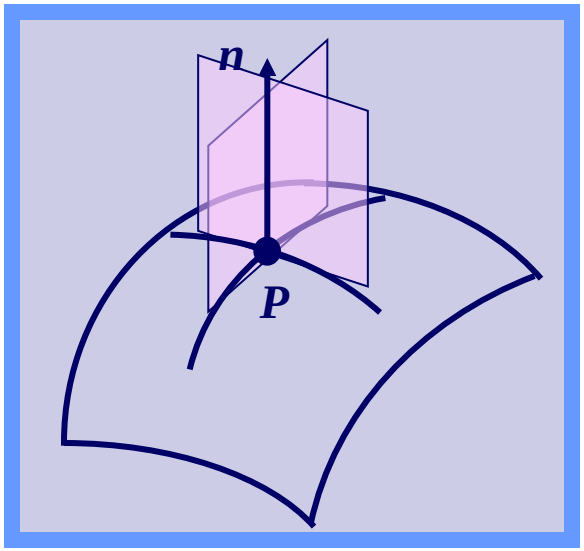
- Curve of intersection of a plane containing the normal to the surface at *P*



Geometric Curvatures

CGVR

- Curve of intersection of a plane containing the normal to the surface at *P*
- Principal, mean, and Gaussian curvatures



$$P : k_{\min}, k_{\max}$$

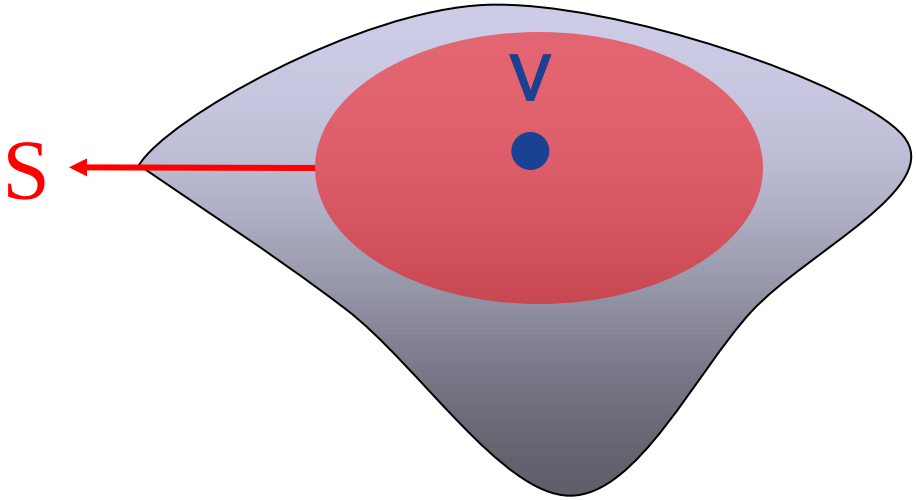
$$H = \frac{k_{\min} + k_{\max}}{2}$$

$$K = k_{\min} k_{\max}$$

Integral Discrete Curvatures

CGVR

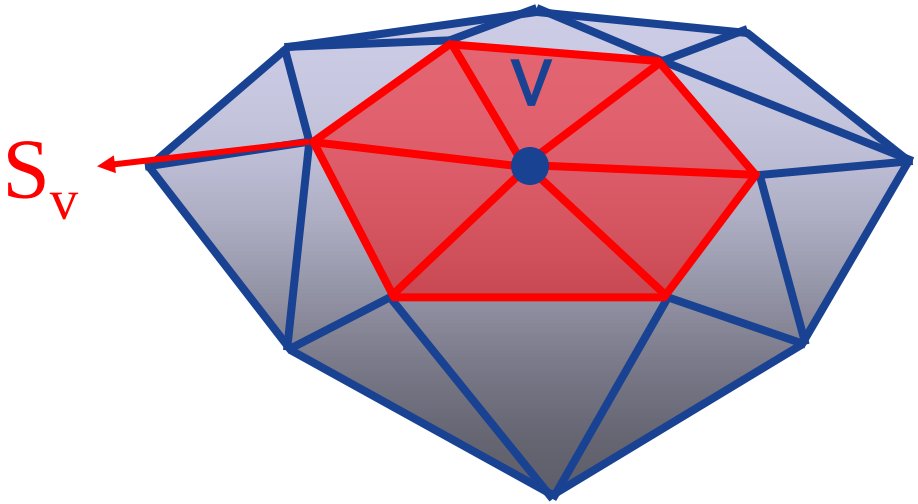
- Discrete curvature with respect to the area $S=S_v$ attributed to v



Integral Discrete Curvatures

CGVR

- Discrete curvature with respect to the area $S=S_v$ attributed to v

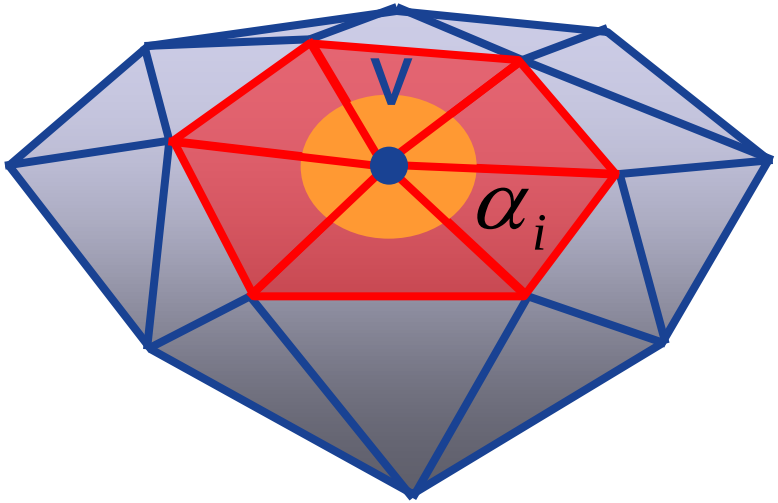


Integral Discrete Curvatures

CGVR

- Discrete curvature with respect to the area $S=S_v$ attributed to v
- Integral Guassian curvature $\overline{K} = \overline{K}_v$

$$\overline{K} = \int_S K = 2\pi - \sum_{i=1}^{n} \alpha_i$$



Integral Discrete Curvatures

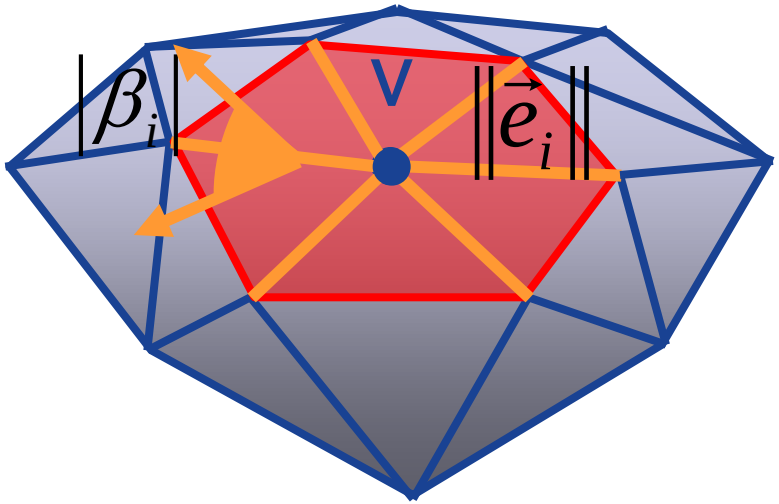
CGVR

- Discrete curvature with respect to the area $S=S_v$ attributed to v
- Integral Guassian curvature $\overline{K}_n = \overline{K}_v$

$$\overline{K} = \int_S K = 2\pi - \sum_{i=1}^{n} \alpha_i$$

- Integral absolute mean curvature $\left|\overline{H}\right| = \left|\overline{H}_v\right|$

$$\left|\overline{H}\right| = \int_S |H| = \frac{1}{4}\sum_{i=1}^{n} \left\|\vec{e}_i\right\| |\beta_i|$$



Discrete Curvature at a Vertex

CGVR

- Assume the curvature to be uniformly distributed around the vertex

Discrete Curvature at a Vertex

CGVR

- Assume the curvature to be uniformly distributed around the vertex
- *Gaussian* and *absolute mean curvatures* are simply normalized by the area

$$K = \frac{\overline{K}}{S} \qquad |H| = \frac{|\overline{H}|}{S}$$

Discrete Curvature at a Vertex

CGVR

- Assume the curvature to be uniformly distributed around the vertex
- ***Gaussian*** and ***absolute mean curvatures*** are simply normalized by the area

$$K = \frac{\overline{K}}{S} \qquad |H| = \frac{|\overline{H}|}{S}$$

- The sum of the *absolute principal curvatures*

$$|\kappa_1| + |\kappa_2| = \begin{cases} 2|H|, & \text{if } K \geq 0 \\ 2\sqrt{|H|^2 - K}, & \text{otherwise} \end{cases}$$

Discrete Curvature Norm

CGVR

- Norm of principal curvatures at all vertices
  - L_1-norm $$\|\kappa\|_1 = \sum_{v \in V} |\kappa_1| + |\kappa_2|$$
  $$= \sum_{v \in V} \begin{cases} 2|H|, & \text{if } K \geq 0 \\ 2\sqrt{|H|^2 - K}, & \text{otherwise} \end{cases}$$
  - L_2-norm $$\|\kappa\|_2^2 = \sum_{v \in V} |\kappa_1|^2 + |\kappa_2|^2$$
  $$= \sum_{v \in V} \left(4H^2 - 2K\right)$$

DDEM

CGVR

$$DDEM^{e}(\mathbf{v}) = Q^{e}(\mathbf{v}) + T^{e}(\mathbf{v}) + C^{e}(\mathbf{v})$$

DDEM

CGVR

$$DDEM^{e}(\mathbf{v}) = Q^{e}(\mathbf{v}) + T^{e}(\mathbf{v}) + C^{e}(\mathbf{v})$$

- Quadric error metric

DDEM

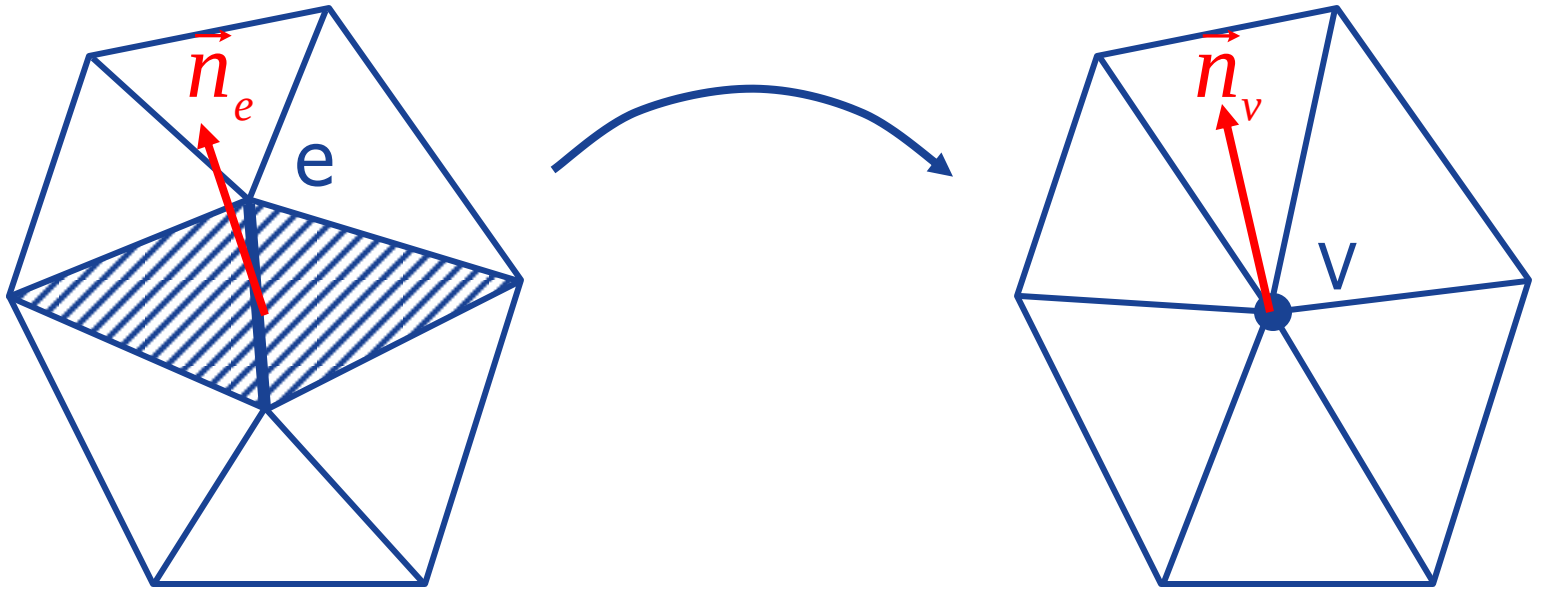
CGVR

$$DDEM^{e}(\mathbf{v}) = Q^{e}(\mathbf{v}) + T^{e}(\mathbf{v}) + C^{e}(\mathbf{v})$$

- Quadric error metric
- Tangential error metric
  - magnitude of a difference vector between average normal to edge e and average normal to vertex v

Tangential Error Metric

CGVR



$$T^e = \left\| \vec{n}_e - \vec{n}_v \right\|$$

DDEM

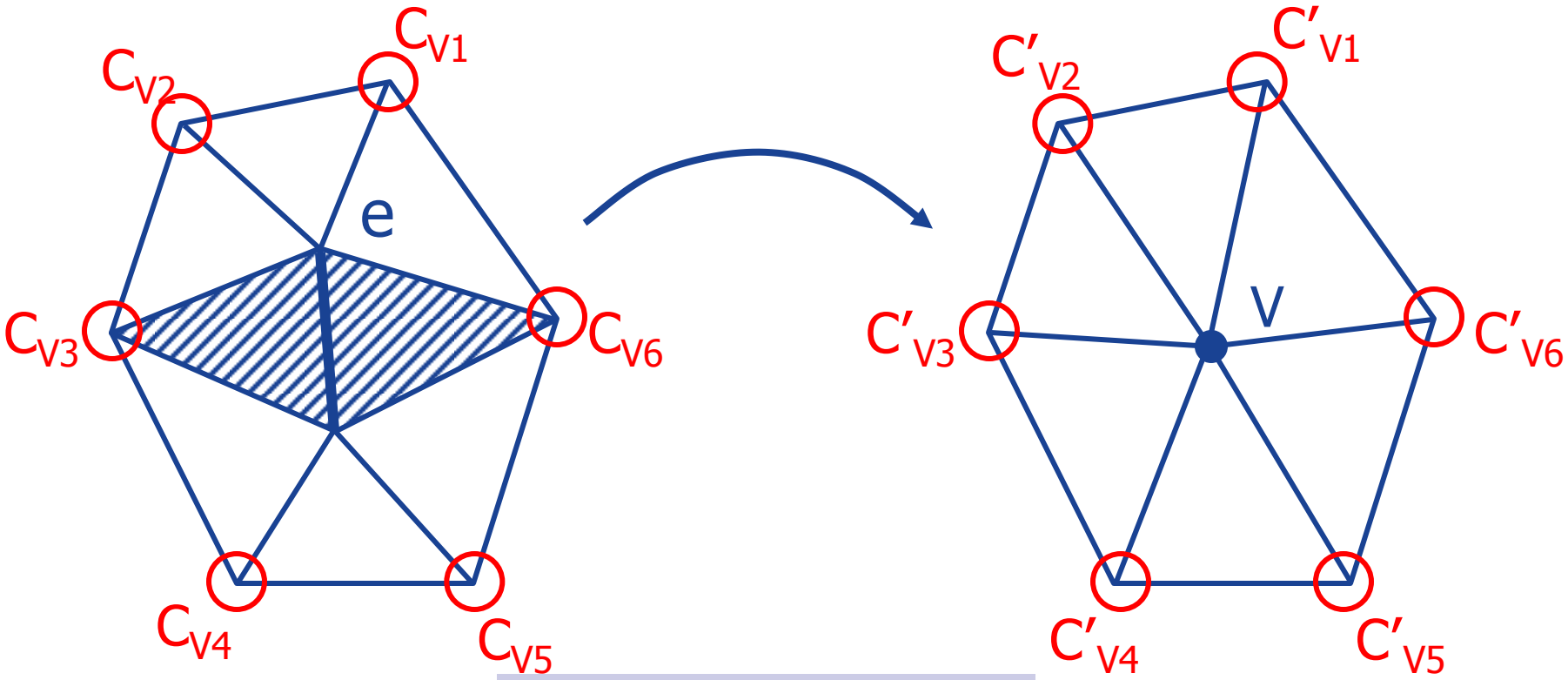
CGVR

$$DDEM^{e}(\mathbf{v}) = Q^{e}(\mathbf{v}) + T^{e}(\mathbf{v}) + C^{e}(\mathbf{v})$$

- Quadric error metric
- Tangential error metric
  - magnitude of a difference vector between average normal to edge e and average normal to vertex v
- Discrete curvature error metric
  - difference between the discrete curvature of the cell of an edge and its discrete curvature after an edge collapse

Discrete Curvature Error Metric

CGVR



$$C^e = \sum_{i=1}^{n} \left| C_{v_i} - C'_{v_i} \right|$$

DDEM Algorithm

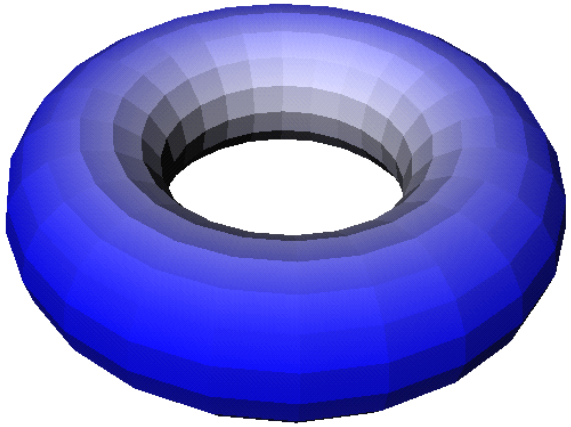
CGVR

- Initialize quadrics Q^e at each vertex
  - (A, b, *c*)
- Select the target position v at each edge
  - $v = -A^{-1}b$
- Compute T^e and C^e using v at each edge
  - (A, b, $c+T^e+C^e$)
- Construct a priority queue for edge collapses with $DDEM^e(v)$ cost function
  - $DDEM^e(v) = v^T A v + 2 b^T v + c + T^e + C^e$
- Collapse the edge with the least cost function

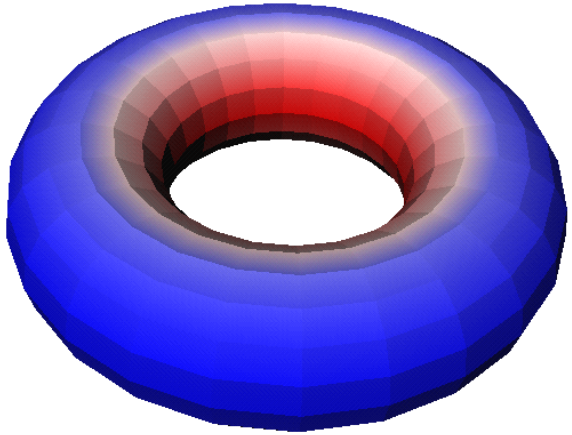
Results – Discrete Curvatures (1/3)

CGVR

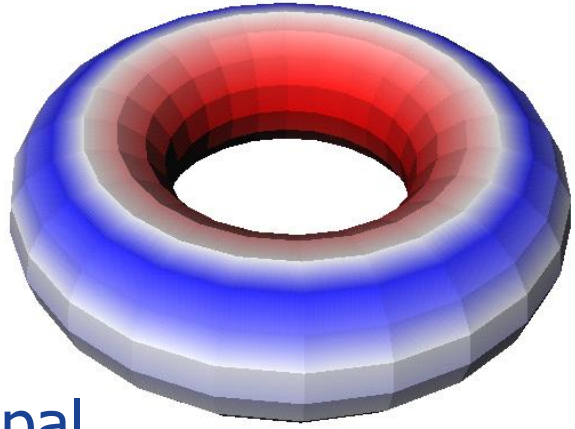
- Torus model (768 faces)



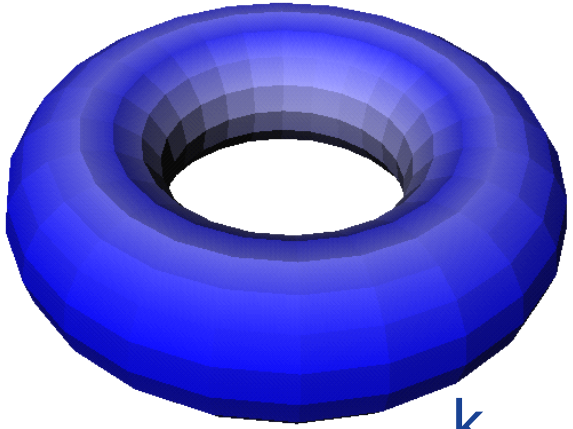
mean



Gaussian



k_{min} principal

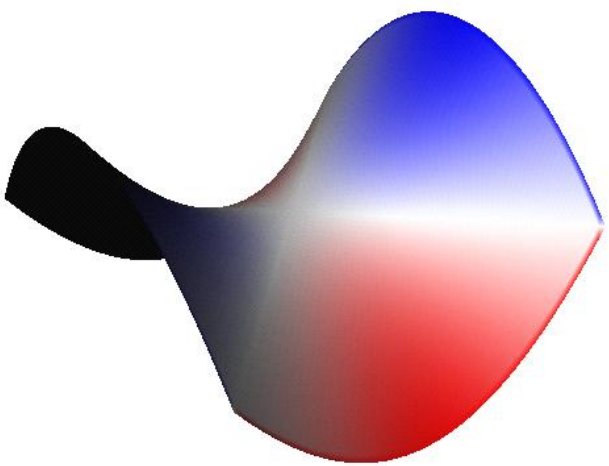


k_{max} principal

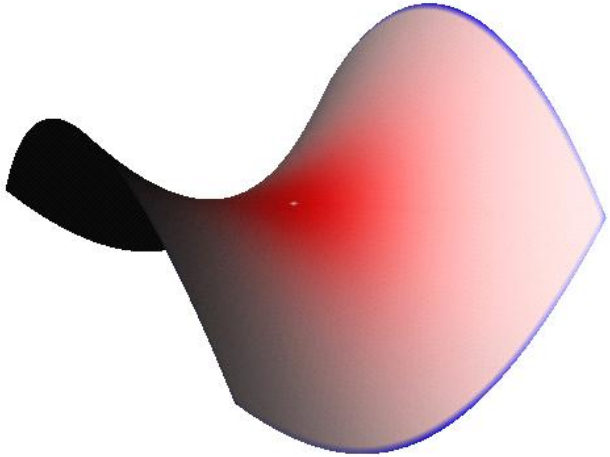
Results – Discrete Curvatures (2/3)

CGVR

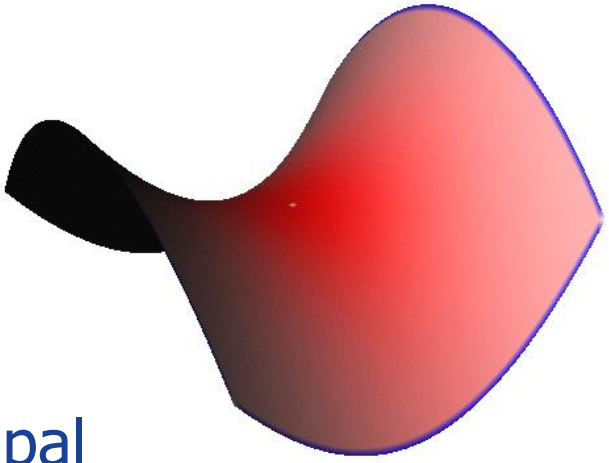
- Saddle model (8,192 faces)



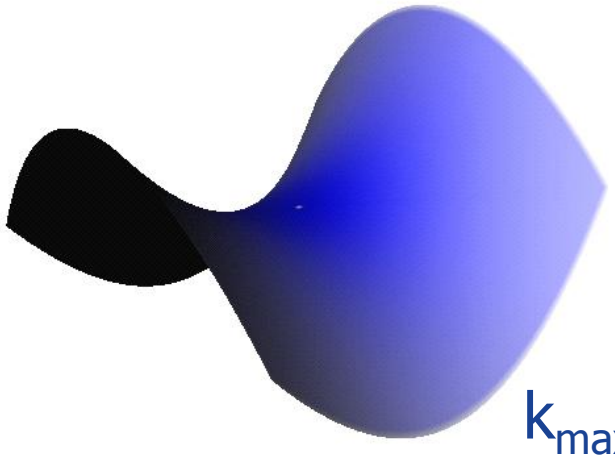
mean



Gaussian



k_{min} principal

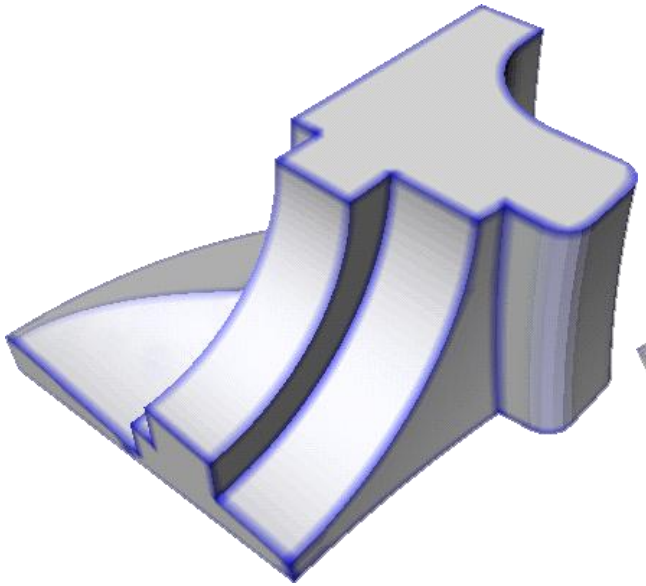


k_{max} principal

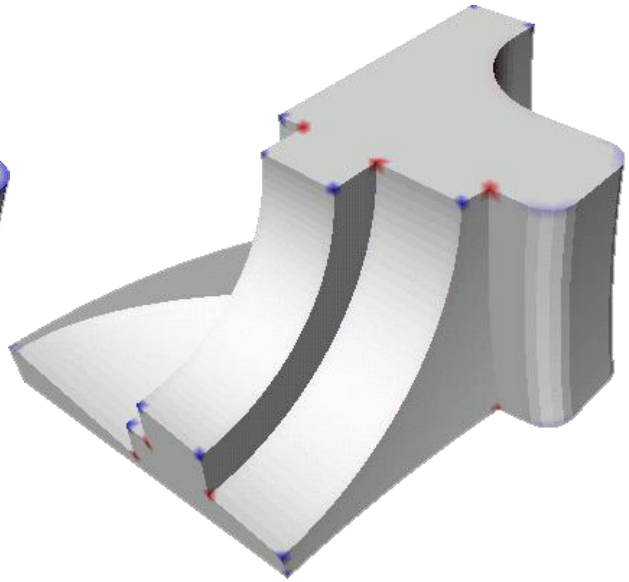
Results – Discrete Curvatures (3/3)

CGVR

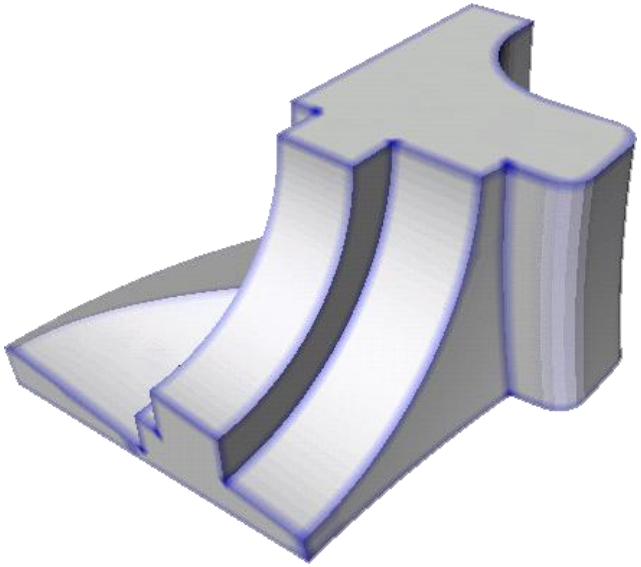
- Fandisk model (12,946 faces)



absolute mean



Gaussian

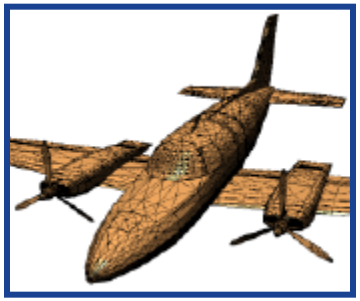


sum of absolute principal

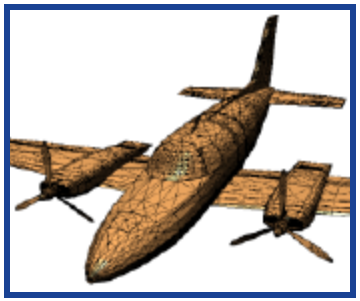
Results – Simplification (3/5)

CGVR

- Step by step



DDEM



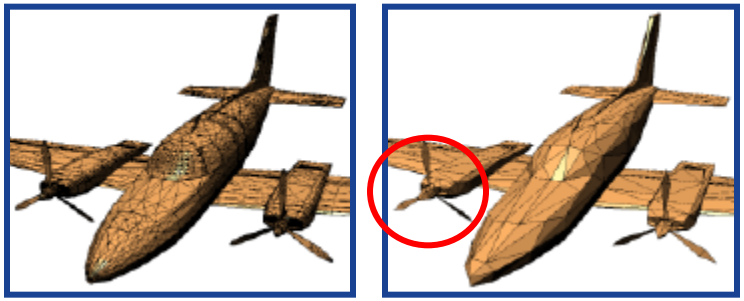
QEM

(a) original
(13,546 faces)

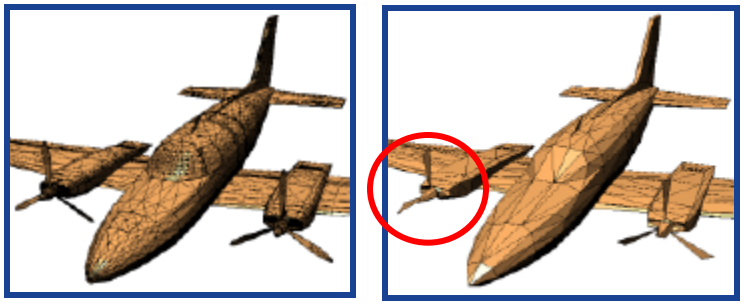
Results – Simplification (3/5)

CGVR

- Step by step



DDEM



QEM

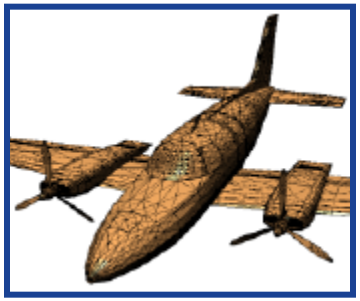
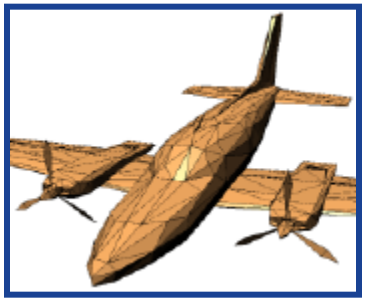
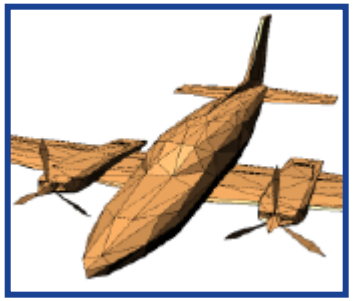
(a) Original (13,546 faces)

(b) Simplified (700 faces)

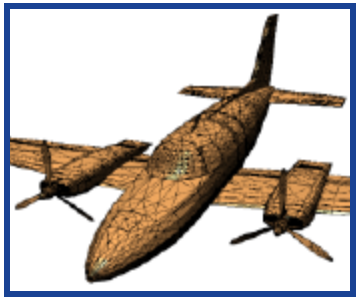
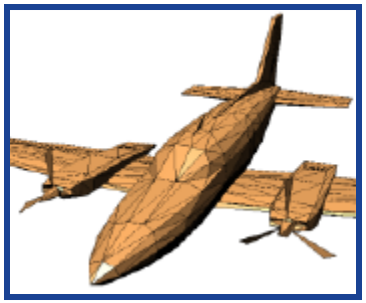
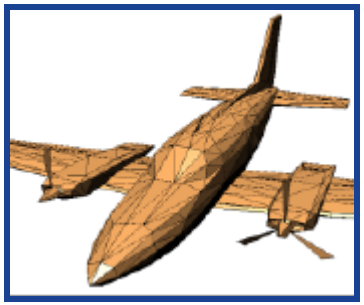
Results – Simplification (3/5)

CGVR

- Step by step

DDEM

QEM

(a) Original (13,546 faces)

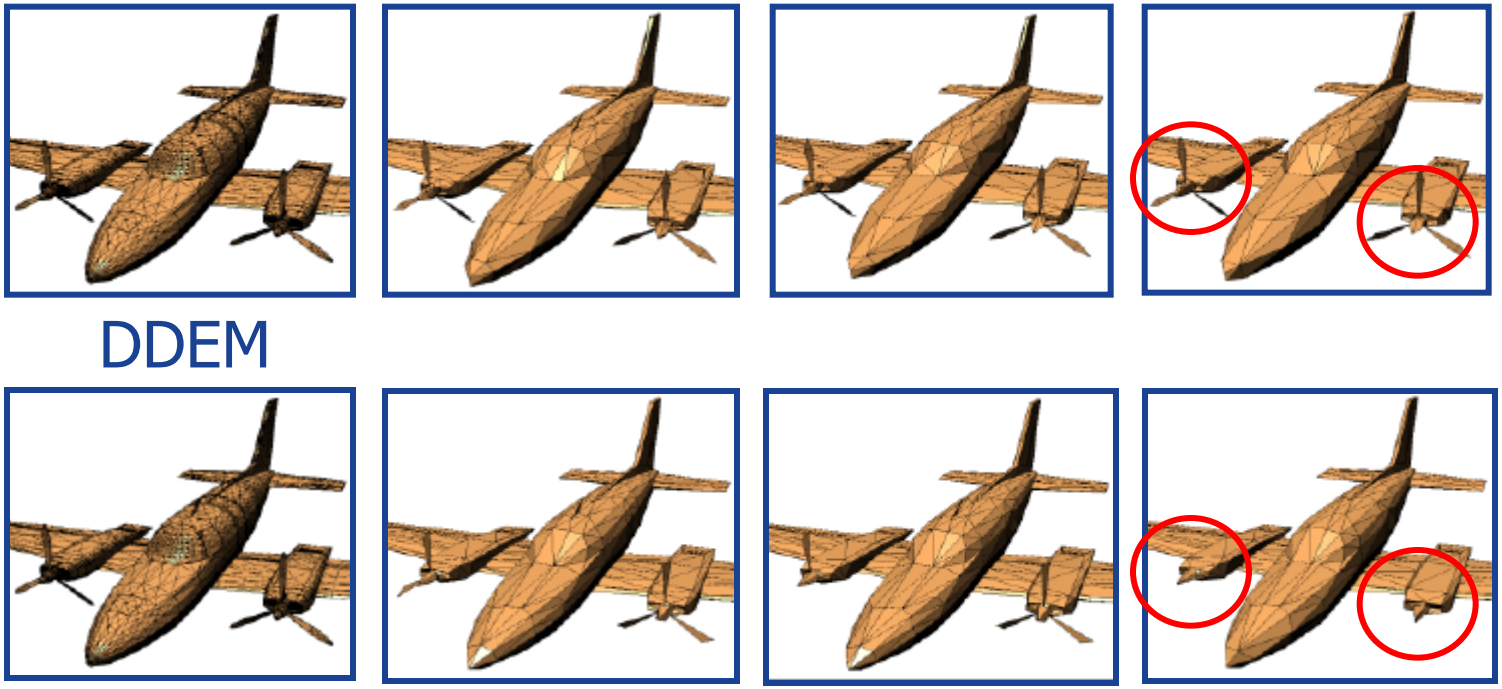
(b) Simplified (700 faces)

(c) 650 faces

Results – Simplification (3/5)

CGVR

- Step by step



DDEM

QEM

(a) Original (13,546 faces)

(b) Simplified (700 faces)

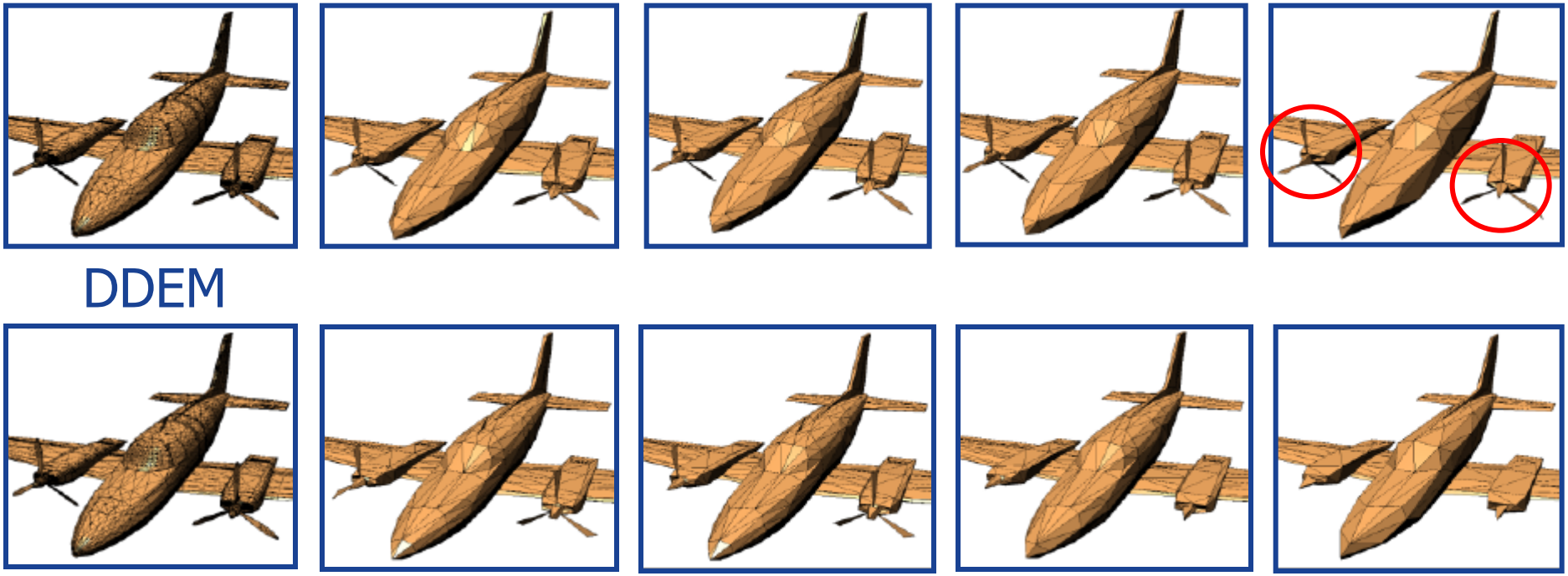
(c) 650 faces

(d) 550 faces

Results – Simplification (3/5)

CGVR

- Step by step



(a) Original (13,546 faces) (b) Simplified (700 faces) (c) 650 faces (d) 550 faces (e) 325 faces

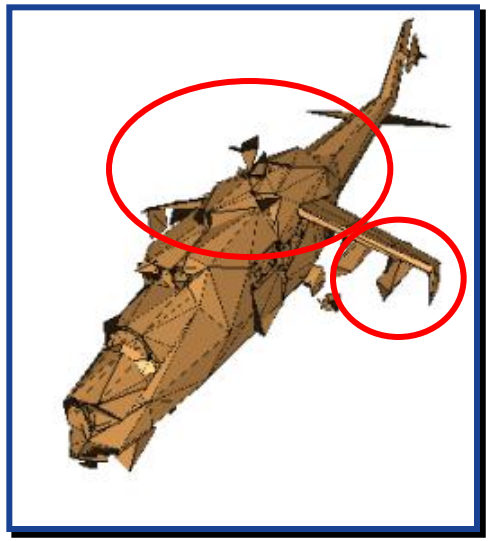
Results – Simplification (4/5)

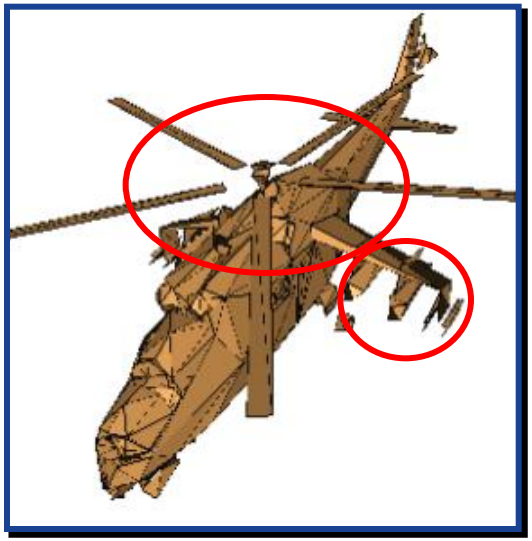
CGVR

- Helicopter model

QEM

DDEM





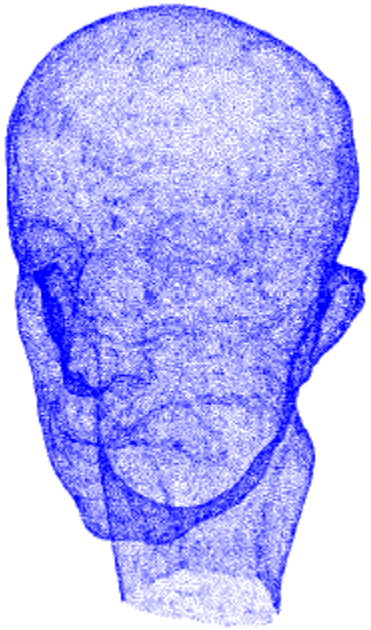
(a) Original
(34,708 faces)

(b) Simplified
(986 faces)

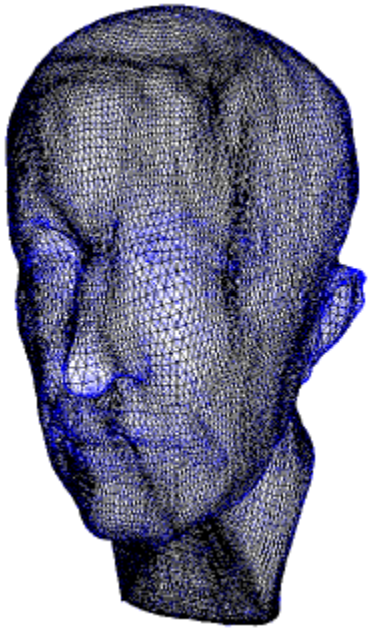
Result – Reconstruction (1/2)

CGVR

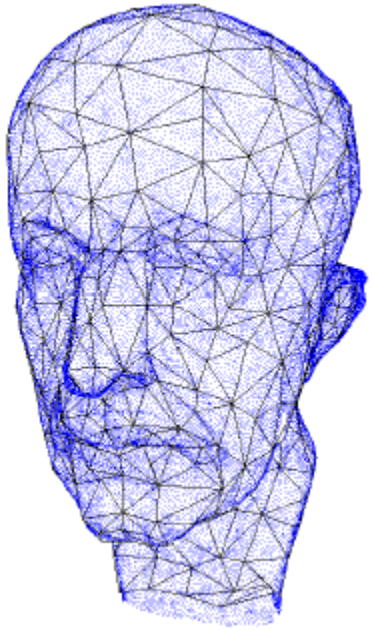
- Max Planck Model



Point Clouds



Shrink-Wrapped Bounding Cube

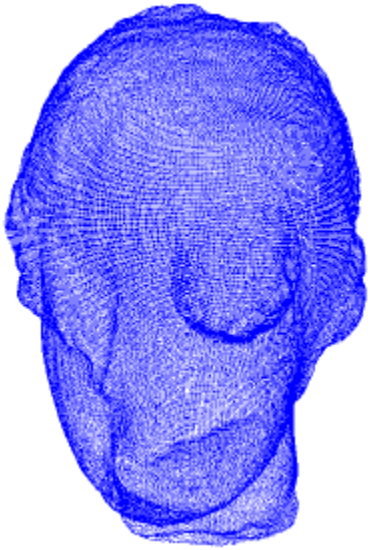


Base Mesh

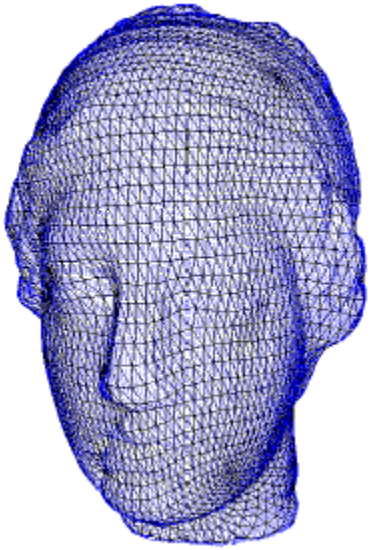
Result – Reconstruction (2/2)

CGVR

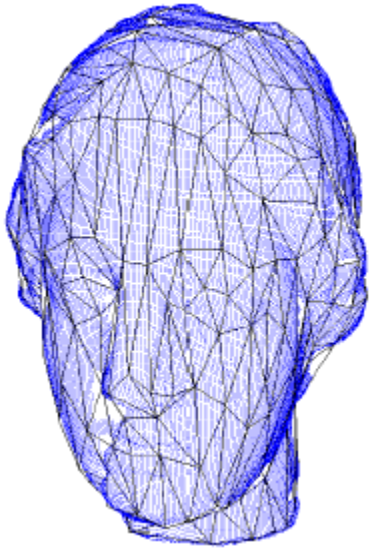
- Igea Model



Point Clouds



Shrink-Wrapped Bounding Cube



Base Mesh

Modeling

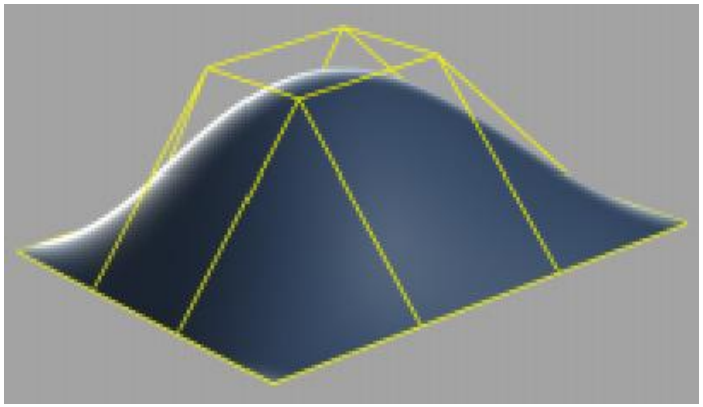
CGVR

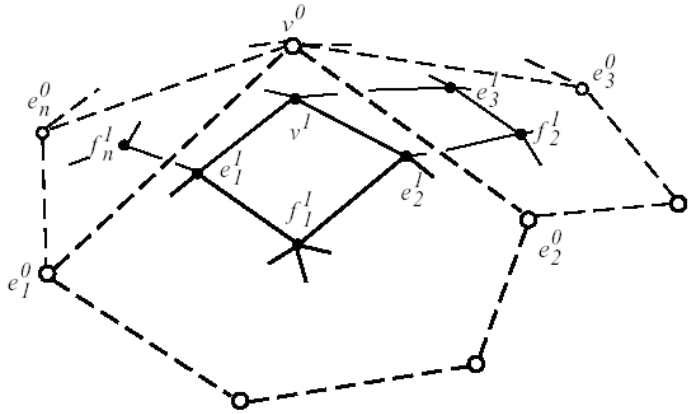
- Reconstruction from a Point Set
- Simplification
- Subdivision

Subdivision

CGVR

- Geri
  - Geri's hand as a piecewise smooth Catmull-Clark surface

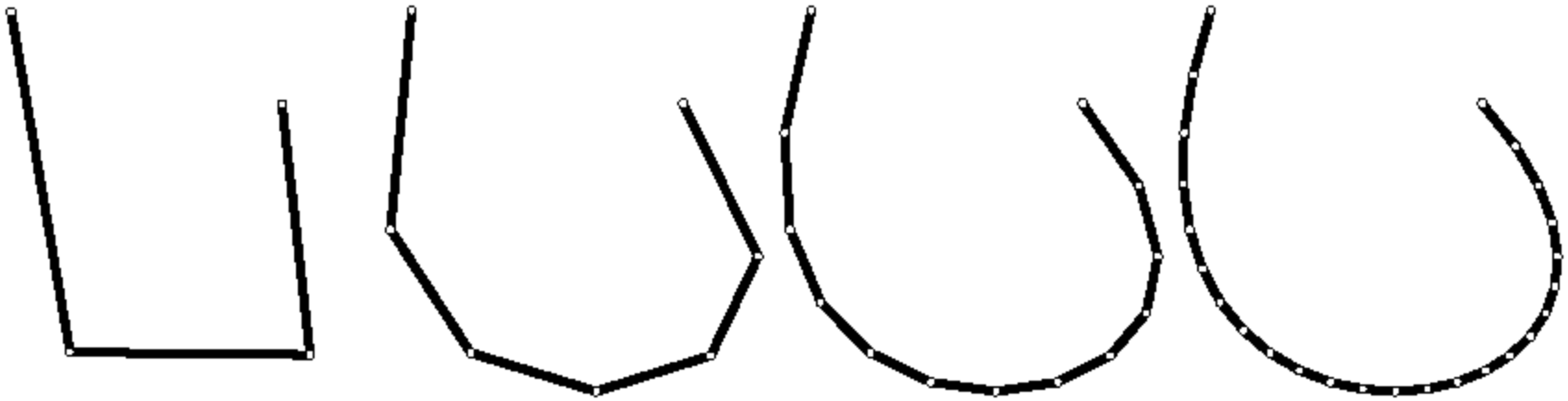




Subdivision Curves

CGVR

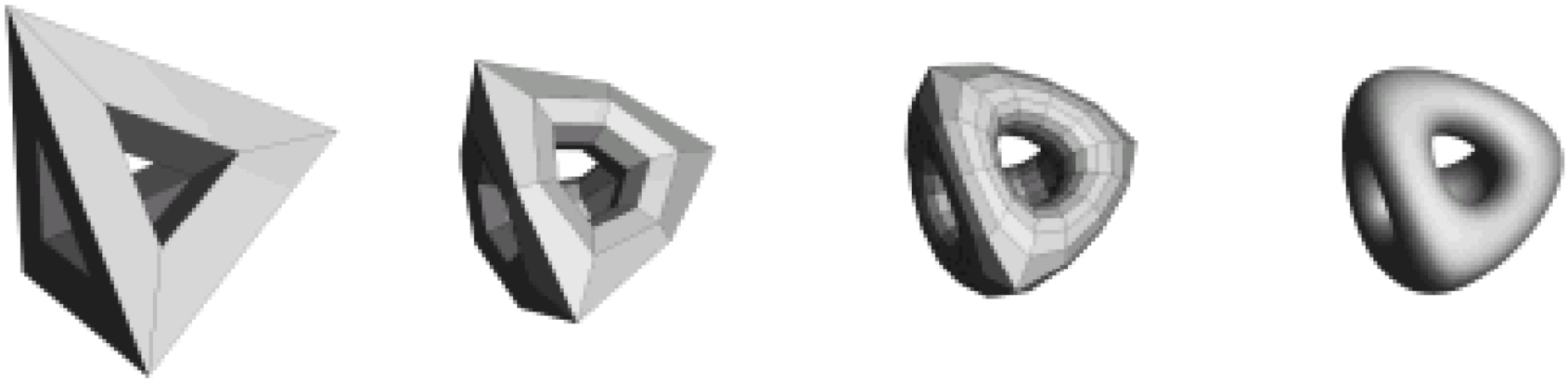
- How do You Make a Smooth Curve?



Subdivision Surfaces

CGVR

- **Coarse Mesh & Subdivision Rule**
  - Define smooth surface as limit of sequence of refinements



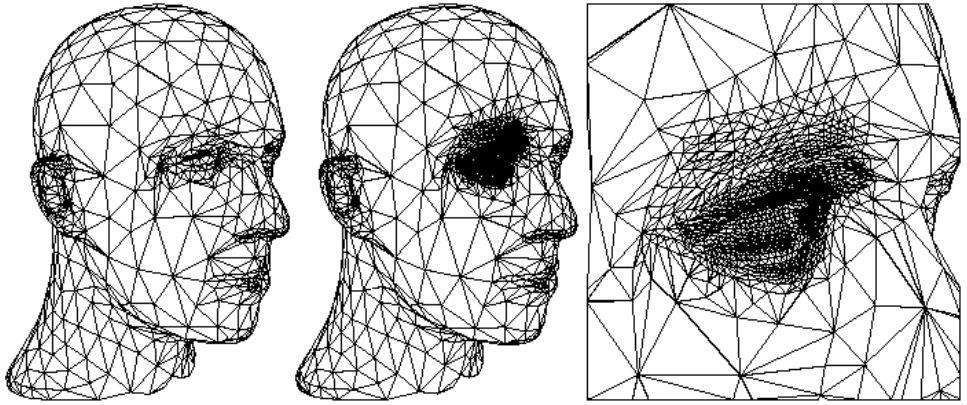
(a) (b) (c) (d)

Advantages & Difficulties

CGVR

- **Advantages:**
  - Simple method for describing complex surfaces
  - Relatively easy to implement
  - Arbitrary topology
  - Smoothness guarantees
  - Multiresolution

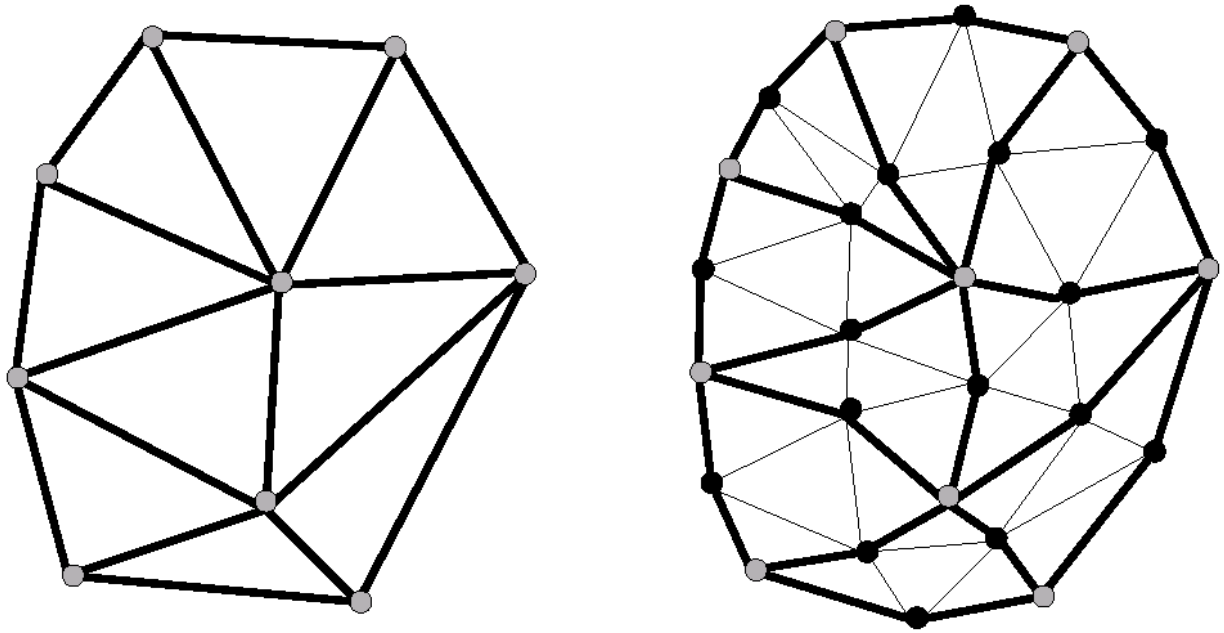
- **Difficulties:**
  - Intuitive specification
  - Parameterization
  - Intersections



Key Questions

CGVR

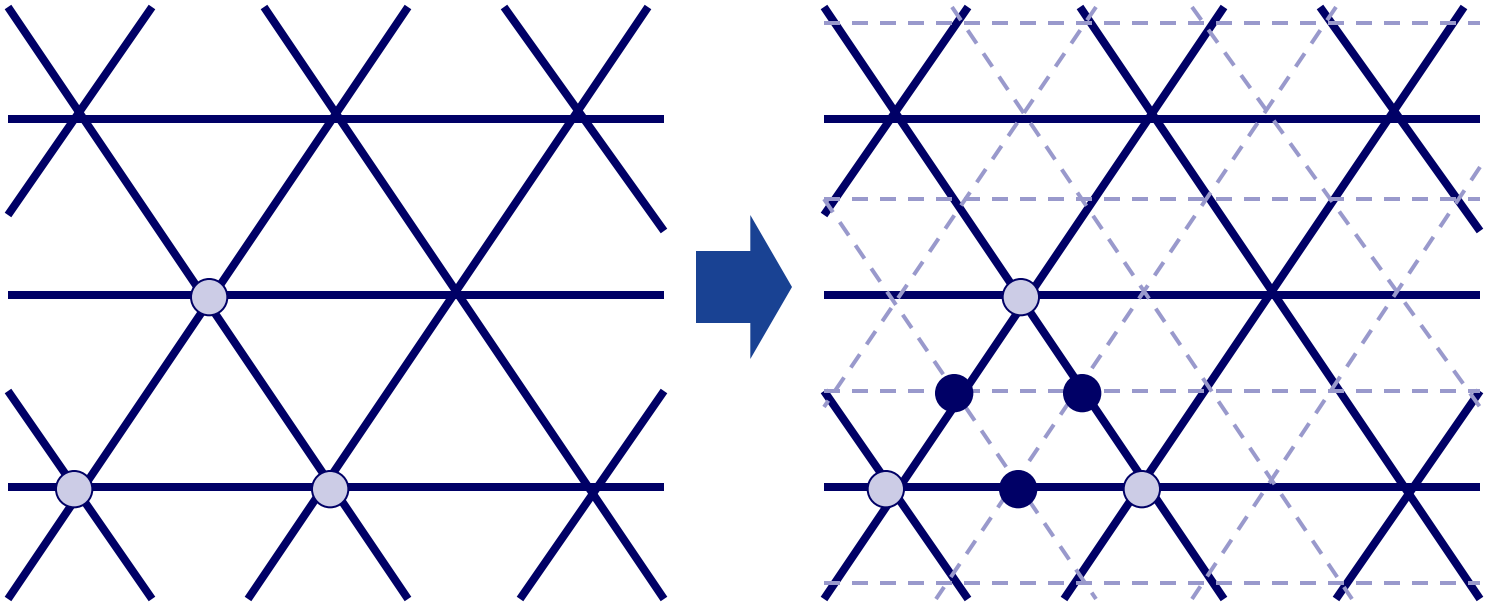
- **How Refine Mesh?**
  - Aim for properties like smoothness



Loop Subdivision Scheme

CGVR

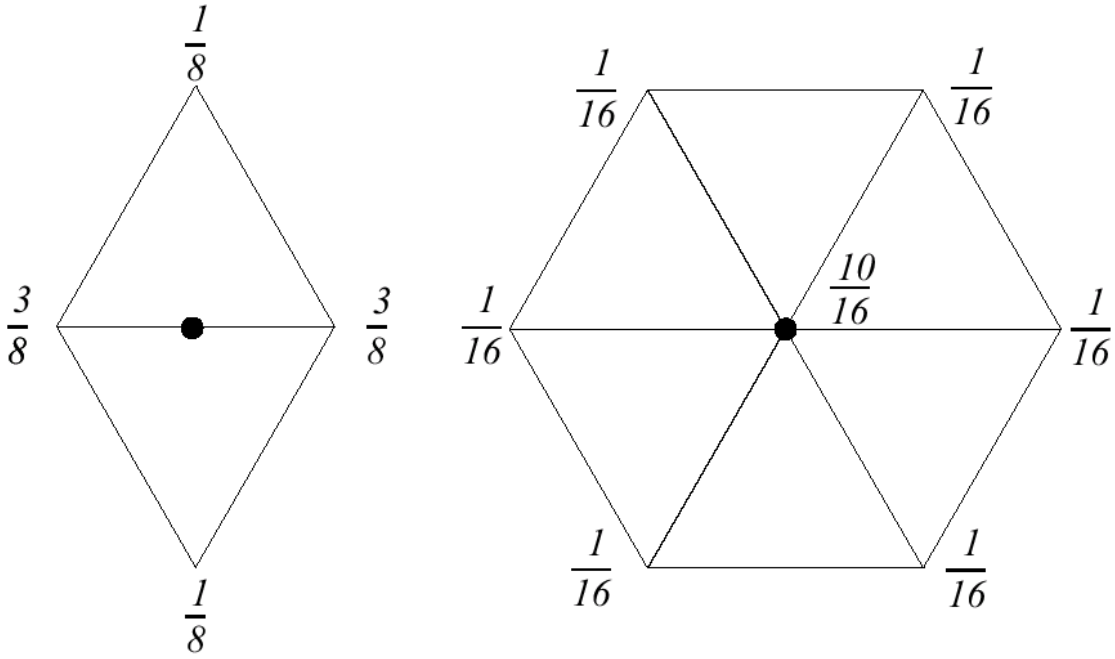
- **How Refine Mesh?**
  - Refine each triangle into 4 triangles by splitting each edge and connecting new vertices



Loop Subdivision Scheme

CGVR

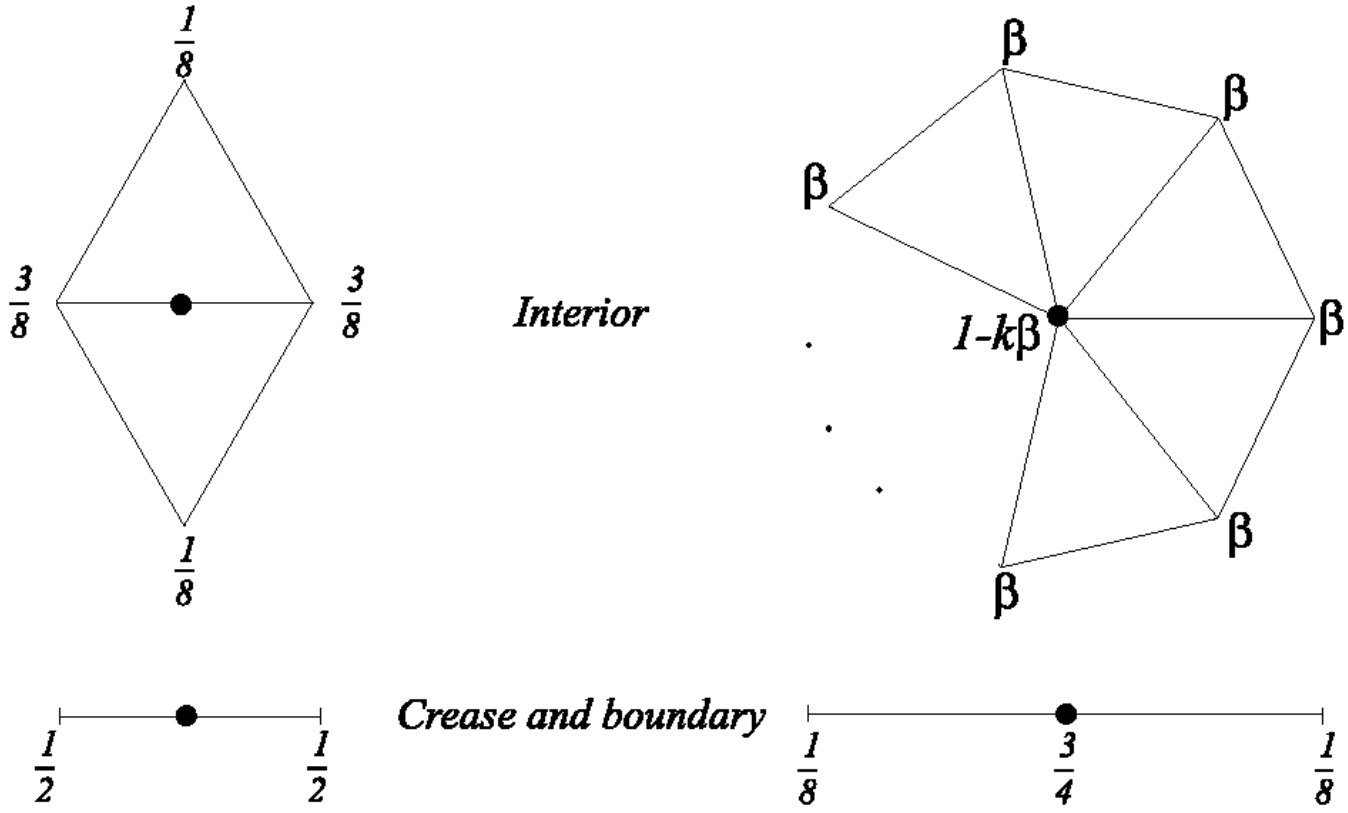
- **How Position New Vertices?**
  - Choose locations for new vertices as weighted average of original vertices in local neighborhood



Loop Subdivision Scheme

CGVR

- **Different Rules for Boundaries:**



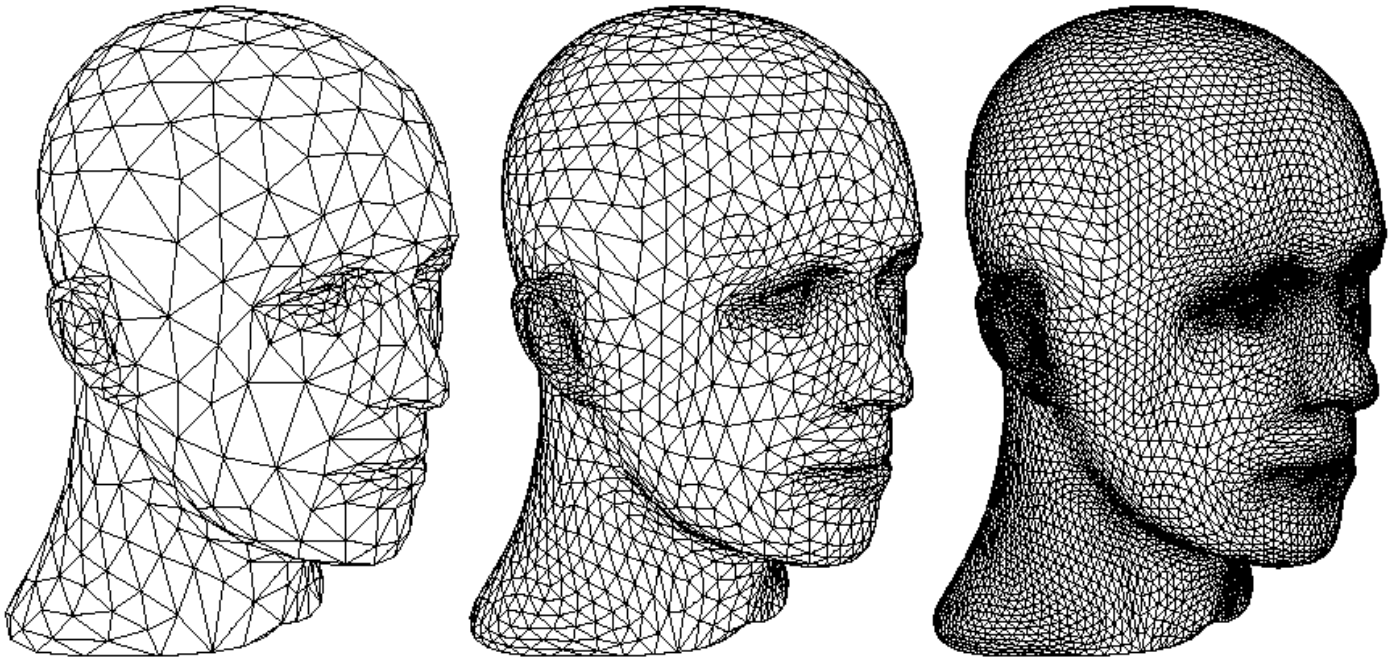
a. *Masks for odd vertices*

b. *Masks for even vertices*

Loop Subdivision Scheme

CGVR

- **Limit Surface Has Provable Smoothness Properties**



Subdivision Schemes

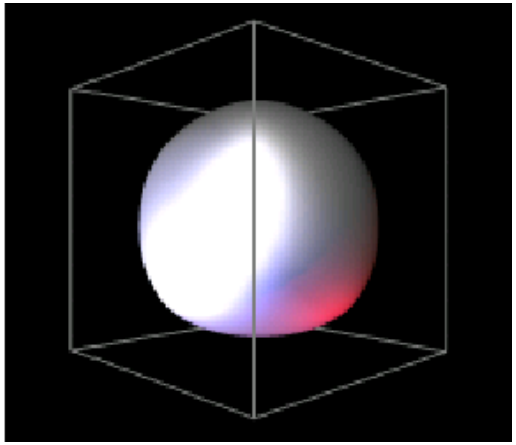
CGVR

- **There are different subdivision schemes**
  - Different methods for refining topology
  - Different rules for positioning vertices
    - Interpolating versus approximating

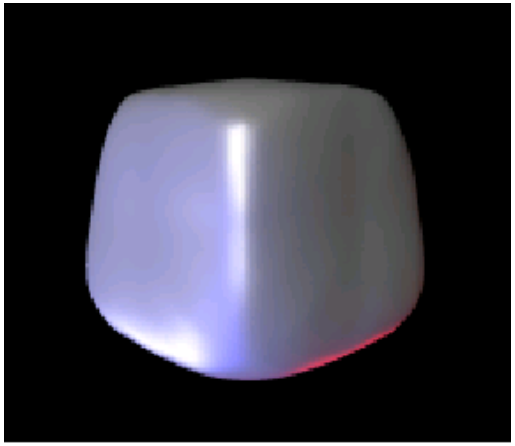
Face Split			Vertex Split
	Triangular Meshes	*Quad. Meshes*	Doo-Sabin, Midege (C^1)
Approximating	Loop (C^2)	Catmull-Clark (C^2)	Biquartic (C^2)
Interpolating	Mod. Butterfly (C^1)	Kobbelt (C^1)	

Subdivision Schemes

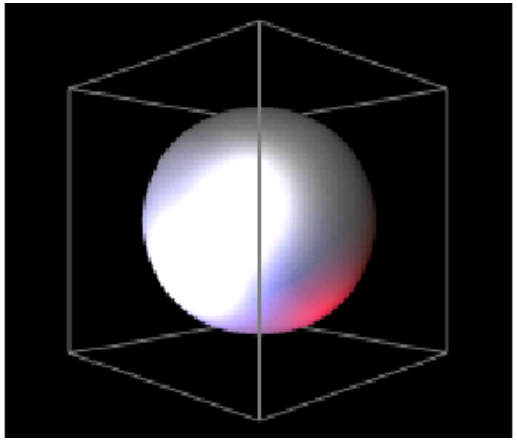
CGVR



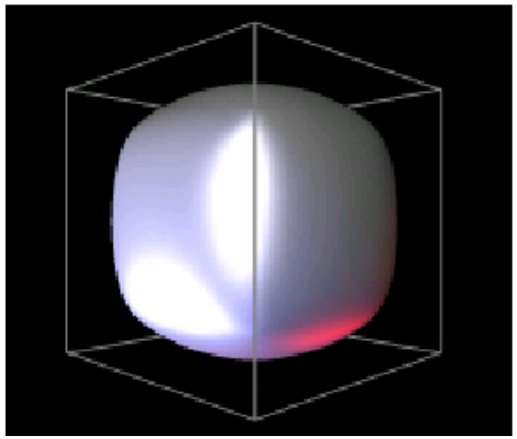
Loop



Butterfly



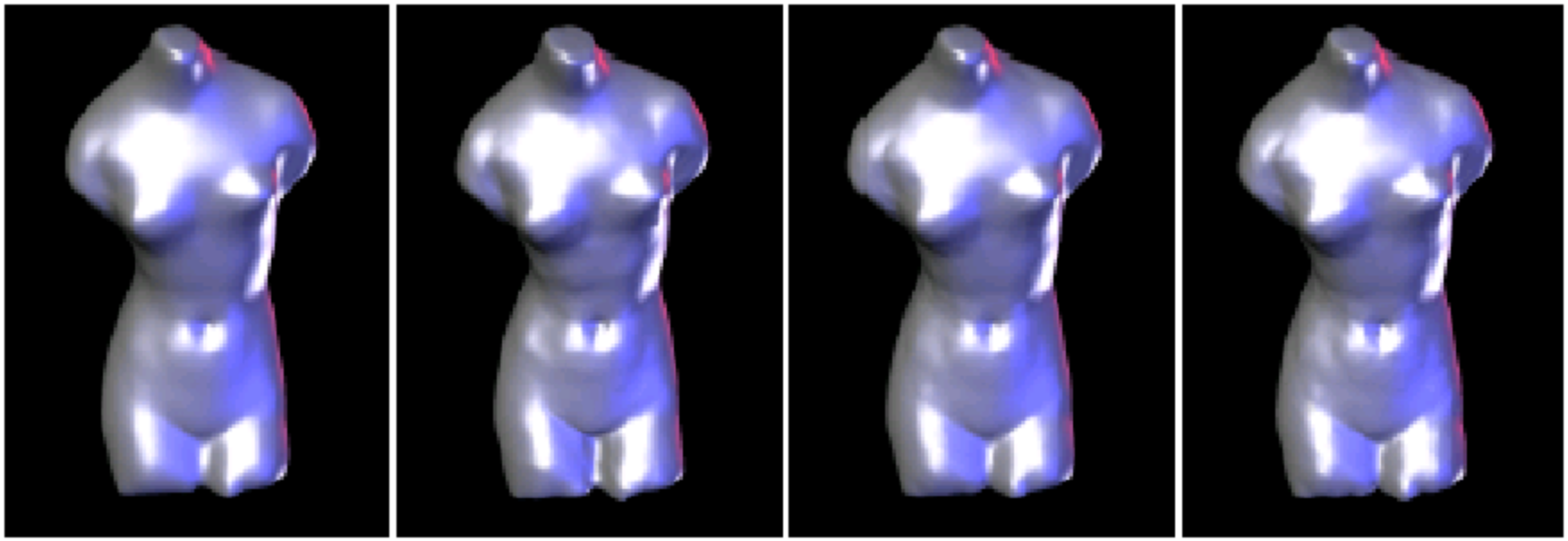
Catmull-Clark



Doo-Sabin

Subdivision Schemes

CGVR



Loop Butterfly Catmull-Clark Doo-Sabin

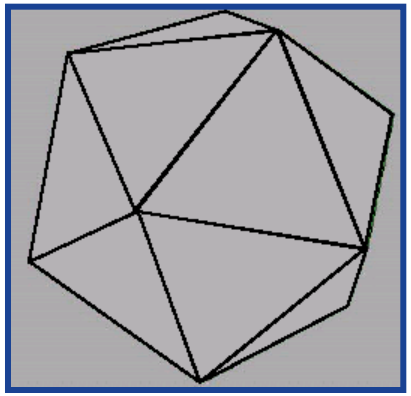
Rendering

CGVR

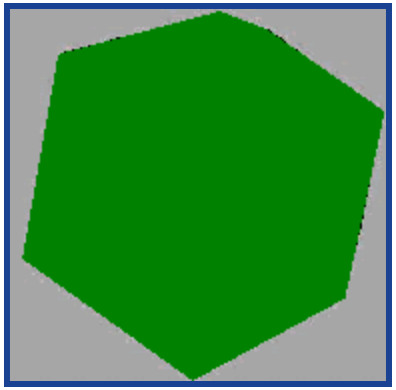
- Illumination Equation
  - Diffuse reflection
  - Specular reflection
  - Emission
  - Ambient

Illumination

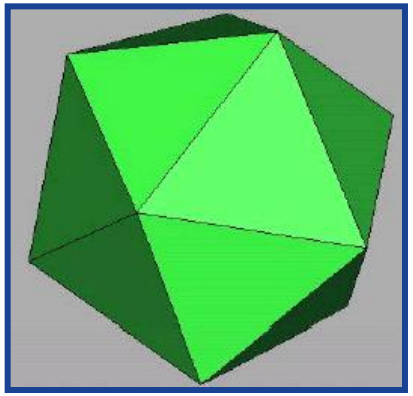
- How do We Compute Radiance for a Sample Ray?
  - Must derive computer models for ...
    - Emission at light sources
    - Scattering at surfaces
    - Reception at the camera



Wireframe



Without Illumination



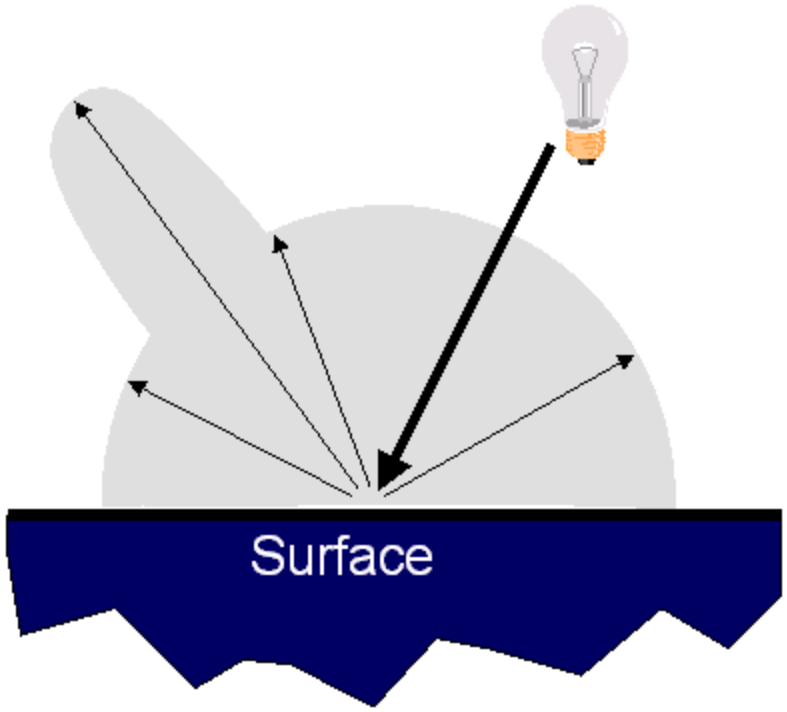
Direct Illumination

Reflectance Model

CGVR

- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”

Based on model proposed by Phong

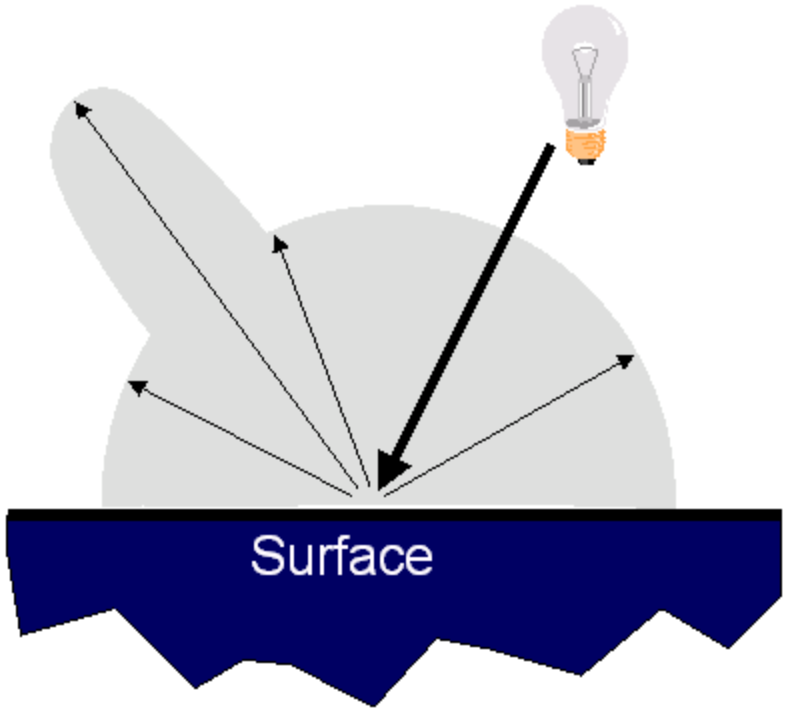


Reflectance Model

CGVR

- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”

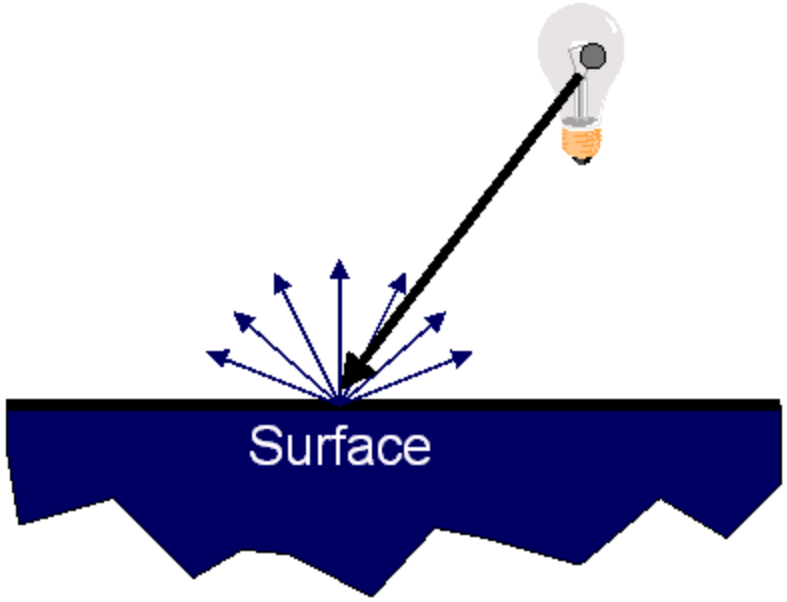
Based on model proposed by Phong



Diffuse Reflection

CGVR

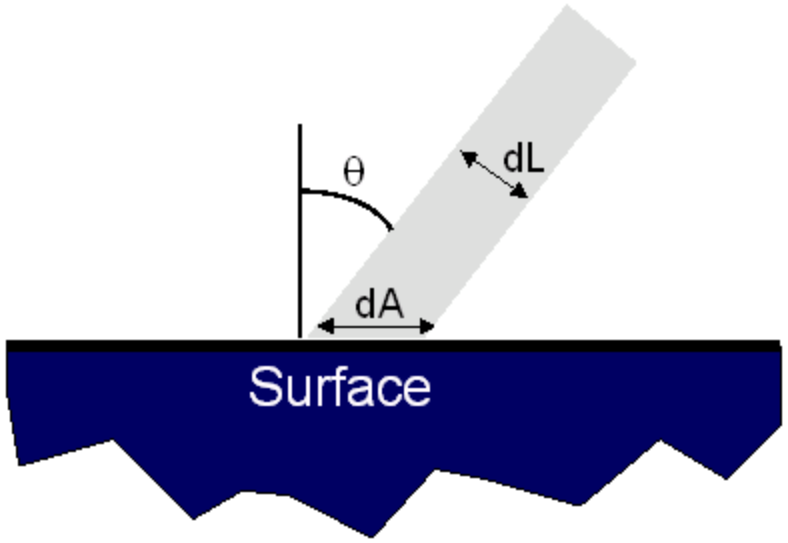
- Assume Surface Reflects Equally in All Directions
  - Examples: chalk, clay



Diffuse Reflection

CGVR

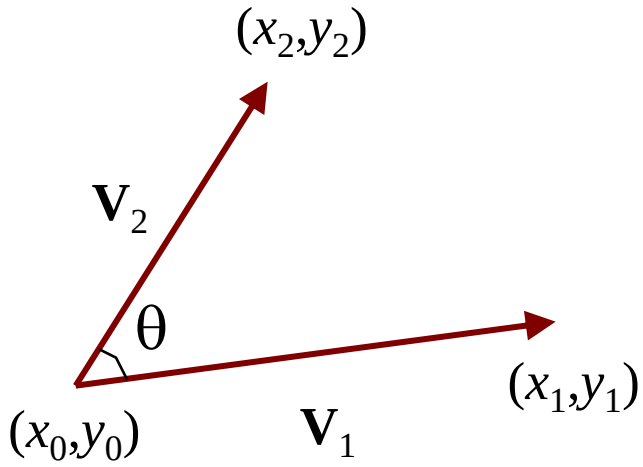
- How Much Light is Reflected?
  - Depends on angle of incident light
  - $dL = dA \cos \Theta$



Vectors

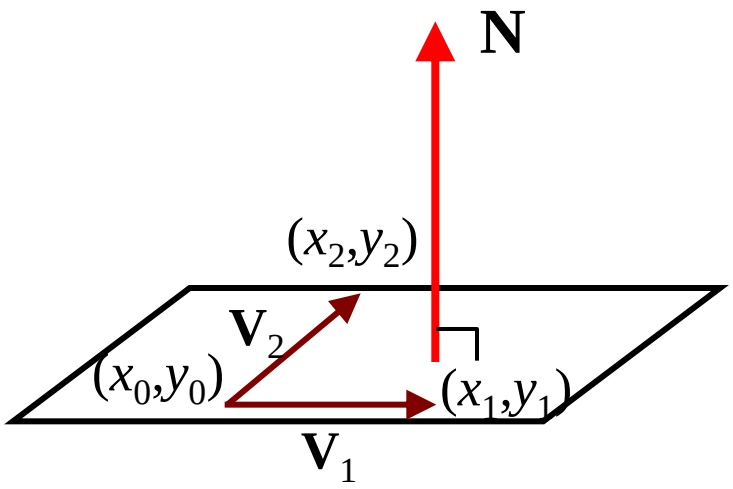
CGVR

- Scalar Product



Angle between Two Edges

- Vector Product

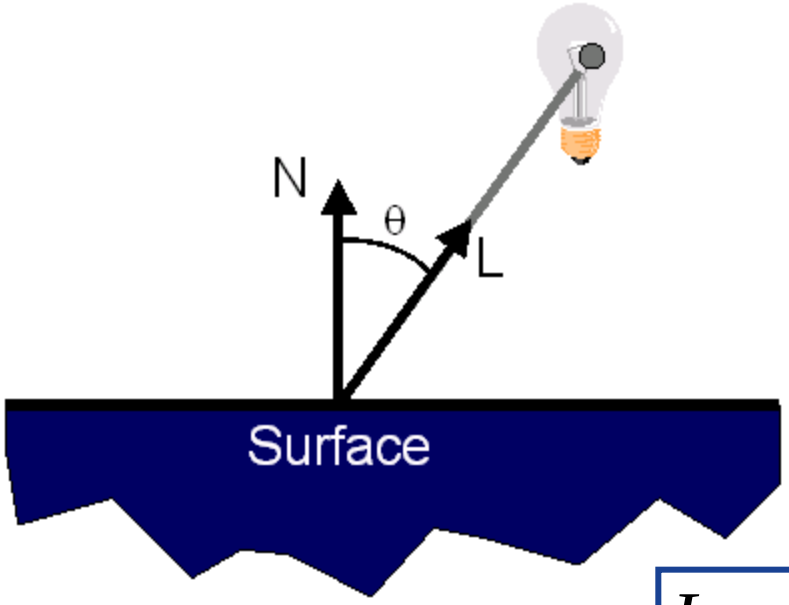


Normal Vector of the Plane

Diffuse Reflection

CGVR

- Lambertian Model
  - Cosine law (dot product)



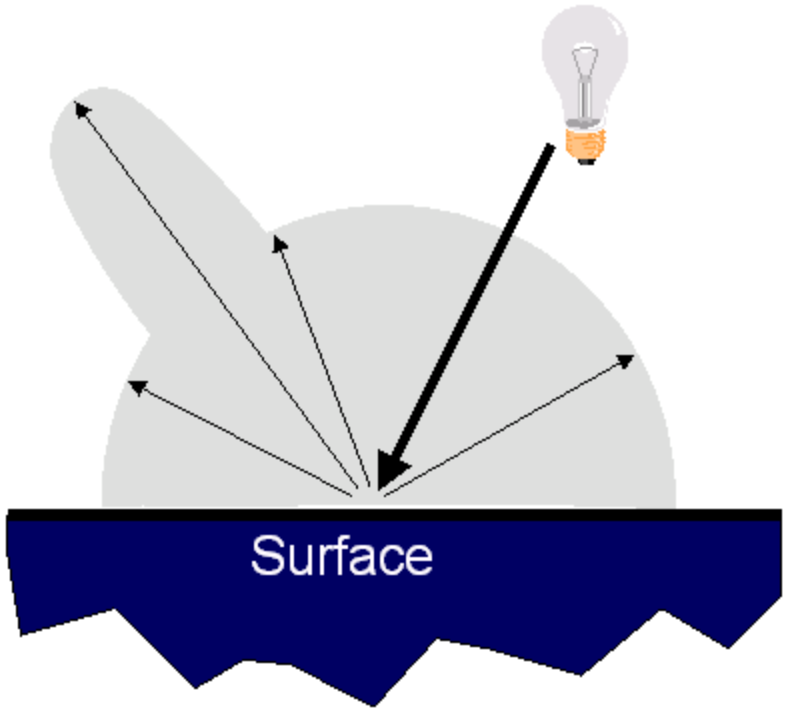
$$I_D = K_D(\mathrm{N}\cdot\mathrm{L})I_L$$

Reflectance Model

CGVR

- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”

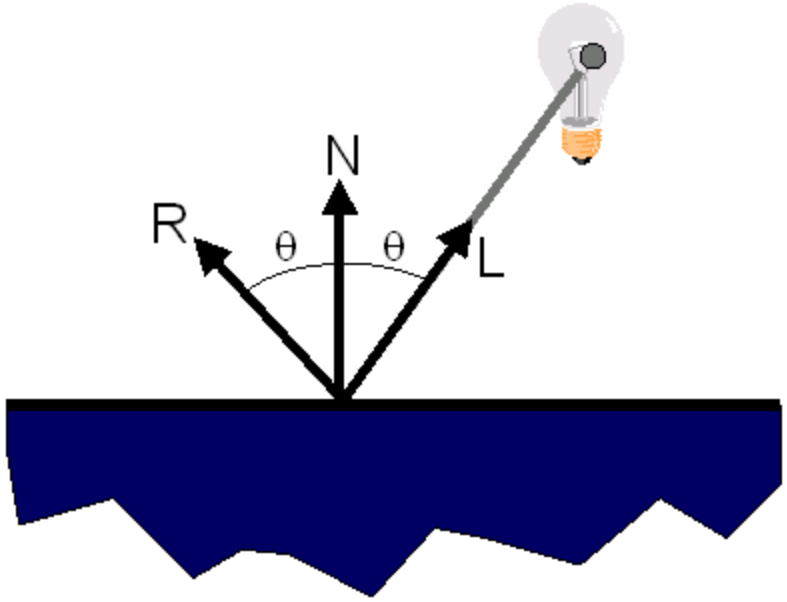
Based on model proposed by Phong



Specular Reflection

CGVR

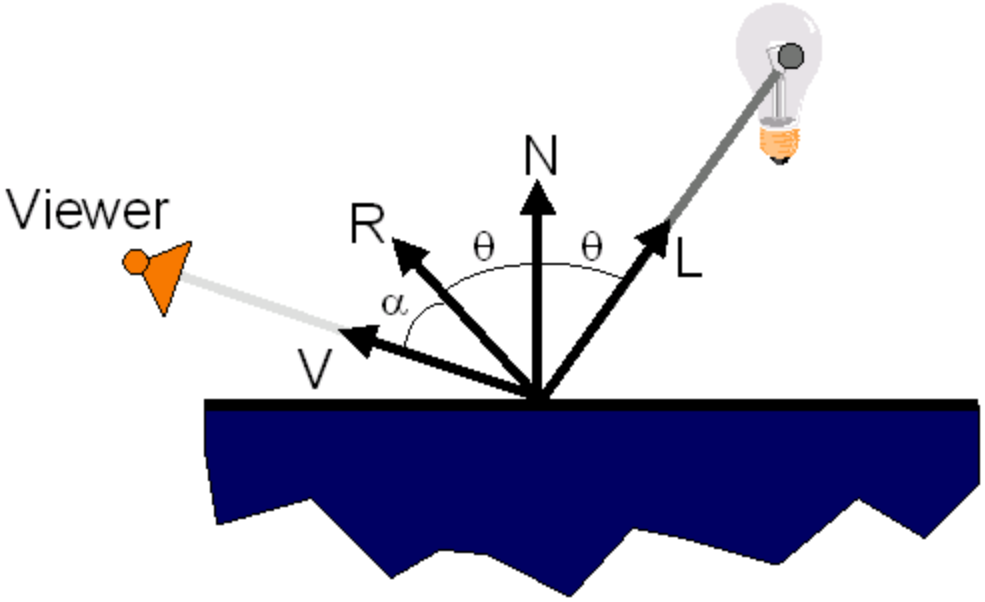
- Reflection is Strongest Near Mirror Angle
  - Examples: mirrors, metals



Specular Reflection

CGVR

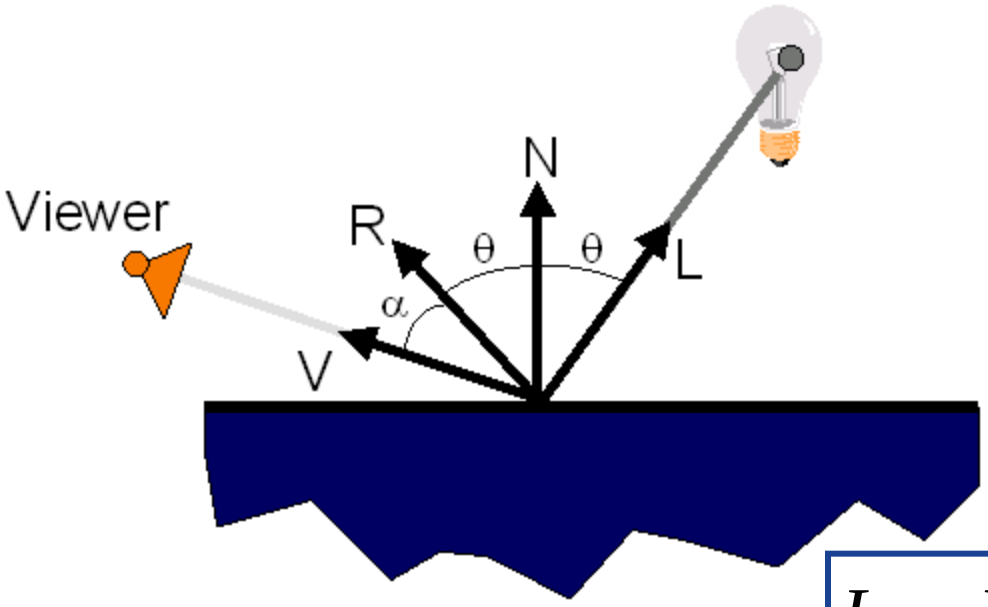
- How Much Light is Seen?
  - Depends on angle of incident light and angle to viewer



Specular Reflection

CGVR

- Phong Model
  - $\{\cos(\alpha)\}^n$



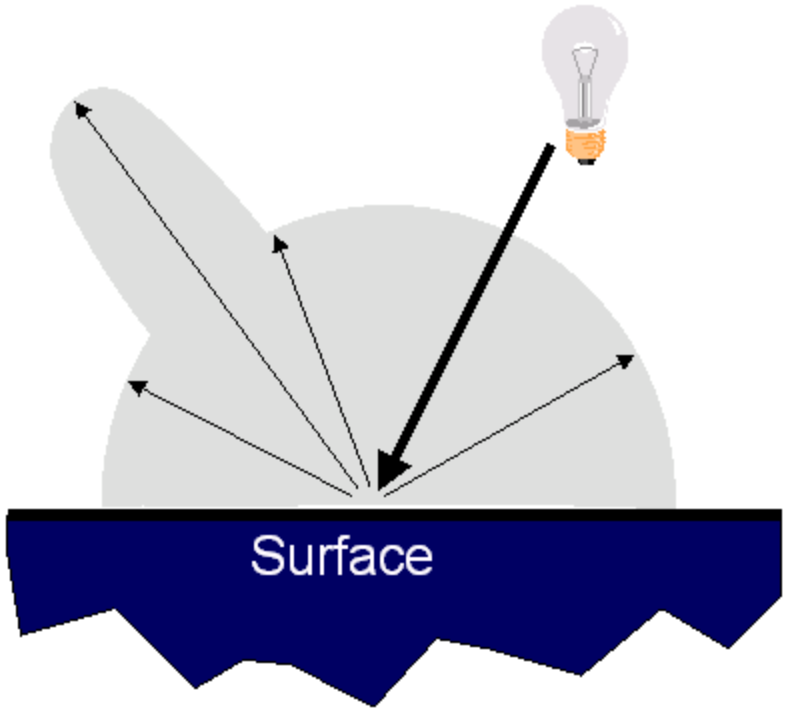
$$I_S = K_S(\mathrm{V}\cdot\mathrm{R})^n I_L$$

Reflectance Model

CGVR

- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”

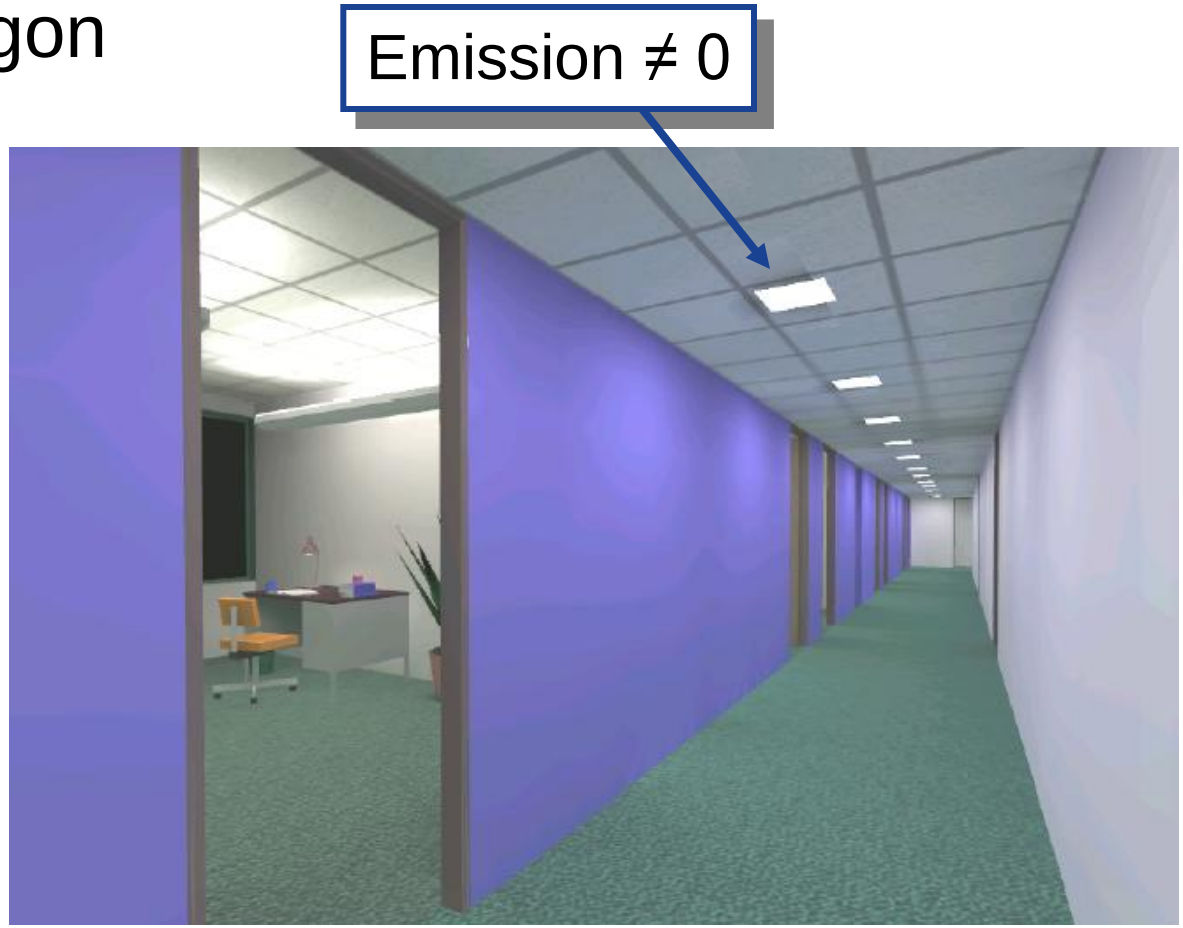
Based on model proposed by Phong



Emission

CGVR

- Represents Light Emitting Directly From Polygon

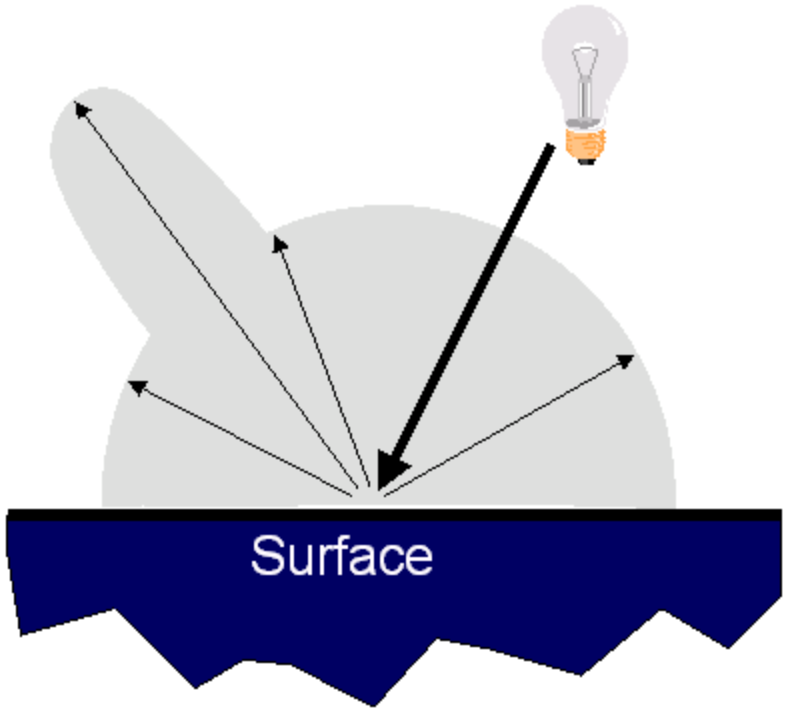


Reflectance Model

CGVR

- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”

Based on model proposed by Phong



Ambient Term

CGVR

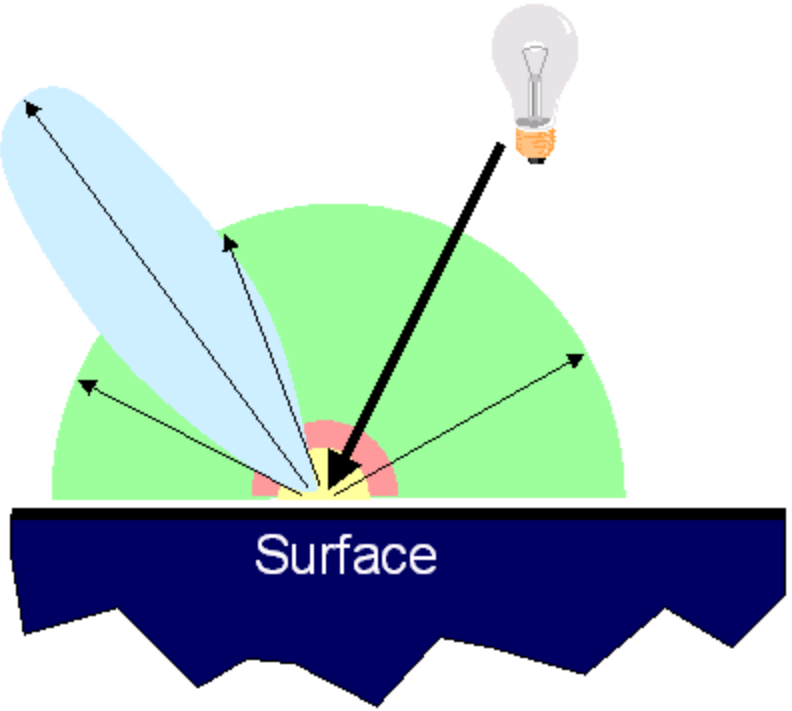
- Represents Reflection of All Indirect Illumination

This is a total hack (avoids complexity of global illumination)!

Reflectance Model

CGVR

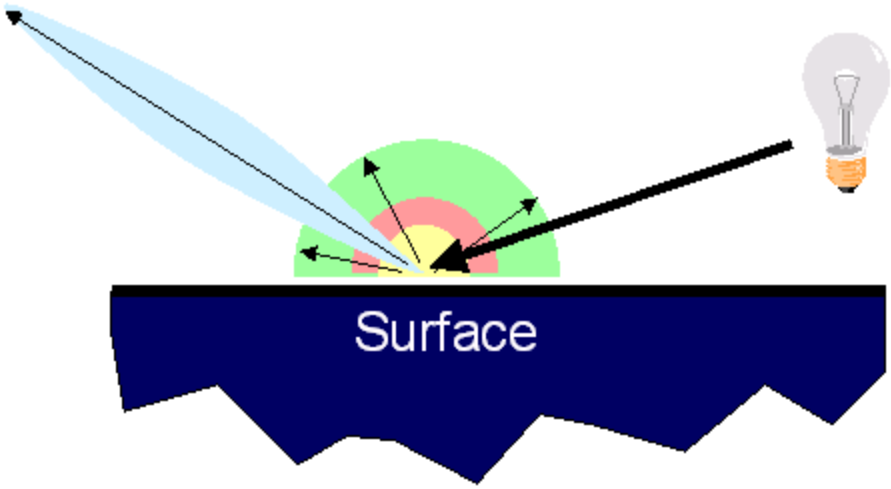
- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”



Reflectance Model

CGVR

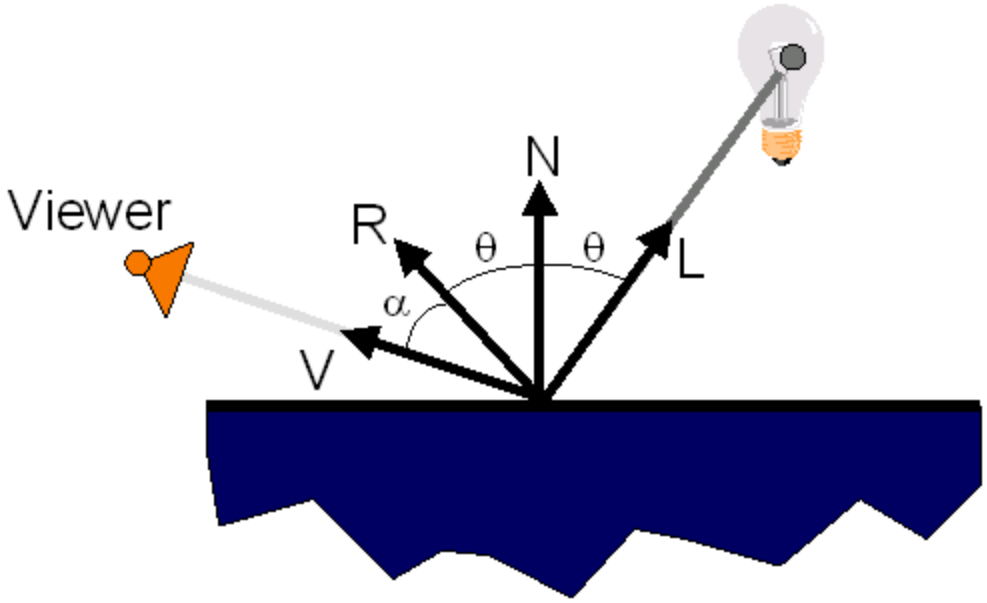
- Simple Analytic Model:
  - Diffuse reflection +
  - Specular reflection +
  - Emission +
  - “Ambient”



Surface Illumination Calculation

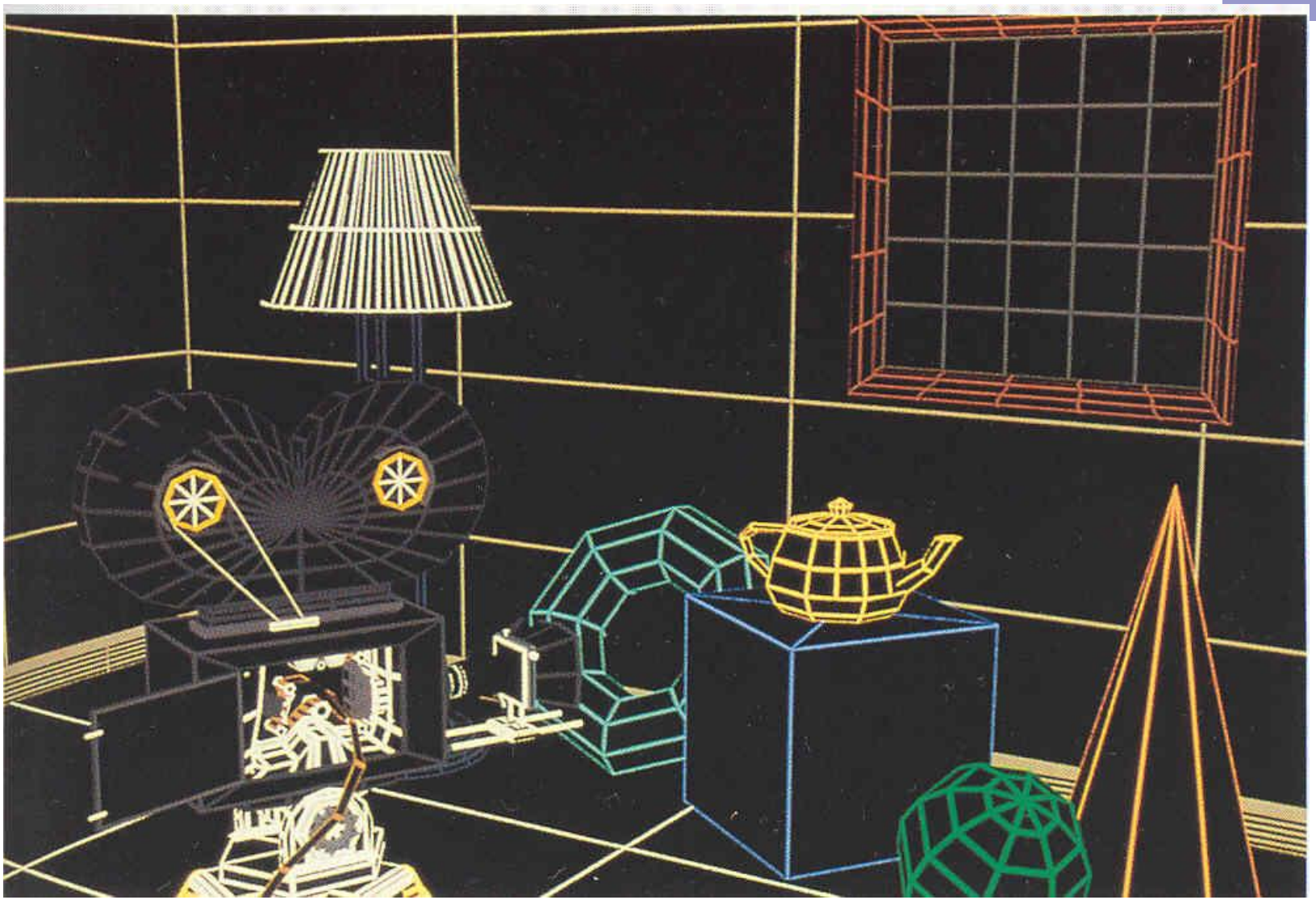
CGVR

- Single Light Source:



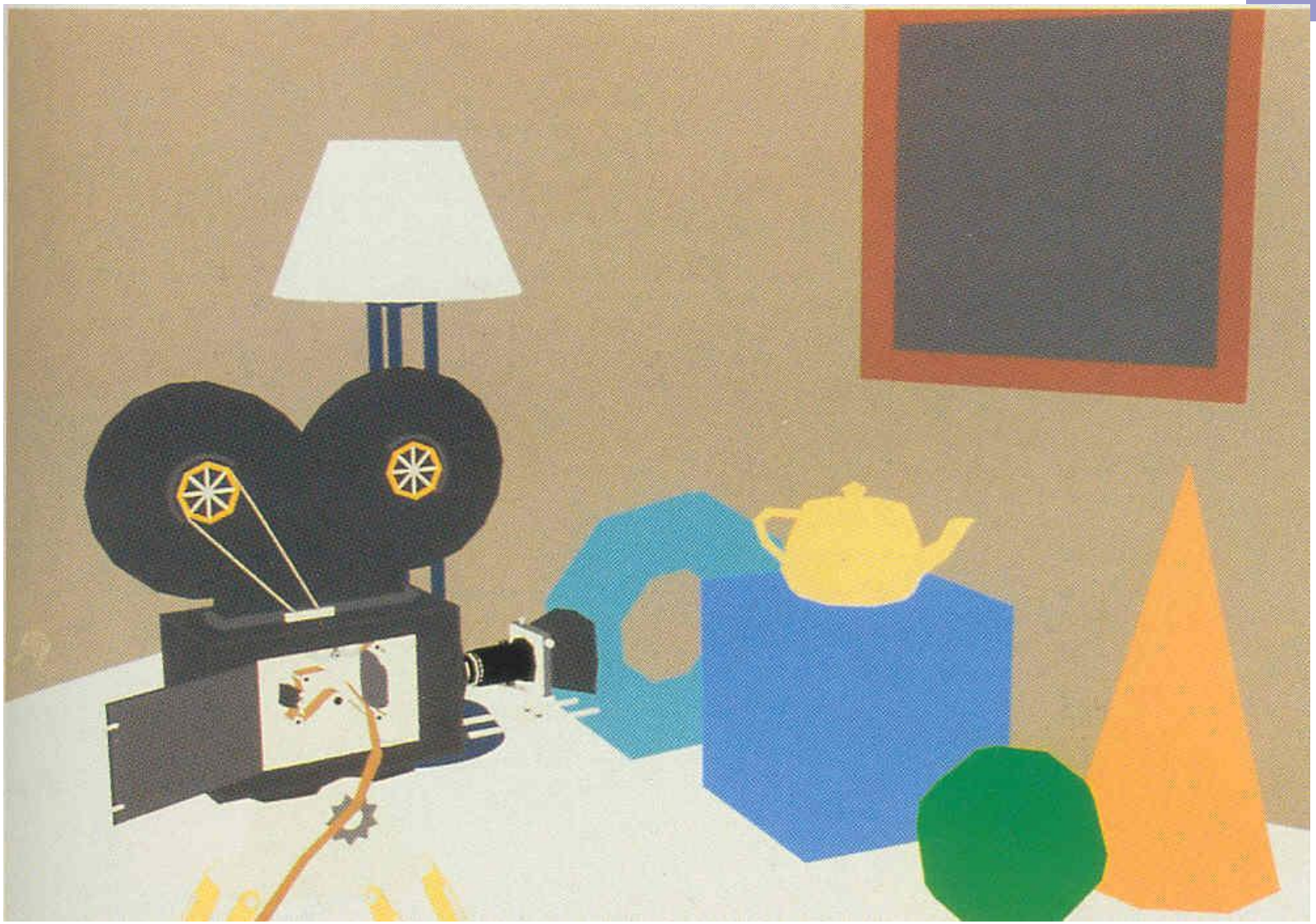
$$I = I_E + K_A I_{AL} + K_D (N \cdot L) I_L + K_S (\mathrm{V} \cdot \mathrm{R})^n I_L$$

Wireframe



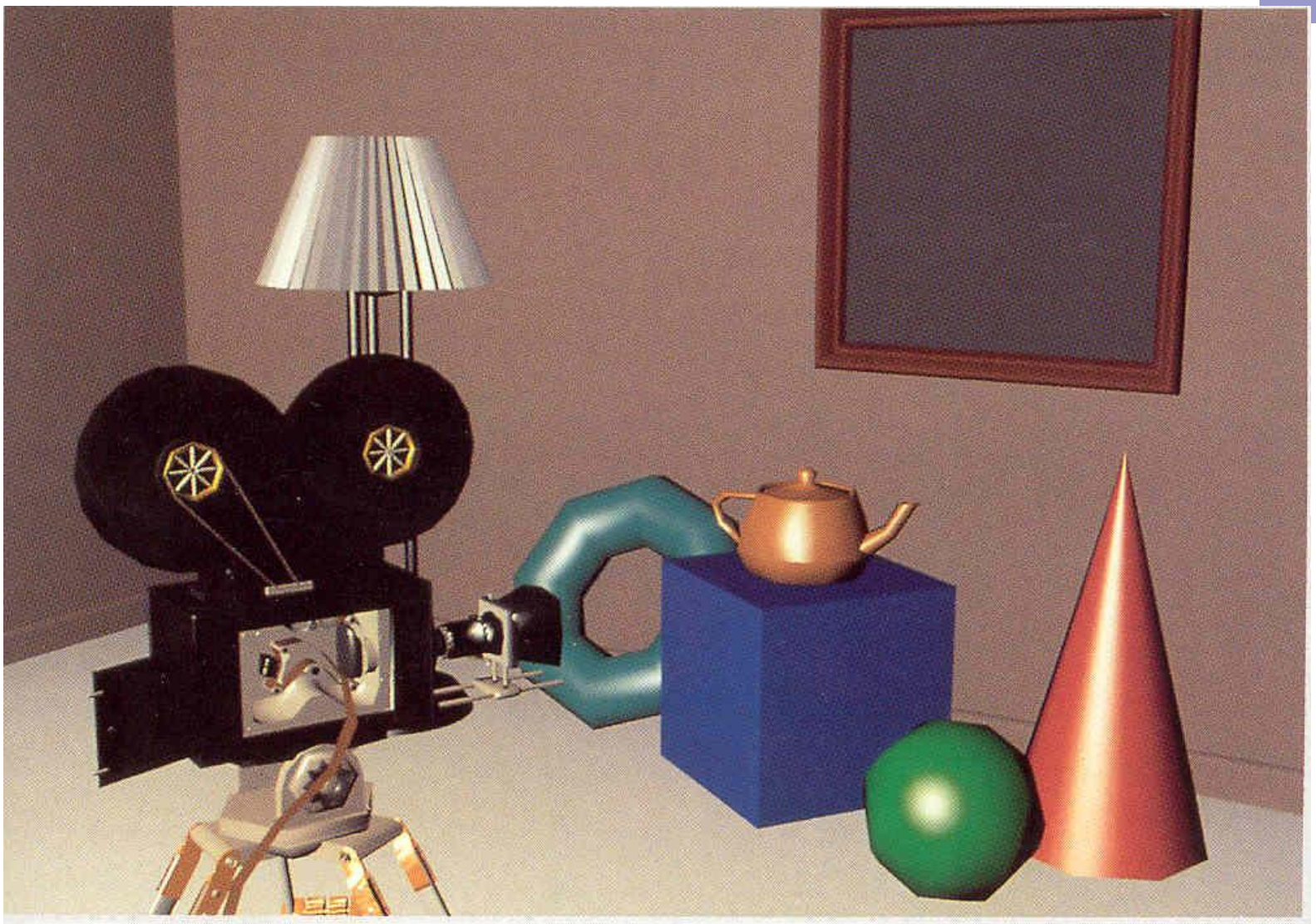
Without Illumination

CGVR



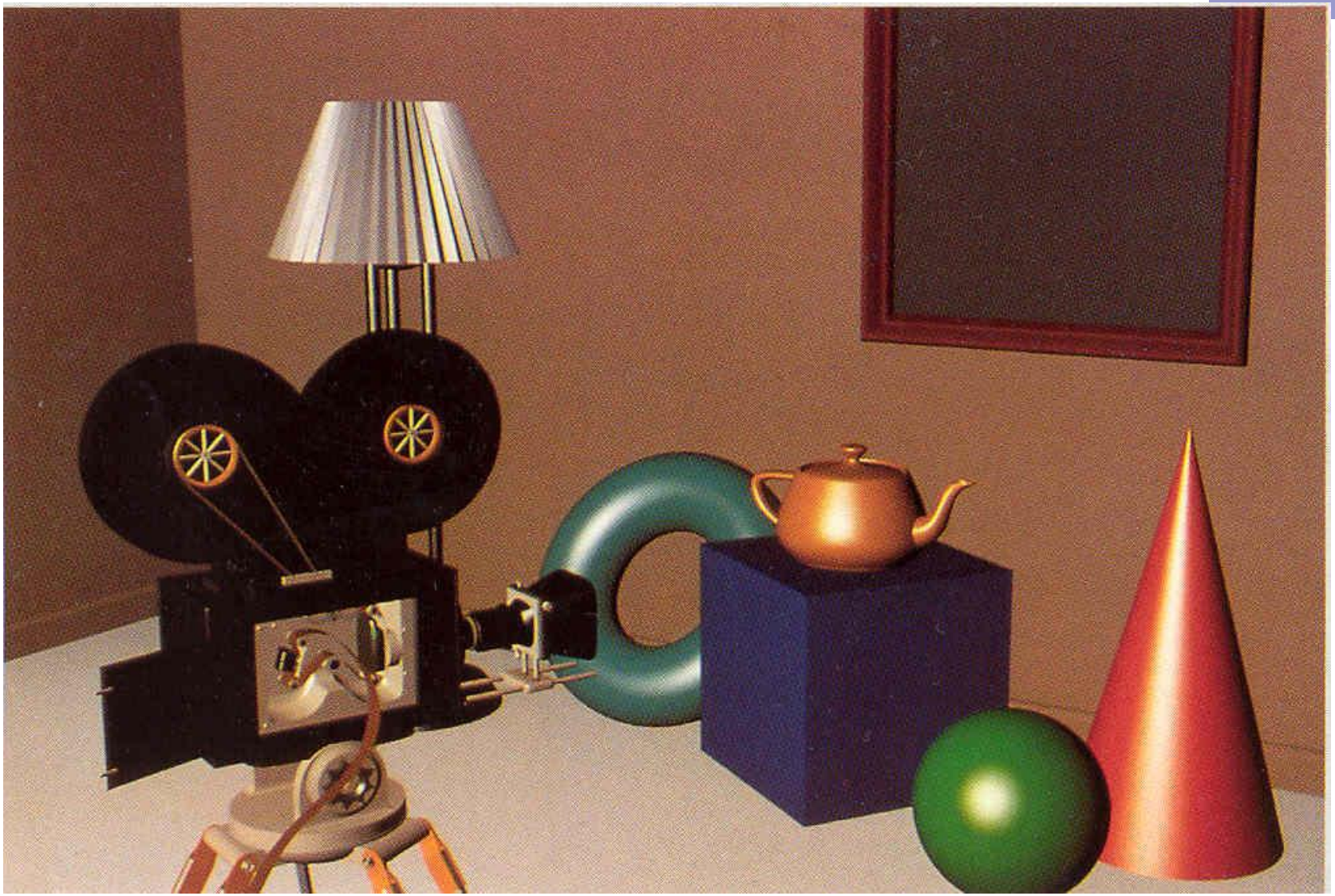
Phong Shading with Polygons

CGVR



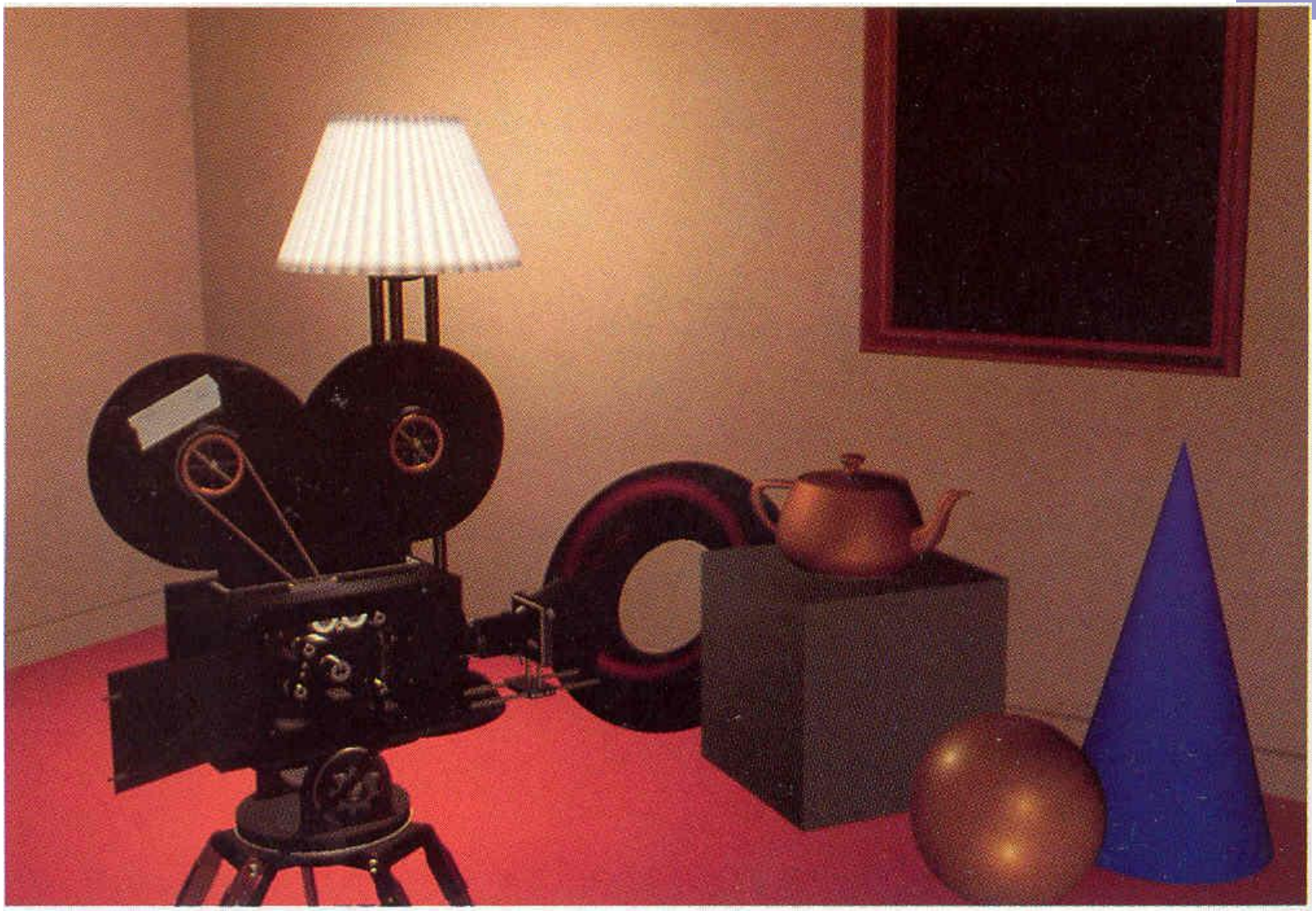
Phong Shading with Curved Surfaces

CGVR



Improved illumination model and multiple lights

CGVR

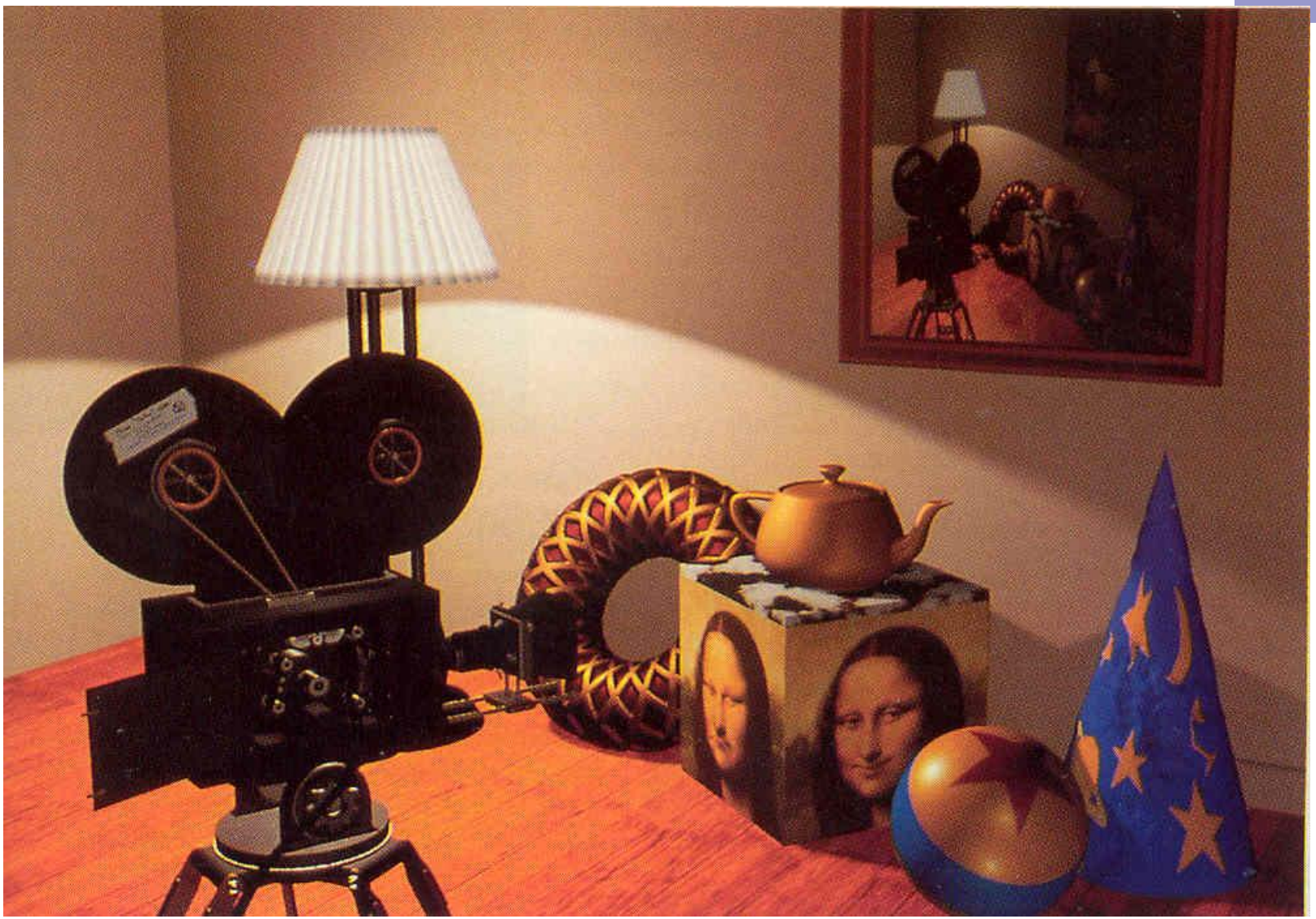


Texture mapping

CGVR

Displacement mapping

CGVR



Animation

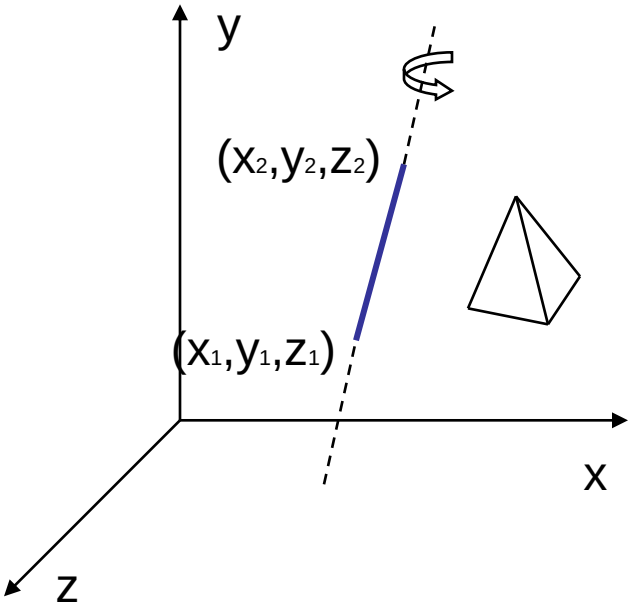
CGVR

- 3D Rotation
  - Euler versus Quaternion
- Keyframe Animation
- Dynamics
  - Space time constraints
  - Retargeting

General 3D Rotations

CGVR

- Rotation about an Arbitrary Axis



T

R

R^{-1}

T^{-1}

Basic Idea

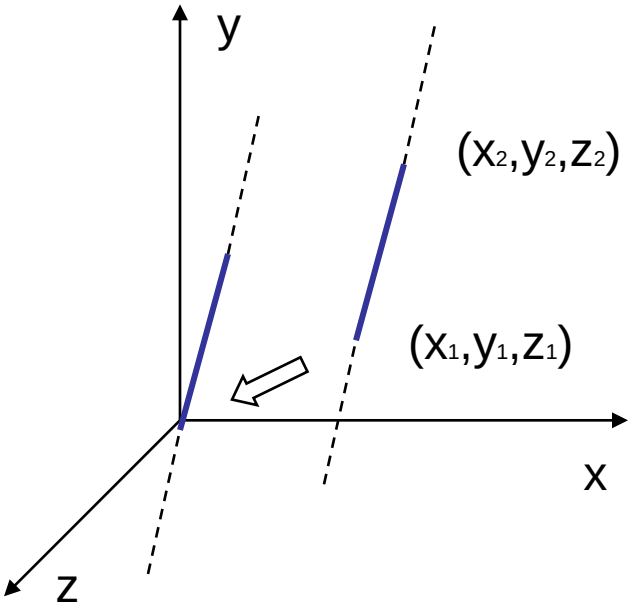
1. Translate (x1, y1, z1) to the origin
2. Rotate (x'2, y'2, z'2) on to the z axis
3. Rotate the object around the z-axis
4. Rotate the axis to the original orientation
5. Translate the rotation axis to the original position

$$[T_R]_{ARB} = [T_{TR}]^{-1}[T_R]_x^{-\alpha}[T_R]_y^{-\phi}[T_R]_z^{\theta}[T_R]_y^{\phi}[T_R]_x^{\alpha}[T_{TR}]$$

General 3D Rotations

CGVR

- Step 1. Translation

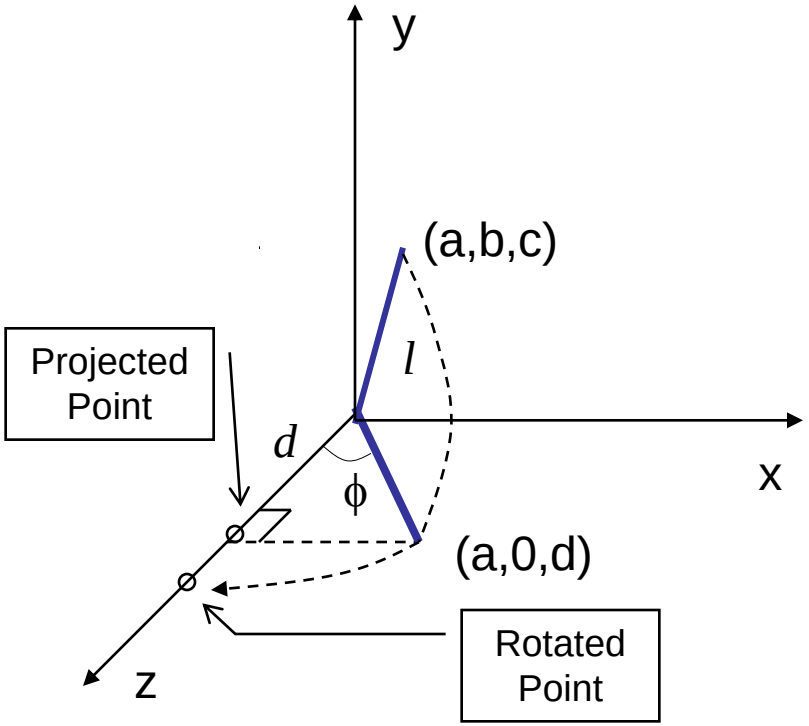


$$T_{TR} = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Arbitrary Axis Rotation

CGVR

- Step 3. Rotate about y axis by ϕ



$$\sin\phi = \frac{a}{l}, \qquad \cos\phi = \frac{d}{l}$$

$$l^2 = a^2 + b^2 + c^2 = a^2 + d^2$$

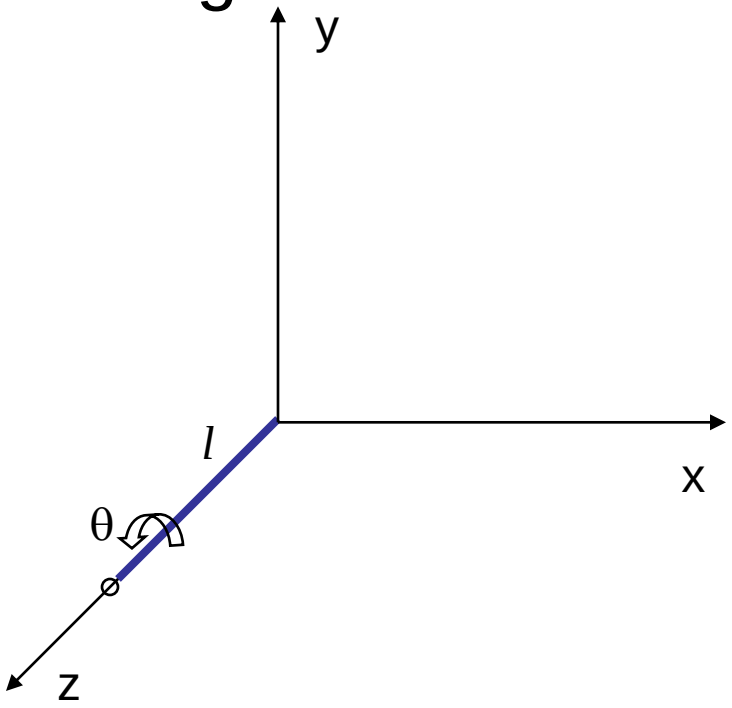
$$d = \sqrt{b^2 + c^2}$$

$$[T_R]_y^{\phi} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d/l & 0 & -a/l & 0 \\ 0 & 1 & 0 & 0 \\ a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Arbitrary Axis Rotation

CGVR

- Step 4. Rotate about z axis by the desired angle θ

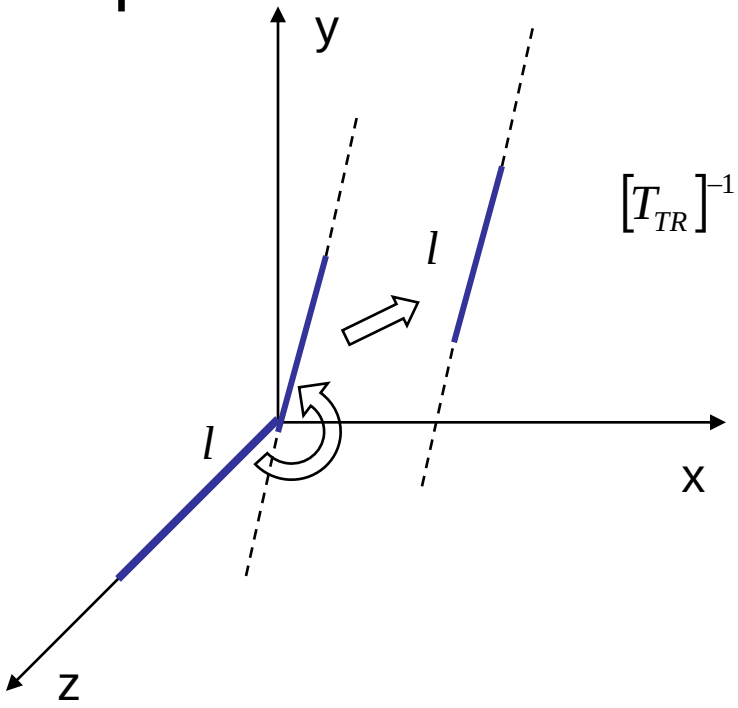


$$[T_R]_z^{\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Arbitrary Axis Rotation

CGVR

- Step 5. Apply the reverse transformation to place the axis back in its initial position



$$[T_{TR}]^{-1}[T_R]_x^{-\alpha}[T_R]_y^{-\phi} = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T_R]_{ARB} = [T_{TR}]^{-1}[T_R]_x^{-\alpha}[T_R]_y^{-\phi}[T_R]_z^{\theta}[T_R]_y^{\phi}[T_R]_x^{\alpha}[T_{TR}]$$

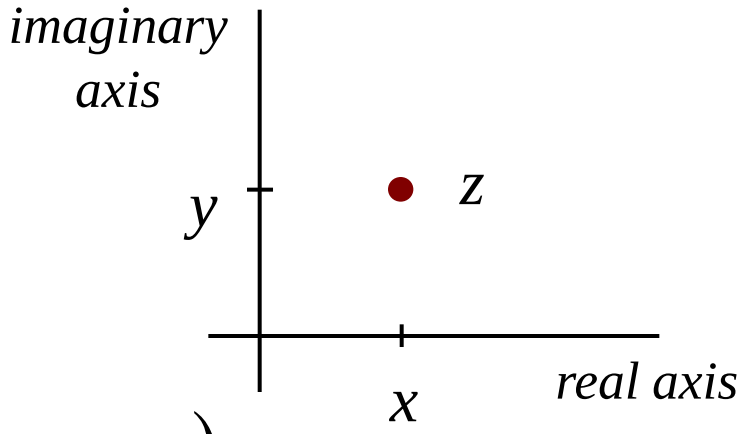
Complex Numbers

CGVR

- Real Part + Imaginary Part: $z = x + iy$

$$z = (x, y)$$

$$x = \mathrm{Re}(z), \quad y = \mathrm{Im}(z)$$



- Addition and Subtraction

$$(x_1, y_1) \pm (x_2, y_2) = (x_1 \pm x_2, y_1 \pm y_2)$$

- Scalar Multiplication

$$k(x_1, y_1) = (kx_1, ky_1)$$

- Multiplication

$$(x_1, y_1)(x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1)$$

Pure Imaginary Number & Complex Conjugate

CGVR

- Imaginary Unit: $i = (0,1)$

$$i^2 = (0,1)(0,1) = (-1,0)$$

$$i = \sqrt{-1}$$

- Complex Conjugate

$$z = x + iy \qquad \bar{z} = x - iy$$

  - Modulus or absolute value

$$|z| = z\bar{z} = \sqrt{x^2 + y^2}$$

- Division

$$\frac{z_1}{z_2} = \frac{z_1\bar{z}_2}{z_2\bar{z}_2} = \frac{(x_1, y_1)(x_2, -y_2)}{x_2^2 + y_2^2} = \left(\frac{x_1x_2 + y_1y_2}{x_2^2 + y_2^2}, \frac{x_2y_1 - x_1y_2}{x_2^2 + y_2^2} \right)$$

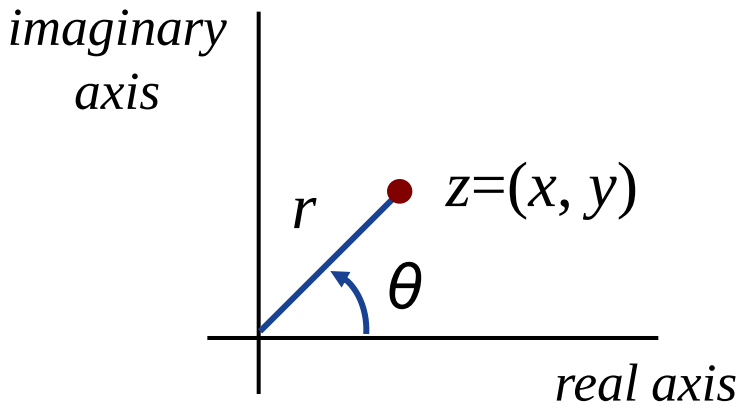
Representation with Polar Coordinates

CGVR

- $z = r(\cos\theta + i\sin\theta)$
- Euler's Formula

$$e^{i\theta} = \cos\theta + i\sin\theta$$

$$z = re^{i\theta}$$



- Complex Multiplication and Division

$$z_1 z_2 = r_1 r_2 e^{i(\theta_1+\theta_2)}, \quad \frac{z_1}{z_2} = \frac{r_1}{r_2} e^{i(\theta_1-\theta_2)}$$

- nth Roots

$$\sqrt[n]{z} = \sqrt[n]{r}\left[\cos\left(\frac{\theta+2k\pi}{n}\right) + i\sin\left(\frac{\theta+2k\pi}{n}\right)\right], \quad k = 0, 1, 2, \cdots, n-1$$

Quaternions

CGVR

- One Real Part + Three Imaginary Part

$$q = s + ia + jb + kc$$

- Properties: $i^2 = j^2 = k^2 = -1$

$$ij = -ji = k$$

$$jk = -kj = i$$

$$ki = -ik = j$$

- Addition and Scalar Multiplication

$$q_1 + q_2 = (s_1 + s_2) + i(a_1 + a_2) + j(b_1 + b_2) + k(c_1 + c_2)$$

$$dq_1 = s_1 d + ia_1 d + jb_1 d + kc_1 d$$

Ordered-Pair Notation

CGVR

- Scalar 's' + Vector "$\mathbf{v} = (a, b, c)$"

$$q = (s, \mathbf{v})$$

- Addition: $q_1 + q_2 = (s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2)$
- Multiplication

$$q_1 q_2 = (s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)$$

- Magnitude

$$|q|^2 = s^2 + \mathbf{v} \cdot \mathbf{v}$$

- Inverse

$$q^{-1} = \frac{1}{|q|^2}(s, \mathbf{v}) \quad \Leftarrow \quad qq^{-1} = q^{-1}q = (1, 0)$$

3D Rotation

CGVR

- For a 3D Point (α, β, γ)
  - A unit quaternion $q = (w, x, y, z)$ its conjugate $\bar{q} = (w, -x, -y, -z)$

$$q \cdot (0, \alpha, \beta, \gamma) \cdot \bar{q} = (0, \alpha', \beta', \gamma')$$

➔ ***Rotating (α, β, γ) by angle 2θ about the axis parallel to (a, b, c)***

- For $q = (\cos\theta, \sin\theta(a, b, c))$

  R_q is a 3D Rotation about (a, b, c) by 2θ

$$R_q(p) = q \cdot p \cdot \bar{q}$$

Animation

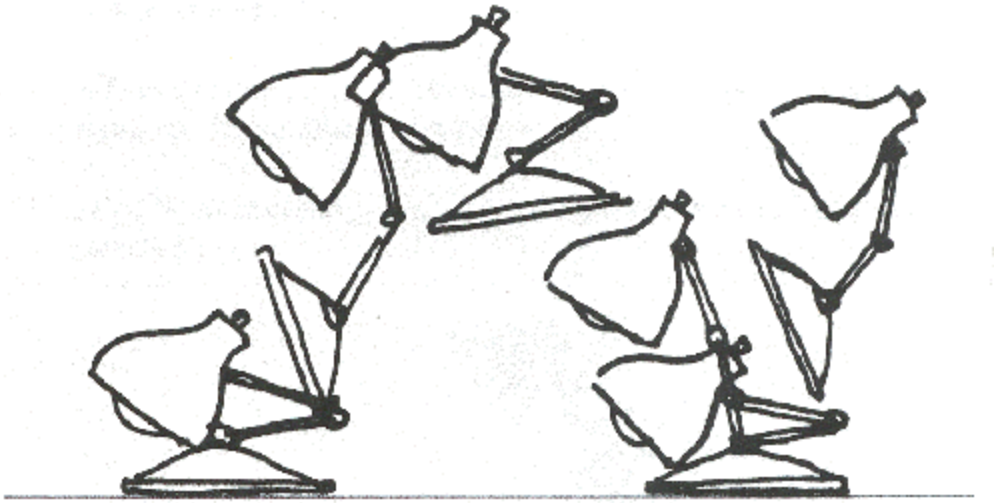
CGVR

- 3D Rotation
  - Euler versus Quaternion
- Keyframe Animation
- Dynamics
  - Space time constraints
  - Retargeting

Keyframe Animation

CGVR

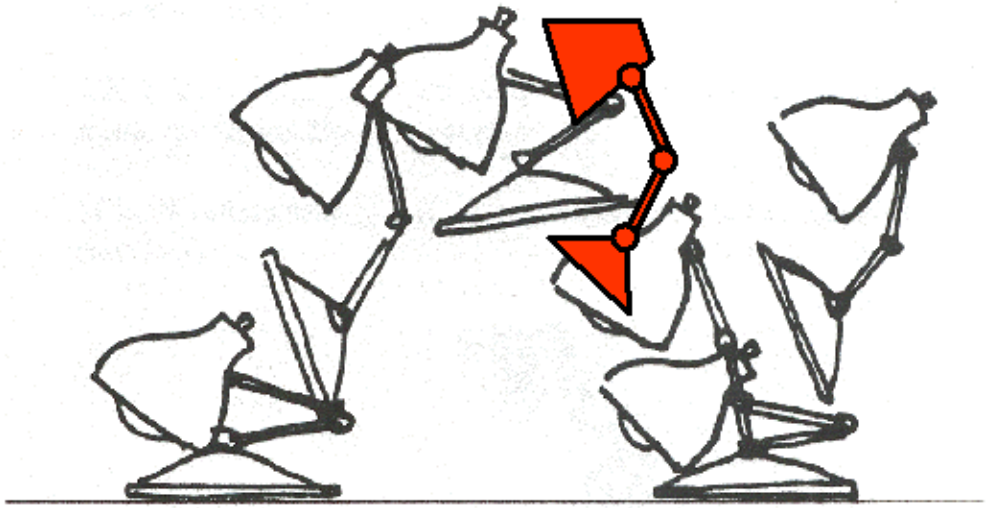
- Define Character Poses at Specific Time Steps Called “Keyframes”



Keyframe Animation

CGVR

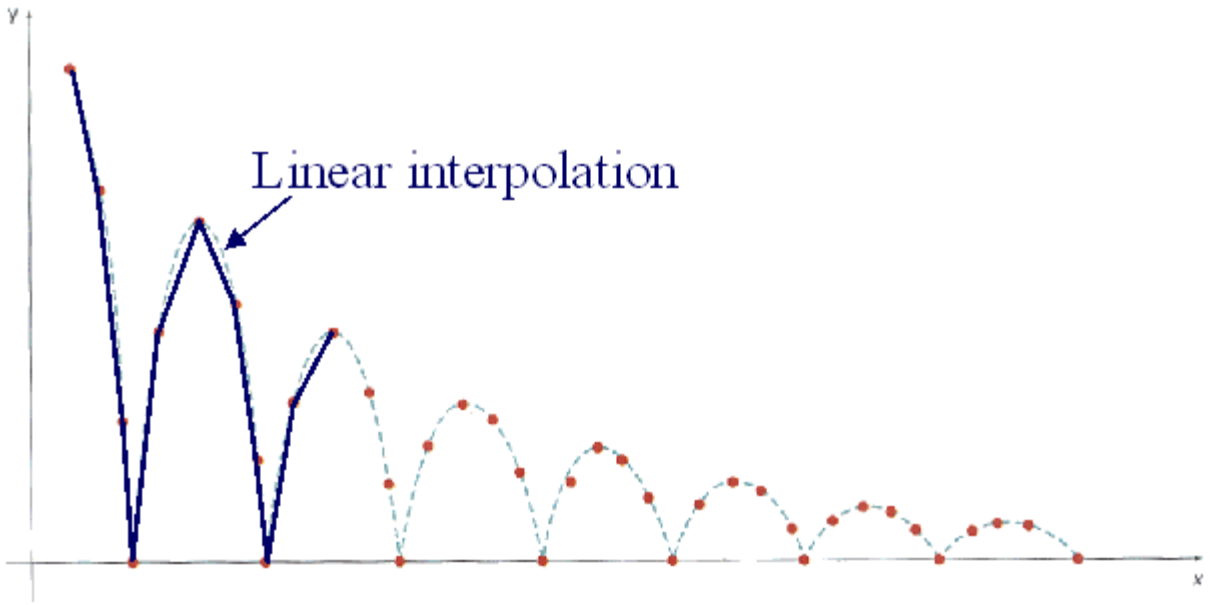
- **Interpolate Variables Describing Keyframes to Determine Poses for Character in between**



Inbetweening

CGVR

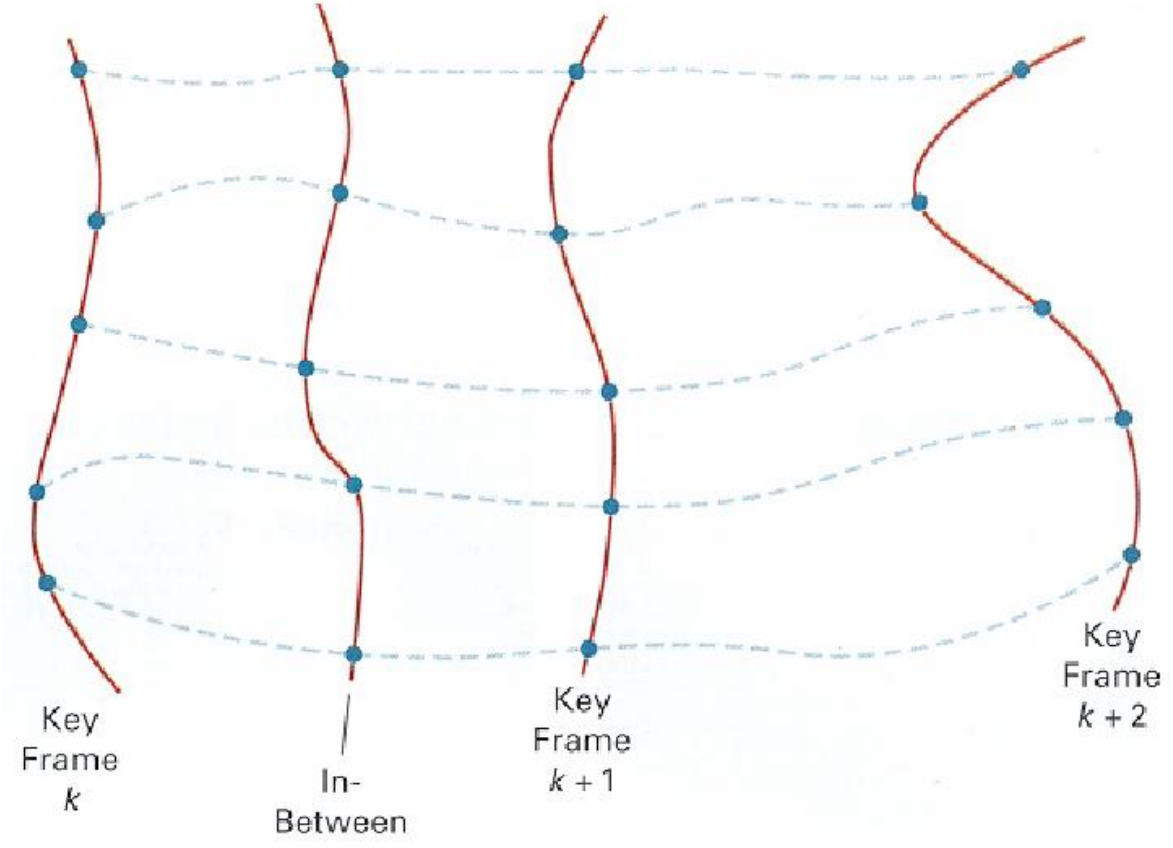
- **Linear Interpolation**
  - Usually not enough continuity



Inbetweening

CGVR

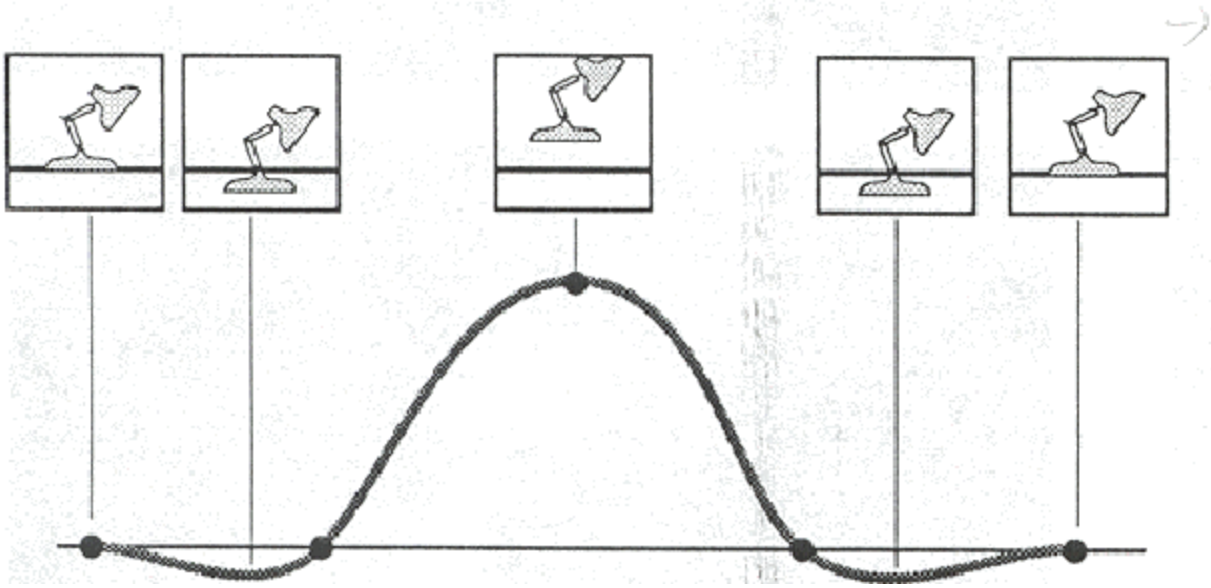
- **Spline Interpolation**
  - Maybe good enough



Inbetweening

CGVR

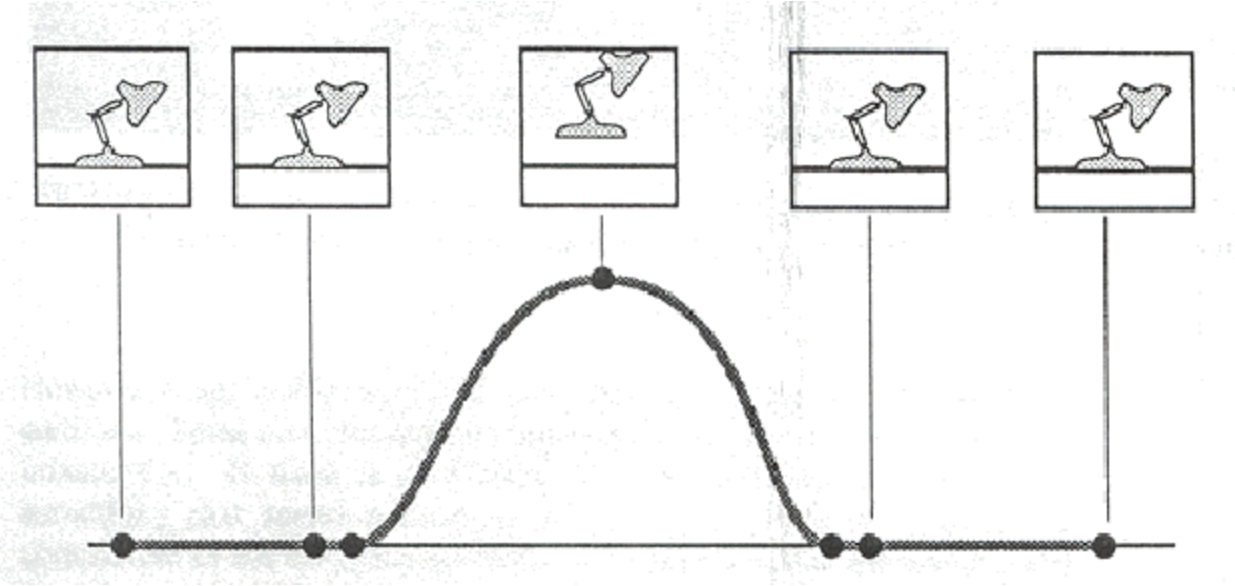
- **Spline Interpolation**
  - Maybe good enough
    - May not follow physical laws



Inbetweening

CGVR

- **Spline Interpolation**
  - Maybe good enough
    - May not follow physical laws



Animation

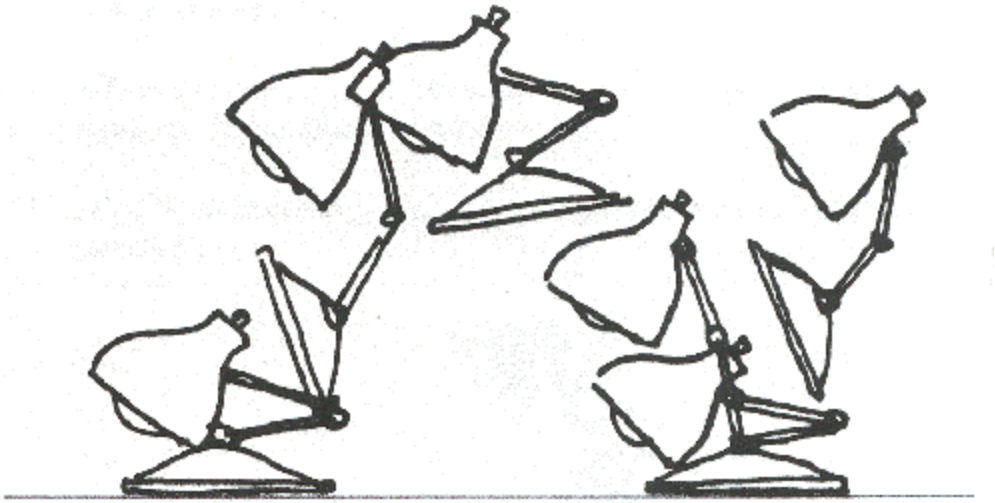
CGVR

- 3D Rotation
  - Euler versus Quaternion
- Keyframe Animation
- Dynamics
  - Space time constraints
  - Retargeting

Dynamics

CGVR

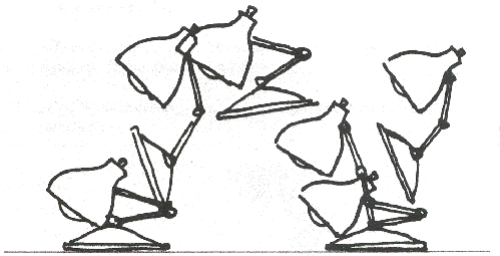
- **Simulation of Physics Insures Realism of Motion**



Space Time Constraints

CGVR

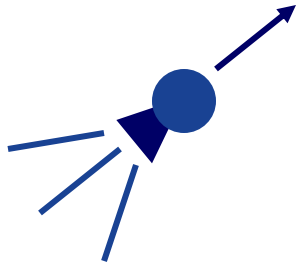
- **Animator Specifies Constraints**
  - What the character's physical structure is
    - e.g., articulated figure
  - What the character has to do
    - e.g., jump from here to there within time t
  - What other physical structures are present
    - e.g., floor to push off and land
  - How the motion should be performed
    - e.g., minimize energy



Space Time Constraints

CGVR

- **Computer Finds the "Best" Physical Motion**
  - Satisfying constraints
- **Example: Particle with Jet Propulsion**
  - $\boldsymbol{x}(t)$ is position of particle at time t
  - $\boldsymbol{f}(t)$ is force of jet propulsion at time t
  - Particle's equation of motion is:



$$mx'' - f - mg = 0$$

  - Suppose we want to move from a to b within t_0 to t_1 with minimum jet fuel:

$$\text{Minimize} \int_{t_0}^{t_1} |f(t)|^2 dt \text{ subject to } x(t_0) = a \text{ and } x(t_1) = b$$

Space Time Constraints

CGVR

- **Discretize Time Steps**

$$x' = \frac{x_i - x_{i-1}}{h}$$

$$x'' = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}$$

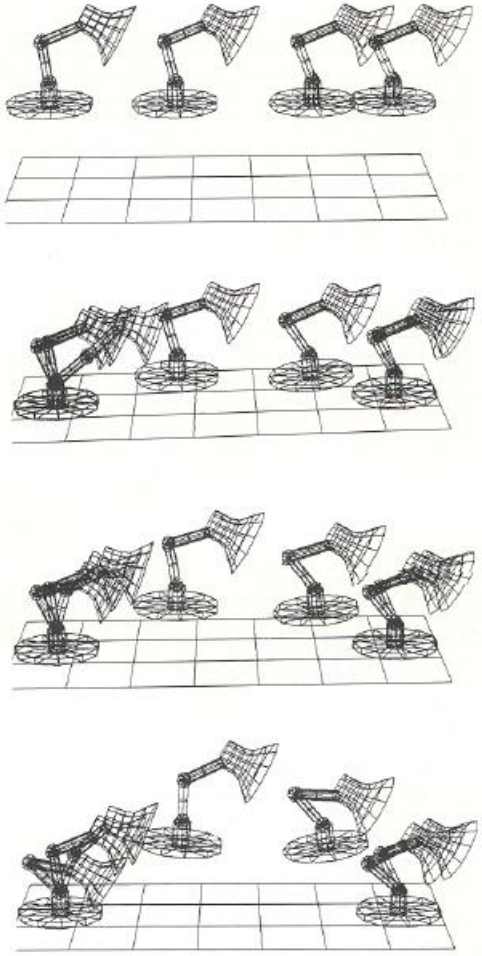
$$m\left(x'' = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} \right) - f_i - mg = 0$$

$$\text{Minimize} \quad h\sum_i \left| f_i \right|^2 \text{ subject to } x_0 = a \text{ and } x_1 = b$$

Space Time Constraints

CGVR

- Solve with Iterative Optimization Methods



Space Time Constraints

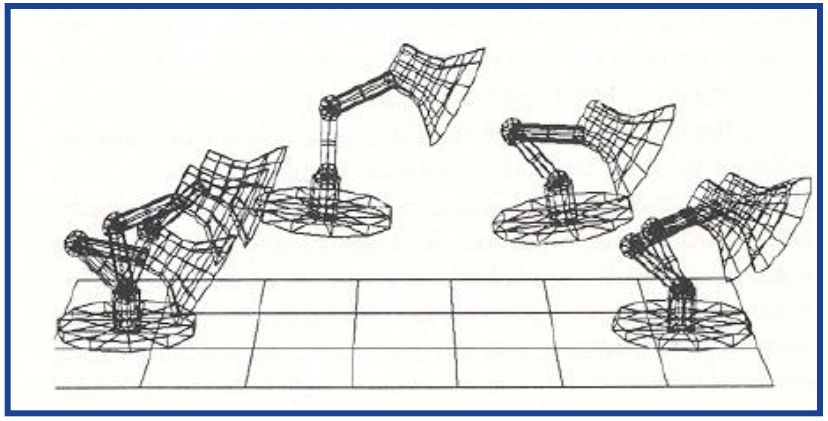
CGVR

- **Advantages**
  - Free animator from having to specify details of physically realistic motion with spline curves
  - Easy to vary motions due to new parameters and/or new constraints
- **Challenges**
  - Specifying constraints and objective functions
  - Avoiding local minima during optimization

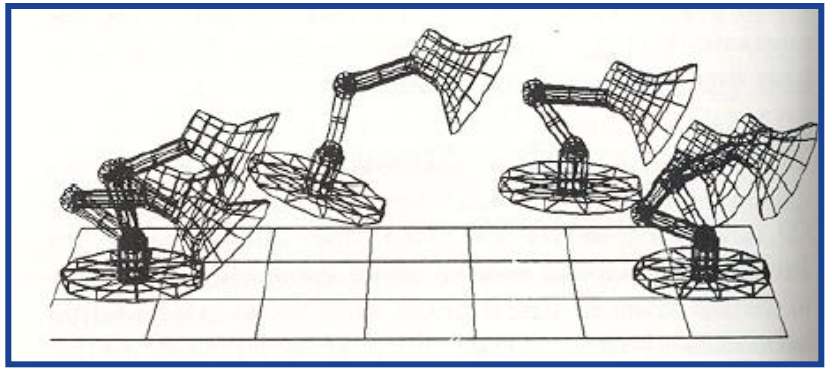
Space Time Constraints

CGVR

- **Adapting Motion**



Original Jump

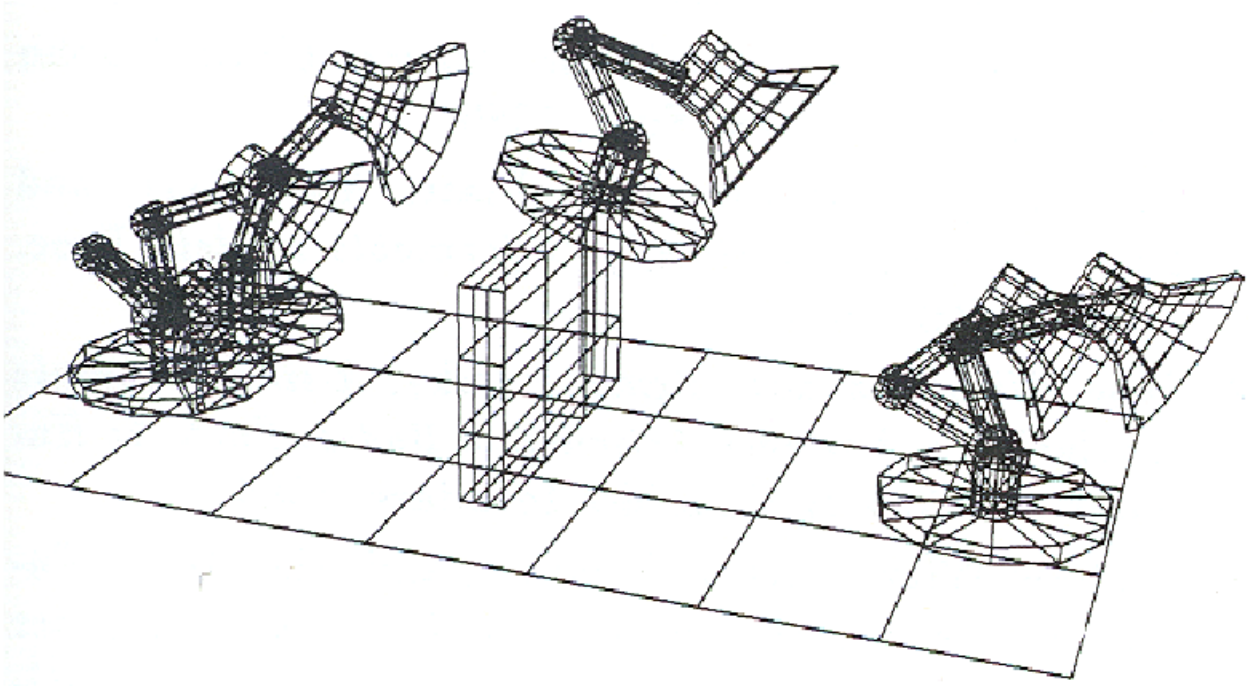


Heavier Base

Space Time Constraints

CGVR

- **Adapting Motion**

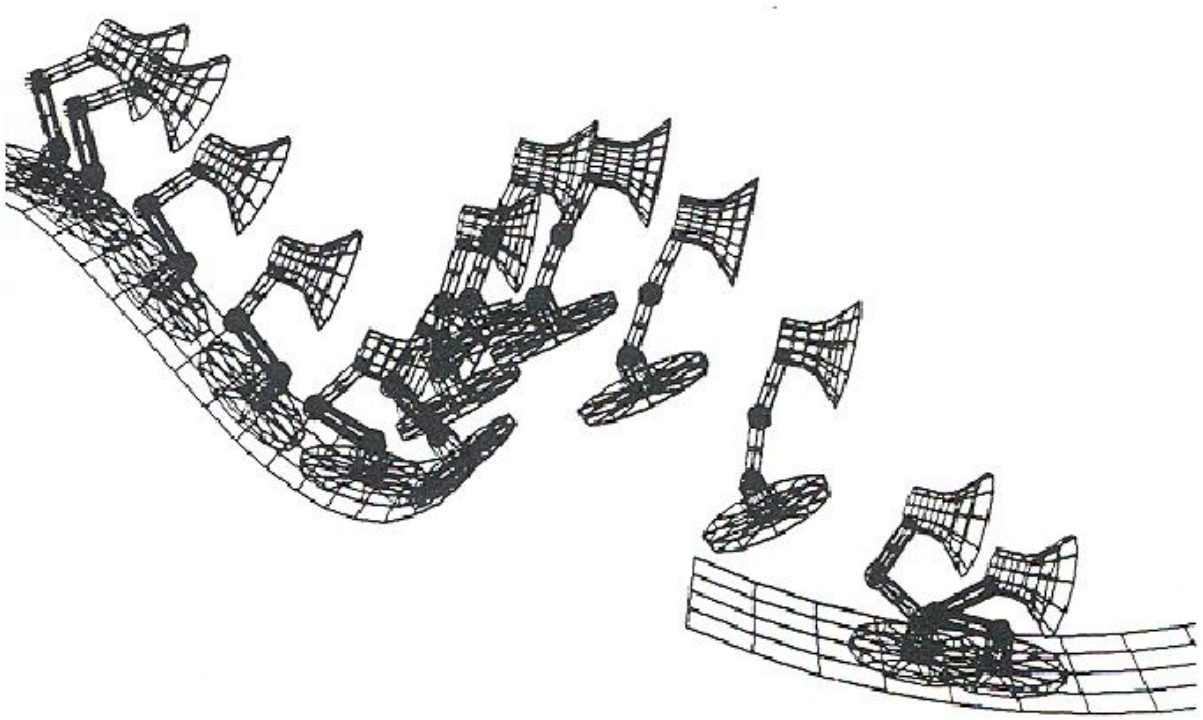


Hurdle

Space Time Constraints

CGVR

- **Adapting Motion**

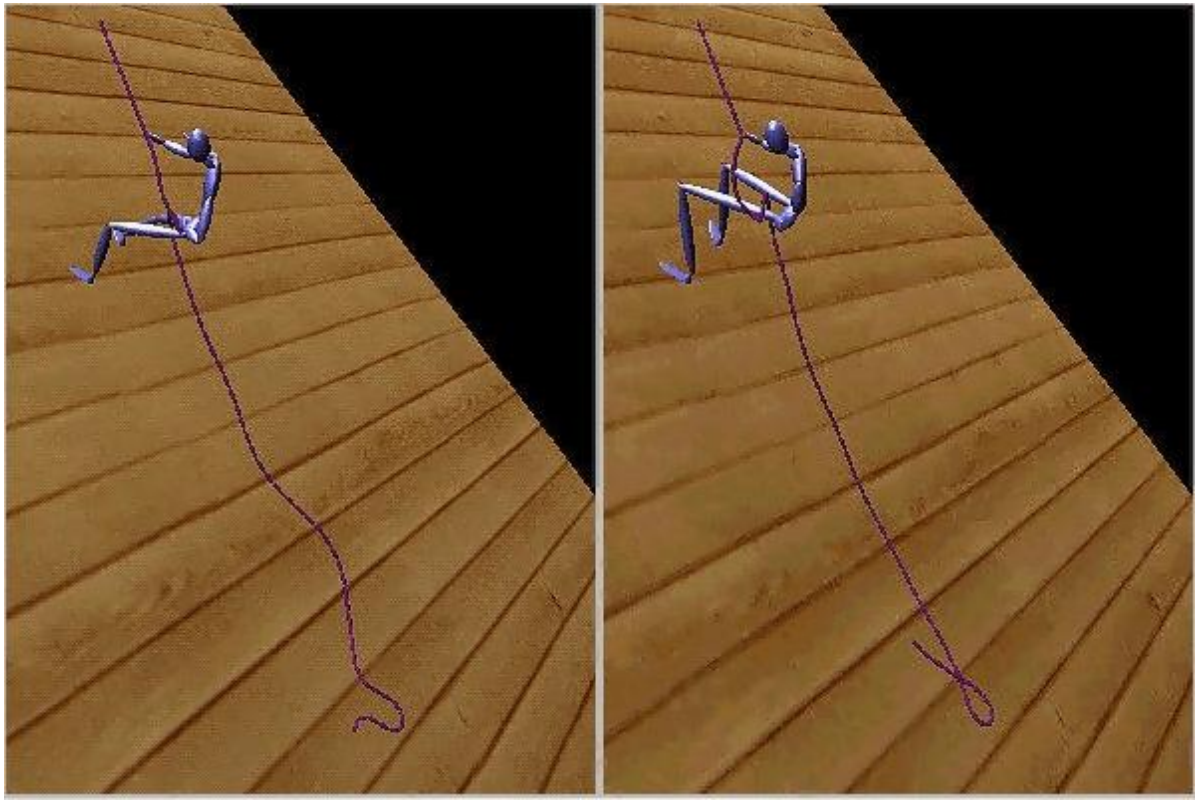


Ski Jump

Space Time Constraints

CGVR

- **Editing Motion**



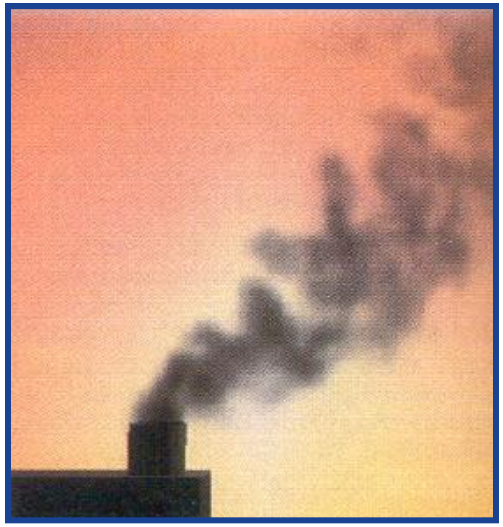
Original Adapted

Dynamics

CGVR

- **Other Physical Simulations**
  - Rigid bodies
  - Soft bodies
  - Cloth
  - Liquids
  - Gases
  - etc.

Cloth



Hot Gases

Summery

CGVR

- Computer Graphics
  - Imaging
  - Modeling
  - Rendering
  - Animation

- What is Your challenge?
  - There are so many problems!!!
  - Open your mind…