

CSE 167:  
Introduction to Computer Graphics  
Lecture #15: Toon Shading

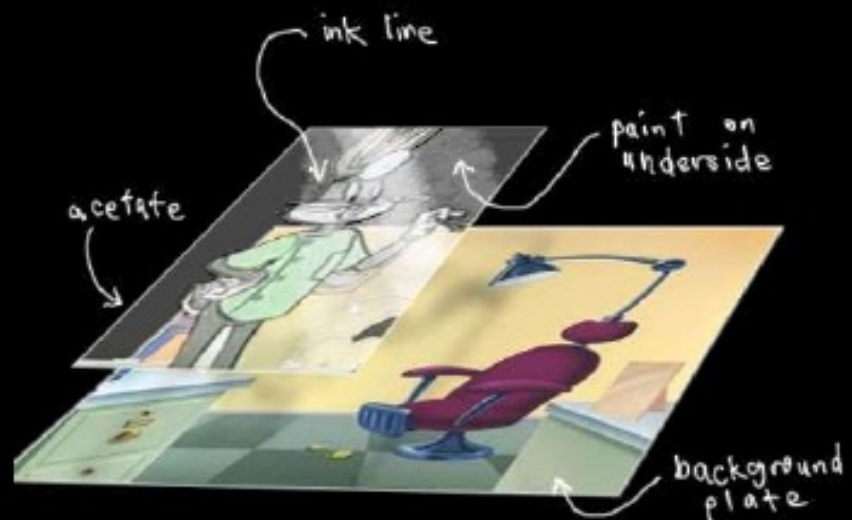
Jürgen P. Schulze, Ph.D.  
University of California, San Diego  
Fall Quarter 2020

# Cel Animation

**DEFINITION:** The process of first drawing each frame of animation onto paper, then transferring these to transparent acetate sheets (cels, short for celluloid). Colour is then added on the reverse of the cel. Layers are built up to create dimension.

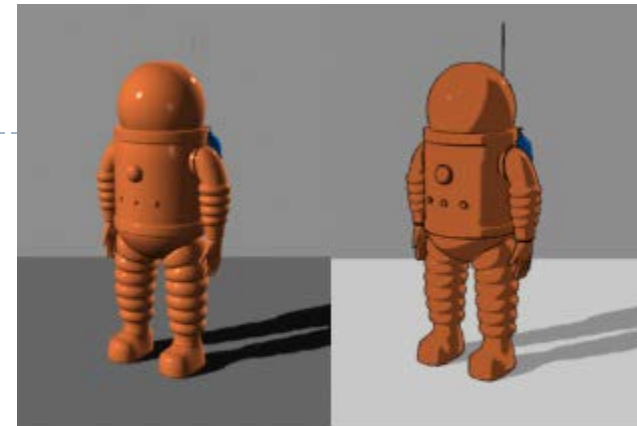


The cels are individually photographed onto film by a rostrum camera.



# Toon Shading

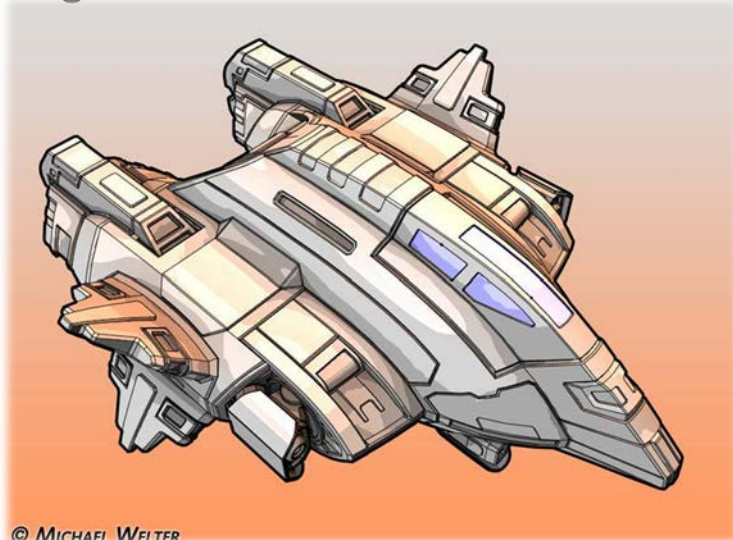
- ▶ A.k.a. Cel Shading (“Cel” is short for “celluloid” sheets)
- ▶ Gives any 3D model a cartoon-style look
- ▶ Main features:
  - ▶ Emphasizes silhouettes
  - ▶ Discrete steps for diffuse shading, highlights
- ▶ Non-photorealistic rendering method (NPR)
- ▶ Programmable shaders allow real-time performance



plastic shader

toon shader

*Source: Wikipedia*



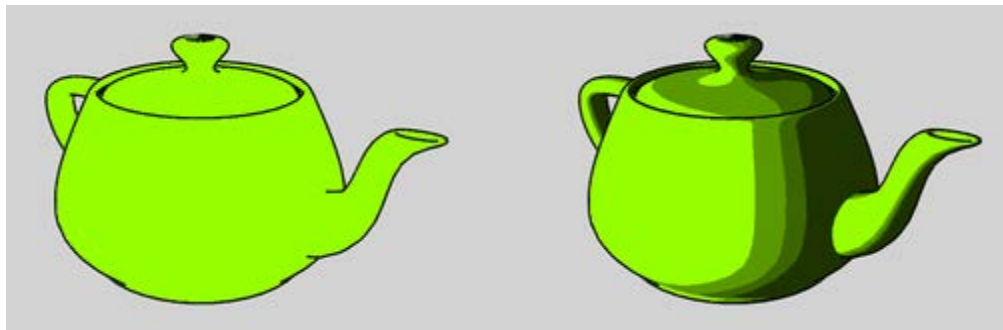
© MICHAEL WELTER



# Approach

---

- ▶ Start with regular 3D model
- ▶ Apply two rendering tricks:
  - ▶ Silhouette edges
    - ▶ Emphasize pixels with normals perpendicular to viewing direction.
  - ▶ Discretized shading
    - ▶ Conventional (smooth) lighting values calculated for each pixel, then mapped to a small number of discrete shades.



*Source: Wikipedia*

# Silhouette Edges

- ▶ Silhouette edge detection

- ▶ Compute dot product of viewing direction  $\mathbf{v}$  and normal  $\mathbf{n}$

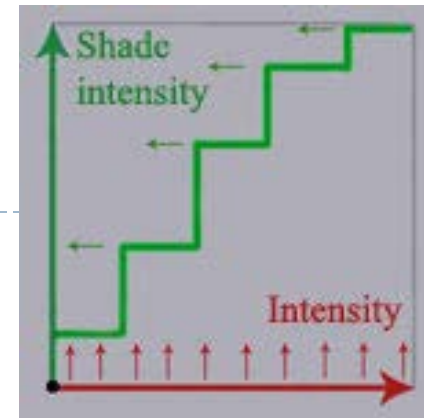
$$\text{edge} = \max(0, \mathbf{n} \cdot \mathbf{v})$$

- ▶ Use cutoff value for edge:

- ▶ if  $\text{edge} < 0.01$  draw black, else don't change pixel color



# Discretized Shading



- ▶ Compute diffuse and specular shading as usual
  - ▶  $diffuse = n \cdot L$
  - ▶  $specular = (R \cdot e)^p$
- ▶ Discretize shading:
  - ▶ If..then..else tree comparing values against thresholds

```
if (intensity > 0.95)
    color = float4(1.0,1,1,1.0) * color;
else if (intensity > 0.5)
    color = float4(0.7,0.7,0.7,1.0) * color;
else if (intensity > 0.05)
    color = float4(0.35,0.35,0.35,1.0) * color;
else
    color = float4(0.1,0.1,0.1,1.0) * color;
```

# Toon Shading Demo

---



<http://www.bonzaisoftware.com/npr.html>