计算机图形学

光照和明暗处理

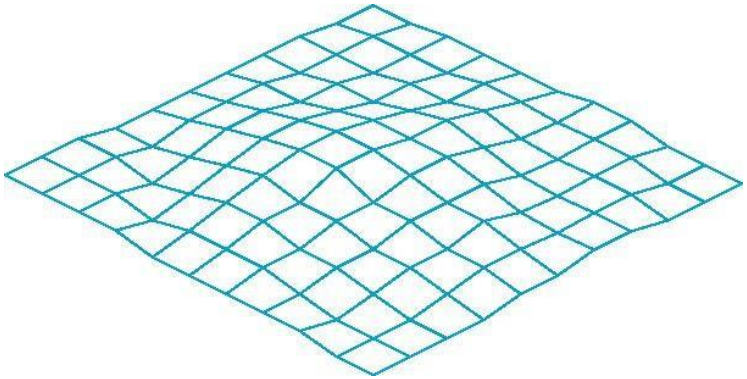
陈中贵
厦门大学

第二节 多边形明暗处理

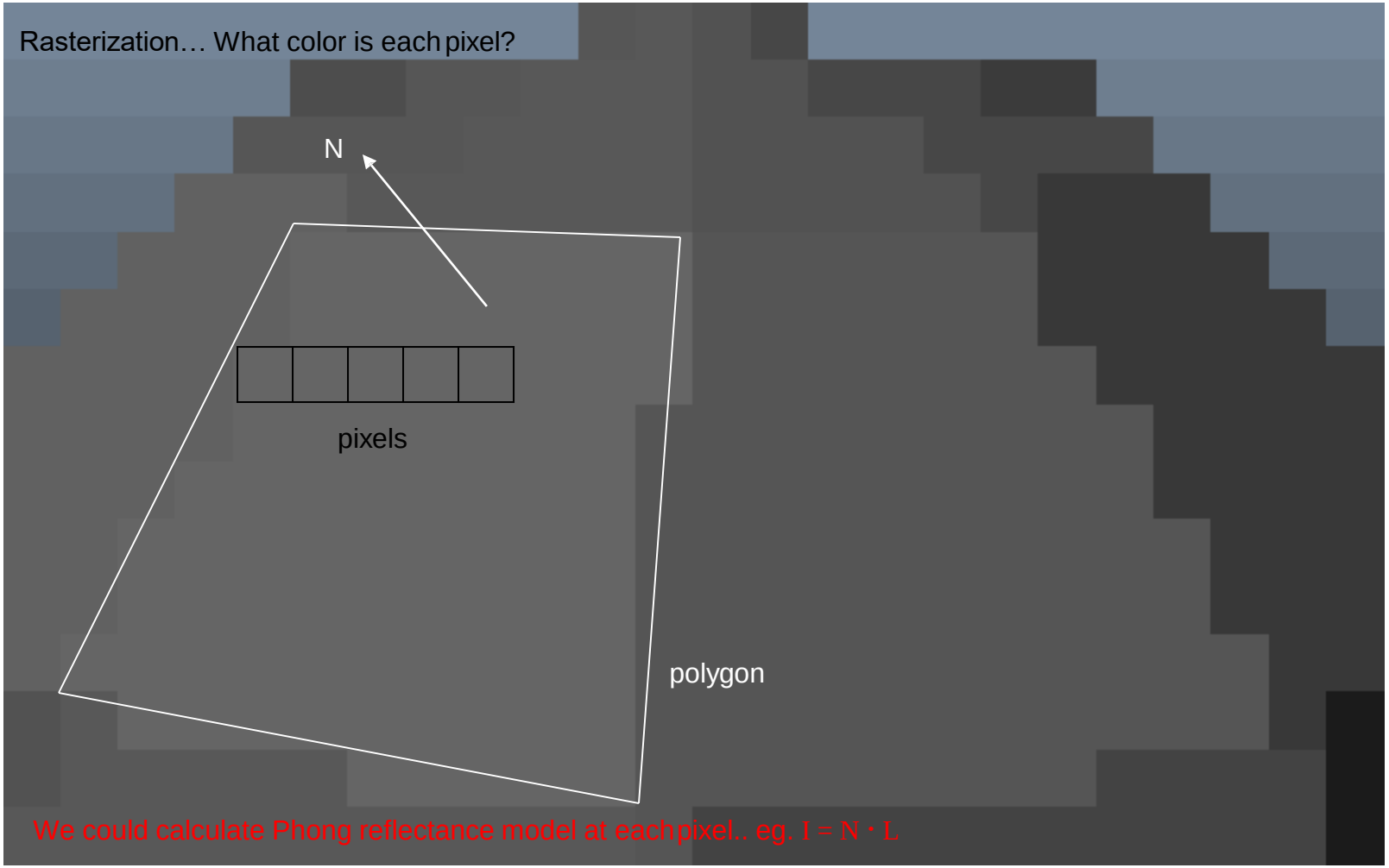
- 多边形的明暗处理
  - 平面着色
  - Gouraud着色
  - Phong着色

多边形网格的明暗处理

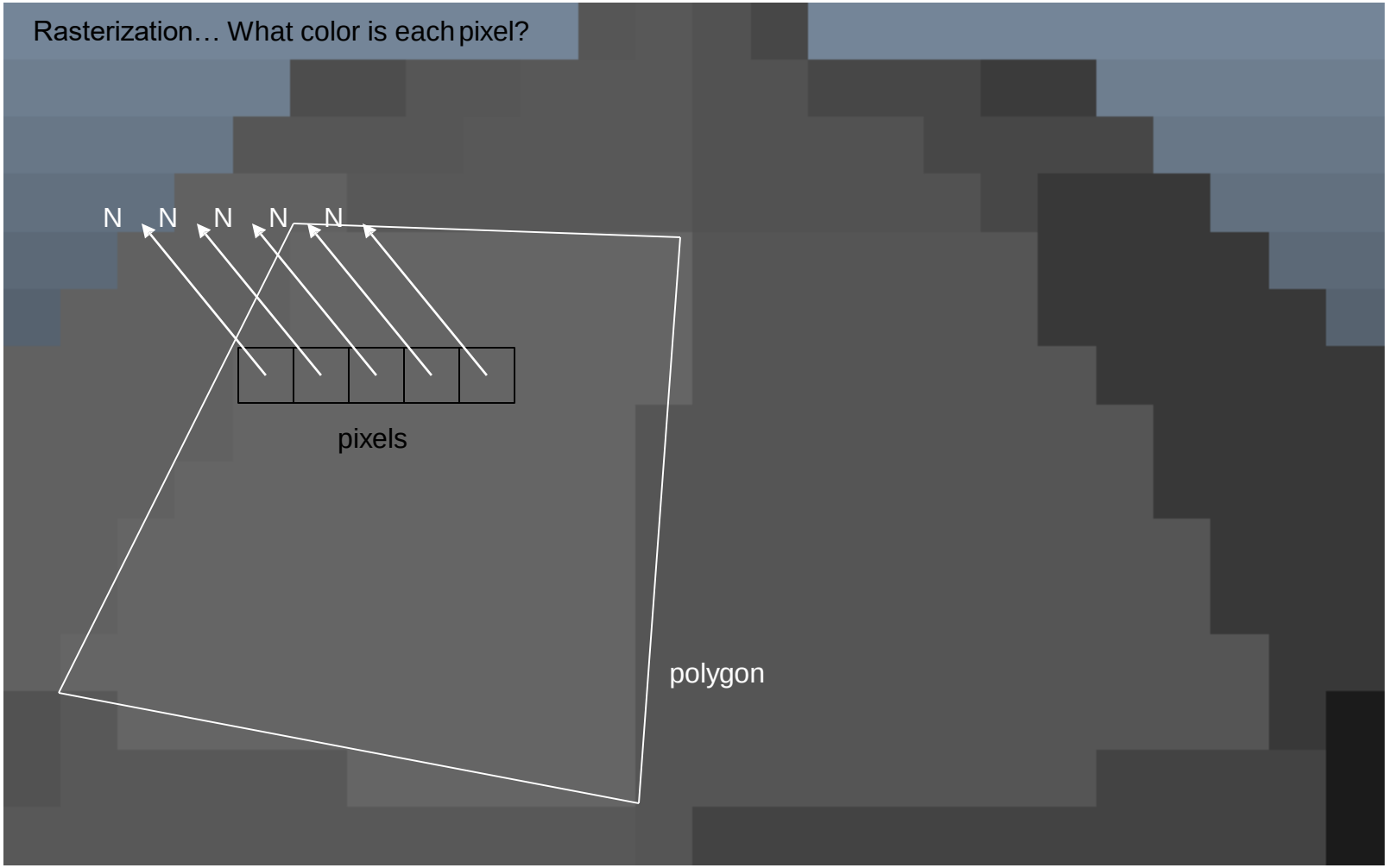
- 在多边形网格中每个多边形为平面，那么存在唯一的法向量
- 三种明暗处理的方法
  - 平面着色(flat shading)
  - Gouraud着色(Gouraud shading)或插值着色
  - Phong着色(Phong shading)



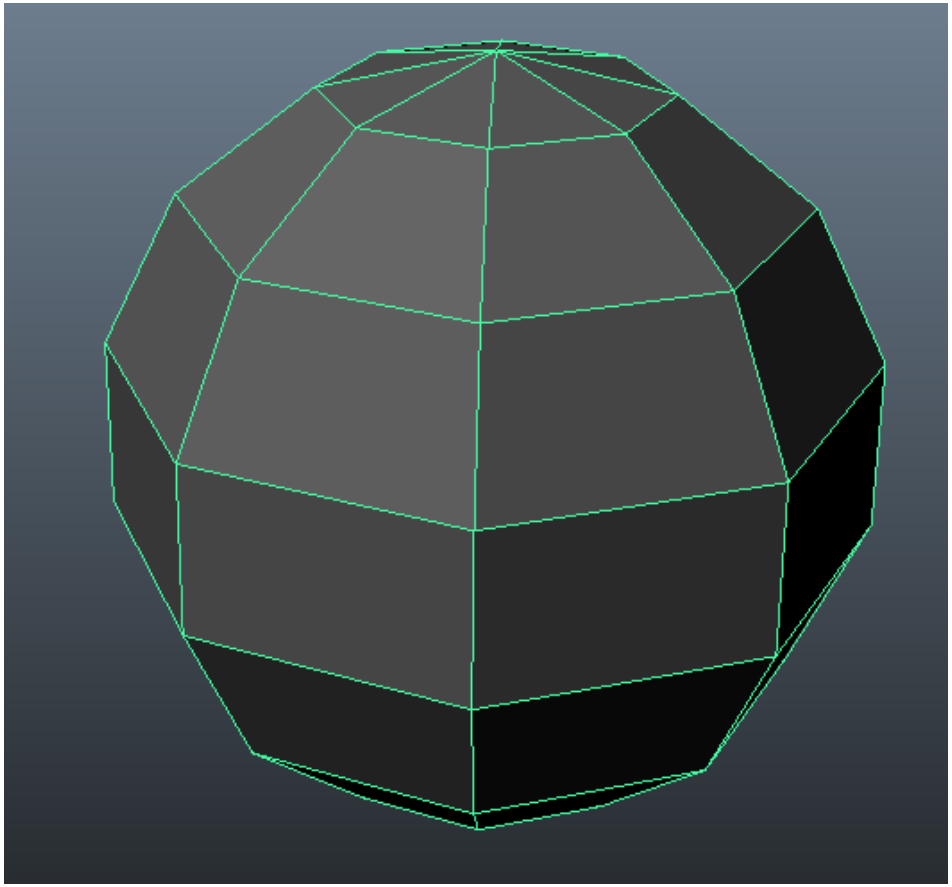
光栅化中像素颜色如何确定?



Question: **What is the surface normal at each pixel?**


Rasterization… What color is each pixel?
N
N
N
N
N
pixels
polygon

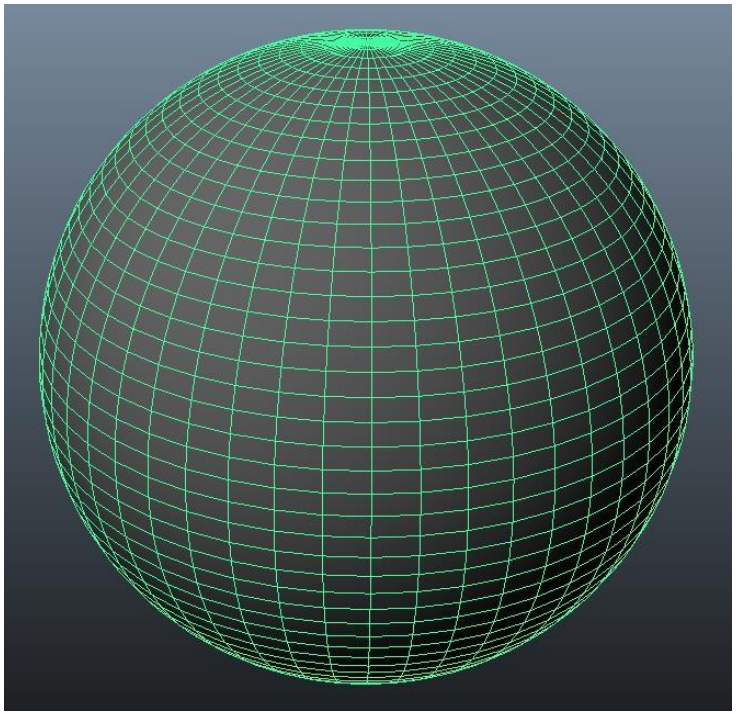
平面着色（Flat Shading）

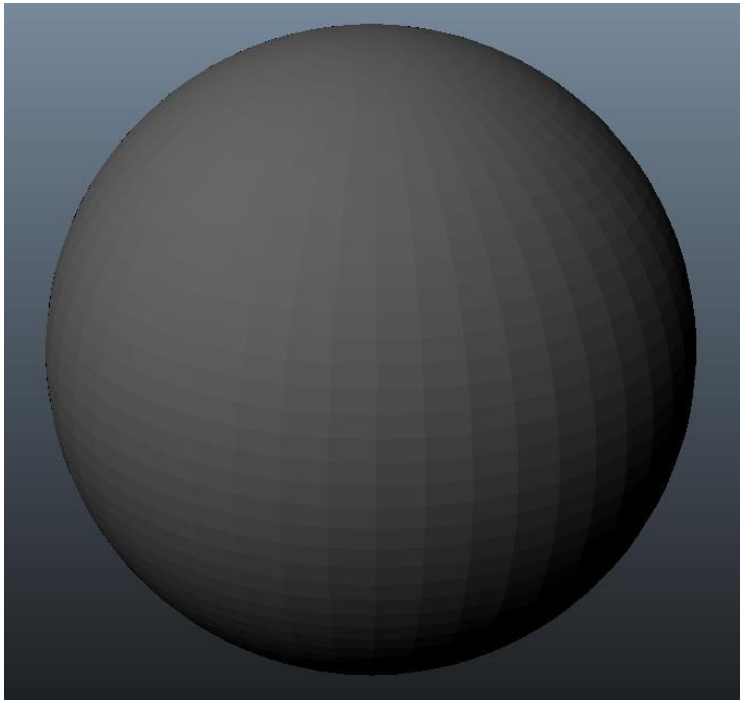


Using the ***face normal*** *for each pixel* is called **flat shading**. Every pixel gets the surface normal of the whole polygon.

glShadeModel (GL_FLAT);

One solution… Increase **tesselation**…

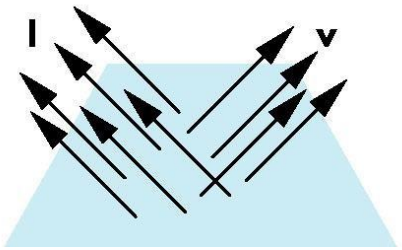


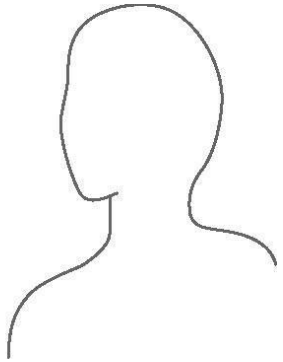


Uses a *lot* of polygons. To get correct images, polygons must be smaller than a pixel.

平面明暗处理

- 在同一多边形上法向**n**为常向量
- 视点在无穷远，视点方向**v**是常向量
- 光源在无穷远，入射方向**l**也是常向量
- 从而对于每个多边形，只需要计算其上一点的颜色，其它点的颜色与它相同





OpenGL平面明暗处理

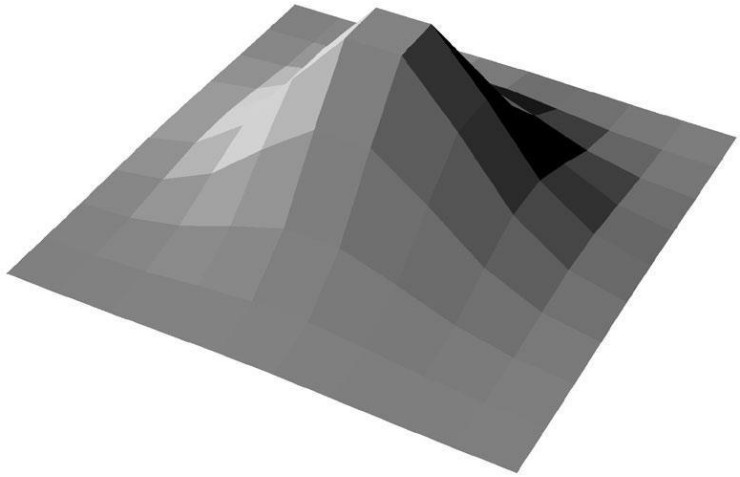
- OpenGL中设置平面着色：

```
glShadeModel(GL_FLAT);
```

- 如何选择多边形的法向或颜色：
  - 单个多边形(GL_POLYGON)　　第1个顶点
  - 独立三角形(GL_TRIANGLES)　　第3i个顶点
  - 独立四边形(GL_QUADS)　　第4i个顶点
  - 四边形带(GL_QUAD_STRIP)　　第2i+2个顶点
  - 三角形带或三角形扇　　第i+2个顶点

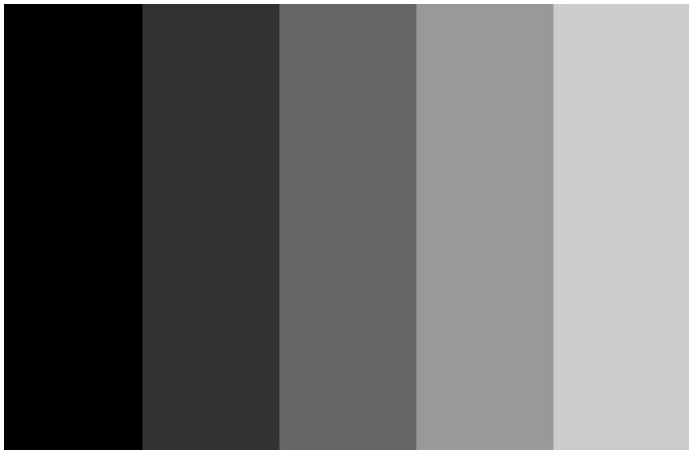
特点

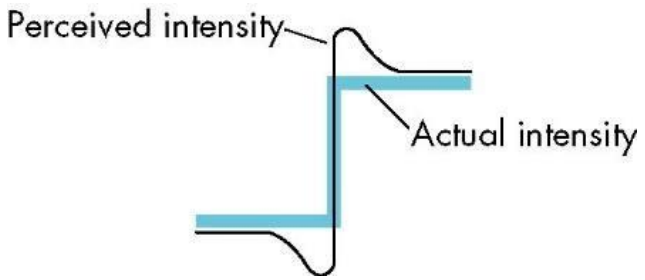
- 网格中每个多边形的颜色不同
  - 如果多边形网格表示的是一个光滑曲面，那么这种效果显然是不令人满意的



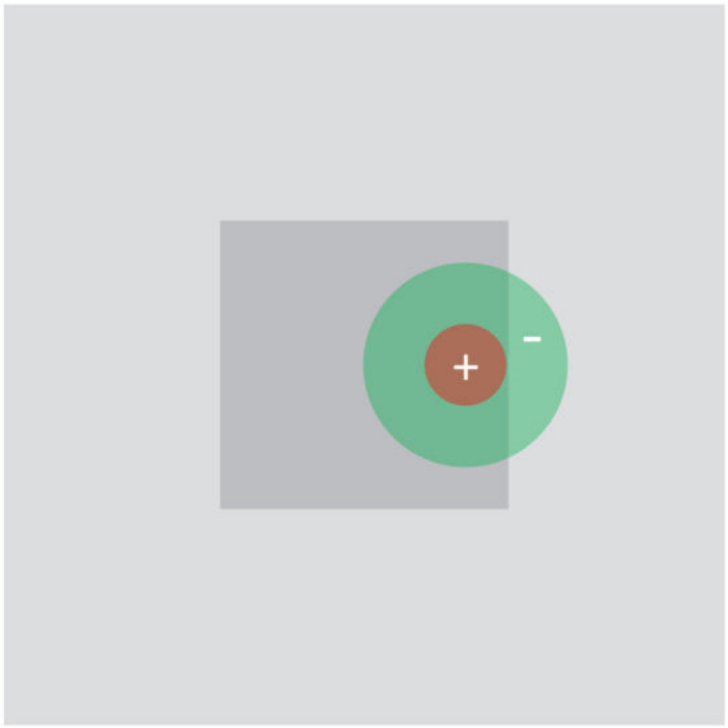
人类视觉系统

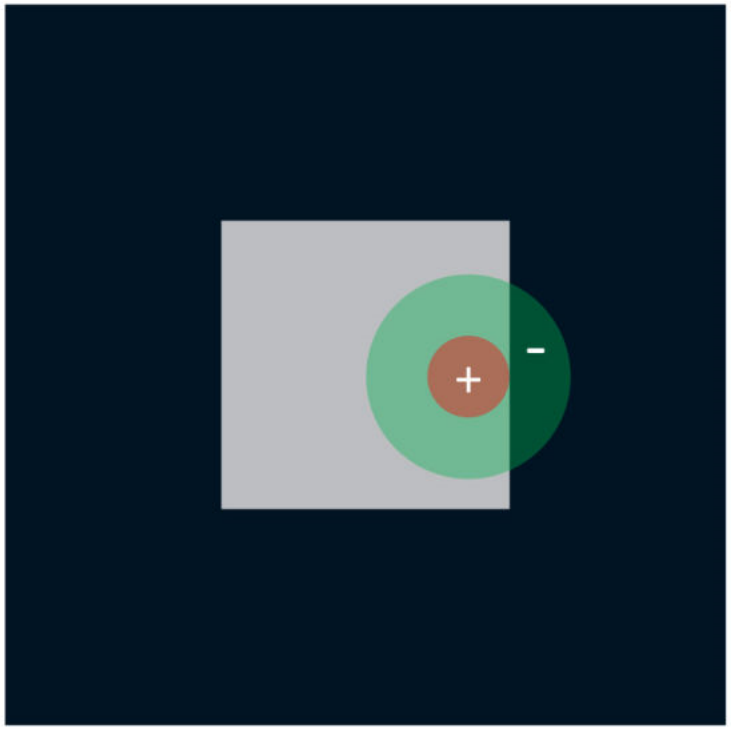
- 人类视觉系统对光强的变化非常敏感，称为旁侧抑制特性(lateral inhibition)
- 注意下图边界的条状效果，称为Mach带
- 没有办法避免这种情形，只有给出更平滑的明暗处理方法



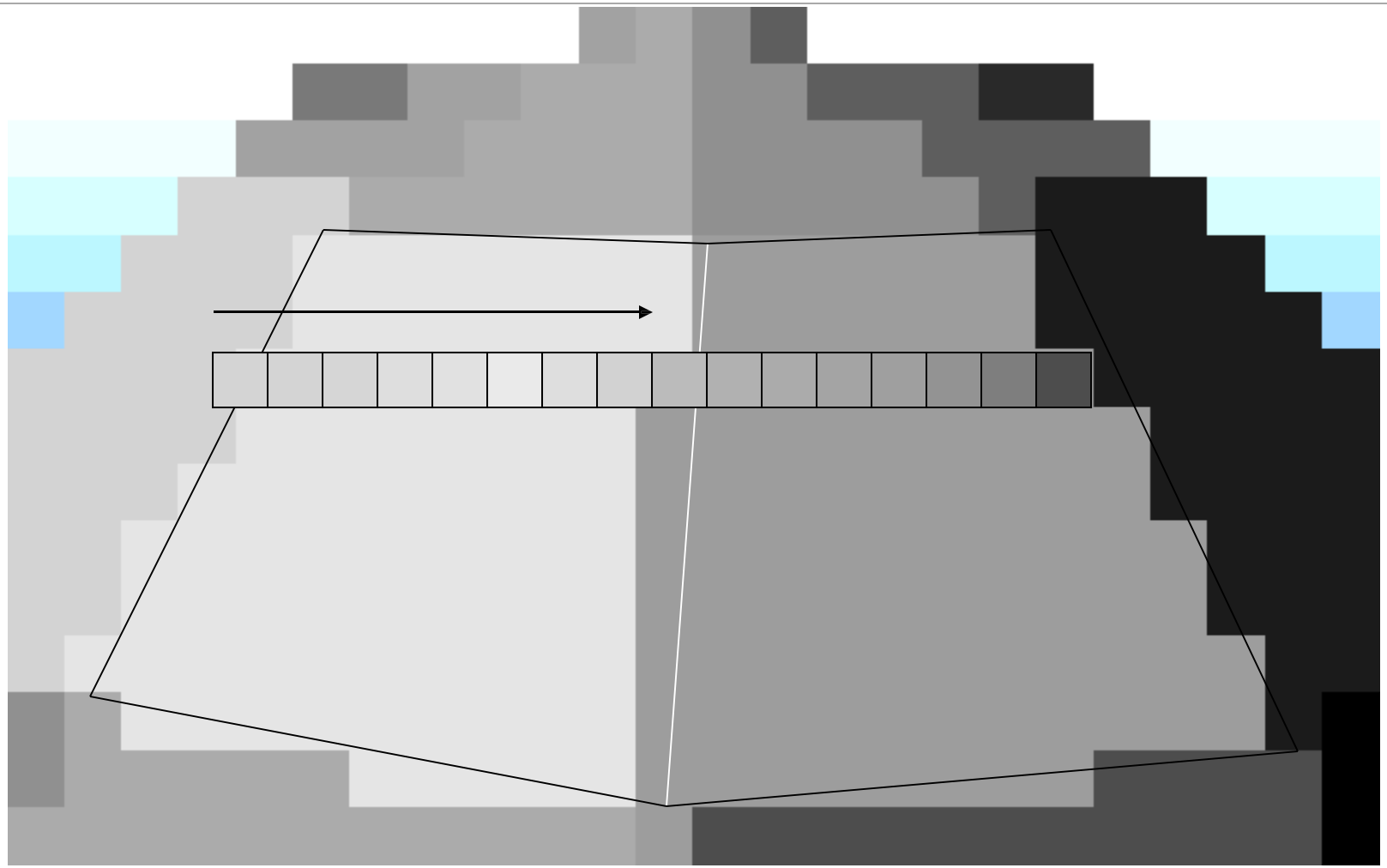


人类视觉系统





Gouraud Shading



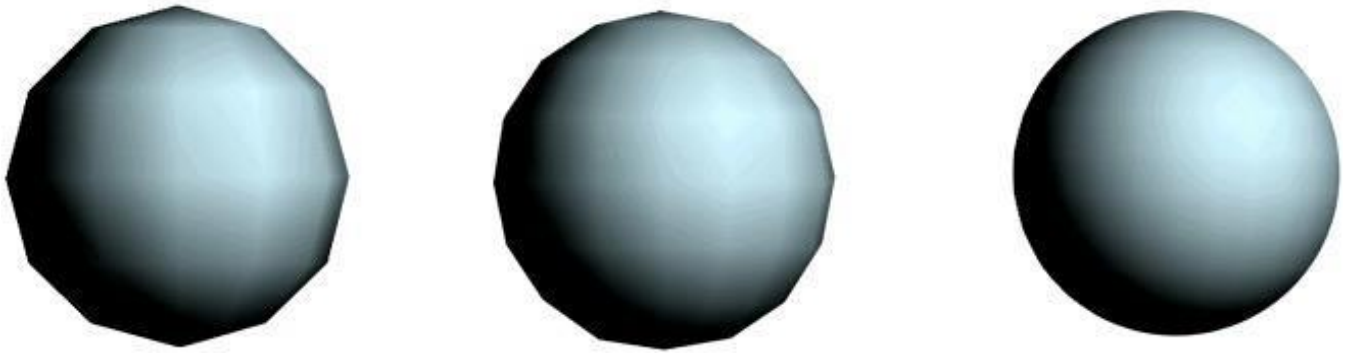
Henri Gouraud (French), 1971 “Continuous shading of curved surfaces”

What if we interpolate the *intensity* across the polygons?

Gouraud Interpolation

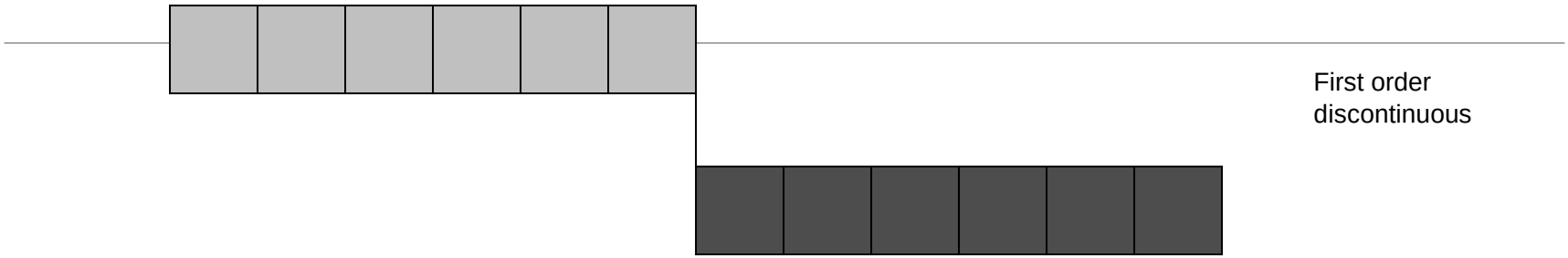
Flat Shading

Gouraud Shading



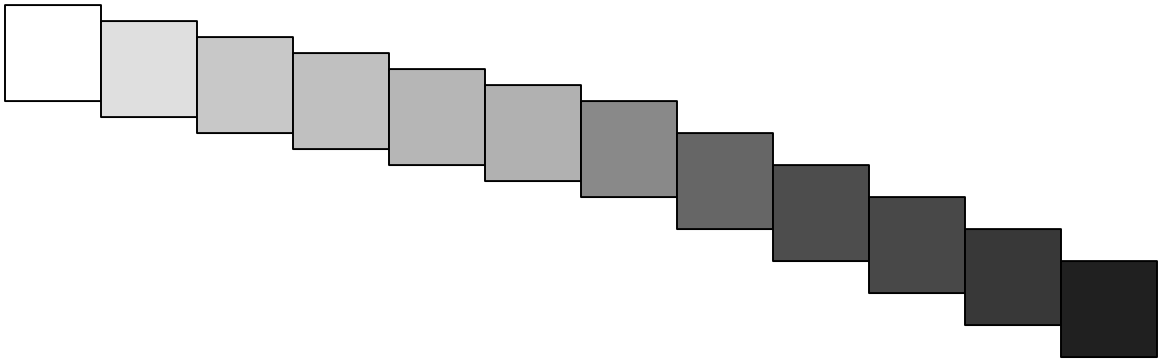
Notice Mach effect at edges of polygons. Mach Bands are an optical illusion.
Our eye notices discontinuous changes in intensity as a *jump* in brightness.

Flat Shading

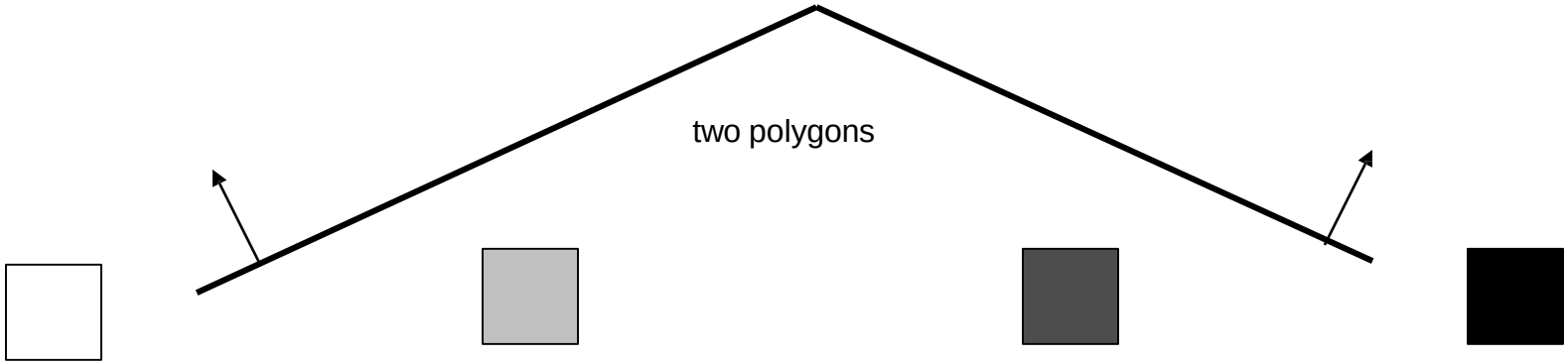


First order
discontinuous

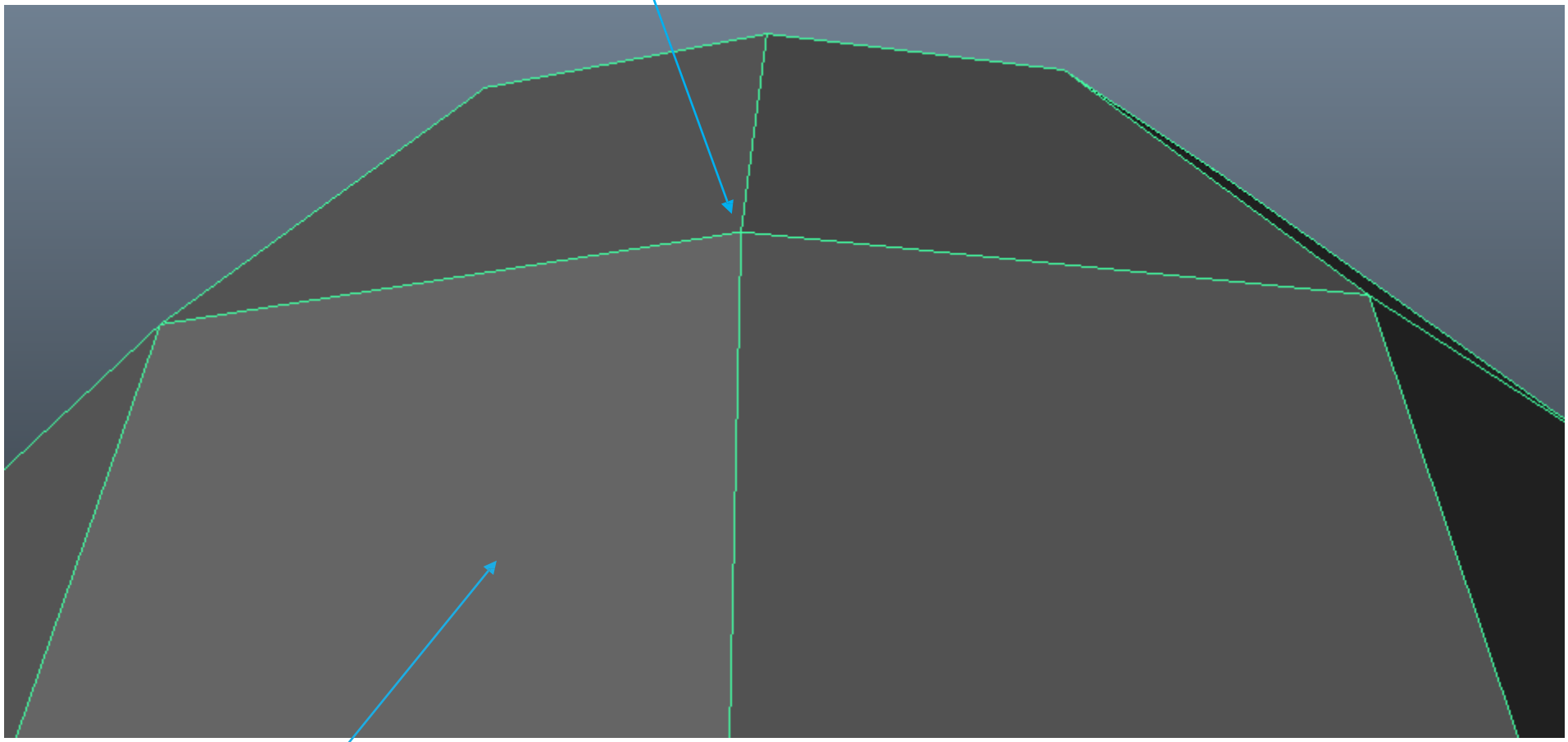
Gouraud Shading



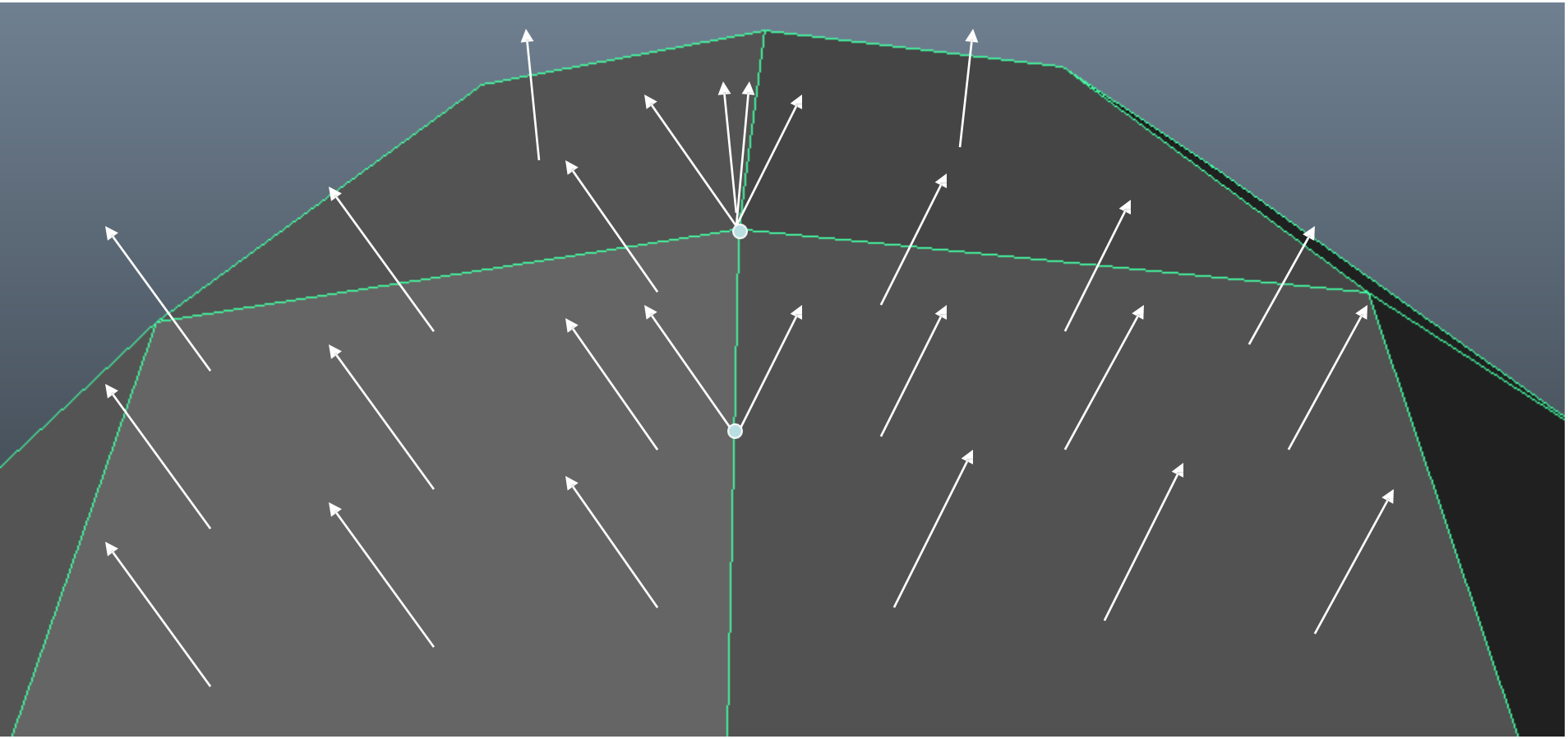
Second order
discontinuous



该点法向是?



这里呢?

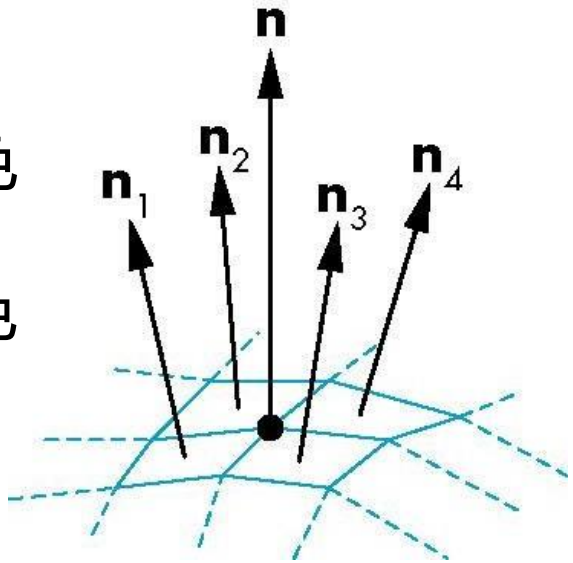


A vertex connecting two or more faces has a *discontinuous normal*.

Gouraud明暗处理

- 在网格中每个顶点处有几个多边形交于该点，每个多边形有一个法向，取这几个法向的平均得到该点的法向

$$n = \frac{n_1 + n_2 + n_3 + n_4}{\left| n_1 + n_2 + n_3 + n_4 \right|}$$

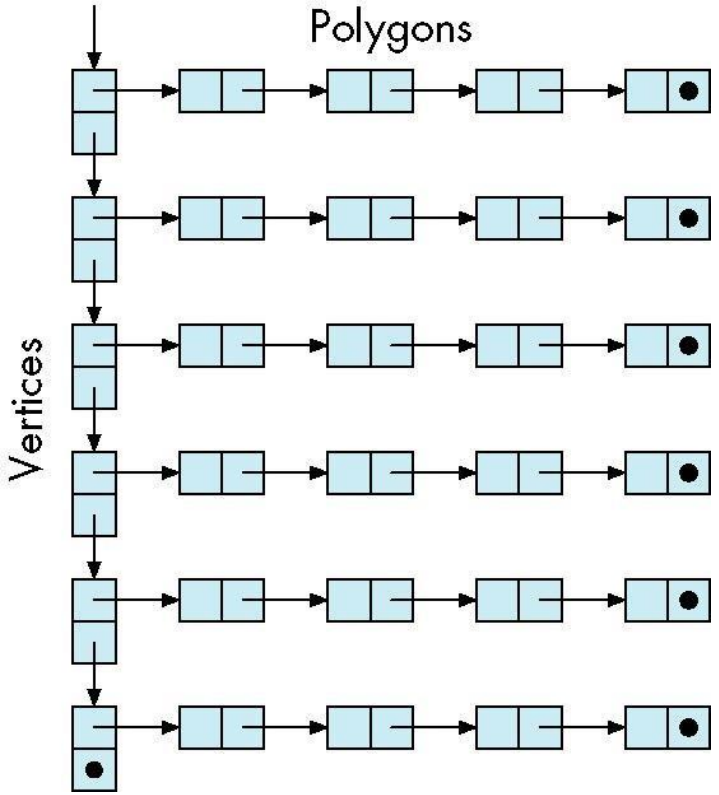


- 然后利用简单光照模型计算出顶点的颜色
- 对多边形内的点，采用线性插值确定颜色
- OpenGL中设置平滑着色：

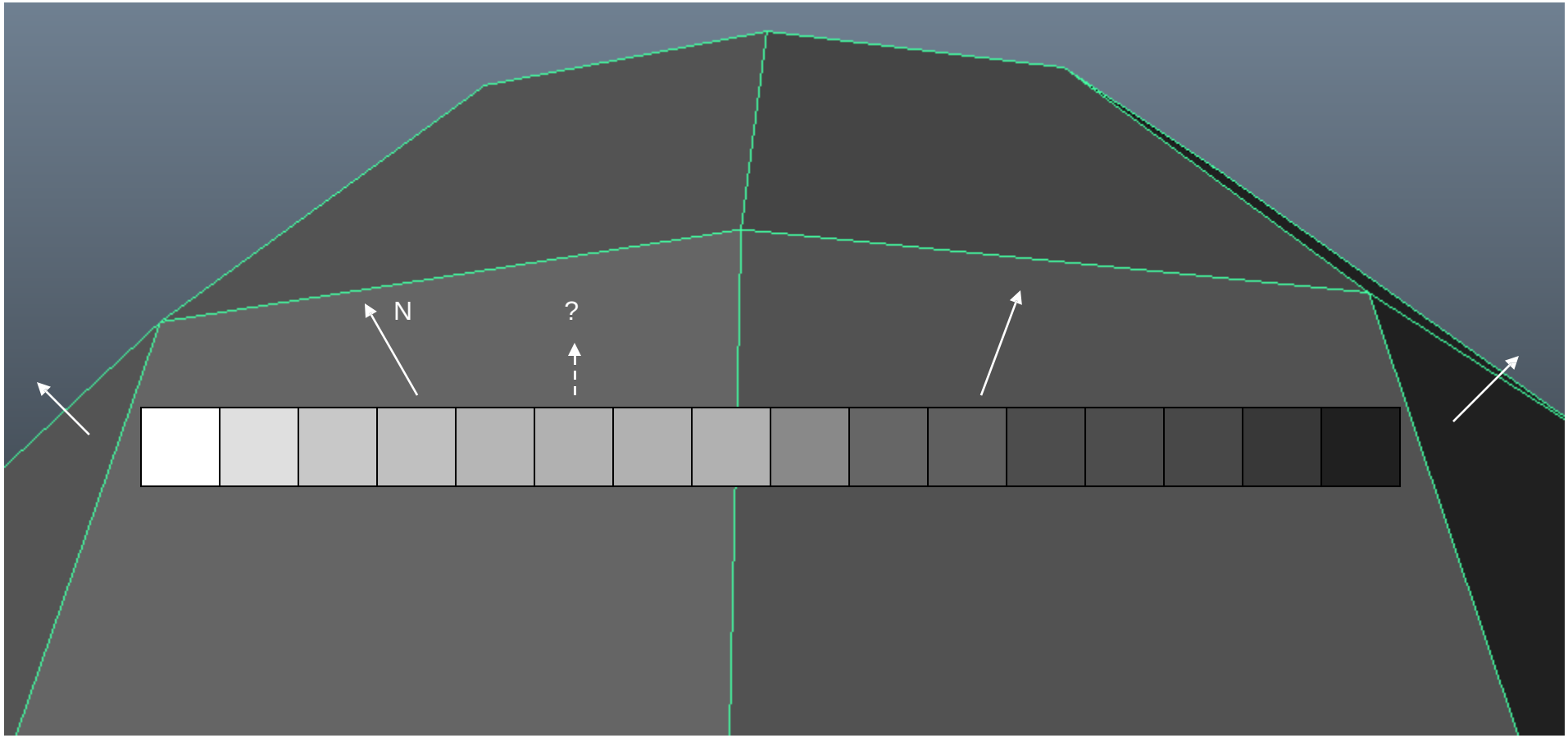
```
glShadeModel(GL_SMOOTH);
```

如何确定法向

- 如何找出与某个顶点相邻的各多边形?
  - 如果程序中只是列出各顶点，那么没有信息找到上述多边形
  - 如右所示数据结构却可以做到这一点



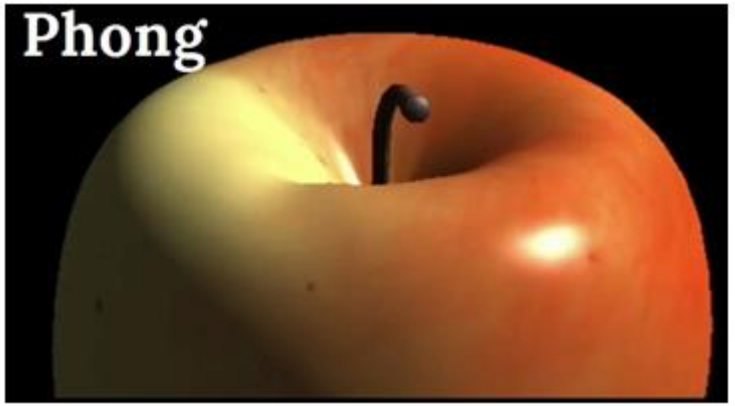
Gouraud Shading

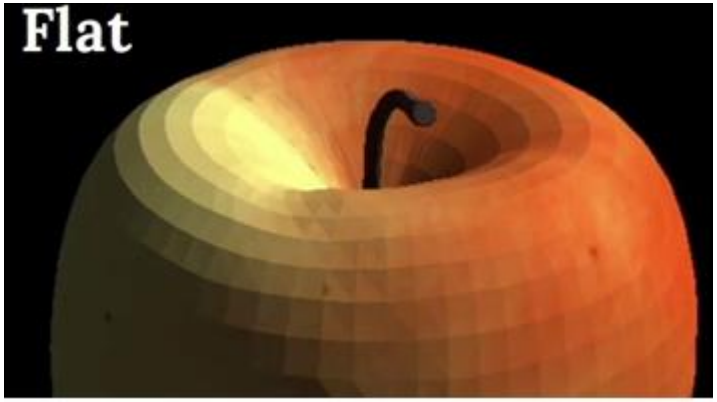


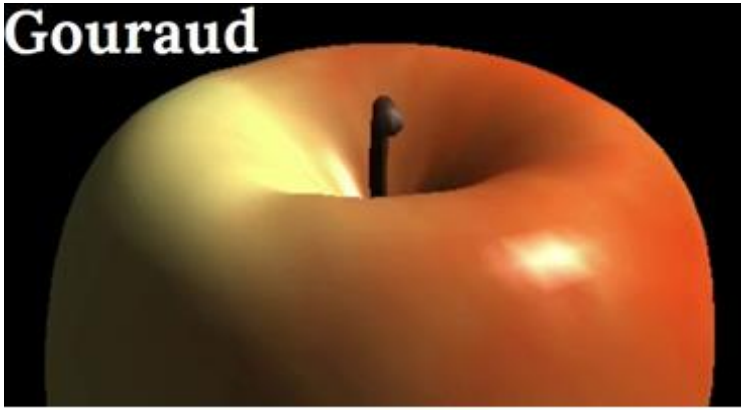
Interpolating the intensity is incorrect, because we are basically guessing intensity instead of using a surface normal…

Gouraud着色法优点

- 效果优于平直着色法
- 所需的处理比Phong着色法少

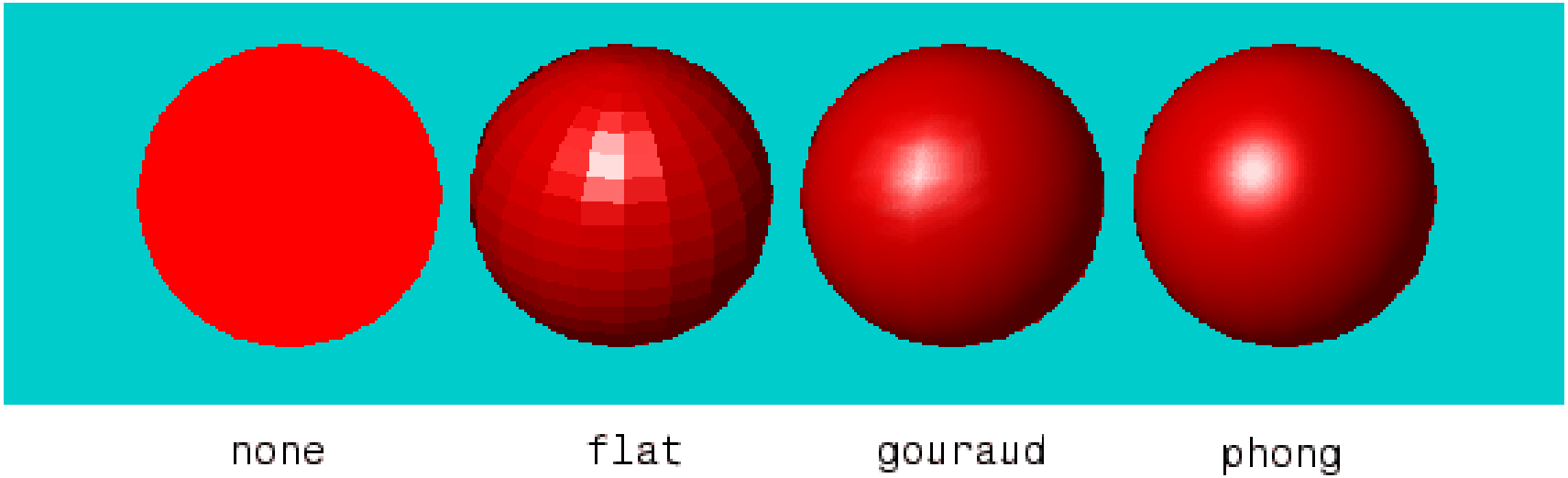






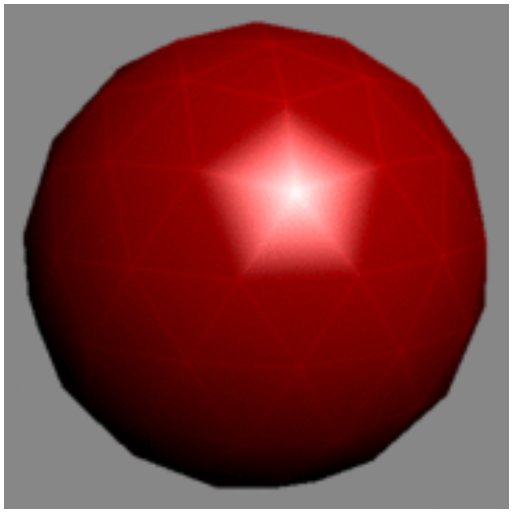
Gouraud着色法缺点

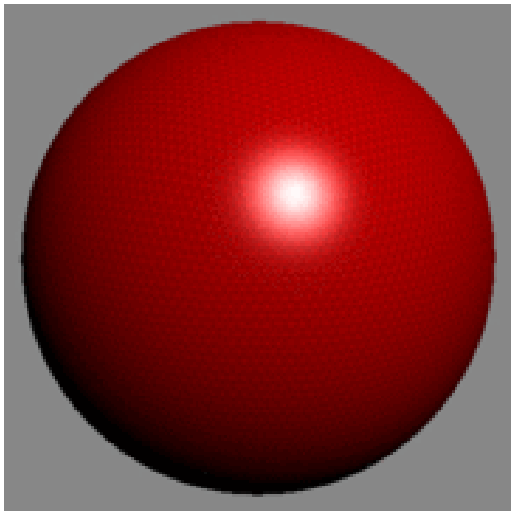
- 着色后仍然可以看出一个个小平面的效
- 渲染一些与位置相关的光照效果（比如高光）时，得到的效果就会有问题

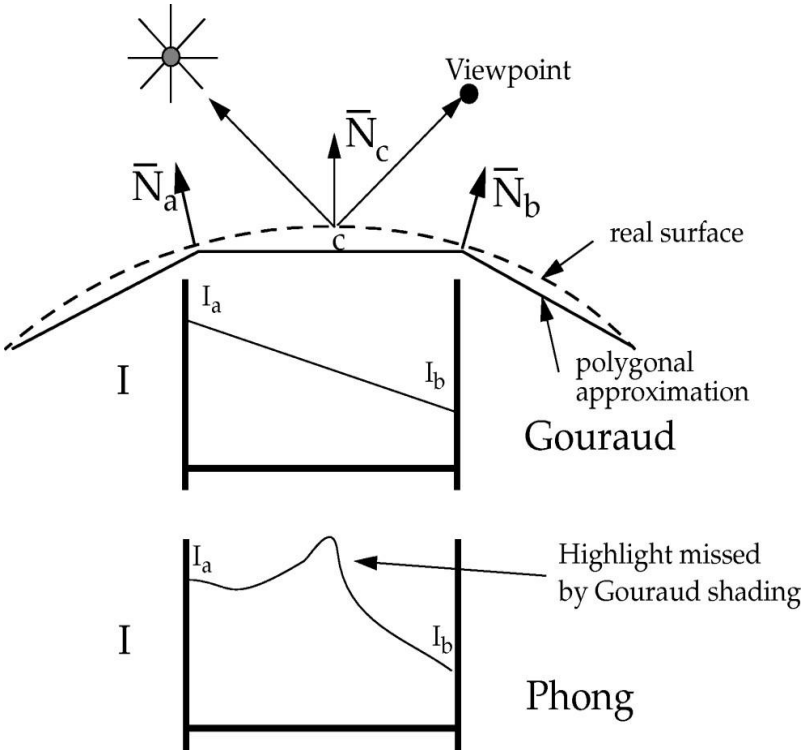


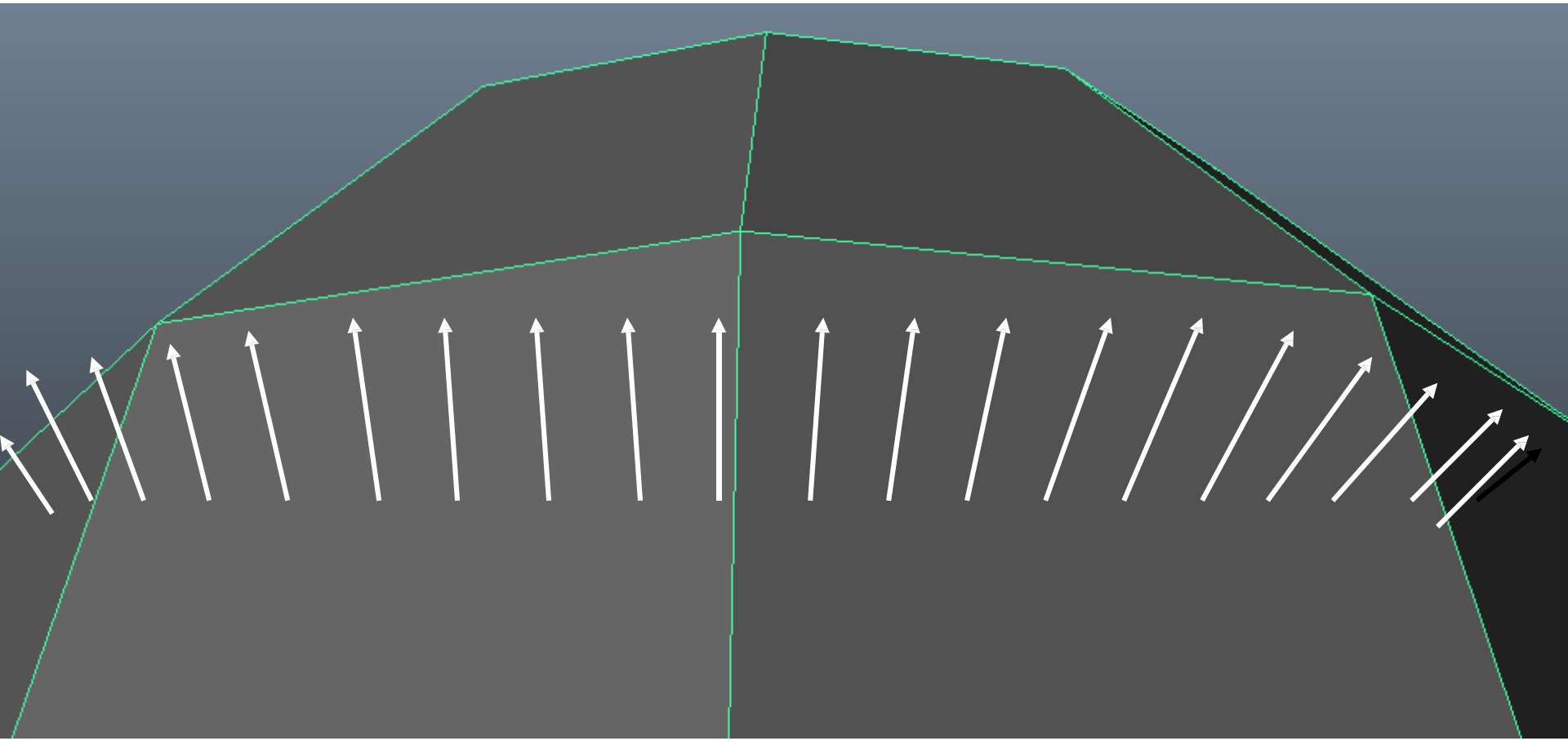
Gouraud着色法缺点

- 着色后仍然可以看出一个个小平面的效
- 渲染一些与位置相关的光照效果（比如高光）时有问题







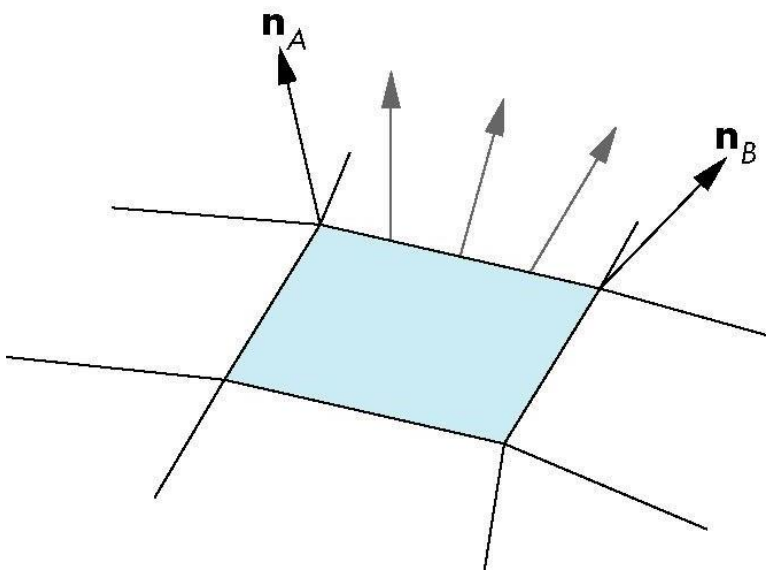


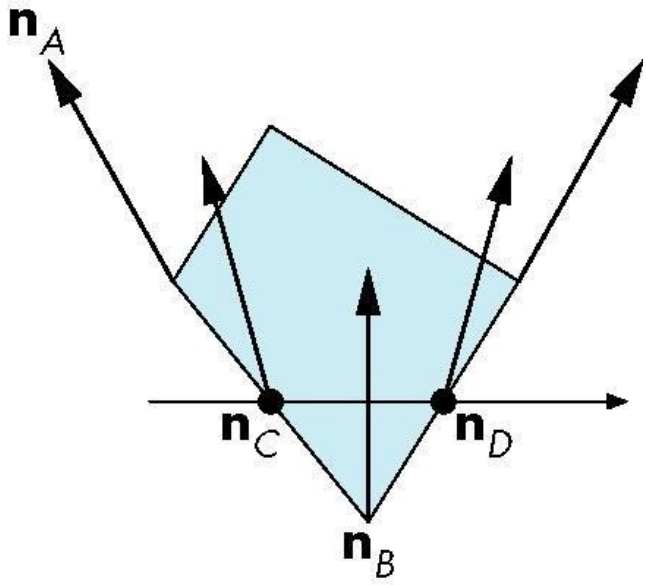
Phong Interpolation.. (do not confuse with Phong Reflectance model)

What we want to do is, ***interpolate the surface normal*** across polygons…

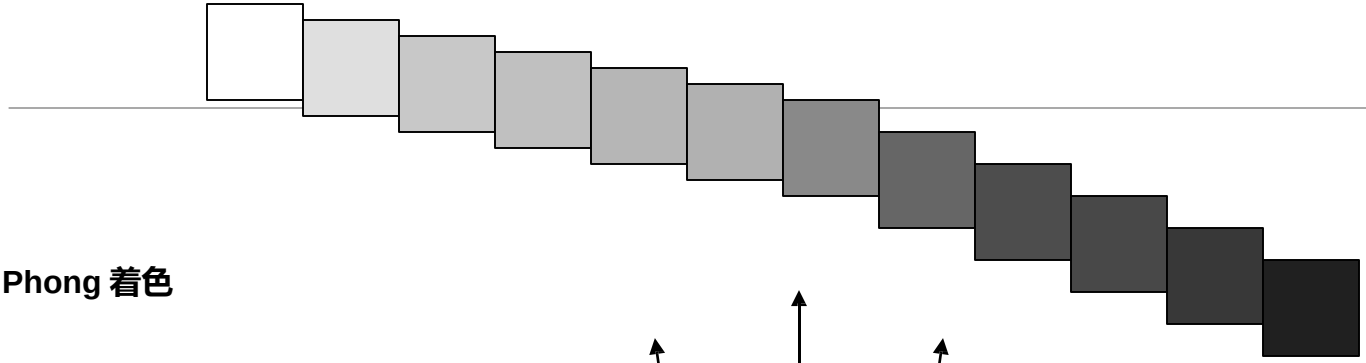
Phong明暗处理

- 与Gouraud方法不同，Phong方法是根据每个顶点的法向，插值出多边形内部各点的法向，然后基于光照模型计算各点的颜色

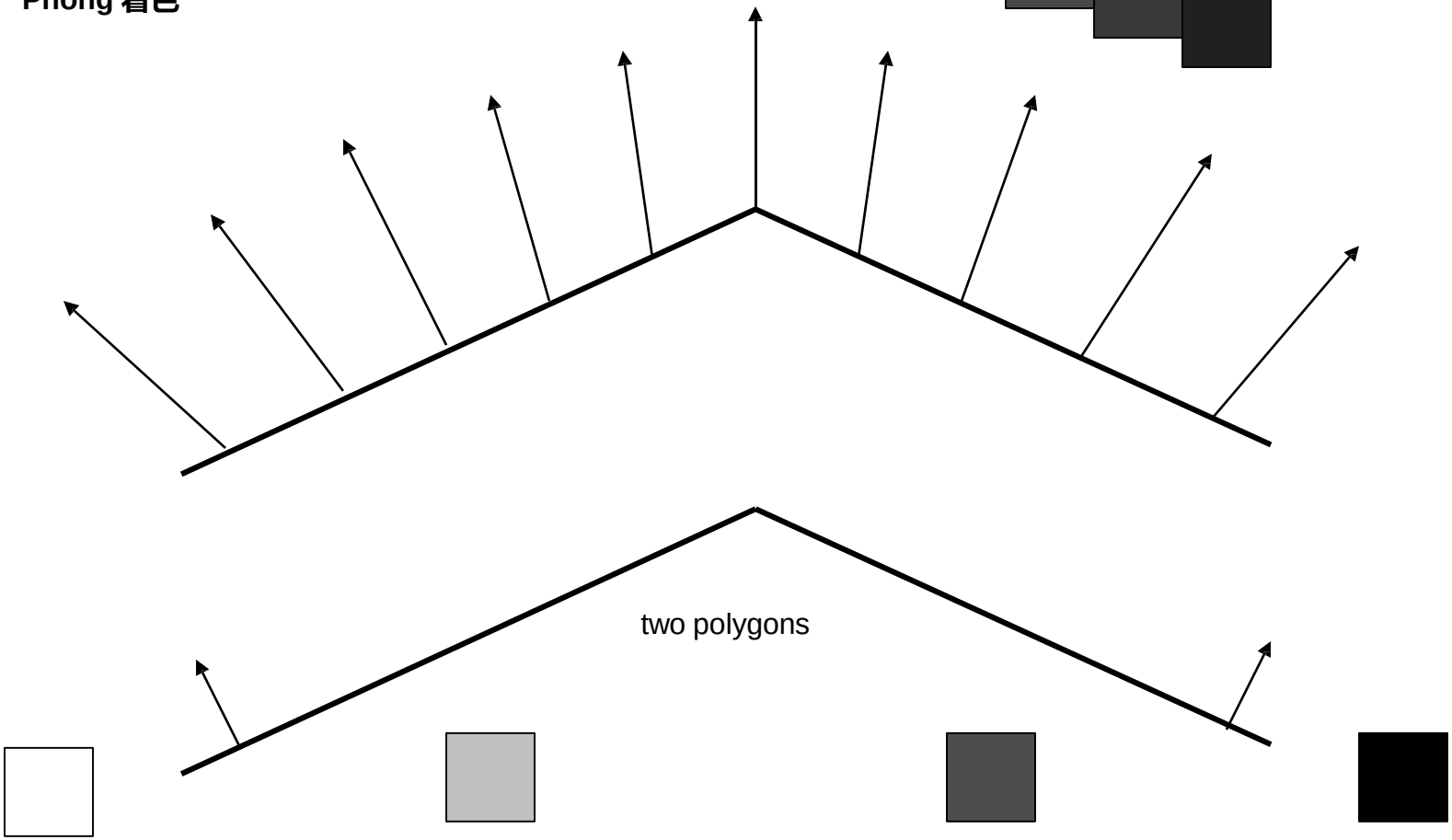




Gouraud 着色

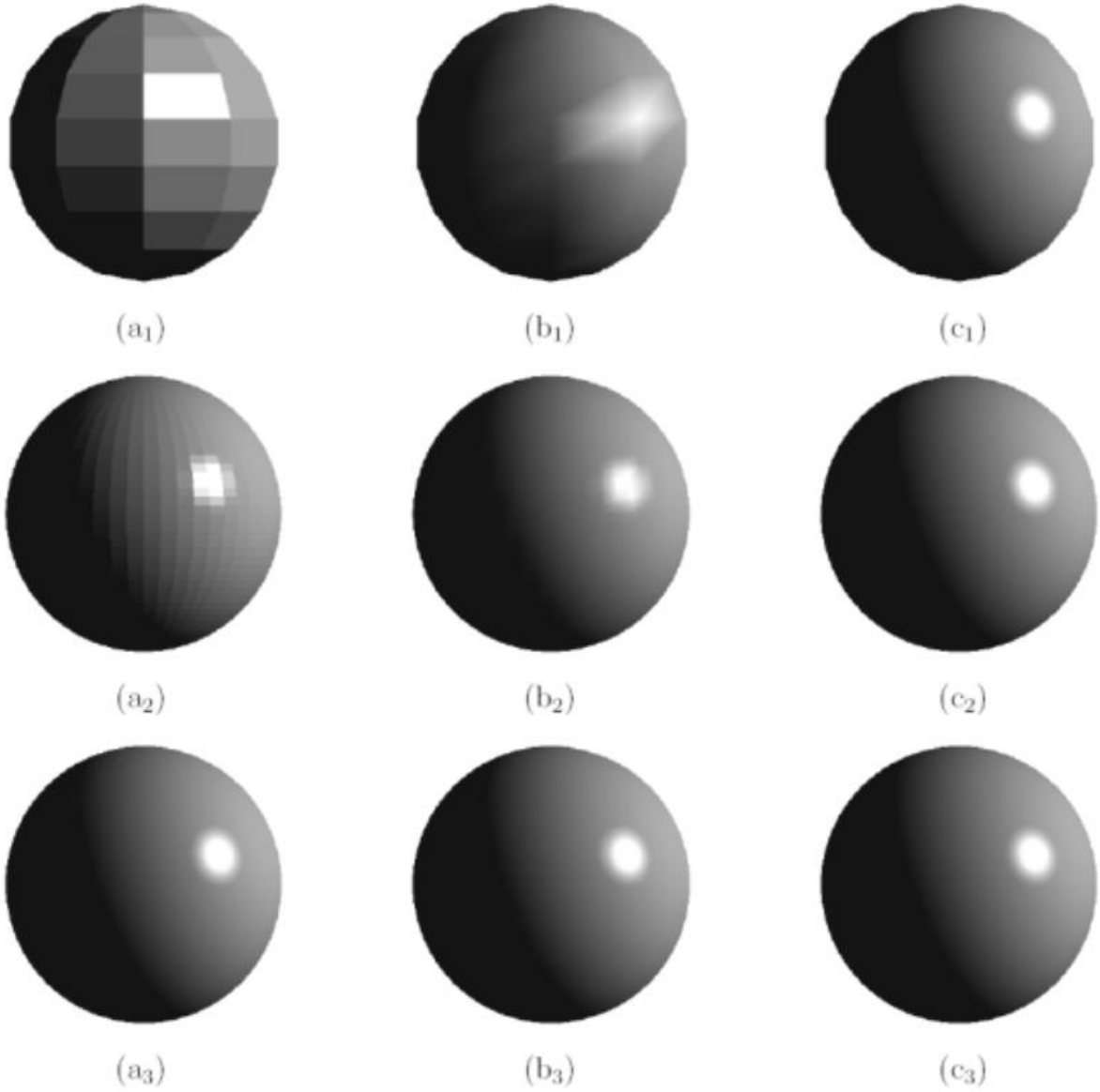


Phong 着色



着色模式

- 平面着色(flat shading)
  - 无插值，每个多边形只会呈现一个颜色，这个颜色由面法向量和光照计算得来。在该模型中，每个多边形中只有多边形的面存在法向量，而其各个顶点没有。
- Gouraud着色(Gouraud shading)或插值着色
  - 插值颜色，通常先计算多边形每个顶点的光照，再通过双线性插值计算三角形区域中其它像素的颜色。
- Phong着色(Phong shading)
  - 插值法向量，多边形中每个顶点都有一个法向量，通过这些法向量与光照计算，来得到每个点的颜色。在使用有限数量的多边形时，对定点法向量进行插值可以给出近似平滑的曲面效果

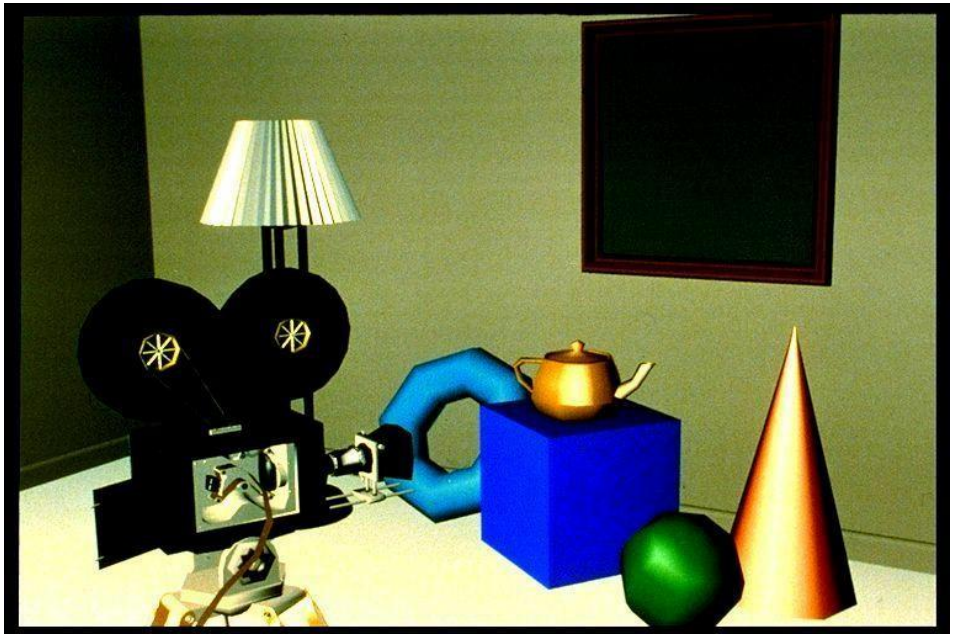


Flat→Gouraud→Phong Shading

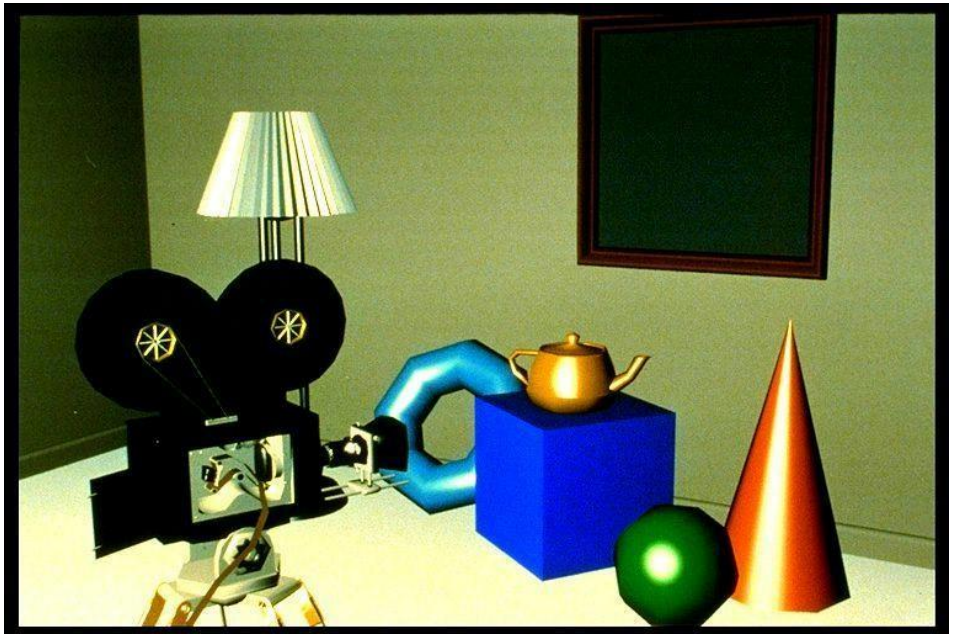
特点

- 通常会有效地降低Mach带效应
- 得到的图形比应用Gouraud方法的结果更光滑
- 但是由于法向的计算还是很复杂
  - 所花费时间通常是Gouraud方法的6到8倍
- OpenGL实现的是Gouraud方法

Gouraud v.s. Phong



Gouraud Surface Rendering



Phong Surface Rendering

比较

- 如果用多边形网格逼近大曲率曲面，Phong方法的结果可能看起来光滑一些，而Gouraud方法就会使边有些明显
- Phong方法比Gouraud方法的复杂度高
  - 可以用片段处理器实现
- 两种方法都需要特定数据结构表示网格，从而可以获取顶点法向