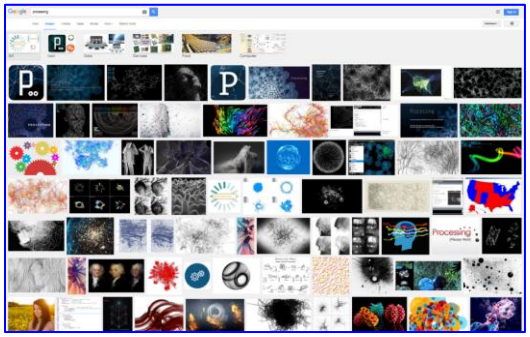
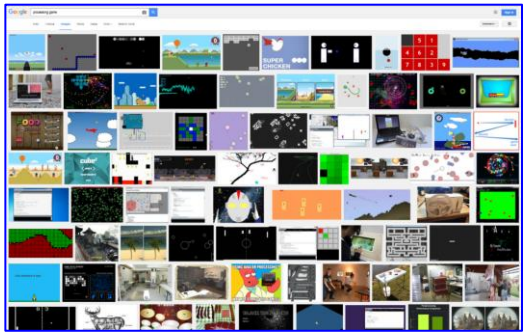


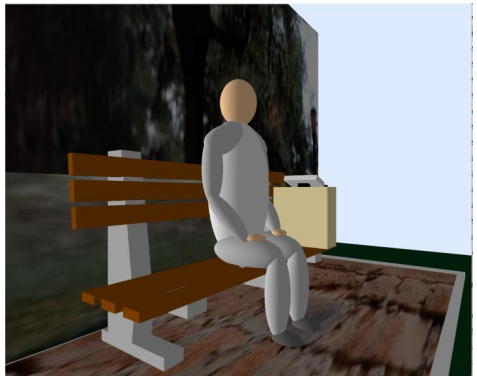


Lab 06

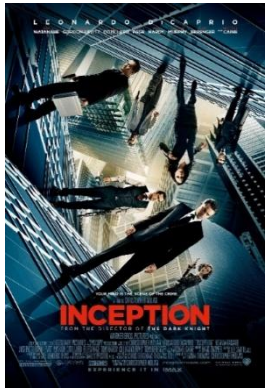
COMP 350

User Interface Design and Programming

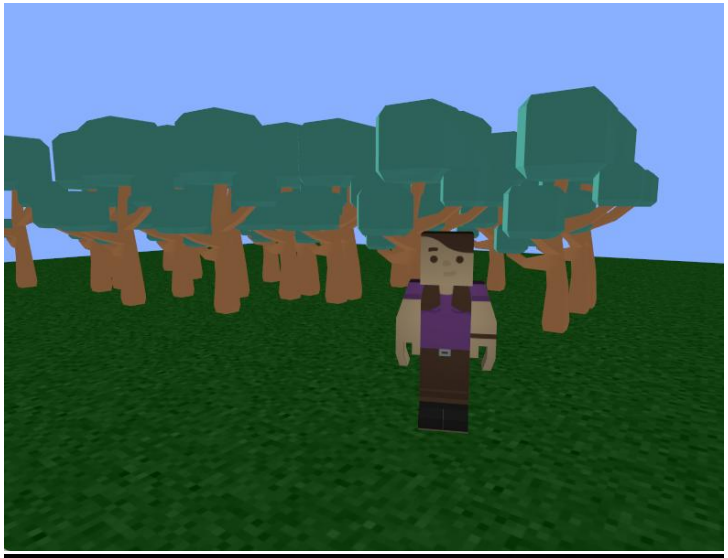
Instructor
Kyungjae Lee (just call me KJ)
KyungJae.Lee@ufv.ca

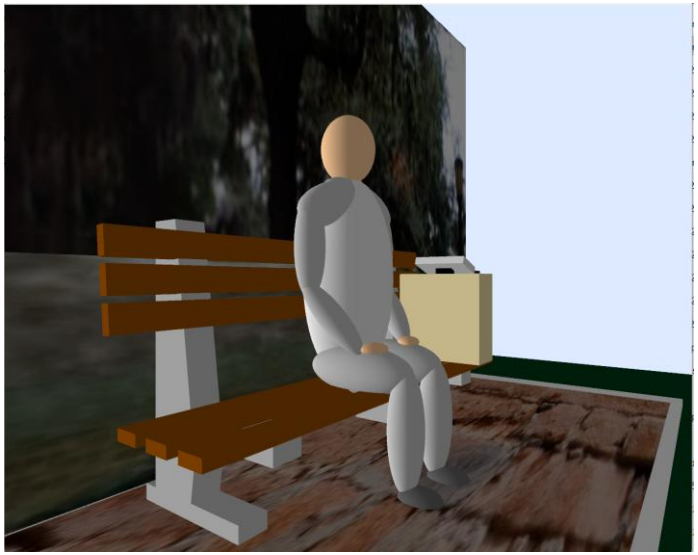


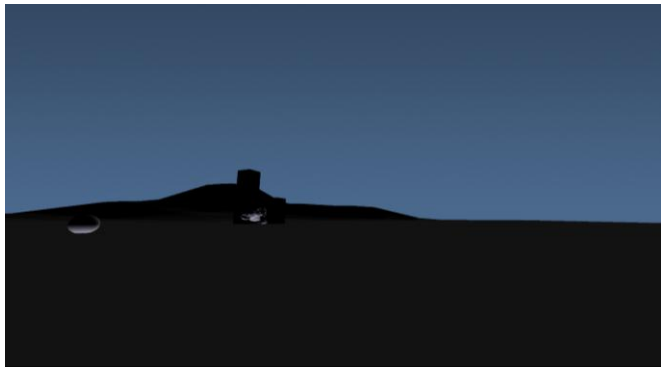
Lab 6
Imitate/parody/mimic a movie/painting/drama scene through 3D environment

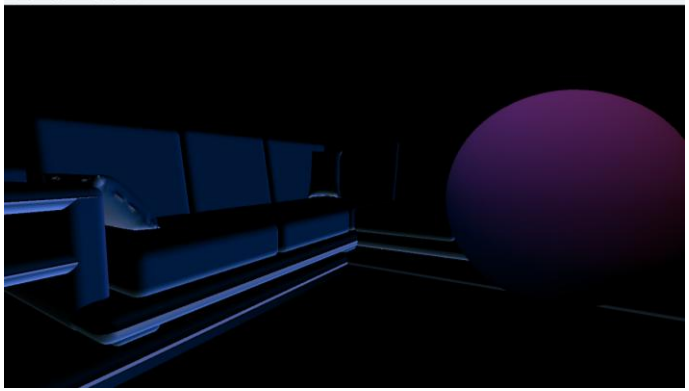


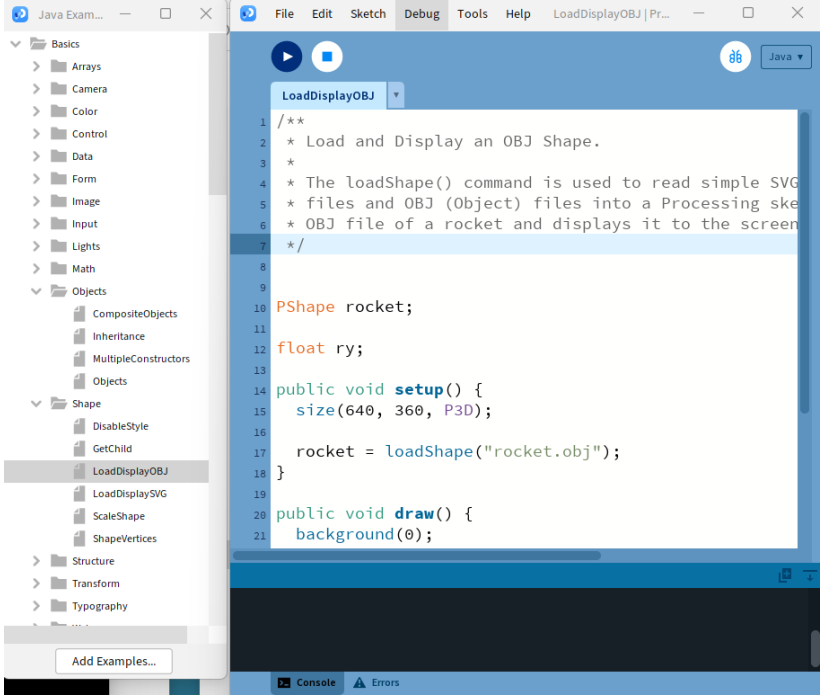
Examples from previous students

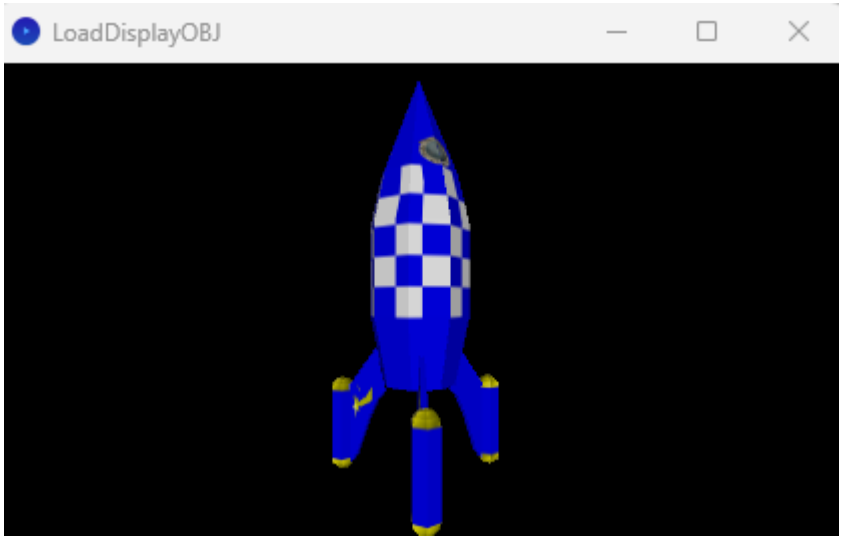


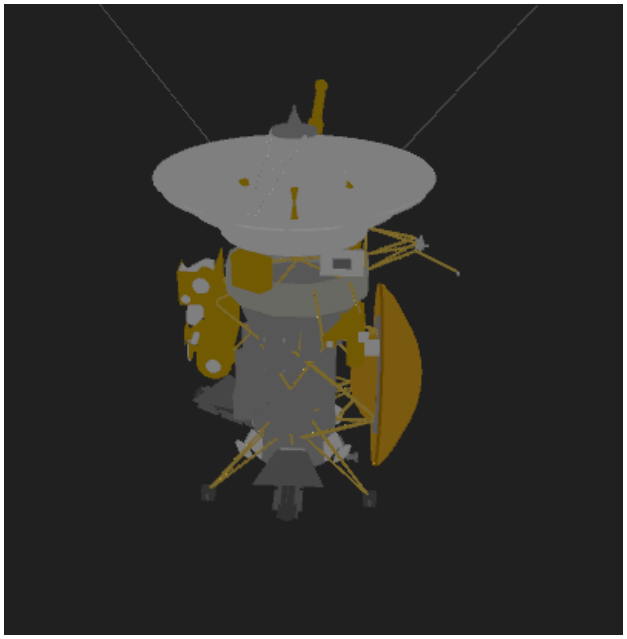


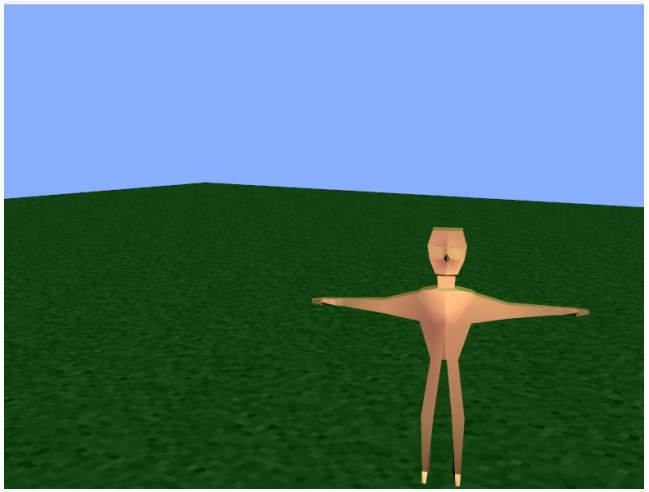


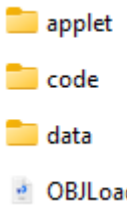






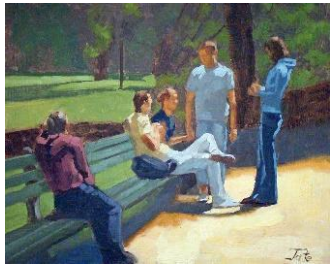


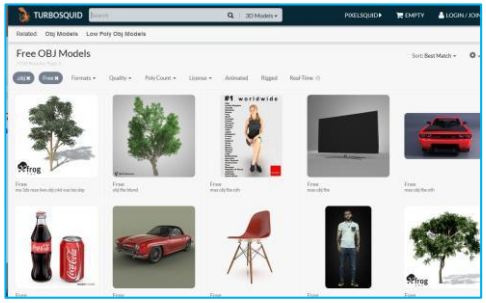


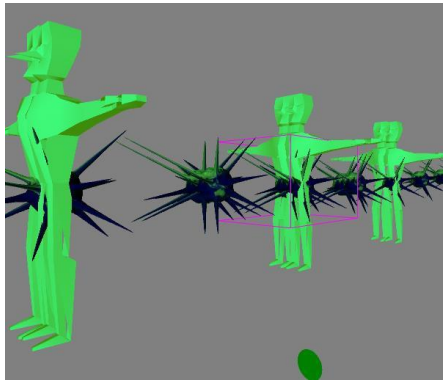


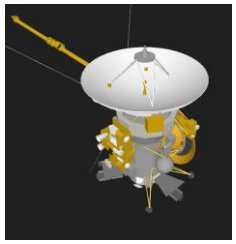
- This sample/documentation can be found here as well (obj loader sample).
- Drag/drop jar file (library > OBJLoader.jar) into Processing workspace.

Lab 6 assignment:
Mimic/parody a movie scene/panting through 3D environment









1) Find a reference image from media (e.g., movie, drama, anime, painting etc.). Fill out MS Word report form.

2) To recreate your reference scene using Processing, find minimum 3D models (e.g., 3D obj format) to roughly match scene complexity in addition to adding primitive shapes available in Processing (e.g., box, sphere, etc.) Multiple boxes can be used to create a character gesture. Feel free to download free 3d models from the following websites.

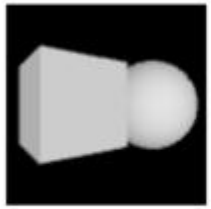
https://www.turbosquid.com/
https://nasa3d.arc.nasa.gov/models

3) Review Processing 3D example. box and sphere and OBJLoader libray

- Play with parameter values across camera, transformation, and color.
- Replace the current 3D object with other obj file format, and add more objects.
- Add multiple objects.
- Position your camera to capture matching shot to your reference image.
- No need for texture mapping.

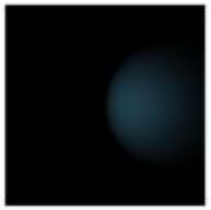
Lights https://processing.org/examples/

On/Off	ambientLight()
Directional	directionalLight()
Spot	pointLight()
Reflection	spotLight()
Mixture	noLights()
Mixture Grid	

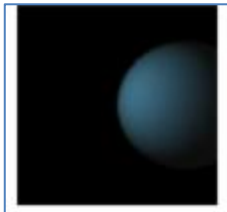


```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(126, 126, 126, 0, 0, -1);
ambientLight(102, 102, 102);
translate(32, 50, 0);
rotateY(PI/5);
box(40);
translate(60, 0, 0);
sphere(30);
```

```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(51, 102, 126, 0, -1, 0);
translate(80, 50, 0);
sphere(30);
```



```
size(100, 100, P3D);
int concentration = 600;  // Try 1 -> 10000
background(0);
noStroke();
spotLight(51, 102, 126, 50, 50, 400,
          0, 0, -1, PI/16, concentration);
translate(80, 50, 0);
sphere(30);
```

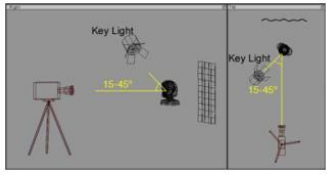


```
size(100, 100, P3D);
background(0);
noStroke();
pointLight(51, 102, 126, 35, 40, 36);
translate(80, 50, 0);
sphere(30);
```

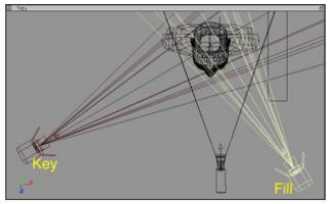
4) Review different types of lighting functions available in Processing.

5) Create two new files, and save it as Lab6a_pointLghting_yourLastNameInitial.pde' and Lab6b_lightingOutdoor_yourLastNameInitial.pde'

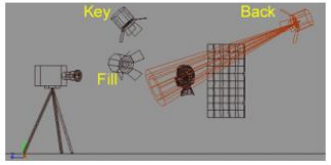
6) For 6a, Create 3D scene simulating 3 point lighting system representing indoor environment. Look for sample image (e.g., studio/portrait lighting), and choose one reference image (submit with your pde files). In addition to placing 3 lights, add both simple primitive shapes and realistic indoor object (e.g., furniture). Consider expressive mood from your scene.



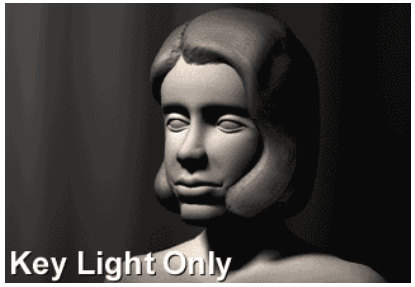
Key Light



Fill Light



Back Light

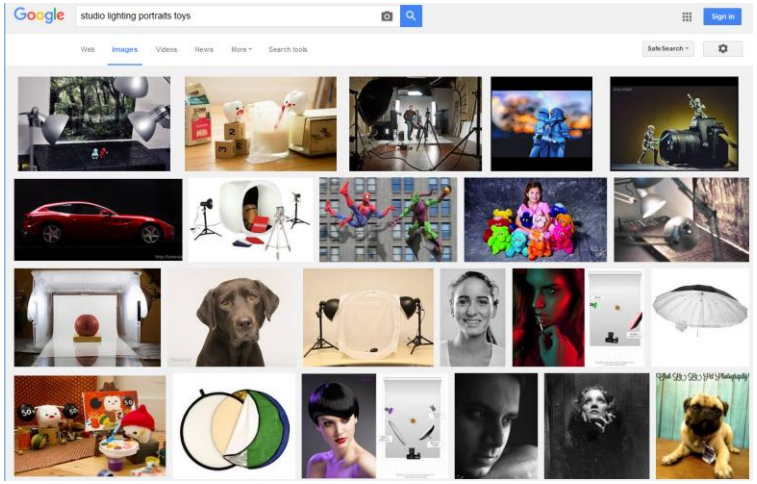


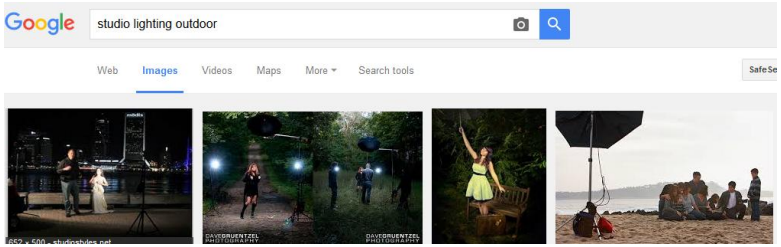
Reading material: Jeremy Birn http://www.3drender.com/light/3point.html

https://processing.org/reference/lights_.html

https://processing.org/examples/

ambientLight()
directionalLight()
pointLight()
spotLight()
noLights()





7) The 6.b aims for outdoor larger scale environment. Review some examples from 3D animation, game, or film (e.g., Minecraft) representing outdoor lighting. Pick one reference image (submit with your pde files), and try to re-create lighting environment (e.g., daytime, night time etc.).

8) Add some interactions (e.g., light on/off key, button, or slider)

- The focus is to simulate lighting, not modeling objects. Download free 3d modeling file (obj format) from the following sites;
  https://www.turbosquid.com/
- **https://nasa3d.arc.nasa.gov/models**

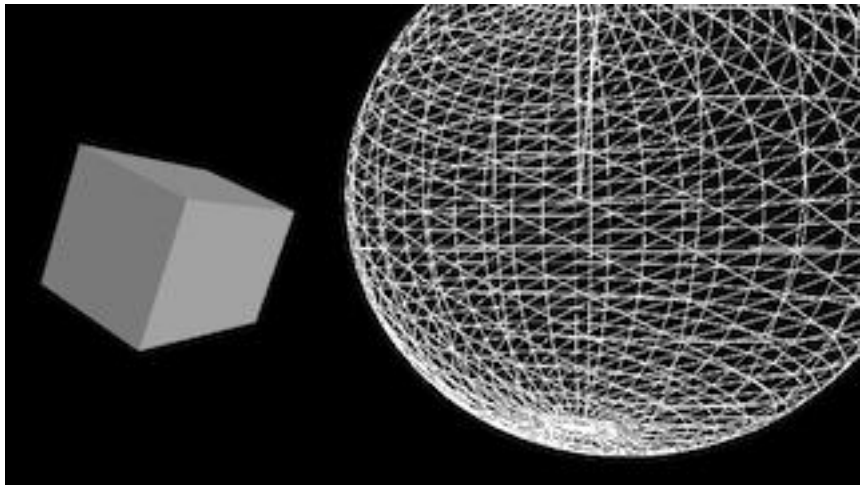
Your work should be different/unique from the examples.

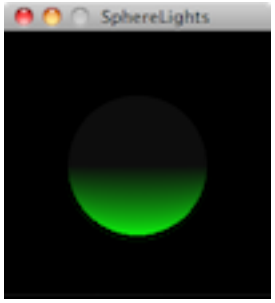
Outdoor Studio Lighting || UofA Men's Tennis Poster

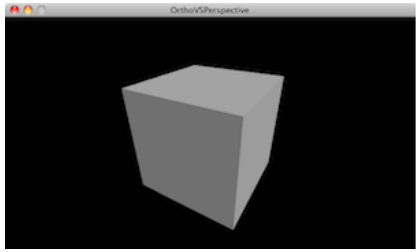
https://processing.org/reference/lights_.html

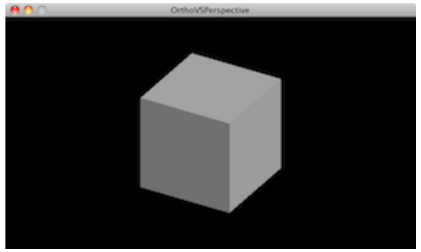
P3D

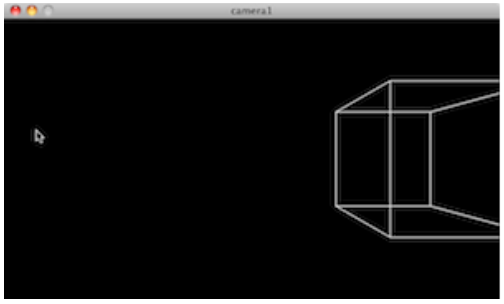
https://processing.org/tutorials/p3d/

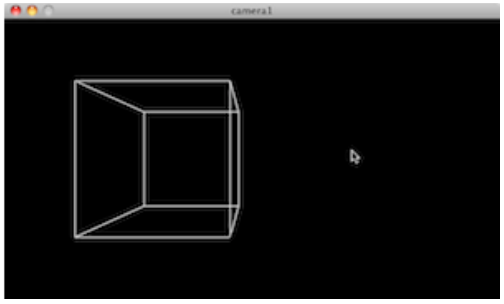








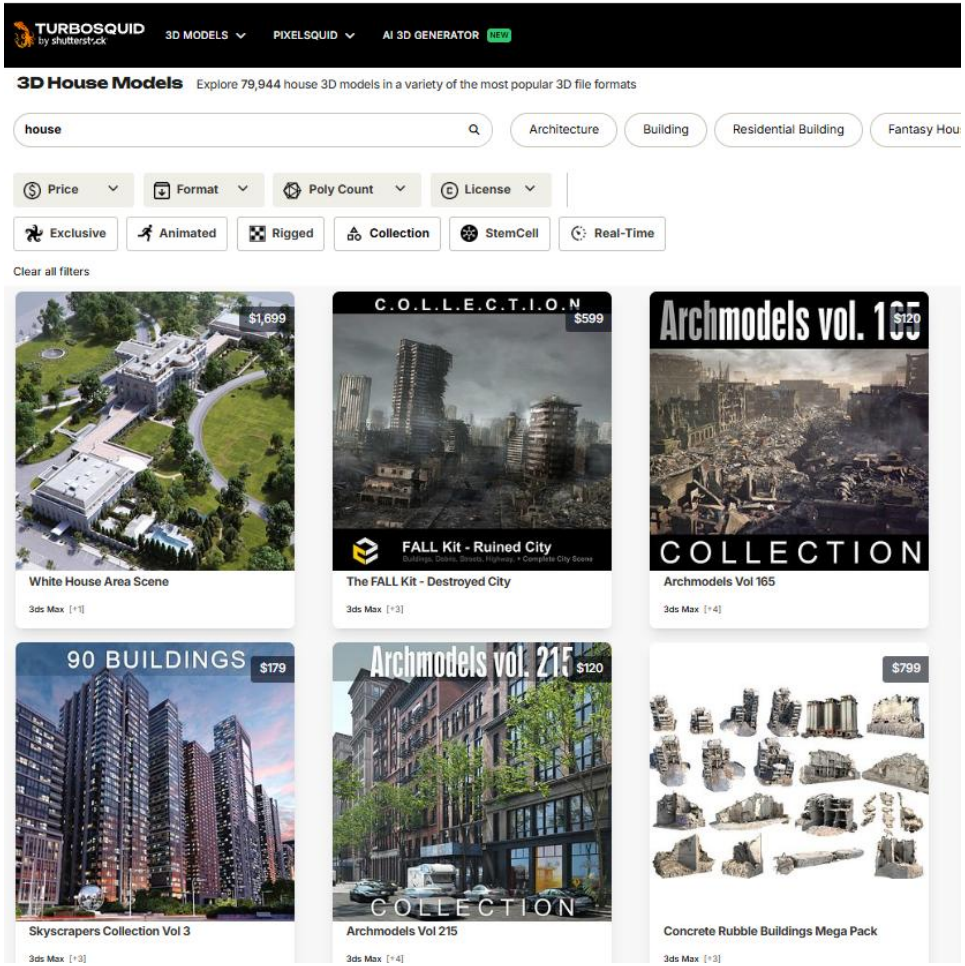


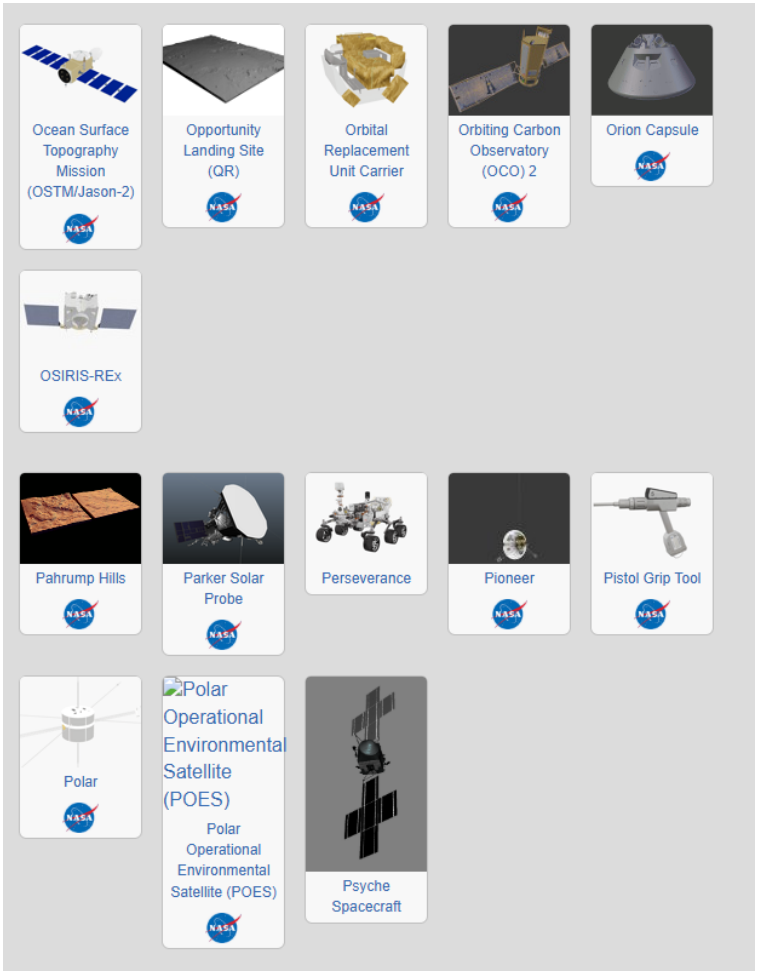


Download 3D models!

https://www.turbosquid.com/

https://nasa3d.arc.nasa.gov/models





camera()

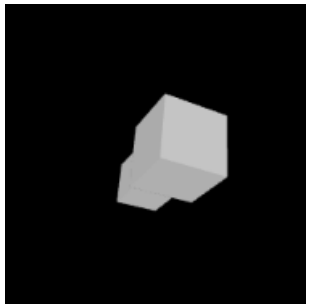
Perspective

Lights, Camera

Lights
ambientLight()
directionalLight()
lightFalloff()
lights()
lightSpecular()
noLights()
normal()
pointLight()
spotLight()

Camera
beginCamera()
camera()
endCamera()
frustum()
ortho()
perspective()
printCamera()
printProjection()

```
size(100, 100, P3D);
noFill();
background(204);
camera(70.0, 35.0, 120.0, 50.0, 50.0, 0.0,
       0.0, 1.0, 0.0);
translate(50, 50, 0);
rotateX(-PI/6);
rotateY(PI/3);
box(45);
```



```
camera()
camera(eyeX, eyeY, eyeZ, centerX, centerY, centerZ, upX, upY, upZ)
```

eyeX	float: x-coordinate for the eye
eyeY	float: y-coordinate for the eye
eyeZ	float: z-coordinate for the eye
centerX	float: x-coordinate for the center of the scene
centerY	float: y-coordinate for the center of the scene
centerZ	float: z-coordinate for the center of the scene
upX	float: usually 0.0, 1.0, or -1.0
upY	float: usually 0.0, 1.0, or -1.0
upZ	float: usually 0.0, 1.0, or -1.0

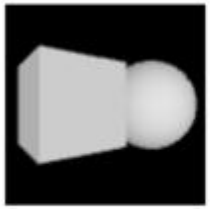
```
perspective()
perspective(fovy, aspect, zNear, zFar)
```

fovy	(float)	field-of-view angle (in radians) for vertical direction
aspect	(float)	ratio of width to height
zNear	(float)	z-position of nearest clipping plane
zFar	(float)	z-position of the farthest clipping plane

```
void
```

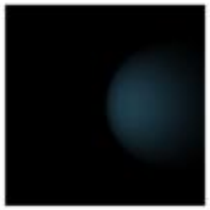
Lights https://processing.org/examples/

On/Off ambientLight()
Directional directionalLight()
Spot pointLight()
Reflection spotLight()
Mixture noLights()
Mixture Grid

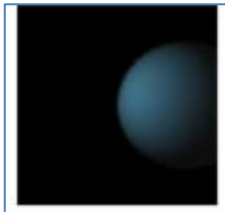


```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(126, 126, 126, 0, 0, -1);
ambientLight(102, 102, 102);
translate(32, 50, 0);
rotateY(PI/5);
box(40);
translate(60, 0, 0);
sphere(30);
```

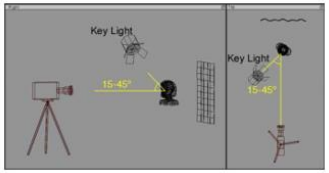
```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(51, 102, 126, 0, -1, 0);
translate(80, 50, 0);
sphere(30);
```



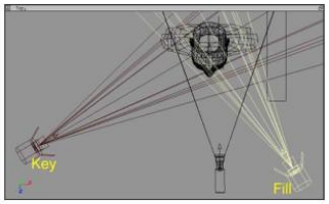
```
size(100, 100, P3D);
int concentration = 600;  // Try 1 -> 10000
background(0);
noStroke();
spotLight(51, 102, 126, 50, 50, 400,
          0, 0, -1, PI/16, concentration);
translate(80, 50, 0);
sphere(30);
```



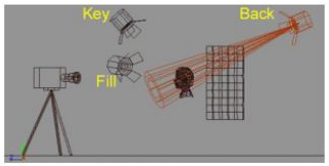
```
size(100, 100, P3D);
background(0);
noStroke();
pointLight(51, 102, 126, 35, 40, 36);
translate(80, 50, 0);
sphere(30);
```



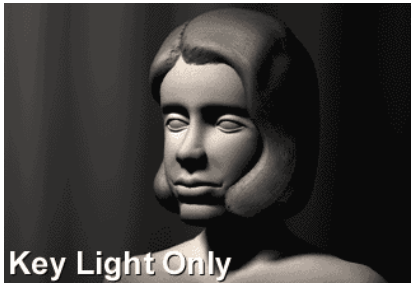
Key Light



Fill Light



Back Light



Reading material: Jeremy Birn
http://www.3drender.com/light/3point.html

https://processing.org/reference/lights_.html

directionalLight(v1, v2, v3, nx, ny, nz)

http://processing.org/reference/directionalLight_.html

```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(51, 102, 126, -1, 0, 0);
translate(20, 50, 0);
sphere(30);
```

```
size(100, 100, P3D);
background(0);
noStroke();
directionalLight(51, 102, 126, 0, -1, 0);
translate(80, 50, 0);
sphere(30);
```

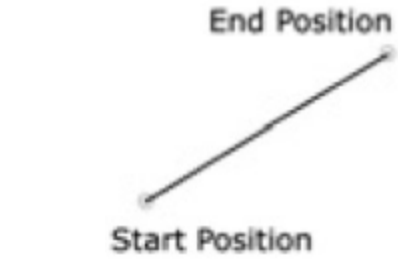
Adds a directional light. Directional light comes from one direction: it is stronger when hitting a surface squarely, and weaker if it hits at a gentle angle. After hitting a surface, directional light scatters in all directions. Lights need to be included in the `draw()` to remain persistent in a looping program. Placing them in the `setup()` of a looping program will cause them to only have an effect the first time through the loop. The `v1`, `v2`, and `v3` parameters are interpreted as either RGB or HSB values, depending on the current color mode. The `nx`, `ny`, and `nz` parameters specify the direction the light is facing. For example, setting `ny` to -1 will cause the geometry to be lit from below (since the light would be facing directly upward).

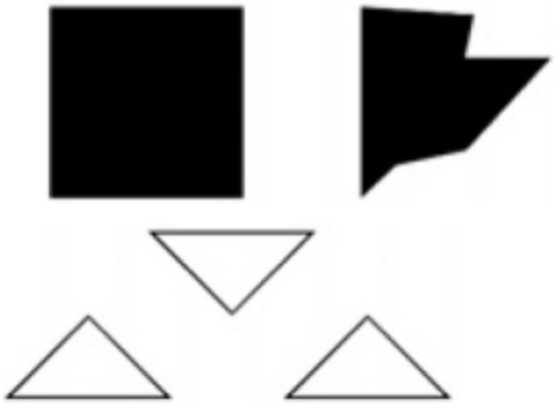
Syntax `directionalLight(v1, v2, v3, nx, ny, nz)`

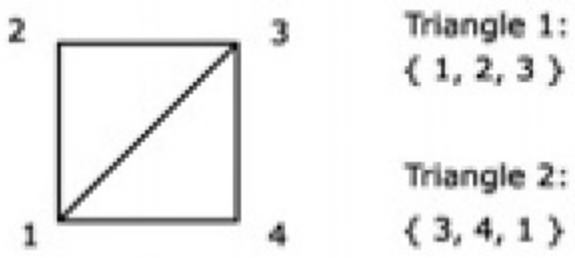
Parameters

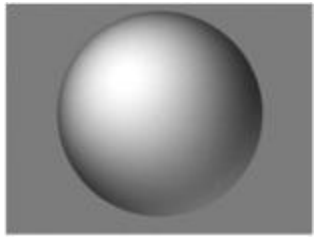
v1	float: red or hue value (depending on current color mode)
v2	float: green or saturation value (depending on current color mode)
v3	float: blue or brightness value (depending on current color mode)
nx	float: direction along the x-axis
ny	float: direction along the y-axis
nz	float: direction along the z-axis

Point, Lines & Primitives









- Point: a location in space plus its color
- Scalar: a number that we multiply other variables by
- Vector: a line that has magnitude, direction and a color

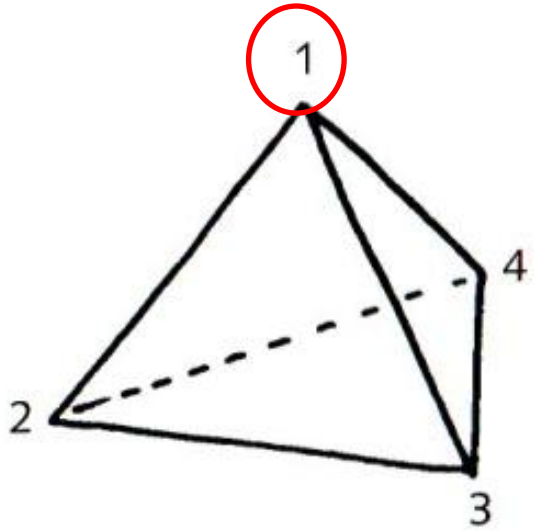
Source: OpenGL graphics through applications / Robert Whitrow

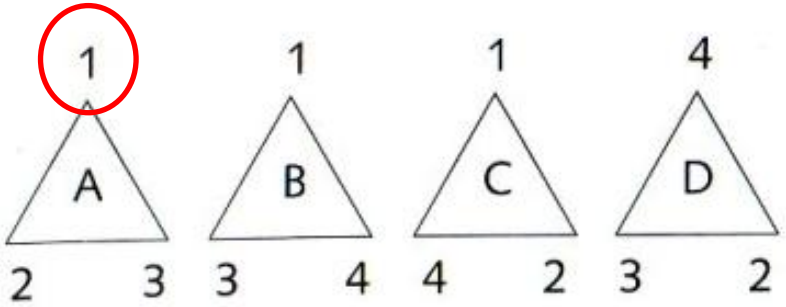
```
struct Point
{
        int x;
        int y;
}

struct Line
{
        Point start;
        Point end;
 }
struct Polygon
{
      int total_points;
      array<Point> points;
}
```

```
struct Vector3D
{
        float x, y, z;
};

struct Vector3D
{
        float pos[3];
}
struct Sphere
{
                int radius;
                Point position;
}
```

Source: Game Graphics Programmingby Allen Sherrod

Description of an Object into data format

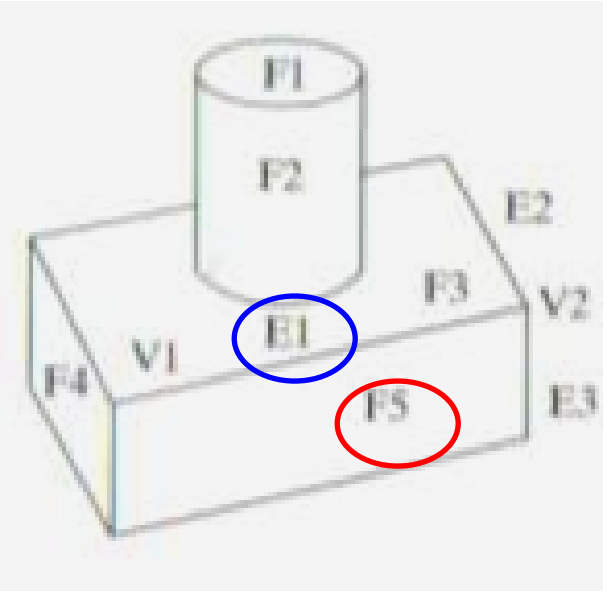




Geometry Format #1

Vertex	X	Y	Z
1	0	0	0
2	-1	-2	1
3	1	-2	1
4	0	-2	-1

Facet	Vertex	Vertex	Vertex
A	1	2	3
B	1	3	4
C	1	4	2
D	3	4	2



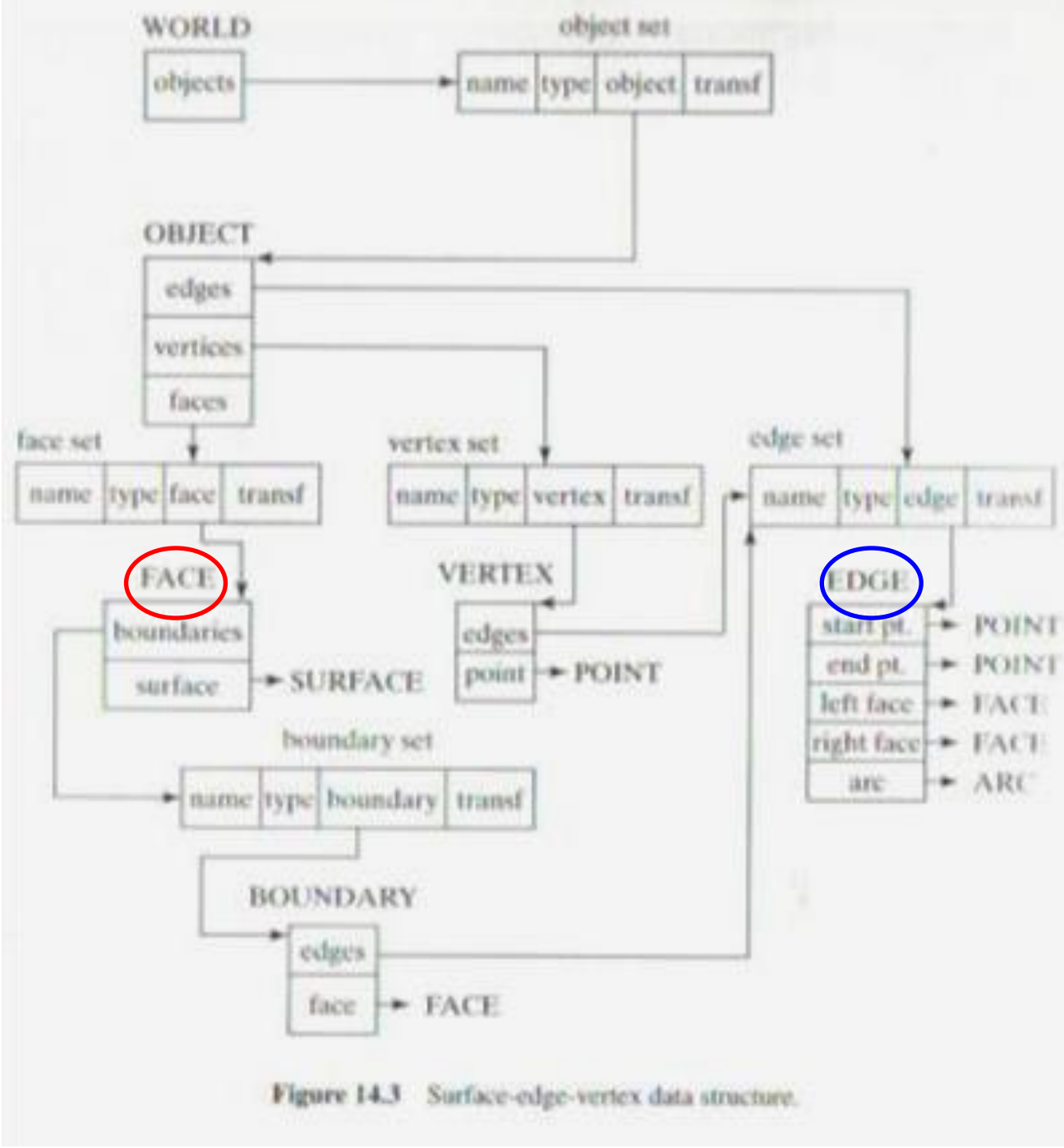


Figure 14.3 Surface-edge-vertex data structure.

(Kerlow,2004)

3D file format example (Obj format)

OBJ (or .OBJ) is a geometry definition file format first developed by Wavefront Technologies for its Advanced Visualizer animation package. The file format is open and has been adopted by other 3D graphics application vendors. For the most part it is a universally accepted format.

The OBJ file format is a simple data-format that represents 3D geometry alone — namely, the position of each vertex, the UV position of each texture coordinate vertex, vertex normals, and the faces that make each polygon defined as a list of vertices, and texture vertices. Vertices are stored in a counter-clockwise order by default, making explicit declaration of face normals unnecessary. OBJ coordinates have no units, but OBJ files can contain scale information in a human readable comment line.

http://www.fileformat.info/format/wavefrontobj/egff.htm

```
3  mtllib cube.mtl
4  g default
5  v 0.000000 0.000000 1.000000
6  v 1.000000 0.000000 1.000000
7  v 0.000000 1.000000 1.000000
8  v 1.000000 1.000000 1.000000
9  v 0.000000 1.000000 0.000000
10 v 1.000000 1.000000 0.000000
11 v 0.000000 0.000000 0.000000
12 v 1.000000 0.000000 0.000000
13 vt 0.375000 0.000000
14 vt 0.625000 0.000000
15 vt 0.375000 0.250000
16 vt 0.625000 0.250000
17 vt 0.375000 0.500000
18 vt 0.625000 0.500000
19 vt 0.375000 0.750000
20 vt 0.625000 0.750000
21 vt 0.375000 1.000000
22 vt 0.625000 1.000000
23 vt 0.875000 0.000000
24 vt 0.875000 0.250000
25 vt 0.125000 0.000000
26 vt 0.125000 0.250000
27 vn 0.000000 0.000000 1.000000
28 vn 0.000000 0.000000 1.000000
29 vn 0.000000 0.000000 1.000000
30 vn 0.000000 0.000000 1.000000
31 vn 0.000000 1.000000 0.000000
32 vn 0.000000 1.000000 0.000000
33 vn 0.000000 1.000000 0.000000
```

Obj format

The **OBJ file extension** is commonly used for a standardized files that contain 3D objects. Usually a simple data-format that represents only the 3D geometry, such as the position of each vertex, the texture coordinate associated with a vertex, the normal at each vertex, and the faces that make each polygon. Sometimes also known as Wavefront 3D objects.

http://www.file-extensions.org/obj-file-extension

http://en.wikipedia.org/wiki/Wavefront_.obj_file

```
3  mtllib cube.mtl
4  g default
5  v 0.000000 0.000000 1.000000
6  v 1.000000 0.000000 1.000000
7  v 0.000000 1.000000 1.000000
8  v 1.000000 1.000000 1.000000
9  v 0.000000 1.000000 0.000000
10 v 1.000000 1.000000 0.000000
11 v 0.000000 0.000000 0.000000
12 v 1.000000 0.000000 0.000000
13 vt 0.375000 0.000000
14 vt 0.625000 0.000000
15 vt 0.375000 0.250000
16 vt 0.625000 0.250000
17 vt 0.375000 0.500000
18 vt 0.625000 0.500000
19 vt 0.375000 0.750000
20 vt 0.625000 0.750000
21 vt 0.375000 1.000000
22 vt 0.625000 1.000000
23 vt 0.875000 0.000000
24 vt 0.875000 0.250000
25 vt 0.125000 0.000000
26 vt 0.125000 0.250000
27 vn 0.000000 0.000000 1.000000
28 vn 0.000000 0.000000 1.000000
29 vn 0.000000 0.000000 1.000000
30 vn 0.000000 0.000000 1.000000
31 vn 0.000000 1.000000 0.000000
32 vn 0.000000 1.000000 0.000000
33 vn 0.000000 1.000000 0.000000
```

Vertex data:

v	Geometric vertices:	v *x y z*
vt	Texture vertices:	vt *u v*
vn	Vertex normals:	vn *dx dy dz*

Elements:

p	Point:	p v_1
l	Line:	l v_1 v_2 ... v_n
f	Face:	f v_1 v_2 ... v_n
f	Face with texture coords:	f v_1/t_1 v_2/t_2 .... v_n/t_n
f	Face with vertex normals:	f $v_1//n_1$ $v_2//n_2$ .... $v_n//n_n$
f	Face with txt and norms:	f $v_1/t_1/n_1$ $v_2/t_2/n_2$ .... $v_n/t_n/n_n$

Grouping:

g	Group name:	g *groupname*

Display/render attributes:

usemtl	Material name:	usemtl *materialname*
mtllib	Material library:	mtllib *materiallibname*.mtl

http://people.cs.clemson.edu/~dhouse/courses/405/docs/brief-obj-file-format.html

How to Use World's Most Advanced GPUs to Bust an Apollo 11 Conspiracy Theory (Video)

https://www.nvidia.com/coolstuff/demos#!/apollo-11

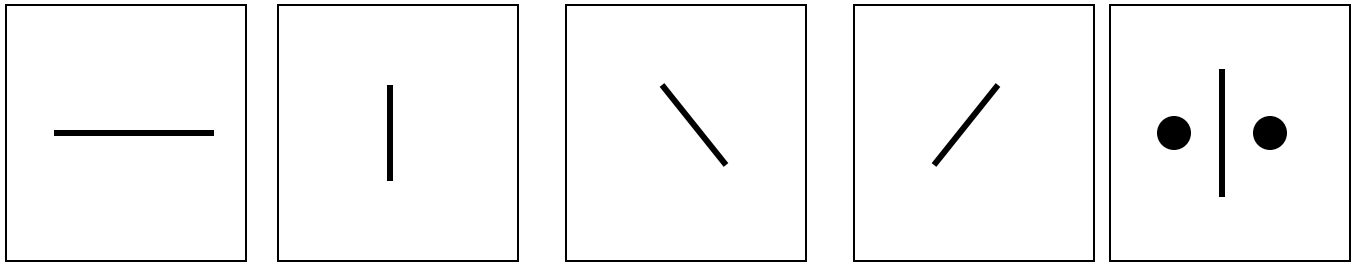
Scene Composition !

(*Think like an art director*! What is the purpose of shot? Good composition?)

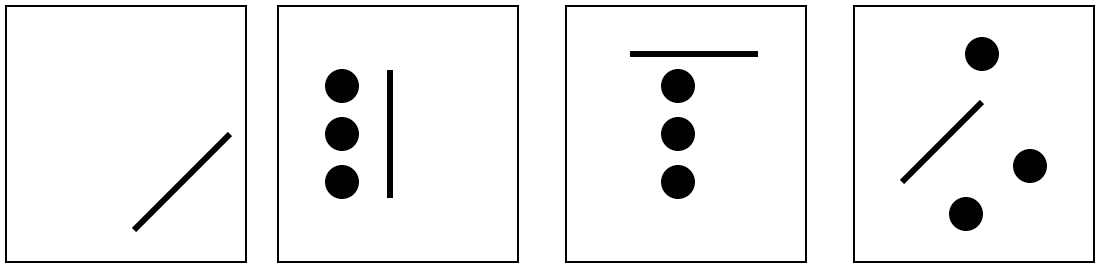
- **Scene Elements (i.e., Character[3], Environment[2])**
  - Portrait (i.e., statue, discussion)
  - Objects (i.e.,, Cups on the table, cars)
  - Scenery (i.e., Playground, mountain, woods)
- **Composition**
  - Balance
  - Shot Flow, direction
  - Camera angle, Perspective Projection
  - Types of Shots (Close Up, Medium Shot, Full Shot)
- **Action**
  - Static
  - Active, passive

Balance can create mood

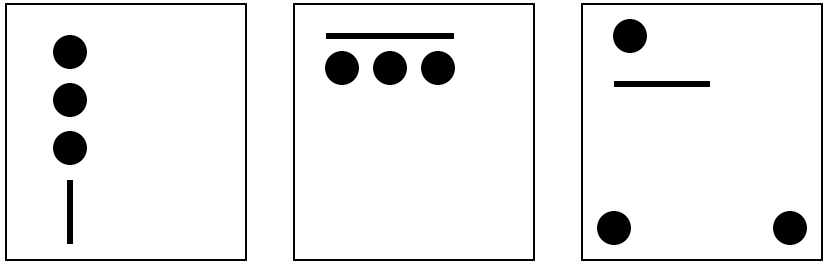
- Symmetry: Absolute/pure balance or different feeling of balance



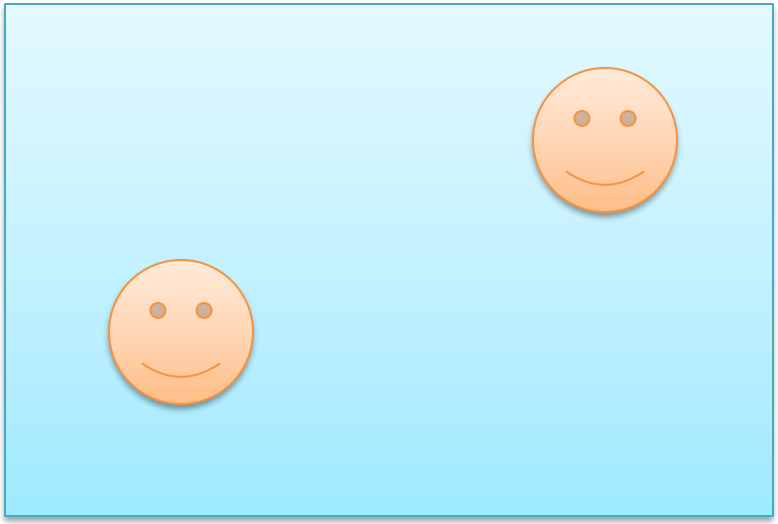
- Asymmetrical Balance: By evenly distributing total mass



- Tension: Sometimes effective design purposely unbalanced



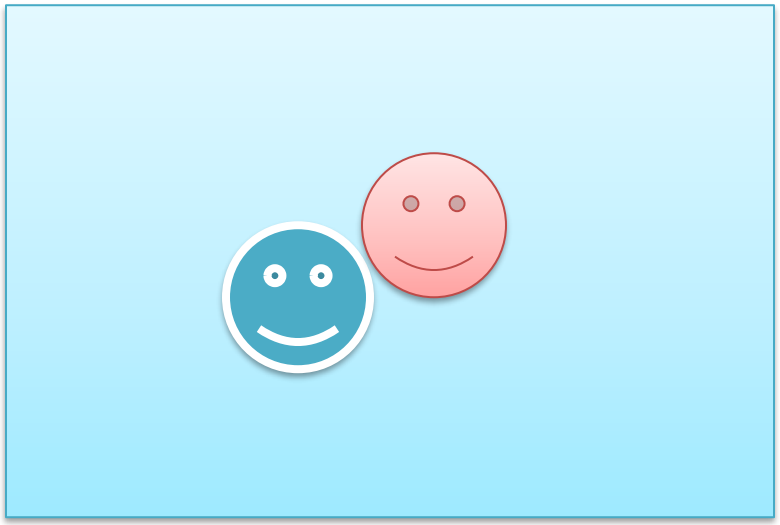
Source: *Using Design Basics to Get Creative Results* by Bryan Peterson

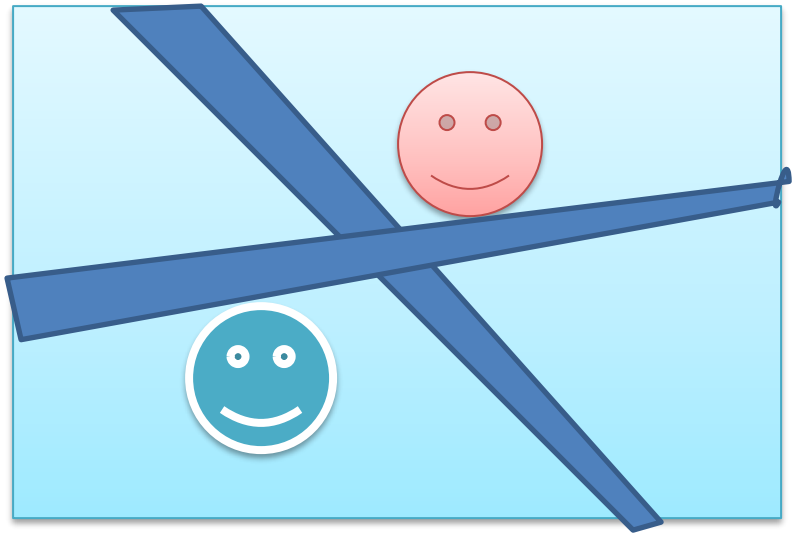


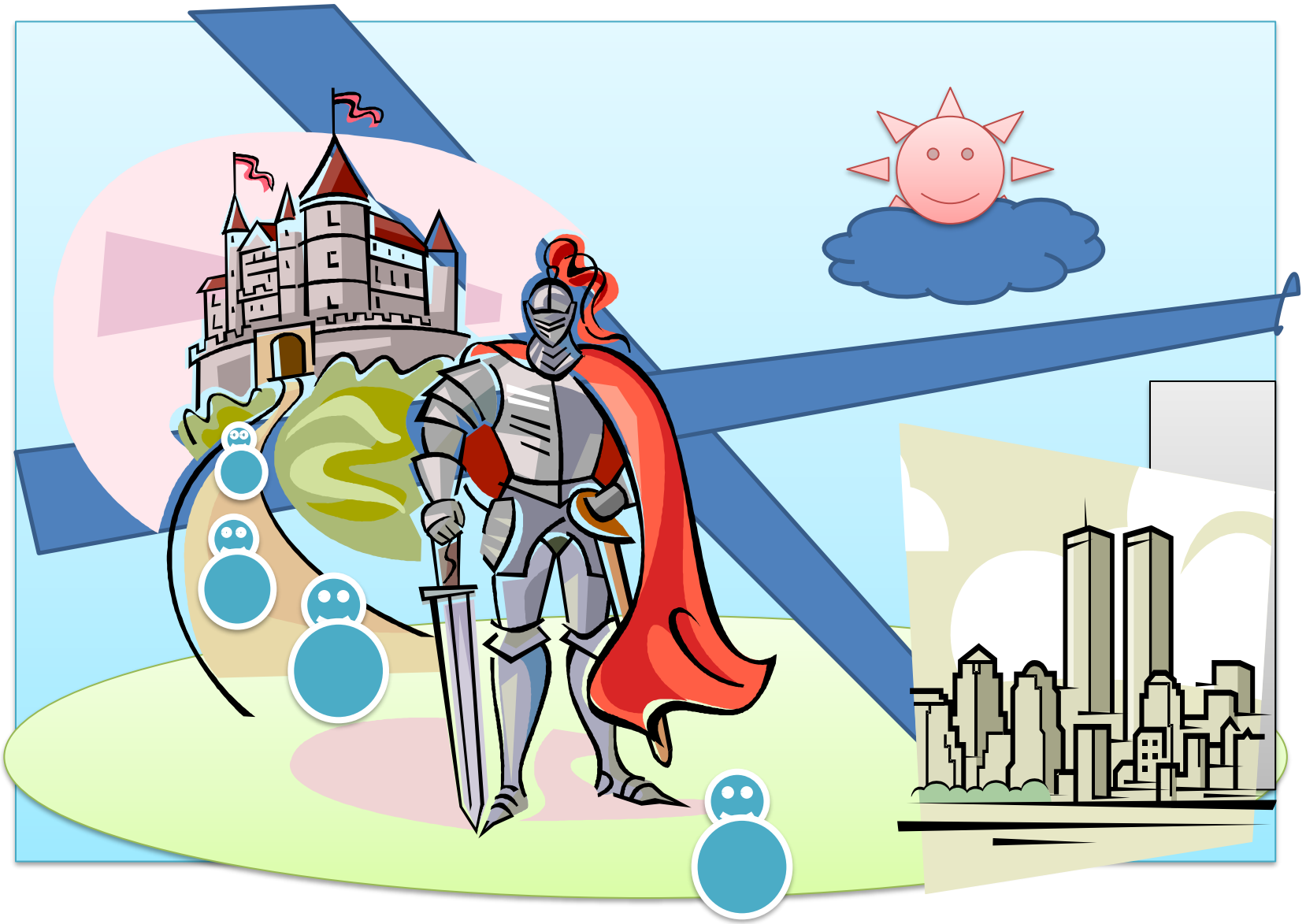
- Every element in a design has personal space. Too far apart and tension is lost.
- Too close and even a simple design seems crowded.
- Picture personal space as a group of physical elements to be avoided and worked around.

Source: Mark Gonyea

http://www.markgonyea.com/







Balance of Positive and Negative space

Mostly Negative Space

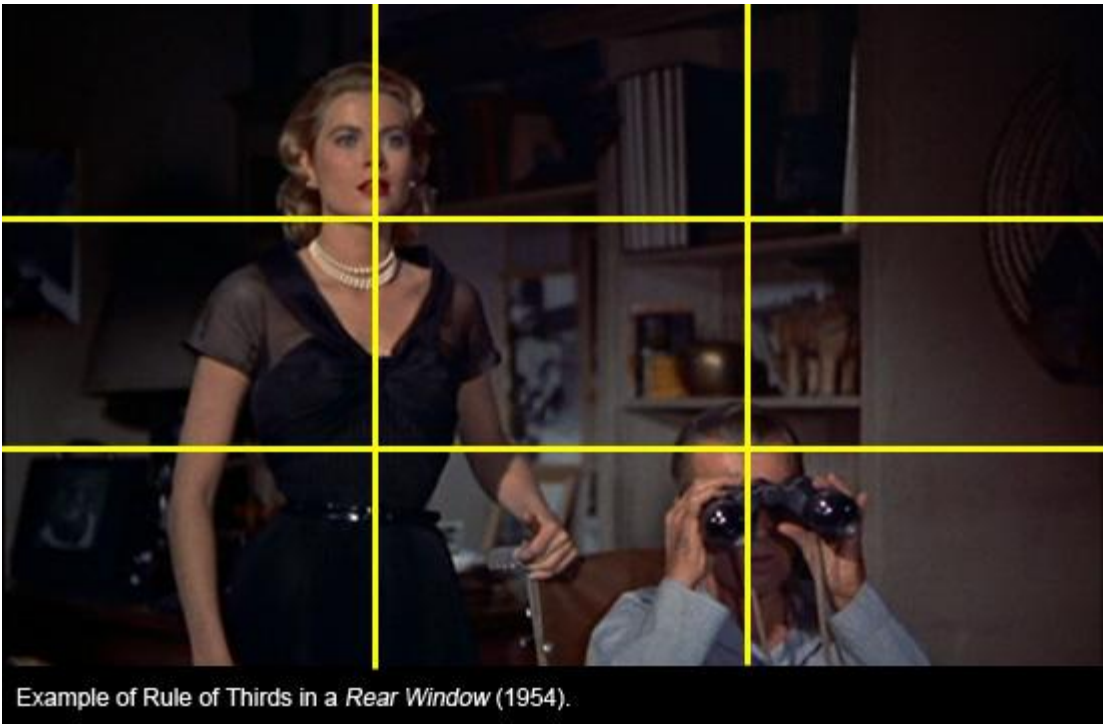
Mostly Positive Space

http://thevirtualinstructor.com/positive-and-negative-space.html

Negative space, in art, is the space around and between the subject(s) of an image. Negative space may be most evident when the space around a subject, and not the subject itself, forms an interesting or artistically relevant shape, and such space is occasionally used to artistic effect as the "real" subject of an image.

Rubin's vase is an optical illusion in which the negative space around the vase forms the silhouettes of two faces in profile, a well-known example of figure-ground reversal.

http://en.wikipedia.org/wiki/Negative_space


Example of Rule of Thirds in a *Rear Window* (1954).

Rule of Thirds

http://youtu.be/HZRXMLoCFrQ

Directing viewer's attention on one of intersection points.

To follow it, one must imagine the frame with two vertical lines and two horizontal lines, as to create three vertical sections of the same dimensions and three vertical sections also of the same size. The result is something like this:

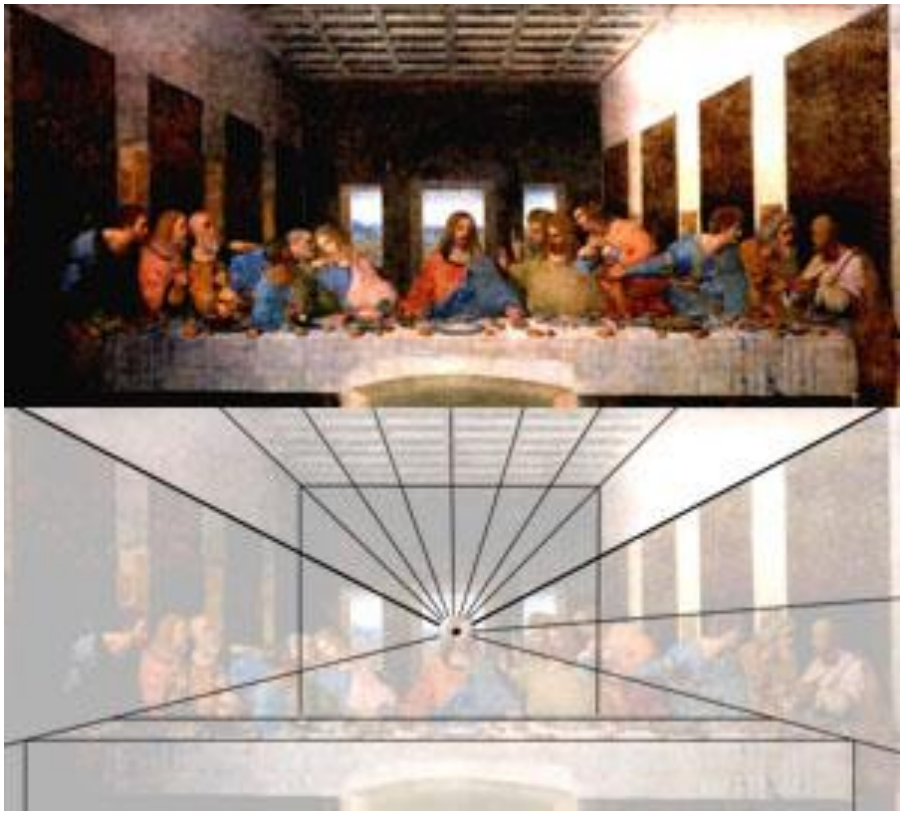
The intersections of the lines are points of interest, where important objects are often placed. These points of interest are comfortable to the eye, thus the middle portion of the frame are kept "empty" or clear.

http://www.elementsofcinema.com/cinematography/composition.html

Rules of Composition

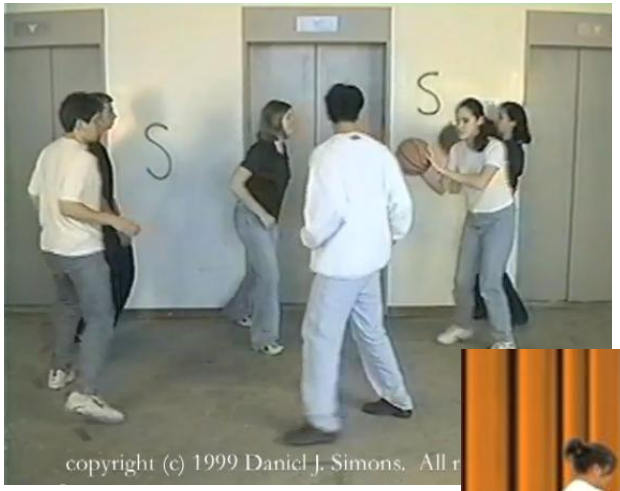
- Avoid symmetry
- Don’t paint objects flat-on to the viewer
- Follow the principles of balance
- Obey the rule of thirds
- Understand the golden mean
  - (Mattingly, Digital Matte Painting Handbook)

Is ‘Rule of Thirds’ absolutely necessary on every single shot/scene?



http://2draw.net/wiki/One-Point_Perspective

Complexity of Scene Composition & Visual Attention



- Selective Attention
- Change Blindness
- Attention Shift

http://www.theinvisiblegorilla.com/videos.html

the invisible gorilla

Christopher Chabris and Daniel Simons

Composition
Where subject should be placed on?

It depends on

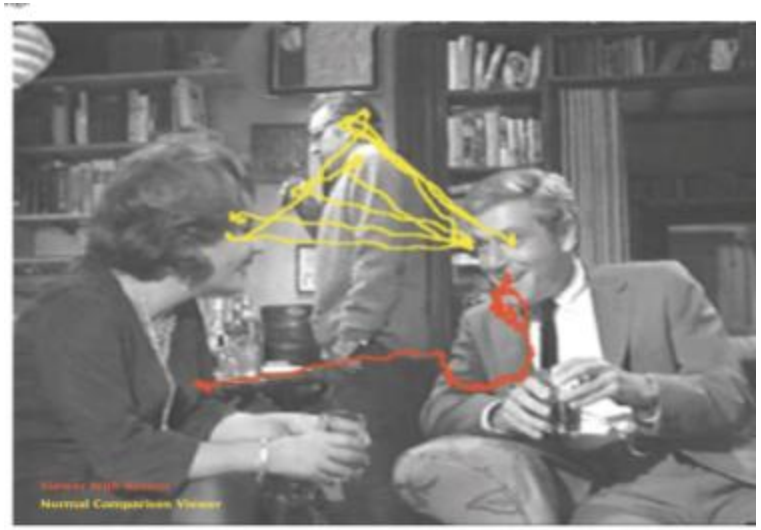
story, dialogue,

lighting situation,

surroundings

Flow of the scene

Your envision

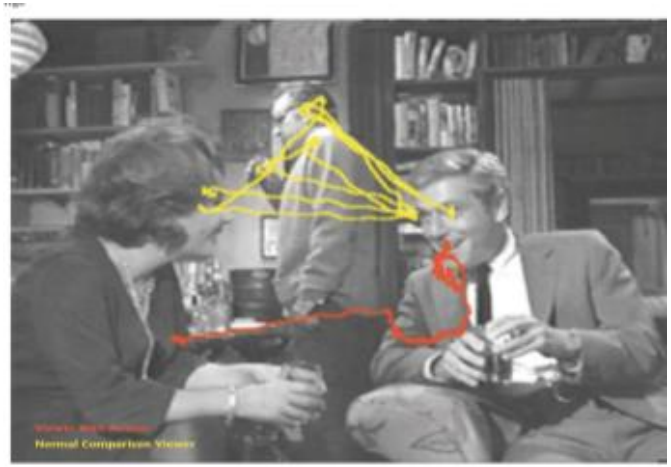


Source: Klin et al. (2002)
Yale University School of Medicine

Rembrandt

Composition

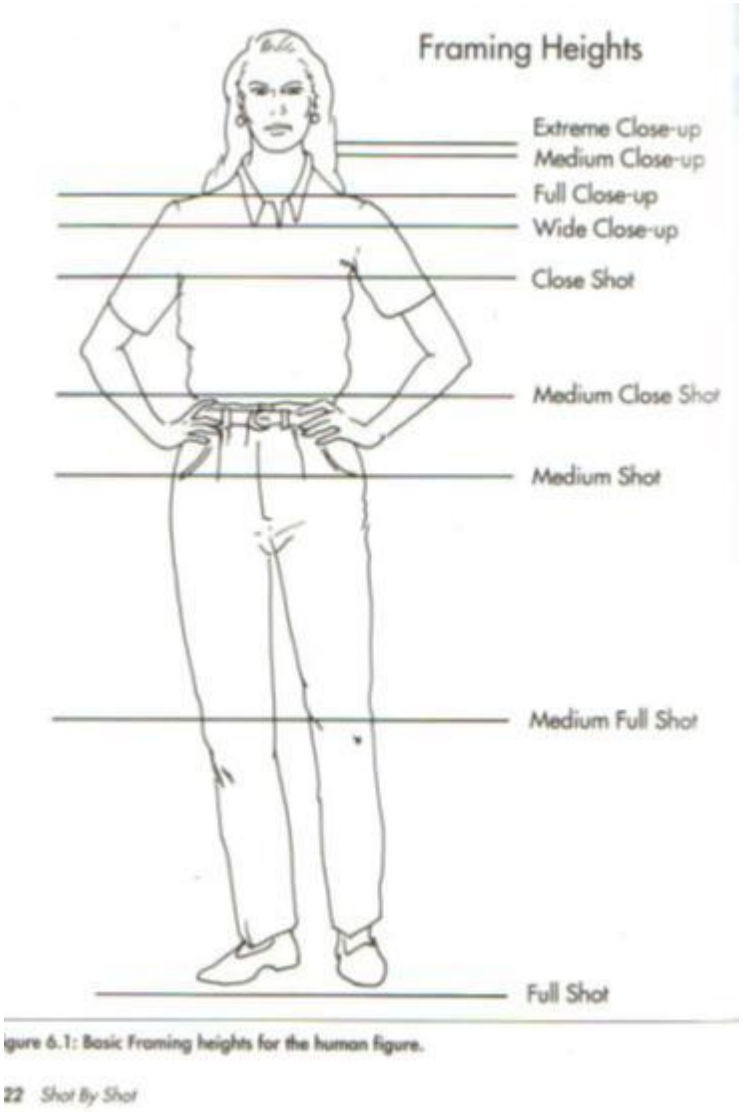
Message, mood?



Source: Klin et al. (2002)
Yale University School of Medicine

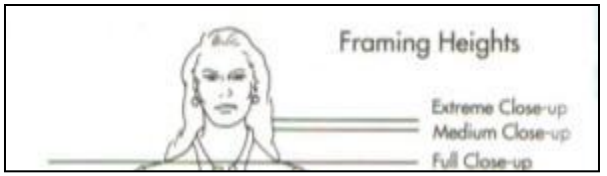
http://www.theinvisiblegorilla.com/videos.html

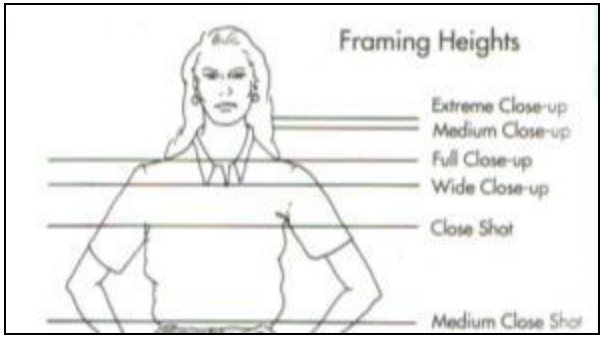
- Is there an interaction between characters (or objects)?
- Is it passive or active? Or emotional or technical exposure?
- Can you identify sender/receiver of message/story?
- Should we put more focus on sender or receiver by camera angle/location or manipulate shape/motion of the character?
- Where is your focus of interest? Should it be located at the center, upper left, 1/3 of the scene?



igure 6.1: Basic Framing heights for the human figure.

22 *Shot By Shot*

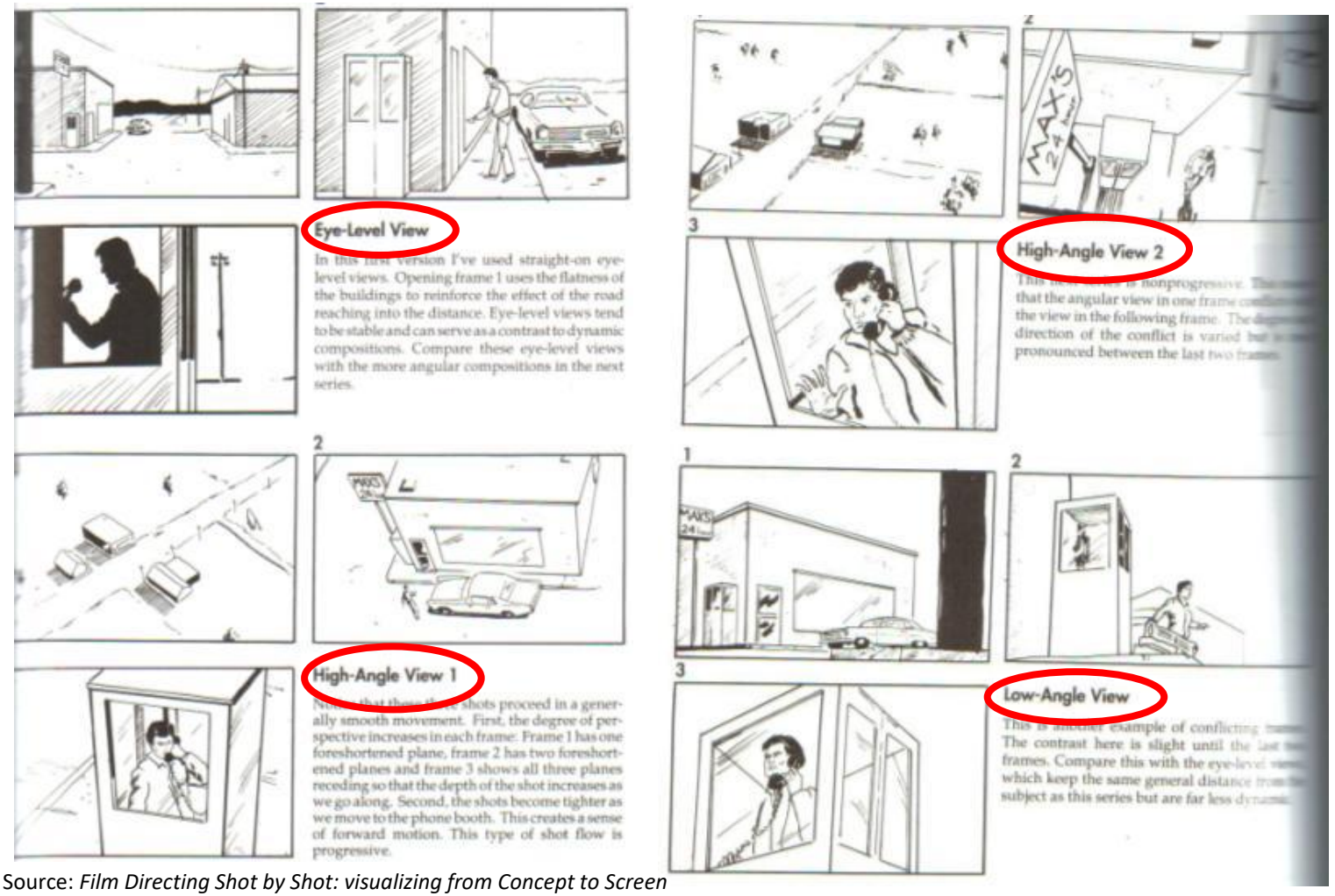




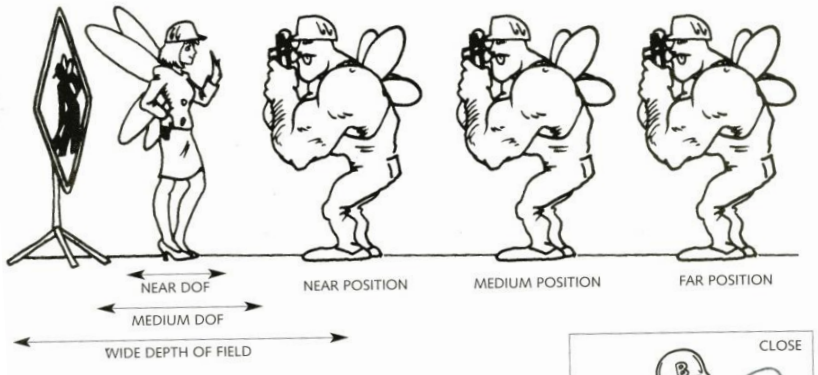
Source: *Film Directing Shot by Shot: visualizing from Concept to Screen* by Steven Katz

Camera Height: Involvement & Power

1. Which view brings a character to have more power (or less power)?
2. Which view brings a viewer to feel more powerful (or less powerful)?
3. Which view brings an object more focused (or global view)?

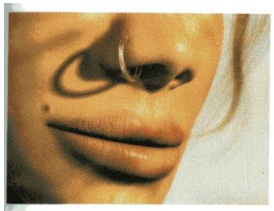


Source: *Film Directing Shot by Shot: visualizing from Concept to Screen* by Steven Katz



CHANGING CAMERA DISTANCE

7.4.4 Both the size of the image area and the depth of field increase when a camera moves away from a fixed subject.



EXTREME CLOSE-UP

CLOSE-UP

MEDIUM CLOSE-UP

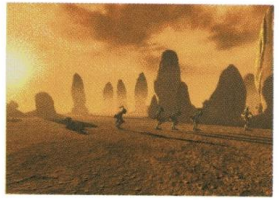
WAIST

KNEE

MEDIUM

WIDE

MEDIUM LONG



LONG

POINT OF VIEW

LOW ANGLE

HIGH ANGLE

3D Lighting

Mood: B/W High contrast

http://oxcgn.com/2012/05/26/top-10-franchises-that-should-be-announced-as-lego-games-at-e3-2012/

http://www.google.ca/imgres?q=james+bond&start=261&hl=en&rlz=1C1AFAB_enCA449CA449&biw=989&bih=590&addh=36&tbm=isch&tbnid=BY6aks9oOTJoRM:&imgrefurl=http://www.karenmeetscelebrities.com/id39.html&docid=IEsGOfEV4f03mM&imgurl=http://i157.photobucket.com/albums/t67/ryu_036/james_bond_collection.jpg&w=405&h=400&ei=uh93UIWuBKKMiALqzoCgCg&zoom=1&iact=hc&vpx=723&vpy=113&dur=43&hovh=223&hovw=226&tx=129&ty=93&sig=112740971482206625992&page=13&tbnh=120&tbnw=120&ndsp=21&ved=1t:429,r:4,s:261,i:284

Ian Fleming

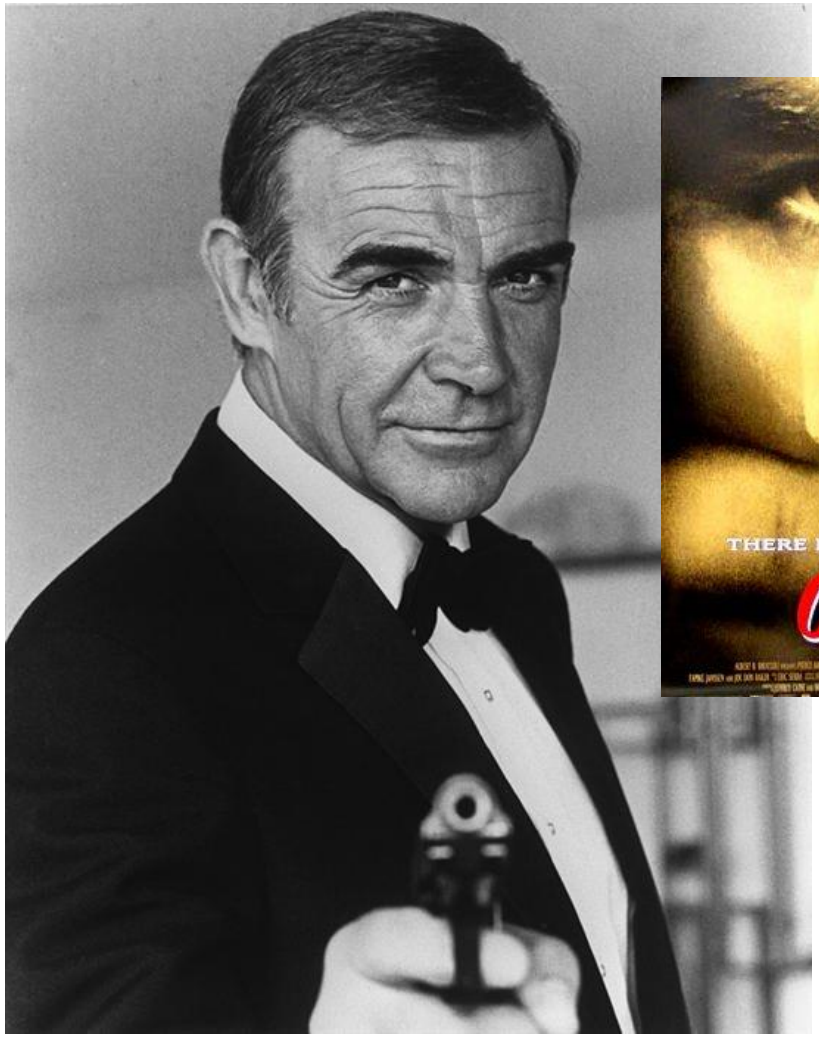
http://rolexblog.blogspot.ca/2008/11/list-of-all-watches-worn-in-james-bond.html

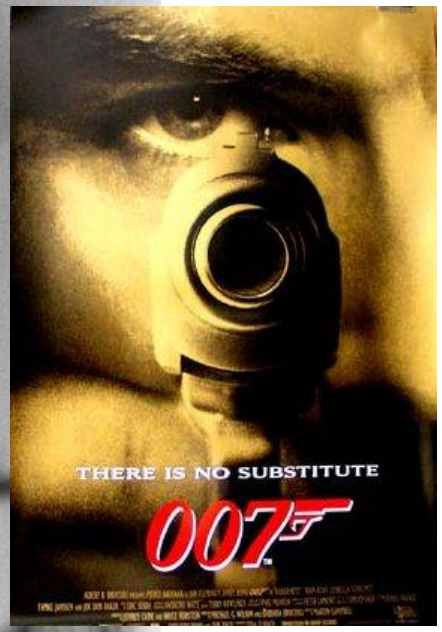
English author, journalist and Naval Intelligence Officer, best known for his James Bond series of spy novels.

http://www.wired.com/underwire/2008/10/golden-thief-st/

http://wallpaperstock.net/james-bond-skyfall-007-gun_wallpapers_33630_1280x800_1.html

http://www.klast.net/bond/





http://rolexblog.blogspot.ca/2008/11/list-of-all-watches-worn-in-james-bond.html

http://www.globalnews.ca/photos/gallery/6442728598/gallery.html

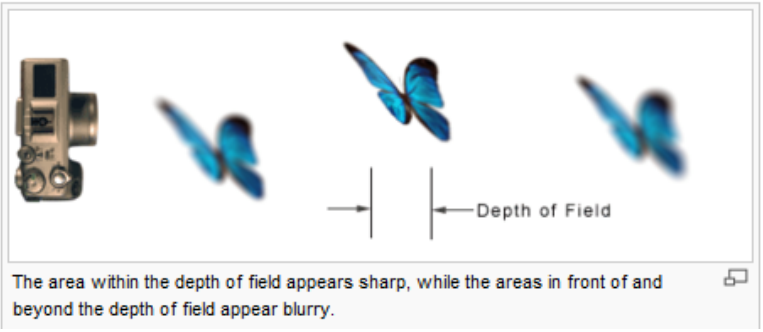
Where is focused area?

Any part out of focus?

http://www.klast.net/bond/

http://rolexblog.blogspot.ca/2008/11/list-of-all-watches-worn-in-james-bond.html

Depth of Field

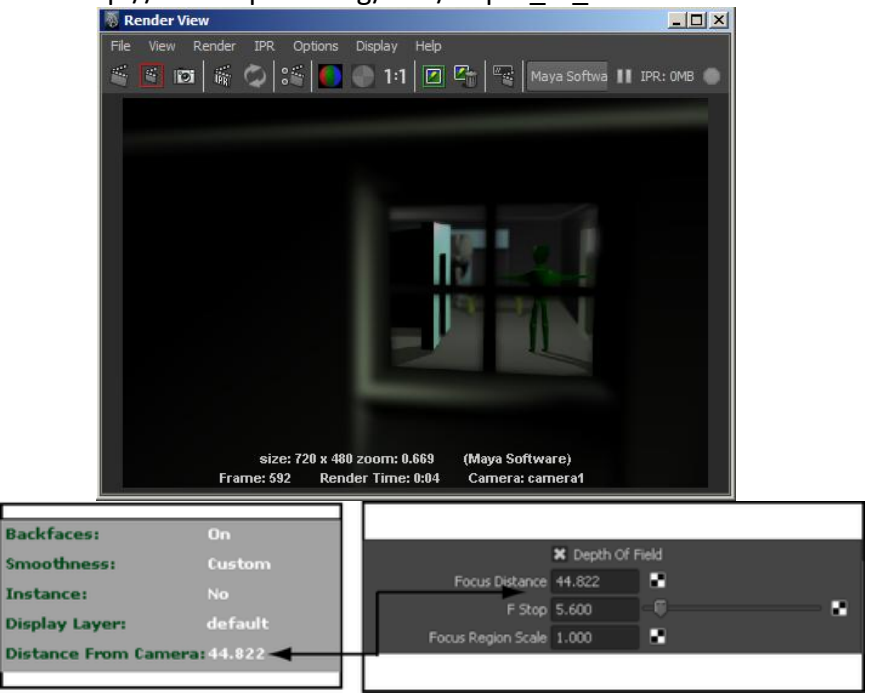


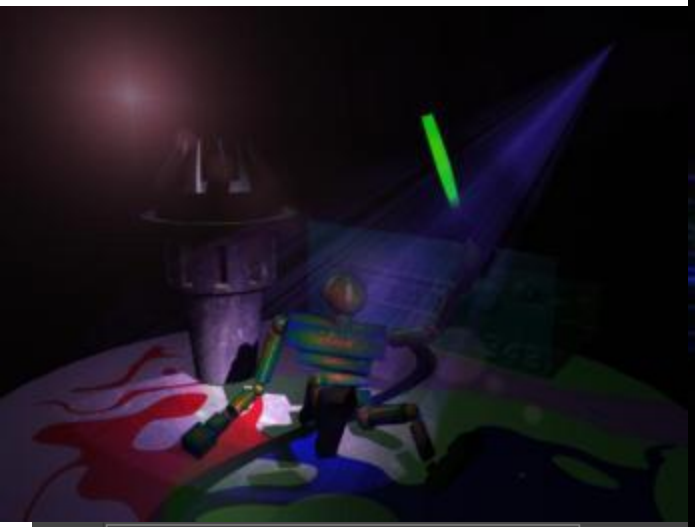
The area within the depth of field appears sharp, while the areas in front of and beyond the depth of field appear blurry.

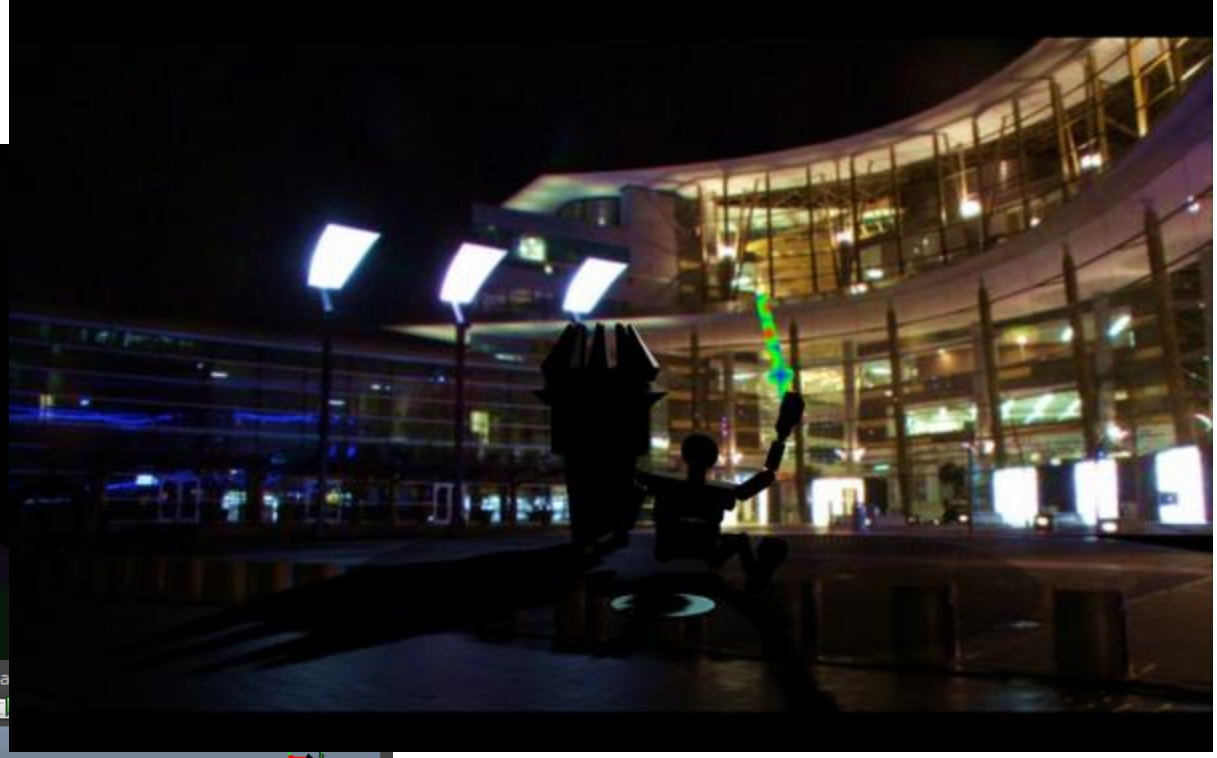
Source image:
http://en.wikipedia.org/wiki/Depth_of_field

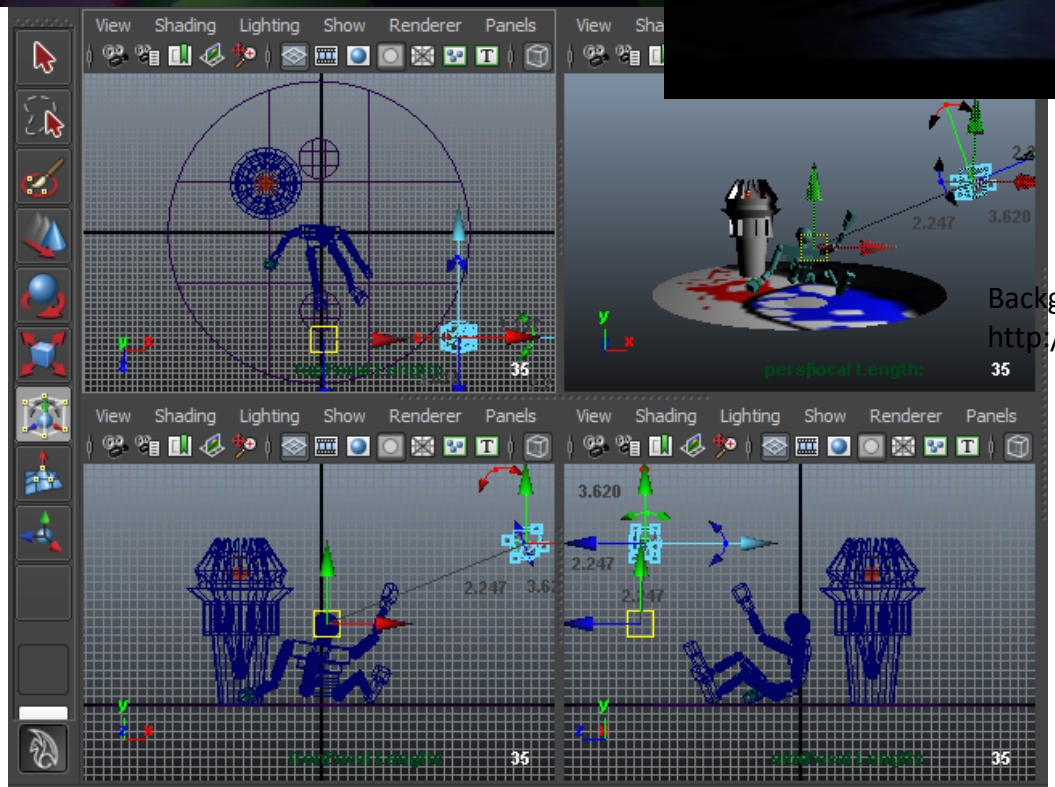
- In photography, the range of distances within which objects will be sharply focused. (Objects outside of this range appear blurred or out of focus.)

- A camera's exposure settings determine depth of field (the region of sharp focus), and whether or not subject matter is crisp or blurred by motion.



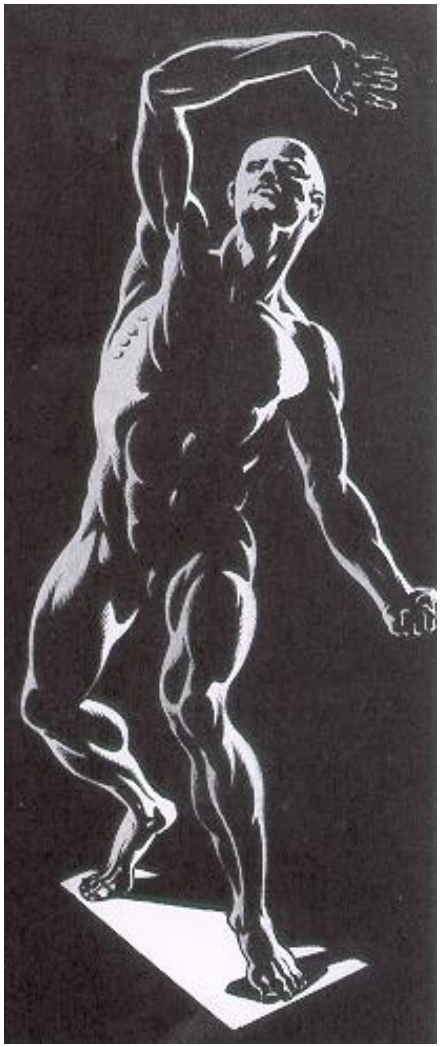






Background Image: Visit Caprica
http://www.visitcaprica.com/2011/01/episode-11-retribution.html

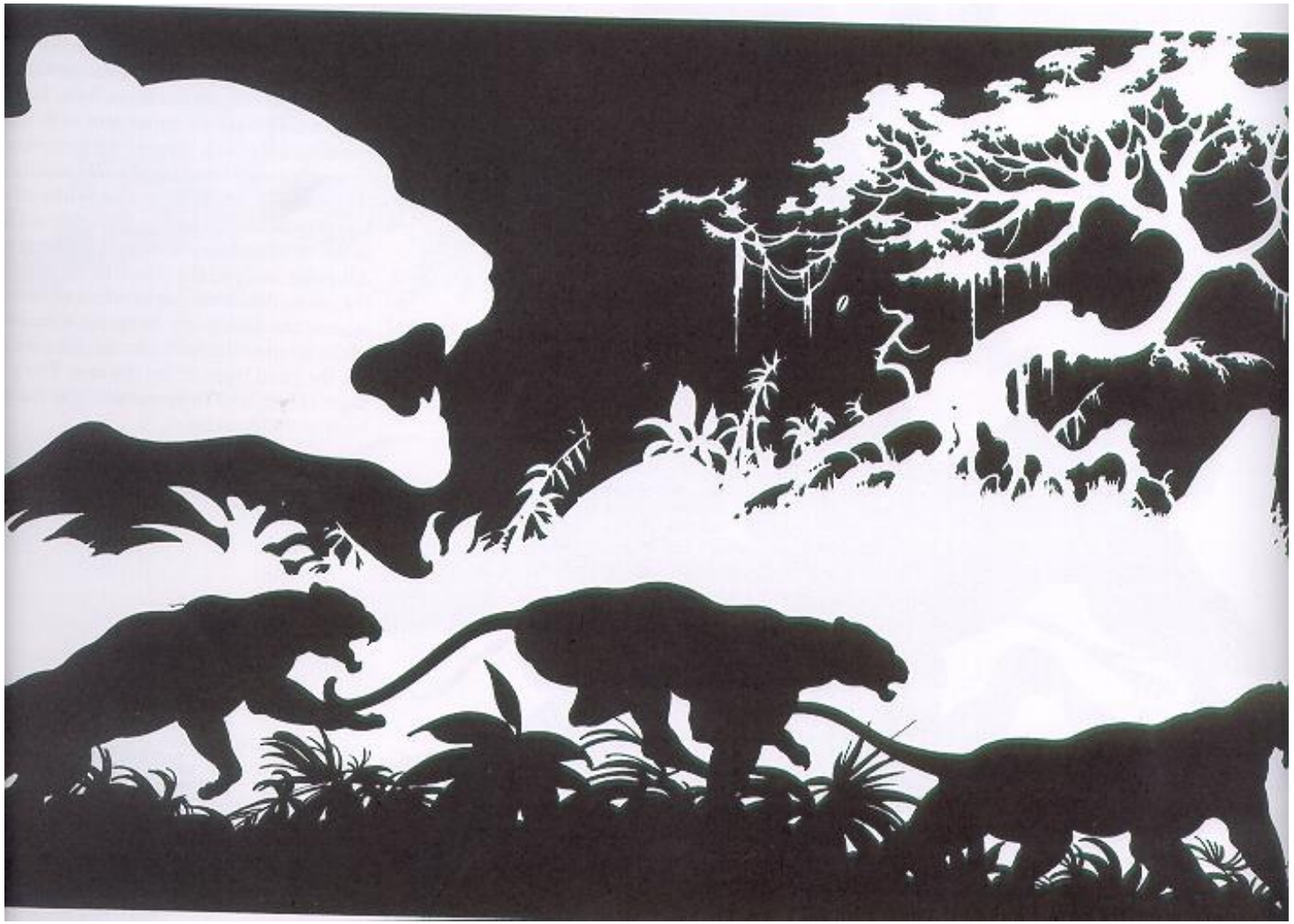
Lighting through Pencil drawing



Source:

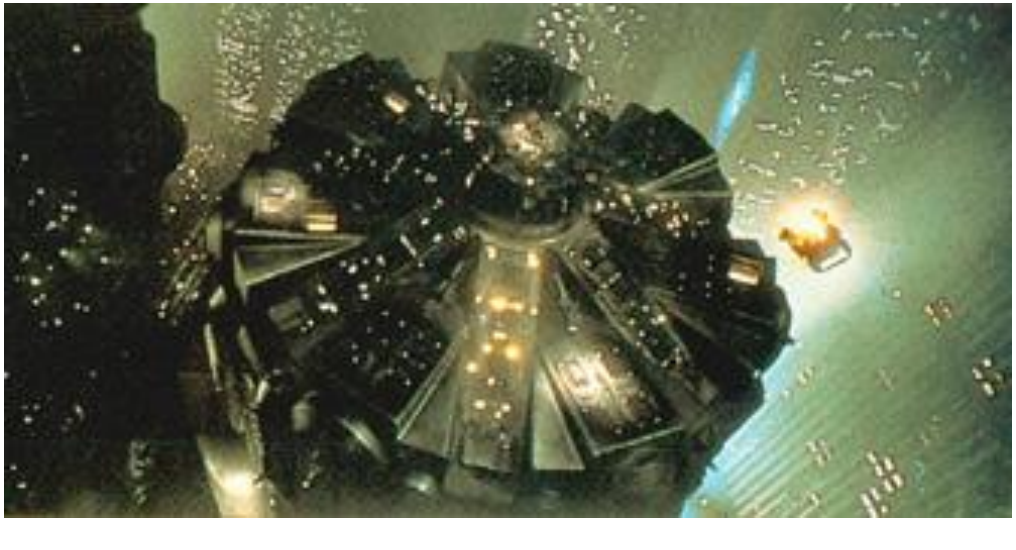
Dynamic Light and Shade (Burne Hogarth)

- Lighting



Source:

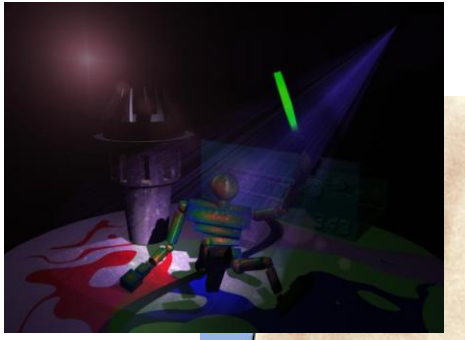
Dynamic Light and Shade (Burne Hogarth)



Source:

- Star Wars
- Blade Runner
- http://www.cityofheroes.com/

Visual Lighting Effect-
light saber

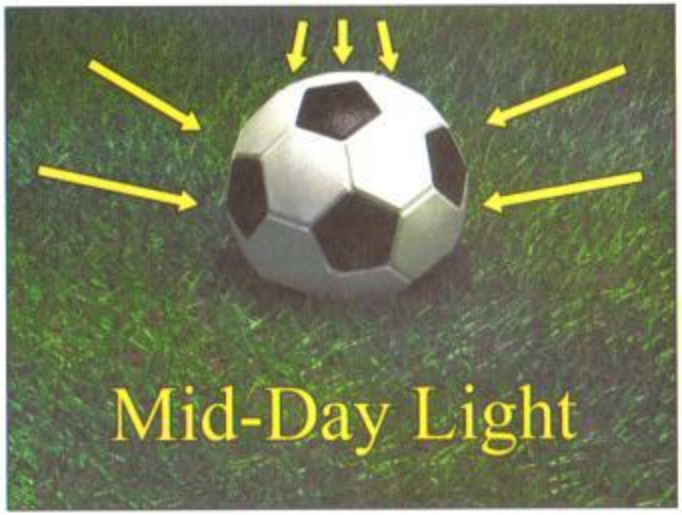


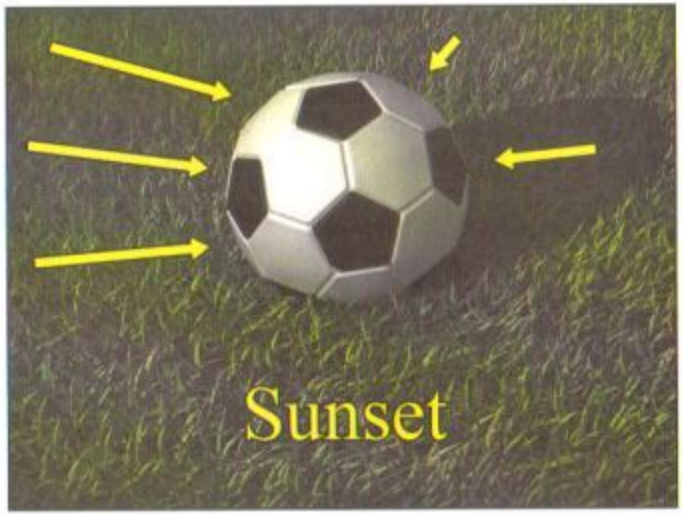
- Who was the director/writer of *Star Wars*?

- Who was the technical animator who **created the lightsaber** blade effect using rotoscope technique?

- Nelson Shin
  - Director of *Transformers* Movie (1975).
  - Animator: Pink Pander
  - Animation Producer
    - Transformers
    - The Simpsons
    - Spiderman, x-men

Sources of Light



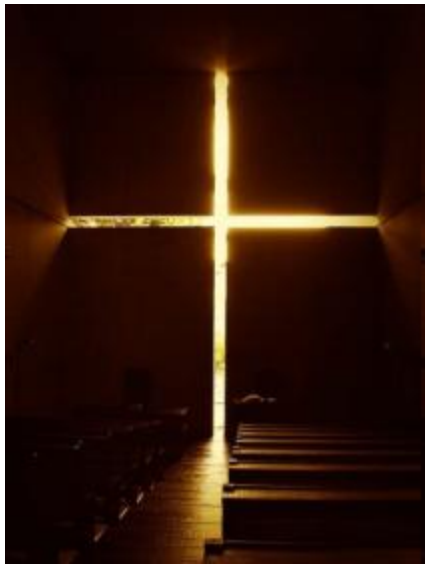


Natural light

- Time
  - Daytime
  - Night sky
  - Sunset/sunrise
- Location
  - City
  - Nature (mountain, ocean etc.)
  - Space

(Donati, 2008)

Relationship between Modeling & Lighting



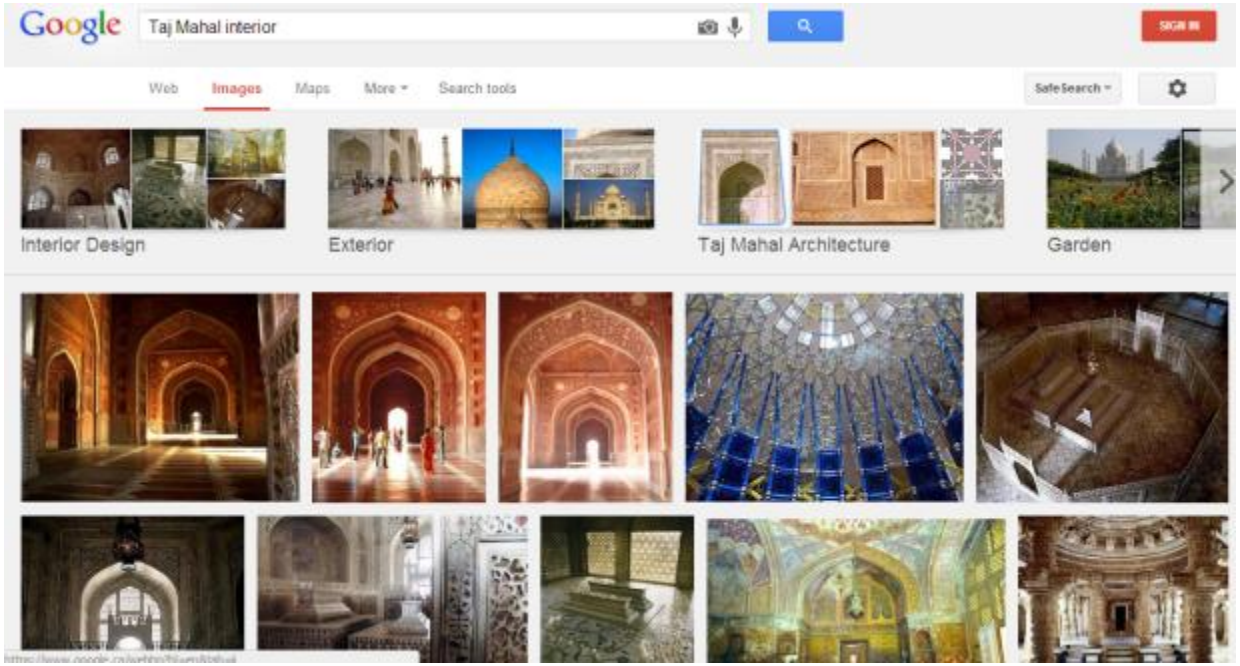
Ando Tadao

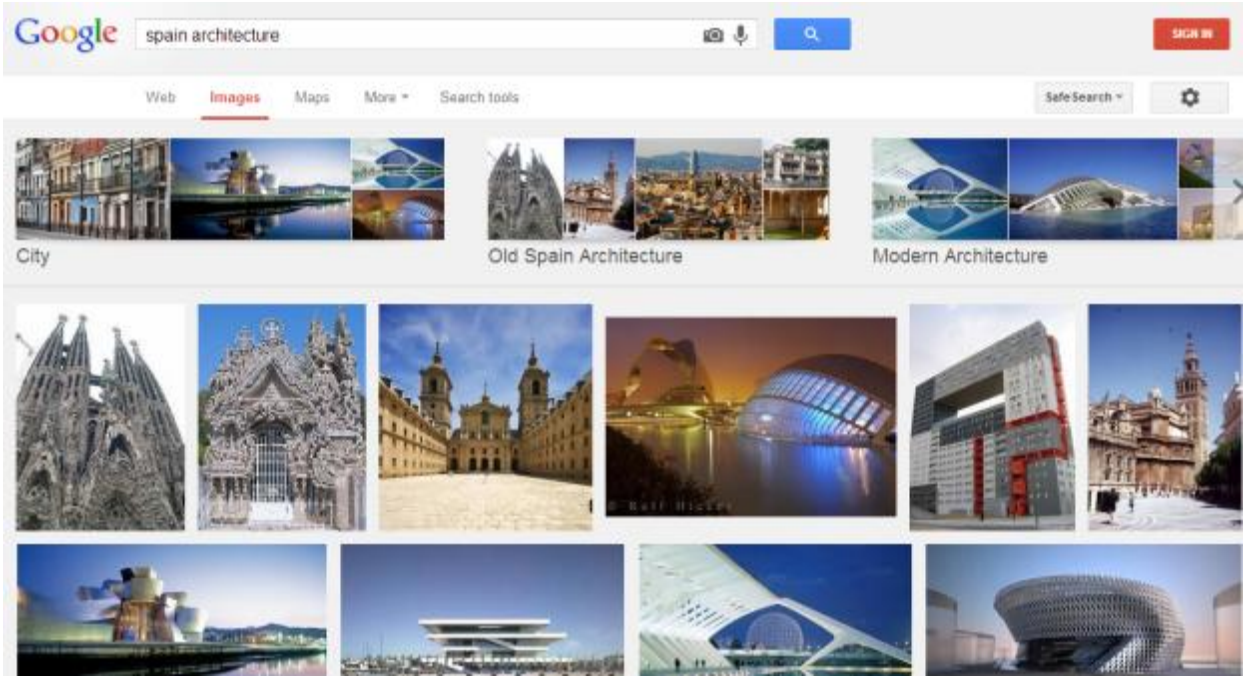
安藤忠雄

http://www.sidhin.com/w_read.html?read_no=644

http://www.minusfive.com/architecture/39/about-tadao-ando

http://www.minusfive.com/architecture/39/about-tadao-ando#mf-image-gallery



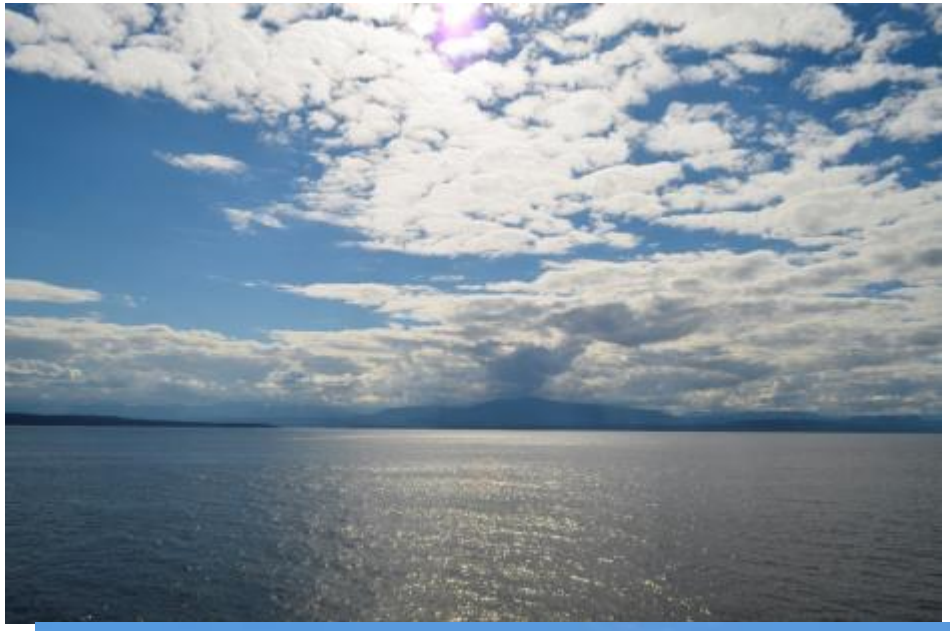


- How do we use CG to lit environment?

Why is the sky blue? *blue sky, white cloud…*

- **Color of mountain?**

Let's imagine you just finish modeling of environment.
Can you properly color/lit this environment?
How many shaders (color palette) are necessary to make it photorealistic?



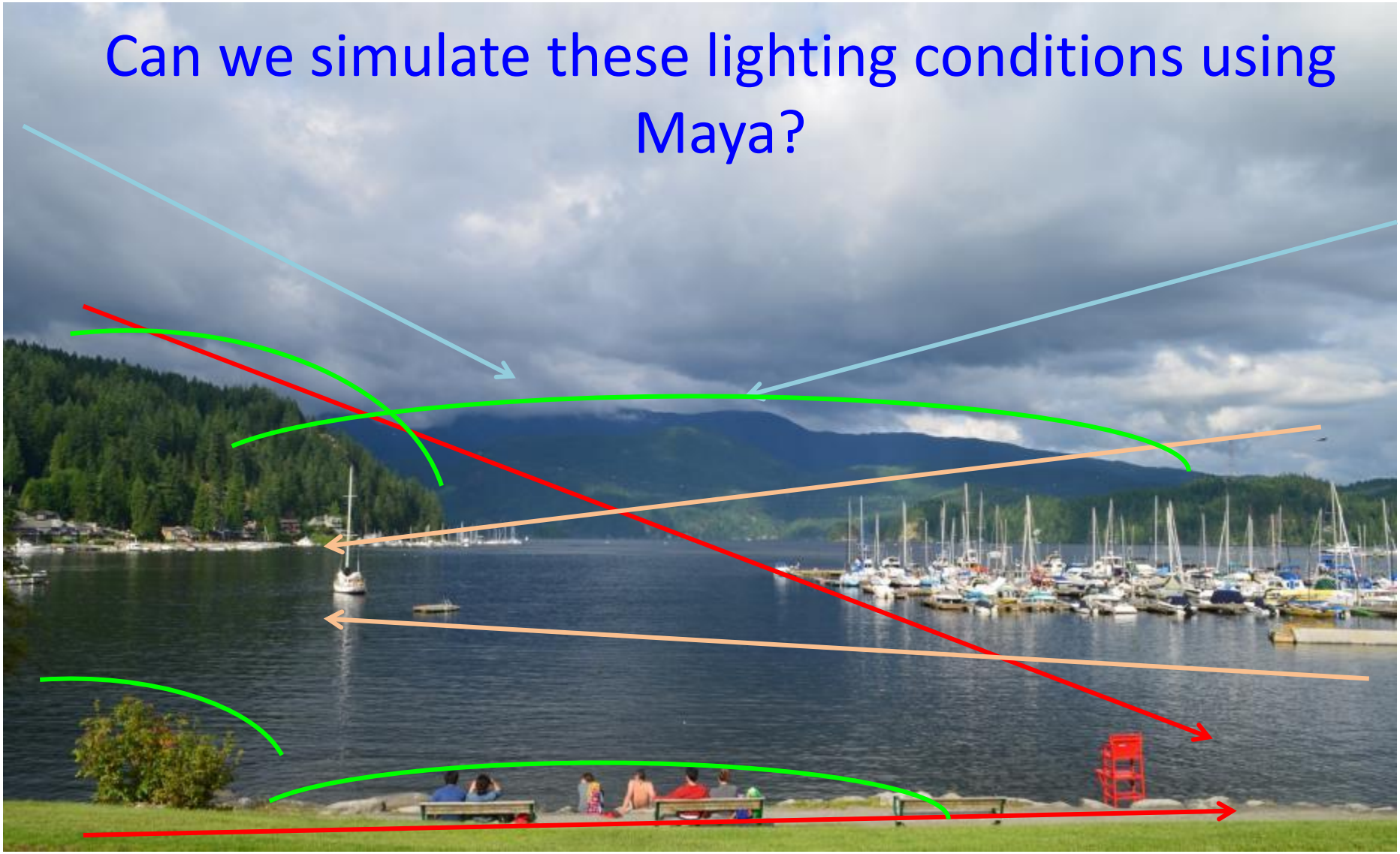
How blue is blue?
Any variation?

What cause these variations?

Can you properly color/lit this environment?

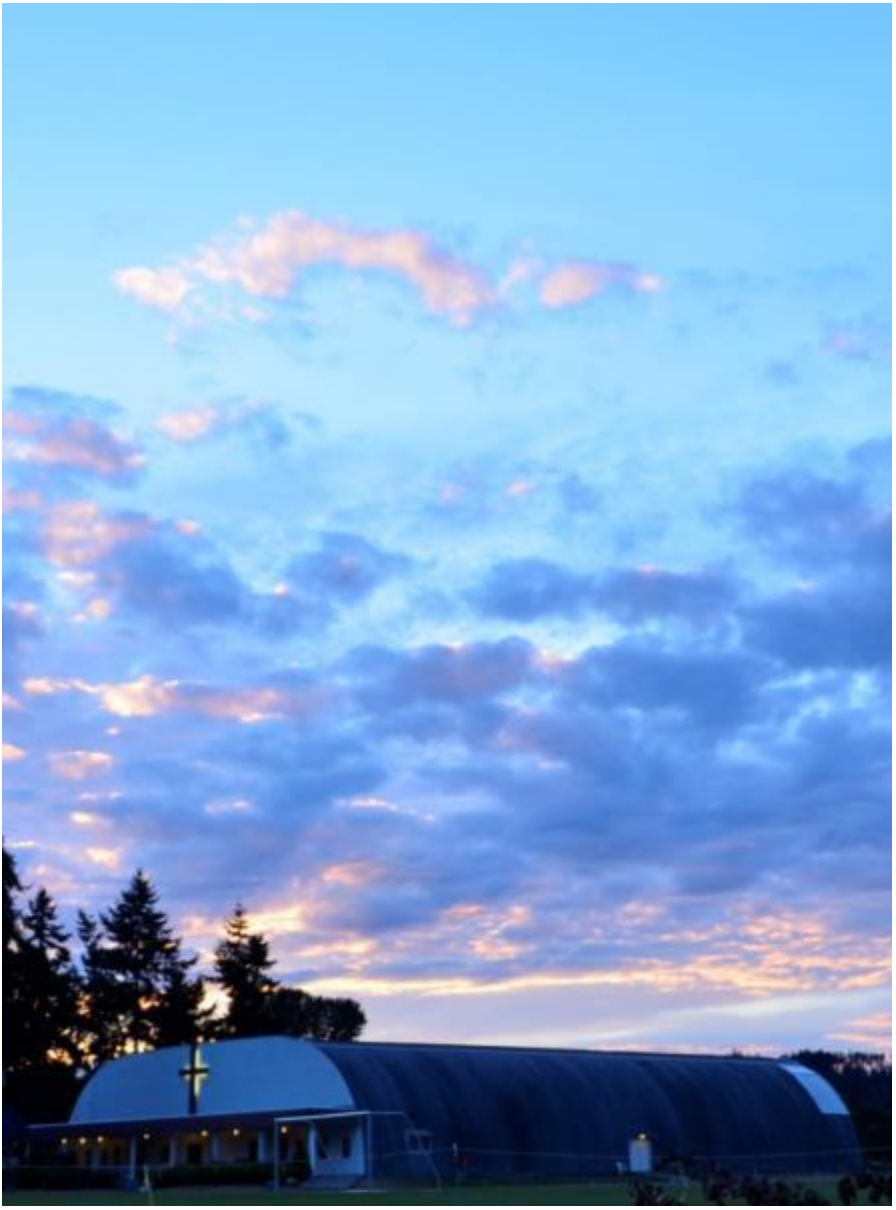
To reconstruct this photo using Maya, how many different green intensities are necessary? Still blue sky & white cloud? Where is the sun?

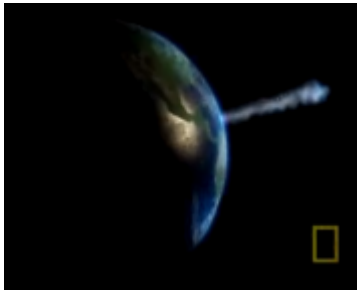
Shot direction? What causes visual focus? Region of Interest?



different green intensities

Sunset, dusk etc…why blue? Some pink? Why?
Where is sun? Alpenglow effects? http://en.wikipedia.org/wiki/Alpenglow





Light Source: Sun

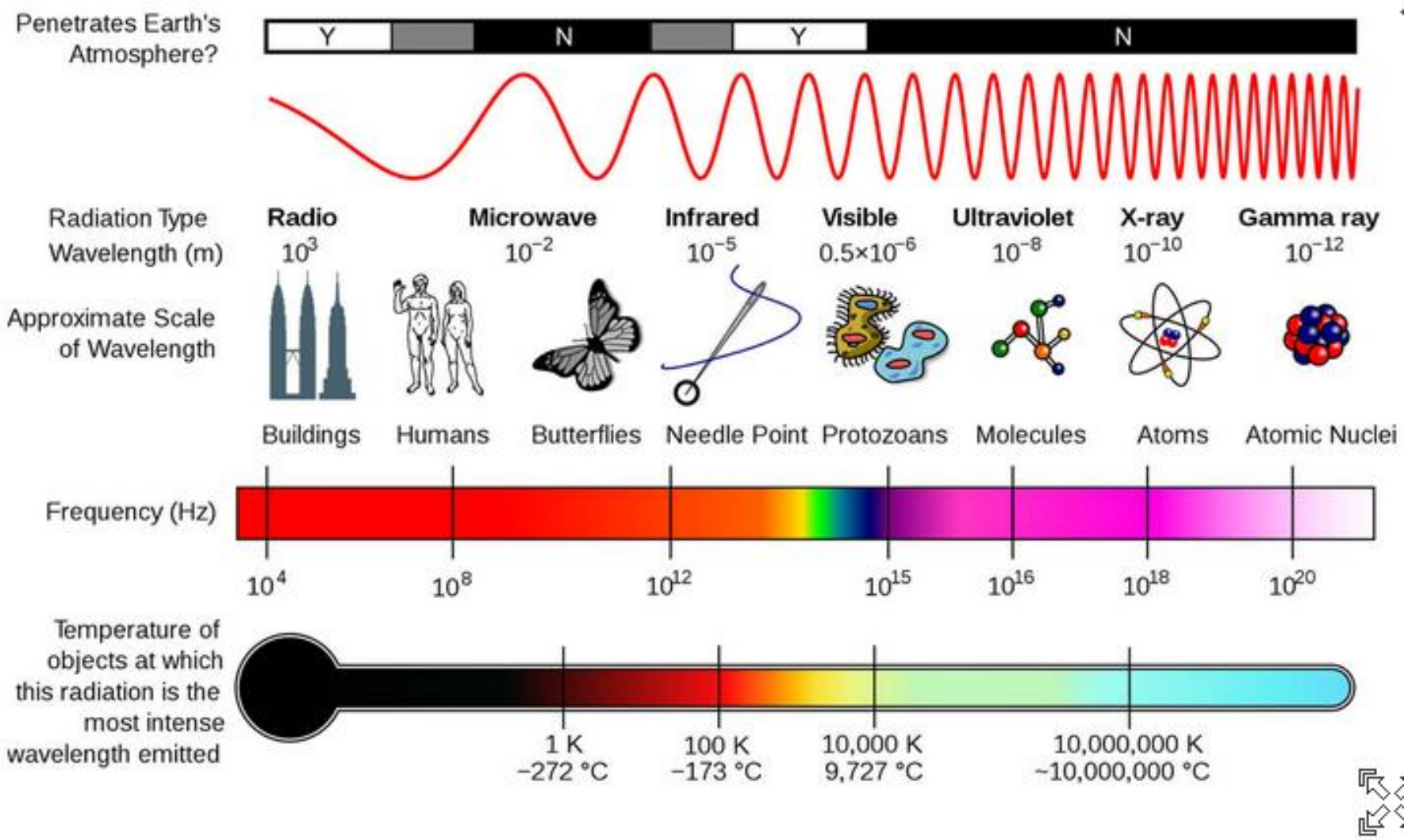
http://youtu.be/1YAOT92wuD8

- Why the Earth looks blue?

- Why is the sky blue?
  - John Tyndall in 1859. He discovered that when light passes through a clear fluid holding small particles in suspension, the shorter blue wavelengths are scattered more strongly than the red.

    http://math.ucr.edu/home/baez/physics/General/BlueSky/blue_sky.html
  - The oceans (and lakes and large rivers) appear blue because of the way sunlight is selectively scattered as it goes through our atmosphere.

    http://mb-soft.com/public/earthblu.html

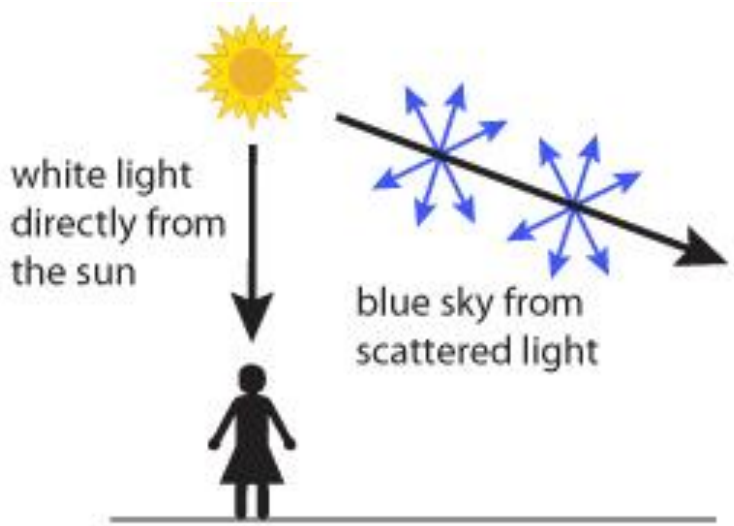


Types of light waves, their relative wavelength and the smallest object they can image.

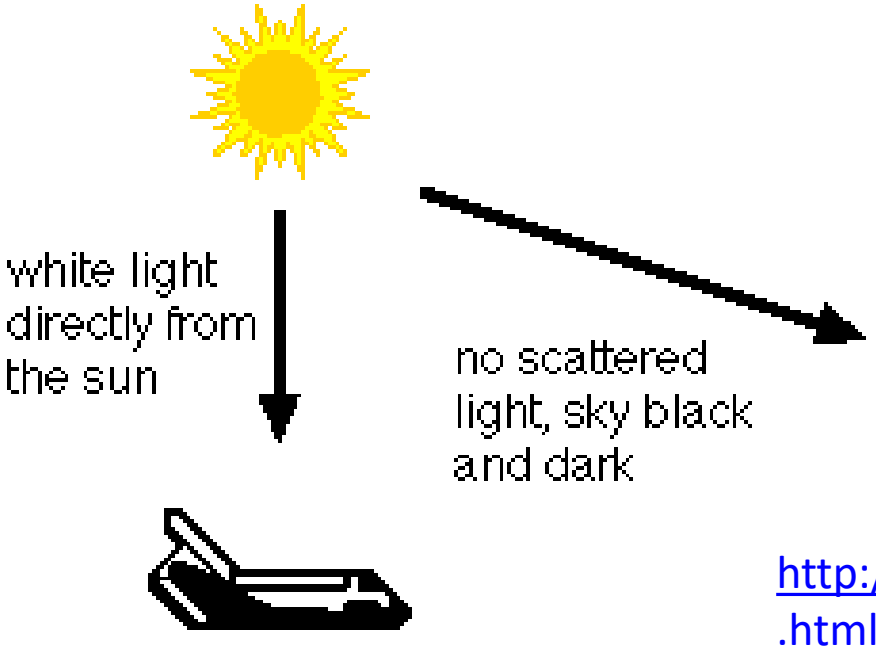
http://www.theozonehole.com/atmosphere.htm

http://askabiologist.asu.edu/quick-class-x-ray-crystallography

LAYERS OF THE ATMOSPHERE



When you look up, some of this blue light reaches your eyes from all over the sky. Since you see blue light from everywhere overhead, the sky looks blue.

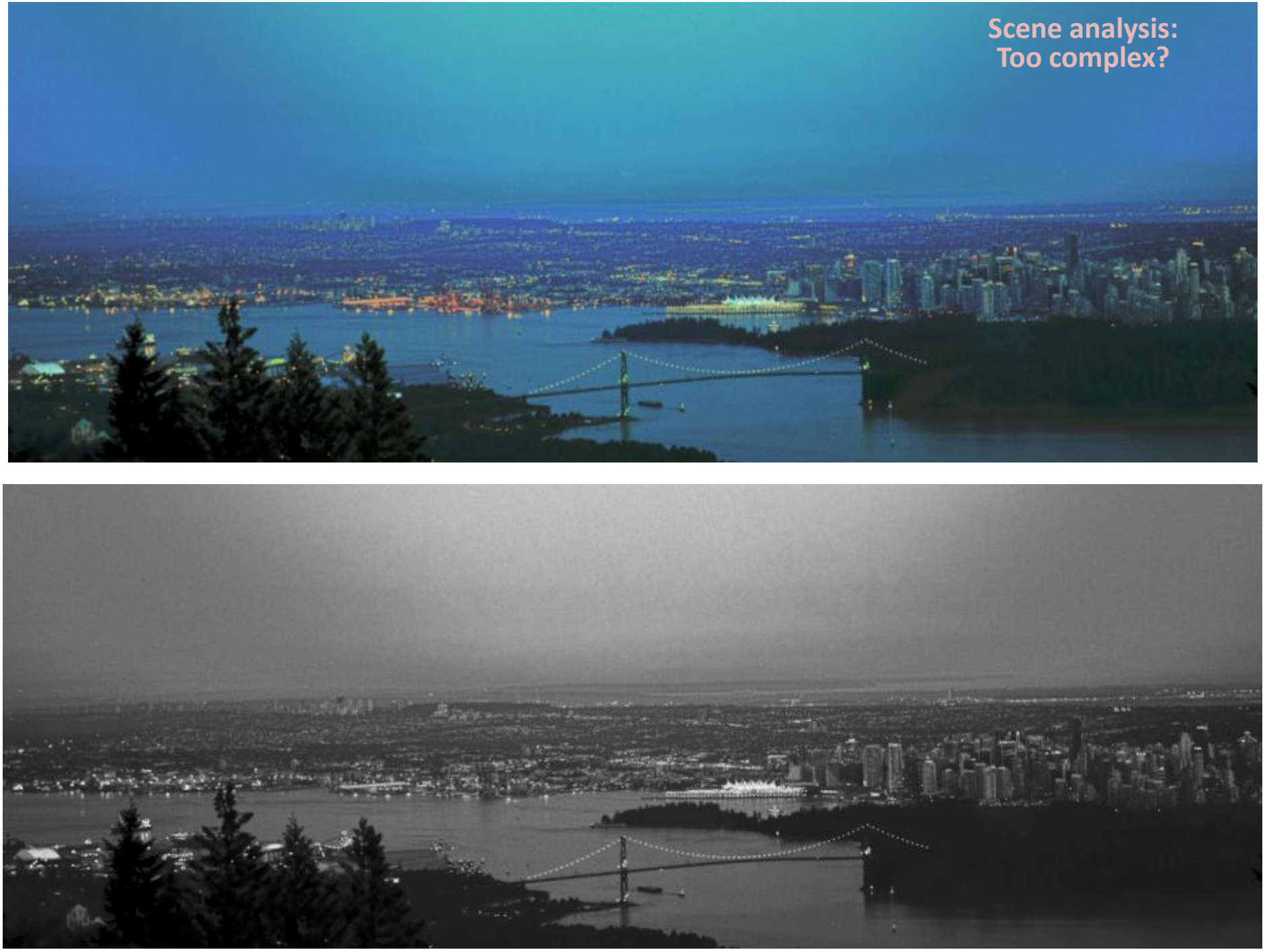


In space, there is no air. Because there is nothing for the light to bounce off, it just goes straight. None of the light gets scattered, and the "sky" looks dark and black.

http://www.sciencemadesimple.com/sky_blue.html

Mirroring? Any variation? What cause it?

Color bleeding.


Scene analysis:
Too complex?

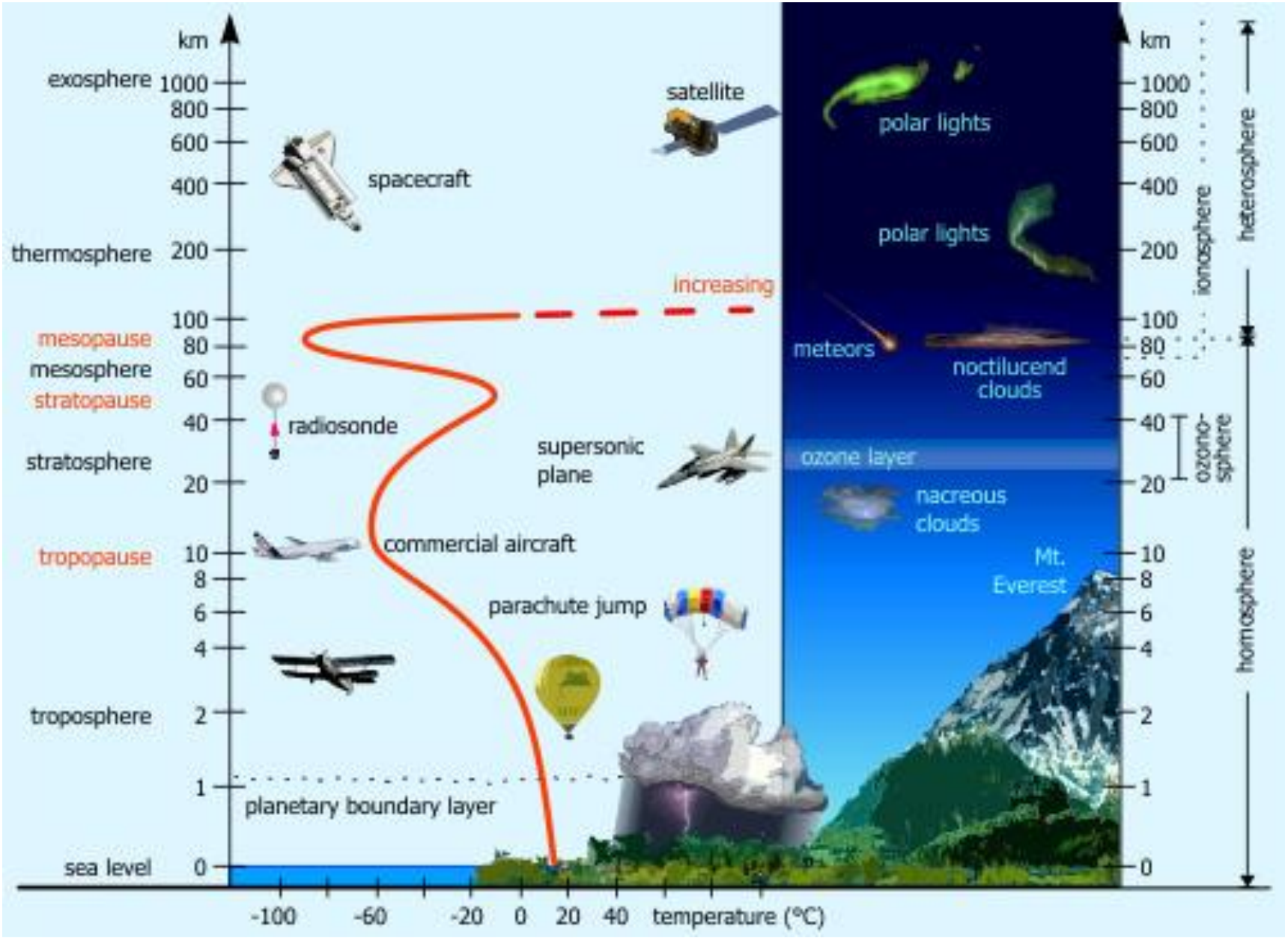
Scene complexity:
static, dynamic or both.

Position of Light source

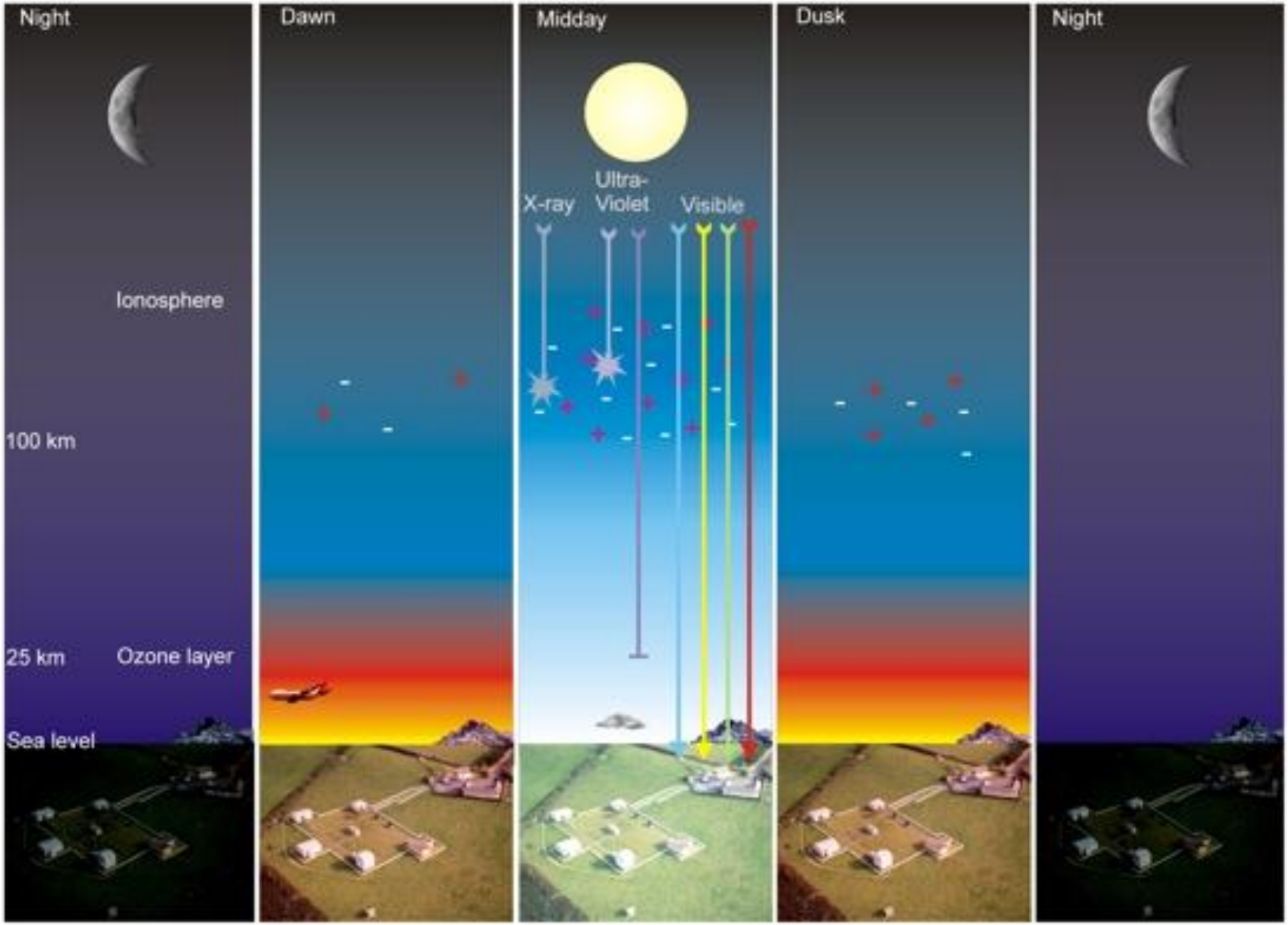
Still blue sky? What color is night sky?

How many light sources? Position of each light?

What is the color of Moon? Any variation?



http://www.theozonehole.com/atmosphere.htm

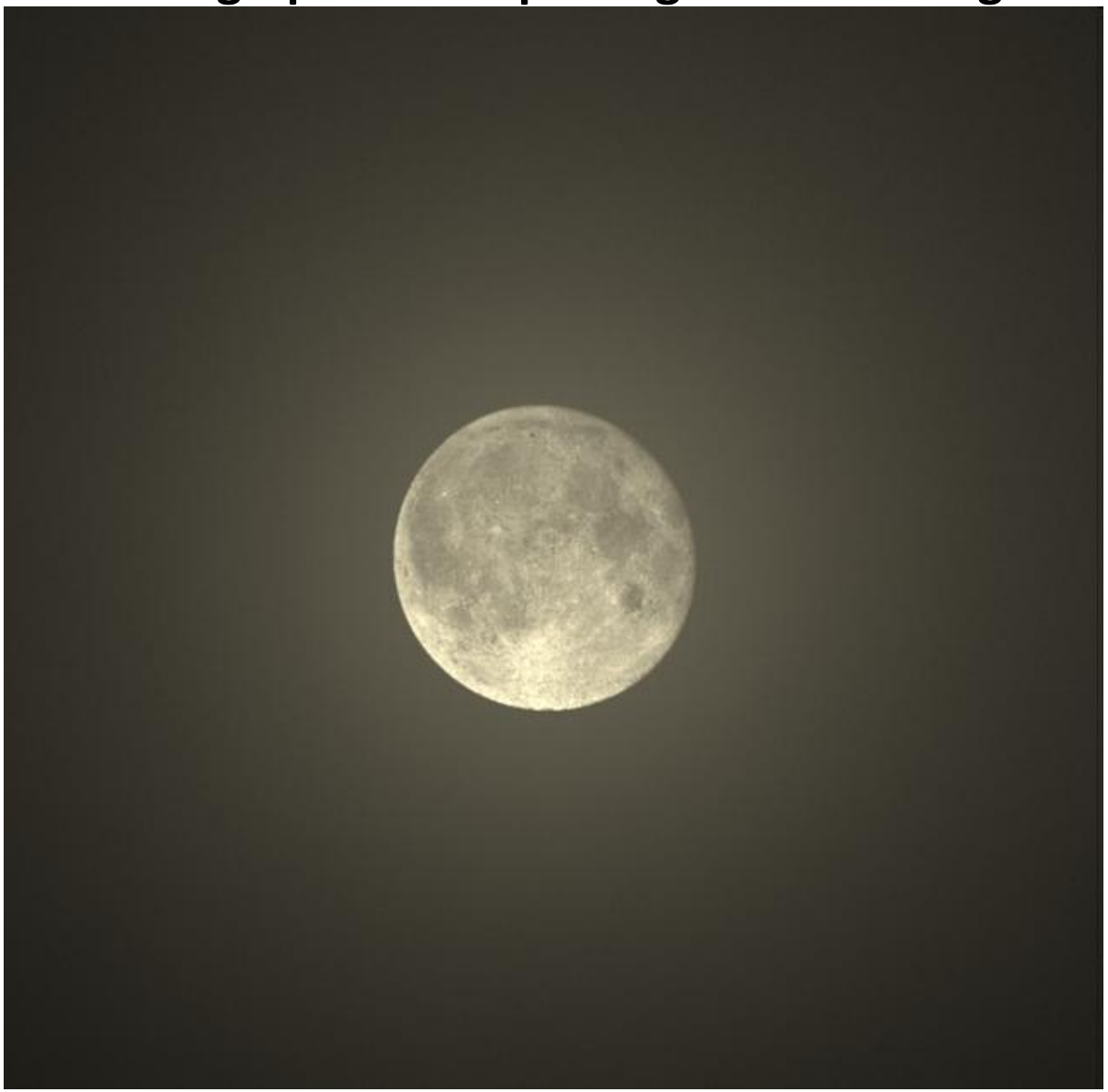


http://www.siriperera.com/wp-content/uploads/2012/08/IonosphericIonisation.jpg

Using CG, is it possible to simulate a large outdoor environment?

Global Illumination in Nature

Photograph or Computer-generated Image?



Global Illumination in Night Sky

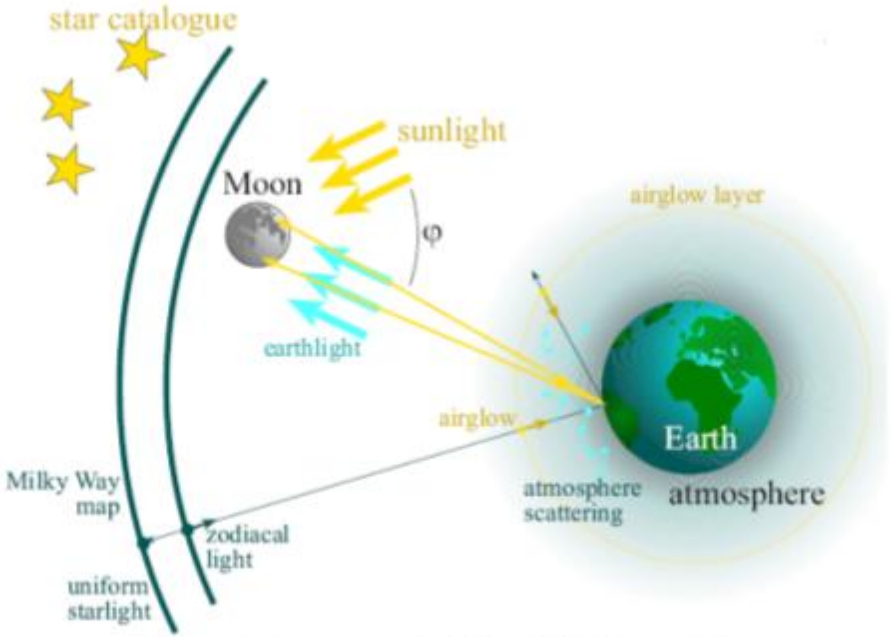


Figure 3: Components of the night sky model.

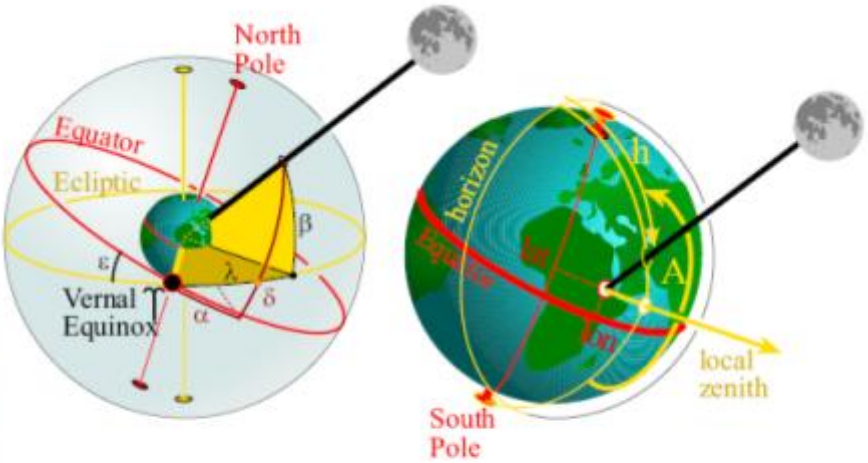
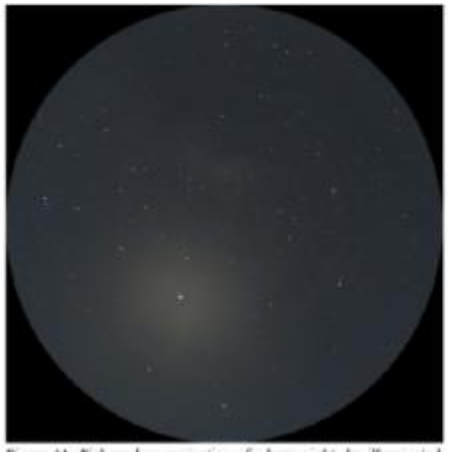
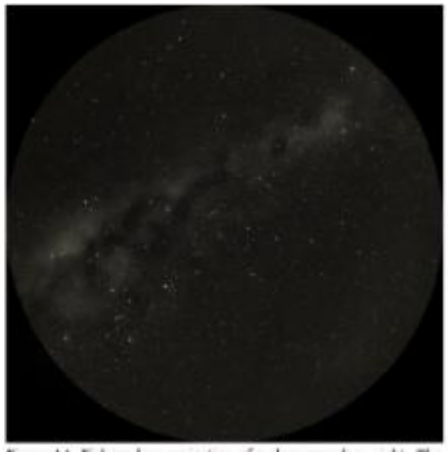


Figure 4: Coordinate systems. Left: equatorial (α, δ) and ecliptic (λ, β). Right: Local coordinates (A, h) for an observer at longitude lon and latitude lat.

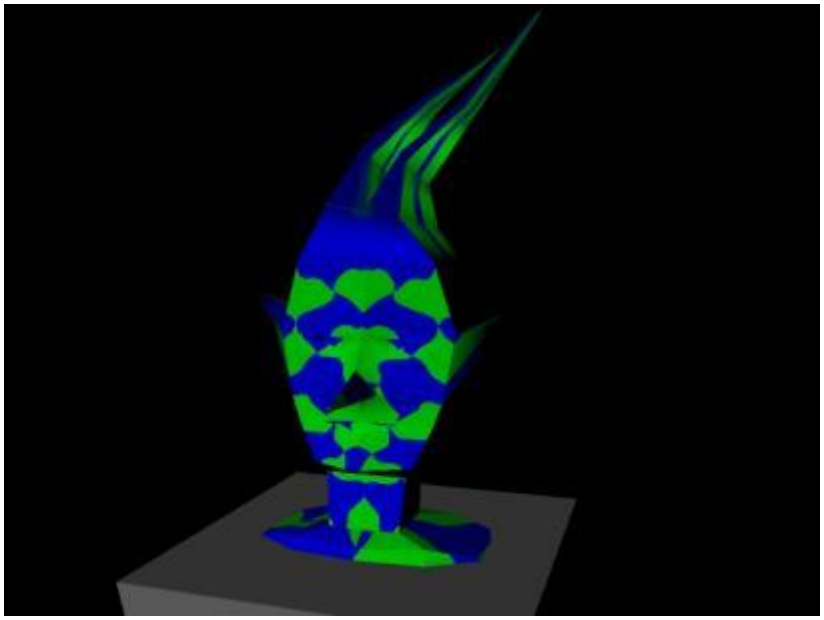
- *A Physically-based Night Sky Model* by Jensen, H. W., Durand, F., Stark, M. M., Premoze, S., Dorsey, J., & Shirly, P. ACM SIGGRAPH 2004

Global Illumination in Night Sky





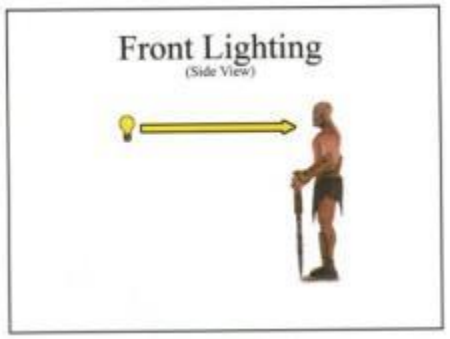
Working on the scene composition of Team project ?

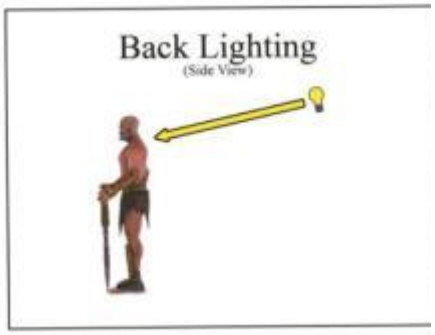


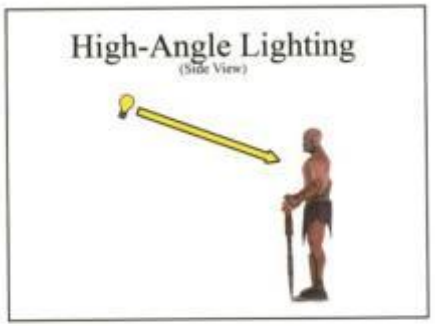
Background color as light source.
(simulating Blue sky? Dusk?)

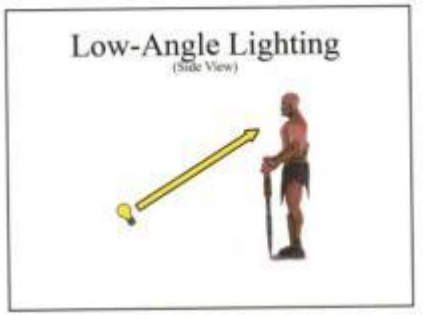
Positioning Lights:
Stage Lighting Design

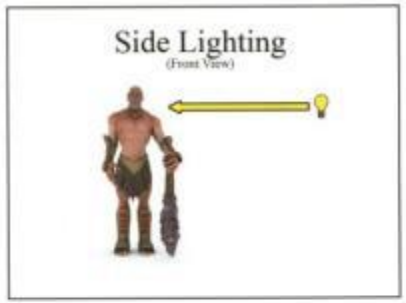
http://en.wikipedia.org/wiki/Stage_lighting











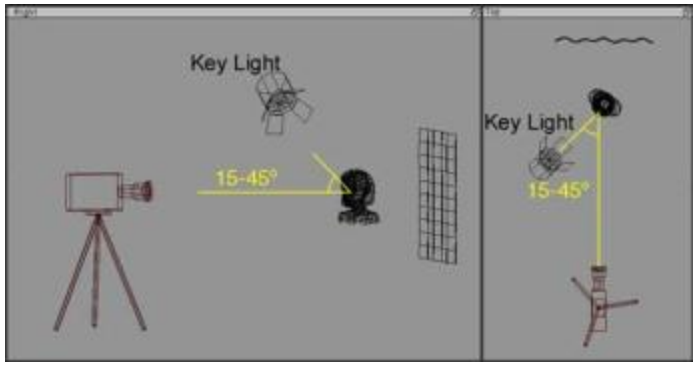
Source:
Donati, J. (2008). Exploring digital cinematography

Goals of Good Lighting

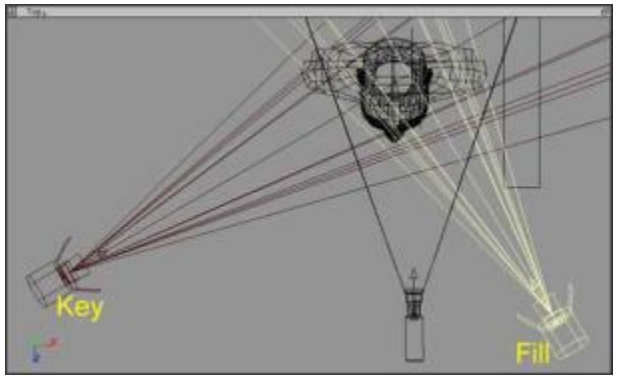
(Brown, 2012)

- A full range of tones and gradations of tone
- Color control and balance
- Shape and dimension in the individual subjects
- Separation: subjects stand out against the background
- Depth and dimension in the frame
- Texture
- Mood and tone: emotional content
- Exposure
- Depth
- Quality
- Altitude

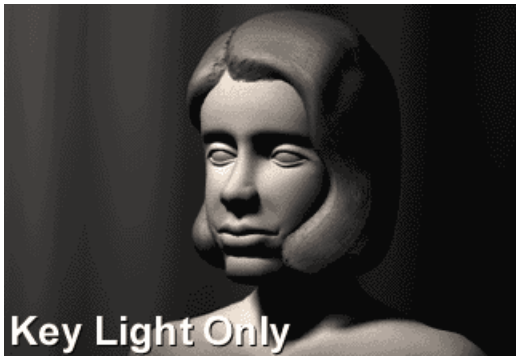
Three Point Lighting

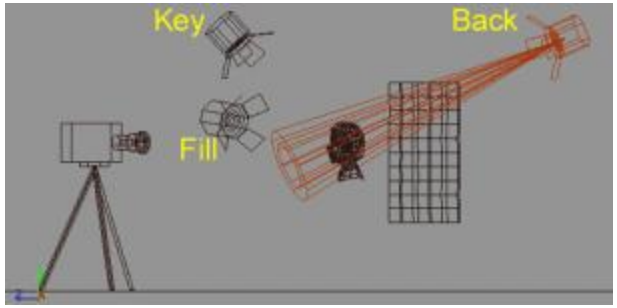


Key Light



Fill Light





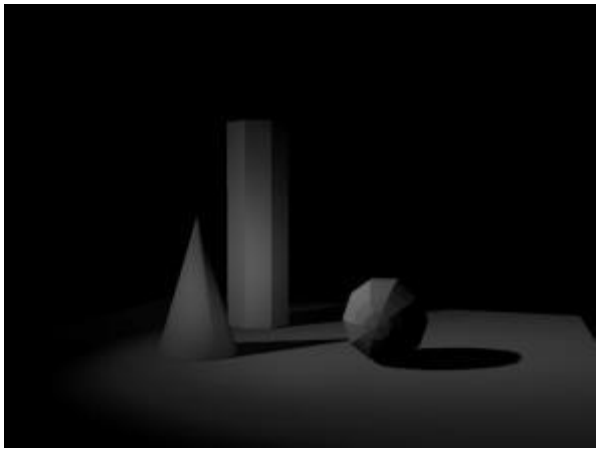
Back Light

Reading material: Jeremy Birn
http://www.3drender.com/light/3point.html

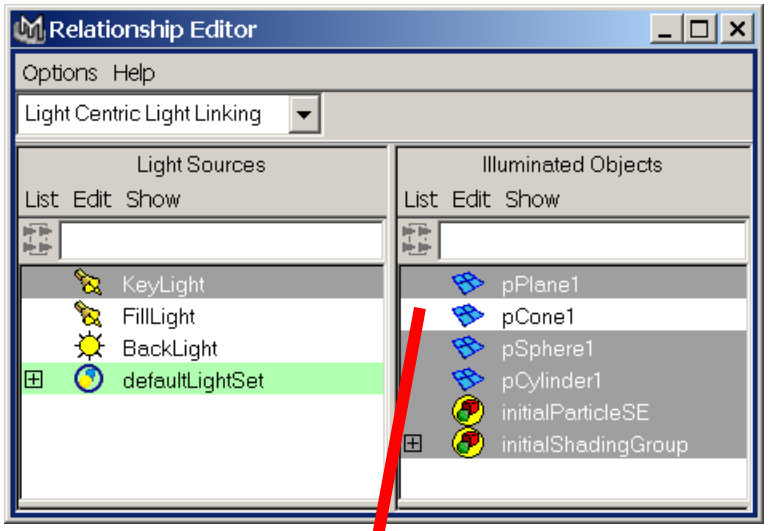
Three point Lighting & Light Linking

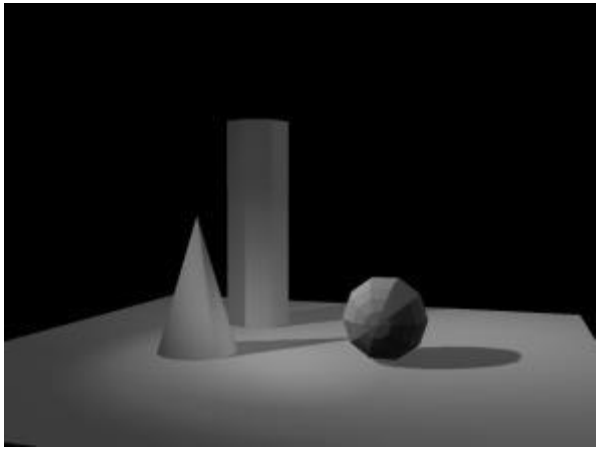
Key Light

Lighting/Shading > Light Linking > Light-Centric

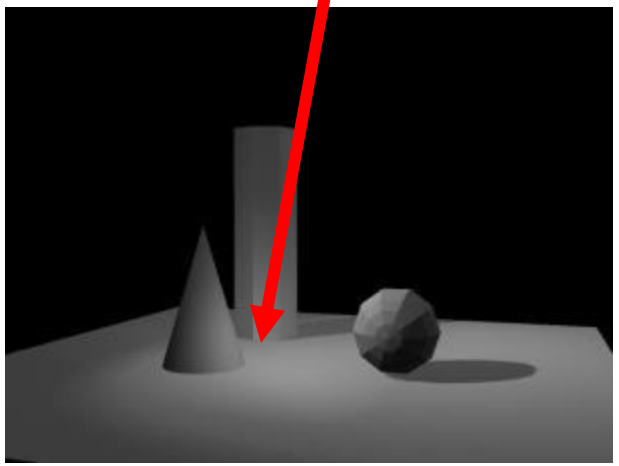


Fill Light





Back Light



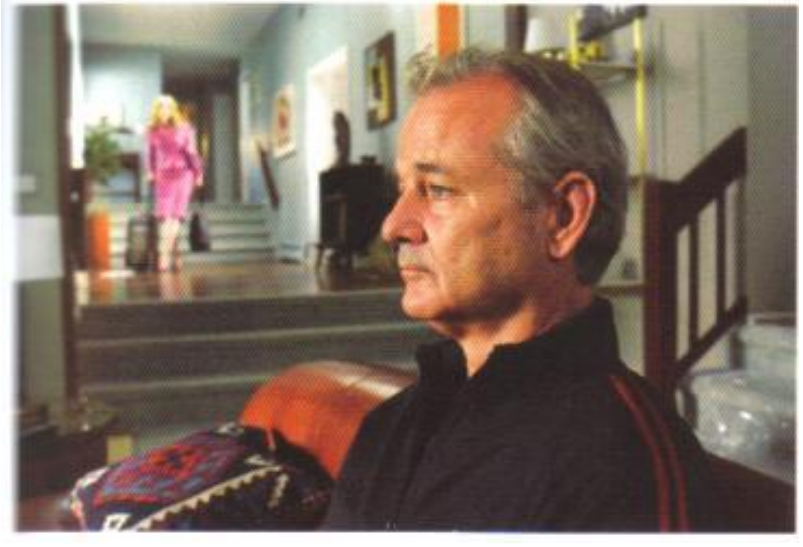
No Shadow

Key to Fill Light Ratio

- **High-key lighting**
  - bright shadowless
  - 2:1 or lower
  - Fill light nearly as intense as the key light
  - Even illumination
  - Facial details washed out

- **Natural-key lighting**
  - 4:1 to 8:1
  - No longer able to illuminate every shadow

- **Low-key lighting**
  - 16:1 to 32:1, high contrast
  - Impossible to eliminate shadows
  - Producing number of distinct shadows



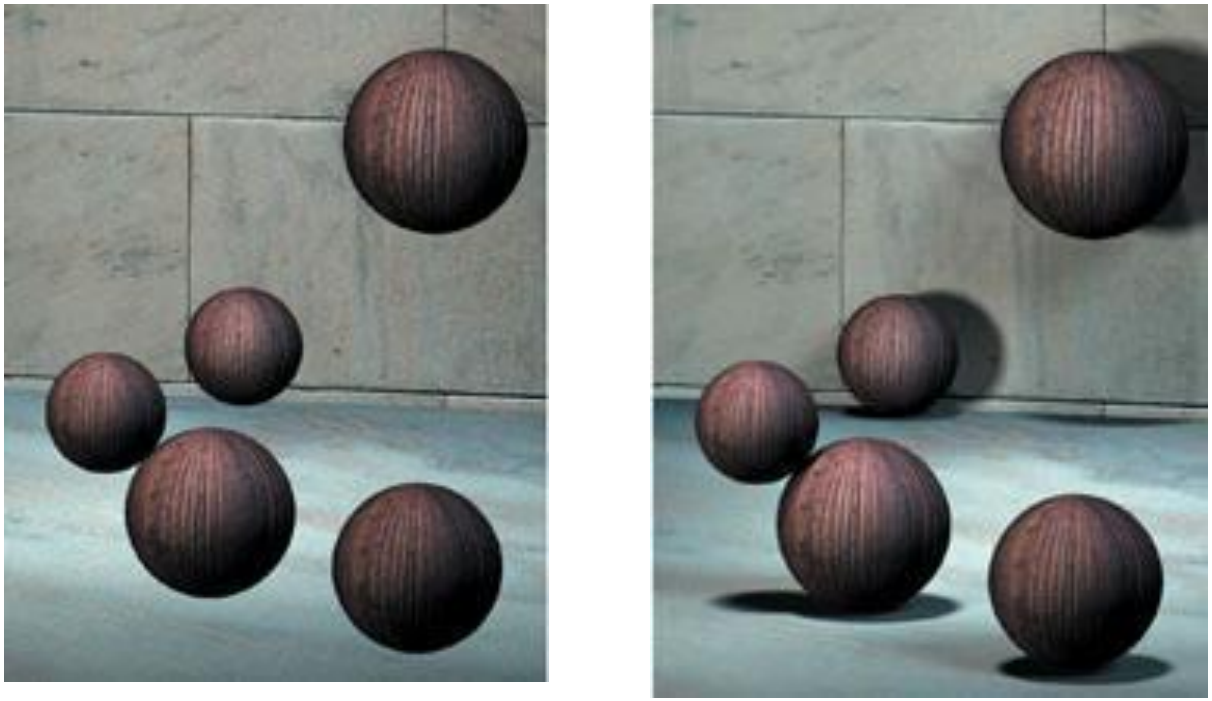
(Pramaggiore & Wallis, 2008)

- The shadow reveals a character's profile, which otherwise would not be seen in the rendering.

Source: Jeremy Birn

Composition Element

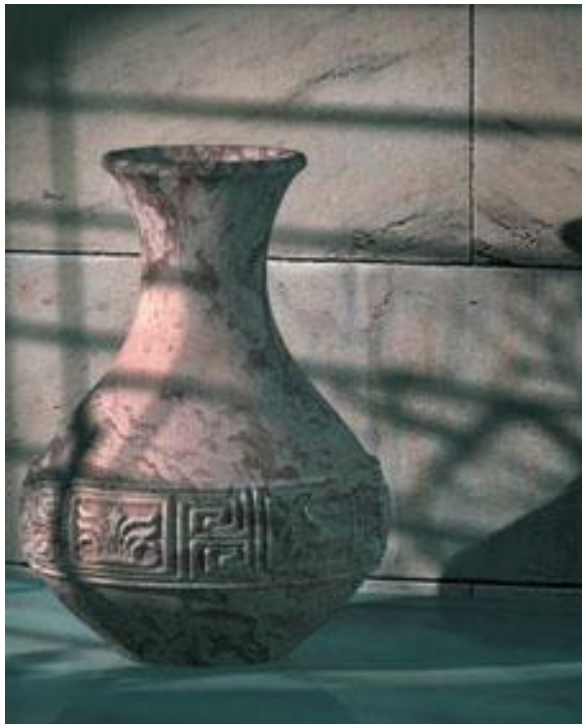
Shadow: Defining Spatial Relationships



- On the left side, you cannot tell how close the large upper ball is to the back wall. The most basic use of shadows is to show spatial relationships between objects, as in the right side of this figure.

Source: Digital Lighting & Rendering (Jeremy Birn)





- A slash breaks up the space and adds to the composition, making the image on the right a more attractive rendering.

- The shadow indicates what might exist in off-screen space.

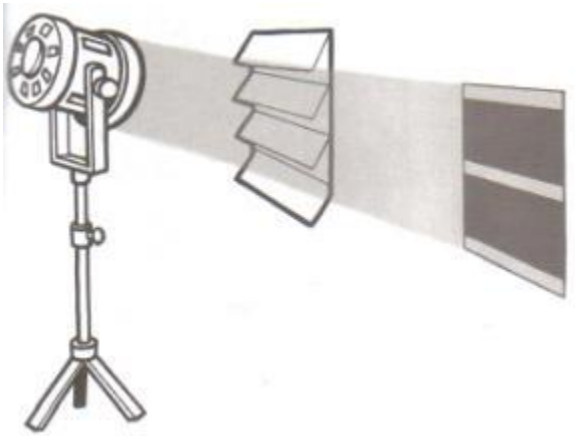
Source: Jeremy Birn

- What size do you want your shadow to be? Move the light farther away for a smaller shadow (left) or up close for a larger shadow

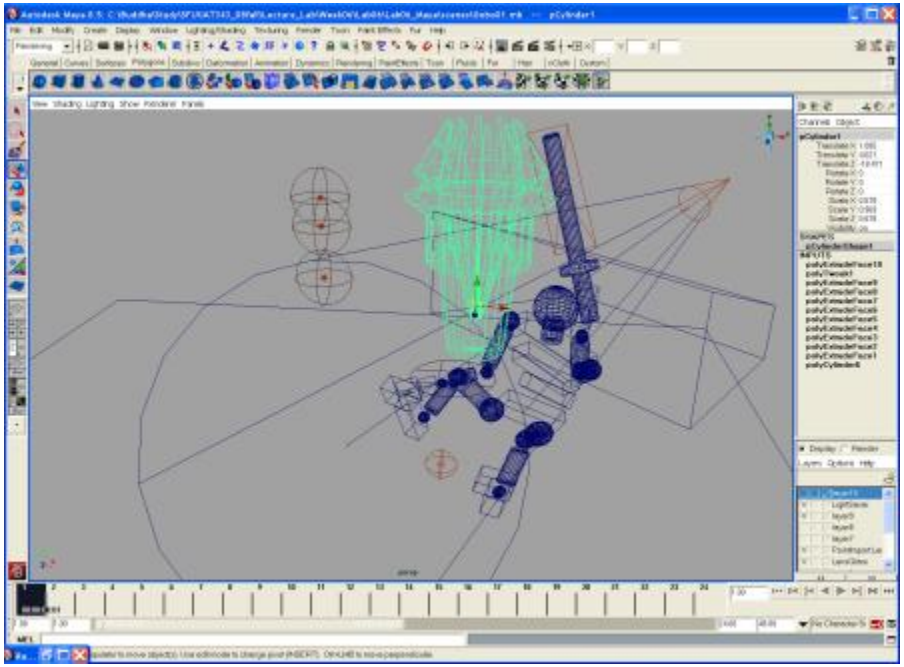
Source: Jeremy Birn


Shadow necessary? Realism?
Interrupt Aesthetics?

Gobo

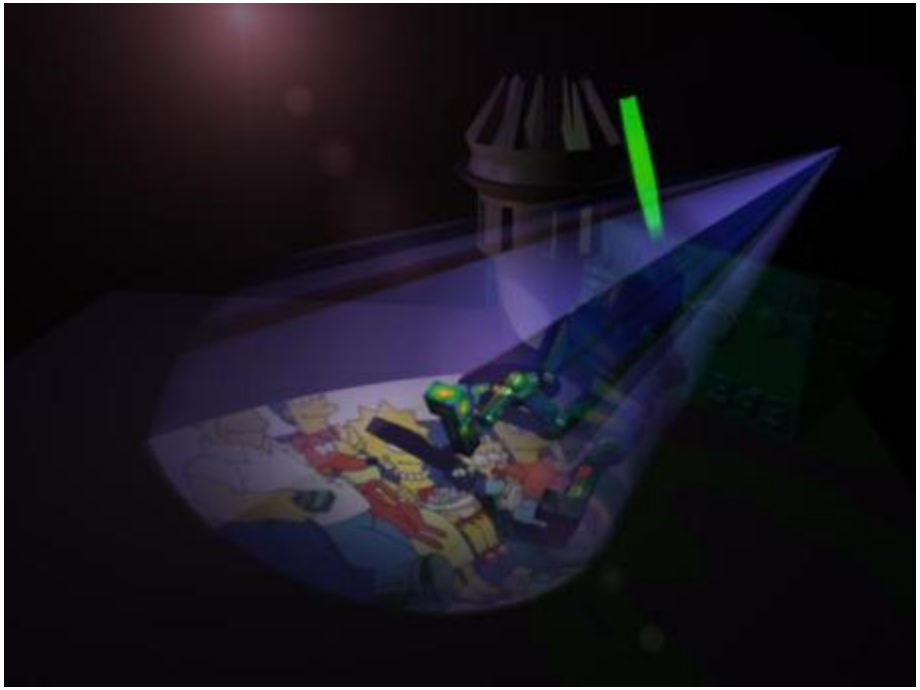


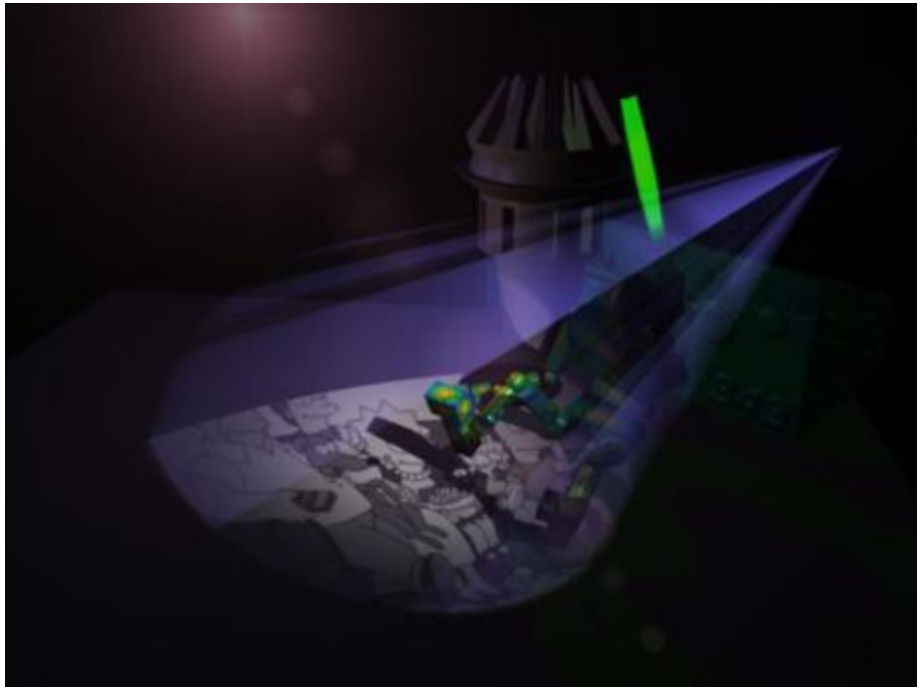
Park, J. E., (2004).



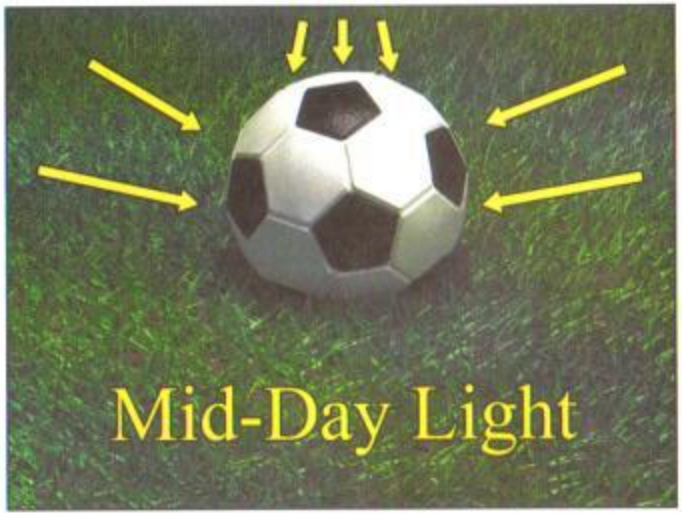
- Shadow pattern between the light and the target
- Provide the presence of objects not really in the scence
- More efficient than a real object

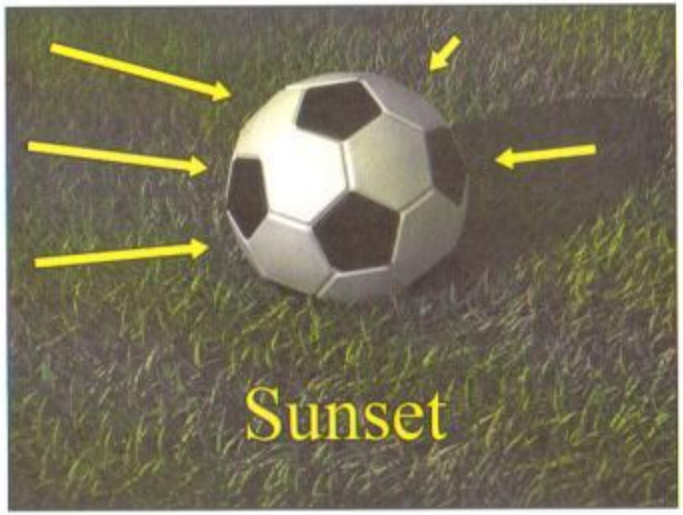
Gobo





Sources of Light



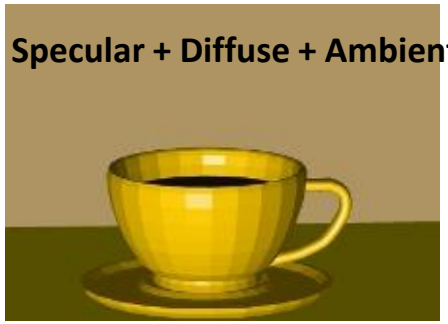


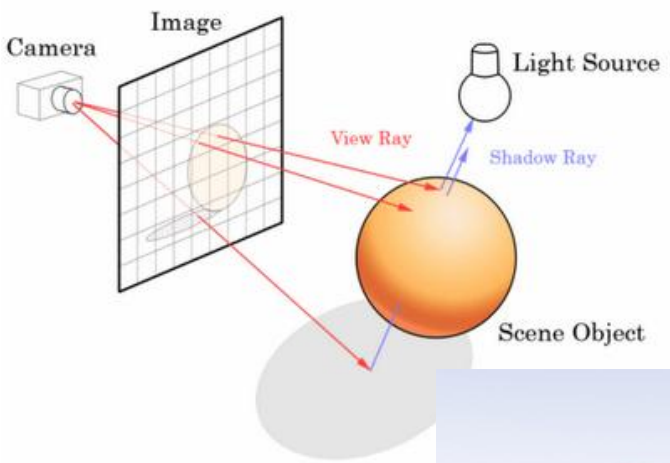
Natural light

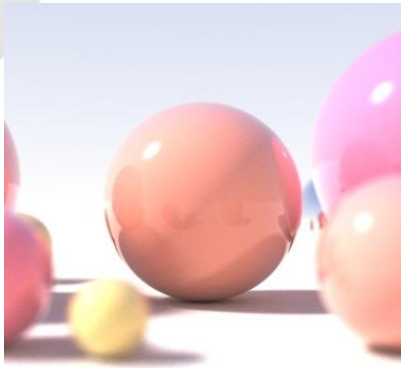
- Time
  - Daytime
  - Night sky
  - Sunset/sunrise
- Location
  - City
  - Nature (mountain, ocean etc.)
  - Space

(Donati, 2008)

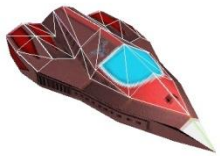
Local & Global Illumination







- **Local Illumination**
  - **Shading**: determining color value of each pixel
  - Light arriving at the surface directly from light source
  - Diffuse/Lambertian,
  - Specular reflection (Phong 1975; Blinn 1977; Cook-Torrance 1981; He et al. 1991)
  - Anisotropic (Kajiya 1985)
- **Global Illumination**
  - Indirect lighting due to reflection, refraction, scattering light from other surfaces or participating media in the scene
  - Ray tracing, radiosity

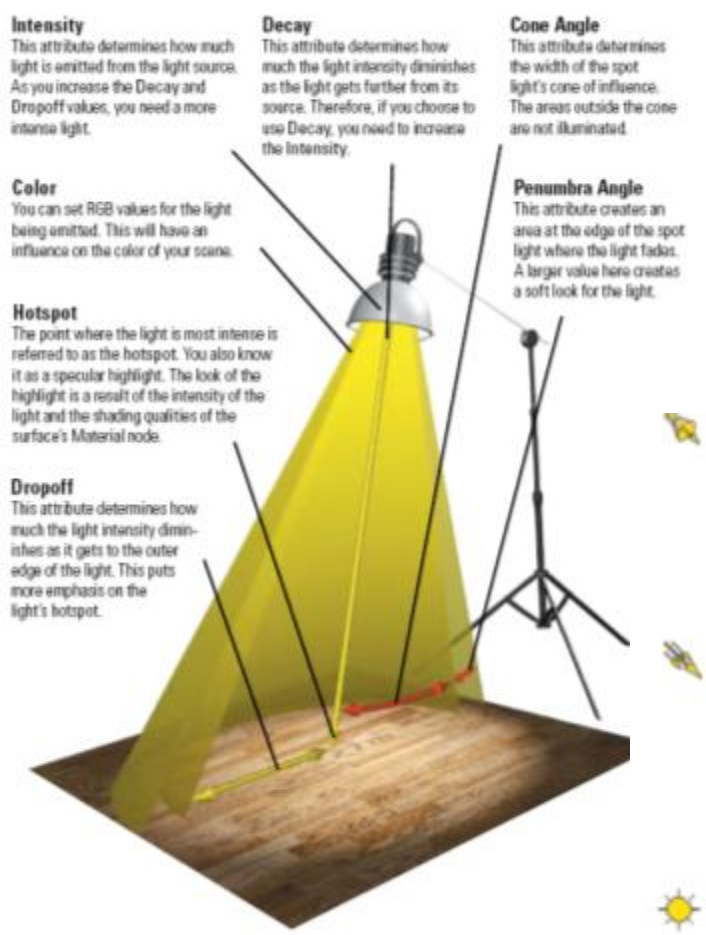


- **Texturing**
  - height, density, size, 2D/3D
  - contrast, regularity, directionality

(David Ebert, 1995)

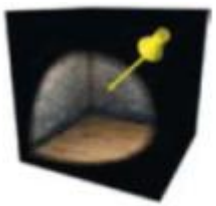
Source:
http://www.techquark.com/2009/08/ray-tracing-creating-stunning-photo.html

Light Types in Computer Graphics



Spot

Spot lights emit light that radiates from a point within a limited cone angle. You can use this cone angle to limit the area receiving light.



Directional

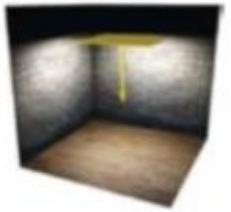
Directional lights use parallel rays of light to illuminate a scene. Shading is very uniform without any hotspots. These rays are similar to the light of the sun, which hits the earth with parallel rays.

Point

Point lights emit light in all directions, radiating from a single point. This creates an effect similar to a light bulb. This light creates subtle shading effects with definite hot spots.

Area

Area lights emit light using a two-dimensional area. The area light's icon can be used to help define the light's direction and intensity. A larger area light has a stronger intensity.



Ambient

Ambient lights emit light uniformly in all directions. The **Ambient Shade** attribute adds positional behavior. Bump maps are not visible with ambient light alone.

Volume

Volume lights emit light in all directions for a finite distance based on a 3D geometric shape. The light shape can be a box, a sphere, a cylinder or a cone.

Ambient Color
This attribute creates the effect of even illumination, without requiring a light source. In this image, the Ambient color has RGB values of 0.25, 0.25, 0.25 on all objects.

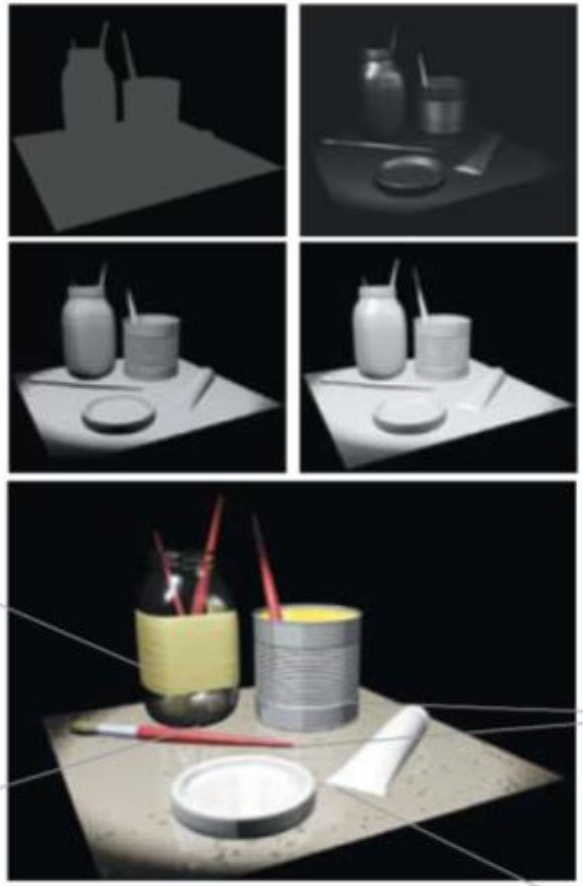
Diffuse
Diffuse determines how much light is absorbed and how much is scattered in all directions by surface imperfections. Rougher surfaces tend to have higher Diffuse values while smooth or mirror-like surfaces have Diffuse values that approach 0.

Color
Color is made up of red, green and blue attributes. The color of light and reflections will influence this base color.

Transparency
White is transparent, black is opaque and other values are semi-transparent. You can also use a texture map to create the appearance of holes in a surface.

Incandescence
This attribute can be used to make a surface appear to emit light. The Incandescence attribute is not actually emitting light and has no effect on other surfaces.

Glow
This attribute, found in the Special Effects section of the Material node, can be used to add the appearance of atmospheric noise to a surface.



Specular Highlights
Specular shading attributes determine the amount of light that is reflected at a consistent angle, resulting in an intensely bright region called a specular highlight. Perfectly smooth surfaces will have very bright, tiny highlights because there are no surface imperfections to distort the reflection angle. Rougher surfaces like brushed metals will have a softer highlight.

Combined Effect
In real life, the proportions of the specular and diffuse components of the total reflected light will vary, depending on the characteristics of the surface.

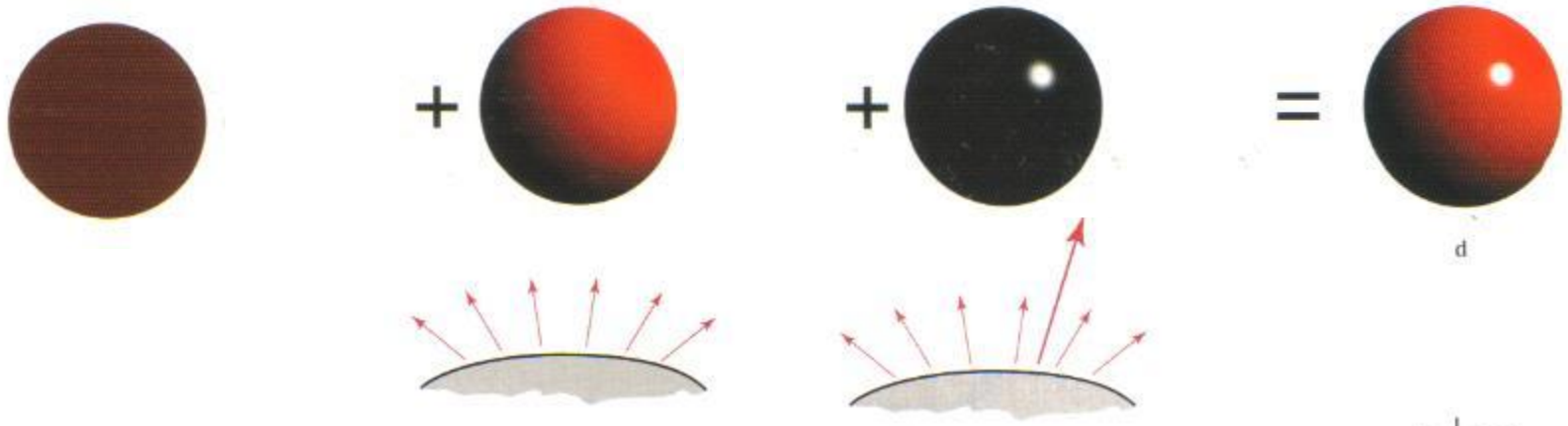
Bump
This attribute lets you add surface relief by using a texture map to alter the direction of the surface Normals.

Reflectivity
This attribute controls the amount a surface reflects its environment. This environment could be a 3D texture map connected to the material's Reflected Color, or actual Raytraced reflections of objects in the scene.

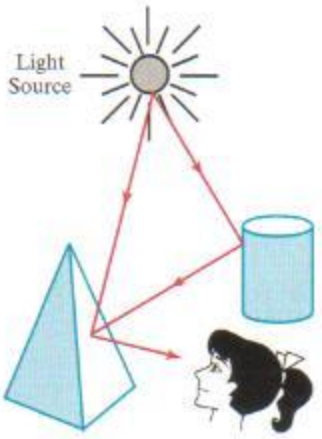
Reflected Color
This attribute can be texture mapped to define a reflected environment without relying on Raytraced reflections. These texture maps are positioned in world space and can be assigned to various materials to make sure the scene's reflections are consistent.

(Autodesk, 2007)

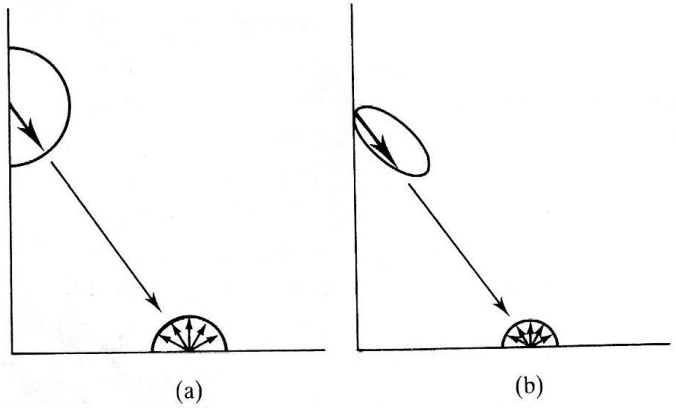
Surface Shaders

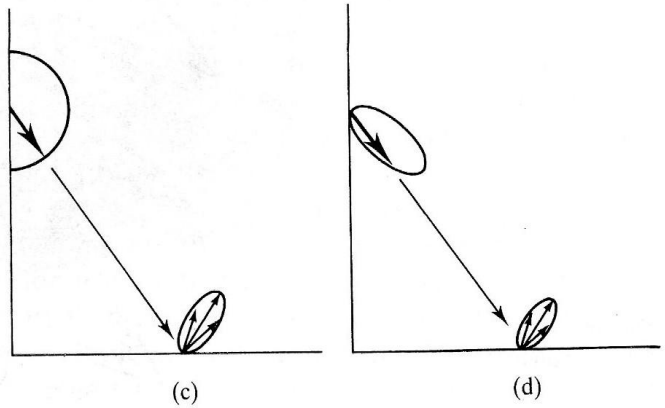


- Ambient
  - Even illumination (environment color)
  - Background light
- Diffuse
  - Main color
  - Scattering
  - Absorption
  - Mirror (~0)
  - Rough surface (>0)
- Specular
  - Highlight
  - Reflection



Four Mechanisms of Light Transport





(a). Diffuse to diffuse transfer

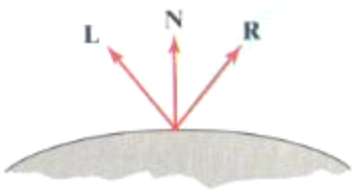
(b). Specular to diffuse transfer

(c). Diffuse to specular transfer

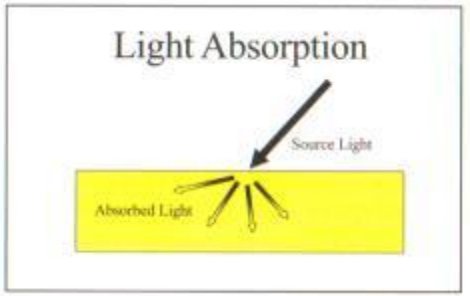
(d). Specular to specular transfer

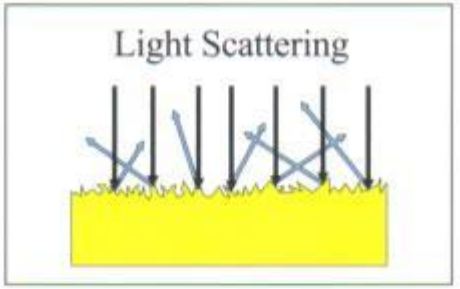
(Watt, 1999)

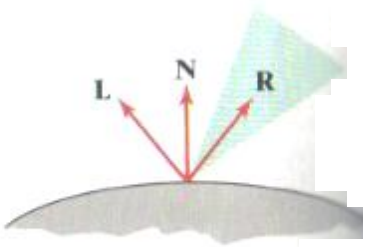
Reflection, Scattering & Absorption.



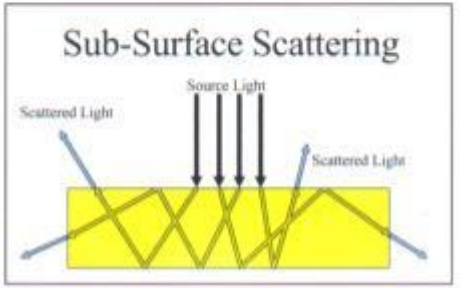
Shiny Surface
(Large n_s)

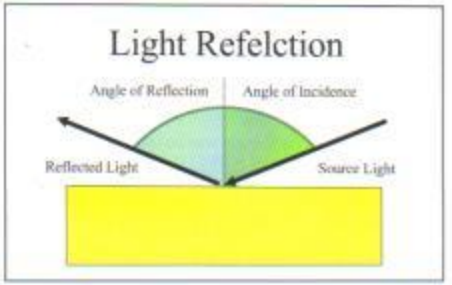






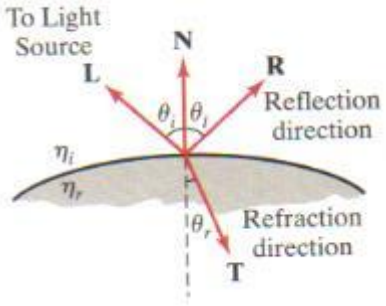
Dull Surface
(Small n_s)

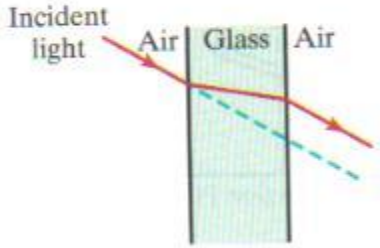




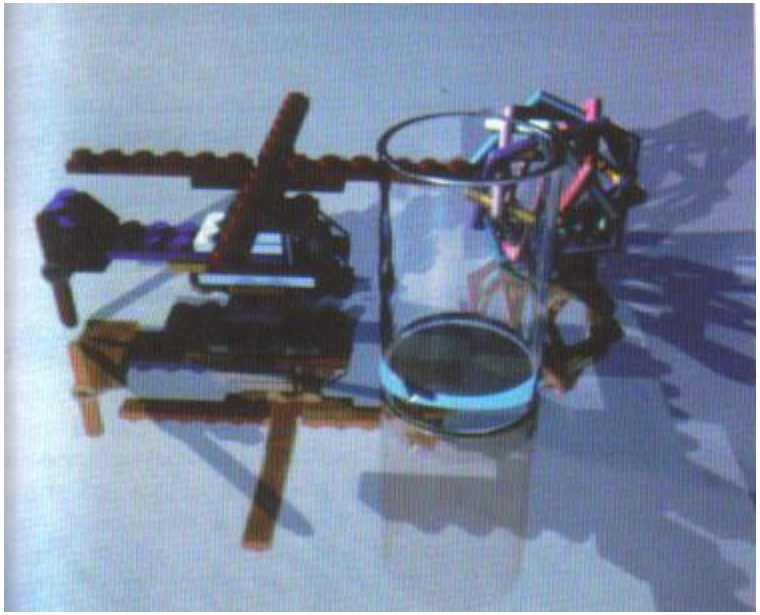
(Hearn, 2004) (Donati, 2008)

Transparent Surfaces (Refraction)





(Hearn & Baker, 2003)

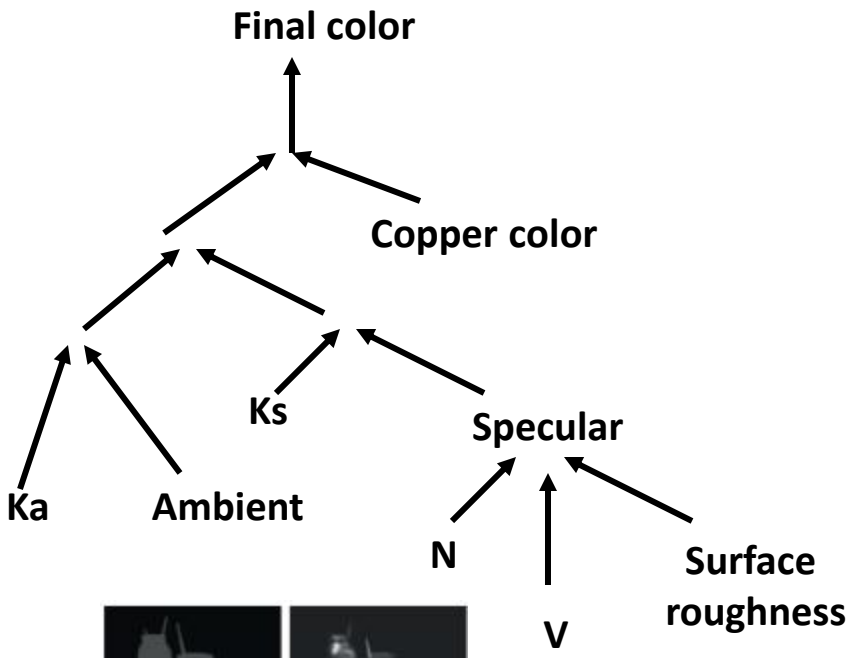


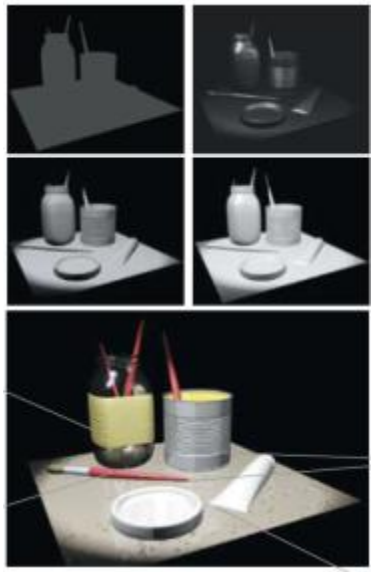
AVERAGE INDEX OF REFRACTION FOR COMMON MATERIALS

Material	*Index of Refr*
Vacuum (Air or Other Gas)	1.00
Ordinary Crown Glass	1.52
Heavy Crown Glass	1.61
Ordinary Flint Glass	1.61
Heavy Flint Glass	1.92
Rock Salt	1.55
Quartz	1.54
Water	1.33
Ice	1.31

How can we build a framework for material rendering?

Shade Tree (Cook, 1984)





- A shade tree is a tree of nodes, each of which takes parameters from its children and produces parameters for its parent.
- *Pixar's RenderMan API* defines a set of key places in the rendering process at which user-defined or system-defined shaders can be called. For example, a surface shader returns the light reflected in a specified direction given a point on the surface, its orientation, and a set of light sources.

Source: Computer Graphics: Principles and Practice
By Foley et al. (1995)

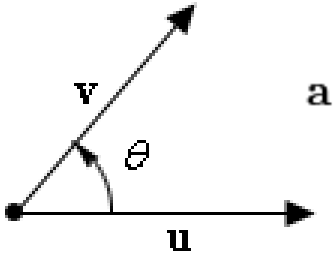
How do we control light intensity?

What is the underlying mechanism of simulating light?

$$A = (x_1, y_1)$$

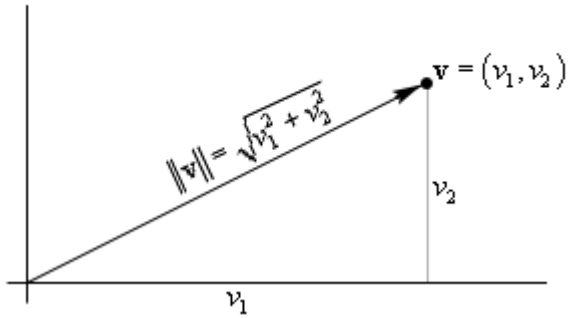
$$B = (x_2, y_2)$$

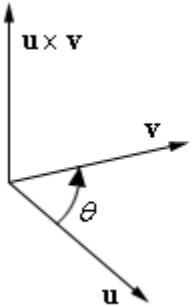
$$\overrightarrow{AB} = (x_2 - x_1, y_2 - y_1)$$

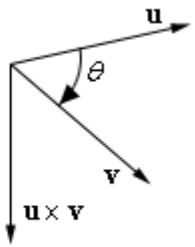


$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}\right)$$



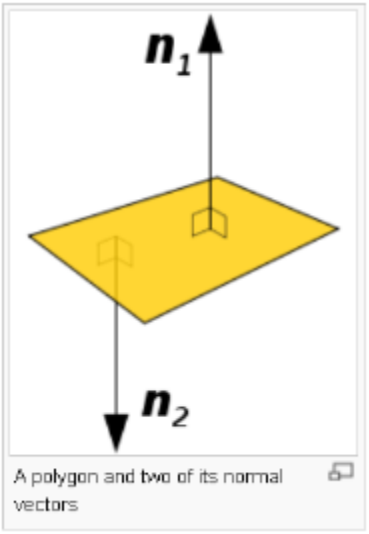




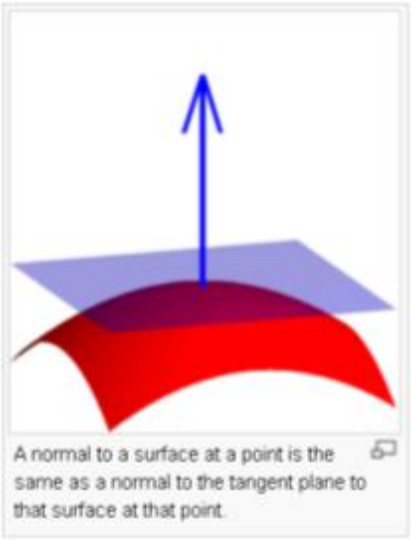
$$\|\mathbf{u} \times \mathbf{v}\| = \|\mathbf{u}\| \, \|\mathbf{v}\| \sin\theta$$

Source: http://www.intmath.com/analytic-trigonometry/7-inverse-trigo-functions.php

Surface Normal (Normal Vector)

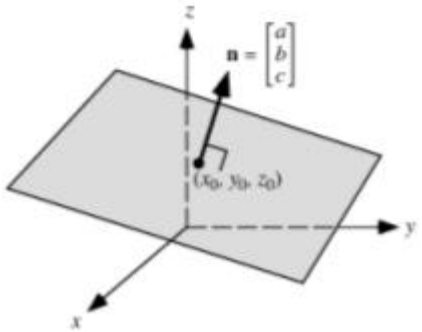


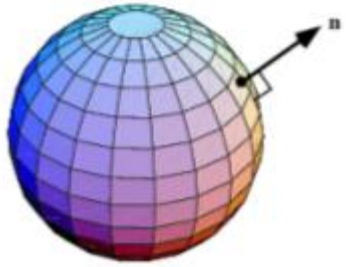
A polygon and two of its normal vectors


A normal to a surface at a point is the same as a normal to the tangent plane to that surface at that point.

Source
http://en.wikipedia.org/wiki/Surface_normal

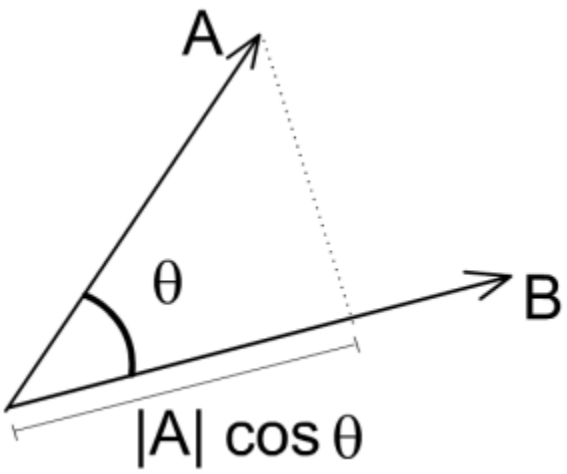
A **surface normal**, or simply **normal**, to a flat surface is a vector that is perpendicular to that surface.





Source:
http://mathworld.wolfram.com/NormalVector.html

Dot product



The dot product of two vectors **a** = [a_1, a_2, ... , a_n] and **b** = [b_1, b_2, ... , b_n] is defined as:

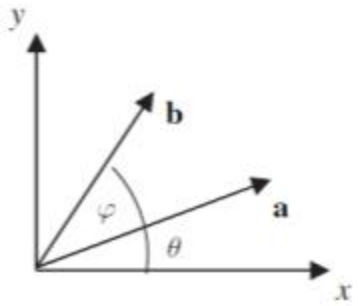
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

where Σ denotes summation notation and *n* is the dimension of the vector space.

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

where ||**a**|| and ||**b**|| denote the length of **a** and **b** and θ is the angle between them.

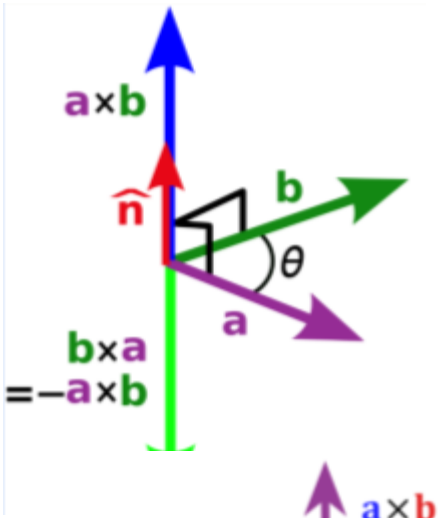
Why we need it?
Find angle between vectors.

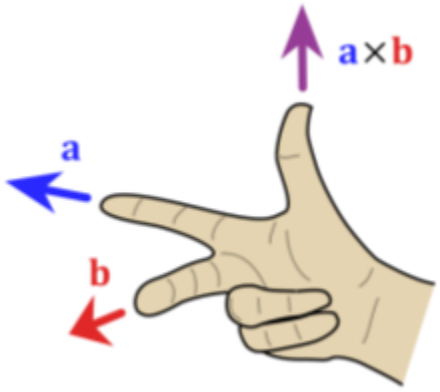


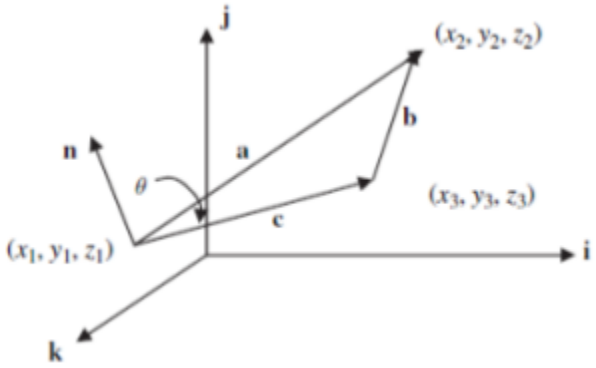
Source:
http://en.wikipedia.org/wiki/Dot_product

$$\begin{aligned}
\mathbf{a} &= (|\mathbf{a}| \cos\theta, |\mathbf{a}| \sin\theta) \\
\mathbf{b} &= (|\mathbf{b}| \cos\varphi, |\mathbf{b}| \sin\varphi) \\
\mathbf{a} \cdot \mathbf{b} &= |\mathbf{a}||\mathbf{b}| \cos\theta \cos\varphi + |\mathbf{a}||\mathbf{b}| \sin\theta \sin\varphi \\
&= |\mathbf{a}||\mathbf{b}| \cos(\varphi - \theta)
\end{aligned}$$

Cross Product





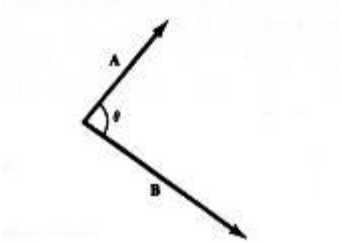


The cross product **a** × **b** is defined as a vector **c** that is perpendicular to both **a** and **b**, with a direction given by the right-hand rule and a magnitude equal to the area of the parallelogram that the vectors span.

$$\mathbf{a} \times \mathbf{b} = ab \sin\theta \, \mathbf{n}$$

Source:
http://en.wikipedia.org/wiki/Cross_product

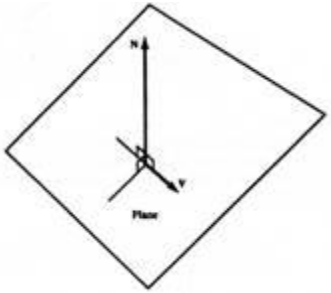
Dot Product & Normal Vector

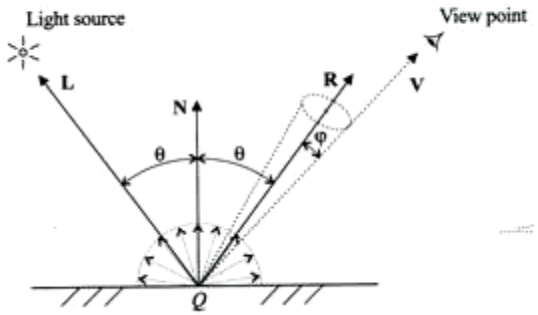


$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}||\mathbf{B}| \cos\theta$$

$$\cos\theta = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|}$$

- Importance of Normal Vector
  - Lighting simulation
  - Polygon facing direction
  - Dynamics & collision

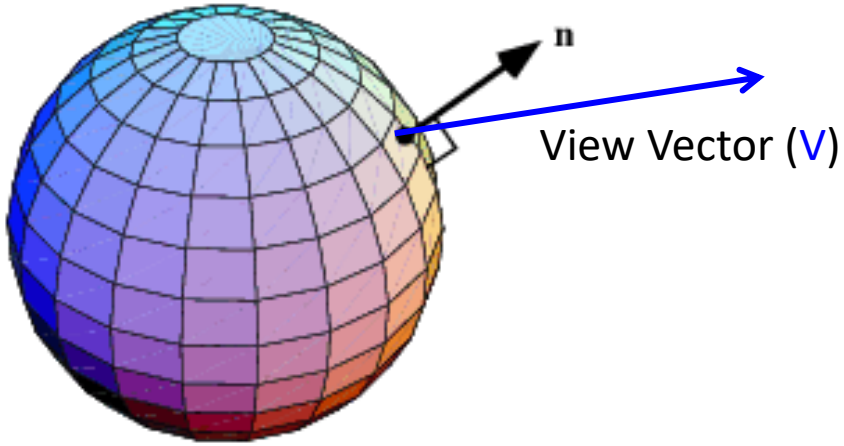


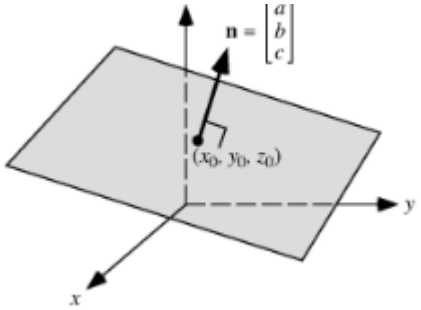


(Xiang & Plastock, 2001)

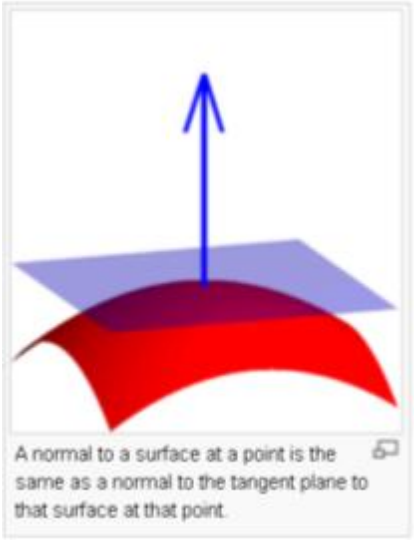
Surface Normal (Normal Vector)

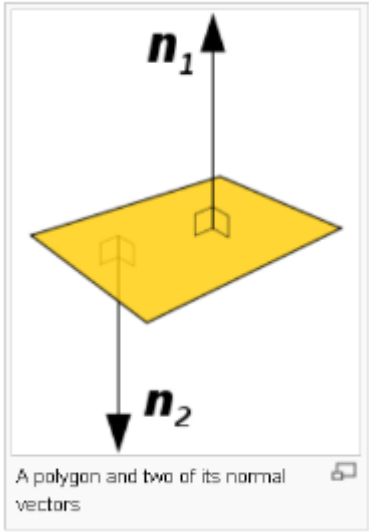
A **surface normal**, or simply **normal**, to a flat surface is a vector that is perpendicular to that surface.





1. Find two vectors
2. Find dot product,
3. Find the angle between two vectors
4. Find the cross product of two vectors


A normal to a surface at a point is the same as a normal to the tangent plane to that surface at that point.

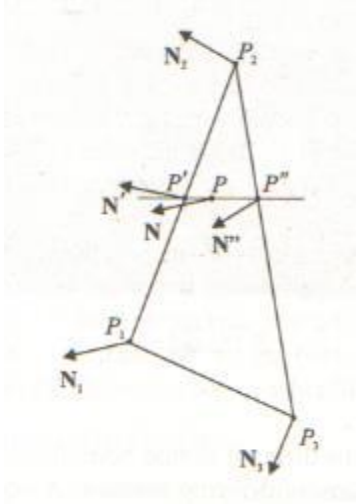


A polygon and two of its normal vectors

Source:
http://mathworld.wolfram.com/NormalVector.html

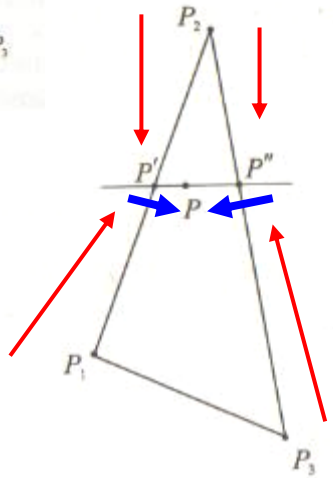
Source
http://en.wikipedia.org/wiki/Surface_normal

Mathematical Shading Models

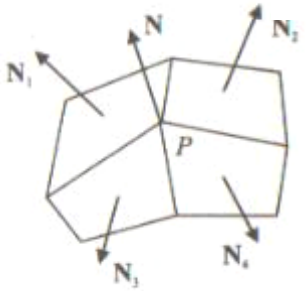


Phong

- Gouraud Shading (1971)
  - Intensity interpolation shading



Illuminate the vertices of polygon mesh

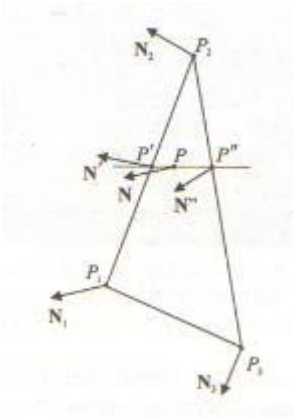


  - Average normal vectors of adjacent polygons (normalize polygon surface normals)

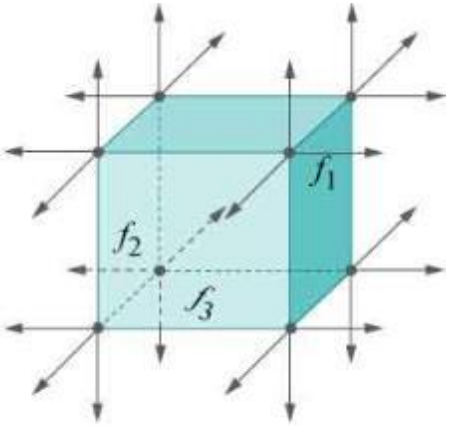
(Xiang, & Plastock, 2001).

Different Normal Vector Effect for Lighting Calculation in 3D Graphics

- **Phong**

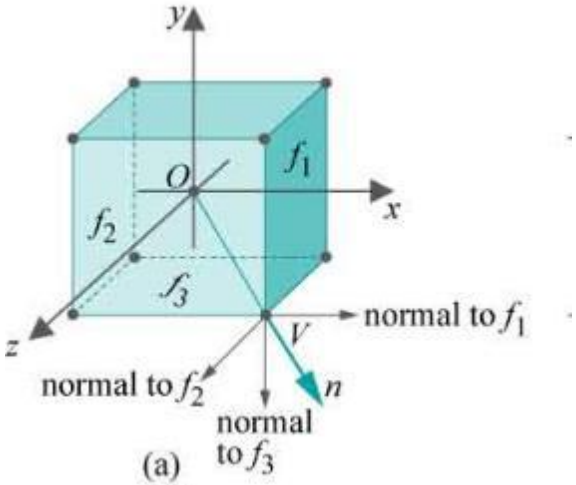


(Xiang, Z, & Plastock, R. A. (2001).



(b)

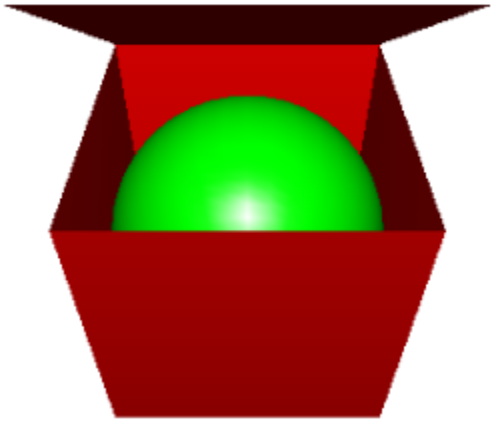
Unaveraged normals!

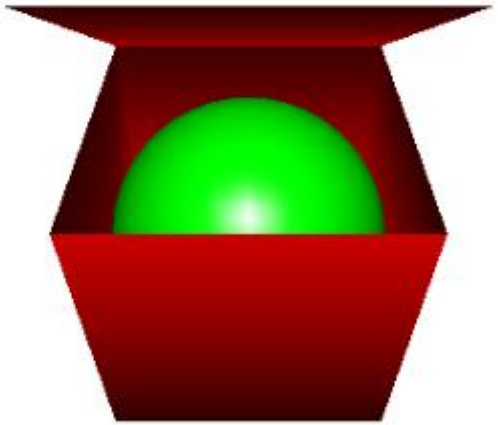


(a)

Averaged normals!

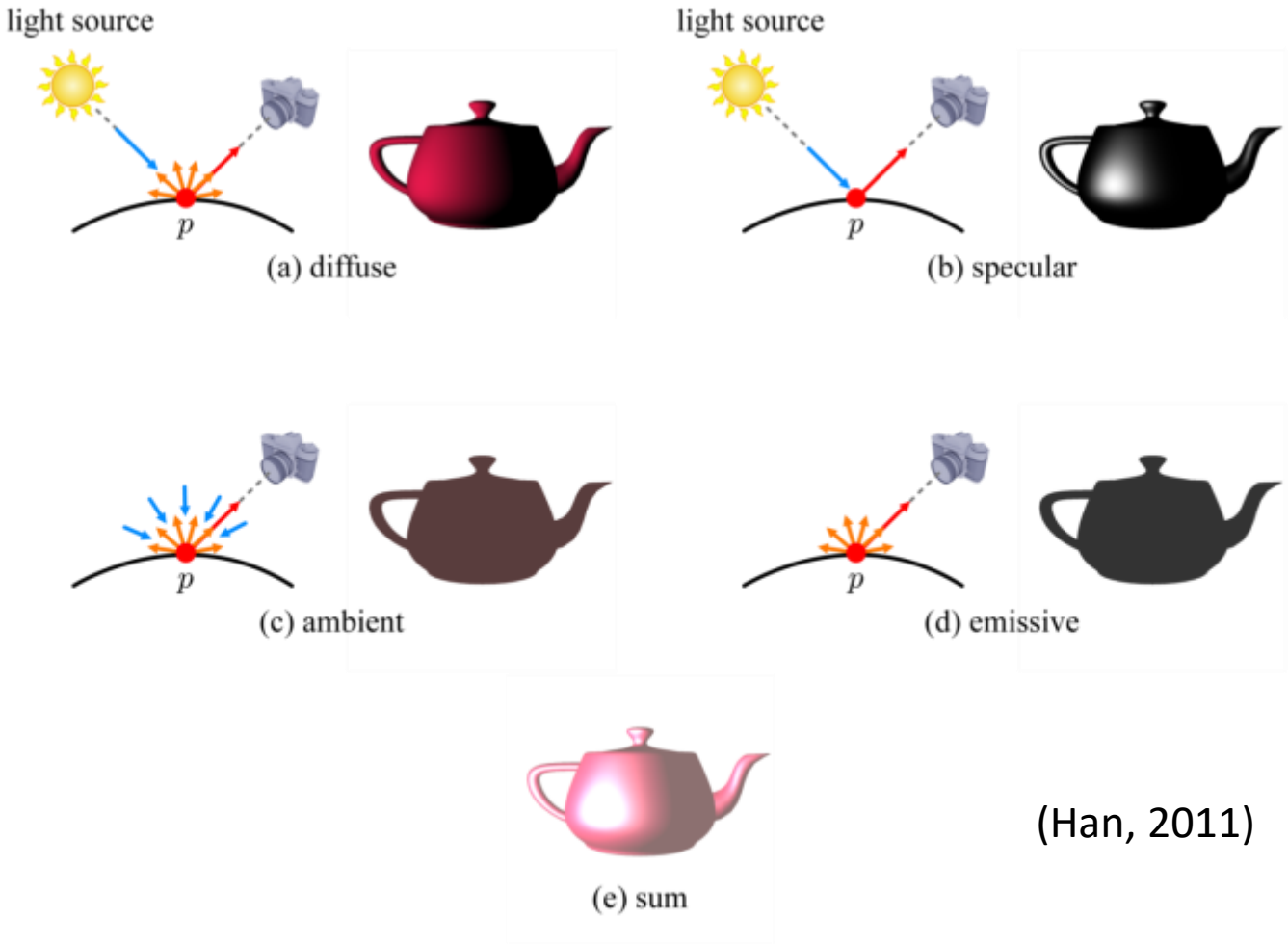
(Guha, 2010)





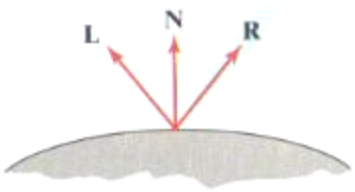
Phong Lighting Model

- The Phong model sums the four terms!!

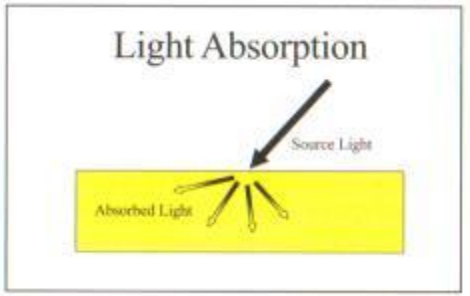


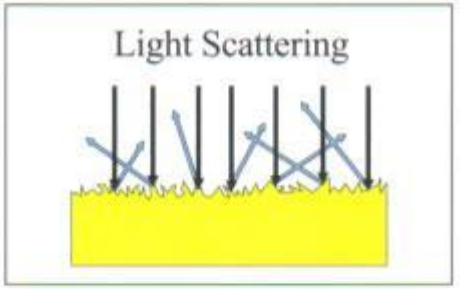
(Han, 2011)

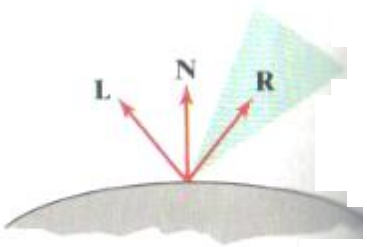
Reflection, Scattering & Absorption.



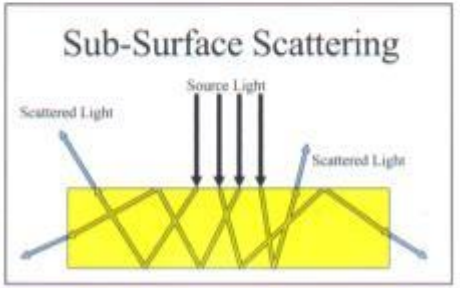
Shiny Surface
(Large n_s)

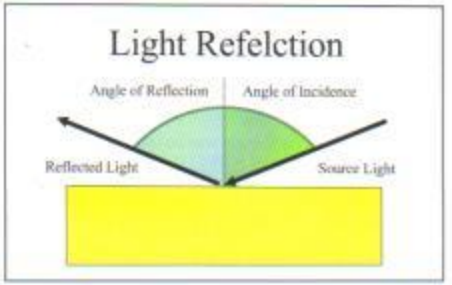






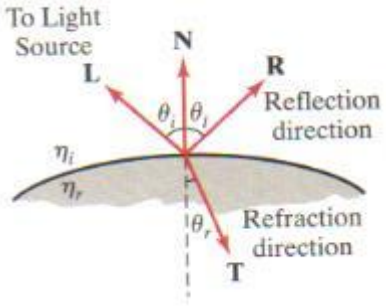
Dull Surface
(Small n_s)

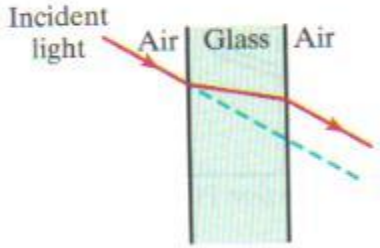




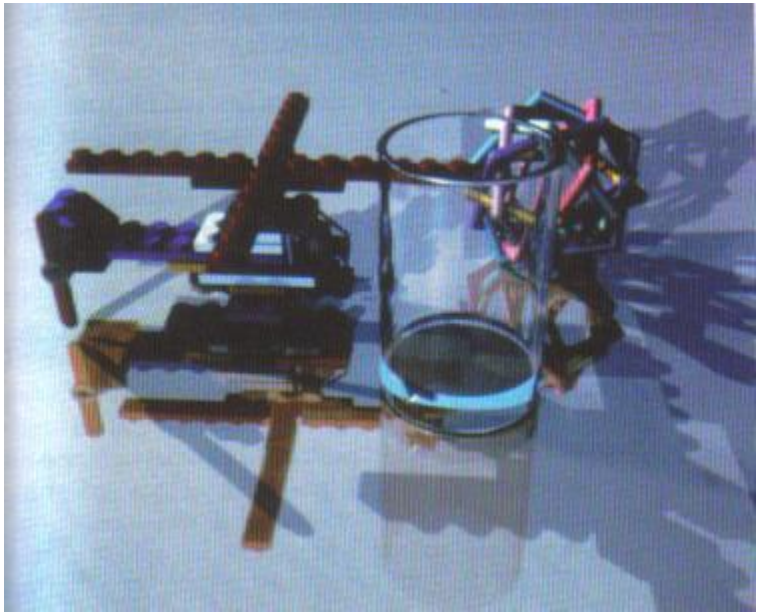
(Hearn, 2004) (Donati, 2008)

Transparent Surfaces (Refraction)





(Hearn & Baker, 2003)



AVERAGE INDEX OF REFRACTION FOR COMMON MATERIALS

Material	*Index of Refr*
Vacuum (Air or Other Gas)	1.00
Ordinary Crown Glass	1.52
Heavy Crown Glass	1.61
Ordinary Flint Glass	1.61
Heavy Flint Glass	1.92
Rock Salt	1.55
Quartz	1.54
Water	1.33
Ice	1.31

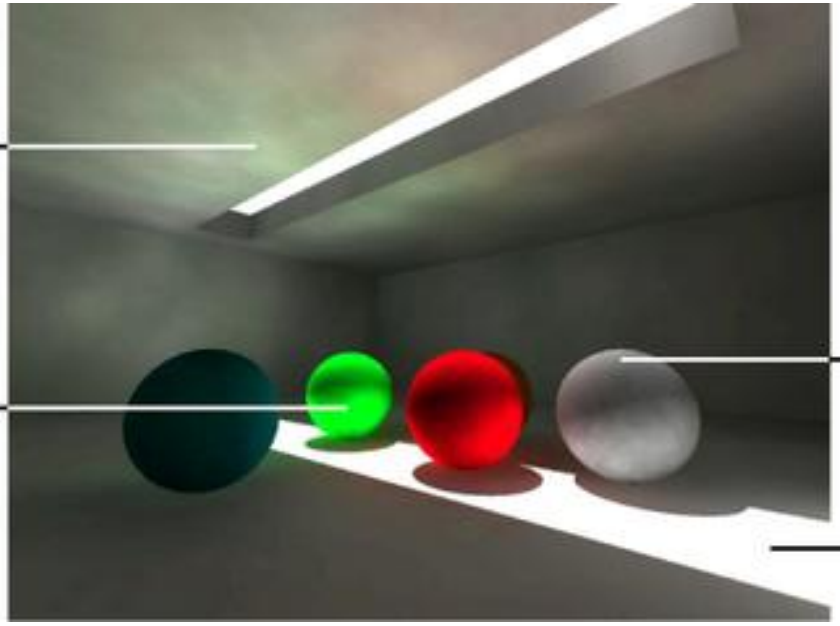
How can we build a framework for material rendering?

Among ambient, diffuse, or specular reflectance model, which one affects the perceptual depth of an object?

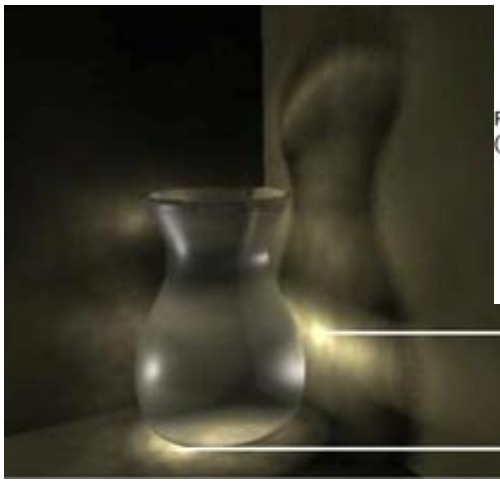
- The ambient intensity, which does not depend on eye or light direction, is uniform across vertices of the ball and cannot, therefore, provide the sense of depth that obtains from a contrast in color values across the surface.
- Diffuse light, on the other hand, which varies across the surface depending on light direction, can provide an illusion of depth.
- Even though there is a specular highlight, sensitive to both eye and light direction, it's too localized to provide much contrast. Reducing the shininess spreads the highlight but the effect is not a realistic perception of depth.
  - **Diffusive reflectance lends three-dimensionality.**
  - **Reduce the diffuse reflectance making an object look flat losing three-dimensionality.**

(Guha, 2010)

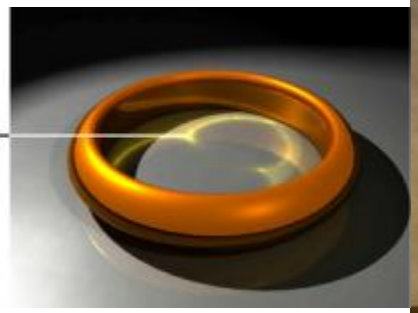
Indirect Lighting



(Autodesk, 2007)



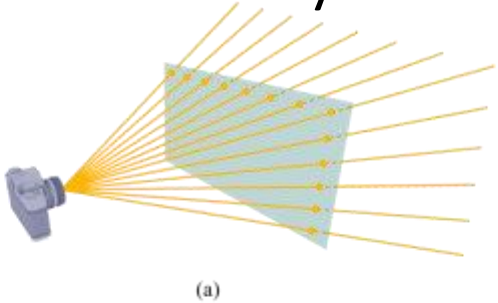
Reflected caustics
(Image by Dan Pressman)

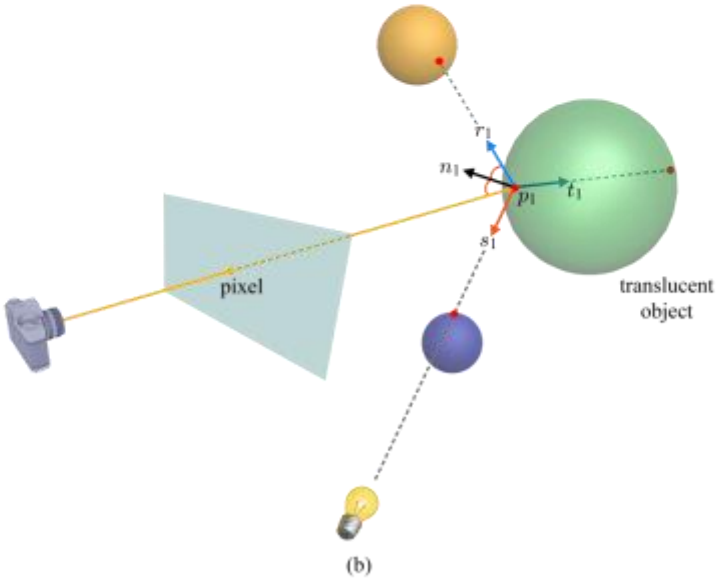


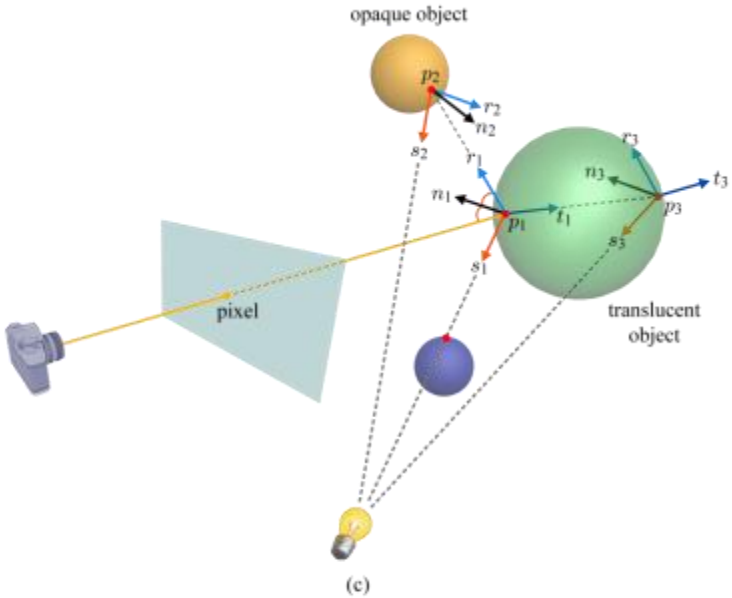
Refracted caustics
(Image by Lisa Williamson)

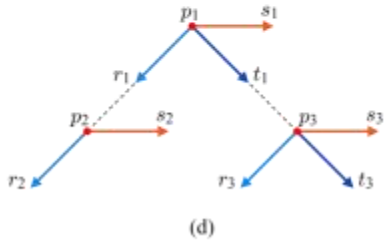
Global Illumination – Ray Tracing (Han, 2011)

- When a primary ray is shot and intersects an object, three secondary rays would be spawned: a shadow ray, a reflection ray, and a refraction ray.

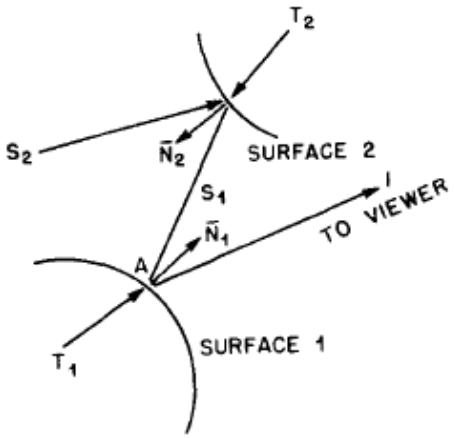


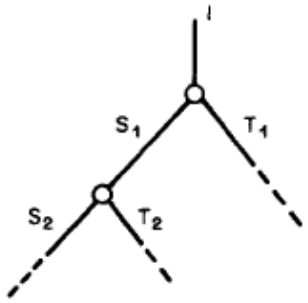


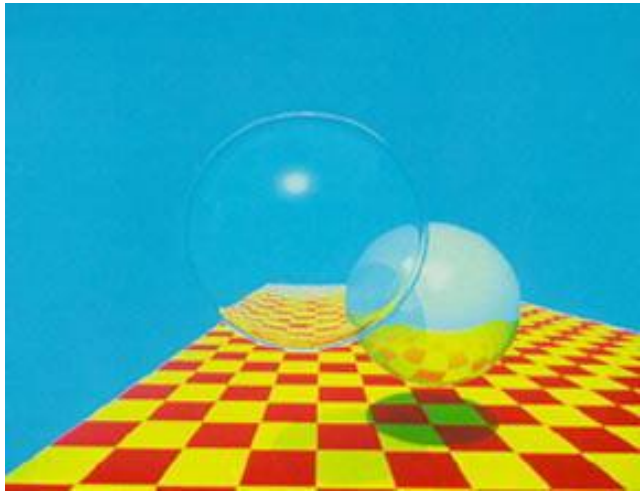




Recursive ray-tracing



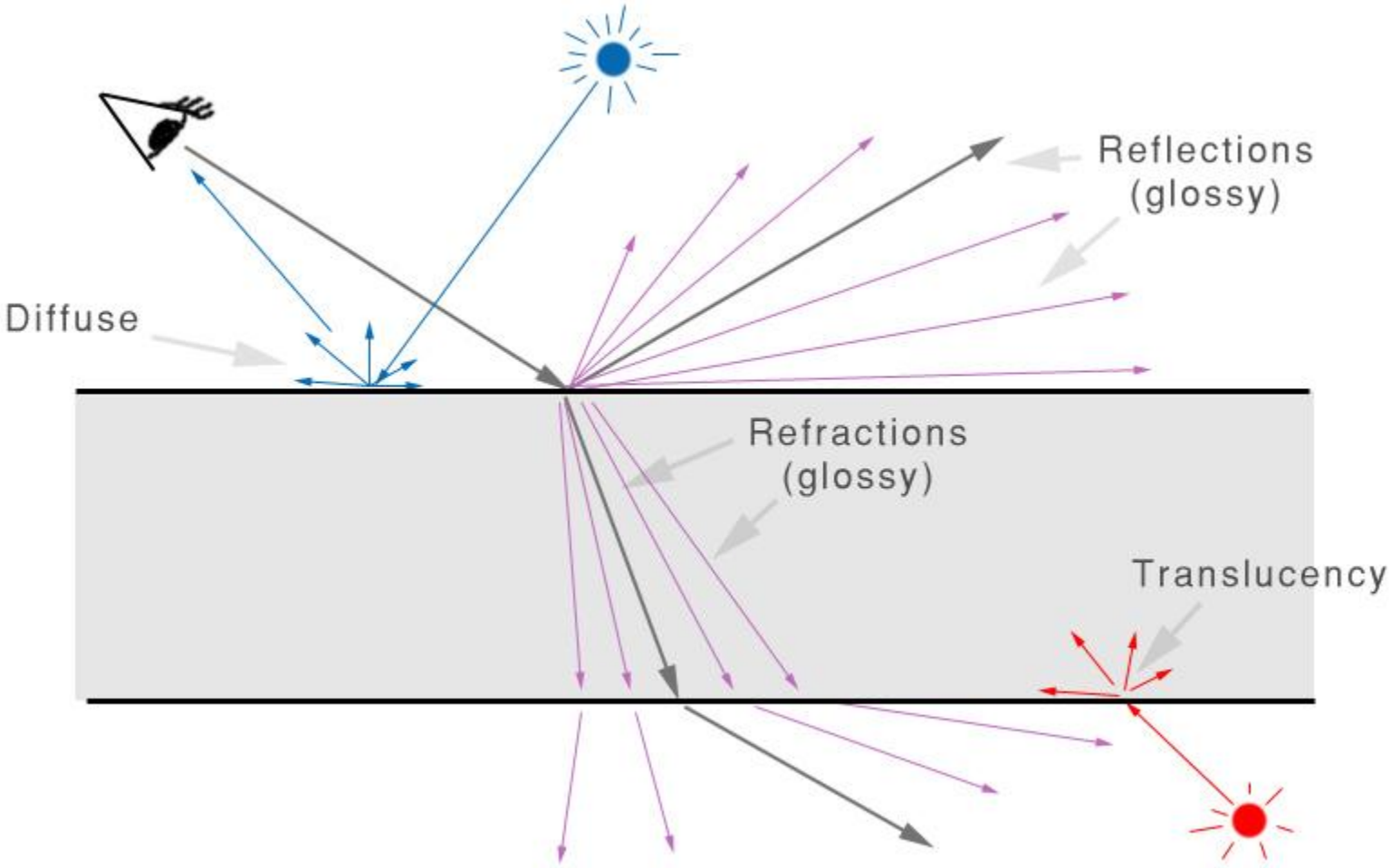




(Turner Whitted, 1978)

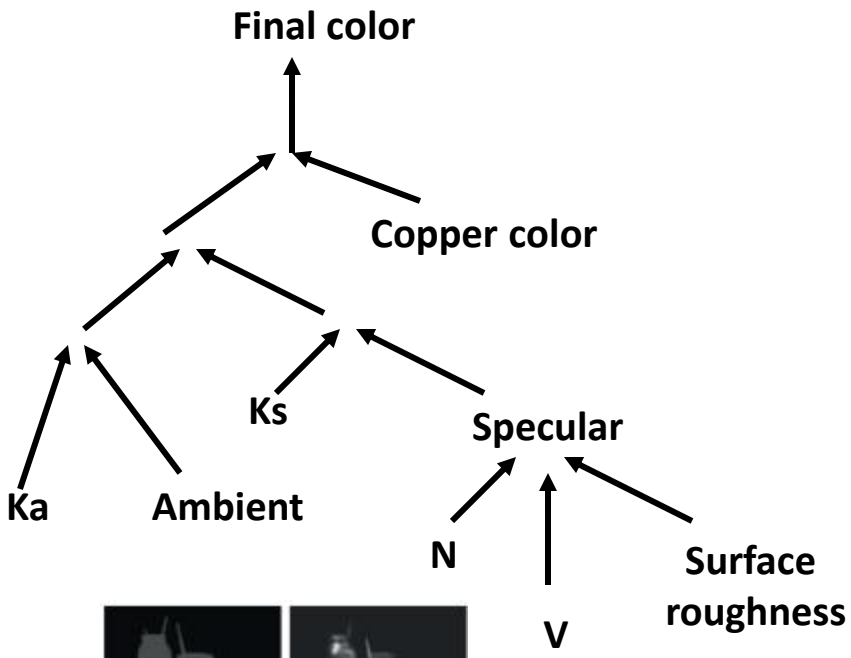
http://dl.acm.org/citation.cfm?id=358882

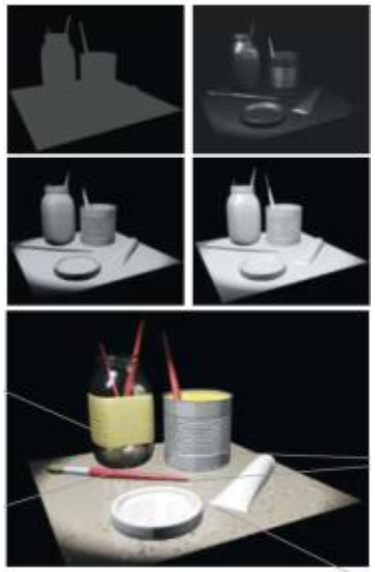
https://design.osu.edu/carlson/history/lesson5.html



Source:
http://docs.autodesk.com/MENTALRAY/2013/ENU/mental-ray-help/

Shade Tree (Cook, 1984)

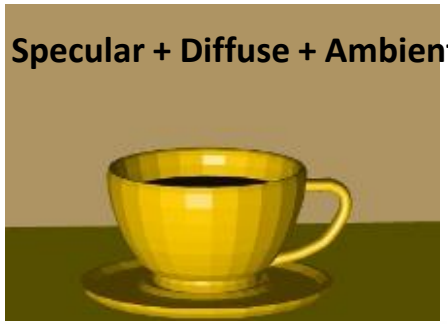


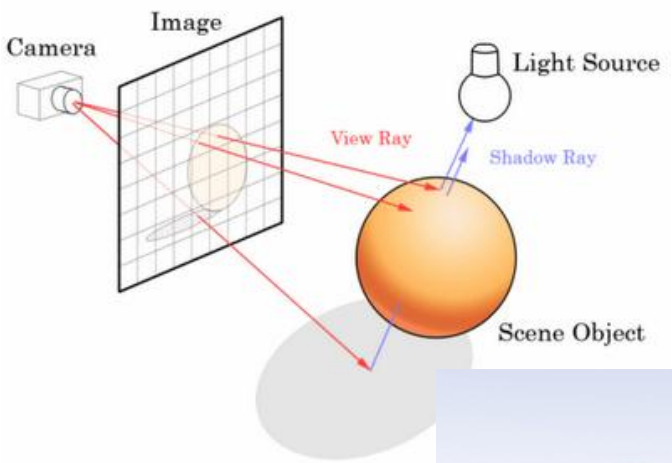


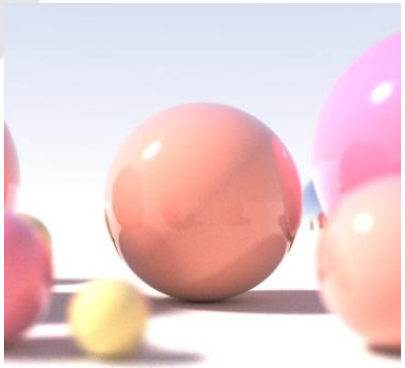
- A shade tree is a tree of nodes, each of which takes parameters from its children and produces parameters for its parent.
- *Pixar's RenderMan API* defines a set of key places in the rendering process at which user-defined or system-defined shaders can be called. For example, a surface shader returns the light reflected in a specified direction given a point on the surface, its orientation, and a set of light sources.

Source: Computer Graphics: Principles and Practice
By Foley et al. (1995)

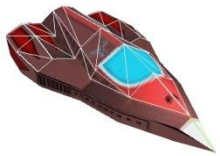
Local & Global Illumination







- **Local Illumination**
  - **Shading**: determining color value of each pixel
  - Light arriving at the surface directly from light source
  - Diffuse/Lambertian,
  - Specular reflection (Phong 1975; Blinn 1977; Cook-Torrance 1981; He et al. 1991)
  - Anisotropic (Kajiya 1985)
- **Global Illumination**
  - Indirect lighting due to reflection, refraction, scattering light from other surfaces or participating media in the scene
  - Ray tracing, radiosity



- **Texturing**
  - height, density, size, 2D/3D
  - contrast, regularity, directionality

(David Ebert, 1995)

Source:
http://www.techquark.com/2009/08/ray-tracing-creating-stunning-photo.html

In summary, what is Illumination? (Han, 2011)

- Illumination or lighting refers to the techniques handling the interaction between light sources and objects.
- **The lighting models are divided into two categories.**
  - Local illumination considers only direct lighting in the sense that the illumination of a surface depends solely on the properties of the light sources and the surface materials. This has been dominant in real-time graphics.
  - In the real world, however, every surface receives light indirectly. (Even though a light source is invisible from a particular point of the scene, light can still be transferred to the point through reflections or refractions from other surfaces of the scene.) For indirect lighting, the global illumination (GI) model considers the scene objects as potential lighting sources.
- **Problems of interactive GI**
  - The cost is often too high to permit interactivity.
  - The rasterization-based architecture of GPU is more suitable for local illumination.
- **Current status of GI**
  - Approximate GI instead of pursuing precise GI.
  - Pre-compute GI, store the result in a texture, and use it at run time.

Image-based lighting and physical shading

CHROME BALL DERIVED ENVIRONMENT MAP EG

CHROME AND GREY SPHERE ON LOCATION

Source:
Ben Snow at Industrial Light and Magic

Final Gather and HDRI
(High Dynamic Range Imaging)

(Debevec, 2000)

Paul Debevec

http://projects.ict.usc.edu/graphics/HDRShop/

Reinhard, Erik; Ward, Greg; Pattanaik, Sumanta; Debevec, Paul (2006). *High dynamic range imaging: acquisition, display, and image-based lighting*. Amsterdam: Elsevier/Morgan Kaufmann. p. 7

In image processing, computer graphics, and photography, **high-dynamic-range imaging** (**HDRI** or just **HDR**) is a set of techniques that allow a greater dynamic range of luminance between the lightest and darkest areas of an image than current standard digital imaging techniques or photographic methods. This wide dynamic range allows HDR images to more accurately represent the range of intensity levels found in real scenes, ranging from direct sunlight to faint starlight.

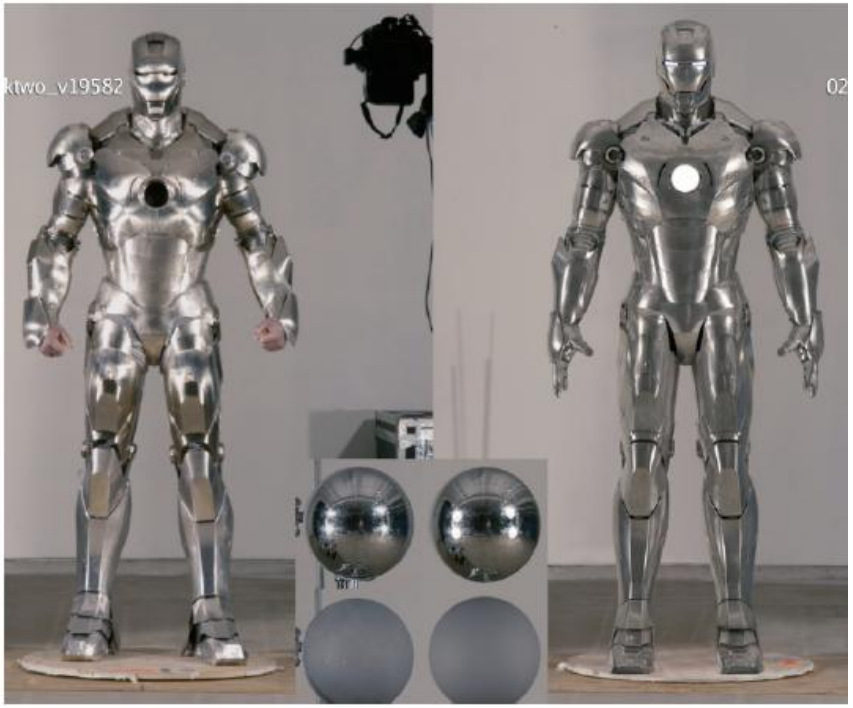
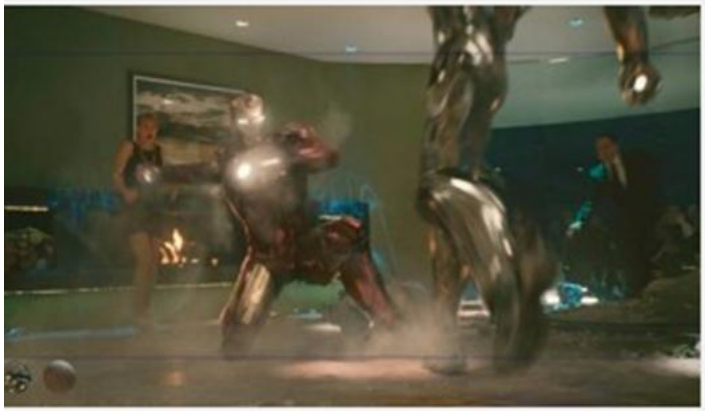
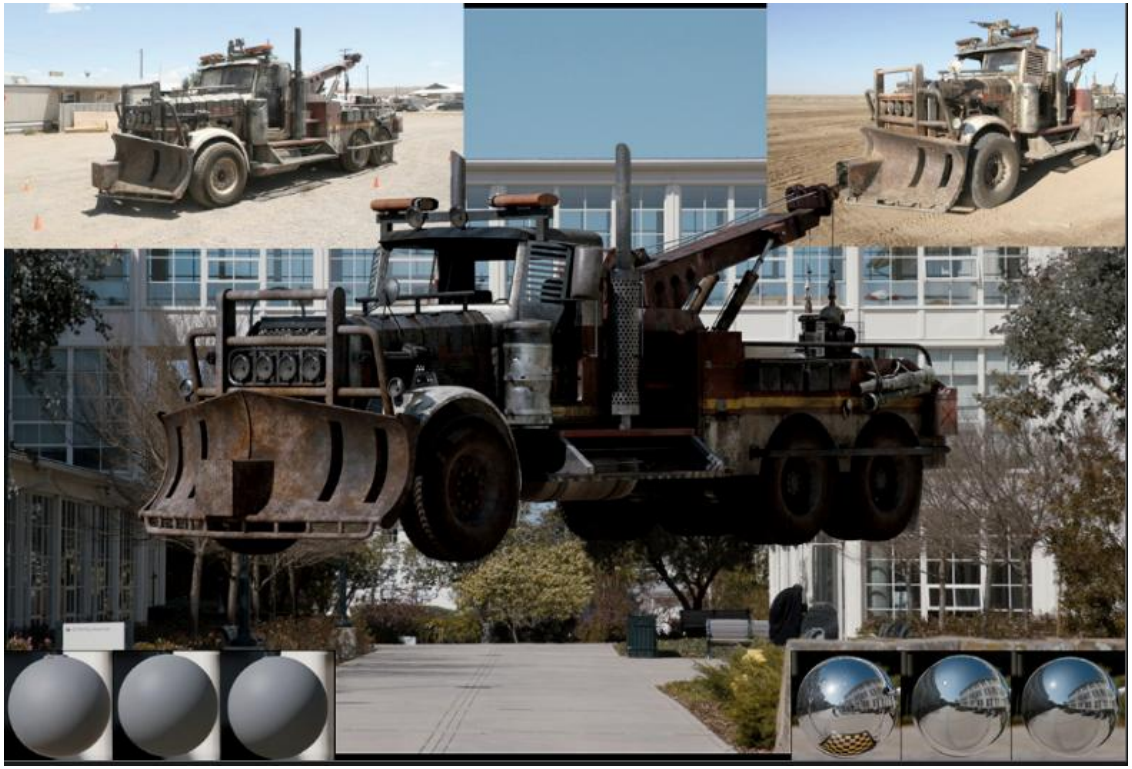


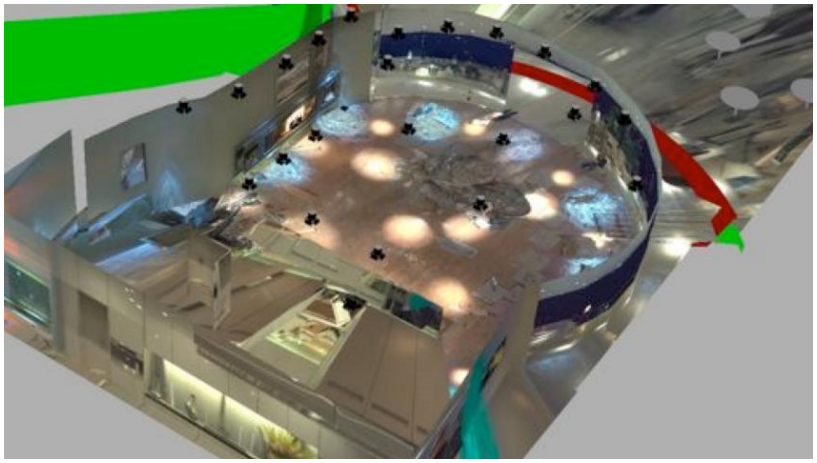
IMAGE: SIDE BY SIDE CG and PRACTICAL



Note the ball reference at the bottom

Source:
Ben Snow at Industrial Light and Magic





Source:
Ben Snow at Industrial Light and Magic