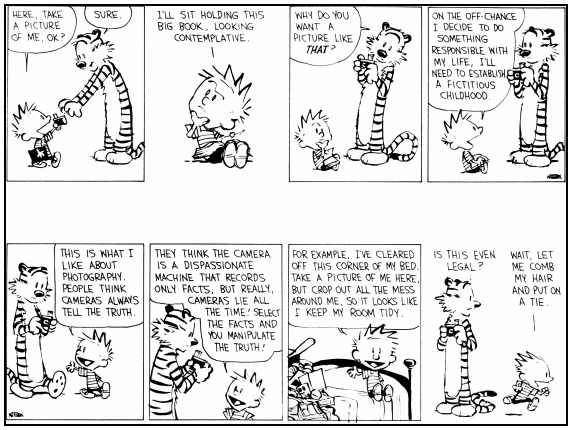


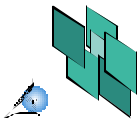
Transformations

MIT EECS 6.837

Frédo Durand and Seth Teller

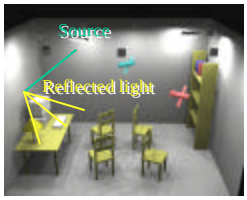
Getting Acquainted

- Frédo Durand
- Research in rendering (image generation)
- A lot of visibility



For real-time rendering

"trivially" hidden



For lighting simulation

Image-based editing

- 3D model from single photo
- Editing a la Photoshop

Input image

New viewpoint

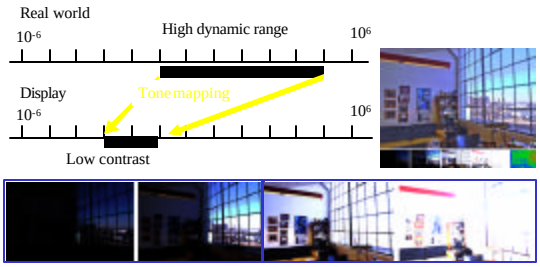
Relighting

Hotel Lobby

Dali painting

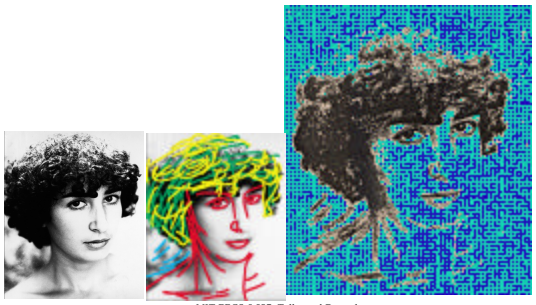
Tone mapping

- Display contrast is limited



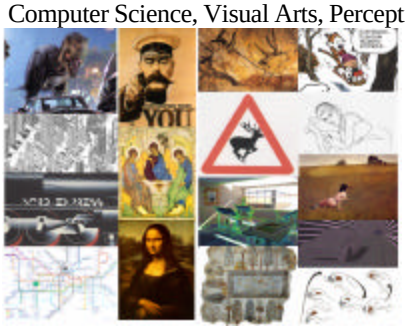
Non-photorealistic rendering

- Qualities of traditional media



Interdisplinary study of depiction

- Computer Science, Visual Arts, Perception



Assignment

- Due Friday 9/20 at 5pm
- Inventor modeling
  - Program that outputs an inventor ascii file
  - With at least 4 parameters
  - Will be used in assignment 2 by others
- More about Inventor modeling on Thursday
  - http://web.mit.edu/acs/www/graphics.html#Open
  - http://web.mit.edu/ivlib/www/iv.html
  - http://www.sgi.com/software/inventor/vrml/TIMSummary.html
  - http://www.sgi.com/software/inventor/manuals.html

Inventor on Windows

- TGS
  - http://www.tgs.com/index.htm?pro_div/oiv_main.hm~main
- GNU
  - http://www.studierstube.org/openinventor/

Questions?

Today's lecture: transformations

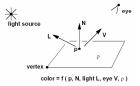
- Graphics involves a lot of matrix algebra
- Important for modeling
- Important for rendering

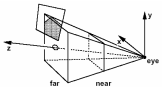
- Shirley 4 & 5
- Some slides courtesy of Leonard McMillan

Today's lecture: transformations

- Motivation
- Transformations
- Matrices
- Composition, object vs. world space
- Playtime (if time permits)

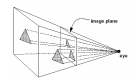
Classical Rendering Pipeline

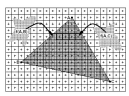


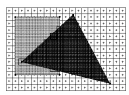


- Model Traversal
  - Graphics primitives issued for display
- Modeling Transformations
  - Object-space -> World-space
- Lighting (Shading)
  - Vertices lit according to local lighting model
- Clipping Transformation
  - World-space -> Eye-space -> Clip-space
  - Portions outside 3D view volume are removed

Classical Rendering pipeline II



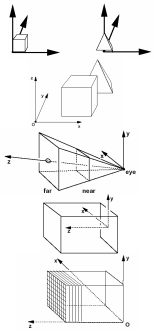




- Perspective Projection
  - 3D scene projected onto 2D image plane
  - Fallof of apparent size with distance is simulated
  - Clip-space -> Screen-space
  - Portion of image plane mapped to raster viewport
- Scan Conversion: Rasterization
  - Object diced into horizontal spans, one per raster
  - Depth, color, other attributes interpolated along object boundaries
  - Depth, color, etc. interpolated across each span, yielding pixels
- Visibility Resolution
  - Established per-pixel, (typically) in hardware
- **Note**: other organizations possible!

Coordinate systems

- Object space
  - coordinate system local to each object
- World space
  - common coordinate system into which all primitives resolved
- Eye space (camera space)
  - viewer-centered coordinate system derived from view frustum
- Clip space
  - parallelepiped [-1,1]x[-1,1]x[-1,1]
- Screen space (3D and 2D)
  - discretized parallelepiped, indexed according to hardware attributes
  - (floating-point, x –y resolution, z resolution; etc.)



Questions?

Today's lecture: transformations

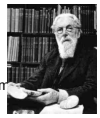
- Motivation
- Transformations
- Matrices
- Composition, object vs. world space
- Playtime (if time permits)

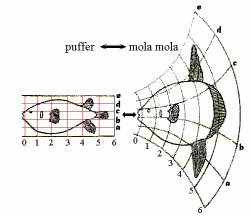
3D Coordinate Transformations

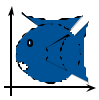
- Transformation: operator on geometric object
- Abstractly:
  - M maps points p to points M(p) = p' = (x'; y'; z')
- Important point: *invariants of a transformation*
- Can be used
  - to move points around
  - To change coordinate systems

Visual tribute to D'arcy Thompson

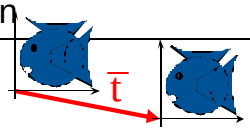
- The first biomathematician
- Transformation & evolution
- http://www-gap.dcs.stand.ac.uk/~history/Miscellaneous/darcy.htm







Translation



- x' = x + tx
- y' = y + ty
- z' = z + tz
- For every translation *there exists an inverse function* which undoes the translation. In our case:
- x = x' - tx
- y = y' – ty
- z = z' - tz
- There also exists a special translation, called the *identity*, that leaves every point unchanged.
- x' = x + 0
- y' = y + 0
- z' = z + 0

Groups and Composition

- Properties of a group
  - ?
  - ?
  - ?
  - ?

Groups and Composition

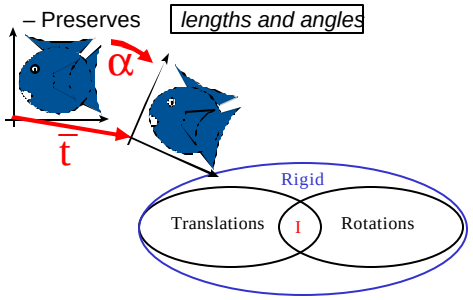
- Properties of a group:
  - There exists an inverse mapping for each function
  - There exists an identity mapping
  - The composition operation is associative
  - The functions are "closed under composition"
- Translations are a group
  - ?
  - ?
  - ?
  - ?

Groups and Composition

- Properties of a group:
  - There exists an inverse mapping for each function
  - There exists an identity mapping
  - The composition operation is associative
  - The functions are "closed under composition"
- These properties might seem trivial at first glance, but they are actually very important, because when these conditions are shown for any class of functions and their two-argument composition operation, then they form an **algebraic group**. One of the consequences is that any series of translations can be composed to a single translation. Another consequence is that the inverse is unique.

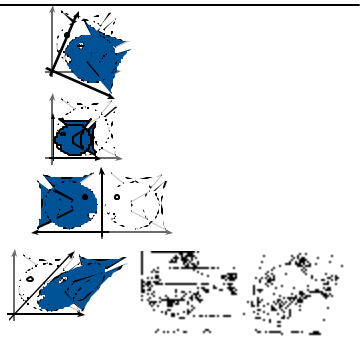
Rigid-Body (Euclidean)

- Translation, rotation (or combination)
  - Preserves *lengths and angles*



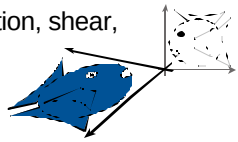
Linear transformations

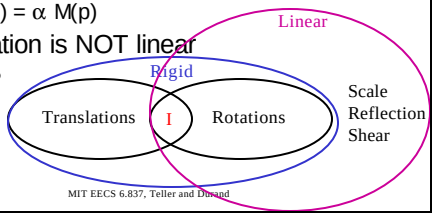
- Rotation,
- Scaling,
- Reflection
- Shear



Linear transformations

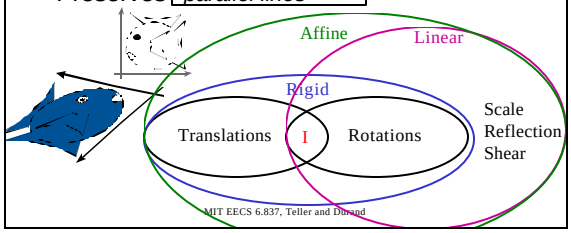
- Rotation, scaling, reflection, shear,
- Any combination
- Linearity:
  - $M(p + q) = M(p) + M(q)$
  - $M(\alpha p) = \alpha M(p)$
- Translation is NOT linear
  - Why?





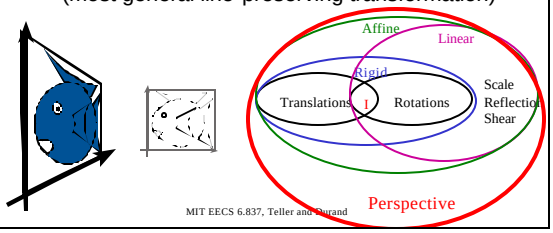
Affine

- Affine: translation, rotation, scaling, reflection, shear (or any combination)
- i.e. linear mappings plus a translation
- Preserves *parallel lines*



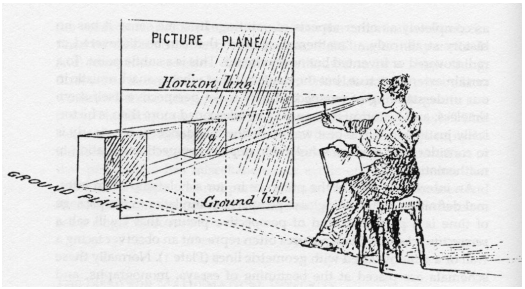
Projective transformations

- Perspective projection
  - Aka homogeneous or rational linear mapping
  - Preserves *lines*
  - (most general line-preserving transformation)



Projective transformations

- Fundamental for projection



Physical 3D perspective

- In Rome
- Walls recede
- Makes it look longer

Side view

Borromini

Questions?

Today's lecture: transformations

- Motivation
- Transformations
- Matrices
- Composition, object vs. world space
- Playtime (if time permits)

Matrix

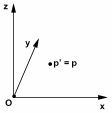
- **Caution**: conventions differ among texts!
  - I'll almost always use right-multiplication (p'= Mp)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{means} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix}$$

Homogeneous Coordinates

- Convenient to describe points as 4-vectors (x; y; z;w)
  - w is "homogeneous coordinate," to be explained later
- For now, we will use w = 1 everywhere
- Trivial example:
  - identity transformation

$$x' = x \qquad y' = y \qquad z' = z$$



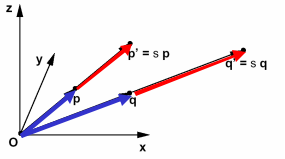
  - In matrix form

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Isotropic Scaling

- Scale all points about origin by factor s:

$$x' = sx$$
$$y' = sy$$
$$z' = sz$$

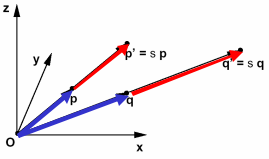


$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Isotropic Scaling

- Scale all points about origin by factor s:

$$x' = sx$$
$$y' = sy$$
$$z' = sz$$

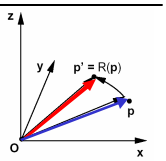


$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotation

- Rotation about +z axis by angle q (note convention)

$$x' = x\cos q - y\sin q$$
$$y' = x\sin q + y\cos q$$
$$z' = z$$



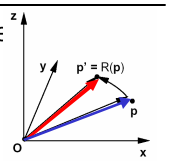
- In matrix form

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotation

- Rotation about +z axis by angle q (note convention)

$$x' = x\cos q - y\sin q$$
$$y' = x\sin q + y\cos q$$
$$z' = z$$



- In matrix form

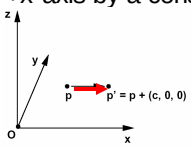
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos q & -\sin q & 0 & 0 \\ \sin q & \cos q & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- Rotation about x, y axes analogous

Translation

- Translation along +x axis by a constant c:

$$x' = x + c$$
$$y' = y$$
$$z' = z$$



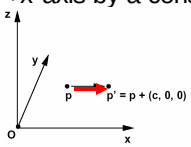
- In Matrix form:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Translation

- Translation along +x axis by a constant c:

$$x' = x + c$$
$$y' = y$$
$$z' = z$$



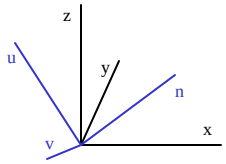
- In Matrix form:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & C \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- **Note: The non-linear operation became linear**

Change of orthobasis (xyz to uvn)

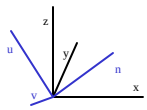
- "Express xyz (i.e., world-space) points in uvn basis."



- We know $p = x\vec{x} + y\vec{y} + z\vec{z} = u\vec{u} + v\vec{v} + n\vec{n}$
- Find transformation M that, given x; y; z; 1, produces u; v; n; 1

Change of orthobasis (xyz to uvn)

$$\begin{aligned}\vec{x} &= (\vec{x}\cdot\vec{u})\vec{u} + (\vec{x}.\vec{v})\vec{v} + (\vec{x}.\vec{n})\vec{n}\\ \vec{y} &= (\vec{y}\cdot\vec{u})\vec{u} + (\vec{y}.\vec{v})\vec{v} + (\vec{y}.\vec{n})\vec{n}\\ \vec{z} &= (\vec{z}\cdot\vec{u})\vec{u} + (\vec{z}.\vec{v})\vec{v} + (\vec{z}.\vec{n})\vec{n}\end{aligned}$$

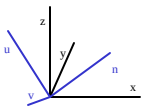


- Rewrite p, then collect terms in u, v, n:

$$\begin{aligned}u &= x(\vec{x}\cdot\vec{u}) + y(\vec{x}.\vec{v}) + z(\vec{x}.\vec{n})\\ v &= x(\vec{y}\cdot\vec{u}) + y(\vec{y}.\vec{v}) + z(\vec{y}.\vec{n})\\ n &= x(\vec{z}\cdot\vec{u})\vec{u} + y(\vec{z}.\vec{v})v + z(\vec{z}.\vec{n})n\end{aligned}$$

Change of orthobasis (xyz to uvn)

- In Matrix form

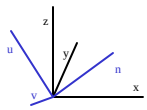


$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_x & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- Verification: $M \begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix} = ?$

Inverse of orthobasis change

$$M = \begin{pmatrix} u_x & u_y & u_x & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



- What is M^{-1} ? (What takes uvn to xyz coords?)
- Simply M^T , since $M^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix}$

Common graphics use

- Transform to a coordinate system using M
- Perform operation
- Transform back using M^{-1}

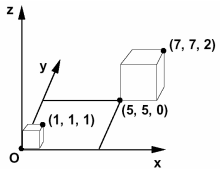
Questions?

Today's lecture: transformations

- Motivation
- Transformations
- Matrices
- Composition, object vs. world space
- Playtime (if time permits)

Composition, Non-Commutativity

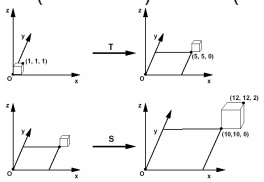
- Given:
  a unit cube at the origin
- Transform to: a side-2 cube at (5, 5, 0) from origin
- We know that
  - T(0,0,0) = (5,5,0) and
  - T(1,1,1) = (7,7,2)
- What do we do?
  - Translate, then scale? Or...
  - Scale, then translate?



Translate, then Scale:

- Compute individual matrices, then compose

$$T_{(5,5,0)} = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad S_2 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

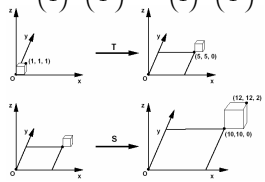


$$ST = \begin{pmatrix} 2 & 0 & 0 & 10 \\ 0 & 2 & 0 & 10 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

What happened?

- S T p = S (T p); S acting on translated p

$$ST\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 0 \\ 1 \end{pmatrix} \quad ST\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \\ 2 \\ 1 \end{pmatrix}$$

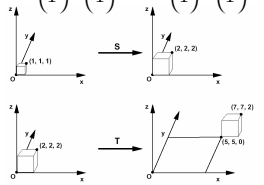


$$ST = \begin{pmatrix} 2 & 0 & 0 & 10 \\ 0 & 2 & 0 & 10 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scale, then Translate

- Apply S first:

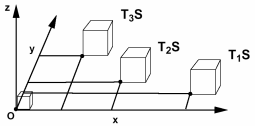
$$TS\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 0 \\ 1 \end{pmatrix} \quad ST\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 7 \\ 2 \\ 1 \end{pmatrix}$$



$$TS = \begin{pmatrix} 2 & 0 & 0 & 5 \\ 0 & 2 & 0 & 5 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

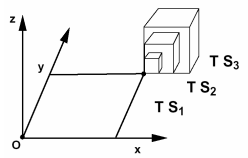
World Space Transformation

- T S p = T (S p); T acting on scaled p
- Think of M as a string of transformations, p' = Mp
- Now apply N to the left of M:
  p'' = NMp = N(Mp)
- We say that N is applied in world space.



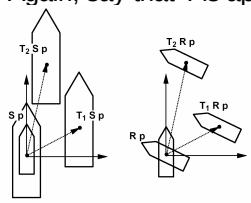
Object-Space transformation

- Apply N to the right of M: p'' = MNp = M(Np)
- We say that N is applied in object space.
- This makes sense
  - it's the first thing to "act upon" p



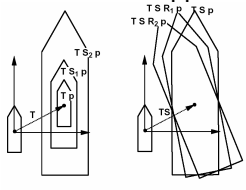
World-Space Translation

- T M p = T (M p); T acting on rotated or scaled p
- Apply T to the left of M: p'' = TMp = T(Mp)
- Again, say that T is applied in "world space."



Object Space Transformation

- Apply M to the right of T: p'' = TMp = T(Mp)
- Again, M is applied in "object space;"
- It's the first thing to "act upon" p
- So it must happen in p's local coordinate



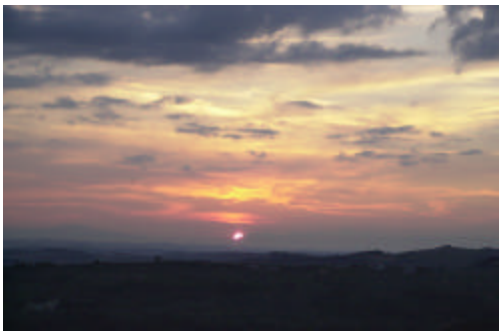
Questions?

Today's lecture: transformations

- Motivation
- Transformations
- Matrices
- Composition, object vs. world space
- Playtime (if time permits)
  - High-level explanation of visual phenomena
  - You can propose the topic

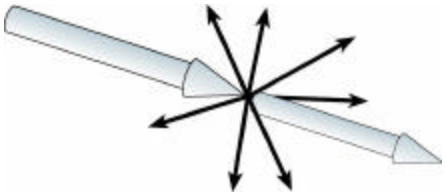
Why is the sky blue?

Answer: because sunset is red



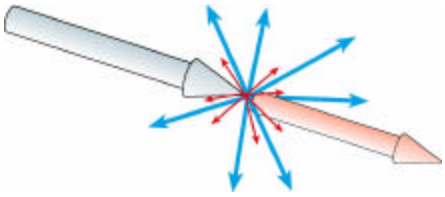
Raileigh scattering

- Lord Raileigh: physicist 19th century
- Light is scattered by air molecules



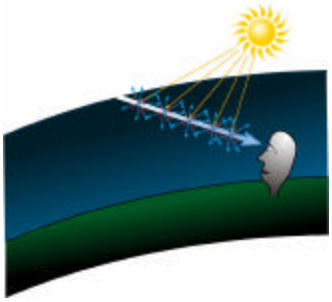
Raileigh scattering

- Lord Raileigh: physicist 19th century
- Light is scattered by air mollecules
- Scattered more in the blue wavelength



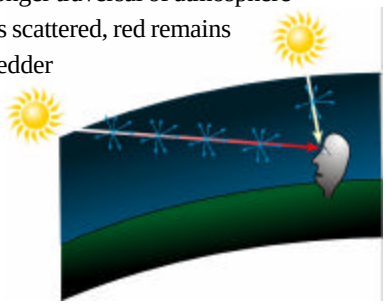
Sky color

- Blue is more scattered



Sun Color

- At sunset, longer traversal of atmosphere
- More blue is scattered, red remains
- Therefore redder

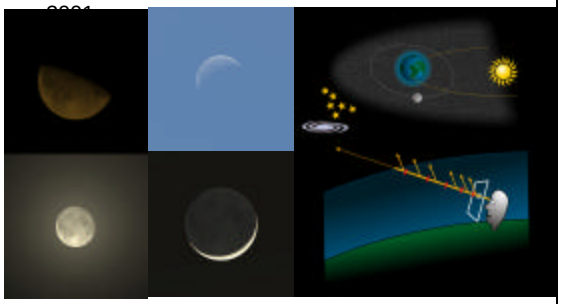


Computer graphics sky models

- E.g. Preetham et al. SIGGRAPH 99.

Night Sky Model

- Jensen, Durand, Stark, Premoze, Dorsey Shirley



Next time

- Transformations and modeling