Review: Separate Warp and Homogenize

- warp requires only standard matrix multiply
  - distort such that orthographic projection of distorted objects shows desired perspective projection
  - w is changed
  - clip after warp, before divide
  - division by w: homogenization

Review: Perspective to NDCS Derivation

- shear
- scale
- projection-normalization

Perspective Example

\[
\begin{bmatrix}
2a & 0 & 0 \\
0 & 2b & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

view volume

- left: \( x = -1 \), right: \( x = 1 \)
- top: \( y = 1 \), bottom: \( y = -1 \)
- near: \( z = 1 \), far: \( z = 4 \)

Perspective Example

tracks in VCS:

- left: \( x = -1 \), right: \( x = 1 \)
- top: \( y = 1 \), bottom: \( y = -1 \)
- near: \( z = 1 \), far: \( z = 4 \)

Perspective Example

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -5/3 & -8/3 \\
0 & 0 & -1 & 0
\end{bmatrix}
\]

Perspective Example

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -5/3 & -8/3 \\
0 & 0 & -1 & 0
\end{bmatrix}
\]

OpenGL Example

Project 1 Grading News

- don't forget to show up 5 min before your slot
- see news item on top of course page for signup sheet scan
- if you have not signed up or need to change your time, contact shailen AT cs.ubc.ca
- you will lose marks if we have to hunt you down!

Review: Perspective Warp/Predistortion

- perspective viewing frustum predistorted to cube
- orthographic rendering of warped objects in cube produces same image as perspective rendering of original frustum

Review: Projective Rendering Pipeline

object \( OCS \)

world \( WC2 \)

viewing \( VCS \)

projection

transformation

modeling

transformations that are applied to object first are specified last

Viewing: Incremental Relative Motion

- how to move relative to current camera coordinate system?
  - what you see in the window
  - computation in coordinate system used to draw previous frame is simple:
    - incremental change \( I \) to current \( C \)
    - at time \( k \), want \( p' = I_k p_k \psi_{k-1} \ldots I_1 p_1 \psi_0 \)
    - each time we just want to premultiply by new matrix
      - \( p' = C p \)
    - but we know that OpenGL only supports postmultiply by new matrix
      - \( p' = C p \)

Viewing: More Camera Motion

- sneaky trick: OpenGL modelview matrix has the info we want!
  - dump out modelview matrix with \( \text{glGetDoublev()} \)
  - \( C \) current camera coordinate matrix
  - wipe the matrix stack with \( \text{glLoadIdentity()} \)
  - apply incremental update matrix \( \psi \)
  - apply current camera model coord \( M \)
  - must leave the modelview matrix unchanged by object transformations after your display call?
    - use push/pop
    - using OpenGL for storage and calculation
      - querying pipeline is expensive
    - but safe to do just once per frame

Project 1 Grading News

- don't forget to show up 5 min before your slot
- see news item on top of course page for signup sheet scan
- if you have not signed up or need to change your time, contact shailen AT cs.ubc.ca
- you will lose marks if we have to hunt you down!

Review: Perspective Warp/Predistortion

- perspective viewing frustum predistorted to cube
- orthographic rendering of warped objects in cube produces same image as perspective rendering of original frustum

Review: Projective Rendering Pipeline

object \( OCS \)

world \( WC2 \)

viewing \( VCS \)

projection

transformation

modeling

transformations that are applied to object first are specified last

Viewing: Incremental Relative Motion

- how to move relative to current camera coordinate system?
  - what you see in the window
  - computation in coordinate system used to draw previous frame is simple:
    - incremental change \( I \) to current \( C \)
    - at time \( k \), want \( p' = I_k p_k \psi_{k-1} \ldots I_1 p_1 \psi_0 \)
    - each time we just want to premultiply by new matrix
      - \( p' = C p \)
    - but we know that OpenGL only supports postmultiply by new matrix
      - \( p' = C p \)

Viewing: More Camera Motion

- sneaky trick: OpenGL modelview matrix has the info we want!
  - dump out modelview matrix with \( \text{glGetDoublev()} \)
  - \( C \) current camera coordinate matrix
  - wipe the matrix stack with \( \text{glLoadIdentity()} \)
  - apply incremental update matrix \( \psi \)
  - apply current camera model coord \( M \)
  - must leave the modelview matrix unchanged by object transformations after your display call?
    - use push/pop
    - using OpenGL for storage and calculation
      - querying pipeline is expensive
    - but safe to do just once per frame
Caution: OpenGL Matrix Storage
• OpenGL internal matrix storage is columnwise, not rowwise.
  - opposite of standard C/C++/Java convention
  - possibly confusing if you look at the matrix from glGetDoublev()

Viewing: Virtual Trackball
• interface for spinning objects around
  - drag mouse to control rotation of view volume
  - orbit/spin metaphor
  - vs. flying/driving
• rolling glass trackball
  - center at screen origin, surrounds world
  - hemisphere "sticks up" in z, out of screen
  - rotate ball = spin world

Virtual Trackball
• know screen click: (x, 0, z)
  - want to infer point on trackball: (x,y,z)
  - ball is unit sphere, so ||x, y, z|| = 1.0
  - solve for y

Trackball Rotation
• correspondence:
  - moving point on plane from (x, 0, z) to (x, 0, c)
  - translating mouse from p1 (mouse down) to p2 (mouse up)
  - correspondences:
    - translating mouse from p1 (mouse down) to p2 (mouse up)
    - rotating about the axis n = p1 x p2

Manual Ray Intersection
• do all computation at application level
  - map selection point to a ray
  - intersect ray with all objects in scene.
• advantages
  - no library dependence
  - difficult to program
  - slow: work to do depends on total number and complexity of objects in scene
• disadvantages
  - hard to program
  - slow: work to do depends on total number and complexity of objects in scene

Bounding Extents
• keep track of axis-aligned bounding rectangles
• advantages
  - conceptually simple
  - easy to keep track of boxes in world space

Backbuffer Example
- use small rectangle around cursor for viewport
- assign per-object integer keys (names)
- redraw in special mode
- store hit list of objects in region
- examine hit list

Select/Hit
- small rectangle around cursor
- change coord sys so fills viewport
- why rectangle instead of point?
  - people aren't great at positioning mouse
  - Fitts' Law: time to acquire a target is function of the distance to and size of the target
  - allow several pixels of slop

Backbuffer Color Coding
- advantages
  - conceptually simple
  - variable precision
- disadvantages
  - introduce 2x redraw delay
  - backbuffer readback very slow

Backbuffer Color Coding
- use backbuffer for picking
  - create image as computational entity
  - never displayed to user
  - save each object's unique color for each pickable object
  - store in table
  - read back pixel at cursor location
  - check against table

Picking
- use small region around cursor for viewport
- assign per-object integer keys (names)
- redraw in special mode
- store hit list of objects in region
- examine hit list

Selection and Feedback Chapter
- all
- Now That You Know Chapter
- only Object Selection Using the Back Buffer

Interactive Object Selection
- move cursor over object, click
  - how to decide what is below?
  - inverse of rendering pipeline flow
  - from pixel back up to object
  - ambiguity
    - many 3D world objects map to same 2D point
- four common approaches
  - manual ray intersection
  - bounding extents
  - backbuffer color coding
  - selection region with hit list

Reading
- Red Book
  - Selection and Feedback Chapter
  - only Object Selection Using the Back Buffer

Bounding Extents
• disadvantages
  - low precision
  - must keep track of object-rectangle relationship
• extensions
  - do more sophisticated bound bookkeeping
    - first level: box check
    - second level: object check
**Viewport**
- nontrivial to compute
  - invert viewport matrix, set up new orthogonal projection
- simple utility command
  - `gluPickMatrix(x,y,w,h,viewport)`
    - `x,y`: cursor point
    - `w,h`: sensitivity/slop (in pixels)
    - push old setup first, so can pop it later

**Render Modes**
- `glRenderMode(mode)`
  - GL_RENDER: normal color buffer
    - default
  - GL_SELECT: selection mode for picking
    - (GL_FEEDBACK: report objects drawn)

**Name Stack**
- again, "names" are just integers
  - `glInitNames()`
- flat list
  - `glLoadIdentity(name)`
- or hierarchy supported by stack
  - `glPushMatrix(name)`, `glPopMatrix`
  - can have multiple names per object

**Viewport**
- `fourth component for transparency`
  - `(r,g,b,α)`
  - fraction we can see through
  - `c = αc_f + (1-α)c_b`
  - more on compositing later

**Hit List**
- `glSelectBuffer(bufferSize, *buffer)`
  - where to store hit list data
  - on hit, copy entire contents of name stack to output buffer.
  - can avoid drawing
  - can have multiple names per object

**Integrated vs. Separate Pick Function**
- integrate: use same function to draw and pick
  - simpler to code
  - name stack commands ignored in render mode
- separate: customize functions for each
  - potentially more efficient
  - can avoid drawing unpickable objects

**Select/Hit**
- advantages
  - faster
  - OpenGL support means hardware acceleration
  - avoid shading overhead
- flexible precision
  - size of region controllable
  - flexible architecture
  - custom code possible, e.g. guaranteed frame rate
- disadvantages
  - more complex

**RGB Color**
- triple `(r, g, b)` represents colors with amount of red, green, and blue
  - hardware-centric
  - used by OpenGL

**Alpha**
- fourth component for transparency
  - `(r,g,b,α)`
  - fraction we can see through
  - `c = αc_f + (1-α)c_b`
  - more on compositing later

**Component Color**
- component-wise multiplication of colors
  - `(a_0,a_1,a_2) * (b_0,b_1,b_2) = (a_0*b_0, a_1*b_1, a_2*b_2)`

**Additive vs. Subtractive Colors**
- additive: light
  - monitors, LCDs
  - `C * M = R`
- RGB model
  - `C * M = R`
- subtractive: pigment
  - printers
  - CMY model
  - dyes absorb light

**Basics Of Color**
- elements of color:
  - light x object = color
  - why does this work?
    - must dive into light, human vision, color spaces

**Hierarchical Names Example**
- `for(int i = 0; i < 2; i++) {
  for(int j = 0; j < 2; j++) {
    glPushName(i);
    for(int k = 0; k < 2; k++) {
      glPushMatrix();
      glPushName(j);
      for(int l = 0; l < 2; l++) {
        glPushMatrix();
        glCallList(i*10.0 + j * 10.0);
      }
      glPopMatrix();
    }
    glPopMatrix();
  }
}

**OpenGL Precision Picking Hints**
- `gluUnproject`
  - transform window coordinates to object coordinates given current projection and modelview matrices
  - use to create ray into scene from cursor location
  - call `gluUnProject` twice with same `(x,y)` mouse location
  - `z = near: (x,y,0)`
  - `z = far: (x,y,1)`
  - subtract near result from far result to get direction vector for ray
  - use this ray for line/polygon intersection

**Reading for Color**
- `RB Chap Color`
- `FCG Sections 3.2-3.3`
- `FCG Chap 20 Color`
- `FCG Chap 21.2.2 Visual Perception (Color)`
Basics of Color
- physics
  - illumination
  - electromagnetic spectra
- reflection
  - material properties
  - surface geometry and microgeometry
- polished versus matte versus brushed
- perception
  - physiology and neurophysiology
  - perceptual psychology

Light Sources
- common light sources differ in kind of spectrum they emit:
  - continuous spectrum
    - energy is emitted at all wavelengths
      - blackbody radiation
    - tungsten light bulbs
    - certain fluorescent lights
    - sunlight
    - electrical arcs
  - line spectrum
    - energy is emitted at certain discrete frequencies

Blackbody Radiation
- black body
  - dark material, so that reflection can be neglected
  - spectrum of emitted light changes with temperature
    - this is the origin of the term "color temperature"
      - e.g. when setting a white point for your monitor
    - cold: mostly infrared
    - hot: reddish
    - very hot: bluish
  - demo:

Electromagnetic Spectrum


Intensity vs. Brightness
- intensity: physical term
  - measured radiance energy emitted per unit time, per unit solid angle, and per unit projected area of the source (related to the luminance of the source)
- lightness/brightness: perceived intensity of light
  - nonlinear

Perceptual vs. Colorimetric Terms
- Perceptual
  - Hue
  - Saturation
  - Lightness
    - reflecting objects
  - Brightness
    - light sources
- Colorimetric
  - Dominant wavelength
  - Excitation purity
  - Luminance

Line Spectrum
- ionized gases
- lasers
- some fluorescent lamps

White Light and Color
- when white light is incident upon an object, some frequencies are reflected and some are absorbed by the object
- combination of frequencies present in the reflected light that determines what we perceive as the color of the object

Sunlight Spectrum
- spectral distribution: power vs. wavelength

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

Sunlight Spectrum
- spectral distribution: power vs. wavelength

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"

Continuous Spectrum
- sunlight
- various "daylight" lamps

Hue
- hue (or simply, "color") is dominant wavelength-frequency
  - integration of energy for all visible wavelengths is proportional to intensity of color

Saturation or Purity of Light
- how washed out or how pure the color of the light appears
  - contribution of dominant light vs. other frequencies producing white light
  - saturation: how far is color from grey
    - pink is less saturated than red
    - sky blue is less saturated than royal blue

White Light
- sun or light bulbs emit all frequencies within visible range to produce what we perceive as "white light"