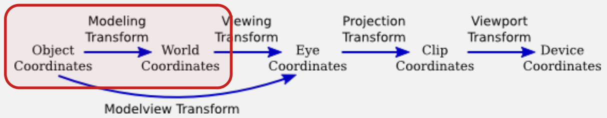
Modeling, Viewing, and Projection

Modeling Transform



translate(0,10)
rotate(-45)
draw house

Isolating Modeling Transforms

```
// body of cart
gl2.glPushMatrix();
gl2.glTranslatef(0f,1f,0f);
gl2.glScalef(6f,2f,1f);
Drawing2D.filledSquare(gl2,1);
gl2.glPopMatrix();

// top of cart
gl2.glPushMatrix();
gl2.glTranslatef(-1f,2.5f,0f);
gl2.glScalef(2.6f,1f,1f);
Drawing2D.filledSquare(gl2,1);
gl2.glPopMatrix();
```

pushMatrix saves the current transformation matrix

popMatrix restores the last-saved transformation matrix

⚠ syntax is OpenGL 1.0, not WebGL

Isolating Modeling Transforms

```
// body of cart
gl2.glPushMatrix();
gl2.glTranslatef(0f,1f,0f);
gl2.glScalef(6f,2f,1f);
Drawing2D.filledSquare(gl2,1);
gl2.glPopMatrix();

// top of cart
gl2.glPushMatrix();
gl2.glTranslatef(-1f,2.5f,0f);
gl2.glScalef(2.6f,1f,1f);
Drawing2D.filledSquare(gl2,1);
gl2.glPopMatrix();
```

applies only to the first square (body of the cart)

applies only to the second square (top of the cart)

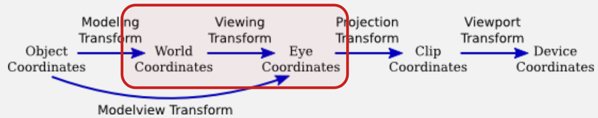
```
// body of cart

gl2.glTranslatef(0f,1f,0f);
gl2.glScalef(6f,2f,1f);
Drawing2D.filledSquare(gl2,1);


// top of cart

gl2.glTranslatef(-1f,2.5f,0f);
gl2.glScalef(2.6f,1f,1f);
Drawing2D.filledSquare(gl2,1);
```

applies to the both squares (body and top of the cart)

applies only to the second square (top of the cart)

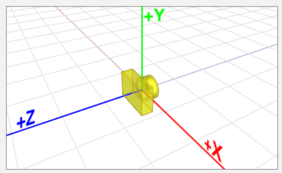
⚠ syntax is OpenGL 1.0, not WebGL

Viewing



eye coordinates (or *view coordinates*) are from the perspective of the viewer

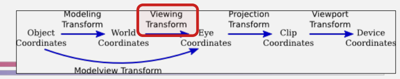
- (0,0,0) at the viewer (viewer's eye)
- viewer looks down negative z
- positive y points up
- positive x points right



viewing transform transforms WC→EC

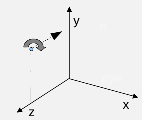
http://www.songho.ca/opengl/gl_camera.html

Viewing Transform



- the *viewing transform* maps WC→EC
  - defined by the position and orientation of the viewer in the world
- manually specifying the viewing transform
  - viewing transform is the inverse of the transform used to position/orient a viewer "object" in the world

```
// position viewer
glTranslatef(0f,2f,2f);
glRotatef(-45,0f,0f,1f);
```




```
// viewing transform
glRotatef(45,0f,0f,1f);
glTranslatef(0f,-2f,-2f);
```

⚠ syntax is OpenGL 1.0, not WebGL

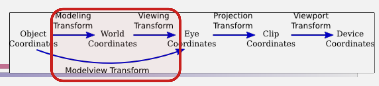
- inverse of a series of transforms is the inverse of each individual transform done in the opposite order

- using the GLU (OpenGL Utility) library

⚠ syntax is OpenGL 1.0, not WebGL

  - `gluLookAt(eyeX,eyeY,eyeZ,refX,refY,refZ,upX,upY,upZ)`
    - viewer at (eyeX,eyeY,eyeZ) - a point
    - viewer looking toward (refX,refY,refZ) - a point
    - up direction is (upX,upY,upZ) - a vector (direction)

Modelview Transform



- modeling transform(s) and viewing transform are distinguished only by their scope
  - a particular modeling transform applies only to a particular object in the scene
  - (the same) viewing transform applies to the whole scene

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

gluLookAt(…);

glPushMatrix();
…
glPopMatrix();
glPushMatrix();
…
glPopMatrix();
…
```

subsequent transforms will affect the modelview matrix

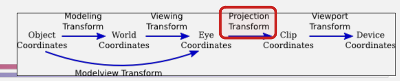
start from a known point

viewing transform

define the scene (modeling transforms and primitives)

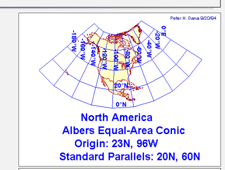
⚠ syntax is OpenGL 1.0, not WebGL

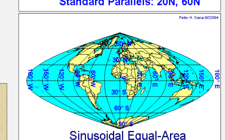
Projection

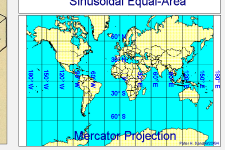


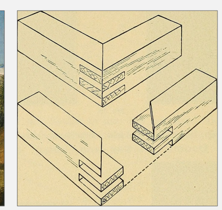
projection transforms higher-dimensional points to lower-dimensional points

(in the viewing pipeline we won't actually drop the z coordinate, but it will subsequently be used only for depth)









https://commons.wikimedia.org/wiki/File:Joos_de_Momper_-_Summer_landscape_with_figures_bringing_in_the_harvest.jpg
https://www.flickr.com/photos/internetarchivebookimages/14596384968
http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj.html

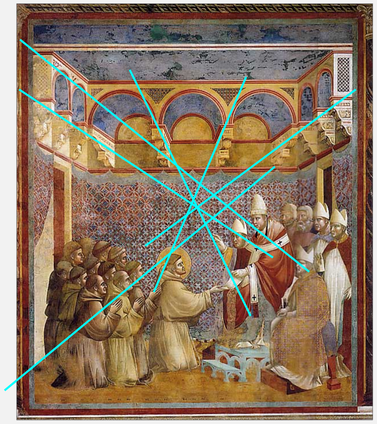
Early Art

size of object not related to distance from viewer

depth shown by overlapping objects or using different horizontal levels

cave paintings at Lascaux (France), c. 15,000 BC

"Heuristic" Perspective

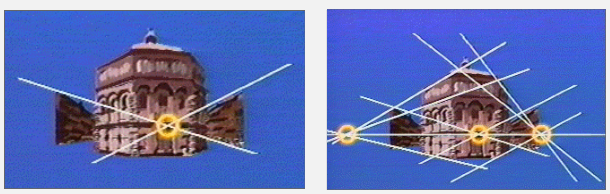


Giotto
Franciscan Rule Approved
c. 1288-1292

e.g.

- incline lines above eye level downward
- incline lines below eye level upward
- incline lines on left or right towards the center

Mathematical Perspective



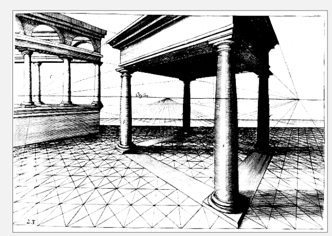
Filippo Brunelleschi, 1420

observations

- lines perpendicular to mirror converged to a central vanishing point
- other oblique lines converged to other vanishing points
- all vanishing points on horizon

Mathematical Perspective

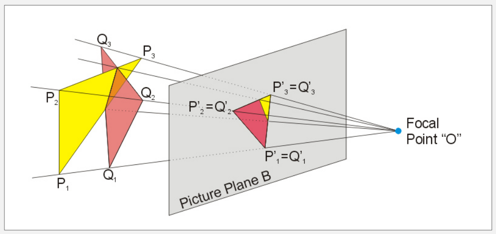
- *principal vanishing points* are derived from the world's primary axes



Vredeman de Vries, *from Perspective*, 1604

Projection in Computer Graphics

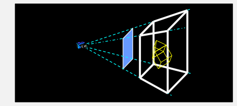
- define a projection by defining a set of projectors
  - every point on a projector ends up at the same point on the projection plane
- *linear projection* – projectors are lines



https://commons.wikimedia.org/wiki/File:Perspective_Projection_Principle.jpg

Perspective Projection

- projectors converge at the eye point



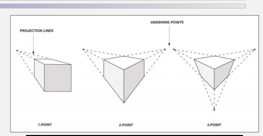
Properties –

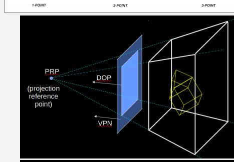
- distant objects appear smaller than near objects
- parallel lines converge

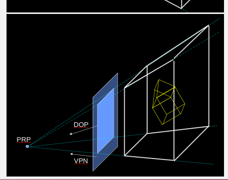
Perspective Projection

Types –

- *one-*, *two-*, *three-point perspective* – the number of principal vanishing points
  - *principal vanishing point* = vanishing point of lines parallel to one of the three coordinate axes
  - *direction of projection* is perpendicular to the projection plane
- *oblique* – direction of projection not perpendicular to the projection plane

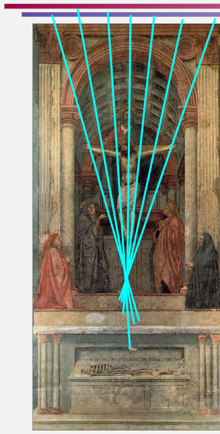




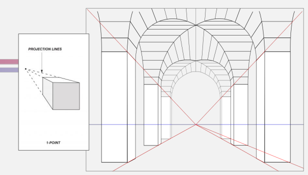


https://blogs.ubc.ca/axonometric/visualglossary/

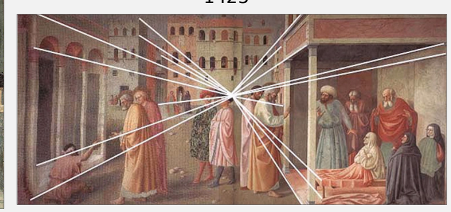
One-Point Perspective



Masaccio
Trinity
1427-28

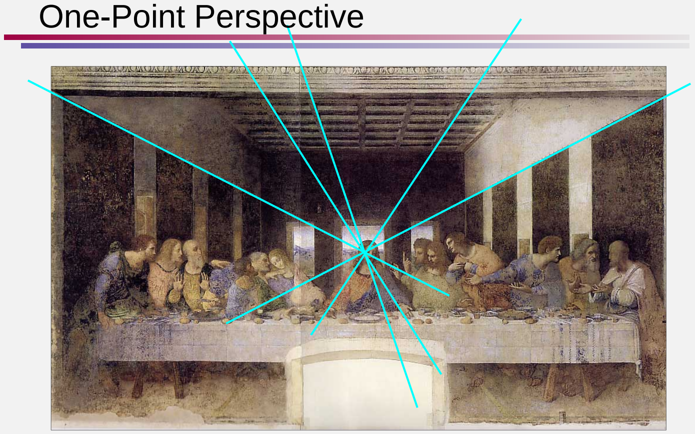


Masolino
The Healing of the Cripple and the Raising of Tabitha
1425



http://www.artchive.com/artchive/M/masaccio/trinity.jpg.html
http://webexhibits.org/sciartperspective/perspective2.html
https://en.wikipedia.org/wiki/3D_projection#/media/File:1ptPerspective.svg
https://blogs.ubc.ca/axonometric/visualglossary/

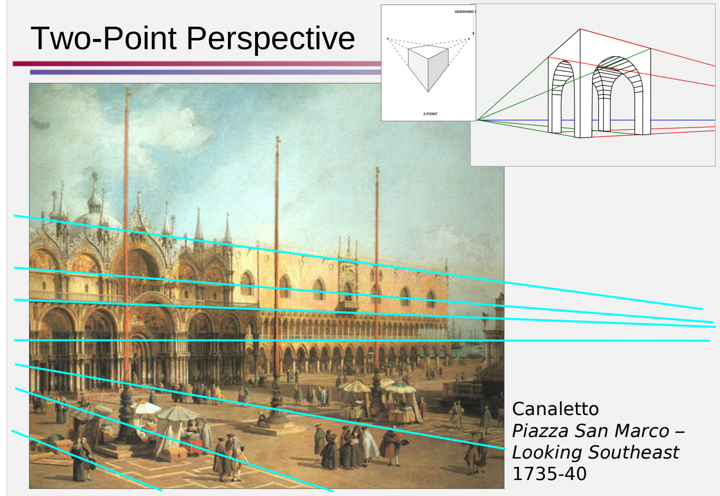
One-Point Perspective



da Vinci, *The Last Supper*, 1498

http://www.artchive.com/artchive/L/leonardo/lastsupp.jpg.html

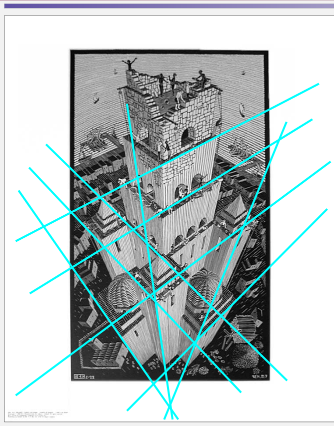
Two-Point Perspective

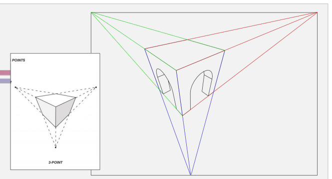


Canaletto
Piazza San Marco – Looking Southeast
1735-40

http://webexhibits.org/sciartperspective/perspective5.html
https://en.wikipedia.org/wiki/3D_projection#/media/File:2-punktperspektive.svg
https://blogs.ubc.ca/axonometric/visualglossary/

Three-Point Perspective

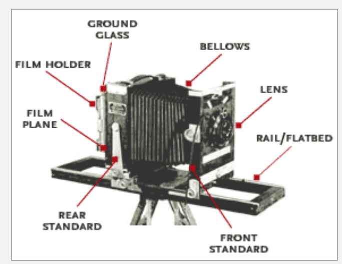




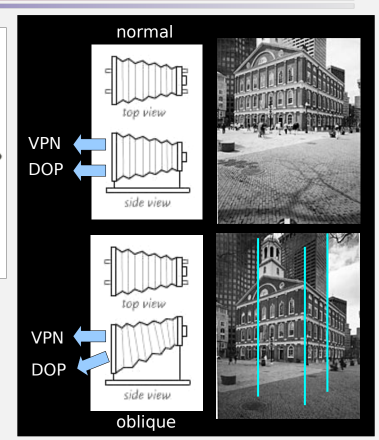
M.C. Escher
Tower of Babel
1928

http://www.worldofescher.com/gallery/A60L.html
https://en.wikipedia.org/wiki/3D_projection#/media/File:3-punktperspektive_1.svg
https://blogs.ubc.ca/axonometric/visualglossary/

Oblique Perspective Projections



view camera

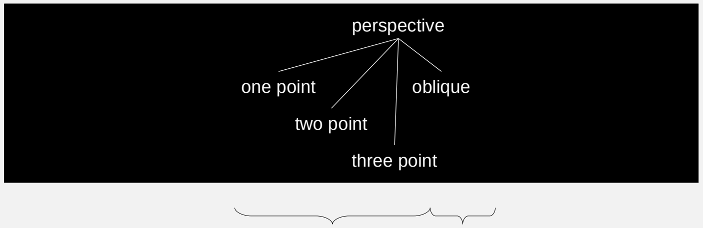


oblique projection results in a taller vertical view, while keeping vertical lines parallel

http://www.pbs.org/wgbh/amex/ansel/sfeature/sf_camera.html

Perspective Projection Recap

projectors converge at eye point



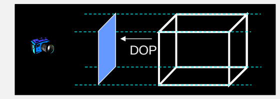
direction of projection is perpendicular to projection plane

distinguished by how many of the coordinate axis intersect the projection plane

direction of projection not perpendicular to projection plane

Parallel Projection

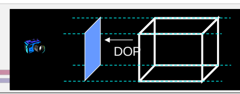
- projectors are parallel lines



Properties –

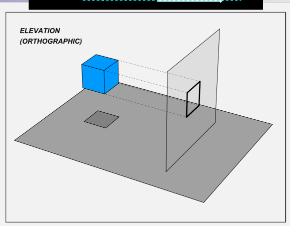
- distant objects appear the same size as near objects
- parallel lines do not converge

Parallel Projection

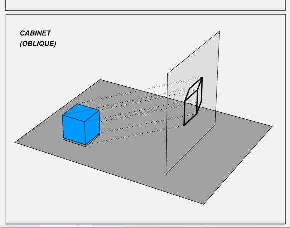


Types –

- *orthographic* – projectors are perpendicular to the projection plane



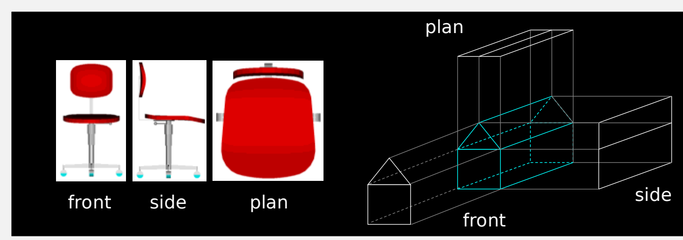
- *oblique* – projectors are not perpendicular to the projection plane



https://commons.wikimedia.org/w/index.php?curid=58480268

Multiview Orthographic Projection

- separate pictures from different sides
  - projection plane is parallel to one of the principal planes defined by the coordinate axes
  - all views use the same scale
- often used for engineering & architectural drawings
- accurate measurements possible
- does not provide realistic view
- need multiple drawings to get 3D feel



Axonometric Projections

- projection plane is not parallel to one of the principal planes defined by the coordinate axes
- *isometric* has single scale factor for all three axes
  - commonly used for catalog illustrations, patent office records, furniture design, structural design
  - illustrates 3D nature without multiple views
  - scale measurements are possible
  - lack of foreshortening creates distorted appearance
  - less useful for curved shapes
- *dimetric* has single scale factor for two axes
- *trimetric* has different scale factors for each axis