Lecture 2 of 41

Viewing 1 of 4: Overview, Projections

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course pages: http://bit.ly/hGvXlH / http://bit.ly/eVizrE
Public mirror web site: http://www.kddresearch.org/Courses/CIS636
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:
Today: Sections 2.2.3 – 2.2.4, 2.8, Eberly 2e – see http://bit.ly/ieUq45
Next class: Section 2.3 (esp. 2.3.4), FVFH slides
Appendices 1-4, Foley, J. D., VanDam, A., Feiner, S. K., & Hughes, J. F. (1991).
Computer Graphics, Principles and Practice, Second Edition in C.

Lecture Outline

- Reading for Last Class: Sections 2.1, 2.2.1 – 2.2.2, Eberly 2e
- Reading for Today: Sections 2.2.3 – 2.2.4, 2.8 Eberly 2e
- Reading for Next Class: Section 2.3 (esp. 2.3.4), Foley *et al.* Slides
- Last Time: Math Foundations, Matrix Transformations
  - Precalculus review: parametric equations of lines
  - Vector spaces and affine spaces
  - Linear systems, linear independence, bases, orthonormality
  - Cumulative Transformation Matrices (CTMs)
    - Translation
    - Rotation
    - Scaling
- Today: Basic Viewing Principles
  - Projections: definitions, history
  - Perspective: optical principles, terminology
- Next Class: Viewing and Normalizing Transformations (VT/NT)

Where We Are

Lecture	Topic	Primary Source(s)
0	Course Overview	Chapter 1, Eberly 2e
1	**CG Basics: Transformation Matrices; Lab 0**	**Sections (§) 2.1, 2.2**
2	Viewing 1: Overview, Projections	§ 2.2.3 – 2.2.4, 2.8
3	Viewing 2: Viewing Transformation	§ 2.3 esp. 2.3.4; FVFH slides
4	**Lab 1a: Flash & OpenGL Basics**	**Ch. 2, 16[1], Angel *Primer***
5	Viewing 3: Graphics Pipeline	§ 2.3 esp. 2.3.7; 2.6, 2.7
6	Scan Conversion 1: Lines, Midpoint Algorithm	§ 2.5.1, 3.1; FVFH slides
7	**Viewing 4: Clipping & Culling; Lab 1b**	**§ 2.3.5, 2.4, 3.1.3**
8	Scan Conversion 2: Polygons, Clipping Intro	§ 2.4, 2.5 esp. 2.5.4, 3.1.6
9	Surface Detail 1: Illumination & Shading	§ 2.5, 2.6.1 – 2.6.2, 4.3.2, 20.2
10	**Lab 2a: Direct3D / DirectX Intro**	**§ 2.7, Direct3D handout**
11	Surface Detail 2: Textures; OpenGL Shading	§ 2.6.3, 20.3 – 20.4, *Primer*
12	Surface Detail 3: Mappings; OpenGL Textures	§ 20.5 – 20.13
13	**Surface Detail 4: Pixel/Vertex Shad.; Lab 2b**	**§ 3.1**
14	Surface Detail 5: Direct3D Shading; OGLSL	§ 3.2 – 3.4, Direct3D handout
15	Demos 1: CGA, Fun; Scene Graphs: State	§ 4.1 – 4.3, **CGA handout**
16	**Lab 3a: Shading & Transparency**	**§ 2.6, 20.1, *Primer***
17	Animation 1: Basics, Keyframes; HW/Exam	§ 5.1 – 5.2
	Exam 1 review; Hour Exam 1 (evening)	**Chapters 1 – 4, 20**
18	**Scene Graphs: Rendering; Lab 3b: Shader**	**§ 4.4 – 4.7**
19	**Demos 2: SFX; Skinning, Morphing**	**§ 5.3 – 5.5, CGA handout**
20	Demos 3: Surfaces; B-reps/Volume Graphics	§ 10.4, 12.7, **Mesh handout**

Lightly-shaded entries denote the due date of a written problem set; heavily-shaded entries, that of a machine problem (programming assignment); blue-shaded entries, that of a paper review; and the green-shaded entry, that of the term project.

Green, **blue** and **red** letters denote **exam review**, **exam**, and **exam solution review** dates.

Acknowledgements

Jim Foley
Professor, College of Computing & Stephen Fleming Chair in Telecommunications
Georgia Institute of Technology

James D. Foley
Georgia Tech
http://bit.ly/ajYf2Q

Andy van Dam
T. J. Watson University Professor of Technology and Education & Professor of Computer Science
Brown University

Andries van Dam
Brown University
http://www.cs.brown.edu/~avd/

Steve Feiner
Professor of Computer Science & Director, Computer Graphics and User Interfaces Laboratory
Columbia University

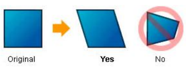
Steven K. Feiner
Columbia University
http://www.cs.columbia.edu/~feiner/

John F. Hughes
Associate Professor of Computer Science
Brown University

John F. Hughes
Brown University
http://www.cs.brown.edu/~jfh/

Background: Basic Linear Algebra for CG

- Reference: Appendix A.1 – A.4, Foley *et al*
- A.1 Vector Spaces and Affine Spaces
  - Equations of lines, planes
  - Vector subspaces and affine subspaces
- A.2 Standard Constructions in Vector Spaces
  - Linear independence and spans
  - Coordinate systems and bases
- A.3 Dot Products and Distances
  - Dot product in $\mathbb{R}^n$
  - Norms in $\mathbb{R}^n$
- A.4 Matrices
  - Binary matrix operations: basic arithmetic
  - Unary matrix operations: transpose and inverse
- Application: Transformations and Change of Coordinate Systems



Affine transformations
© 2005 Trevor McCauley (Senocular)

Review: Basic T, R, S Transformations

- T: Translation (see http://en.wikipedia.org/wiki/Translation_matrix)
  - Given
    - Point to be moved – e.g., vertex of polygon or polyhedron
    - Displacement vector (also represented as point)
  - Return: new, displaced (translated) point of rigid body
- R: Rotation (see http://en.wikipedia.org/wiki/Rotation_matrix)
  - Given
    - Point to be rotated about axis
    - Axis of rotation
    - Degrees to be rotated
  - Return: new, displaced (rotated) point of rigid body
- S: Scaling (see http://en.wikipedia.org/wiki/Scaling_matrix)
  - Given
    - Set of points centered at origin
    - Scaling factor
  - Return: new, displaced (scaled) point
- General: http://en.wikipedia.org/wiki/Transformation_matrix

Review: Lab 0

- Warm-Up Lab
  - Account set-up
  - Linux environment
  - Simple OpenGL exercise
- Basic Account Set-Up
  - See http://support.cis.ksu.edu to understand KSU Department of CIS setup
  - Make sure your CIS department account is set up
  - If not, use SelfServ: https://selfserv.cis.ksu.edu/selfserv/requestAccount
- Linux Environment
  - Make sure your CIS department account is set up
  - Learn how to navigate, set your shell (see KSOL, http://unixhelp.ed.ac.uk)
  - Lab 1 and first homeworks will ask you to render to local XWindows server
- Simple OpenGL exercise
  - Watch OpenGL Primer Part 1 as needed
  - Follow intro tutorials on "NeHe" (http://nehe.gamedev.net) as instructed
  - Turn in: source code, screenshot as instructed in Lab 0 handout

Projections From 3-D to 2-D: Orthographic & Perspective

- History
- Geometrical Constructions
- Types of Projection
- Projection in Computer Graphics

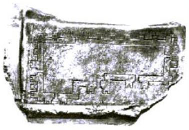
Drawing as Projection

- Painting based on mythical tale as told by Pliny the Elder: Corinthian man traces shadow of departing lover

Detail from **The Invention of Drawing**, 1830: Karl Friedrich Schinkle (Mitchell p.1)

Early Examples of Perspective

- Plan view (orthographic projection) from Mesopotamia, 2150 BC: earliest known technical drawing in existence



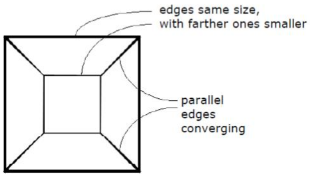
Carlbom Fig. 1-1

Theseus Killing the Minotaur by the Kleophrades Painter

- Greek vases from late 6th century BC show perspective(!)
- Roman architect Vitruvius wrote specifications of plan with architectural illustrations, De Architectura (rediscovered in 1414). The original illustrations for these writings have been lost.

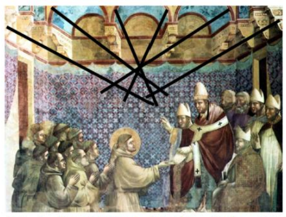
Key Features of Linear Perspective

- || lines converge (in 1, 2, or 3 axes) to *vanishing point*
- Objects farther away are more *foreshortened* (i.e., smaller) than closer ones
- Example: perspective cube



Early Perspective: *Ad Hoc*

- Ways of invoking three dimensional space: shading suggests rounded, volumetric forms; converging lines suggest spatial depth of room
- Not systematic—lines do not converge to single vanishing point



Giotto, **Franciscan Rule Approved**, Assisi, Upper Basilica, c.1295-1300

13

Historical Setting for Invention of Perspective

- The Renaissance: new emphasis on importance of individual viewpoint and world interpretation, power of observation—particularly of nature (astronomy, anatomy, botany, etc.)
  - Massaccio
  - Donatello
  - Leonardo
  - Newton

- Universe as clockwork: intellectual rebuilding of universe along mechanical lines

Ender, **Tycho Brahe and Rudolph II in Prague** (detail of clockwork), c. 1855
url: http://www.mhs.ox.ac.uk/tycho/catfm.htm?image10a

14

Brunelleschi and Vermeer

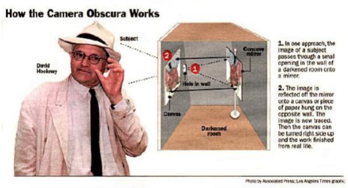
- Brunelleschi invented systematic method of determining perspective projections (early 1400's). Evidence that he created demonstration panels, with specific viewing constraints for complete accuracy of reproduction. Note the perspective is accurate only from one POV (see Last Supper)
- Vermeer created *perspective boxes* where picture, when viewed through viewing hole, had correct perspective
- Vermeer on the web:
  - http://www.grand-illusions.com/articles/mystery_in_the_mirror/
  - http://essentialvermeer.20m.com/
  - http://brightbytes.com/cosite/what.html

Vermeer, **The Music Lesson**, c.1662-1665 (left) and reconstruction (right)

15

Stork vs. Hockney

- An artist named David Hockney proposed that many Renaissance artists, including Vermeer, might have been aided by *camera obscura* while painting their masterpieces, raising a big controversy

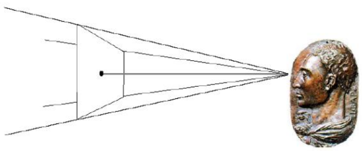


- David Stork, a Stanford optics expert, refuted Hockney's claim in the heated 2001 debate about the subject among artists, museum curators and scientists. He also wrote the article "Optics and Realism in Renaissance Art", using scientific techniques to disprove Hockney's theory

16

Alberti

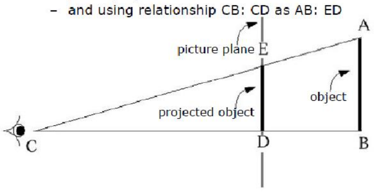
- Published first treatise on perspective, *Della Pittura*, in 1435
- "A painting **[the projection plane]** is the intersection of a visual pyramid **[view volume]** at a given distance, with a fixed center **[center of projection]** and a defined position of light, represented by art with lines and colors on a given surface **[the rendering]**." (Leono Battista Alberti (1404-1472), *On Painting*, pp. 32-33)



17

Visual Pyramid and Similar Triangles [1]

- Projected image is easy to calculate based on
  - height of object (AB)
  - distance from eye to object (CB)
  - distance from eye to picture (projection) plane (CD)
  - and using relationship CB: CD as AB: ED

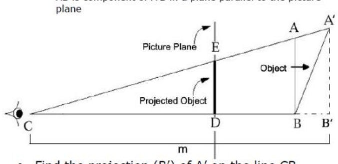


CB : CD as AB : ED

18

Visual Pyramid and Similar Triangles [2]

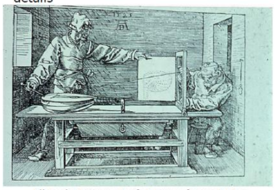
- The general case: the object we're considering is not parallel to the picture plane
  - AB is component of A'B in a plane parallel to the picture plane



- Find the projection (B') of A' on the line CB.
  - Normalize CB
  - dot(CA', normalize(CB)) gives magnitude, m, of projection of CA' in the direction of CB
  - Travel from C in the direction of B for distance m to get B'
- A'B':ED as CB':CD
  - We can use this relationship to calculate the projection of A'B on ED

Dürer

- Concept of similar triangles described both geometrically and mechanically in widely read treatise by Albrecht Dürer (1471-1528)
- Refer to chapter 3 of the book for more details



Albrecht Dürer, **Artist Drawing a Lute**

Woodcut from Dürer's work about the Art of Measurement. 'Underweysung der messung', Nurenberg, 1525

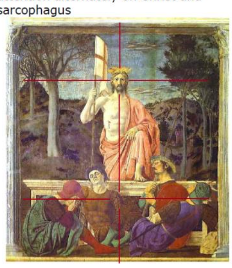
Las Meninas (1656)
By Diego Velàzquez

- Point of view influences content and meaning of what is seen
- Are royal couple in mirror about to enter room? Or is their image a reflection of painting on far left?
- Analysis through computer reconstruction of the painted space
  - verdict: royal couple in mirror is reflection from canvas in foreground, not reflection of actual people (Kemp pp. 105-108)

Robert Campin
The Annunciation Triptych (c. 1425)

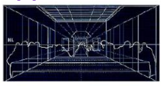
Piero della Francesca
The Resurrection (1460)

- Perspective can be used in unnatural ways to control perception
- Use of two viewpoints concentrates viewer's attention alternately on Christ and sarcophagus

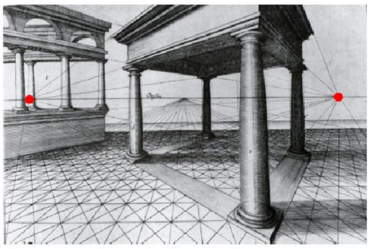


Leonardo da Vinci
The Last Supper (1495)

- Perspective plays very large role in this painting



Geometrical Construction Of Projections

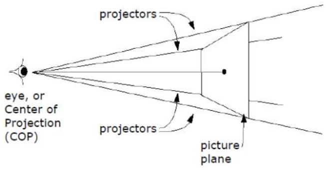


from Vredeman de Vries's **Perspective**, Kemp p.117

- 2 point perspective—two vanishing points

Planar Geometric Projection

- Projectors are straight lines, like the string in Dürer's "Artist Drawing a Lute".
- Projection surface is plane (picture plane, projection plane)

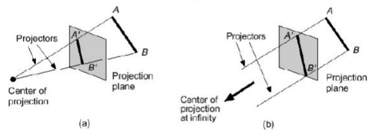


- This drawing itself is perspective projection
- What other types of projections do you know?
  - hint: maps

Planar Geometric Projection

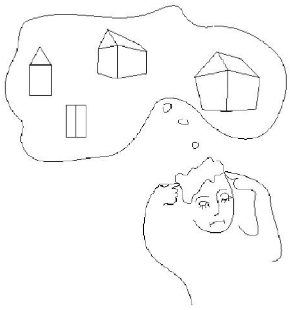
a) Perspective: determined by Center of Projection (COP) (in our diagrams, the "eye")

b) Parallel: determined by Direction of Projection (DOP) (projectors are parallel—do not converge to "eye" or COP). Alternatively, COP is at ∞

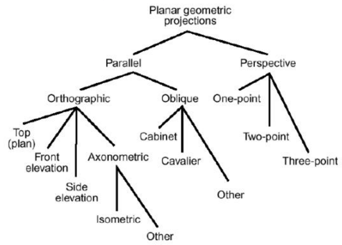


- In general, a projection is determined by where you place the projection plane relative to principal axes of object (relative angle and position), and what angle the projectors make with the projection plane

Types of Projection



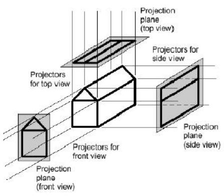
Logical Relationships Among Types of Projections



- Parallel projections used for engineering and architecture because they can be used for measurements
- Perspective imitates eyes or camera and looks more natural

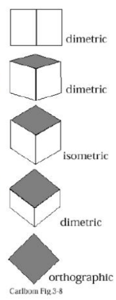
Multiview Orthographic

- Used for:
  - engineering drawings of machines, machine parts
  - working architectural drawings
- Pros:
  - accurate measurement possible
  - all views are at same scale
- Cons:
  - does not provide "realistic" view or sense of 3D form
- Usually need multiple views to get a three-dimensional feeling for object

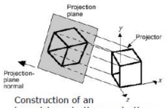


Axonometric Projections

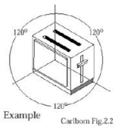
- Same method as multiview orthographic projections, except projection plane not parallel to any of **coordinate planes**; parallel lines equally foreshortened
- *Isometric:* Angles between all three principal axes equal (120°). Same scale ratio applies along each axis
- *Dimetric:* Angles between two of the principal axes equal; need two scale ratios
- *Trimetric:* Angles different between three principal axes; need three scale ratios
- Note: different names for different views, but all part of a continuum of parallel projections of cube; these differ in where projection plane is relative to its cube



Isometric Projection [1]



Construction of an isometric projection: projection plane cuts each principal axis by 45°



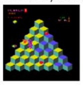
Example

Carlbom Fig.2.2

- Used for:
  - catalogue illustrations
  - patent office records
  - furniture design
  - structural design
  - 3d Modeling in real time (Maya, AutoCad, etc.)
- Pros:
  - don't need multiple views
  - illustrates 3D nature of object
  - measurements can be made to scale along principal axes
- Cons:
  - lack of foreshortening creates distorted appearance
  - more useful for rectangular than curved shapes

Isometric Projection [2]

Video games have been using isometric projection for ages. It all started in 1982 with *Q*Bert* and *Zaxxon* which were made possible by advances in raster graphics hardware



- Still in use today when you want to see things in distance as well as things close up (e.g. strategy, simulation games)

SimCity IV (Trimetric)

StarCraft II

- *Technically* some games today aren't isometric but instead are a general axonometric (trimetric) with arbitrary angles, but people still call them isometric to avoid learning a new word. Other inappropriate terms used for axonometric views are "2.5D" and "three-quarter." http://simcity.ea.com/about/inside_scoop/3d1.php

Oblique Projections

- Projectors at oblique angle to projection plane; view cameras have accordion housing, used for skyscrapers
- Pros:
  - can present exact shape of one face of an object (can take accurate measurements): better for elliptical shapes than axonometric projections, better for "mechanical" viewing
  - lack of perspective foreshortening makes comparison of sizes easier
  - displays some of object's 3D appearance
- Cons:
  - objects can look distorted if careful choice not made about position of projection plane (e.g., circles become ellipses)
  - lack of foreshortening (not realistic looking)

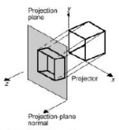
perspective

oblique

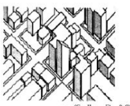
View Camera

Source: http://users.usinternet.com/rniederman/star01.htm

Examples of Oblique Projections

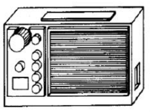


Construction of oblique parallel projection



(Carlbom Fig. 2-3)

Plan oblique projection of city



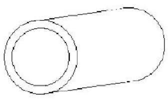
(Carlbom Fig. 2-4)

Front oblique projection of radio

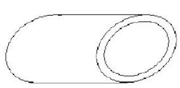
Examples of Oblique Projections

Rules for placing projection plane for oblique views: projection plane should be chosen according to one or several of following:

- parallel to most irregular of principal faces, or to one which contains circular or curved surfaces
- parallel to longest principal face of object
- parallel to face of interest



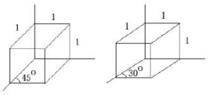
Projection plane parallel to circular face



Projection plane not parallel to circular face

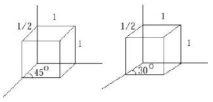
Main Types of Oblique Projections

- *Cavalier:* Angle between projectors and projection plane is 45º. Perpendicular faces projected at full scale



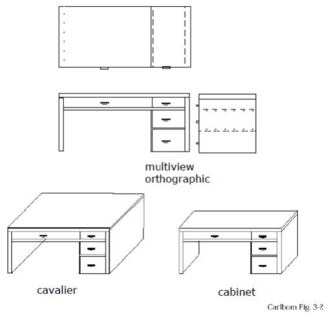
cavalier projection of unit cube

- *Cabinet:* Angle between projectors & projection plane: arctan(2) = 63.4º. Perpendicular faces projected at 50% scale

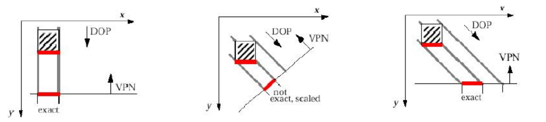


cabinet projection of unit cube

Examples of Orthographic And Oblique Projections



Summary of Parallel Projections



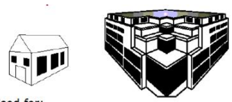
1) Multiview Orthographic
- VPN || a principal coordinate axis
- DOP || VPN
- shows single face, exact measurements

2) Axonometric
- VPN ∦ a principal coordinate axis
- DOP || VPN
- adjacent faces, none exact, uniformly foreshortened (function of angle between face normal and DOP)

3) Oblique
- VPN || a principal coordinate axis
- DOP ∦ VPN
- adjacent faces, one exact, others uniformly foreshortened

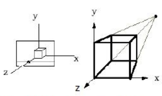
Assume object face of interest lies in principal plane, i.e., parallel to *xy*, *yz*, or *zx* planes. (DOP = Direction of Projection, VPN = View Plane Normal)

Perspective Projections

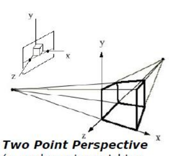


- Used for:
  - advertising
  - presentation drawings for architecture, industrial design, engineering
  - fine art
- Pros:
  - gives a realistic view and feeling for 3D form of object
- Cons:
  - does not preserve shape of object or scale (except where object intersects projection plane)
- Different from a parallel projection because
  - parallel lines not parallel to the projection plane converge
  - size of object is diminished with distance
  - foreshortening is not uniform

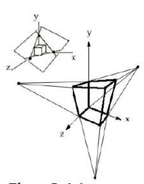
Vanishing Points [1]



One Point Perspective
(*z*-axis vanishing point)



Two Point Perspective
(*z*, and *x*-axis vanishing points)

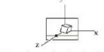


Three Point Perspective
(*z*, *x*, and *y*-axis vanishing points)

For right-angled forms whose face normals are perpendicular to the *x*, *y*, *z* coordinate axes, number of vanishing points = number of principal coordinate axes intersected by projection plane

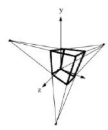
Vanishing Points [2]

- What happens if same form is turned so its face normals are *not* perpendicular to *x*, *y*, *z* coordinate axes?



Unprojected cube depicted here with parallel projection

- New viewing situation: cube is rotated, face normals no longer perpendicular to any principal axes

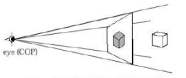


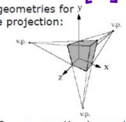
Perspective drawing of the rotated cube

- Although projection plane only intersects one axis (*z*), three vanishing points created
- But… can achieve final results identical to previous situation in which projection plane intersected all three axes
  - Note: the projection plane still intersects all three of the cube's edges, so if you pretend the cube is unrotated, and it's edges the axes, then your projection plane is intersecting the three axes

Vanishing Points and The View Point [1]

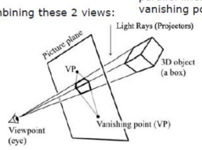
- We've seen two pyramid geometries for understanding perspective projection:



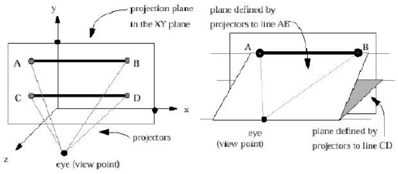


1. perspective image is intersection of a plane with light rays from object to eye (COP)
2. perspective image is result of foreshortening due to convergence of some parallel lines toward vanishing points

- Combining these 2 views:

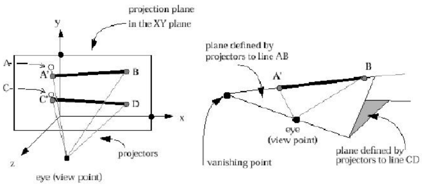


Vanishing Points and The View Point [2]



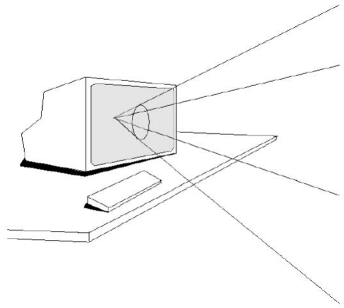
- Project parallel lines *AB*, *CD* on *xy* plane
- Projectors from eye to *AB* and *CD* define two planes, which meet in a line which contains the view point, or eye
- This line **does not** intersect projection plane (*XY*), because parallel to it. Therefore there is no vanishing point

Vanishing Points and The View Point [3]



- Lines *AB* and *CD* (this time with *A* and *C* behind the projection plane) projected on *xy* plane: *A'B* and *C'D*
- Note: *A'B* not parallel to *C'D*
- Projectors from eye to *A'B* and *C'D* define two planes which meet in a line which contains the view point
- This line **does** intersect projection plane
- Point of intersection is vanishing point

Next Time: Projection in Computer Graphics



Summary

- **A Brief History of Viewing**
  - **Ancient and classical views of projections: orthographic, pseudo-perspective**
  - **Perspective and the Renaissance**
  - **The Enlightenment and optics**
- **Taxonomy of Projections**
  - **Multiview orthographic**
  - **Parallel**
    - **Orthographic: top, front, side; axonometric (iso- di-, tri-metric)**
    - **Oblique: cabinet, cavalier**
  - **Perspective: one-, two-, three-point**
- **Projections and Viewing**
  - **Projectors**
  - **Vanishing points**
  - **Center of projection (COP = eye/camera)**
  - **Direction of projection (DOP) *vs.* view plane normal (VPN)**
- **Next: View Volumes, Viewing and Normalizing Transformations**

Terminology

- **Points and Vectors**
  - **Center of projection (COP = eye/camera)**
  - **Direction of projection (DOP) *vs.* view plane normal (VPN)**
- **Kinds of Projections**
  - **Parallel – no foreshortening, projectors stay parallel**
  - **Perspective – foreshortening, projectors converge on vanishing point(s)**
- **Parallel Projections**
  - **Orthographic: "dead on", *i.e.,* (DOP || VPN)**
  - **Oblique: "at an angle" , *i.e.,* ¬ (DOP || VPN)**
- **Orthographic Projections**
  - **Multiview: top, front, side**
  - **Isometric (one measure): 120° angles among each pair of axes**
  - **Other axonometric: dimetric (two different angles), trimetric (three different)**
- **Perspective Projection**
  - **Projectors – lines running parallel to DOP (enclosing view volume)**
  - **Vanishing point(s) – intersection(s) of COP baseline & projection plane**