


Rendering Scenarios

- אצווה (batch)
 - כל תמונה מיוצרת ברמת פירוט גבוהה ככל האפשר עבור סט ספציפי של פרמטרים
 - לוקח כמה זמן שאריך
 - שימושי לפנטוראליזם, סרטים וכו'
- אינטראקטיבי
 - מייצרים תמונות בשבריר שניה (לפחות 10 בשניה) כאשר המשתמש שולט בפרמטרים של הרינדור
 - יש צורך להשיג את האיכות הגבוהה ביותר בתחשב בזמן הנתון (הקצב הנדרש)
 - שימושי לויזואליזציות, משחקים וכו'



קורס גרפיקה ממוחשבת

2009 סמסטר ב'
ליאור שפירא



3D Polygon Rendering Pipeline



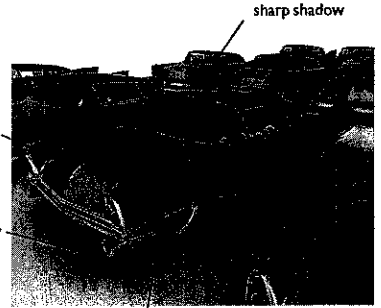
1 חלק מהשקפים מעובדים משקפים של פרוז דוראדו, טומס פוקהאוסר ודיאל כהן-אור

הגורים 25-30 פיימיג Real time 88 כולל כולל עבודה עם רנדור

Why do we need anything else?

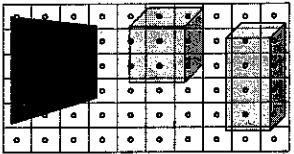
- Ray tracing isn't interactive (yet)
- Need the whole scene in memory
- Hard to achieve
 - Motion blur 
 - Depth of field 
 - הכי יפה תראה... עסקים

What's so good about ray tracing?




Ray Casting Revisited

- For each sample ...
 - Construct ray from eye position through view plane
 - Find first surface intersected by ray through pixel
 - Compute color of sample based on surface radiance



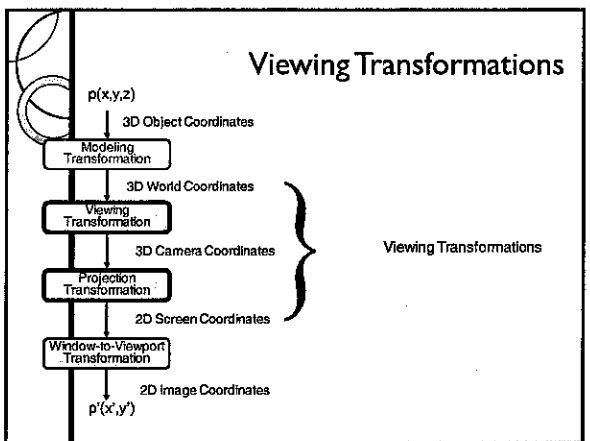
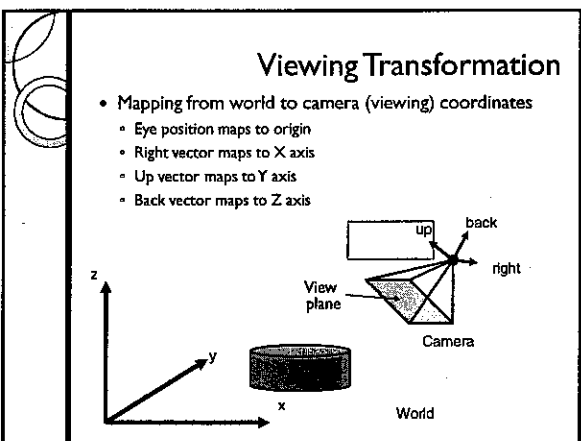
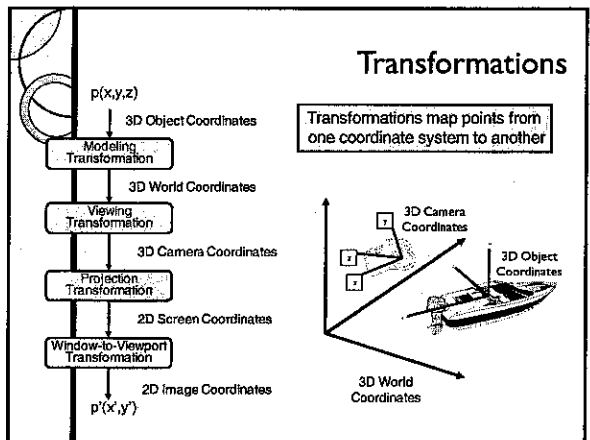
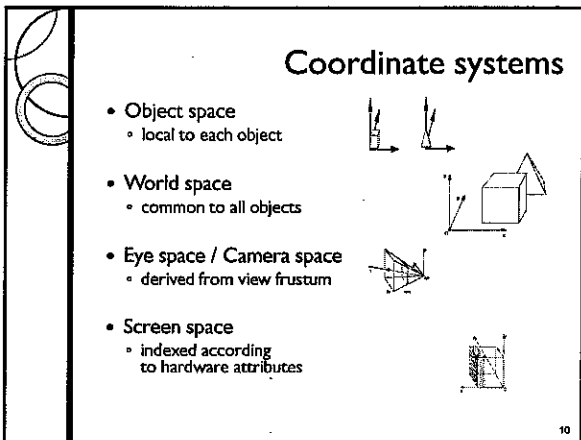
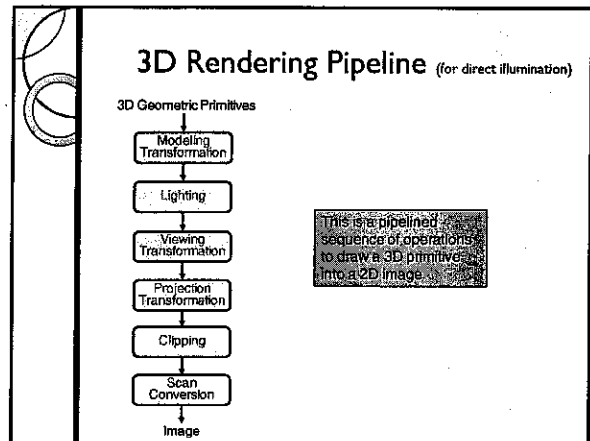
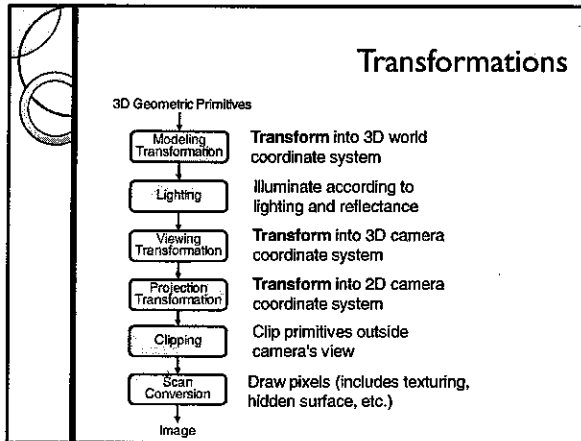
More efficient algorithms utilize spatial coherence!

RASTERIZATION BASED RENDERING



ברוי סרטיים 88 התחשני בוצע סיכור 6 הפיזיקליים לבסוף (אם איזי סים פיימיג אפני)

נקודת תחילת אונרלי בו ניקח סקנה הדיסקט הפיזיקליים שאני עולה מסוימת בליים נוא אפיק אחר עתמינג תחסיסר 6 ביקאים



Finding viewing transformation

- We have the camera (world coordinates)
- We want a transformation T taking objects from world to camera

$$p^c = Tp^w$$

- Trick: find T^{-1} taking objects in camera to world

$$p^w = T^{-1}p^c$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} ?$$

Camera Coordinates

- Canonical coordinate system
 - Convention is right-handed (looking down -z axis)
 - Convenient for projection, clipping, etc.

אנחנו מבינים/מחפשים את T . את T^{-1} קל למצוא כי אנחנו יודעים את הצירים של הצילום. הבעיה היא שהצירים של הצילום הם ימניים (right-handed) והצירים של העולם הם שמאליים (left-handed). לכן אנחנו צריכים להפוך את הצירים של העולם כדי שיהיו ימניים.

Finding viewing transformation

- Lets give it a try...

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} R_x & U_x & B_x & E_x \\ R_y & U_y & B_y & E_y \\ R_z & U_z & B_z & E_z \\ R_w & U_w & B_w & E_w \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

$$\begin{bmatrix} E_x \\ E_y \\ E_z \\ E_w \end{bmatrix} = \begin{bmatrix} R_x & U_x & B_x & E_x \\ R_y & U_y & B_y & E_y \\ R_z & U_z & B_z & E_z \\ R_w & U_w & B_w & E_w \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} R_x + E_x \\ R_y + E_y \\ R_z + E_z \\ R_w + E_w \end{bmatrix} = \begin{bmatrix} R_x & U_x & B_x & E_x \\ R_y & U_y & B_y & E_y \\ R_z & U_z & B_z & E_z \\ R_w & U_w & B_w & E_w \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

עבור לנק' העין, הצירים עוברים לקוטרים המכונים את מערכת הצירים של המצלמה (0,0,0)

Finding viewing transformation

- Trick: map from camera coordinates to world
 - Origin \rightarrow eye position
 - Z axis \rightarrow back vector
 - Y axis \rightarrow up vector
 - X axis \rightarrow right vector

Invert to get T, Easy!

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} R_x & U_x & B_x & E_x \\ R_y & U_y & B_y & E_y \\ R_z & U_z & B_z & E_z \\ R_w & U_w & B_w & E_w \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Projection

- General definition:
 - Transform points in n -space to m -space ($m < n$)
- In computer graphics:
 - Map 3D camera coordinates to 2D screen coordinates
 - Projection is formed by the intersection of certain lines with the view plane

Viewing Transformations

Parallel Projection

- Center of projection is at infinity
- Direction of projection (DOP) same for all points

View Plane

Angel Figure 5.4

Taxonomy of Projections

Planar geometric projections

Parallel

Perspective

Orthographic

Oblique

One-point

Two-point

Three-point

Top (plan)

Front elevation

Side elevation

Axonometric

Isometric

Cabinet

Cavalier

Other

Other

FVFHP Figure 6.10

סוגי הפרויקציות השונים של
 כפרויקציות Parallel
 וזרימה וזרימה וקווים מקבילים

Orthographic Projections

- Since the viewing plane is aligned with (x,y) , orthographic projection is performed by:

$$\begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

22

Orthographic Projections

- DOP perpendicular to view plane

Top

Front

Side

Angel Figure 5.5

לצד של הפרויקציות המקבילים וזרימה
 של זרימה וזרימה הם אופייניים משום שהם
 קווי הפרויקציה הם מקבילים וזרימה
 וזרימה וזרימה וזרימה וזרימה וזרימה

Oblique Projections

- DOP not perpendicular to view plane

Cavalier (DOP at 45°)

Cabinet (DOP at 63.4°)

View plane

DOP

H&B Figure 12.24

Orthographic Projections

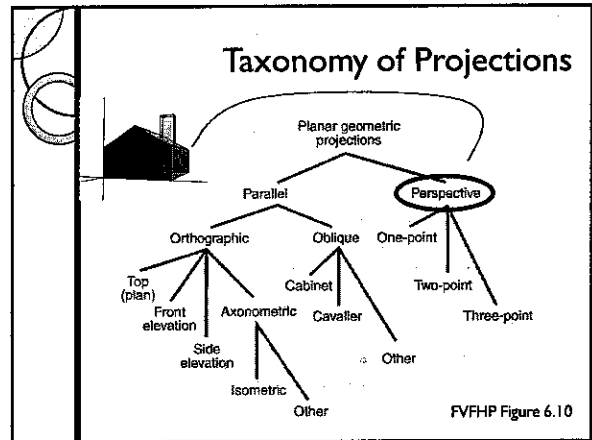
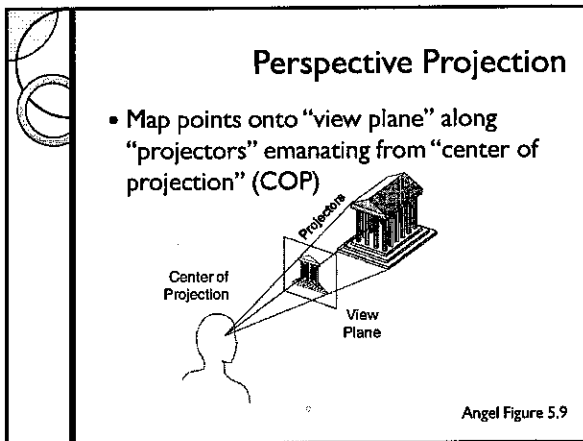
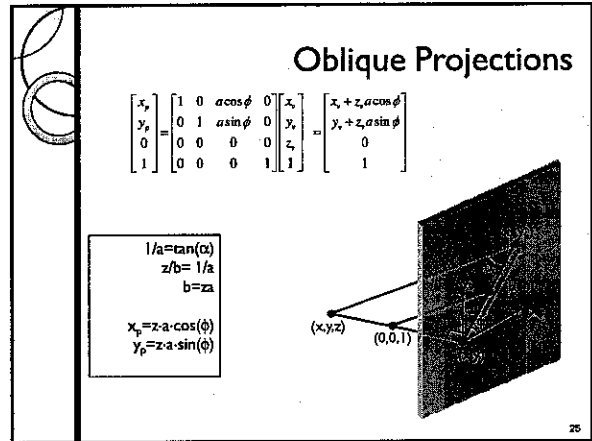
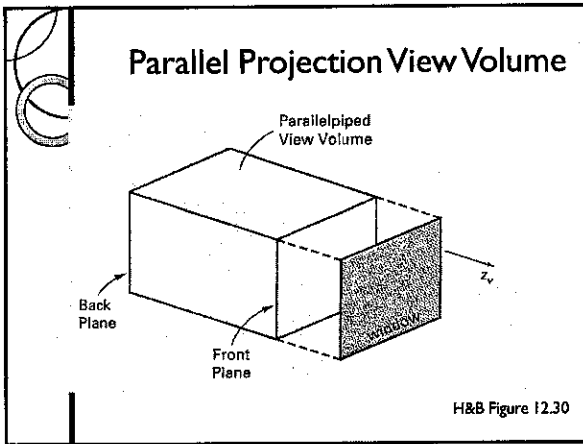
- Lengths and angles of faces parallel to the viewing planes are preserved.
- Problem:** 3D nature of projected objects is difficult to deduce.

Top View

23

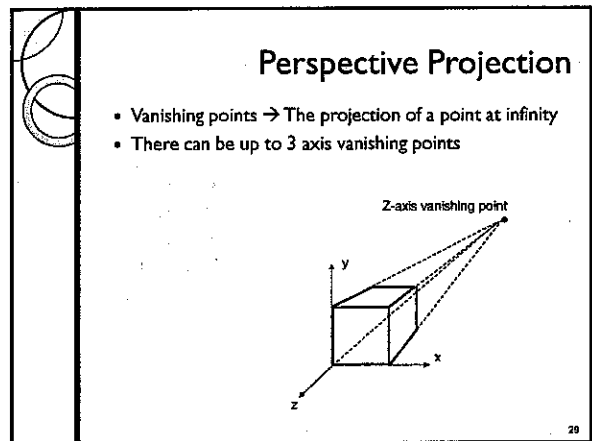
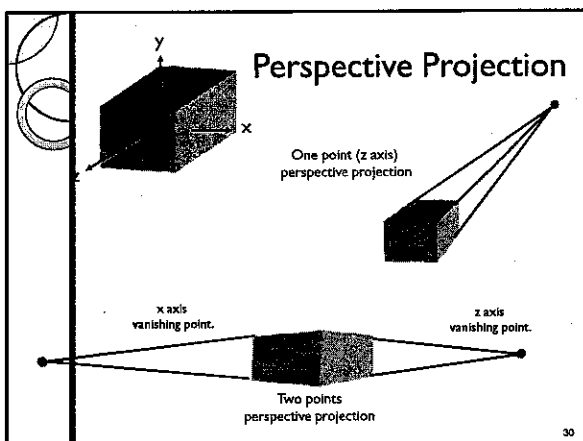
בצורה של הפרויקציה המקבילים וזרימה
 של זרימה וזרימה הם אופייניים משום שהם
 קווי הפרויקציה הם מקבילים וזרימה

הכיתה של זרימה וזרימה של זרימה וזרימה




קטעים של ווריום של הקוים אל תפ"ן
אל תדמה הקוים.

הם תפ"ן - הם'ה פרוסקטיבה הן ואלה
הם - ואלה תפ"ן



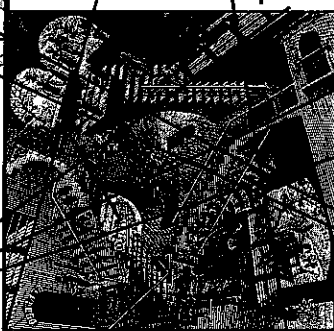
Perspective Projection

- How many vanishing points?



Angel Figure 5.10

3 point perspective

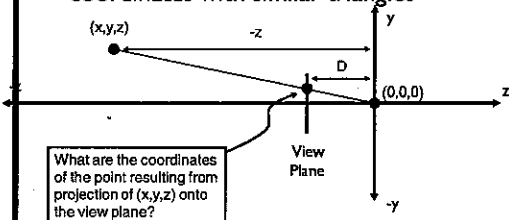


M.C. Escher's "Relativity" where 3 worlds co-exist thanks to 3-point perspective.

31

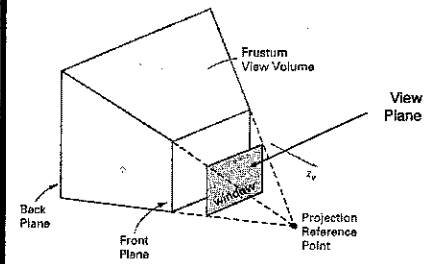
Perspective Projection

- Compute 2D coordinates from 3D coordinates with similar triangles



What are the coordinates of the point resulting from projection of (x,y,z) onto the view plane?

Perspective Projection View Volume



H&B Figure 12.30

יש מנקודת מבט אחת הנקראת נקודת המבט
 כל הנקודות שנמצאות על אותו המישור
 יפגשו בנקודה אחת זו הנקודה הנקראת
 נקודת המבט (נקודת המפגש)

Perspective Projection Matrix

- 4x4 matrix representation?

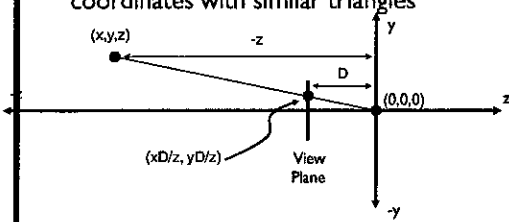
$$\begin{matrix} x_s = x_c D / z_c & x' = x_c \\ y_s = y_c D / z_c & y' = y_c \\ z_s = D & z' = z_c \\ w_s = 1 & w' = z_c / D \end{matrix}$$

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ w_s \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/D & 0 \end{pmatrix}$$

Perspective Projection

- Compute 2D coordinates from 3D coordinates with similar triangles



Changing Perspective

What is the difference between moving the center of projection and moving the projection plane?

Original

Center of Projection Projection plane z

38

Perspective Projections

- Some observations
 1. The matrix representation is singular, therefore a many to one mapping
 2. Points on the viewing plane do not change
 3. When d goes to infinity, we get an orthogonal projection

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	0

37

המטריצה שניצרה אינה הפיכה. זה הפירוט כיוון שאם שהתבונן אליהם כל המושגים, איבדנו מידע.

Changing Perspective

What is the difference between moving the center of projection and moving the projection plane?

Moving the Projection Plane

Center of Projection Projection plane z

40

Changing Perspective

What is the difference between moving the center of projection and moving the projection plane?

Moving the Center of Projection

Center of Projection Projection plane z

39

Perspective vs. Parallel

- Perspective projection
 - + Size varies inversely with distance - looks realistic
 - Distance and angles are not (in general) preserved
 - Parallel lines do not (in general) remain parallel
- Parallel projection
 - + Good for exact measurements
 - + Parallel lines remain parallel
 - Angles are not (in general) preserved
 - Less realistic looking

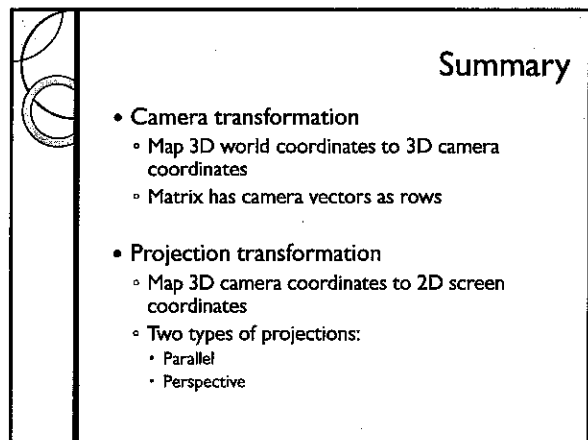
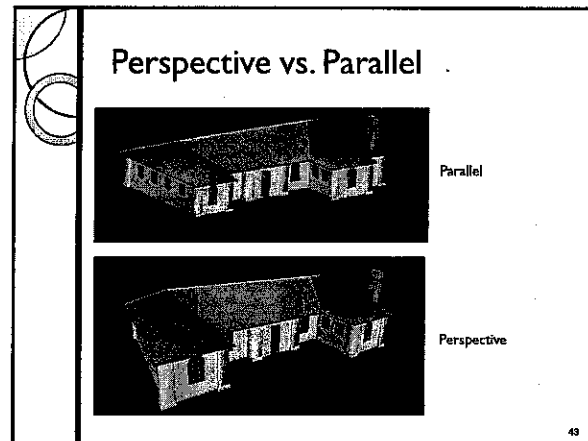
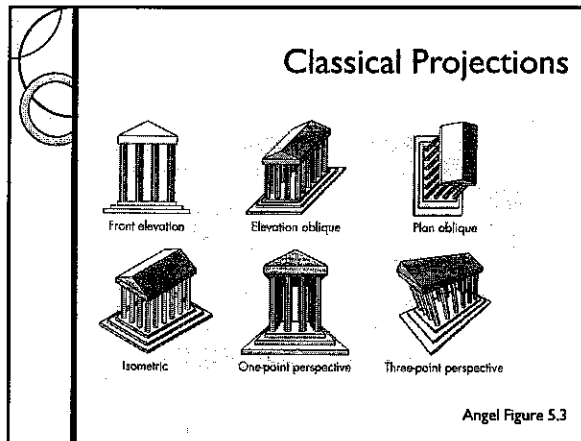
10

Taxonomy of Projections

```

    Planar geometric projections
    /      \
    Parallel  Perspective
    /  \    /  \
    Orthographic  Oblique  One-point
    /  |  \      /  \      /  \
    Top Front Axonometric  Cabinet  Two-point
    (plan) elevation  /  \      /  \
                    Side Cavalier  Three-point
                    elevation  /  \
                    /  \      /  \
                    Isometric  Other
            
```

10



קורס גרפיקה ממוחשבת
 2009 סמסטר ב'
 ליאור שפירא

שיעור 6 חלק שני

3D Polygon Rendering Pipeline

3D Rendering Pipeline (for direct illumination)

2D Rendering Pipeline

Clipping

- Avoid drawing parts of primitives outside window
- Window defines part of scene being viewed
- Must draw geometric primitives only inside window

Clipping

- Avoid drawing parts of primitives outside window
- Points
- Lines
- Polygons
- Circles
- etc.

Point Clipping

- Is point (x,y) inside the clip window?

```

inside =
(x >= wx1) &&
(x <= wx2) &&
(y >= wy1) &&
(y <= wy2);
    
```

Line Clipping

- Find the part of a line inside the clip window

Before Clipping

Line Clipping

- Find the part of a line inside the clip window

After Clipping

Cohen Sutherland Line Clipping

- Use simple tests to classify easy cases first

Cohen Sutherland Line Clipping

- Classify some lines quickly by AND of bit codes representing regions of two endpoints (must be 0)

1001	P0001	0101	
			Bit 4
P1 1000	P3 0000	P4 0100	
P2			Bit 3
1010	P5 0010	P9 0110	
			Bit 2
			Bit 1

אנו מחלקים את המישור ל-16 ריבועים על ידי שתי קווים אופקיים ושתי קווים אנכיים. כל ריבוע מיוחס קוד ביטוני. קוד זה מיוחס לפי המיקום של הריבוע ביחס לקווים. קוד זה מיוחס לפי המיקום של הריבוע ביחס לקווים. קוד זה מיוחס לפי המיקום של הריבוע ביחס לקווים.

Clipping

- Avoid drawing parts of primitives outside window
 - Points
 - Lines
 - Polygons
 - Circles
 - etc.

2D Screen Coordinates

Polygon Clipping

- Find the part of a polygon inside the clip window?

Before Clipping

Polygon Clipping

- Find the part of a polygon inside the clip window?

After Clipping

Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time

Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time

Clipping to a Boundary

- Do inside test for each point in sequence, Insert new points when cross window boundary, Remove points outside window boundary

הנקודה P' היא הנקודה שבה הקו חוצה את הגבול החיצוני של הפוליגון, והנקודה P'' היא הנקודה שבה הקו חוצה את הגבול הפנימי של הפוליגון.

Clipping to a Boundary

- Do inside test for each point in sequence, Insert new points when cross window boundary, Remove points outside window boundary

2D Rendering Pipeline

```

    graph TD
      A[3D Primitives] --> B[2D Primitives]
      B --> C[Clipping]
      C --> D[Viewport Transformation]
      D --> E[Scan Conversion]
      E --> F[Image]
  
```

- Clipping**: Clip portions of geometric primitives residing outside the window
- Viewport Transformation**: Transform the clipped primitives from screen to image coordinates
- Scan Conversion**: Fill pixels representing primitives in screen coordinates

Viewport Transformation

- Transform 2D geometric primitives from screen coordinate system (normalized device coordinates) to image coordinate system (pixels)

Screen

Window

Image

Viewport

Viewport Transformation

- Window-to-viewport mapping

Window

Screen Coordinates

Viewport

Image Coordinates

$$vx = vx1 + (vx - wx1) * (vx2 - vx1) / (wx2 - wx1);$$

$$vy = vy1 + (vy - wy1) * (vy2 - vy1) / (wy2 - wy1);$$

Summary of Transformations

$p(x,y,z)$

3D Object Coordinates

Modeling Transformation

3D World Coordinates

Viewing Transformation

3D Camera Coordinates

Projection Transformation

2D Screen Coordinates

Window-to-Viewport Transformation

2D Image Coordinates

$p'(x',y')$

} Modeling transformation

} Viewing transformations

} Viewport transformation

Summary

3D Primitives

↓

2D Primitives

Clipping

↓

Viewport Transformation

↓

Scan Conversion

↓

Image

Clip portions of geometric primitives residing outside the window

Transform the clipped primitives from screen to image coordinates

Fill pixels representing primitives in screen coordinates

Summary

3D Primitives

↓

3D Modeling Coordinates

Modeling Transformation

3D World Coordinates

Lighting

3D World Coordinates

Viewing Transformation

3D Camera Coordinates

Projection Transformation

2D Screen Coordinates

Clipping

2D Screen Coordinates

Viewport Transformation

2D Image Coordinates

Scan Conversion

2D Image Coordinates

Image

Viewing Window