COMPUTER GRAPHICS AND SCIENTIFIC VISUALIZATION

Atul Singh
Computer Science and Engineering
IIT Kanpur
Mentor: Dr. Suman Chakraborty
OVERVIEW

- Motivation
- Fundamentals of raster graphics
- Transformations
- Projections
- Viewing and Clipping
- Hidden surface elimination
- Color Models and Image Processing
- Animations
- Hardware Aspects
COMPUTER GRAPHICS: NEEDS

- **Modeling**
  - Creating and representing the geometry of objects in 3D world by performing a sequence of transformations on basic geometric shapes

- **Rendering**
  - Generating 2D images from the 3D models and displaying it on visual displays

- **Animation**
  - Describing motion of objects using a sequence of rendered images
Visualization : needs

- The size of the dataset generated by computers is usually order of millions of data values.
- Need to analyze these huge datasets to determine trends and relationships.
- Usually after conversion to visual form, the trends and patterns are immediately apparent.
- Some graphical visualization techniques usually used are color coding, contour plots, graphs and charts.
Visualization : needs (Contd.)

- Huge Datasets
- Critical Insight
- Graphical Visualization
- Meaning transmitted

Color Coded Data

Computer Graphics and Scientific Visualization
FUNDAMENTALS OF RASTER GRAPHICS

- Display screen is imagined to be split into a very fine mesh of elementary rectangles called pixels
- Graphic is displayed by controlling the intensity of the electron beam which is traversing throughout the screen
- Two ways of beam traversal
  - Raster Scan Display
  - Random Scan Display
- Scan conversion algorithms
RASTER SCAN DISPLAY

- Based on television technology
- Electron beam is swept across the screen one row at a time from top to bottom
- As the beam sweeps across the screen the beam intensity is turned on/off to create pattern of illuminated spots
- Picture definition is stored in a memory area called the refresh buffer or frame buffer
- Stored intensity values are retrieved and then painted on the screen
- Constant refresh rate
Random Scan Display

- Draw one line at a time and thus also referred to as vector displays
- Component lines of a picture can be drawn in any order
- Picture definition is stored as a set of line-drawing commands in the *refresh display file*
- Refresh rate dependent on the number of lines to be displayed
- Cannot display realistic shaded scenes
- However produce smoother line drawings (raster scan produces jagged lines)
SCAN CONVERSION

- Process of digitizing a picture definition into a set of pixel-intensity values for storage in the frame buffer
- Not possible to generate exact shapes due to the finite size of pixels
- Need to locate point closest to the given point and store its intensity in the frame buffer
- Well known algorithms for efficient scan conversion of lines, circles and other conic curves
LINE GENERATION USING DDA

- Digital Differential Analyzer (DDA) is an algorithm for efficient scans of straight lines
- Direct use of \( y = mx + c \) for generation
- Need scan between \((x_1, y_1)\) and \((x_2, y_2)\)
- Idea is to obtain next point from previous one
- For \((x_i, y_i)\) next point chosen has x coordinate as \(x_i+1\)
- So \(y_{i+1} = mx_{i+1} + c = m(x_i + 1) + c\)
  \[= (mx_i + c) + m\]
  \(y_{i+1} = y_i + m\)
TRANSFORMATIONS

- Affine transformations
- Cartesian coordinate and homogeneous coordinate
- Transformation in 2D
  - Mathematical representation of Translation, Scaling, Rotation, Reflection, Shearing
- Transformation in 3D
  - Similar to 2D
- Composition of transformations
AFFINE TRANSFORMATIONS

A coordinate transformation of the form
- $x' = a_{xx}x + a_{xy}y + b_x$
- $y' = a_{yx}x + a_{yy}y + b_y$

is called a two-dimensional affine transformation

- General properties of affine transformations: parallel lines transform to parallel lines and finite points map to finite points
- Translation, rotation, scaling, reflection and shear are affine transformations
- Any other affine transformation can be written as a composition of these basic transformations
Homogeneous Coordinates: Need

- The rotational transform can be expressed as
  \[
  \begin{bmatrix}
  \cos\theta & -\sin\theta \\
  \sin\theta & \cos\theta
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  =
  \begin{bmatrix}
  x \cos\theta - y \sin\theta \\
  x \sin\theta + y \cos\theta
  \end{bmatrix}
  \]

- Similarly, scaling and reflection can be expressed in the form \( P' = T \cdot P \)

- However, translation cannot be expressed in this form.

- Need for a different coordinate system so that every transform can be expressed in consistent way.

Computer Graphics and Scientific Visualization
HOMOGENEOUS COORDINATES

- (x, y) in Cartesian coordinate is (x_h, y_h, h) in Homogeneous coordinate
  - Where x_h = x * h and y_h = y * h
- Using the homogeneous coordinates all the affine transformations can be represented in similar ways i.e.
  \[ P' = T \cdot P \]
  where T is the transformation matrix
- In 2D case, for simplicity we may chose h = 1
TRANSLATION

- Repositioning an object along a straight line path from one coordinate location to another:

\[
\begin{bmatrix}
1 & 0 & T_x \\
0 & 1 & T_y \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1 \\
\end{bmatrix} =
\begin{bmatrix}
x + T_x \\
y + T_y \\
1 \\
\end{bmatrix}
\]

- \((T_x, T_y)\) is called a translation/shift vector \(P' = T(T_x, T_y). P\)
SCALING/RESIZE

- Altering the size of an object
- Uniform scaling i.e. $s_x = s_y$

\[
\begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= \begin{bmatrix}
x \cdot s_x \\
y \cdot s_y \\
1
\end{bmatrix}
\]

$P' = S (s_x, s_y) P$

Computer Graphics and Scientific Visualization
**Rotation**

- Repositioning an object along a circular path by an angle $\theta$

\[
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= \begin{bmatrix}
x \cos \theta - y \sin \theta \\
x \sin \theta + y \cos \theta \\
1
\end{bmatrix}
\]

\[P' = R(\theta) \cdot P\]
**Reflection**

- It produces the mirror image of an object.
- The transformation matrices for reflections along x and y axes are:
  
  $\begin{bmatrix}
  1 & 0 & 0 \\
  0 & -1 & 0 \\
  0 & 0 & 1
  \end{bmatrix}$
  $\begin{bmatrix}
  -1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
  \end{bmatrix}$
**Shear**

- A transformation that distorts the shape of an object so that the transformed object appears as if the object were composed of internal layers that had been caused to slip over each other.

- The transformation matrices for the shear in x and y directions are:

  \[
  \begin{bmatrix}
  1 & s_h_x & -s_h_x \cdot y_{ref} \\
  0 & 1 & 0 \\
  0 & 0 & 1 \\
  \end{bmatrix} \quad \begin{bmatrix}
  1 & 0 & 0 \\
  s_h_y & 1 & -s_h_y \cdot y_{ref} \\
  0 & 0 & 1 \\
  \end{bmatrix}
  \]
TRANSFORMATIONS IN 3D

- Very much similar to those in 2D
- Need to add z dimension to the transformation matrices
- More number of rotations (along each axis), reflections (planes, axes and origin)
- An example: Translation

\[
\begin{bmatrix}
1 & 0 & 0 & \Delta x \\
0 & 1 & 0 & \Delta y \\
0 & 0 & 1 & \Delta z \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
COMPOSITION OF TRANSFORMATIONS

- If two transformations (say P and then Q) are performed in succession. So a point ‘x’ is transformed to ‘P.x’ after transformation P and then to ‘Q.P.x’ after transformation Q.
- Result is equivalent to using only one matrix “Q.P”
- This can be generalized to

\[ P_n \cdot P_{n-1} \cdot \ldots \cdot P_2 \cdot P_1 \cdot v_{\text{old}} = v_{\text{new}} \]

where transformations performed are \( P_1, P_2, \ldots, P_n \) successively
COMPOSITION OF TRANSFORMATIONS

1. Leaves of tree are standard size object primitives

2. We transform them

3. To make sub-groups

4. Transform subgroups

5. To get final scene

Computer Graphics and Scientific Visualization
PROJECTIONS

- All our current display technologies are two dimensional
- Reduce the three dimensions of a real or virtual (simulated) object to two dimensions
- Analogous to the role of a camera to capture 2D photographs of the 3D world
- Two basic ways
  - Parallel – Line of sight is parallel to the projection lines
  - Perspective – Line of sight is neither parallel nor perpendicular to the projection lines
Parallel Projection

- Parallel edges remain parallel
- Least distortion is obtained
- Measurement can be done from the projection itself
- Two types
  - Multi view
  - Pictorial
PARALLEL PROJECTION : MULTI VIEW

- In any projected view some of the features are usually missing
- So usually 3 projection along orthogonal axes are taken
- Leads to no distortion and true size
- Standard drawing practice
- Three common projections
  - Top
  - Side
  - Front

Three parallel-projection views of an object, showing relative proportions from different viewing positions.

Computer Graphics and Scientific Visualization
PARALLEL PROJECTION: PICTORIAL

- All the dimensions are visible in one diagram
- Gives somewhat distorted features
- Some of the commonly used pictorial projections are isometric, dimetric and trimetric
Our real world perception is quite different from the parallel projection

Need to make the projection more realistic

- Farther objects look smaller
- Rail tracks, corridors look as though parallel edges are getting closer together
**PERSPECTIVE PROJECTION**

- The line of sight is neither parallel nor perpendicular
- The rate at which parallel edges converge is perspective angle
- Distortion increases with increase in perspective angle
VIEWING TRANSFORMATION

- Displaying an image involves mapping the coordinates of points and lines from the picture to appropriate coordinates on the device where image is to be displayed
- Some terminology
  - World coordinate system
  - WINDOW
  - VIEWPORT
VIEWING TRANSFORMATION/CLIPPING

- Viewing coordinates
  - Generating a view of an object in three dimensions is similar to photographing the object
  - First pick the “view reference point” which act as origin of our viewing coordinate system
  - Now decide the axes of orientation for the camera

- Transformation from World coordinates to Viewing coordinates
  - Translate the view reference point to the origin of the world-coordinate system
  - Apply rotations to align the viewing axes to the world axes
VIEWING TRANSFORMATION

- Once world-coordinate descriptions of the objects in a scene are converted to viewing coordinates, we can project the 3D objects onto the 2D plane.
- Two projection methods – parallel or perspective.
- Intersection points of projected lines on the view plane for the projection of the 2D object on the plane.
CLIPPING

- Sometimes need to display only a portion of a picture
- Need to effectively erase/eliminate part of picture
- Various algorithms for efficient clipping
  - Cohen-Sutherland line clipping algorithm
  - Cyrus-Beck line clipping algorithm
  - Liang-Barsky line clipping algorithm
  - Sutherland-Hodgman polygon clipping algorithm
**Hidden Surface Elimination**

- Also referred as visible-surface detection (although some subtle differences)
- Need to determine which surfaces are visible/hidden from the point of view
- Eliminate the invisible surfaces to give a real picture
- Several Algorithms
  - Depth-buffering, A-buffering, Scan-line method, Depth Sorting, BSP tree, Area subdivision method, ray-casting, wireframe method
**Depth-Buffer Method**

- Known as z-buffer method
- The idea is to compare the surface depths at each pixel position on the projection plane
- For each pixel position \((x, y)\) the object depths can be compared by comparing the \(z\) values
- Assuming a surface to be planar \(Ax + By + Cz + D = 0\), \(z\) can be computed as \(z = (-Ax -By -D)/C\)
- Other surfaces might have complicated equations (explicit or implicit)
Depth Sorting

- Also known as Painter’s Algorithm
- Usually applied to surfaces in form of polygons
- Polygons sorted by the depth of their centroid and then drawn from back to front
- Every paint layer covers up the previous layer
- Works very well for polygons of small and similar size
- Major drawback is the cost of sorting step
**Area Subdivision Method**

- Divide each viewing area into smaller parts until each small area is the part of a single visible surface or no surface at all.
- Surrounding, Overlapping, Inside and Outside surfaces.
- At the limit, subdivision may occur down to the pixel level.
- Can be easily implemented using quadtree/octree data structure.

*Computer Graphics and Scientific Visualization*
COLOR MODELS

- A color model is a method for explaining the properties or behavior of the color within some particular context
- No single color model can explain all the aspects of color, so we can make use of different models to help describe the different perceived characteristics of color
- Some commonly used terms – primary colors, complimentary colors, color gamut, brightness, hue, saturation
- Color models – RGB, YIQ, CMY, HSV, HLS
COLOR MODELS: RGB

Based on the tri-stimulus theory of vision

The color model is represented in unit cube defined on R, G and B axes

The grayscale is the diagonal between white (1, 1, 1) and black (0, 0, 0)

Almost every software support this model

RGB color space:

R, G, B in range [0.0, 1.0]
R, G, B at black = 0.0

Problem:
Specification of hue, saturation, and brightness not intuitive in RGB space
COLOR MODELS : HSV

- More intuitive in nature
- Mimic human perception
- Represented in a cone
- Hue is angle which define the color ($0^\circ$ is red)
- Saturation is radial distance (color purity)
- Brightness varies along the axis of cone

The HSV hexcone

Computer Graphics and Scientific Visualization
Image Processing

- Image processing applies techniques to modify or interpret existing pictures
- Two principal applications
  - Improving image quality
  - Machine perception of visual information
- Major usage
  - Enhance certain features
  - De-emphasize certain other features
- Usually implemented as filters or transformations
FILTERS AND TRANSFORMS

- Used to filter out some of the features from an image or alternately to enhance certain other features

- Transformations
  - Intensity transformations
  - Color transformations
  - Image histogram
  - Discrete convolution
  - Noise introduction
FILTERS AND TRANSFORMS

- Original image
- Increased brightness
- Reduced brightness
- Histogram
- Color inversion
- Color transformation
- Emboss
- Edge Detection
- Sharpen
- Blur
- Noise
- Ripple
FILTERS AND TRANSFORMS (Contd.)
ANIMATION

- Generally implicitly defined as to make the objects change over time
- However the term “Computer Animation” refers to any time sequence of visual changes in a scene
- Rapid display of similar frames
- Typical applications
  - Entertainment (motion pictures and cartoons)
  - Advertising
  - Scientific and Engineering studies
  - Training and education
- Simulation (described ahead)
Animation: Simulation

- Prediction of changes in graphical representation of mathematical formulations
- Prediction of how objects change over time according to physical laws
- Need to consider the effects of other objects/forces on the object in the simulated world
- Maybe real-time!
Animation Sequences: Design

- Major steps involved
  - Storyboard layout
  - Object definitions
  - Key-frame specifications
  - Intermediate frames

- Real time animations may not follow these steps e.g. flight simulator
ANIMATION SEQUENCES: CONSTRUCTION

- **Kinematics**
  - Considers the motion regardless the underlying forces
  - Determined by motion parameters such as position, velocity, acceleration

- **Dynamics**
  - Considers the underlying forces on a system
  - Based on the physics of the problem. Major forces under consideration include electromagnetic, gravitational, friction and other mechanical forces
ANIMATION: KINEMATICS

- Forward Kinematics
  - Motion parameters are specified
  - Special constraints are mentioned
  - Other parameters are computed

- Inverse Kinematics
  - Goals are specified
  - Require to compute intermediate stages using goals are given constraints
ANIMATION : DYNAMICS

- **Forward Dynamics**
  - The forces acting are defined
  - Initial values are known
  - Need to compute subsequent motion
  - Easy to vary motion due to introduction of parameters

- **Inverse Dynamics**
  - Initial and final position of objects are given
  - Often the type of motion is also known
  - Need to obtain the nature of forces acting on the system

Computer Graphics and Scientific Visualization
Hardware Aspects

- GPUs are popular for computer graphics applications used particularly in scientific simulations

- Difference from CPUs
  - In GPUs nearly all the transistors are used in ALUs whereas in CPUs most of the transistors are used in cache
  - Caching less important for computation intensive graphing applications
Power of GPUs have grown much more rapidly than that of CPUs in recent years
Rate of growth of GPUs is ten times the rate of growth of CPUs !!!
**SUMMARY**

- Modeling of objects – Computer Graphics
- Analysis of huge datasets – Scientific Visualization
- Raster and Random scans
- Transformations and projections
- Viewing and clipping
- Visible/hidden surfaces
- Color models and image filters/transforms
- Animations
- Computational Intensive – Need for special hardware
REFERENCES

- Fundamentals of Computer Graphics – S. Chakraborty
Thank You