

Transformationen:

- Translation (Verschiebung): Punkte oder Objekte werden per Vektoraddition bewegt ($x+dx=x'$, $y+dy=y'$, $z+dz=z'$)
- Rotation (um Winkel ϕ): Die Polarkoordinaten sind $x = r \cdot \cos \alpha$, $y = r \cdot \sin \alpha$. Nach der Rotation sind die neuen Koordinaten: $x' = r \cdot \cos(\alpha + \phi)$, $y' = r \cdot \sin(\alpha + \phi)$, also $x' = r \cdot \cos \alpha \cdot \cos \phi - r \cdot \sin \alpha \cdot \sin \phi$, $y' = r \cdot \cos \alpha \cdot \sin \phi + r \cdot \sin \alpha \cdot \cos \phi$,
3D-
Rotationsmatrix:
$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$
- Skalierung: Jeder Eckpunkt, der das Objekt beschreibt, wird skaliert ($x' = s_x \cdot x$, $y' = s_y \cdot y$, $z' = s_z \cdot z$) oder multipliziert
$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$
- Scherung (auch Transvektion): Überführung einer Figur in eine andere Figur unter Beibehaltung der Höhe. Die Fläche bleibt erhalten.

Koordinatensystem: Dient der Positionsangabe von Punkten im Raum. Rechtshändige Koordinatensysteme bezeichnen den positiven Drehsinn (Drei-Finger-Regel)

Homogene Koordinaten: Bei der Überführung eines Punktes in sein homogenes Äquivalent wird er als Gerade dargestellt. Eine zusätzliche Dimension muss eingeführt werden, die einen beliebigen, aber nicht komplexen Wert ungleich 0 enthalten darf. Bei der Rückrechnung werden einfach die übrigen Koordinaten durch den aktuellen Wert der nun überflüssigen Dimension geteilt.

- Weltkoordinaten sind immer durch ein Fenster der Dimension begrenzt. Weltkoordinaten werden auch Datenkoordinaten genannt.
- Gerätekoordinaten beschreiben die Koordinaten der Ausgabegeräte
- normierte Koordinaten stellen ein neutrales Koordinatensystem dar, das unabhängig von den Gerätekoordinaten funktioniert

RGB: Farben werden additiv gemischt, rot, grün und blau ergeben gemeinsam weiß.

CMY: Farben werden subtraktiv gemischt, also ergeben cyan, magenta und yellow schwarz.

CMYK: Schwarz als weitere Farbe, die das „gemischte“ Schwarz ablöst, weil es eine schmutzige Darstellung ist

geometrische Projektion

- ist eine Abbildung von $n+1$ Dimensionen auf n oder weniger Dimensionen. Sie werden anhand ihrer Projektionsstrahlen (Objektpunkt und zugehörige Bildposition) klassifiziert.
- orthogonale Projektion: die Punkte der Objektebene sind parallel und stehen normal auf der Bildebene ($(x, y, z) \rightarrow (x, y)$)
- Zentralprojektion: Alle Projektionsstrahlen treffen sich in einem Projektionszentrum, die Bildpunkte ergeben sich als Schnittpunkt der Projektionsstrahlen mit einer Bildebene. Die Größe des Bildes ist deshalb abhängig vom Abstand zwischen Projektionszentrum und Bildebene. Parallelität und Winkeltreue gehen dabei verloren.
- Parallelprojektion: Die Projektionsstrahlen sind parallel (das Projektionszentrum ist im Unendlichen)

Abbildung des Modells:

- Rechtskoordinatensystem zur Beschreibung des Objekts
- Linkskoordinatensystem zur Beschreibung des Bildes
- Abbildungsparameter: view reference point (Ursprung des Bildkoordinatensystems), view plane

normal (Blickrichtung auf das Objekt) und Vektor up (natürliche Aufwärtsrichtung) (\Rightarrow gluLookAt)

Polyeder:

- wird in 3D beschrieben, obwohl es eine 2D-Fläche darstellt
- Darstellung durch: - Liste der Eckpunkte – Liste der Kanten – Liste der Polygonflächen
- „verbundene“ Darstellung, wenn jedes Polygon mindestens eine Kante mit mindestens einem anderen Polygon teilt (der eingeschlossene Körper heißt Polyeder)

Reflexionsmodelle

- sind Modelle zur Berechnung des Lichts, das ein Objekt abstrahlt. Sie sind abhängig von einfallendem Licht auf die Oberfläche, der Oberflächenbeschaffenheit und dem Beobachterstandpunkt

Diffuse Reflexion:

- das Licht verteilt sich gleichmäßig in alle Richtungen
- Reflexion ist unabhängig vom Beobachtungspunkt, aber abhängig vom Winkel zwischen Lichtstrahl und Normalenvektor
- Oberfläche, die das Licht gleichmäßig in alle Richtungen verteilen, nennt man „matt“

Ambiente Reflexion:

- Hintergrundbeleuchtung, indirekte Beleuchtung, die sich aus von anderen Objekten reflektiertem Licht zusammensetzt
- die Beleuchtung besitzt keine Richtung und wird deshalb als Konstante für eine bestimmte Abbildungsumgebung berechnet
- ohne ambiente Reflexion sind Schattentiefscharf, der Übergang von Schatten zu Beleuchtung ist abrupt

spiegelnde Reflexion

- Menge der spiegelnden Reflexion, die vom Beobachter aus bemerkbar ist, hängt ab vom Winkel zwischen dem auf der Oberfläche auftreffenden Blick des Betrachters und der Lichtquelle
- der Spiegelkoeffizient ist abhängig von der Oberflächenbeschaffenheit

Schattierung

- konstante Schattierung
Lichtquelle ist im Unendlichen, keine entfernungsabhängige Lichtabnahme wird berücksichtigt, nur ambiente und diffuse Beleuchtung werden modelliert
 \Rightarrow die Normale des Polygons bleibt gleich, Lichteinfall bleibt gleich, es wird kein Beobachtungspunkt berücksichtigt. Es gibt einen Schattierungswert pro Polygon (unrealistische Darstellung, Kanten werden verstärkt)
- Gouraud-Schattierung
Schattierung wird an den Eckpunkten des Polygons berechnet und im Inneren interpoliert. Dazu wird zu einem bestimmten Eckpunkt E die Normale n aller angrenzenden Polygone berechnet und interpoliert
- bilineare Interpolation
Schattierungswerte sind an den Eckpunkten bekannt, Gouraud-Schattierung wirkt realistischer. Wenn nur diffuse Reflexion auftritt, ist Gouraud ausreichend.
Sind die Polygone groß, gibt es ähnliche Probleme wie mit konstanter Schattierung. Spiegelnde Reflexion ist nicht zufriedenstellend berücksichtigt.
- Phong-Schattierung
die Oberflächennormalen werden für jeden Punkt im Inneren des Polygons interpoliert und explizit die Schattierungswerte für jedes Pixel berechnet.
1. zu allen Eckpunkten der Polygone die dazugehörigen Oberflächennormalen berechnen
2. mit bilinearer Interpolation die Oberflächennormalen für jeden Punkt im Inneren eines Polygons berechnen

3. die Schattierungswerte zu jeder Oberflächennormalen mittels des Phong-Reflexionsmodells berechnen

Normalenvektoren

- in den Ecken und Kanten eines Polyeders stoßen verschiedene Polygone mit verschiedenen Normalenvektoren aufeinander. Dadurch sind die Ecken und Kanten sichtbar
- zur Berechnung der Normalen werden die Vektoren links und rechts einer Ecke berechnet. Diese werden dann normalisiert mittels des Vektorprodukts

Texturen

- Texturen bestehen aus einer Beschreibung der Textur (meist ein 2D-Bild) und der Abbildung der Textur („wie kommt die Tapete an die Wand?“)
- Abbildung: 2D-Textur-Koordinaten \rightarrow 3D-Objekt-Koordinaten
- Orthogonalprojektion und Skalierung sorgt für unetliche Kanten
- für Polyeder: jeder Ecke werden explizit die Texturkoordinaten u, v zugeordnet und linear oder bilinear zwischen den Ecken interpoliert
- für parametrische Flächen: Parameterwerte der parametrischen Fläche werden als Texturkoordinaten benutzt
- Mip Mapping speichert 2D-Textur-Bilder in verschiedenen Auflösungen und gibt an entfernte/kleinere Objekte kleinere Auflösungen weiter, um Aliasing-Effekte zu verwässern
- 1D-Texturen werden benutzt, wenn Veränderungen der Farbe nur in eine Richtung erfolgt
- 3D-Texturen werden für die Modellierung von ungleichmäßigem Material (z.B. Holz) verwendet. Sie sind sehr speicherintensiv
- Bump Mapping modelliert die Oberflächenbeschaffenheit, z.B. Rauheit.

Szenegraph

- dient der hierarchischen Modellierung von Objekten und letztlich der „Welt“
- Koordinaten: Objektkoordinaten, Weltkoordinaten und Kamerakoordinaten werden als Transformationen festgehalten
- ist ein gerichteter Baum, also ein gerichteter Graph ohne Kreise
- Knoten repräsentieren einen transformierten Teil der Szene
- innere Knoten enthalten die Struktur der Szene
- Blätter enthalten die Geometrie (Einzelobjekte)
- Kanten sind eine „besteht aus“-Beziehung und sorgen für die Verknüpfung der Transformationen

Normalen von „krummen“ Flächen

- Approximation der Normalen: in der Mitte einer glatten Fläche wird der Mittelwert gebildet und am Rand modifiziert
- Normale an einer ebenen Kurve: Wenn die Funktion vorliegt, steht die Normale senkrecht auf der Tangente, deren Steigung aus der Ableitung der Funktion hervorgeht

Entfernen verdeckter Linien und Oberflächen

- das Eliminieren verdeckter Oberflächen bei konvexen Polygondarstellungen wird als Culling bezeichnet
- Z-Buffering (Tiefenpuffer) braucht zusätzlichen Speicher, um Tiefe (also die Distanz von Bild zu Objekt) zu speichern. Zusätzlich zu jedem Bildpunkt wird die Entfernung zum Objektpunkt berechnet. Wenn ein zweiter Bildpunkt an die gleiche Stelle gezeichnet werden soll, werden die z-Werte verglichen. Nachteil: Zusätzlicher Speicherplatz
- List Priority Algorithm: Polygone werden nach ihrer Distanz zur Bildebene sortiert, dann werden Objekte „von hinten nach vorne“ abgebildet, damit die vorderen die hinteren überschreiben. Wenn sich Polygone überschneiden, werden sie geteilt und nochmals sortiert.

Nachteil: Eventuell großer Sortieraufwand und Mehraufwand durch Berechnung aller Schattierungen (auch solcher, die überschrieben werden)

- Raytracing: Eine Gerade wird vom Beobachtungspunkt aus durch ein Pixel gelegt. Schnittpunkt mit Objektpunkt definiert den auf diesem Pixel abzubildenden Punkt. Nur der erste (nächste) Punkt zählt. Für diesen Objektpunkt wird eine Schattierung berechnet und im Pixel abgelegt. Ergebnisse bei einer Kugel: Bei imaginären Zahlen wurde kein Schnittpunkt gefunden, bei einem reellen Schnittpunkt ist der Strahl die Tangente, bei zwei reellen Schnittpunkten schneidet der Strahl die Kugel; es wird der Schnittpunkt mit kleinerem t gewählt

Transparenz und Schatten

- Transparenz: linearer Parameter t bestimmt, wie viel Gewicht zwei Objekten zufällt.
- Schatten: Schattenwurf wird von OpenGL nicht unterstützt. Er wird berechnet mittels Hidden Surface Algorithmen: Beobachtungsstandpunkt wird in die Lichtquelle gesetzt: Verdeckte Flächen sind Schatten (die Fläche wird auf die nicht sichtbaren Ebenen projiziert)

Rasterung einer Linie

Digital Differential Analyzer Algorithm:

- Berechnung der Punkte einer Linie aus dem vorher berechneten Startpunkt

Bresenham Algorithmus:

- In die „schnelle“ Richtung wird immer ein Schritt gemacht und je nach Steigung hin und wieder zusätzlich ein Schritt in der „langsameren“ Richtung
- Verwendung einer Fehlervariable, die bei einem Schritt in x -Richtung den Wert dy subtrahiert bekommt und bei Unterschreitung des Nullwerts einen y -Schritt setzt und den größeren Wert dx addiert

Midpoint-Algorithmus

- Linie läuft zwischen zwei ganzzwertigen Koordinaten. Von diesen wird die Mitte ermittelt und geprüft, ob die Linie sich ober- oder unterhalb der Mitte befindet

Scan-Line Algorithmus

- Paritäts-Flag beginnt bei „even“ (gerade, kein Pixel zeichnen) und ändert bei Übertretung einer Linie zu „odd“ (ungerade, Pixel zeichnen)
- obere und rechte Ränder werden nicht gezeichnet, untere und linke Ränder werden gezeichnet
- horizontale Kanten ändern die Parität nicht
- Eckpunkte:
 - Eckpunkt für zwei Linien mit $y_{\min/\max}$ in A wird 2x gezählt (eine Spitze ragt von oben oder unten in die Figur)
 - Eckpunkt für 1 Linie mit y_{\min} in B und 1 Linie mit y_{\max} in B wird 1x gezählt (Spitze ragt horizontal in die Figur hinein oder aus der Figur heraus)
 - Eckpunkt für 2 Linien mit y_{\max} in C wird nicht gezählt (Spitze ragt oben oder unten aus der Figur heraus)

Clipping

- wichtige Eigenschaften:
 - beide Endpunkte sind im Fenster (wird akzeptiert)
 - beide Endpunkte sind außerhalb des Fensters (wenn die Linie auch außerhalb ist, wird die Linie verworfen, sonst werden zwei neue Endpunkte berechnet)
 - ein Punkt ist im Fenster, einer außerhalb (neuer Endpunkt wird berechnet)
- Cohen-Sutherland-Algorithmus
 - funktioniert beim Clipping einer Linie gegen einen Quader
 - 9 Felder im Format YYXX (01, wenn kleiner als min, 10, wenn größer als max, 00, wenn innerhalb)
 - Code1 OR Code2 = 0 => Linie innerhalb des Quaders
 - Code1 AND Code2 ungleich 0 => Linie außerhalb des Quaders

- Code1 OR Code2 ungleich 0 und Code1 AND Code2 = 0 => Linie wird mit Quader geschnitten

Freiformkurven und -flächen

- Jede stetige Funktion auf einem Intervall kann durch Polynome beliebig genau angenähert werden
- zu jeder Knotenverteilung existiert eine stetige Funktion, so daß die Polynominterpolation „versagt“
- Bézierkurve: numerische parametrisch modellierte Kurve. Sie liegt innerhalb der konvexen Hülle des Kontrollpolygons
- Splines: Mittels Interpolation und Approximation werden Kurven gebildet

OpenGL

Transformationen: glTranslatef, glScalef, glRotatef

Texturen: glTexCoord2f (links unten, rechts unten, rechts oben, links oben)

Projektion: glOrtho und gluLookAt, glFrustum und gluLookAt, gluPerspective (Gesichts-Winkel, Bildverhältnis, zNear, zFar) und gluLookAt

glEnable(GL_COLOR_MATERIAL);	spezifiziert, welche Materialeigenschaften sich auf die aktuelle Farbe auswirken sollen
glEnable(GL_LIGHTING);	beleuchtete Objekte zeichnen
glEnable(GL_NORMALIZE);	Normale werden automatisch normalisiert
glLightModel(GL_SMOOTH);	Farbverlauf über die Polygonfläche wird bei der Rasterisierung berechnet
glPolygonMode	wählt aus, wie die einzelnen Polygone beim Rastern gezeichnet werden sollen
glDepthFunc	legt fest, wann ein Fragment den Tiefentest im Tiefenpuffer besteht
glClearDepth();	legt den Wert fest, mit dem der Tiefenpuffer überschrieben wird
glMaterialfv();	Lichteigenschaften des Materials übergeben
glLightfv();	Verhalten und Positionierung des Lichtes
glutSwapBuffers();	tauscht berechneten mit sichtbarem Buffer
glViewport();	passt das Format dem Betrachtungsfenster an
glGenTextures();	legt Speicherplätze eines Arrays (Anzahl und Name) für Texturen fest
glBindTexture();	alle folgenden texturrenbasierten Operationen werden diesem zugeordnet
glPixelStorei();	bestimmt die Bedingungen für die Ausrichtung von Pixelzeilen im Speicher