

Computergraphik

Prof. Dr.-Ing. Kerstin Müller

Kapitel 4

Transformationen und Projektionen

Gerl's Game



Translationen

- Translationen müssen in 2D und in 3D durchgeführt werden.

- Schieben der Schachfiguren auf dem Brett (2D).
- Schachfigur bewegt sich von der Ursprungs-Position auf einer 3D - Raumkurve zum Ziel-feld (3D).
- Vereinfacht: Stückweise lineare Transformation bildet die Raumkurve.



Weitere Transformationen

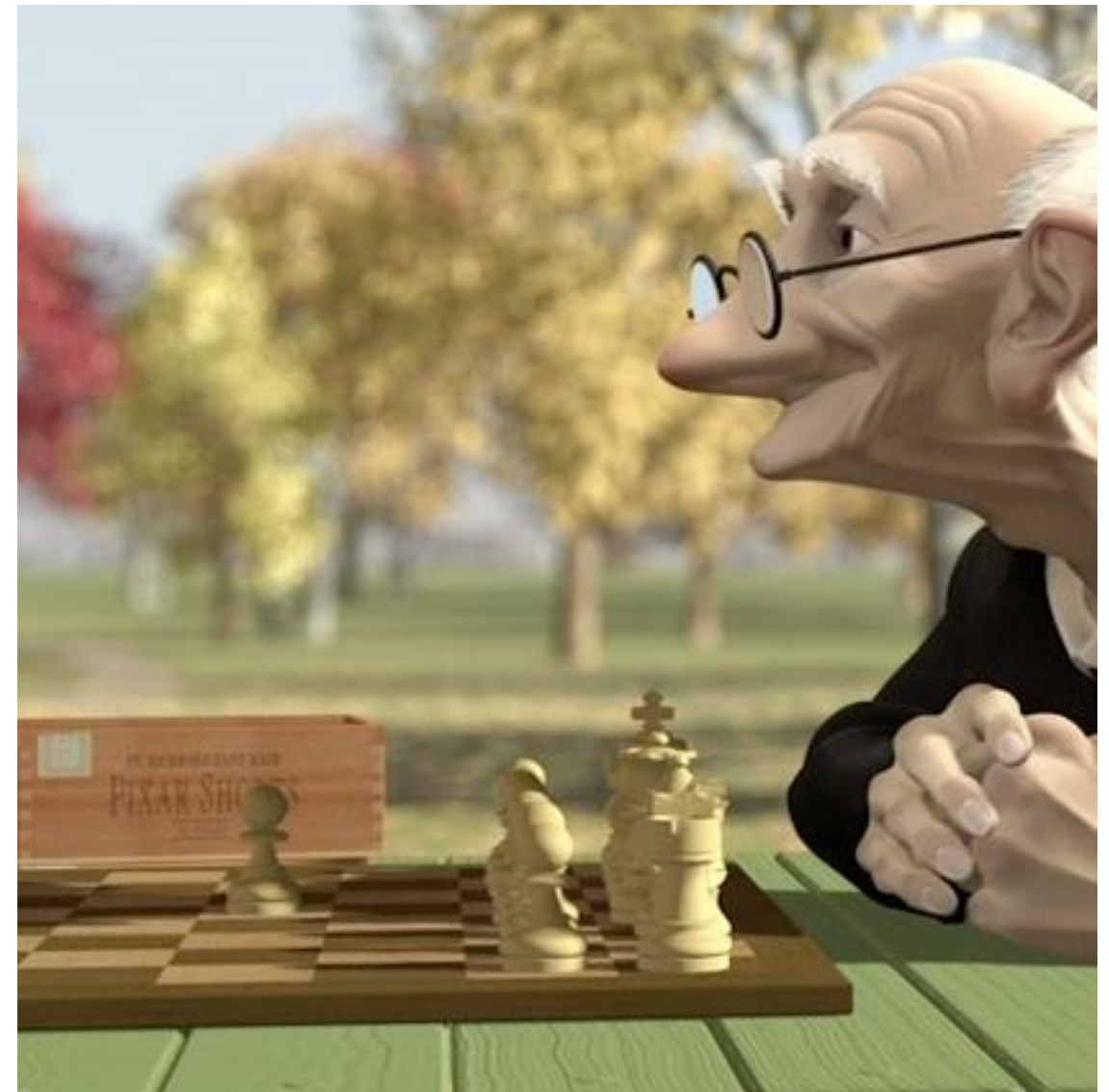
- Neben Translationen müssen auch Rotationen durchgeführt werden, falls eine Schachfigur geschlagen wird und in der Kiste landet oder der König umgeworfen wird, wenn das Spiel verloren ist.
- Dieses Kapitel behandelt Translationen, Rotationen und Scherungen.



Transformation 3D zu 2D

■ Integration eines Kamerakonzeptes in die Berechnung des 2D-Bildes aus dem 3D-Modell.

- Einführung eines Kamerakoor-
dinatensystems.
- Welcher Teil der Gesamt-Szene
ist überhaupt sichtbar?
(Clipping)
- Berechnung einer perspekti-
vischen Projektion (weit entfernte
Objekte sollen kleiner erscheinen)
- Hinzufügen von Tiefenunschärfe.



Kapitelübersicht

- 4.1 Die Computergraphik Pipeline
- 4.2 Koordinatentransformationen
- 4.3 Transformationen in der Ebene
- 4.4 Transformationen im Raum
- 4.5 Projektionen
- 4.6 Windowing
- 4.7 Clipping

4.1 Die Computergraphik Pipeline

- Was ist prinzipiell nötig, um ein computergraphisch erzeugtes Bild auf dem Monitor/Drucker auszugeben?
- Definition (Computergraphik-Pipeline): Modellvorstellung, welche die einzelnen Verarbeitungsschritte vom Modell bis hin zum fertigen Bild verknüpft und zeitlich anordnet.
- Die tatsächliche Implementierung kann in Hardware, in Software oder teils-teils erfolgen.
- Das Konzept von Pipelining kann auch konkret umgesetzt werden.

Das Objekt-Koordinatensystem

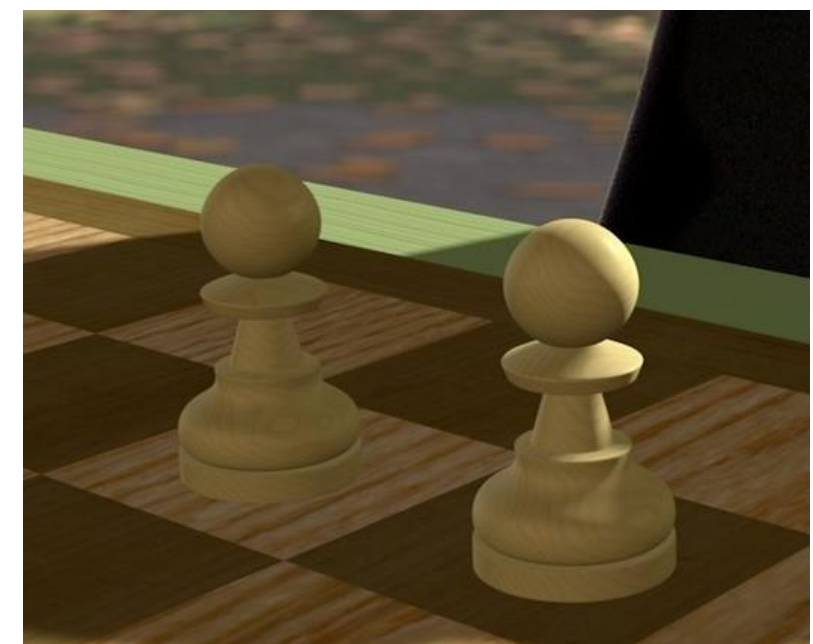
■ Schritt 1: Beschreibung der 3D-Objekte

- Objektbeschreibungen können aus einer Datenbank geladen oder selbst beschrieben werden.
- Dies geschieht aus Gründen der Einfachheit in einem eigenen Objekt Koordinatensystem oder lokalen Koordinatensystem.
- Man kann so z.B. den Ursprung des lokalen Koordinatensystems in die Mitte eines symmetrischen Objekts legen.
- Die Objekte werden nun mittels Transformationen, beispielsweise Translationen (d.h. Verschiebungen), Rotationen oder Skalierungen an die ihnen zugedachten Plätze in der Szene platziert.
- Mathematisch wird dies durch Multiplikationen der Transformationsmatrizen mit den die Objekte definierenden Punkten durchgeführt.

Das Objekt-Koordinatensystem

■ Beispiel: Modellierung der Schachfiguren

- Jeder Typ von Schachfigur wird nur einmal in der Datenbank abgelegt, d.h. es gibt z.B. nur einen Bauern in der Datenbank obwohl er 16 mal auf dem Spielfeld (mit verschiedenen Positionen und Farben) vorkommt.
- Das lokale Koordinatensystem ist definiert durch die Standebene und die Hochachse der Figur.
- Für das verschiedene Auftreten der Figuren wird je eine Kopie mit der entsprechenden Transformation (zunächst Verschiebung) versehen.
- Zunächst einfach: 2D-Verschiebung auf dem Brett.



Das Objekt-Koordinatensystem

■ Beispiel: Modellierung der Schachfiguren

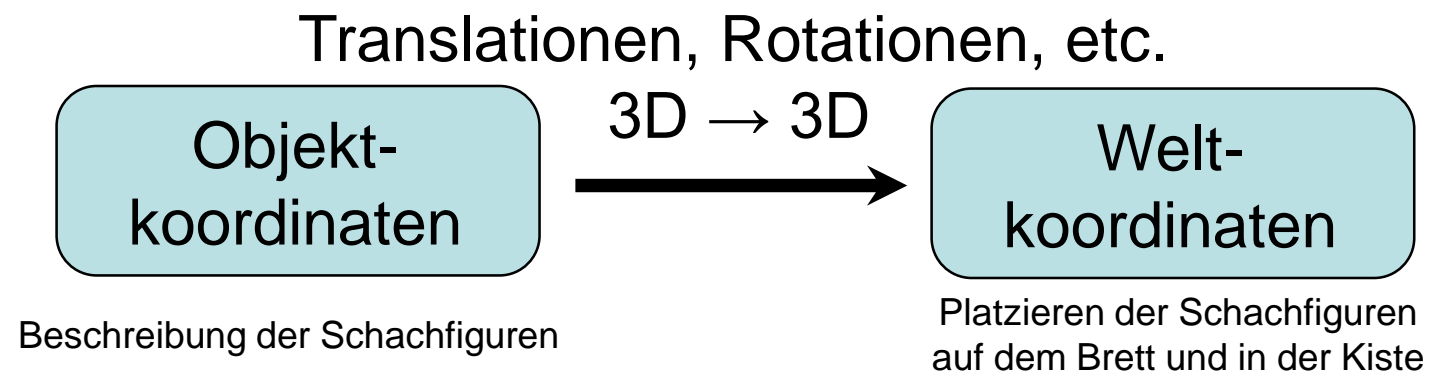
- Falls eine Figur geschlagen wird, muss sie in beliebiger Orientierung (z.B. liegend in der Holzkiste) dargestellt werden können.
- Daher sind Rotationen und Verschiebungen im 3D Raum (die oben in der Kiste liegende Figur ist „höher“ positioniert als das Spielfeld) notwendig.
- Ein lokales Koordinatensystem ist insbesondere bei spiegelsymmetrischen Objekten sinnvoll, wenn eine Koordinatenebene die Spiegelebene ist.
 - Hier kann bei interaktiver Veränderung des Objektes nur eine Seite modifiziert werden, und die Symmetrie durch Spiegelung an der Koordinatenebene (sehr einfach: nur das Vorzeichen einer Koordinate invertieren) beibehalten werden.

Das Welt Koordinatensystem

■ Schritt 2: Zusammenstellung aller Objekte im Raum

- Sind alle Objekte in der Szene platziert worden, ist das Ergebnis eine Beschreibung der Szene im Welt-Koordinatensystem oder globalen Koordinatensystem.
- Technisch wird dieser Schritt in der heutigen Graphik-Hardware ausgeführt.
- Die Regeln für den Zusammenbau der Szene , d.h. welche Transformationen und in welcher Reihenfolge werden in einem so genannten Szenegraphen gespeichert.
- Auf diese Art und Weise werden nicht nur Objekte, sondern zum Beispiel auch virtuelle Lichtquellen platziert.

Computergraphik-Pipeline Stufe 1



Das Kamera-Koordinatensystem

■ Schritt 3: Platzierung einer virtuellen Kamera

- Der Kamera wird ein eigenes lokales Koordinatensystem das Kamera-Koordinatensystem oder Eye-Koordinatensystem zugeordnet, aus dessen Sicht sich alle im Welt-Koordinatensystem beschriebene Objekte betrachten lassen.
- Also Betrachtungsweise der Szene aus der Sicht einer Kamera.
- Analogie: Der Pilot eines Flugzeugs schaut in eine Szene
 - Was er sieht hängt von seiner momentanen Position, der Flugrichtung und der Orientierung (Rotation bei Kurvenflug) ab.
 - Es wird hier noch davon ausgegangen dass er unendlich weit blicken kann.
- Prinzip Flugzeug wird oft in virtuellen Rundflügen durch eine Szene verwendet:

Beispiel: Virtueller Flug durch eine Szene

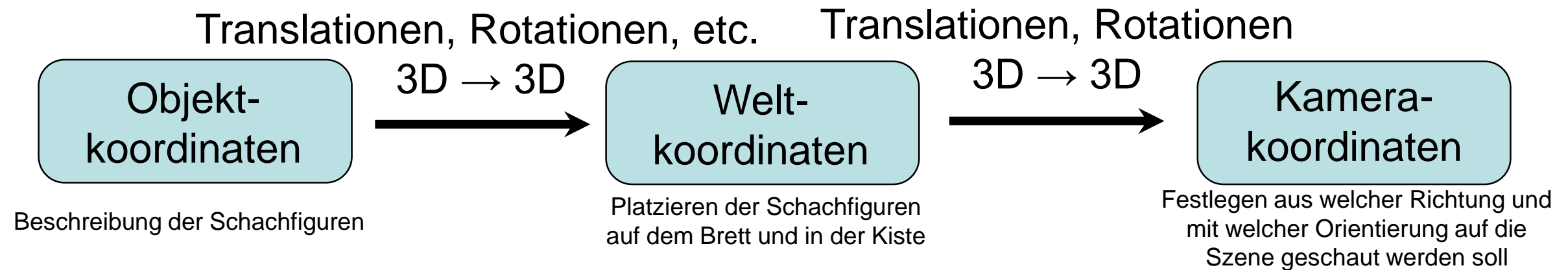
- Umsetzung in Computerspielen:
 - Die Szene (hier Mine) ist fix, die Sichtweise ändert sich.
- Umsetzung in Animationen, hier z.B. in [Ocean`s Eleven](#)



Welt- und Kamera-Koordinaten kombiniert

- Wegen der Analogie
Platzierung der Kamera in einer Szene und
Szene vor der Kamera ausrichten
wird die Transformation von Objekt in Weltkoordinaten
und weiter in Kamerakoordinaten oft in einem Schritt
ausgeführt.
- Ein Beispiel ist die Modelview-Matrix in OpenGL.
- Im Bereich dieser beiden Stufen ist dann auch das
Konzept des Szenegraphen (nicht Teil von OpenGL)
angesiedelt.

Computergraphik-Pipeline Stufe 1-2



Clipping-Koordinaten

- Stufe 3: Clipping Koordinaten
- Festlegen wie groß das Sichtbarkeitsvolumen ist.
- Üblicherweise wird ein Würfel oder ein Pyramidenstumpf (sogenanntes Frustum) verwendet.
- Unproblematisch falls Objekte ganz oder gar nicht innerhalb des Sichtbarkeitsvolumens liegen.
- Bei Objekten die teilweise sichtbar sind ist eine Sonderbehandlung notwendig.
 - Entweder Objekte als innen definieren und mitprojizieren, dann in 2D Ränder zuschneiden
 - Oder Schnittoperationen in 3D durchführen.

Computergraphik-Pipeline Stufe 1-3

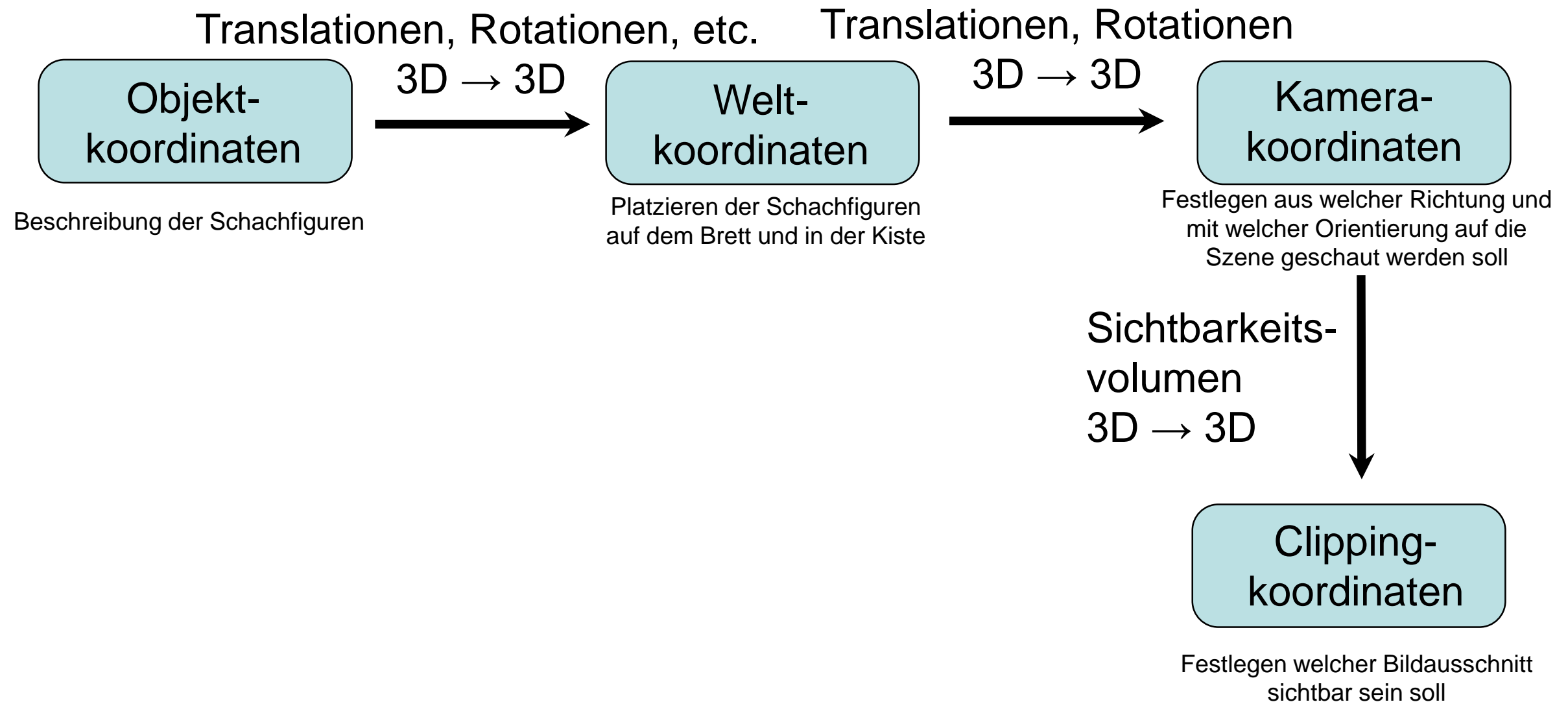
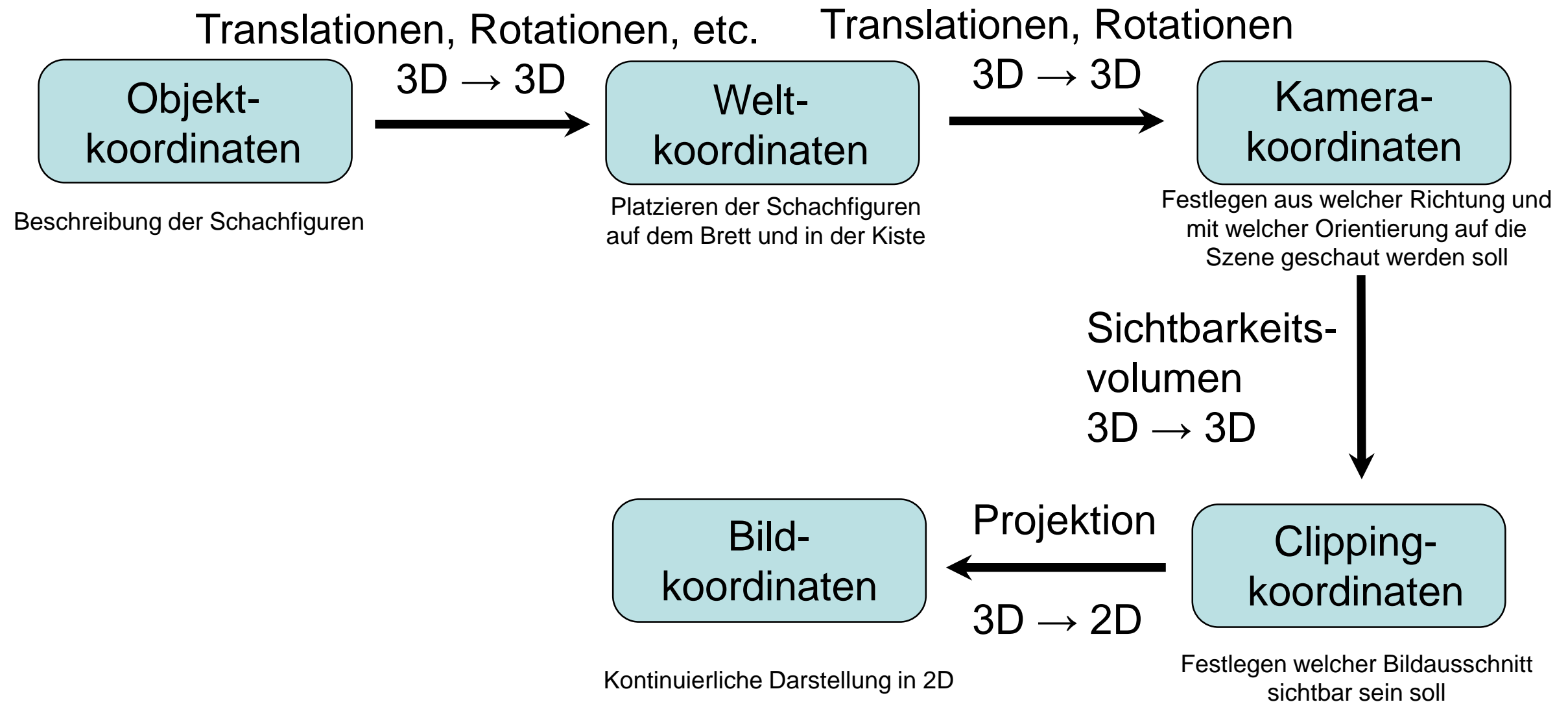


Bild Koordinaten

- Stufe 4: Beim Übergang von Clipping Koordinaten zu Bild Koordinaten wird der logische Übergang von 3D auf 2D Koordinaten vollzogen.
- Die Projektion wird ebenfalls durch eine Matrixmultiplikation durchgeführt (i.d.R. in Hardware realisiert).
- Es entstehen Koordinaten in der Projektions- oder Bildebene, im Bild-Koordinatensystem oder screen space. Die Darstellung ist noch immer kontinuierlich!!!
- Dabei interessieren allerdings nur Punkte innerhalb eines rechteckigen Bereichs, des Windows oder Fensters.
- Das Zuschneiden wird also auch in dieser Stufe realisiert.

Computergraphik-Pipeline Stufe 1-4



Geräte-Koordinaten

- Ausgabegeräte (meist Bildschirm), verwenden ein diskretes 2D-Raster von einzelnen Bildpunkten, den Pixeln .
 - Die Bildpunkte sollen quadratisch sein und identische Größe haben.
- Als Organisation zur Speicherung des Rasterbildes dient eine zweidimensionale Speichermatrix, der Bildschirm-speicher oder frame buffer.
 - Die Größe (Höhe,Breite) des frame buffers bestimmt die virtuelle Bildschirmauflösung.
 - Diese kann von der festen physikalischen Auflösung des Ausgabegeräts abweichen.
 - Die Größe des Framebuffers ist oft per Software einstellbar.

Geräte-Koordinaten

- Die Gerätekoordinaten beziehen sich auf die virtuelle Bildschirmauflösung nicht die physikalische!!!
- Stufe 5: Die Umrechnung der kontinuierlichen 2D Darstellung in eine Rasterdarstellung wird heute ebenfalls komplett in Hardware realisiert.
- Ebenfalls in dieser Stufe wird die so genannte Window-Viewport Transformation durchgeführt.
 - Der Viewport gibt den Bildschirmbereich an, in dem der Inhalt eines Fensters abgebildet werden soll. Window gibt an was gezeichnet werden soll.

Zusammenfassung: Graphik-Pipeline

