Computergrafik Zusammenfassung

Dreiecksnetze und Licht

* Kreuzprodukt
* Diffuses Licht
* Halbkanten
* Algorithmen
  + Für einen Vertex die 2-Ring-Nachbarn bestimmen
  + Oberfläche eines Dreiecksnetzes berechnen
* Freiheitsgrade
* Szenengraphen zeichnen

Kurven

* Kontrollpunkte und -polygon
* Kurven
* Bezier-Basisfunktion,-Kurve und -Spline
* Interpolation
* Konvexe Hülle

Datenstrukturen

* BSP-Baum
* Laufzeitkomplexität
* Culling
* Octree

Simulation/Tracking

* Featurepunkte
* Homogene Matrix
* Explizietes Euler-Verfahren, Integrationsschritte
* Grundlagen
  + Vektoren
    - Normierung: Veränderung eines Vektors, sodass er seine Richtung beibehält aber die Länge 1 bekommt
    - Ortsvektor: Position im Raum
    - Richtungsvektor: Verschiebung im Raum
    - Skalarprodukt   
      
    - Orthogonalität

, Überprüfung: Skalarprodukt der beiden Vektoren ergibt 0

* + - Kreuzprodukt



* + - * Länge des Ergebnisvektors entspricht der Fläche des von den beiden Vektoren aufgespannten Parallelogramms
      * Ergibt den Sinus des Winkels der aufgespannten Vektoren
      * Ergebnisvektor steht senkrecht auf den beiden Vektoren (Normalvektor)
      * Rechte-Hand-Regel: Daumen , Zeigefinger , Mittelfinger
  + Abstand Punkt-Gerade







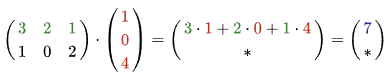
* + Kamera
    - Bestandteile
      * Position
      * Blickrichtung (Referenzpunkt)
      * Oben-Vektor
    - Sichtbarkeit
      * Near-Clipping: Kleinere Entfernungen werden ignoriert
      * Far-Cipping: Größere Entfernungen werden ignoriert
      * Öffnungswinkel (fovy): Verhältnis von Breit und Höhe (aspect)
* Dreiecksnetze
  + Matrizen
    - Transponierte

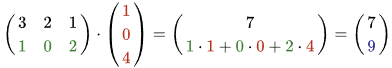


* + - Inverse

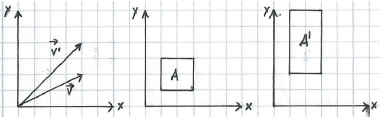
 , 

* + - Matrix-Vektor-Produkt





* + - Skalierungsmatrix
      * Dimensionsrichtungen auf der Hauptdiagonalen



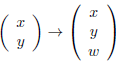
* + - Rotationsmatrix



* + Baryzentrische Koordinaten
    - Ein beliebiger Punkt in einem Dreieck kann mit B. K. angegeben werden
      * Mit Abstand zur Kante gegenüber usw.
      * als Punkte im Dreieck
* Normalen
* Transformationen



* + Homogene Koordinaten
    - Zusammenfassung affiner Transformation und Translation in einer Matrix
    - Erweiterung aller Matrizen und Vektoren um eine Zeile und eine Spalte
    - Die zusätzliche Komponente nennt man -Komponente



* + - Beim Übergang reguläre Vektoren 🡪 homogene Vektoren muss der Wert der -Koordinate 1 sein, ist das nicht der Fall, teilt man den gesamten Vektor durch .



* + - Bei homogenen Matrizen werden Werte mit 0 aufgefüllt, außer der auf der Hauptdiagonalen. Dieser wird 1



* + - Die Translationsmatrix um den Vektor also



* + Basisvektoren
    - Kanonische Basisvektoren



* + - Ein 2D-Vektor  bedeutet also eigentlich:



* + Transformationsmatrizen
    - Transformation zwischen beliebigen Koordinatensystemen
      * 
    - Kameratransformation
      * Punk p eines Objekts
        + Homogene Koordinaten:
        + Translationskomponente: Verschiebung um , homogen:
        + Rotationskomponente: um die z-Achse
        + ist also:
        + Für die Transformation in Weltkoordinaten ergibt sich
    - View-Transformationen
      * Blickrichtung der Kamera wird auf die Richtung der z-Achse überführt
      * #TODO
    - Perspektivische Transformation
      * #TODO
    - Pixel
      * #TODO
  + Tracking
    - #TODO
* Kurven und Flächen
  + Basisfunktionen
* Datenstrukturen
* Simulation
* Prozedurale Generierung