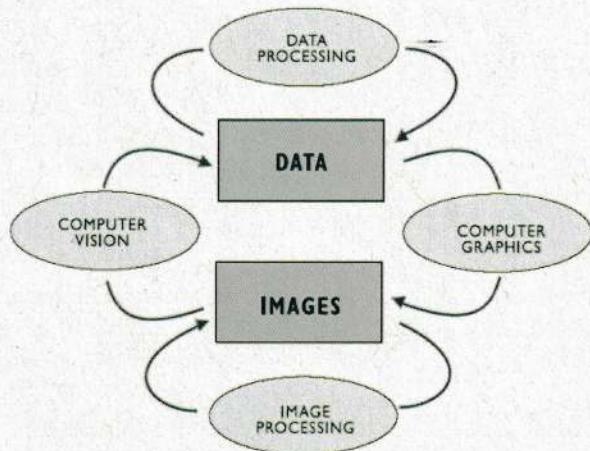


Einführung in Visual Computing

Teilgebiete:



Data Processing

nicht Teilgebiet von Visual Computing

Image Processing

Bildaufnahme, Bildmanipulation / Bildverarbeitung

Computer Graphics

Das Erzeugen von Grafiken, Animationen
Das Visualisieren von Daten

Computer Vision

Entwickelte sich als Teilgebiet von AI-Artificial Intelligence
Bildverstehen, das Erfassen von Inhalten im Bild

CV1: Einführung in Computer Vision

Definition Computer Vision

1. Bilderkennung - Datenerfassung

Maschinelles Sehen / Machine Vision

Typische Einsatzgebiete sind industrielle Herstellungsprozesse, die Automatisierungstechnik und die Qualitätssicherung

Mustererkennung

zB OCR - Optical Character Recognition

Detection von Ereignissen

Videotracking

Objekterkennung

Maschinelles Lernen

Katalogisierung

Bewegungsabschätzung

2. Reproduktion

Szenenreproduktion

Bildrestaurierung

Reproduktion des menschlichen Sehens

Das menschliche Auge ist aber fehlerbehaftet und wir wissen nicht was nach der Reizaufnahme im Menschlichen Auge passiert.

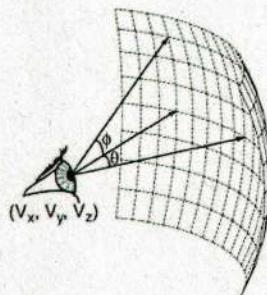
Plenoptische Funktion (idealisiertes Modell)

Was kann potentiell gesehen werden? Was sehen wir, wenn wir in die Welt schauen?

Der Raum ist voll mit Lichtstrahlen. Eine plenoptische Kamera, auch Lichtfeldkamera genannt, erfasst die Richtung einfallender Lichtstrahlen.

$$p = P(\Theta, \Phi, \lambda, x, y, z, t)$$

mit: Θ, Φ Raumwinkel, x, y, z Position im Raum, t Zeit



Die plenoptische Funktion ermittelt die Intensität des Lichtes und hat theoretisch unendliche Schärfentiefe.

Θ, Φ Raumwinkel / Perspektive

λ Wellenlänge des Farbspektrums (konkretisiert welches Licht)

V_x, V_y, V_z Position im Raum

t spezifischer Zeitpunkt

Camera Obscura (idealisiertes Modell)

Apertur = ∞ klein

Geschlossene Box mit kleiner Öffnung / Apertur „Pinhole“ projiziert Objekt verdreht im inneren.

Die einfachste Art der perspektivischen Projektion:

In der Computergrafik wird meist ein idealisiertes Kameramodell mit einem unendlich kleinen Loch verwendet.

Dieses Kameramodell kann keine Tiefunschärfe simulieren, d.h. alle Objekte werden scharf abgebildet.

Es ist eine ideale Abbildung einer 3D-Szene auf einem 2D-Bild.

Umgekehrte &
Seitenverkehrte
Projektion

CV2: Bildaufnahme

Das menschliche Auge

Das visuelle System:

Linse	Fokussierung des Abbildes auf die Membran
Netzhaut	besteht aus Fotorezeptoren
Stäbchen	Schwarz-Weiß-Sehen
Zäpfchen	Farbwahrnehmung (R G B)

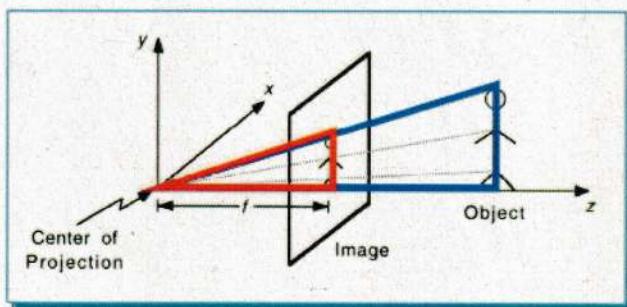
25% 50% 25%

Perspektivische Projektion

in einer Lochkamera („zentrale Projektion“)
Fortsetzung: Camera Obscura

Es werden folgende Annahmen getroffen:

- (1) Das Zentrum der Projektion stimmt mit dem Ursprung des Kamerakoordinatensystems überein.
- (2) Die Kameraachse (optische Achse) ist an der z-Achse der Kamera ausgerichtet.
- (3) Das Bild wird auf einer imaginären Bildebene vor dem Projektionszentrum erzeugt, sodass es Bild nicht mehr auf dem Kopf steht.



0 Zentrum der Projektion

f fokale Länge, stellt Brennweite dar

Bei digitalen Kameras wäre die fokale Länge f die Entfernung der Linse zum Kamerاسensor.

x x-Größe der Bildebene

X x-Größe des Objektes

y y-Größe der Bildebene

Y y-Größe des Objektes

f die Entfernung der Bildebene zu 0

Z die Entfernung des Objektes zu 0

Es gelten folgende Dreiecksbeziehungen:

$$\frac{y}{Y} = \frac{f}{Z} \quad x = \frac{f}{Z} X \quad y = \frac{f}{Z} Y \quad \frac{x}{X} = \frac{f}{Z}$$

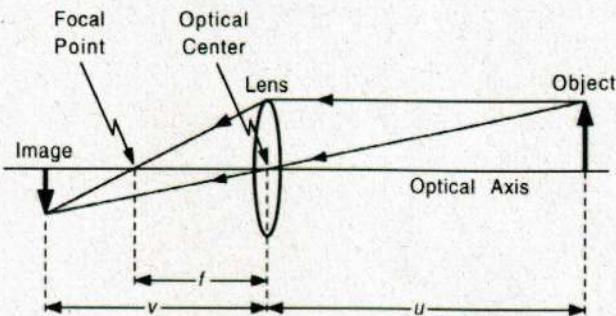
Linsen

Die Lochkamera ist idealisiert - Die Lochblende ist unendlich klein und lässt beliebig wenig Licht durch aber die Belichtungszeit wäre sehr lange.

In Wirklichkeit braucht man optische Linsen.

Idealisierte Linse: Unendlich dünn.

Jeder Lichtstrahl wird an einer virtuellen Ebene in der Linsenmitte gebrochen → Parallelle Strahlen werden im Brennpunkt / Focal Point mit Abstand f zur Linse an der Optical Axis gebrochen.



Die einfache Linsengleichung bestimmt das Verhältnis zwischen u (Abstand von Objekt und Linse) und v (Abstand von Bildebene und Linse) sodass Objekte im Brennpunkt / Focal Point scharf abgebildet werden:

$$1/u + 1/v = 1/f$$

Tiefenschärfe Bereich (DOF: Depth of Field)

In der Praxis werden Objekte scharf abgebildet die im richtigen Tiefenschärfenbereich sind. DOF ist der Abstand der am nächsten und am weitesten Entfernten Objekte im Bild.

Der Tiefenschärfe Bereich wird bestimmt durch:

1. Sensorauflösung
2. Blendenöffnung

Die Blende ist eine ringförmige Öffnung vor der Linse, welche den Kegelwinkel des Strahlenbündels bestimmt, welches auf die Bildebene trifft.

Je kleiner die Blende, desto weniger Strahlen treffen die Bildebene und desto dunkler ist das Bild für eine gegebene Belichtungszeit, aber desto größer ist der DOF.

3. Linsengröße

Je größer die Linse, desto mehr Strahlen eines einzigen Objektpunktes gelangen auf die Bildebene und durch den falschen Abstand zur Linse fallen die Strahlen nicht mehr zu einem Punkt zusammen.

Radiometrie

Radiometrie ist die Wissenschaft von der Messung elektromagnetischer Strahlung.

Im Gegensatz zu Photometrie „Messung des Lichts“ auch Bereiche des Infraroten und Ultravioletten, aber auch der Gammastrahlen.

Man misst Strahlen aus allen Frequenzen.

Sampling

Unter Sampling versteht man den Umwandlungsprozess von einem Signal zu einer numerischen Folge.

Die Bildauflösung ist ein Sammelbegriff für den Detailreichtum eines Bildes und wird von der radiometrischen-, Sensor-, räumlichen- und zeitlichen Auflösung bestimmt.

Radiometrische Auflösung

bestimmt, wie fein ein System Helligkeitsunterschiede unterscheidet

Sensor Auflösung

steht für die Anzahl der Bildelemente (Picture-elements = Pixel) in digitalen Bildern

Räumliche Auflösung

Sie wird durch die minimale Entfernung zwischen zwei benachbarten Merkmalen oder die minimale Größe eines Merkmals definiert

Zeitliche Auflösung

Frames per second

Bildsensoren

Ein Bildsensor ist ein Bauelement, das ein optisches Bild in ein elektronisches Signal umwandelt.

Es sind zwei Arten von Pixelsensoren in Verwendung:

Charge-Coupled Device (CCD)

Complementary Metal-Oxide-Semiconductor (CMOS)

Farbe auf CCD-Sensoren

Digitale Farbkameras können Farbe auf drei verschiedenen Wegen erfassen:

Field Sequential Technik

mithilfe von Rot-, Grün- und Blau-Filtern drei Schwarz-Weiß Bilder nacheinander aufnehmen (Es können Fehler entstehen weil die Bilder zu unterschiedlichen Zeitpunkten gemacht werden).

Wird in Mikroskopen benutzt.

3 Chip

Durch die Verwendung eines strahlenbrechenden Prismas kann eine korrekte Farbtrennung erzielt werden, da das Eingangssignal in einen roten, grünen und blauen Anteil aufgespalten wird.

Wird in Camcordern benutzt.

Color Filter Array (CFA) oder Color Filter Mosaic (CFM)

ein Mosaik aus winzigen Farbfiltern, die über jedes Pixel eines Bildsensors platziert werden, um Farbinformation aufzunehmen.

Bayer Filter:

Hat unterschiedliche RGB-Anteile im Mosaic.

Der rohe (engl. raw) Output einer Bayer-Filter Kamera wird als Bayer Muster Bild (engl. Bayer pattern image) bezeichnet.

Jedes Pixel nimmt nur eine der drei Farben auf, somit muss die tatsächlich Farbe des Pixels, mit Hilfe von Interpolationsalgorithmen berechnet werden, um die fehlenden Rot, Grün und Blauanteile je Farbkanal für jeden fehlenden Punkt aus dessen Nachbarn zu berechnen (engl. demosaicing).

Ideale vs. Reale Kamera

Einige Störungen:

optische- / chromatische- / sphärische- Aberration

Linsen Verzerrung

Weißabgleich

GRGB / RG BG

50% Grün

25% Rot & Blau

An der menschl.
Physiologie angepasst

CV3: Bildkodierung und Kompression

Digitale Bildformate

Wenn eine Zahl unendlich viele mögliche Werte annehmen kann, sprechen wir von kontinuierlichen oder analogen Daten. Computer können mit analogen Daten nicht rechnen, die Daten müssen daher in eine für den Computer bearbeitbare Version umgewandelt werden. Diesen Umwandlungsprozess bezeichnet man als Digitalisierung, der beim Scannen (Abtasten) von Fotos oder eines dreidimensionalen Raumes beim Fotografieren mit digitalen Kameras erfolgt.

Raster Bildformat

Ein Rasterbild ist ein Raster von diskreten Werten (jeder Bildpunkt eines Bildrasters muss mit seinem Helligkeitswert bzw. Farbwert abgespeichert werden). Rasterbilder beinhalten eine endliche Menge an digitalen Werten, die Bildelemente oder Pixel genannt werden, welche die kleinsten, individuellen Elemente in einem Bild darstellen.

Durch die Verwendung des Raw-Bildformates ermöglichen uns digitale Kameras den Zugang zu den tatsächlich von der Kamera aufgenommenen Bilddaten.

Dieses Format speichert demnach für jeden Pixel den entsprechenden Farbwert ohne Nachbearbeitung (bei 1-Chip Kameras Rot-, Grün- und Blauwerte entsprechend dem CFA-Muster). Dieses RAW Format stellt somit kein „richtiges“ Farbbild dar, es muss zur farbigen Anzeige umgewandelt werden.

Beispiele: BMP Bitmap, PNG Portable network graphics

Vektor Bildformat

Bei einem Vektorbild wird der Bildinhalt in Form von geometrischen Objekten repräsentiert und die Rasterung erfolgt erst bei der Darstellung auf einem konkreten Endgerät (z.B. einem Display oder Drucker).

Beispiele: CGM Computer Graphics Metafile, WMF Windows Metafile

Bildkompression

Bildkompression kann verlustfrei (engl. lossless) oder verlustbehaftet (engl. lossy) sein.

Bilder werden lossy komprimiert aber sie sind für den Menschen dennoch „visually lossless“.

Verlustfreie Kompression

Run Length Encoding (RLE)

Dabei werden Sequenzen von Datenwerten, in der ein gleicher Datenwert in vielen aufeinanderfolgenden Datenelementen vorkommt (engl. runs), als einzelner Datenwert und dessen Anzahl an Wiederholungen abgespeichert.

Huffman Encoding

Entropiedecodierung, kodiert Dinge die am öftesten vorkommen mit kleineren Zeichen zB „00“ und Dinge die seltener vorkommen mit größeren Zeichen zB „101“

Lempel Ziv (LZ)

Bildformate, die verlustfreie Kompression verwenden: z.B. GIF, PNG und TIFF

Verlustbehaftete Kompression

„Joint Photographic Experts Group“ (JPEG)

Vorgang besteht aus mehreren Schritten und sie nutzt die Schwächen des Auges.

1. Farbraumkonversion und Downsampling

Das Ausgangsbild wird von RGB in den YCbCr-Farbraum umgerechnet.

Dieser Schritt ist durch die geringere Empfindlichkeit des Auges gegenüber feinen Farbdetails im Gegensatz zu feinen Helligkeitsunterschieden begründet.

2. Kosinustransformation und Quantisierung im Spektralraum

Diskrete Cosinus Transformation (DCT) für 8x8 Blöcke

3. Verlustfreie Kompression

JPEG Kompression kann ein Bild zu einem Fünftel seiner Originalgröße reduzieren (ohne visuelle Verschlechterung). Je höher die Kompression (Quantisierung), desto mehr Artefakte treten auf (Blockformation). Das JPEG-Verfahren kombiniert also mehrere verschiedene, sich ergänzende Kompressionsmethoden.

Diskrete Kosinus Transformation (DCT)
Variation der Fourier-Transformation

Videokompression

Die Mehrheit der Algorithmen verwendet verlustbehaftete Kompression, bei der große Mengen an Daten entfernt werden, der Unterschied aber visuell kaum erkennbar ist.

MPEG (Moving Picture Experts Group)

Interframe-Kompression

Analysiert Veränderung in einzelnen Frames und komprimiert Frames einzeln.

CV4: Punktoperationen

Punktoperationen

Jeder neue Pixelwert ist ausschließlich vom ursprünglichen Pixelwert an derselben Position abhängig.

Inhomogene Punktoperatoren

Abhängig von Pixelkoordinaten

Homogene Punktoperatoren

Unabhängig von Pixelkoordinaten, (für jede Bildposition gleich)

Affine Punktoperatoren

Abhängig vom Wert des zu ändernden Pixels

Die Abbildungsfunktion

(engl. mapping function) Jedem Pixelwert im Eingangsbild wird entsprechend der Abbildungsfunktion ein Helligkeitswert im Ausgangsbild zugewiesen.

Identitätsfunktion / Invertierung

Die Abbildungsfunktion invertiert Werte oder passt sie einem erlaubten Wertebereich an.

Schwellwertoperation

Teilt Pixel einen bestimmten Wert zu wenn sie Schwellwert (threshold value) überschreiten (1) oder unterschreiten (0)

Clipping

Wenn man zB Kontrast und Helligkeit zu manipuliert können Pixelwerte die Zahl 255 überschreiten oder 0 unterschreiten wodurch Clipping notwendig ist.

Histogramme

sind Bildstatistiken.

Statistisch gesehen sind Histogramme Häufigkeitsverteilungen, daher beschreiben Histogramme von Bildern die Häufigkeit der einzelnen Intensitätswerte im Bild.

Die x-Achse beschreibt in einem Schwarz-Weiß-Bild alle Grauwerte von 0 bis 255.

Farbhistogramme werden in 3 Channels aufgeteilt.

Erhöhung der Helligkeit --> Histogramm rückt nach Rechts

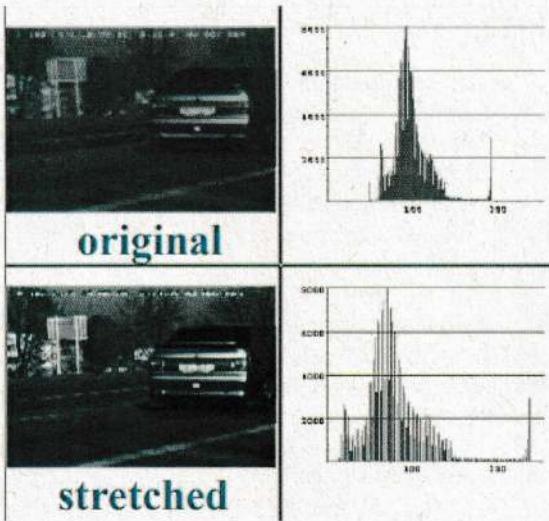
Kontrasterhöhung --> Histogramm wird breiter

Histogramm-Normalisierung / -Spreizung --> „Autokontrast“

= Kontrasterhöhung sodass der gesamte Wertebereich abgedeckt wird

Dazu wird das aktuell dunkelste Pixel auf den niedrigsten und das aktuell hellste Pixel auf den höchsten Intensitätswert abgebildet und alle dazwischenliegenden Pixelwerte linear verteilt

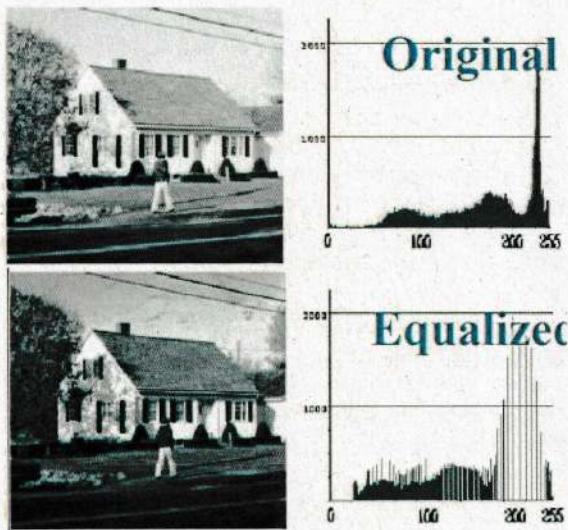
Alternative Verteilungsmethoden: Aufteilung nach Quantillen etc.



Histogrammequalisierung / Histogrammausgleich

Im Gegensatz zur Normalisierung keine affine Punktoperation
Da Spitzen im Histogramm nicht entfernt werden können ist eine echte Gleichverteilung nicht möglich.

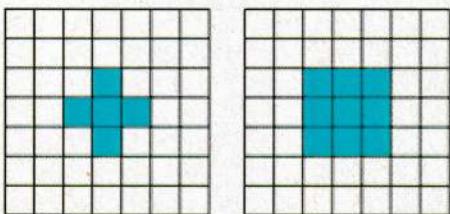
Basierend auf der rel. Häufigkeit werden Lücken verteilt (nicht linear).



CV5: Lokale Operationen

Nachbarschaften

Vierer und Achter Nachbarschaft um Pixel herum.



lokale Operationen / Filter

Wikipedia: „In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image.“

Lineare Filter

Pixelwerte innerhalb der Filterregion in linearer Form verknüpfen, d.h. durch eine gewichtete Summation verknüpfen. Ein einfaches Beispiel dazu ist die lokale Mittelwertbildung bei der alle neun Pixel der 3×3 -Filterregion mit der Gewichtung 1/9 summiert werden.

$$I'(u, v) = \frac{1}{9} \sum_{j=-1}^1 \sum_{i=-1}^1 I(u+i, v+j)$$

Bei linearen Filtern werden die Größe und Form der Filterregion, wie auch die zugehörigen Gewichte, durch eine Matrix von Filterkoeffizienten spezifiziert, der sogenannten „Filtermatrix“ oder „Filtermaske“ --> unterschiedliche Nachbaren können unterschiedliche Multiplikatoren erhalten aus der Filtermaske

$$I'(u, v) = \sum_{j=-1}^1 \sum_{i=-1}^1 I(u+i, v+j) \cdot F(i, j)$$

Beispiele für lineare Filter:

Tief- / Hoch-Passfilter

Tiefpassfilter sind solche, die hohe Frequenzen herausfiltern und niedrige Frequenzen passieren lassen. → Rauschunterdrückung bzw. als Glättungsoperatoren.

Beispielsweise:

Mittelwertfilter

Berechnet Mittelwert, nennt man auch Box-Filter wenn alle Gewichte 1 sind. Ist aber nicht sinnvoll da das Zentrum stärker bewertet werden sollte als die Ränder.

Gaußfilter

Wie Gaußfunktion, in der Mitte am stärksten gewichtet.

Standardabweichung σ bestimmt die Glättungs-Intensität (nicht die Größe der Maske / Nachbarschaft).

Umgekehrt filtern Hochpassfilter tiefe Frequenzen heraus und lassen hohe Frequenzen passieren. → Detektion von Kanten

Differenzfilter

Wenn manche Gewichte positiv und andere negativ sind ist die Summe dann quasi die Differenz der positiven und negativen Anteile.

Beispiel: Gewichtung eines Laplace-Filters (in einer Vierernachbarschaft)

$$F_{\text{Laplace}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Ziel ist es nicht verschiedene Intensitäten zu glätten, sondern zu verstärken. Sie sind deshalb Hochpassfilter und eignen sich dazu Kanten und Konturen hervorzuheben.

Danach muss immer auf 0..255 normiert werden.

Mehr über Hochpassfilter

Hochpassfilter basieren auf der ersten oder zweiten Ableitung der Bildfunktion (weil sich max und min in der Ableitung zeigen) in unterschiedlichen Richtungen mit unterschiedlichen Gewichtungen.

Filter die auf der ersten Ableitung beruhen: Gradienten-Filter

Filter die auf der zweiten Ableitung beruhen (also Gradient vom Gradienten): Laplace-Filter

Umgang mit dem Bildrändern

Bildränder haben keine Werte

Möglichkeit: Auch die Randbereiche berechnen unter der Annahme, dass:

- die Pixel außerhalb des Bilds einen konstanten Wert (z.B. „schwarz“ oder „grau“) aufweisen. → ungünstig
- sich die Randpixel außerhalb des Bilds fortsetzen. (also wenn Rand rot ist dann noch 5 weitere Pixel rot) → geringe Verfälschungen
- sich das Bild in beiden Richtungen (horizontal und vertikal) zyklisch wiederholt (zB das Bild ist an den Rändern gespiegelt)

Formale Eigenschaften linearer Filter

Mathematischer Ursprung: „Lineare Faltung“ (engl. linear convolution)

verknüpft zwei Funktionen gleicher Dimensionalität, kontinuierlich oder diskret. Für diskrete, zweidimensionale Funktionen I und F ist die Faltungsoperation definiert als:

$$I'(u, v) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} I(u-i, v-j) \cdot F(i, j)$$

Man bezeichnet die Faltung auch als „Black-Box“-Operation.

Nicht-lineare Filter

Lineare Filter verwischen das Bild (Ecken und Kanten) und können durch Gewichten und Addieren charakterisiert werden.

Beispiele für Nicht-Lineare Filter

Rangordnungsfilter (englisch: rank value filter)

werden charakterisiert durch die Kombination benachbarter Pixel durch das Vergleichen und Selektieren.

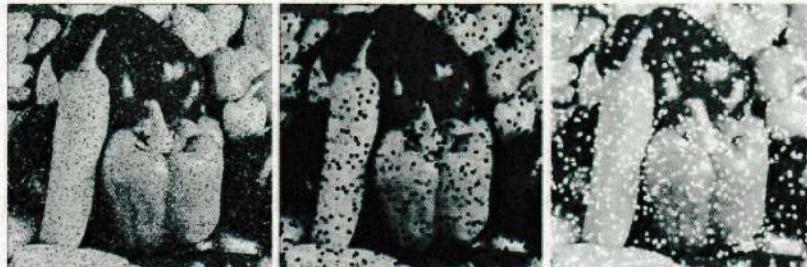
Sie sortieren beispielsweise Grauwerte innerhalb der Filtermaske in aufsteigender Reihenfolge. Sie unterscheiden sich nur durch die Position in der Liste, von welcher der Grauwert genommen und in das

zentrale Pixel zurückgeschrieben wird. Diejenige Filteroperation, die den Mittelwert selektiert, wird Medianfilter genannt, die Filter, die das Minimum bzw. das Maximum selektieren, werden als Minimum- bzw. Maximumfilter bezeichnet

sind folgendermaßen definiert:

$$I^-(u, v) = \min \{I(u+i, v+j) \mid (i, j) \in R\} = \min(R_{u,v})$$

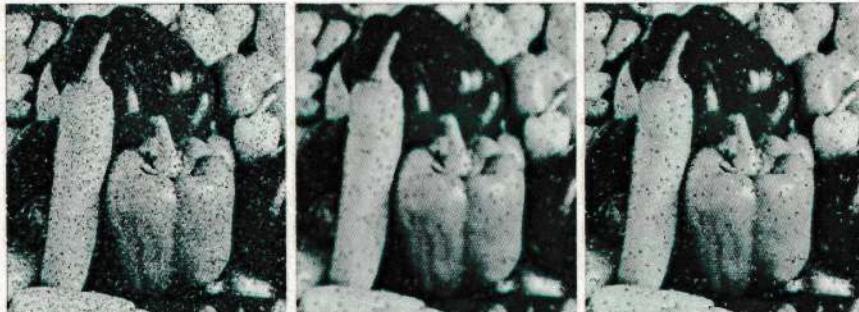
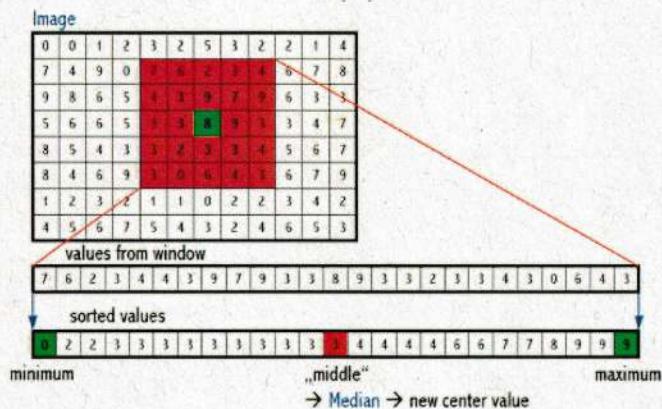
$$I^+(u, v) = \max \{I(u+i, v+j) \mid (i, j) \in R\} = \max(R_{u,v})$$



$(R_{u,v}) \rightarrow$ Region der Bildwerte die an der aktuellen Position (u, v) von der Filterregion überdeckt werden

Der Medianfilter ersetzt jedes Bildelement durch den Median der Pixelwerte innerhalb der Filterregion R , also:

$$I^-(u, v) = \text{median}(R_{u,v})$$



Links: Bild mit zufällig gesetzten weißen und schwarzen Punkten (pepper and salt)
 --> soll verbessert werden

Mitte: Mittelwertfilter

Rechts: Medianfilter

CV6: Kantenfilterung

Bildschärfung

Die übliche Methode des Schärfens ist das Anheben der hochfrequenten Bildanteile, die für den Schärfeeindruck des Bildes verantwortlich sind.

Für die Bildschärfung im zweidimensionalen Fall verwendet man die zweiten Ableitungen in horizontaler und vertikaler Richtung in Form des so genannten Laplace-Operators.

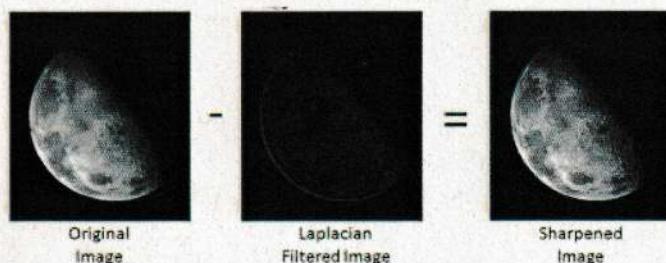
Der Laplace-Operator ∇^2 einer zweidimensionalen Funktion $f(x,y)$ ist definiert als Summe der zweiten partiellen Ableitungen in x- und y-Richtung, d.h.:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2}(x,y) + \frac{\partial^2 f}{\partial y^2}(x,y) \quad \text{wobei} \quad \frac{\partial^2 f}{\partial x^2}(x,y) = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

Für die Schärfung wird das Bild I mit dem Laplace-Filter HL gefiltert und anschließend wird das Ergebnis vom ursprünglichen Bild subtrahiert:

$$I' = I - w \cdot (HL * I)$$

Der Faktor w bestimmt dabei den Anteil der Laplace-Komponente und damit die Stärke der Schärfung.



Kanten

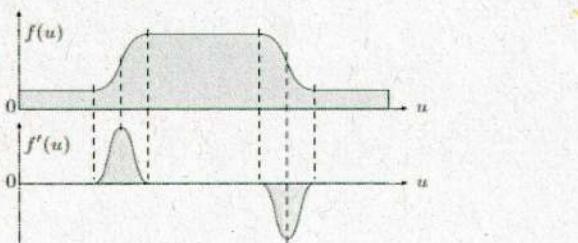
In Bildern bezeichnen Kanten Diskontinuitäten der Intensität oder Farbe.

Die subjektive „Schärfe“ eines Bilds steht in direktem Zusammenhang mit der Ausgeprägtheit der darin enthaltenen Diskontinuitäten und der Deutlichkeit seiner Strukturen.

Kanten kann man grob als jene Orte im Bild beschreiben, an denen sich die Intensität auf kleinem Raum und entlang einer ausgeprägten Richtung stark ändert. Je stärker sich die Intensität ändert, umso stärker ist auch der Hinweis auf eine Kante an der entsprechenden Stelle. Die Stärke der Änderung bezogen auf die Distanz ist aber nichts anderes als die erste Ableitung.

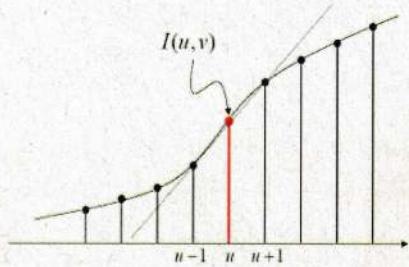
Gradientenbasierte Kantendetektion

Beispiel: Bild in einer Dimension



so ergibt sich in der Ableitung ein positiver Ausschlag überall dort, wo die Intensität ansteigt, und ein negativer Ausschlag, wo der Wert der Funktion abnimmt.

Bei einer diskreten Funktion kann der Anstieg der Tangente an der Stelle u dadurch geschätzt werden, dass eine Gerade durch die Abtastwerte an den benachbarten Stellen $u-1$ und $u+1$ gelegt und deren Anstieg berechnet wird:



Der gleiche Vorgang wird auch in der vertikalen Richtung, also entlang der x und y Achse durchgeführt (partiellen Ableitung)

z.B. für die partiellen Ableitungen der Bildfunktion $I(u,v)$ entlang der u - bzw. v -Koordinate.

$$\frac{\partial I}{\partial u}(u,v), \quad \frac{\partial I}{\partial v}(u,v) \quad \nabla I = \begin{pmatrix} \frac{\partial I}{\partial u} \\ \frac{\partial I}{\partial v} \end{pmatrix} \quad |\nabla I| = \sqrt{\left(\frac{\partial I}{\partial u}\right)^2 + \left(\frac{\partial I}{\partial v}\right)^2}$$

Den Vektor ∇I aus beiden partiellen Ableitungen bezeichnet man als Gradient (= Steigung) der Funktion I an der Stelle (u,v) .

Die Filterantwort ist richtungsabhängig (durch den Betrag) aber:

- der horizontale Gradientenfilter reagiert am stärksten auf Änderungen der Helligkeit in horizontaler Richtung
- der vertikale Gradientenfilter auf vertikale Kanten

In flachen Bildregionen (Regionen mit konstanten Intensitätswerten) ist die Filterantwort null, da sich dort die Ergebnisse des positiven und des negativen Filterkoeffizienten jeweils aufheben.

Kantendetektionsfilter

Gute Kantendetektoren sind Richtungsunabhängig „isotrop“ (wie zB der Betrag des Gradientenfilters / Laplace-Operators).

Sobel und Prewitt-Operator

sind aber richtungsabhängig und filtern in u - oder v -Richtung und unterscheiden sich geringfügig.

Bei beiden erfolgt eine einfache (Box-) Glättung über drei Zeilen bzw. drei Spalten, bevor der Gradient berechnet wird.

Prewitt-Operator:

$$H_x^P = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad H_y^P = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Sobel-Operator:

$$\Phi(u,v) = \tan^{-1} \left(\frac{D_y(u,v)}{D_x(u,v)} \right)$$

Die Filter des Sobel-Operators sind fast identisch, geben allerdings durch eine doppelt so hohe Glättung mehr Gewicht auf die zentrale Zeile bzw. Spalte.

- Vor allem der Sobel-Operator ist aufgrund seiner guten Ergebnisse und seiner Einfachheit weit verbreitet.

Roberts-Operator

Der älteste Kantenoperator. Benutzt die kleinste Filtergröße (2×2), um die Richtungsableitungen entlang der beiden Diagonalen zu schätzen (reagiert stärker auf Diagonale).

$$H_1^R = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad H_2^R = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Der Roberts-Operator ist kantenunabhängiger als Sobel.

Alternative:

nicht nur ein Paar „breite“ Filtern für zwei (orthogonale) Richtungen, sondern mehrere „engere“ Filter für mehrere Richtungen wie beim Kirsch-Operator.

Kirsch-Operator („Kompass-Operator“)

acht verschiedene Richtungen im Abstand von 45°

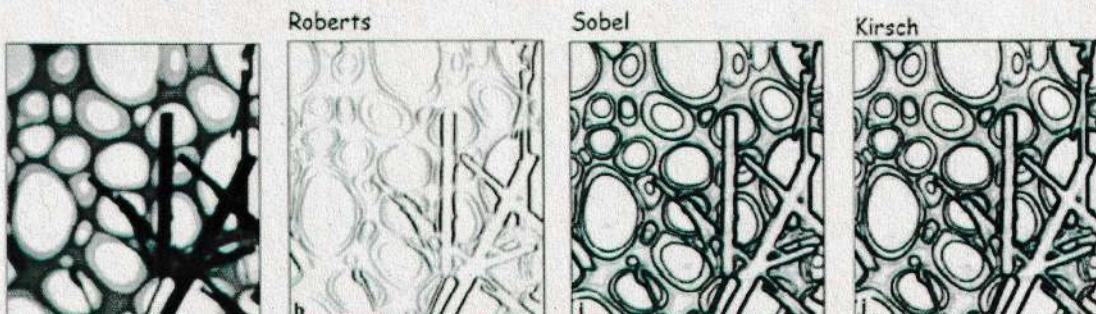
$$H_1^K = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad H_2^K = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad H_3^K = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad H_4^K = \begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{pmatrix} \quad H_5^K = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad H_6^K = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix} \quad H_7^K = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad H_8^K = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}$$

Die Kantenstärke E^K an der Stelle (u, v) ist als Maximum der einzelnen Filterergebnisse definiert:

$$E^K(u, v) = \max(D1(u, v), D2(u, v), \dots, D8(u, v))$$

und der am stärksten ansprechende Filter bestimmt auch die zugehörige Kantenrichtung.

Derartige Kompass-Operatoren bieten allerdings kaum Vorteile gegenüber einfacheren Operatoren, wie z.B. dem Sobel-Operator.



Das Design eines guten Kantenfilters ist ein Kompromiss: Je besser ein Filter auf „kantenartige“ Bildstrukturen reagiert, desto stärker ist auch seine Richtungsabhängigkeit (umso enger ist der Winkelbereich, auf den das Filter anspricht).

Der Ablauf der Kantendetektion

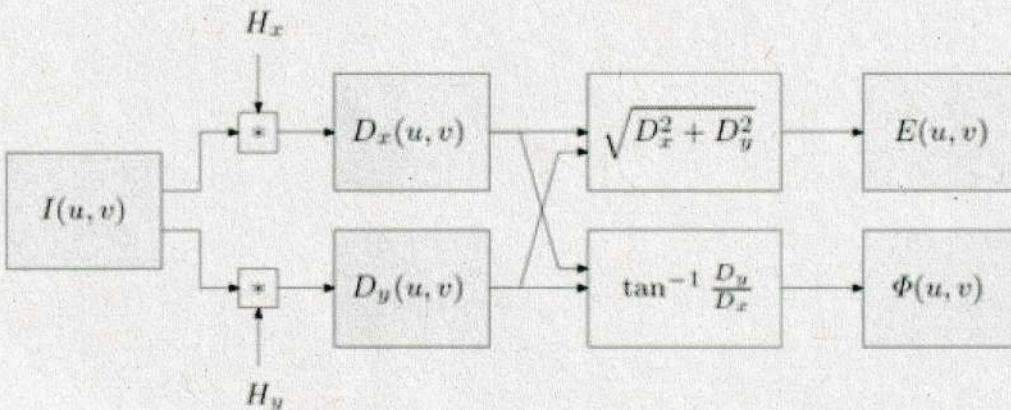
Man kann nicht nur die Stärke (den Betrag) eines Kantenpunkts berechnen, sondern auch die lokale Richtung der zugehörigen Kante, da beide Informationen im Gradienten enthalten sind.

Die Kantenstärke (Gradientenbetrag): $E(u,v)$

$$E(u,v) = \sqrt{D_x(u,v)^2 + D_y(u,v)^2}$$

Die Kantenrichtung Φ (Gradientenvektor): df/du (u)

Zuerst wird das ursprüngliche Bild I mit den beiden Gradientenfiltern H_x und H_y gefiltert und nachfolgend aus den Filterergebnissen die Kantenstärke E und die Kantenrichtung Φ berechnet.



Kantendetektion mit der zweiten Ableitung

Ein Max / Min ist in der zweiten Ableitung eine Nullstelle.

Wenn man die zweite Ableitung nimmt dann ist man unabhängig vom Wert des Maximums oder Minimums (also bildet nicht die Ableitung ab) und bildet mit 0/1 die Nullstellen ab.

Die detektierten Kanten sind dann im Ausgangsbild nur 1 Pixel breit.

Laplacian-of-Gaussian-Operator (LoG)

Der Laplace-Operator arbeitet mit der zweiten Ableitung.

Der Laplace-Operator ist Richtungsunabhängig aber rauschanfällig weshalb das Bild vor der Anwendung immer geglättet werden sollte:

Kombiniert Gauss und Laplace mit einem gemeinsamen linearen Filter.

Canny-Operator

Durch verschiedenen Auflösungen desselben Bildes „Multi-Resolution“ / „Multi-Scale“ - Techniken erhält man unterschiedliche Resultate.

Der wichtigste Vertreter für ein derartiges Verfahren ist der Kantendetektor von Canny, der einen Satz von gerichteten (und relativ großen) Filtern auf mehreren Auflösungsebenen verwendet und deren Ergebnisse in ein gemeinsames Kantenbild („edge map“) zusammenfügt.

CV7: Globale Operationen

Bisher haben wir uns mit Punkt- und lokalen Operationen befasst, Ausgangspunkt war entweder ein Pixel oder die Umgebung eines Pixels inklusive des betrachteten Pixels.

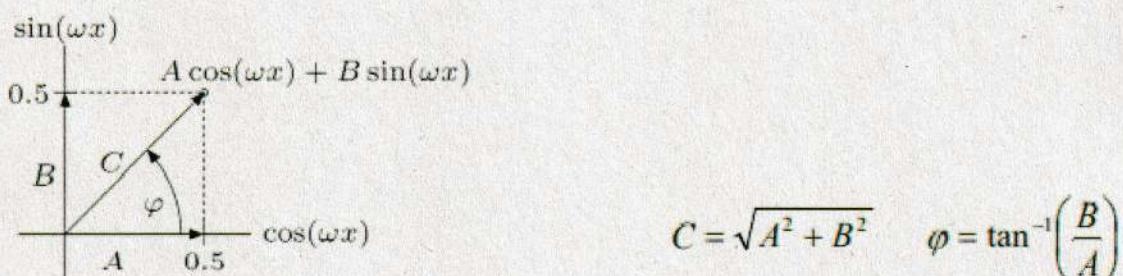
Globale Operationen verwenden hingegen das gesamte Bild als Ausgangsbasis.

Das gesamte Bild wird aus dem Ortsraum (Bildraum) in einen anderen Raum (z.B. Frequenzraum) transformiert, da dort manche Eigenschaften in Bildern besser sichtbar sind. So kann man dort z.B. Daten besser dekorrelieren.

Fouriertransformation

Zerlegung von Bildsignalen in Sinus- und Kosinusfunktionen, mithilfe der Fouriertransformation.

Insbesondere entsteht durch die Addition einer Kosinus- und Sinusfunktion mit identischer Frequenz ω und Amplituden A bzw. B eine weitere sinusoidale Funktion mit derselben Frequenz ω , wobei die resultierende Amplitude C und der Phasenwinkel φ ausschließlich durch die beiden Amplituden A und B bestimmt sind:



So kann man jede beliebige periodische Funktion als Kombination von sin und cos darstellen.

$$g(x) = \sum_{k=0}^{\infty} [A_k \cos(k\omega_0 x) + B_k \sin(k\omega_0 x)]$$

Dies bezeichnet man als Fourierreihe, die Gewichte A_k , B_k als Fourierreffizienten der Funktion $g(x)$. Diese Gewichte könnten mit der Fourieranalyse berechnet werden.

$$A_\omega = A(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cos(\omega x) dx \quad B_\omega = B(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \sin(\omega x) dx$$

Die Koeffizientenfunktionen $A(\omega)$ und $B(\omega)$ sind kontinuierlich, sie enthalten das Spektrum der im ursprünglichen Signal enthaltenen Frequenzkomponenten. Das Fourierintegral beschreibt die Funktion $g(x)$ als Summe unendlich vieler Kosinus-/Sinusfunktionen mit kontinuierlichen (positiven) Frequenzwerten $\rightarrow G(\omega)$

Der Übergang von der Funktion $g(x)$ zu ihrem Fourierspektrum $G(\omega)$ bezeichnet man als Fouriertransformation. Umgekehrt kann die ursprüngliche Funktion $g(x)$ aus dem Fourierspektrum $G(\omega)$ durch die inverse Fouriertransformation berechnet werden.

Ortsraum und Frequenzraum sind zueinander "duale" Darstellungsformen, die sich nicht unterscheiden.

Diskrete Fourier-Transformation (DFT)

Bei Bildern liegen immer diskrete Daten vor und daher wird eine Version der Fouriertransformation benötigt, in der sowohl das Signal wie auch das zugehörige Spektrum als endliche Vektoren dargestellt werden - die DFT.

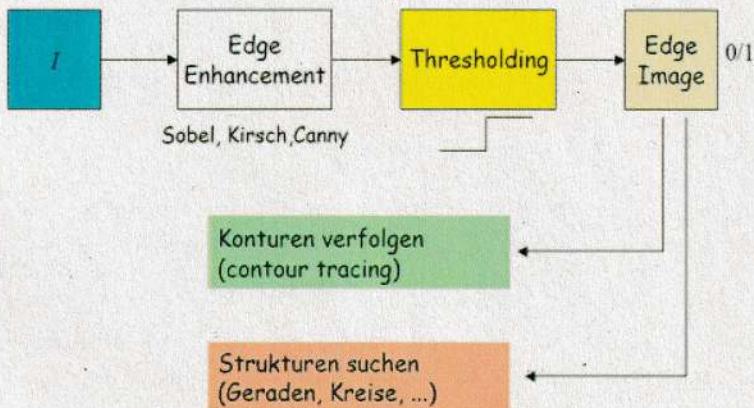
Es werden Abtastwerte benötigt.

Zur praktischen Berechnung der DFT existieren schnelle Algorithmen.

Hough-Transformation

Bildkanten im kontinuierlichen Kantenbild werden zuerst auf einem „Edge Image“ (binär) dargestellt indem nur Punkte gewählt werden die einen Schwellwert überschreiten.
Anschließend werden Bildstrukturen im „Edge Image“ gesucht.

Die Hough-Transformation untersucht Verteilungen von Punkten und sucht nach Konturen / Strukturen.



Hough Transformation ist ein allgemeiner Ansatz, um beliebige, parametrisierbare Formen in Punktverteilungen zu lokalisieren. Zum Beispiel können geometrische Formen wie Geraden, Kreise und Ellipsen mit wenigen Parametern beschrieben werden.

Die Detektion von Geraden in binären Kantenbildern ist eine häufige Anwendung. Eine Gerade in 2D kann mit zwei reell-wertigen Parametern beschrieben werden, z. B. in der klassischen Form $y = kx + d$.

Ziel ist es, jene Geradenparameter k und d zu finden, auf denen möglichst viele Kantenpunkte liegen.

Dafür ermittelt die HT alle möglichen Geraden die durch einen einzelnen Bildpunkt laufen könnten und ermittelt die Geraden die sich mit den meisten Punkten schneiden.

Dazu verwendet die Hessesche Normalform: $r = x \cos(\theta) + y \sin(\theta)$

Es gibt an jedem Punkt unendlich viele Geraden im sogenannten „Hough-Raum“ wobei die Parameter r und θ variieren.

Jede einzelne Gerade dieser Menge der durch die Koordinaten r , θ aufgespannt wird, wird durch einen einzigen Punkt repräsentiert, der genau den Parametern der betrachteten Geraden entspricht. (im kartesischen System)

Hough-Tracking für Linien

Vom Input I (Eingangsbild) wird ein Kantenbild mit Richtung und Betrag berechnet (zB mit dem Sobelfilter).

Danach wird für die Hough-Transformation ein zweidimensionales Akkumulator-Array erzeugt. Dazu wird der (r, θ) -Raum diskretisiert (z.B. r in Pixel, θ in 1° oder 5° Stufen), sodass θ in der x-Richtung und r in y-Richtung eingetragen werden kann.

Das daraus entstehende endliche, zweidimensionale Feld (r, θ) wird als Akkumulator bezeichnet, jedes Element im Akkumulator entspricht einer Geraden im (x, y) -Koordinatensystem.

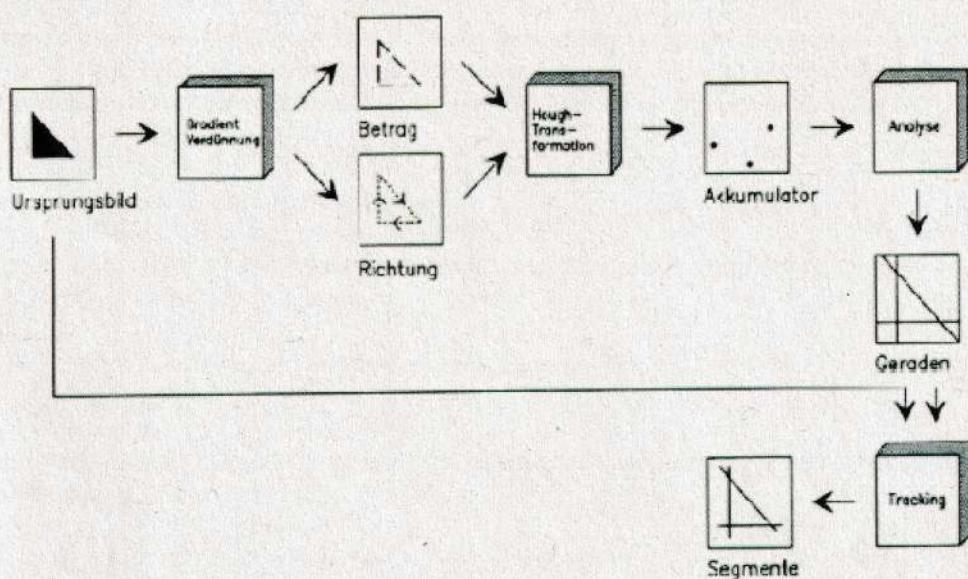
Der Akkumulator wird zunächst mit Null initialisiert, anschließend werden alle Kantenpunkte mit der Hough-Transformation in den (r, θ) -Raum überführt und in den Akkumulator eingetragen (die betroffenen Zellen werden inkrementiert).

--> Alle Pixel beeinflussen die Einträge im (r, θ) -Raum den wir suchen die Parameter der Gerade welche sich mit den meisten Punkten schneidet.

Danach filtert man alle Punkte im (r, θ) -Raum aus die einen Schwellwert überschreiten.

Das Ergebnis der Hough-Raum-Analyse ist ein Bild mit Geraden ohne Anfangs- und Endpunkte. Um deren Anfangs- und Endpunkte zu finden, wird ein sogenanntes Tracking durchgeführt. Dabei dienen die durch die Hough-Transformation gefundenen Geraden als "Wegweiser" durch das Originalbild.

Man fährt auf den Geraden das Originalbild mit einem Fenster ab und registriert wann die Grauwertdifferenz (des Bereichs rechts und links von der Geraden) im Fenster einen Schwellwert überschreitet. An diesen Stellen befindet sich das Fenster zu einer hohen Wahrscheinlichkeit auf einer Objektkontur. Der Vorteil bei der Verwendung von Betrag und Richtung von Kantenpixeln ist, dass man die Richtung der Kante im Originalbild schon kennt.



Weitere Hough-Transformationen

Prinzipiell kann man jede parametrisierbare Struktur mit Hilfe der Hough Transformation finden, die häufigsten sind Kreise und Ellipsen. Die Kreiserkennung funktioniert ähnlich der Geradenerkennung.

Die Hough-Räume für komplexe Formen haben mehrere Dimensionen.

CV8: Bildmerkmale - Interest Points

Bildmerkmale

Bildmerkmale (engl. image features) sind mathematische Beschreibungen eines Bildes oder Bildteiles, die eine leichter unterscheidbare Repräsentation als reine Pixelwerte liefern und helfen, relevante Information für gewisse Aufgaben zu extrahieren

Globale Bildmerkmale

Wie zB Grauwerthistogramm - Daten schwer unterscheidbar, zu allgemein

Lokale Bildmerkmale

Beschreiben kleine interessante (hoher visueller Informationsgrad) Bildregionen, um diese zu erkennen:

1. Merkmalsdetektion: Räumliche Bestimmung, auch die räumliche Ausdehnung oder Skalierung. Die erkannten Regionen werden als Interest Points oder Keypoints bezeichnet.
2. Merkmalsbeschreibung: Es wird ein Merkmalsvektor berechnet, der die lokale Bildstruktur beschreibt. Die Menge aller möglichen Merkmalsvektoren bildet einen Merkmalsraum (engl. feature space).

Kantenerkennung

Kanten: starke Helligkeitsveränderungen / Unstetigkeiten

Ein Kantendetektor reduziert die Menge an zu verarbeitenden Daten.

Allerdings sind Kanten zur Erfassung der relevanten Information nur beschränkt geeignet, da sie weder rotations- noch skalierungs invariant sind, gleiche Merkmale in verschiedenen Bildern können somit nur schwer lokalisiert werden.

Interest Points

1. hat eine mathematisch eindeutige Definition
2. hat eine klar definierte Position im Bildraum
3. ihr umgebende lokale Bildstruktur hat einen hohen Informationsgehalt
4. ist gegenüber lokalen und globalen Störungen des Bildes stabil - inklusive Deformationen, die durch perspektivische Transformationen oder Variationen in der Beleuchtung/Helligkeit entstehen.
5. sollte skalierungs invariant sein.

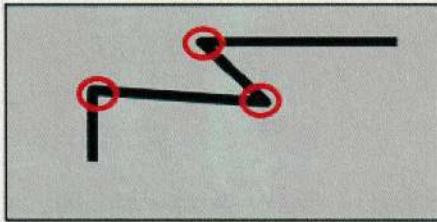
Eckendetektion

Eckpunkte: Ein Eckpunkt kann als Schnittpunkt zweier Kanten definiert werden oder auch als Punkt, in dessen lokaler Nachbarschaft zwei verschiedene und dominante Kantenrichtungen existieren. Das bedeutet, dass ein Eckpunkt beispielsweise auch der Endpunkt einer Linie oder ein Punkt auf einer Kurve, an dem die Krümmung maximal ist, sein kann.

Von verschiedenen Blickwinkeln sowie unter verschiedenen Beleuchtungsbedingungen zuverlässig zu lokalisieren.

„Corner detector“: Findet Eckpunkte (oder Interest Points im weiteren Sinn) unabhängig von Orientierung (isotrop)

Ein Eckpunkt weist im Gegensatz zu einer Kante einen Helligkeitsunterschied in mehr als einer Richtung auf. (Gradient des Bildes hat in mehr als einer Richtung einen hohen Wert - bei nur einer Richtung wäre es eine Kante)



Moravec-Eckendetektor

Ein Fenster wird verschoben und es werden Intensitäts-Unterschiede zwischen der aktuellen $I(x+u,y+v)$ und ursprünglichen Region $I(x,y)$ bestimmt. (Wird in jede Richtung wiederholt)

Sind Veränderungen in alle Richtungen hoch, bleibt auch das Minimum aller Veränderungen hoch und es handelt sich um einem Eckpunkt.

Nachteile:

Anisotropie (Nicht isotrop: Richtungsabhängigkeit)

Wenn eine Kante vorhanden ist, die nicht in der Richtung der vier Nachbarn verläuft (zB diagonal), dann wird diese auch einen vergleichsweise hohen Interest Value aufweisen.

Rauschempfindlichkeit

Abhängig von Skalierung

Harris-Eckendetektor

Verbesserung des Moravec-Kantendetektors: Es wird die Variation der lokalen Bildstruktur betrachtet und nicht Pixeldifferenzen. (isotrop)

Grundlage des Harris-Detektors sind die Gradienten des Bildes, an einem Eckpunkt sollten Gradienten sowohl in der Hauptrichtung als auch normal dazu vorhanden sein.

Nachteile:

Höherer Rechenaufwand

Abhängig von Skalierung (nicht Skalierungsinvariant)

Scale-Invariant Feature Transform (SIFT)

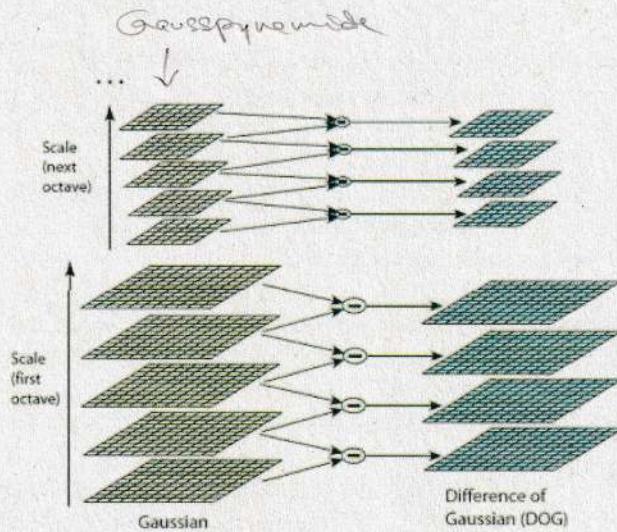
Algorithmus zur Beschreibung lokaler Merkmale in Bildern

Die Methode ist invariant gegenüber der Skalierung und Rotation des Bildes sowie bis zu einem gewissen Grad invariant gegenüber affinen Transformationen und Beleuchtungsveränderungen. Der SIFT-Algorithmus besteht aus vier Schritten:

1. Finden von Interest Points - Skalierung

Durch Difference-of-Gaussians (DoG) werden beispielsweise 5 Mal der Gauss-Filter hintereinander verwendet und dadurch 4 „4 bandpassgefilterte Ergebnisse“ erzeugt.

Dies wird auch als Oktave bezeichnet. Anschließend wird die nächste Ebene der Gaußpyramide (2. Oktave) in der gleichen Weise behandelt. Die einzelnen Oktaven werden benötigt, um Extrema finden zu können. Die Ebenen der ersten Oktave haben die Auflösung des Originalbildes, die Ebenen der zweiten Oktave die halbe Auflösung des Originalbildes in beiden Richtungen und so weiter. In den DoG Bildern – oder auch Laplacebildern – werden die Frequenzen separiert. Dieser Vorgang lokalisiert Kanten und Ecken im Bild.



2. Finden von Interest Points - Position

2.1. Lokalisierung der Extrema (Maxima/Minima) in den DoG-Bildern

In allen Ebenen des DoG Raumes wird nach Extrema gesucht indem dieselben Punkten in verschiedenen Ebenen miteinander verglichen werden → grobe Lokalisierung

2.2. Bestimmung der Position der Extrema mit Subpixel-Genauigkeit

da die Maxima/Minima nicht notwendigerweise genau auf einem Pixel, sondern auch zwischen den Pixeln liegen können.

Um den approximierten Interest-Point werden dafür eigene Subpixel generiert und anschließend wird sich diesen angenähert.

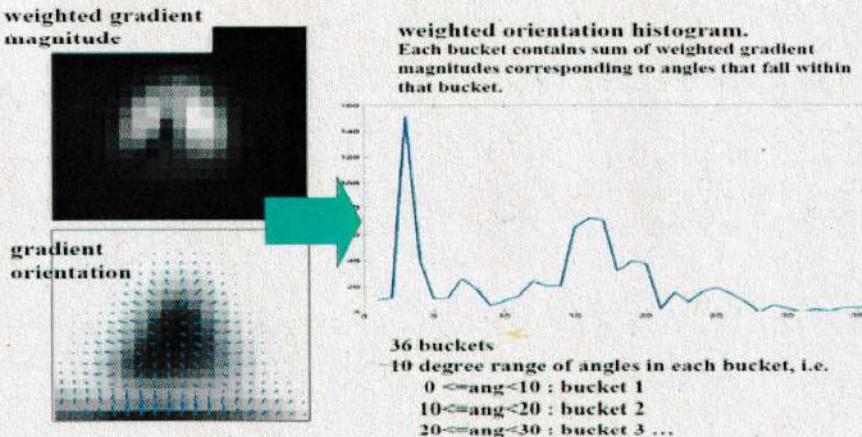
2.3. Eliminierung ungeeigneter Interest Points

Manche der Interest Points liegen entlang einer Kante oder besitzen nicht genügend Kontrast → eliminieren nachdem Intensitätswerte im DoG geprüft werden

3. Finden von Interest Points - Orientierung

Dazu werden rund um jeden Interest Point die Längen (Gradientenbeträge) und Richtungen der Gradienten gesammelt und anschließend die Orientierung der stärksten Gradienten dieser Region bestimmt (die „dominante“ Gradientenrichtung dieser Region).

Dazu wird ein Histogramm mit allen Winkeln erstellt (alle 10 Grad) und der dominierende Richtung wird der Region zugeteilt.



4. Erstellen einer Beschreibung der Merkmale

Um diese Signatur (Beschreibung) zu generieren, wird ein 16x16 Fenster um den Interest Point in dessen Skalierung definiert und in 16 4x4 Fenster unterteilt.

Längere Gradienten (d.h. ein höherer Kontrast) werden stärker gewichtet, weiter entfernte Gradienten werden mithilfe einer Gaußschen Gewichtungsfunktion weniger stark gewichtet als näher liegende Gradienten.

Diese Werte bilden den "Merkmalsvektor".

Der Interest Point ist somit durch diesen Merkmalsvektor identifizierbar. Jedoch hat dieser zwei Einschränkungen:

1. Rotationsabhängigkeit: Da der Merkmalsvektor Gradientenrichtungen verwendet, verändern sich diese bei einer Rotation des Bildes.
2. Beleuchtungsabhängigkeit: Lichteffekte können einen starken Anstieg der Länge einzelner Gradienten bewirken.

CV9: Multiskalen Repräsentation

Multiskalenanalyse

Nachbarschaftsoperationen sind nur Ausgangspunkt für die nachfolgende Bildanalyse, da sie lediglich lokale Merkmale in der Größenordnung von höchstens einigen Pixeln extrahieren. Bilder enthalten aber auch großskalige Information.

Eine Möglichkeit wäre große Filtermasken zu benutzen aber der Rechenaufwand steigt stark an.

Daher hängt die Detektion bestimmter Merkmale in einem Bild von der richtigen Skala ab. Während die Auswertung von feinen Details im Bild die maximale verfügbare Auflösung benötigt, lassen sich grobe Strukturen mit geringerem Aufwand bei reduzierter Auflösung analysieren.

Die Multiskalenanalyse ist ein mathematisches Konzept, das die Signalanalyse auf unterschiedlichen Auflösungsstufen beschreibt.

Abtastung (Diskretisierung von Signalen)

Bei der Bildaufnahme wird eine kontinuierliche Funktion in eine diskrete Funktion umgewandelt, was als Abtastung (Sampling) bezeichnet wird.

Es ist die Entnahme von Abtastwerten der kontinuierlichen Funktion an bestimmten Punkten in der Zeit oder im Raum, üblicherweise in regelmäßigen Abständen.

Zur Diskretisierung eines kontinuierlichen Signals $g(x)$ benötigen wir eine Abtastfrequenz, die mindestens doppelt so hoch wie die maximale Signalfrequenz ist.

Wird diese Bedingung nicht eingehalten, dann überlappen sich die replizierten Spektralteile im Spektrum des abgetasteten Signals und das Spektrum wird verfälscht mit der Folge, dass das ursprüngliche Signal nicht mehr fehlerfrei aus dem Spektrum rekonstruiert werden kann. Dieser Effekt wird als Aliasing bezeichnet.

Bildskalierung

Subsampling – Bilder herunterskalieren

Wie sollen Bilder zB angepasst werden, wenn sie nicht auf einem Bildschirm passen?

Jede zweite oder dritte Spalte löschen erzeugt Aliasing-Effekte, deshalb müssen die Bilder zuerst mit einem Gauß Filter gefiltert werden, um anschließend verkleinert zu werden, wobei der Filter nach dem Abtasttheorem doppelt so groß wie die gewünschte Verkleinerung sein muss.

Upsampling / Resampling – Bilder hochskalieren

Man muss sich die Abtastfunktion ansehen und bei zusätzlich hinzugefügten Datenpunkten deren Werte aus den benachbarten interpolieren.

Dazu sind zwei Resamplingverfahren allgemein verbreitet:

Bilineare Interpolation

Bilineare Interpolation errechnet den Wert einfach aus den Nachbarwerten

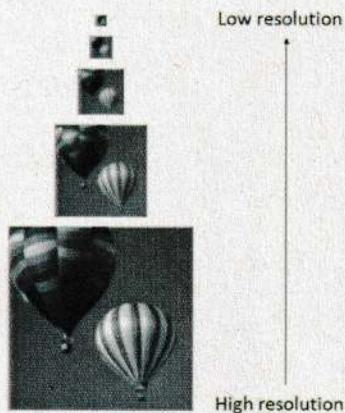
Bikubische Interpolation

Bei der bikubischen Interpolation geht man von der vergrößerten neuen Bildmatrix aus und rechnet mit Hilfe der Transformationsgleichungen von den Mitten der dortigen Bildelemente in das Eingabebild zurück.

Bildpyramiden

Für viele Aufgaben der Bildauswertung ist es wünschenswert, Strukturen auf unterschiedlichen Skalen effizient zu analysieren, um verschiedene Objekteigenschaften untersuchen zu können und eine gewisse Invarianz bezüglich der Größe zu erreichen. Wendet man jedoch Bildauswertungsverfahren naiv auf mehreren Skalen an, so ist dies mit beträchtlichem Rechenaufwand verbunden.

Bildpyramiden: Mehrgitterdarstellung zur Verarbeitung der Signale in unterschiedlichen Skalen. In der Pyramide liegt das Bild in verschiedenen Auflösungen vor. Dadurch können sowohl grobe als auch feine Bildstrukturen mittels kleiner lokaler Operatoren ausgewertet werden.



Gauss-Pyramide

Eine mit einem Tiefpass (Gaußfilter) konstruierte Bildpyramide heißt Gaußpyramide. Dadurch bleiben auf den obersten Pyramidenebenen mit der niedrigsten Auflösung nur noch niedrigere Ortsfrequenzen.

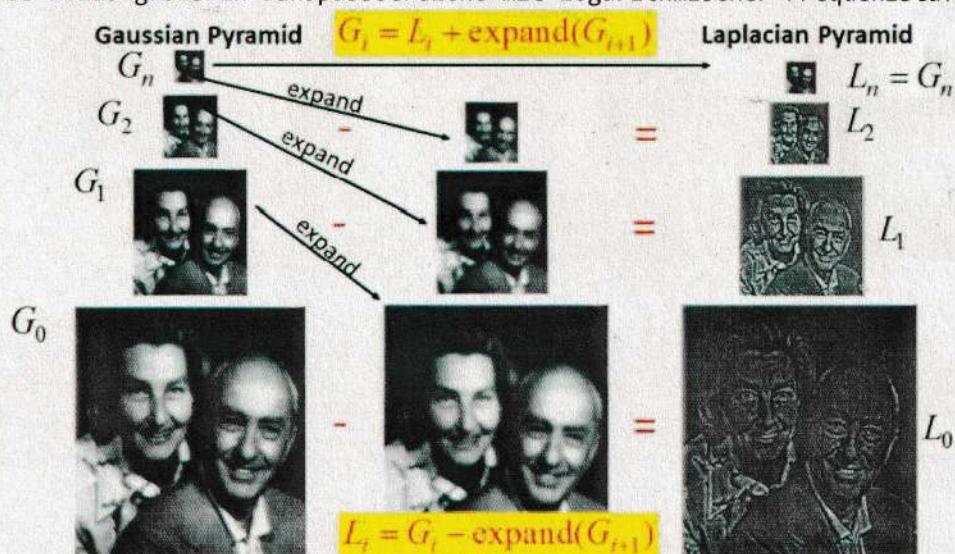
Laplace-Pyramide

Alternativerweise kann eine Bandpassfilterung (Frequenzfilter, mit beliebiger Höhe und Tiefe) des Originalbildes auch eine Pyramide aufbauen.

Eine einfache Möglichkeit, um eine solche Bandpassfilterung zu realisieren, ist die Subtraktion zweier aufeinanderfolgender Bilder einer Gauß Pyramide (Difference-of-Gaussians DoG).

Dafür muss immer von einer Ebene hochskaliert (Interpolierung der fehlenden Informationen) werden.

Auf den ersten Stufen der Laplace-Pyramide werden feine Kantenstrukturen hervorgehoben, während auf den höheren Stufen gröbere Kantenstrukturen des Originalbildes sichtbar sind. Die Laplace-Pyramide ist somit eine Zerlegung des Bildsignals in Bandpassbereiche mit logarithmischer Frequenzstufung.



Stereo Vision

Dieses zusammengesetzte Wort bezeichnet das räumliche Sehen, welches auch als menschliches bzw. binokulares Sehen bezeichnet wird.

Dadurch wird ein Tiefenbild aus zwei Bildern einer Szene erstellt.

Der Versatz der in den entstandenen Bildern der beiden Kameras zeigt sich in weiter entfernten Objekten weniger als in näheren Objekten. --> „Disparität“

Durch Zuweisung der unterschiedlichen Disparitäten zu jedem Bildpunkt wird eine Disparitätsmatrix erstellt. Aus dieser lässt sich für jeden Bildpunkt die Tiefe ableiten.

Stereoskopie

Umgangssprachlich (falschlicherweise) „3D“ genannt obwohl es sich nur um zweidimensionale Abbildungen handelt, die einen räumlichen Eindruck vermitteln.

Wie funktionieren nun Stereoskope?

Halbbilder nebeneinander mit einer speziellen Blicktechnik
Parallelblick oder Kreuzblick

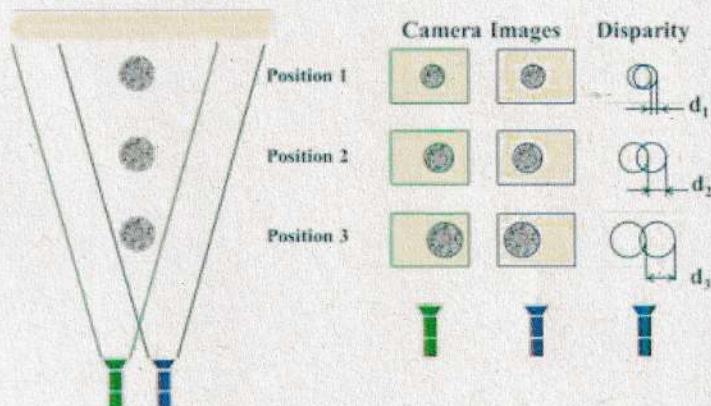
Halbbilder in Komplementärfarben

Zur Trennung der beiden Einzelbilder werden verschiedene Farbfilter in 3D-Brillen verwendet (R- Rot, L- Grün) Dies reduziert jedoch den Farbeindruck, daher arbeitet man heute mit polarisiertem Licht. Es befinden sich dazu jeweils entsprechend versetzte Polarisationsfilter vor den Projektionsobjektiven und in den 3D-Brillen der Betrachter.

Disparität

Ein einzelner Punkt wird in beiden Bildern des Stereosystems auf unterschiedliche Bildkoordinaten abgebildet. Die horizontale Differenz zwischen diesen nennt man Disparität.

Daher wird die Disparität größer, je näher das Objekt zur Kamera ist, und ist 0 im Unendlichen.



Stereosysteme

Kameraanordnungen mit zwei Kameras werden Stereosysteme genannt und lassen sich bezüglich ihrer räumlichen Anordnung in zwei grundsätzliche Klassen einteilen.

Normalfall

Ein achsparalleles Stereosystem zeichnet sich durch zwei Kameras aus, die horizontal verschoben und deren Koordinatensysteme nicht gegeneinander verdreht sind. Der Abstand zwischen den beiden optischen Zentren wird Basislänge B genannt und die Verbindungsgerade Basislinie (engl. baseline).

Epipolargeometrie

(Allgemeine Form) Bei der konvergenten Anordnung richten sich die optischen Achsen auf einen Konvergenzpunkt aus. Aufgrund der gedrehten Bildebenen schneidet eine Gerade beide Bildebenen.

Korrespondenzproblem

Das Korrespondenzproblem stellt das zentrale Problem der Stereo Vision dar.

Es bezeichnet die Aufgabe, für jeden Bildpunkt im linken Bild jenen Punkt im rechten Bild zu finden, der denselben Objektpunkt abbildet. Das entsprechende Suchverfahren bezeichnet man als Korrespondenzanalyse oder auch als Stereo Matching.

„Epipolar-Rektifizierung“ (Entzerrung) bringt Objekte aus beiden Bildern auf eine gemeinsame Horizontale Linie um den Prozess zu beschleunigen.

Regionenbasiertes Matching (engl. Area-Based Matching - ABM)

Vergleich von 2 Ausschnitten aus rektifizierten Bildern und Analyse entlang einer „Scanline“ während Grauwerte verglichen werden.

Sum of Absolute Differences - SAD

Sum of Squared Differences - SSD

Normalized Cross Correlation - NCC

--> Um die Ähnlichkeiten der beiden Bilder zu messen

Mit Hilfe der Position des Maximums oder Minimums der Ähnlichkeitsfunktion wird die Position des korrespondierenden Punktes bestimmt (Disparität).

regionenbasierte Matching-Methode --> ein dichteres Tiefenbild

Hier werden die Tiefenwerte für alle Pixel direkt ausgerechnet, nicht nur für einige ausgewählte Merkmalspunkte, dafür aber mehr Rechenaufwand

Merkmalsbasiertes Matching

Merkmale zuerst individuell extrahieren vor Matching. Lokale Merkmale können Ecken, Kanten oder andere Interest Points sein.

Structure from Motion SfM

Die Kamerabewegungen und die Kamerageometrie wird erst aus Korrespondenzen im Film bestimmt (die selbst auch gefunden werden müssen).

Bewegungsfeld

Je weiter ein Objekt vom Betrachter entfernt ist, desto geringer ist dessen Bewegung:
Die Sterne und der Mond bewegen sich nicht, die Straße direkt neben dem Beobachter bewegt sich sehr schnell.

Bewegt sich die Kamera ohne zu rotieren, bewirkt dies ein „nach außen“ oder „nach innen“ Zeigen aller Vektoren zu einem einzigen Punkt, auch Focus of Expansion (FoE) oder Focus of Contraction (FoC) genannt.

Multi-View Geometry

3D-Rekonstruktion mit mehr als 2 Bildern bis eine stabile 3D Punktwolke entsteht, die durch Bündelausgleich verbessert werden kann. Der Begriff Bündelausgleich (engl. Bundle Adjustment), oder auch Bündelblockausgleichung, stammt aus der Photogrammetrie.

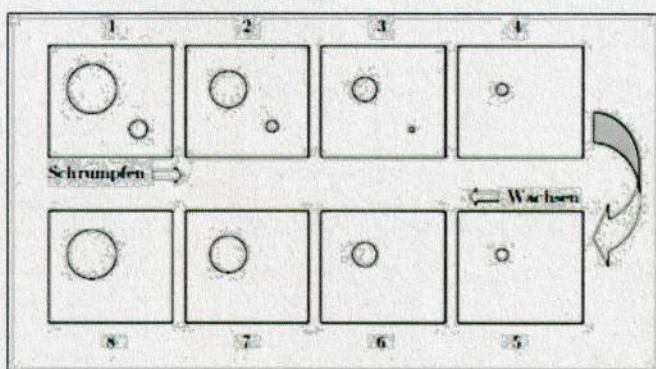
Grundlegende Annahme für den Bündelausgleich sind die Starrheit der Szene zwischen den einzelnen Ansichten und die Erfüllung der Kollinearitätsgleichung (engl. Collinearity Condition).

CV11: Morphologische Operationen und Bildsegmentierung

Morphologische (Morphologie = Lehre der „Gestalt“) Filter sind – in ihrer ursprünglichen Form – primär für Binärbilder gedacht, d.h. für Bilder mit nur zwei verschiedenen Pixelwerten 0 und 1. Binärbilder können z.B. aus Grauwertbildern

durch eine einfache Schwellwertoperation erzeugt werden, wobei wir Bildelemente mit dem Wert 1 als VordergrundPixel (Foreground) bzw. mit dem Wert 0 als Hintergrund-Pixel (Background) definieren. Ausgangspunkt für die weiteren Betrachtungen ist die Beobachtung, dass, wenn wir einen 3×3 -Medianfilter auf ein Binärbild anwenden, sich größere Bildstrukturen abrunden und kleinere Bildstrukturen, wie Punkte und dünne Linien, vollständig verschwinden.

Dies ist also nützlich, um Strukturen unterhalb einer bestimmten Größe aus dem Bild zu eliminieren.



Schrumpf-Operation – „Erosion“

Wachs-Operation – „Dilation“

Diese Operationen sind allgemeiner als im obigen Beispiel illustriert. Insbesondere gehen sie über das Abschälen oder Anfügen einer einzelnen Schicht von Bildelementen hinaus und erlauben komplexere Veränderungen.

Strukturelement

Ähnlich der Koeffizientenmatrix eines linearen Filters, wird auch das Verhalten von morphologischen Filtern durch eine Matrix spezifiziert, die man hier als „Strukturelement“ bezeichnet. Die Matrix beinhaltet nur 1/0.

Zusammengesetzte Operationen

Opening: Eine Erosion gefolgt von einer Dilation mit demselben Strukturelement

Ein Opening bewirkt, dass alle Vordergrundstrukturen, die kleiner sind als das Strukturelement, im ersten Teilschritt (Erosion) eliminiert werden – siehe oberes Bild

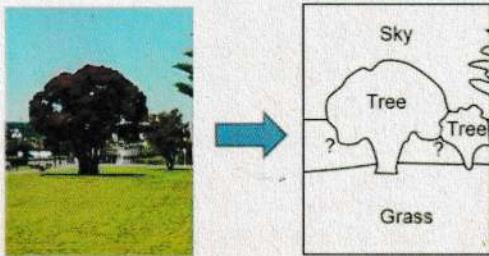
Closing: Dilation und dann Erosion

Durch ein Closing werden Löcher in Vordergrundstrukturen und Zwischenräumen, die kleiner als das Strukturelement H sind, gefüllt

Beide Operationen, Opening und Closing, sind idempotent, d.h., ihre Ergebnisse sind insofern „final“, als jede weitere Anwendung derselben Operation das Bild nicht mehr verändert.

Segmentierung

Unter Segmentierung versteht man eine Operation, die für jeden einzelnen Bildpunkt entscheidet, ob er zu einem Objekt gehört oder nicht. Dabei entsteht ein Binärbild (bei nur einem Objekt im Bild), oder ein Bild mit mehreren eindeutigen Nummern (Identifikatoren oder Labels).

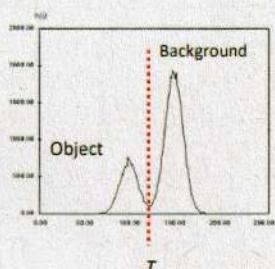


Segmentierungsverfahren lassen sich generell in 3 Klassen einteilen:

Punktorientierte Verfahren

verwenden die Grauwerte der einzelnen Pixel.

Das Segmentierungskriterium ist die Intensität eines isoliert betrachteten Pixels basierend auf einem Histogramm von allen Werten im Bild.



Im 3D-Raum sucht man dann nach Punktwolken (Clustern) die den einzelnen Segmenten zugeordnet werden.

Kantenorientierte Verfahren

erkennen Kanten und versuchen ihnen zu folgen.

Regionenorientierte Verfahren

untersuchen Grauwerte in zusammenhängenden Regionen --> Das entscheidende Merkmal eines Objekts ist aber, dass es zusammenhängend ist.

Region Growing Algorithmus

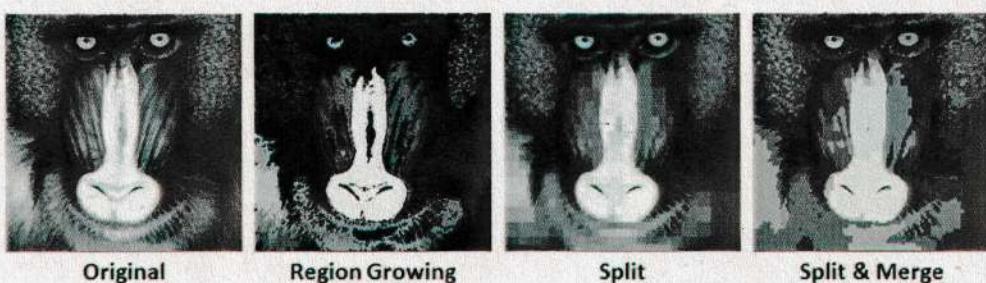
Der zentrale Teil des Algorithmus besteht darin, die angrenzenden Nachbarn bereits gefundener Regionen auf ein bestimmtes Homogenitätskriterium zu überprüfen. Wird dieses Kriterium erfüllt, wird das jeweilige Pixel der Region zugeordnet. Erfüllt kein Nachbar der bereits gefundenen Regionen das Homogenitätskriterium, wird ein neuer Saatpunkt generiert.

Split & Merge

Als Eingabeparameter werden hier das Bild, ein Homogenitätskriterium (bspw. Farbgleichheit oder Farbähnlichkeit) und eine Datenstruktur wie ein Quadtree erwartet.

1. Wenn Kriterium nicht erfüllt wird wird gesplittet (rekursiv tiefer in das Quadtree) und wenn es erfüllt wird bleibt man auf der Ebene stehen.

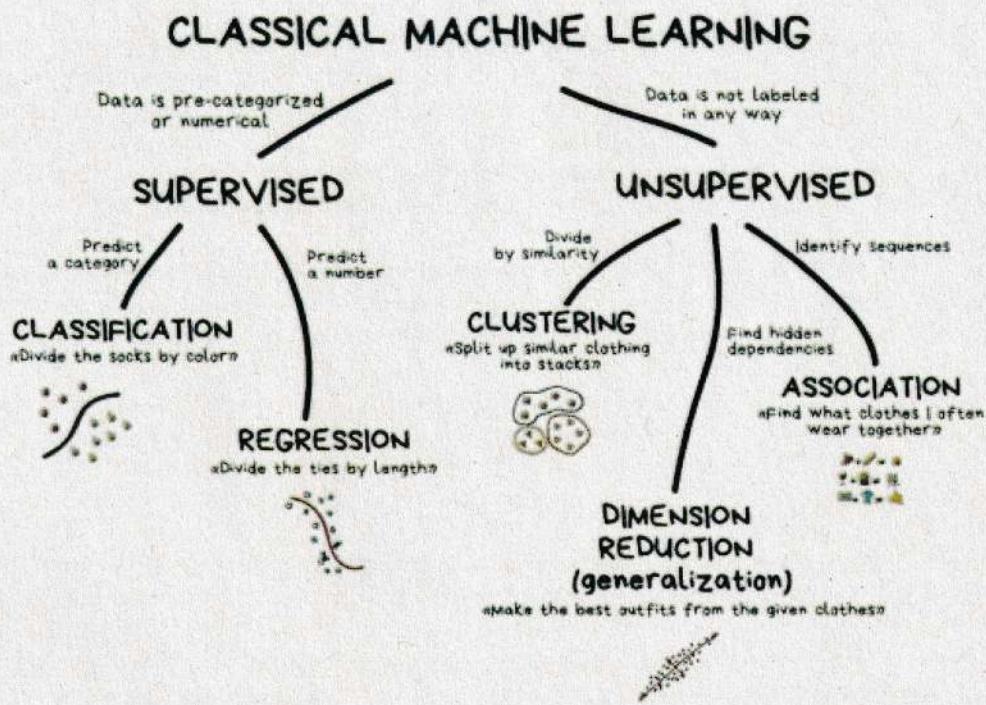
2. Dann muss man zusammengehörige Quadtree-Nodes miteinander mergen



Maschinelles Lernen

Teilgebiet der künstlichen Intelligenz

Maschinelle Lernverfahren werden in folgende Kategorien gegliedert:



Supervised learning – überwachtes Lernen

Im Allgemeinen besteht das eigentliche Ziel des Supervised Learning darin, die Genauigkeit von Vorhersagen zu maximieren.

Unsupervised learning – unüberwachtes Lernen

Findet bisher unbekannte Muster und Zusammenhänge im Datensatz

Reinforcement learning – bestärkendes Lernen

Lernprozess des Algorithmus basiert auf Belohnung und Bestrafung mit dem Ziel den Nutzen zu maximieren.

Supervised learning

Optimierung des Ausgabewertes basierend auf den Eingabewerten („Attributen“)

Der eigentliche Lernprozess zur Vorhersage der Zielvariable basiert auf dem Training Set, wohingegen die Evaluierung dieses trainierten Modells mithilfe des Testsets erfolgt. (Um Genauigkeit und Fehlerquote zu bestimmen).

Klassifikation

Wir haben ein Set an Daten die „labeled“ sind.

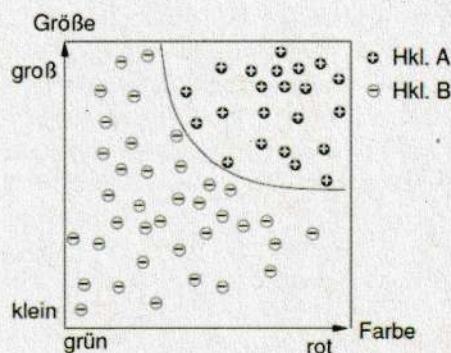
Beispiel:

Äpfel sollen in 2 Klassen eingeteilt werden, A und B

Wir haben pro Apfel 2 Merkmale „features“: Größe und Farbe (Merkmalsvektor)
Ein Classifier soll die Äpfel einteilen lernen.

Zur Einstellung der Maschine werden Äpfel von einem Fachmann handverlesen, das heißt klassifiziert.

Die Aufgabe beim maschinellen Lernen besteht nun darin, aus den gesammelten klassifizierten Daten eine Funktion zu generieren, die für neue Apfel aus den beiden Merkmalen Größe und Farbe den Wert der Klasse (A oder B) berechnet.



In diesem Beispiel ist es noch recht einfach, solch eine Trennlinie für die beiden Klassen zu finden. Deutlich schwieriger, vor allem aber weniger anschaulich, wird die Aufgabe, wenn die zu klassifizierenden Objekte nicht nur durch zwei, sondern durch viele Merkmale beschrieben werden.

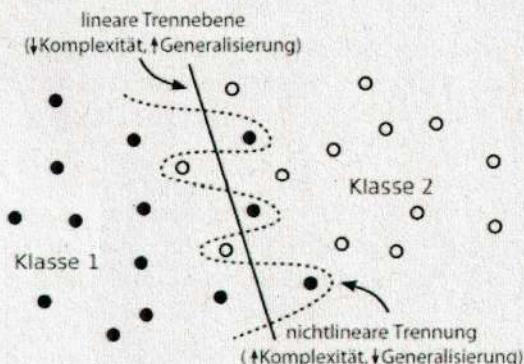
Bei n Merkmalen besteht die Aufgabe darin, in dem n -dimensionalen Merkmalsraum eine $(n-1)$ -dimensionale Hyperfläche zu finden, welche die beiden Klassen möglichst gut trennt.

Zusammengefasst:

Ein Klassifikator bildet somit einen Merkmalsvektor auf einen Klassenwert ab. Hierbei hat er eine feste, meist kleine, Anzahl von Alternativen. Diese Abbildung wird auch Zielfunktion genannt. Aufgabe des verwendeten Lernverfahrens ist es, eine solche Abbildung zu lernen.

Generalisierung

Beispiel: „Modellkomplexität“ vs „Generalisierung“



Man sollte nicht immer die komplexesten Modelle nehmen auch wenn sie präziser sind weil es schnell zu einer Überanpassung führen kann.

Ein komplexes Modell wird immer einen niedrigen Bias-Term (Fehlenschätzung) haben, da es sich flexibel an gegebene Lerndaten anpassen kann, dieser Tatsache steht aber immer eine Erhöhung des Variance-Term (Varianz der Schätzungen bei unterschiedlichen Lerndatensätzen - Maß der Komplexität des Entscheidungs-Modells) entgegen.

Künstliche neuronale Netze

Das Gehirn dient als Vorbild für das mathematische Modell der NN (Perceptron und Multilayer-Perceptron).

Die Neuronen sind die Knoten des Netzes, die Synapsen sind die Kanten zwischen den Knoten, die mit Gewichten versehen werden (weil die Synapsen im Gehirn auch adaptiv sind, nicht die Neuronen). Somit kann über eine Adaption der Gewichte das „Verhalten“ des Neurons verändert werden.

Perceptron - Eines der berühmtesten neuronalen Netze

Ein einfaches Perceptron besteht aus einem einzelnen künstlichen Neuron und einer Reihe von Eingängen I_1 bis I_k . (input)

Die Signale an den Eingängen werden mit Gewichten w_1 bis w_k (weight) multipliziert bevor sie beim Neuron eintreffen.

Die Gewichte verstärken die Signale oder schwächen sie ab und entsprechen den Verstärkungen und Abschwächungen der natürlichen Signale durch die Synapsen.

Die gewichteten Eingaben werden aufsummiert: $\phi(x) = \sum w_i \cdot I_i$

Darauf wird dann eine Aktivierungsfunktion angewendet: $O = f(\sum w_i \cdot I_i)$

Die Funktion f ist eine Schwellwertfunktion:

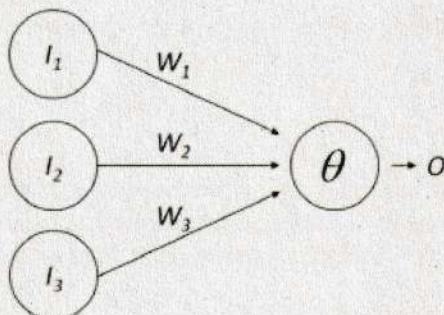
Die gewichteten Eingangssignale $\phi(x)$ des Neurons werden mit einem Schwellwert θ verglichen.

Nur wenn der Schwellwert überschritten wird, feuert das Neuron:

$$O = \begin{cases} 1: & \left(\sum_i w_i I_i \right) + \theta > 0 \\ 0: & \text{otherwise} \end{cases}$$

$$\begin{array}{ll} 1 & \text{if } (\sum w_i \cdot I_i) + \theta > 0 \\ 0 & \text{otherwise} \end{array}$$

Seine Ausgabe hängt also von den Eingangssignalen I , den zu lernenden Gewichten w und dem Schwellwert θ ab.



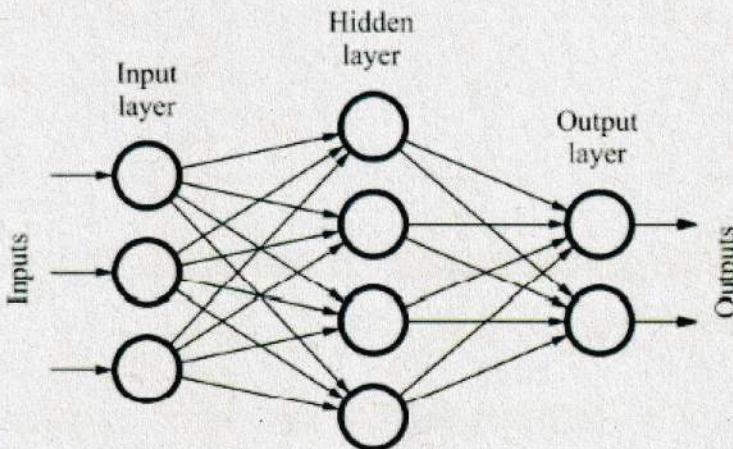
Das Ziel des Perceptrons besteht darin, Daten, die zu zwei unterschiedlichen Klassen gehören, voneinander zu trennen.

Gegeben sind Datenbeispiele mit Klassenzugehörigkeiten. Der Perceptron- Lernalgorithmus soll die Gewichte so anpassen, dass das Neuron bei Daten der ersten Klasse 0 und bei Daten der zweiten Klasse 1 ausgibt.

Die Anpassung der Weights:

Die Anpassung der Gewichte kann mit der δ -Regel, die auch unter der Bezeichnung Widrow-Hoff-Regel bekannt ist erfolgen.

Multilayer Perceptron



Es gibt viele Abwandlungen von dieser Struktur, beispielsweise kann es Direktverbindungen zwischen der Eingangsschicht und der Ausgabeschicht geben, so genannte Shortcut Verbindungen uvm.

Bei Multilayer-Perceptron-Netzen werden beim Lernen die Gewichtsfaktoren zwischen den Verarbeitungseinheiten in einem überwachten Lernvorgang verändert.

Das bekannteste Trainingsverfahren ist die Fehler-Rückvermittlung, besser bekannt als Backpropagation-Training, bei dem in jedem Lernschritt die Gewichte in Richtung abnehmenden Fehlers verändert werden.

Gesucht ist das globale Minimum der Gesamtfehlerfunktion (Summe der Fehler aller Einzmuster). Zu diesem Zweck werden die Gewichte stets in Richtung des abnehmenden Fehlers verändert, d. h. entgegen dem Gradienten des Fehlers.

Die Ausgänge der Zwischenschicht aber sind eine Funktion der Gewichte zwischen Eingangs- und Zwischenschicht, und damit ist letztlich der Fehler F auch eine Funktion dieser Gewichte. Hiervon ausgehend ist es möglich, durch partielle Differenzieren nach der Kettenregel auch für die Zwischenschicht einen Fehler zu definieren, aus dem ein Gradient zur Veränderung der Gewichte bestimmt werden kann.

Der Fehlerwert wird also zuerst für die Ausgangseinheiten berechnet, dann für die verdeckten Einheiten. „Error-Backpropagation“

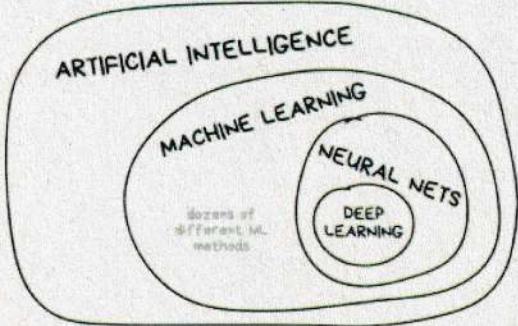
Deep Learning

Aber warum verwenden wir nicht direkt alle verfügbaren Sensordaten, das heißt direkt das Bild wie es vom Sensor kommt? Wir könnten alle 10 Millionen Pixel eines Fotos als Eingabevektor verwenden, der dann eine Länge von 30 Millionen (im Fall von RGB-Bildern mit jeweils drei Farbwerten) hätte. Wo liegt das Problem? Die Trainingszeiten wachsen sehr schnell, oft exponentiell, mit der Dimension der Eingabedaten.

Um die Rechenzeiten in Grenzen zu halten, müssen daher die Eingabedaten zuerst auf kurze Merkmalsvektoren abgebildet werden.

Allerdings kann es für viele komplexe Anwendungen wie etwa bei der Objektklassifikation auf Bildern sehr schwierig bis unmöglich sein, manuell Formeln für gute Merkmale zu finden.

Daher verfolgt Deep Learning das Prinzips des End-to-End-Learning: nicht nur die optimale Verarbeitung der Merkmale wird gelernt, sondern auch simultan die optimale Herleitung solcher Merkmale.



Im Bereich Computer Vision kommen in der Regel Convolutional Neural Networks (CNNs) zum Einsatz, bei denen Filterkoeffizienten gelernt werden, mit denen die Bilder zu Beginn gefaltet (eng. Convolved) werden. Andere – auch außerhalb der Computer Vision – häufig eingesetzte Deep Learning Verfahren sind Recurrent Neural Networks (RNNs) oder Generative Adversarial Networks (GANs) und Variationen davon.

In der Praxis werden viele Schichten eingesetzt und manche Schichten werden supervised und manche sind unsupervised.

Ein Teil des Systems sind aber oft Autoencoder die sich darauf fokussieren sich in einzelnen Schichten einzelnen Merkmalen zu widmen.

CV13: Computational Photography

Computational Photography

Ziel ist es, durch das Zusammenspiel von Bildverarbeitung, Computer Vision und Computergrafik neuartige Funktionalitäten und Anwendungsgebiete für die Photographie zu erschließen. Dabei sollen mittels algorithmischer („computational“) Ansätze die physikalischen Grenzen der traditionellen projektiven Kamera, wie begrenzte Tiefenschärfe, Auflösung und Verlust der dritten Dimension, überwunden werden. Damit können Bilder erzeugt werden, die mit einer herkömmlichen Kamera nicht erfassbar gewesen wären. Durch die Kombination von Software, digitalen Sensoren, moderner Optik sowie intelligenter Beleuchtung geht Computational Photography über die Grenzen der traditionellen Fotografie hinaus.

Lichtfeld

Einfallende Beleuchtung kann mittels eines 4D-Lichtfelds parametrisiert werden. Dazu stellen wir uns um das Objekt eine Kugel vor, die das Objekt einschließt. Wir verwenden (u_i, v_i) um die Position auf der Kugel zu beschreiben, auf der das Licht auftrifft, und die Winkel (θ_i, ϕ_i) beschreiben die Richtung des eintreffenden Lichtes. Daraus ergibt sich das einfallende Lichtfeld aus: $R_i(u_i, v_i, \theta_i, \phi_i)$. Das ausgehende Licht kann auf gleiche Weise mittels eines 4D Lichtfelds parametrisiert werden, der Unterschied ist nur, dass wir jetzt betrachten, wie das Licht die objektumgebende Oberfläche verlässt.

Das Reflexionsfeld beinhaltet daher die nötigen Informationen, um das Objekt unter anderen Beleuchtungsgegebenheiten und von verschiedenen Blickwinkeln aus zu rendern.

HDR – High dynamic range

HDR beschreibt eine Fülle an Techniken, die aufeinander abgestimmt wurden, um Bilder mit mehr Kontrast repräsentieren zu können.

HDR kompensiert diesen Detailverlust durch die Aufnahme mehrerer Bilder mit verschiedenen Belichtungsstufen und ordnet diese, sodass wir ein Bild erhalten, dessen dunkle als auch helle Bildbereiche im Detail darstellbar sind.

Image Inpainting

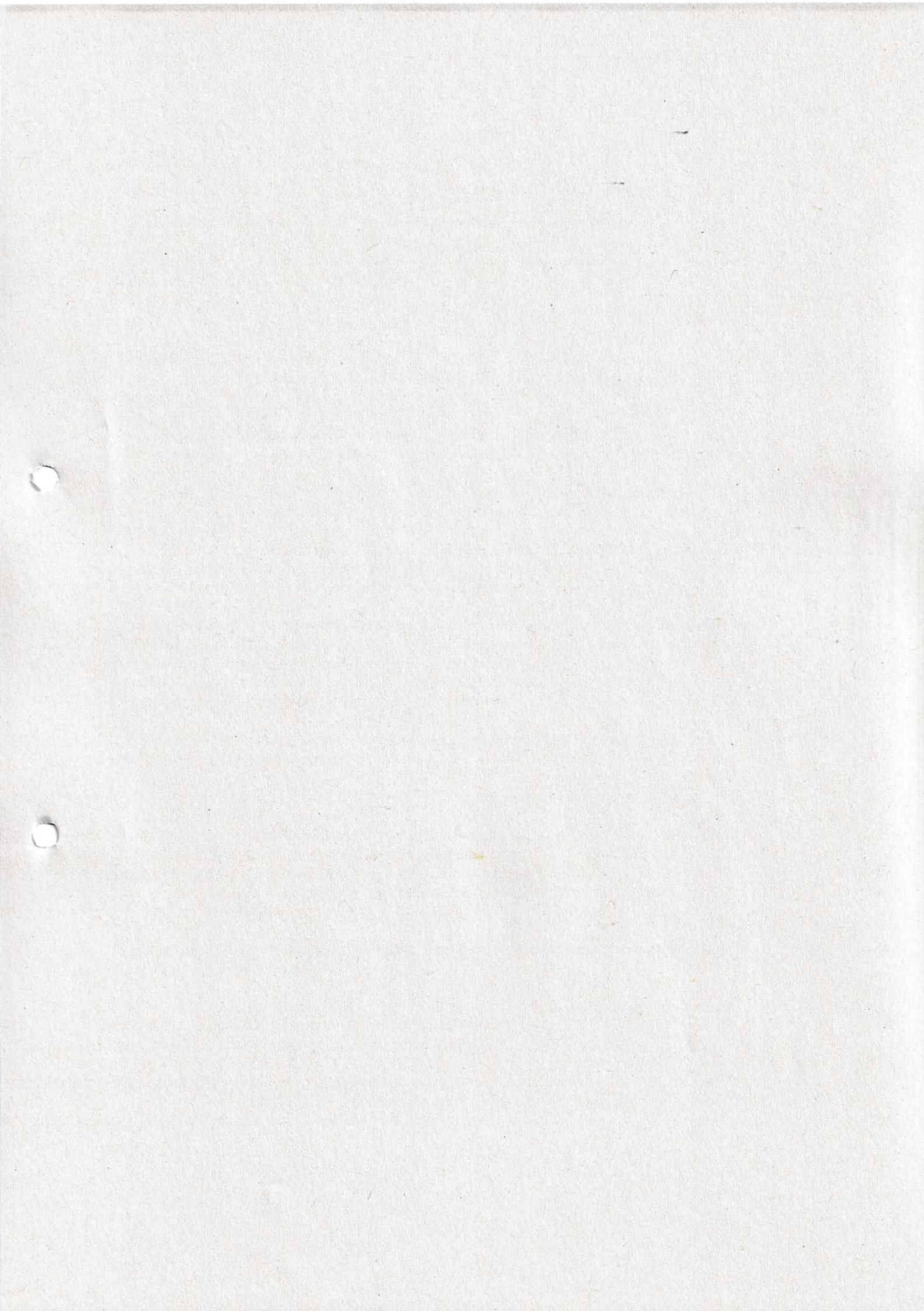
Inpainting wird jener Bildbearbeitungsprozess genannt, der bei der Rekonstruktion von schlecht erhaltenen oder nicht vorhandenen Bild oder Videoteilen Anwendung findet.

Image Mosaicing

Panorama durch das stitchen von Bildern.

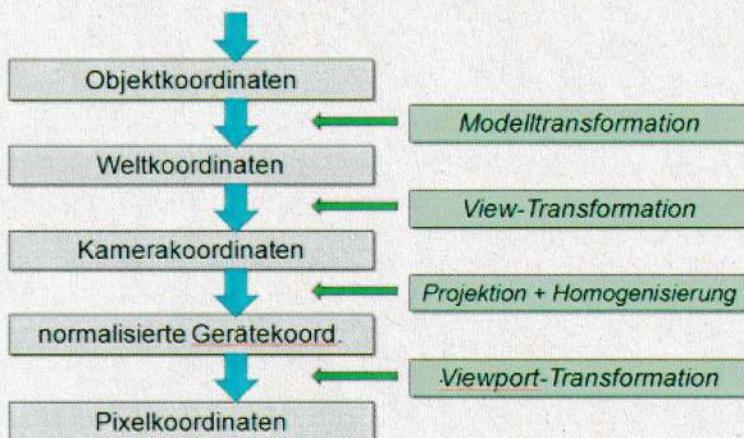
Morphing

Image Morphing ist ein spezieller Bildeffekt, der nahtlos ein Bild zu einem anderen verändert.



CG1: Einführung in die Computergraphik
Künstliche Erzeugung von Bildern, digitale Repräsentation, Erzeugung und Manipulation der Daten.

Komponenten von einer Computergraphik-Software
Komponenten eines Graphiksystems: „Grafik-Pipeline“



Modellierung mit graphischen Primitiven

Objekte werden durch Kombination von geometrischen Primitiven modelliert und in unterschiedlichen geometrischen Datenstrukturen gespeichert.

Graphische Primitive sind graphische Grundformen die unabhängig vom Ausgabegerät sind. In der 2D-Computergrafik sind Punkte, Strecken, Polygone, Kreise und Ellipsen gebräuchlich. In der 3D-Computergrafik sind Dreiecke oder Polygone die gebräuchlichsten Primitive, wenngleich auch Punktfolgen und weitere Formen Verbreitung finden.

Rasterisierung (2D) / Rendering (3D) der graphischen Grundformen

Grundformen werden bei vektorgestützter Grafikbearbeitungssoftware und 3D-Modellierungswerkzeugen verwendet.

Durch die Rasterisierungs-Algorithmen werden sie als Pixelinformation gespeichert.

Weltkoordinaten und Kameraparameter

Nach der Platzierung der Objekte in einem Weltkoordinatensystem wird die Projektion durch Definition entsprechender Kameraparameter festgelegt.

Für beide Operationen benötigt man geometrische Transformationen, die üblicherweise als homogene Matrizen repräsentiert sind.

Clipping

Bildteile, die außerhalb des Betrachtungsfensters liegen, müssen weggelassen werden (Clipping) wie Bildinformationen, die von anderen Objekten verdeckt werden (Sichtbarkeitsberechnung).

Beleuchtungsmodelle

Es gibt einfache Beleuchtungsmodelle aber auch realistischere, wie Ray-Tracing und/oder die globale Beleuchtungsberechnung.

Texturen

Auf Oberflächen werden Texturen aufgebracht, die auch mit lokalen geometrischen Strukturen kombiniert sein können (zB Wölbungen auf einer Kugel).

Anti-Aliasing

Schließlich werden bei der Bildausgabe noch Verfahren angewendet, die den Rastereindruck des Ausgabemediums reduzieren.

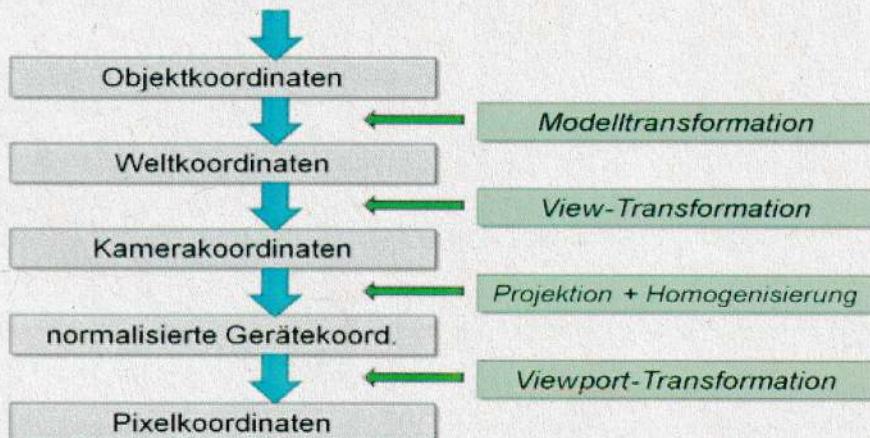
CG2: Graphikpipeline, Objektrepräsentation

Graphikpipeline

Es gibt viele Abbildungen die dieses Grundprinzip beschreiben.

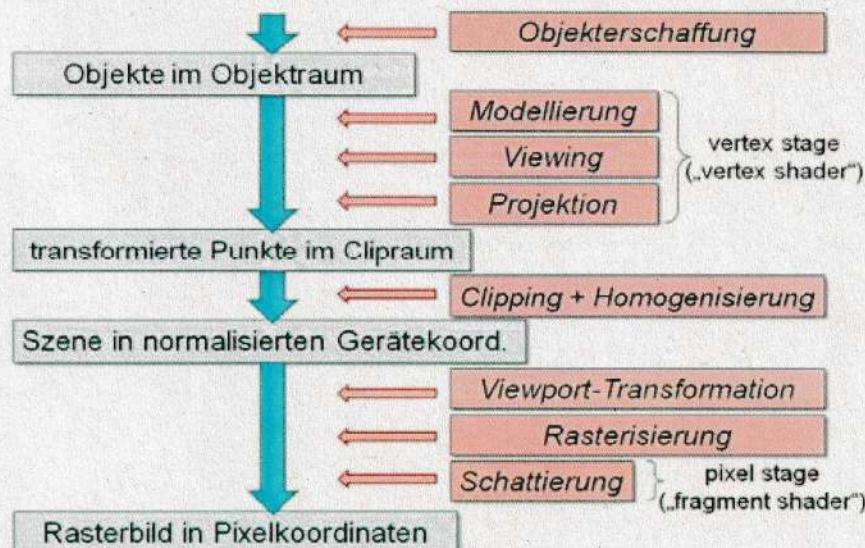
Graphik / Viewing - Pipeline: Umwandlung von Informationen zu einem Bild

1. Objekte und Szene Beschreiben
2. Blickrichtung festlegen
3. Objekte projizieren
4. Ergebnis in Rasterpunkte umwandeln



Transformations / Rendering - Pipeline

Die Begriffe „vertex shader“ und „fragment shader“ werden für programmierbare Teile von Graphikkarten verwendet.



Datenstrukturen

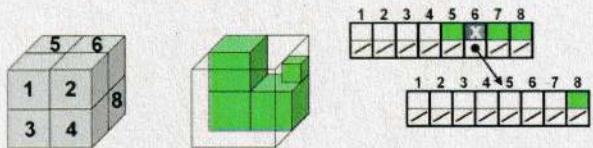
Graphen und Bäume

Rekursive Datenstrukturen die sich auf unterschiedlichen Arten traversieren lassen

Quadtree und Octree

Lässt sich auch in linearisierter Form angeben

In diesen Datenstrukturen sind Mengenoperationen sehr einfach aber geometrische Datenstrukturen sehr aufwändig weil dann die gesamte Datenstruktur neu entworfen werden muss.



Linearisierung: X(EEEEESX(EEEEEEES)SS)

E ... Empty, S ... Solid, X ... Mixed

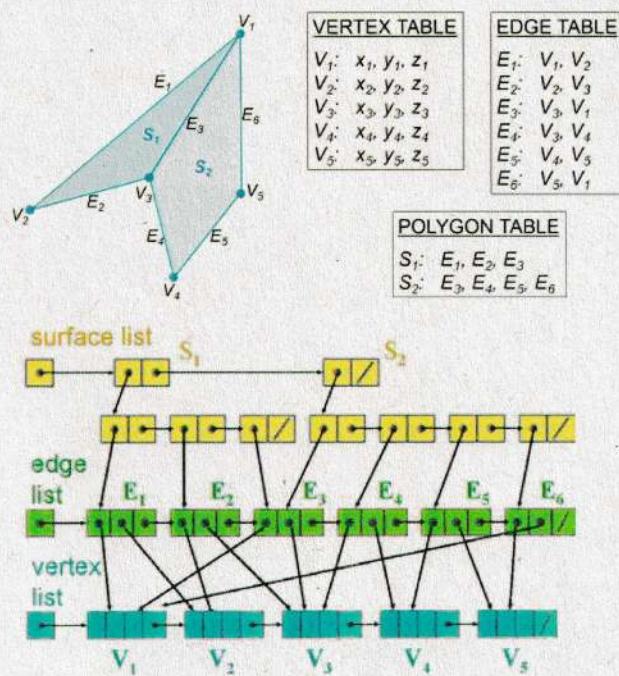
Polygonlisten, Boundary Representations (B-Reps)

Dreidimensionale Objekte werden meist durch Polygonlisten repräsentiert (oft Dreiecke). Eine Menge von Polygonen, die die Oberfläche eines Objektes beschreibt, nennt man Boundary-Representation („B-Rep“).

Sehr häufig werden ausschließlich Dreiecke als Polygone verwendet --> Dreiecks-Mesh, Dreiecks-Strip

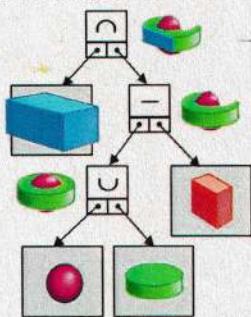
Datenstrukturen für B-Reps enthalten neben geometrischer Information auch Attribute (Eigenschaften). Die Geometrie besteht aus Punktlisten, Kantenlisten und muss auf Konsistenz und Vollständigkeit überprüft werden.

Nur in der Punktliste ist die tatsächliche geometrische Information enthalten, die anderen Listen beschreiben lediglich die Topologie --> deshalb reicht es für eine Transformation einfach die Verteces zu manipulieren



Constructive Solid Geometry (CSG)

Bei CSG werden Objekte aus dreidimensionalen Primitiven mit Hilfe von Mengenoperationen konstruiert. --> CSG-Baum



Als Primitive dienen einfach geometrische Formen wie Kugel, Tetraeder, Würfel, Zylinder, die mit den Operatoren Vereinigung, Durchschnitt und Differenz verknüpft werden.

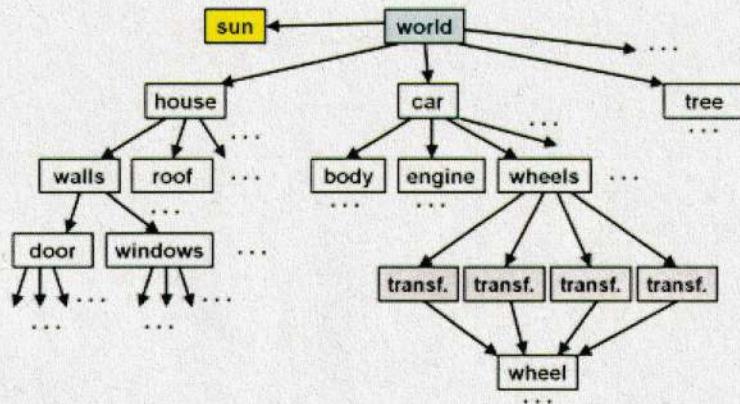
Jeder Knoten des Baumes enthält noch Transformationsmatrizen die auf den darunterliegenden Teilbaum angewandt werden.

Nachteil: kompliziertes Rendering. Würde mit Ray-Casting (Untermenge von Ray-Tracing) einfacher fallen.

Szenengraphen

Einzelne Nodes können mehrfach benutzt werden.

Zwischenknoten enthalten solche Informationen, die den gesamten Teilgraphen betreffen, wie das Material oder die Farbe, die Position und Lage im Raum, die Größe, eine eventuelle Verzerrung. Alle geometrischen Transformationen kann man sehr elegant in Matrizen ablegen.



CG3: Geometrische Transformation

Unter geometrischen Transformationen versteht man das Verschieben, Vergrößern und Verkleinern, Drehen, Spiegeln usw. von Objekten innerhalb eines Koordinatensystems oder zwischen Koordinatensystemen.

Einfache 2D Transformationen

- Translation (Verschiebung)
- Rotation (Drehung)
- Skalierung (Vergrößerung oder Verkleinerung)
- Reflexion (Spiegelung)

Scherung (Verzerrung) - lässt sich aus den oberen zusammensetzen

Homogene Koordinaten

Damit auch die Translation in Matrixschreibweise angegeben werden kann, verwendet man homogene Koordinaten.

Das ist Vorteilhaft weil man auch größere Teile als Ganzes (eine Menge an Punkten) gleichzeitig transformieren kann und auch eine einzige Matrix finden kann die das Produkt von mehreren Operationen zugleich ist.

Jedem Punkt wird eine zusätzliche Koordinate h zugeordnet, wobei die Umrechnung in 2D-Koordinaten durch Division der x - und y -Komponente durch h erfolgt. Daher verwendet man meist $h=1$. Für den Punkt (x,y) schreiben wir daher $(x,y,1)$. Die Transformationsmatrizen werden um eine Zeile und Spalte mit Einheitswerten erweitert:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Affine Transformationen

Alle hier behandelten Transformationen sind affine Transformationen, d.h. die Koordinaten lassen sich durch lineare Funktionen plus einer Translation ineinander überführen.

Affine Abbildungen erhalten **Kolinearität**, d.h. (je) 3 Punkte auf einer geraden Linie sind auch nach der Abbildung auf einer geraden Linie, und die **Proportionalität** von Abständen entlang einer gerade Linie, d.h. das Verhältnis von Längen auf einer Geraden, bleibt erhalten. Weiters bleiben parallele Linien immer parallel und endliche Punkte bleiben im Endlichen.

Alle affinen Transformationen lassen sich aus Skalierung, Rotation und Translation zusammensetzen (auch die Scherung!).

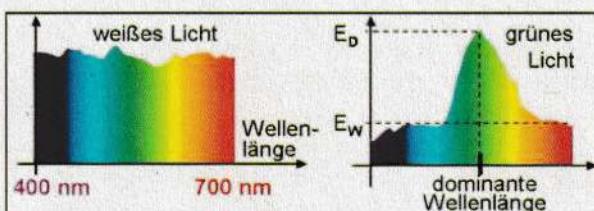
CG4: Farbe

Unser Auge kann innerhalb des sichtbaren Bereiches unterscheiden, welche Frequenz die Strahlung hat, das empfinden wir dann als unterschiedliche Farben. Langwelligeres Licht (also niedrigere Frequenz) empfinden wir als rot, kurzwelligeres Licht (also höhere Frequenz) als blau bis violett.

Tatsächlich kommt in der Natur aber höchst selten spektralreines Licht vor (das nur genau eine Wellenlänge hat), sondern meist sehen wir eine Mischung aus vielen Farben (Spektrum).

Frequenzen mit mehr Energie bestimmen dann welche Farbe wir wahrnehmen, man spricht von dominanter Wellenlänge.

Sind alle Anteile (ungefähr) gleich groß, so sehen wir ein farbloses Licht (also weiß oder grau).



Kolorimetrie

Die Kolorimetrie ist die Wissenschaft von der technischen Beschreibung von Farben. Kolorimetrie berücksichtigt also nur die visuelle Unterscheidbarkeit von elektromagnetischer Strahlung. Alle Spektren, die den gleichen Farbeindruck erzeugen, sind in diesem Sinn nicht unterscheidbar, bilden eine Äquivalenzklasse im Farbraum.

Unser Farbempfinden setzt sich folglich aus der Kombination von drei getrennten „nicht-farbigen“ skalaren Signalen zusammen, daher bezeichnet man das menschliche Farbempfinden als Tristimulus. Die Empfindlichkeitskurven der drei Zapfenarten haben ihre Maxima bei Rot, Grün und Blau.

Empfindlichkeit der Zapfen:

Farbmodelle

CIE 1931 XYZ-Farbmodell

CIE-Diagramm hält eine Approximation von allen Farben fest die das menschliche Auge wahrnehmen kann.

Die mit einem RGB-Monitor darstellbaren Farben, also die Linearkombinationen der Farben Rot, Grün und Blau, die der Monitor erzeugen kann, liegen alle innerhalb des von diesen 3 Punkten aufgespannten Dreieckes. Da es keine drei Farben gibt, die das ganze Diagramm enthalten, kann kein Monitor alle Farben darstellen.

RGB-Farbmodell

Für Bildschirme wird fast immer das RGB-Modell verwendet. Dabei wird ein Pixel aus drei kleinen Farbpunkten zusammengesetzt, deren Lichtsumme (additive Farbmischung)

CMY-Farbmodell

subtraktive Farbmischung: Das CMY-Modell dazu ist das Komplement des RGB-Raumes. Für einfache Anwendungen gilt daher

$$[C, M, Y] = [1, 1, 1] - [R, G, B]$$

HSV- und HLS- Farbmodelle

Eine Art den Farbraum möglichst für den Menschen intuitiv darzustellen

HSV: steht für Hue, Saturation und Value

HLS / HSL: bei dem H=Hue, L=Lightness oder Luminance, S=Saturation stehen

CG5: Rasterisierung

Graphik Bausteine / Graphikprimitive müssen rasterisiert werden damit ein graphisches Output erzeugt werden kann

In 2D:

- Punkte, Linien
- Polygone, Kreise, Ellipsen und andere Kurven, alles auch gefüllt
- Bitmap-Operationen
- Buchstaben und Zeichen

In 3D:

- Dreiecke und andere Polygone
- Freiformflächen

Linien-Algorithmen zum rasterisieren:

DDA (Digital Differential Analyzer)

Wir haben eine Linie der Form $y = mx + b$

Die Koeffizienten m und b lassen sich mit 2 ausgewählten Punkten der Linie berechnen.

Das DDA Verfahren berechnet einfach die eigentlichen Werte und rundet sie ab.

Bresenham-Verfahren

Ident aber benutzt „Integer-Arithmetik“

Attribute

Graphikprimitive können diverse Eigenschaften haben

Linien: Strichdicke, Strichierungsmuster, Farbe oder Pinseltyp

Punkte / Linienenden: Breit, schmal, kantig und spitz, flach

Text: Font, Stil, Größe, Richtung, Farbe, Bündigkeit

Flächen: Können in sich wiederholende Muster haben die mit einem Seedpoint gesetzt werden

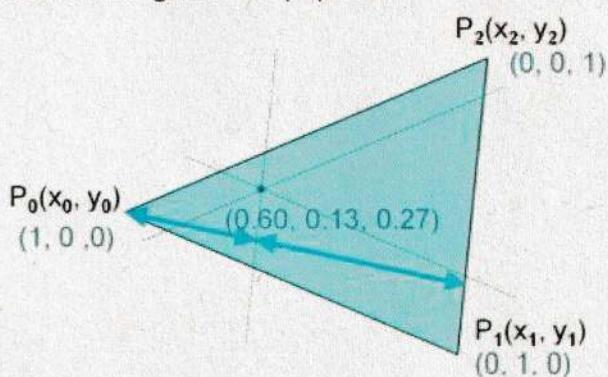
Baryzentrische Koordinaten

Baryzentrische Koordinaten sind eine Grundlage für die Interpolation von Pixeln in Dreiecken.

Jeder Punkt der Ebene wird dabei als gewichtetes Mittel der drei Eckpunkte des Dreiecks dargestellt:

$$P = \alpha P_0 + \beta P_1 + \gamma P_2$$

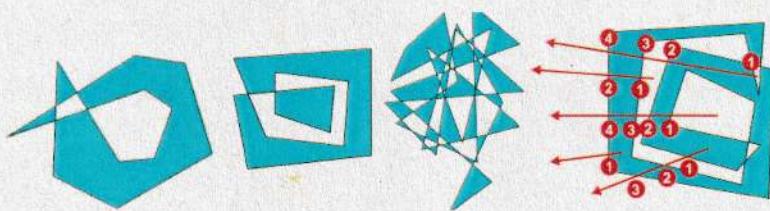
wobei immer gilt: $\alpha + \beta + \gamma = 1$



Bestimmung von „innen“ und „außen“ bei Polygonen

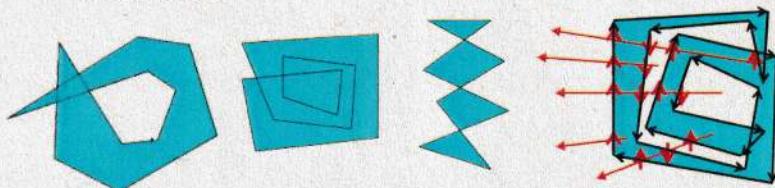
Odd-Even-Rule:

zieht man von einem Punkt aus einen beliebigen Halbstrahl, so ist der Punkt innerhalb, wenn die Zahl der Schnitte mit der Kurve ungerade ist, ansonsten ist der Punkt außerhalb.



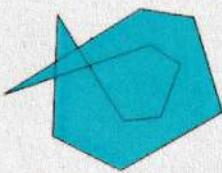
Nonzero-Winding-Number-Rule:

Punkte sind außerhalb, wenn sich auf einem beliebigen Halbstrahl gleich viele im Uhrzeigersinn und gegen den Uhrzeigersinn verlaufende Kurvenkanten befinden, ansonsten innerhalb.



All-In-Rule:

alles, was irgendwie umschlossen ist, ist innen.



CG6: Polygonfüllen

Scanline-Verfahren

Bei den Scanline-Verfahren wird jede Scanlinie mit dem Polygon (oder mit der Flächenumrandung) geschnitten, und innere Teile (Spans) werden gefüllt.

Jede Rasterzeile (Scanline) wird getrennt berechnet und es wird die „Odd-Even-Regel“ angewandt wenn es mehrere Grenzen gibt.

Ziel ist es so viele Informationen aus der vorherigen Zeile zu benutzen wie möglich um den Vorgang zu beschleunigen.

Wir wollen die Zeilen von unten nach oben bearbeiten.

Floodfill-Verfahren

Bei den Floodfill-Verfahren wird die Fläche ausgehend von einem Anfangspunkt in alle Richtungen gefüllt, bis eine Grenzbedingung erfüllt ist (z.B. man stößt an eine Grenzlinie).

Füllt rekursiv alle Pixel von einem Startpunkt aus (zuletzt in EP1 implementiert) bis eine Grenze erreicht wird.

Man speichert bei jeder Kante:

[maximaler y-Wert, x-Wert der 1.Schnittpunktes, inverse Steigung]

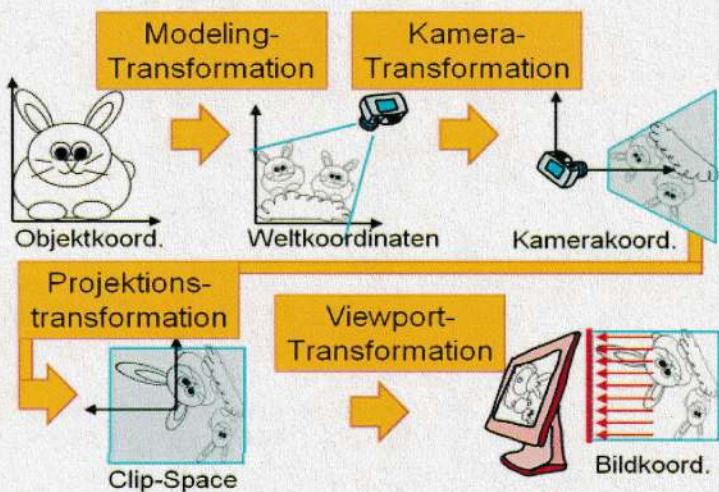
Es kann nämlich vorkommen dass ein Punkt genau auf der Trennlinie für ein Scansegment oder zwischen zwei Trennlinien liegt.

Paralleles Füllen

Durch parallel computing

CG7: Viewing

Viewing in der Graphik-Pipeline



Grundsätzlich erfolgt die Modellierung der darzustellenden Szene in Weltkoordinaten.

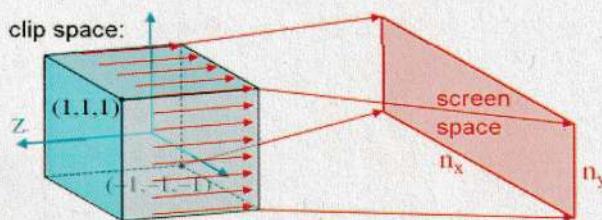
Nach der Definition der Kameraparameter werden die Koordinaten in Kamerakoordinaten verwandelt, auf die anschließend die Projektion angewandt wird.

Das Ergebnis liegt in einem normierten Würfel (häufig der Seitenlänge 2), von wo es durch die sogenannte Viewport-Transformation in die Gerätekordinaten des jeweils verwendeten visuellen Outputs transformiert wird.

[VON HINTEN NACH VORNE]

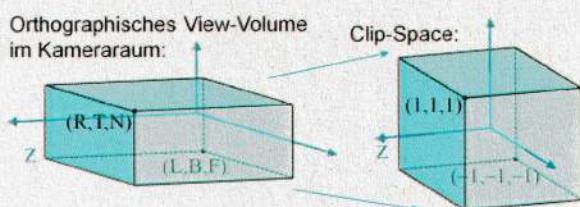
Viewport-Transformation

Wir nehmen also an, dass die Szene bereits im Clip-Space vorhanden ist, d.h. alle relevanten Koordinaten befinden sich in einem achsenparallelen Würfel der Seitenlänge 2 mit Mittelpunkt $(0,0,0)$. Wir wollen eine orthographische Abbildung (parallele Normalprojektion) mit Blickrichtung $-z$ auf einen Bildschirm mit Abmessungen $n_x \times n_y$ (Pixel) durchführen.

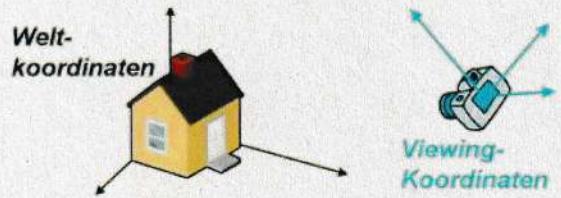


Projektionstransformation

Wie bereits oben erwähnt, nehmen wir für den Moment an, dass eine orthographische Projektion durchzuführen ist. Das vereinfacht die Transformation gewaltig, denn jetzt müssen wir nur einen achsenparallelen Quader mit den Grenzen L(eft), R(ight), B(ottom), T(op), N(ear), F(ar) so verschieben und verzerren, dass der Würfel $[-1, -1]^3$ entsteht.



Kamera Transformation



Alle diese Transformationen und ihre Transformationsmatrizen lassen sich gemeinsam multiplizieren und wenn man eine perspektivische statt orthogonale transformation bevorzugt kann man auch eine weitere Transformation hinzufügen.

CG8: Clipping und Antialiasing

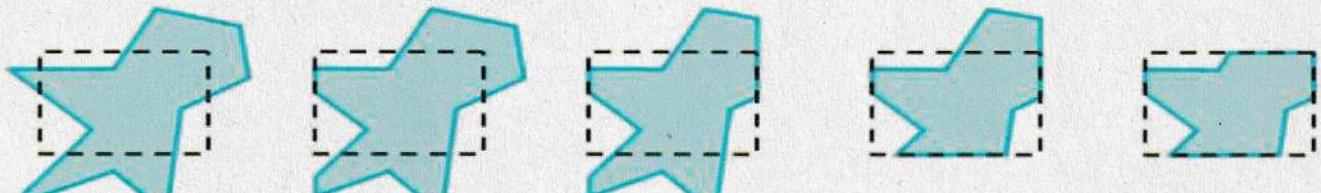
Je früher man die Clippingoperation in der Viewing-Pipeline durchführt, desto mehr unnötige nachfolgende Umformungen von ohnehin nicht sichtbaren Teilen erspart man sich.

Linien-Clipping: Cohen-Sutherland-Verfahren

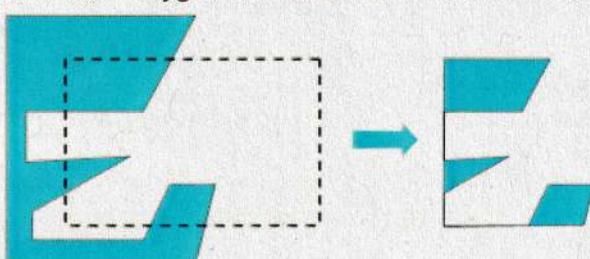
Clippt Linien innerhalb eines Clippingfensters am intersection Punkt mit der Fenstergrenze --> nur x und y Koordinaten die das Fenster nicht überschreiten

Polygon-Clipping: Sutherland-Hodgman-Verfahren

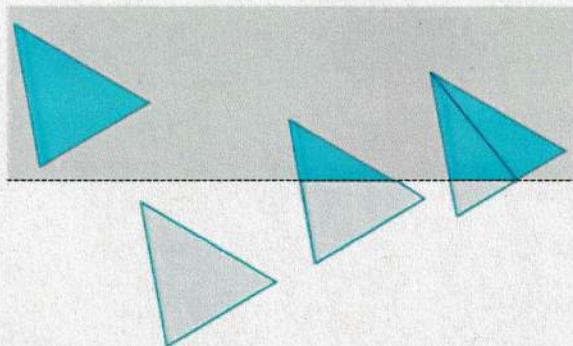
Rekursiver Aufruf für 4 Fenstergrenzen



Wenn das Polygon in mehreren Teilen zerfällt kann sie mit einer Linie verbunden werden.



Wenn man beim Clippen ausschließlich mit Dreiecken arbeiten möchte kann es vorkommen dass beim Clippen eines Dreiecks ein Vieleck entsteht. Dabei muss man aus dem Vieleck wieder (eine redundante Anzahl an) Dreiecken erzeugen.



Aliasing und Anti-Aliasing

Aliasing-Effekte sind Fehler, die bei der Umwandlung (Diskretisierung) von analogen in digitale Informationen auftreten.

Antialiasing nennt man Methoden zur Reduktion unerwünschter Aliasing-Effekte.

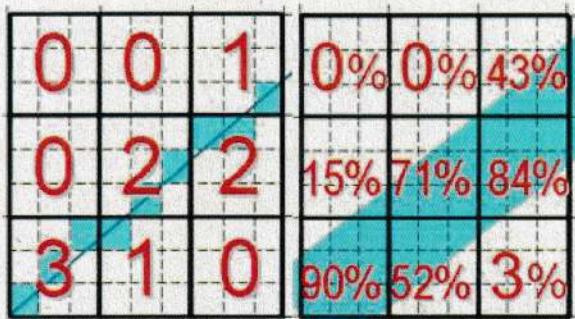
Die Ursache von Aliasing ist ungenügend feine Abtastung des wahren Bildes.

Die theoretische Grundlage dazu bildet das **Nyquist-Shannon Abtasttheorem**: Demnach kann eine Information nur dann korrekt rekonstruiert werden, wenn eine Abtastfrequenz (sampling rate) verwendet wird, die mindestens **doppelt so hoch** ist wie die höchste zu übertragende Informationsfrequenz.

Antialiasing von Linien

Pixel, die von einer Linie weiter durchkreuzt werden, sollen mehr Linienfarbe bekommen, als Pixel, die von einer Linie nur leicht gestreift werden. --> so entsteht ein Übergangseffekt

Dazu unterteilt man jedes Pixel in Subpixel, zählt, wie viele Subpixel auf der Linie liegen, und wählt eine Intensität die proportional zu dieser Anzahl ist. Aufbauend auf der Erkenntnis, dass die Mitte eines Pixels wichtiger ist als dessen Rand, werden auch manchmal die Subpixel in der Mitte stärker gewichtet als die am Rand („weighted oversampling“).



Antialiasing von Polygone

Für Polygonkanten gibt es dieselben Alternativen wie für Linien: entweder man arbeitet mit Supersampling oder man berechnet analytisch den Überdeckungsgrad des Pixels durch das Polygon. --> Dafür muss man mehr Information haben als ausschließlich Pixelwerte

CG9: Sichtbarkeitsverfahren

Methoden um nur Objekte / Objektteile die im sichtbaren Bereich sind zu berechnen.

Objektraum-Verfahren: Backface Detection, Depth Sorting, Octree-Methode

Bildraum-Verfahren: Z-Puffer, Scanline-Methode, Area Subdivision, Ray Casting

Backface Detection / Culling

Backface Culling ist kein vollständiges Sichtbarkeits-verfahren. Es werden lediglich alle Polygone, deren Oberflächennormale vom Betrachter weg zeigt und die daher ganz sicher nicht sichtbar sein können (nach dem Backface Prinzip), eliminiert.

Z-Puffer-Verfahren (Depth Buffer)

Basierend auf Distanz zwischen beobachtenden Punkt (meist auf der Z-Achse)

Scanline-Methode

Es werden die Flächen mit der Farbe vom Polygon gefüllt das am nächsten zum Beobachter ist.

Depth-Sorting-Methode: „Painters Algorithm“

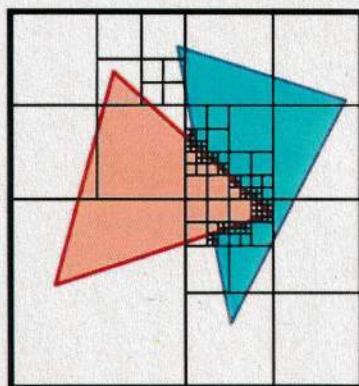
Alle Objekte werden nach depth sortiert und die die am weitesten Hinten sind werden zuerst gezeichnet.

Area-Subdivision Methode

Basiert auf Quadtree und Octree

Es gibt drei einfache Sichtbarkeitsentscheidungen:

1. alle Polygone liegen außerhalb des Fensters → fertig
2. nur ein Polygon hat mit dem Fenster einen nichtleeren Schnitt → dieses Polygon zeichnen
3. ein Polygon überdeckt das Fenster und liegt vor allen anderen Polygonen im Fensterbereich → dieses Polygon zeichnen



Raycasting

Dazu wird vom Pixel aus in Blickrichtung ein Blickstrahl gelegt, das ist eine gerade Linie durch das Pixel in die Szene. In umgekehrter Richtung gelangt auf dieser Linie das Licht aus der Szene auf die Bildebene, also zum Betrachter. Schneidet man diesen Blickstrahl mit allen Objekten/Polygonen der Szene, so erhält man eine Menge von Schnittpunkten, aus denen man den auswählt, der zum Betrachter am nächsten liegt. Die Farbe der Oberfläche an dieser Stelle bestimmt die Farbe des Pixels, durch das man den Strahl gelegt hat.

CG10: Beleuchtung und Schattierung

Ein Beleuchtungsmodell oder auch Schattierungsmodell (Illumination Model oder Lighting Model oder Shading Model) dient dazu, aus der Beschreibung der Lichtverhältnisse und den Oberflächeneigenschaften eines Objektes zu berechnen, welche Farbe/Helligkeit der Betrachter wahrnimmt, also welche Farbe das zugehörige Pixel erhalten soll.

Lichtquellen

Merkmale von Lichtquellen können sein:

Form: Lichtrichtung, Punktlichtquellen, gerichtete Punktlichtquellen, flächige Lichtquellen, ...

Eigenschaften: Helligkeit, Farbe, Entfernung, ...

Objektoberflächen

Oberflächen können das einfallende Licht entweder diffus reflektieren, d.h. es wird in jede Richtung gleich viel Licht reflektiert (z.B. Papier, Kreide), oder spiegelnd reflektieren, d.h. in die Spiegelungsrichtung wird mehr Licht reflektiert als in die anderen Richtungen (z.B. Lack, Metall), oder transparent sein, d.h. das Licht geht durch die Oberfläche durch und kommt auf der anderen Seite wieder heraus (z.B. Glas, Wasser). Reale Oberflächen besitzen meist eine Mischung aus diesen Eigenschaften.

Einfaches Berechnungsmodell

Ambientes Licht

ist Hintergrundlicht (sehr gleichmäßig)

Lambert'sches Gesetz

Dieses Gesetz besagt, dass je flacher Licht auf eine Oberfläche auffällt, desto dunkler erscheint diese Oberfläche. Erst durch diesen Effekt erhalten wir den Eindruck einer räumlichen Form.

Glanzpunkte

Fast jede Oberfläche ist auch etwas spiegelnd. Wenn man diesen Aspekt nicht mitmodelliert, dann wirken alle Materialien gleich stumpf.

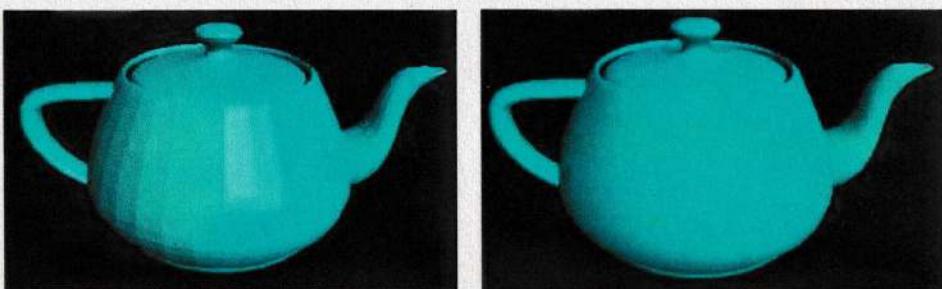
Da die exakte spiegelnde Reflexion äußerst kompliziert zu berechnen ist, behilft man sich mit einer einfachen Funktion, die einen ähnlichen Verlauf hat wie das Highlight: \cos^p . Mit dem freien Parameter p lässt sich dabei die „Poliertheit“ der Oberfläche steuern

Dazu kann man das Beleuchtungsmodell von Phong oder die Fresnel'schen Reflexionsgesetze verwenden

Schattierung von Polygonen

Flat-Shading

Durch die Polygonkanten wirken Objekte nach der Schattierung noch kantiger. Zur Korrektur sind zwei Verfahren üblich: Gouraud-Schattierung und Phong-Schattierung.



Gouraud-Schattierung

Die Gouraud-Schattierung interpoliert die berechneten Helligkeitswerte über die Polygonflächen. Dazu werden an den Eckpunkten der Polygone Helligkeitswerte berechnet und von diesen aus wird durch lineare Interpolation jedes Polygon gefüllt.

Phong-Schattierung

Als Alternative zur Gouraud-Schattierung erzeugt die Phong-Schattierung (nicht mit dem Phong-Beleuchtungsmodell verwechseln!) viel konsistenter Glanzeffekte.

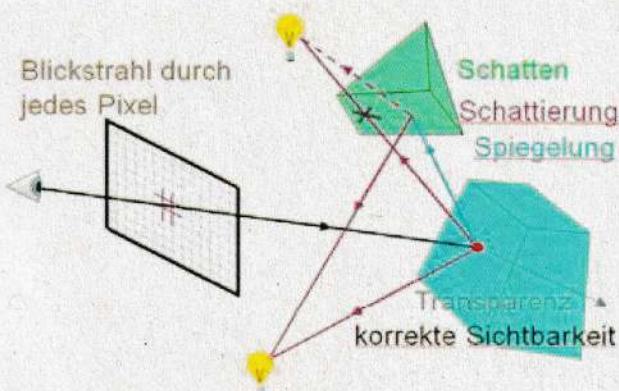
CG11: Ray-Tracing

neben der korrekten Sichtbarkeit einige wichtige optische Effekte simuliert werden können: Schattierung, Schatten, Spiegelbilder, Lichtbrechung. Die Einfachheit des Verfahrens erlaubt es, auch sehr komplexe Objekte auf diese Art darzustellen, wie Freiformflächen, fraktale Oberflächen, mathematische Funktionen aller Art usw.

Die Basisidee ist es, das Licht, das auf einen Bildpunkt trifft, in umgekehrter Richtung zu verfolgen (also zu untersuchen, wo es herkommt) und daraus auf das Aussehen dieses Bildpunktes (Pixels) zu schließen. Tut man dies für alle Bildpunkte (also im einfachsten Fall für alle Pixel), so erhält man eine Abbildung der Szene in korrekter Sichtbarkeit. Dabei kann ein beliebiges Schattierungsmodell verwendet werden, also z.B. das Phong-Modell.

Schatten

Um die Schattierung eines Punktes zu berechnen braucht man neben der Oberflächennormale auch die Richtungen zu allen Lichtquellen. Eine Lichtquelle hat aber nur dann einen direkten Einfluss auf die Schattierung eines Punktes, wenn der Lichteinfall nicht durch andere Objekte verdeckt ist, wenn also keine Objekte zwischen dem Punkt und der Lichtquelle liegen. Um dies festzustellen, legt man einen Schattenfühler (das ist ein „Sekundärstrahl“) vom zu schattierenden Punkt zur Position der Lichtquelle, schneidet diese Gerade mit allen Objekten der Szene und berücksichtigt die Lichtquelle nicht, wenn man einen Schnittpunkt zwischen Objekt und Lichtquelle erhält. --> Auch Simulation von Spiegelungen mit Sekundärstrahl möglich sowie Transparenz (Reflexions und Transparenzstrahl)



Perspektive

Verwendet man parallele Strahlen, die normal auf die Bildebene stehen, so erhält man eine orthonormale Parallelprojektion der Objekte. Lässt man alle Blickstrahlen von einem fiktiven Augpunkt ausgehen, so wird das Bild in Perspektive erzeugt.

Beschleunigung

Beschreibung
Ray-Tracing ist ein extrem aufwändiges Verfahren.

Daher ist es notwendig, das Verfahren signifikant zu beschleunigen. Die wichtigste Methode dazu ist es, die Anzahl der notwendigen Schnittberechnungen durch Ausnutzung von Kohärenz zu reduzieren.

Definition von Objektumgebungen

Bevor man alle Einzelteile eines komplexeren Objektes (oder eines Szeneteiles) mit einem Strahl schneidet, kann man überprüfen, ob der Strahl überhaupt in die Nähe dieses Objektes kommt. Dazu werden in der Datenstruktur zu solchen komplexeren Objekten Umgebungskugeln (bounding spheres) eingefügt, die das Objekt ganz umschließen. Strahlen, die diese Umgebungskugel nicht treffen, treffen auch das Objekt sicher nicht, und man erspart sich viele unnötige Schnittberechnungen.

Auf diese Weise verringert man die tatsächlichen Schnittversuche von $O(n)$ auf etwa $O(\log n)$. Statt Umgebungskugeln kann man beliebige andere Objektumgebungen verwenden, z.B. Umgebungsquader. Es gilt hier abzuwegen, ob der Mehraufwand durch deren kompliziertere Form den Gewinn durch deren engeres Anliegen rechtfertigt. (Auch Octree möglich)

CG12: Globale Beleuchtung und Texturen

Globale Beleuchtung

Radiosity

Das Radiosity-Verfahren stammt ursprünglich aus der Wärmelehre und modelliert die Lichtausbreitung unter Beachtung des Energiegleichgewichtes in einem geschlossenen System.

Das Verfahren beschreibt den physikalischen Vorgang der Ausbreitung von Licht in einer diffus reflektierenden Umgebung, also die Berechnung der Helligkeiten aller Flächen einer Szene unter Berücksichtigung der gegenseitigen Beeinflussung. Somit werden auch Flächen, die nicht direkt beleuchtet sind, eine gewisse Helligkeit erhalten. Jeder beleuchtete Gegenstand wirkt als sekundäre Lichtquelle.

Radiosity ist eine blickpunktunabhängige Methode zur Berechnung der Helligkeit der einzelnen (diffusen) Patches, nach der noch ein Renderingschritt notwendig ist. Meist verwendet man ein einfaches Polygon-Verfahren mit Gouraud-Schattierung. Um im fertigen Bild auch die mit Ray-Tracing möglichen Effekte erzielen zu können, kann man die so erzielten diffusen Schattierungswerte aber auch als Basiswerte für ein Ray-Tracing verwenden. Dadurch lassen sich dann auch Spiegelungen und Schatten usw. schön darstellen.

Path Tracing

Erweiterung von Ray-Tracing (wird auch „Monte Carlo Ray-Tracing“ genannt), bei der an jedem Auftreffpunkt nicht in alle relevanten Richtungen Sekundärstrahlen gelegt werden, sondern entsprechend der dort gültigen Verteilungsfunktion zufällig nur eine Richtung ausgewählt wird.

Photon Mapping

Bei der Photon-Mapping-Methode werden Lichtstrahlen von den Lichtquellen aus verfolgt, also ähnlich wie bei RayTracing, jedoch in Vorwärtsrichtung. An Auftreffpunkten wird die Wirkung des Lichtes gespeichert und später wird zwischen diesen Werten das Aussehen des Objektes interpoliert.

Texturen

Environment Mapping

Um effizient Objektoberflächen die perfekt spiegeln in einer komplexen Umgebung zu rendern ohne diese ganze Umgebung vollständig modellieren zu müssen und ohne Anspruch auf Exaktheit, verwendet man Environment Mapping.

Texture Mapping

Das Aufbringen von Texturen wird Texture Mapping genannt.

Solid Texturing

Eine Textur kann außer in 2-dimensionaler Form auch als 3D-Volumen gegeben sein. Das heißt, es wird in einem 3-dimensionalen Parameterraum für jeden Raumpunkt eine Farbe definiert, die dann von einer Oberfläche, die sich an diesem Raumpunkt befindet, abgerufen wird.

Bump Mapping

Die Grundidee ist es, die Oberfläche unverändert zu lassen, aber den Normalvektor entsprechend der Unebenheiten zu verändern. Dadurch entspricht die Schattierung den Bumps, aber geometrisch braucht man nichts verändern.

Natürlich ist Bump-Mapping lediglich ein großer Schwindel, bei dem die Schattierung verändert wird ohne die Geometrie zu korrigieren. Dementsprechend verbleiben auch sichtbare Fehler, die umso deutlicher werden, je höher die Bumps sind.

Displacement Mapping

Für jeden dieser Fehler gibt es mehr oder weniger aufwändige Hilfslösungen, korrekt ist es aber, die Oberfläche tatsächlich um die Bumphöhe zu verändern. Diese Methode nennt

sich dann Displacement Mapping. Dabei werden die Oberflächenpunkte tatsächlich verschoben und es entsteht natürlich auch eine korrekte Silhouette. Dies ist aber viel aufwändiger zu implementieren, wird aber von neueren Graphikkarten unterstützt. Das geschieht mittels einer zusätzlichen programmierbaren Hardware-Stage in der Rendering-Pipeline zwischen Vertex und Pixel-Stage ("Tesselation-Stage"), die Dreiecke direkt auf der Hardware in viele kleinere Dreiecke unterteilt, und entsprechend einer Displacement-Map verschieben kann

Multitexturing

Kombination von Mappings

CG13: Kurven und Flächen

Darstellung von Freiformflächen (allgemeine Flächen)

Kurven

Allgemeine Kurven können entweder durch eine Formel (analytische Darstellung) definiert werden oder durch Angabe einer Menge von Stützpunkten oder von Kontrollpunkten, die den Verlauf der Kurve bestimmen. Da eine analytische Definition für den Benutzer viel schwerer verständlich ist, wird normalerweise die zweite Methode verwendet.

Kurven, die durch Stütz- oder Kontrollpunkte definiert sind, nennt man auch Splines. Im Folgenden werden einige einfache gebräuchliche Spline-Verfahren beschrieben.