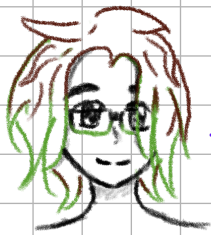


## Disclaimer

Das ist nur eine Zusammenfassung und kein Ersatz für das Skriptum / die VOs. Keine Garantie darauf, dass alles so stimmt, wie es hier steht. Das ist nur meine Interpretation der Inhalte. Falls etwas unklar sein sollte, bitte im Skriptum nachschauen. Einige Unterkapitel (wie z.B. "Andere Objektrepräsentationen") könnten fehlen, da ich sie als unwichtig/nicht prüfungsrelevant erachtet hab. Jegliche Bilder gehören den Urhebern des Skriptums.



Viel Spaß beim Lernen!

## Inhaltsverzeichnis

Globale Operationen.....	2
Bildmerkmale - Interest Points.....	4
Multiskalenrepräsentation.....	6
Stereo und Motion.....	8
Deep Learning für Computer Vision.....	11
Computational Photography.....	14

# Bildtransformationen

Globale Operationen: verändern ganzes Bild

Bild-Frequenzraum-Transformationen: umwandlung von Bild- in Frequenzraum

Transformation = Matrizenmultiplikation - Eingabe mal  $N \times N$  Transf. matrix  $A_N$

# Fourier-Transformationen

⇒ notwendig für Bilddarstellung & -analyse im Frequenzbereich

Wellen besitzen: Amplitude, Frequenz ( $f$ ) & Phasenlänge

⇒ lässt sich als gewichtete sin/cos darstellen; sin/cos = orthogonal

Kreisfrequenz  $\omega = 2\pi f$ , Phasenwinkel  $\varphi = \tan^{-1}\left(\frac{B}{A}\right)$

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

Fourierkoeffizienten:  $A_n, B_n$ ; Grundfrequenz:  $\omega_0$

Fourierintegral:  $g(x) = \int_0^{\infty} A_\omega \cos(\omega x) + B_\omega \sin(\omega x) d\omega$ ;  $A_\omega, B_\omega =$  Gewichte

Bestimmung der Koeffizienten:  $A_\omega = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cos(\omega x) dx$ ;  $B_\omega = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \sin(\omega x) dx$

Fourierspektrum:  $G(\omega) = \sqrt{\frac{\pi}{2}} [A(\omega) - iB(\omega)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [\cos(\omega x) - i \sin(\omega x)] dx$

⇒ in Euler'scher Schreibweise:  $G(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot e^{-i\omega x} dx$

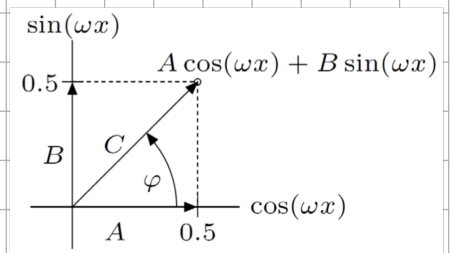
Fouriertransformation = Übergang von  $g(x)$  zu  $G(x)$ ; inverse Ft.:  $G(x) \rightarrow g(x)$

# Diskrete Fouriertransformation (DFT)

$M =$  Anzahl der Abtastwerte;  $g(u) =$  Signal

Fourierspektrum von  $g(u)$ :  $G(m) = \frac{1}{\sqrt{M}} \sum_{u=0}^{M-1} g(u) \cdot e^{-i2\pi \frac{mu}{M}}$  für  $0 \leq m \leq M$

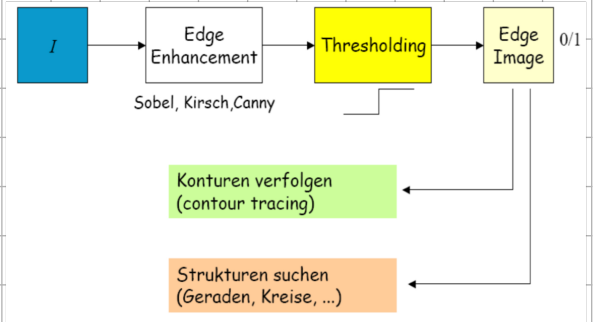
Betrag/Magnitude:  $|G(m)| = \sqrt{G_{RE}^2(m) + G_{IM}^2(m)} =$  Leistungsspektrum



=> beschreibt, wie sehr die Frequenzkomponenten zum Signal beitragen  
 Fast Fourier Transformation (FFT): Reduziert Laufzeit  $O(M^2) \rightarrow O(M \cdot \log M)$

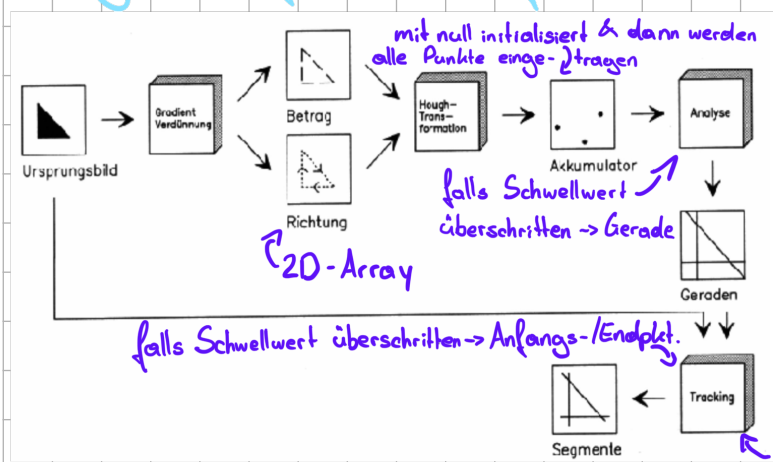
## Hough-Transformationen

= Art der Kantendetektion, die Strukturen im Kantenbild anhand ihrer Parameter sucht



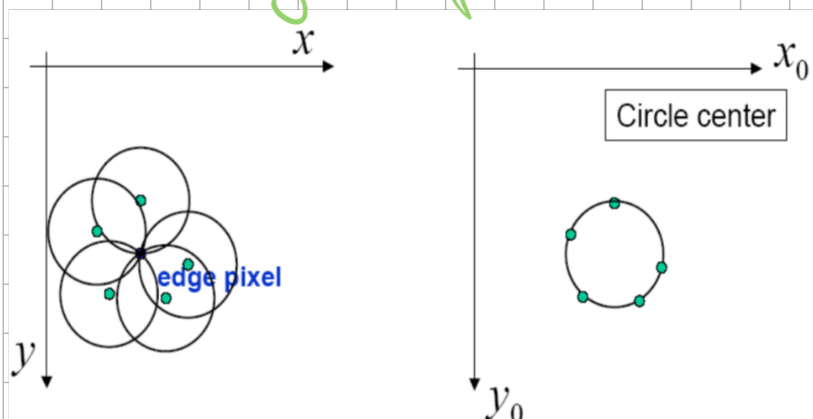
=> ermittelt alle möglichen Geraden, die durch den betrachteten Punkt laufen bzw.  $r = x \cos \theta + y \sin \theta$  erfüllen;  $\theta$  = Winkel zur x-Achse,  $r$  = Abstand zum Ursprung => Ergebnis: Lösungsmenge  $\{(r_j, \theta_j)\}$

## Hough-Transformationen für Linien



Gradientverdünnung: Betrag & Richtung des Gradienten berechn.  
 Analyse-Ergebnis: Geraden ohne Anfangs- & Endpunkten  
 Ermittelt diese

## Weitere Hough-Transformationen



Punkte, in denen sich die meisten Kreise schneiden => Mittelpunkte  
 bei Ellipsen: Berücksichtigung der Halbachsen notwendig

## Bildmerkmale

= mathematische Beschreibungen von Bildteilen (lokal) / Bildern (global)

1. Merkmalsdetektion: Ermittlung von Key/Interest Points (s.u.)

2. Merkmalsbeschreibung: Berechnung von Merkmalsvektoren

Menge all dieser = Merkmalsraum

Ermittlung  
lokaler Merkmale

## Interest Points

= Bildbereiche mit hohem visuellen Informationsgrad (z.B. Kanten, Eckpunkte)

Charakterisierung von Interest Points:

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

Kriterien, die ein guter "Corner Detector" erfüllen muss:

- Eckpunkte zuverlässig unter realistischem Bildrauschen finden
- gefundene Eckpunkte lokalisieren
- effizient arbeiten

Eckpunkt = Punkt mit zwei Kanten in seiner Nachbarschaft

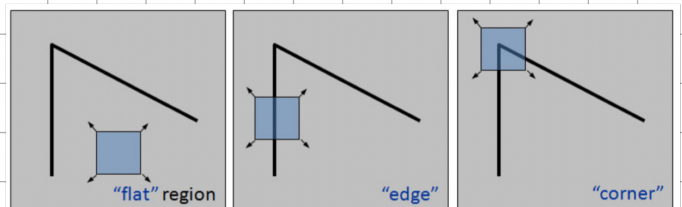
## Moravec-Eckendetektor

Berechnung der Intensitätsunterschiede:  $E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$

$w(x, y)$  = Fensterfunktion;  $I(x+u, y+v)$  =

= Intensität an der verschobenen Stelle

Nachteil: empfindlich gegenüber Bild-



rauschen, anisotropische Antwort (erkennt nur Kanten in Nachbarsrichtung)

## Harris Eckendetektor

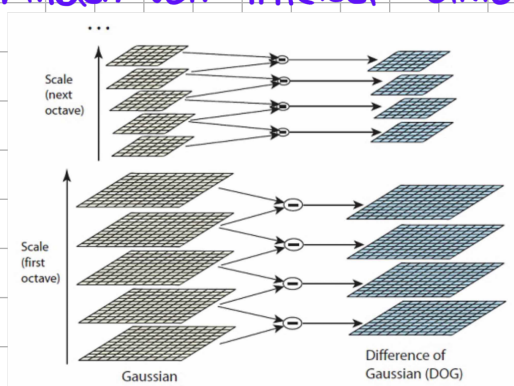
= Verbesserung des Moravec-Eckendetektors, sodass er die Variation der lokalen Bildstruktur betrachtet

⇒ zuverlässiger, aber höhere Berechnungskosten; rotationsinvariant

## Scale-Invariant Feature Transform (SIFT)

(siehe Kantenfilterung)

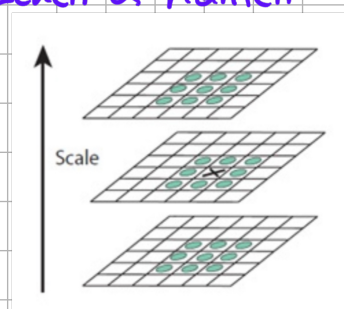
1. Finden von Interest Points - Skalierung:



Octaven nötig, um Extrema zu finden  
Seperierung der Frequenzen in den DoG/  
Laplacebildern

Ziel: Lokalisierung von Ecken & Kanten

2. Finden von Interest Points - Position:



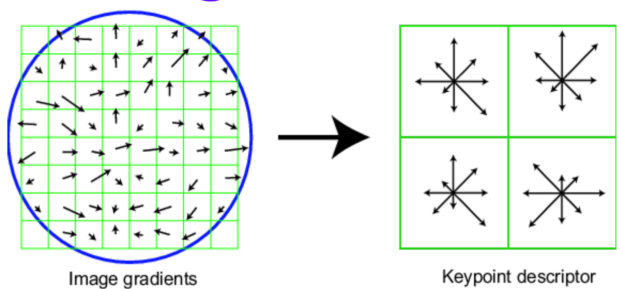
- grobe Lokalisierung der Extrema durch iterative Überprüfung folgender 26 Nachbarn:

- Positionsbestimmung mit Subpixel-Genauigkeit mittels Taylorentwicklungen
- Eliminierung ungeeigneter Interest Points (z.B. solchen an Kanten/mit zu wenig Kontrast) dessen Taylorentwicklung einen Schwellwert unterschreitet

3. Finden von Interest Points - Orientierung

⇒ Zuweisung von Merkmalsvektoren (Gradientenlänge und -richtung des Interest Points), um ihn rotationsinvariant zu halten

## 4. Erstellung einer Beschreibung der Merkmale



blauer Kreis = Gauß'sche Gewichtungsfunktion, die längere Gradienten stärker gewichtet als kürzere

↖ = Merkmalsvektor, sobald normalisiert

zwei Einschränkungen:

- Rotationsabhängigkeit, da es Gradientenrichtungen verwendet
- Beleuchtungsabhängigkeit, wegen der Normalisierung

## Multiskalenanalyse

bei großskaliger Information größere Filter notwendig  
⇒ Rechenaufwand größer, wo es nicht zwingend notwendig wäre, daher:  
Multiskalenanalyse = Verarbeitung in mehreren Auflösungen

## Abtastung

Impulsfunktion/Deltafunktion:  $\delta(x) = 0$  und  $\int_{-\infty}^{\infty} \delta(x) \cdot dx = 1$  für  $x \neq 0$

Abtastung von  $g(x)$ :  $\bar{g}(x) = g(x) \cdot \sum_{i=-1}^N \delta(i-1)$  = Pulsfolge, wenn unendlich:

Kamm/Shah-Funktion:  $S(x) = \sum_{i=-\infty}^{\infty} \delta(x-i)$ , dabei gilt  $f(x) * \delta(x) = f(x)$

⇒ Rekonstruktion der ursprünglichen Signale möglich, solange sich keine

Spektralkomponenten überlappen ⇒ Problem wegen Überlappung, wenn die

Abtastrate zu niedrig ist (Nyquist-Shannon-Theorem) ⇒ Aliasing Effekte

## Bildskalierung

Subsampling: Bildverkleinerung anhand von Gauß-Filter & anschließendes Löschen jeder zweiten Zeile & Spalte

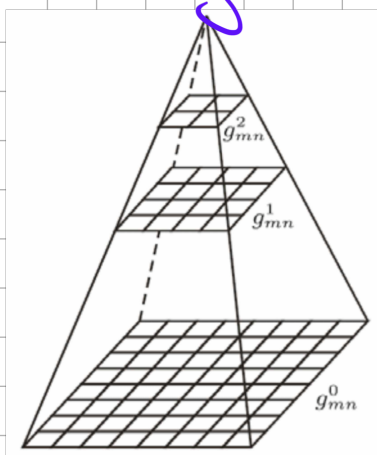
Re-/Upsampling: Bildvergrößerung, indem man neue Werte bilinear oder bikubisch interpoliert

-bilinear: errechnet neue Werte aus ihren Nachbarn

-bikubisch: geht von vergrößerter Matrix aus und rechnet dann ins Eingabebild zurück

## Bildpyramiden

Skalierung ohne Pyramide  $\Rightarrow$  erheblicher Rechenaufwand

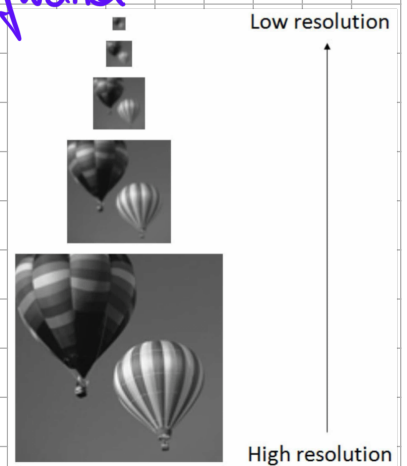


1. Tiefpassfilterung von  $g_{mn}^1$  auf halbe Bandbreite

2. Unterabtastung um Faktor 2

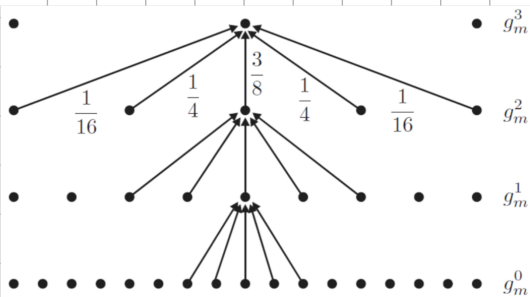
3. Iterative Wiederholung aller

Ebenen bis  $g_{mn}^{v-1}$



$\Rightarrow$  geringerer Rechenaufwand, überschaubarer Speicheraufwand

## Gaußpyramide



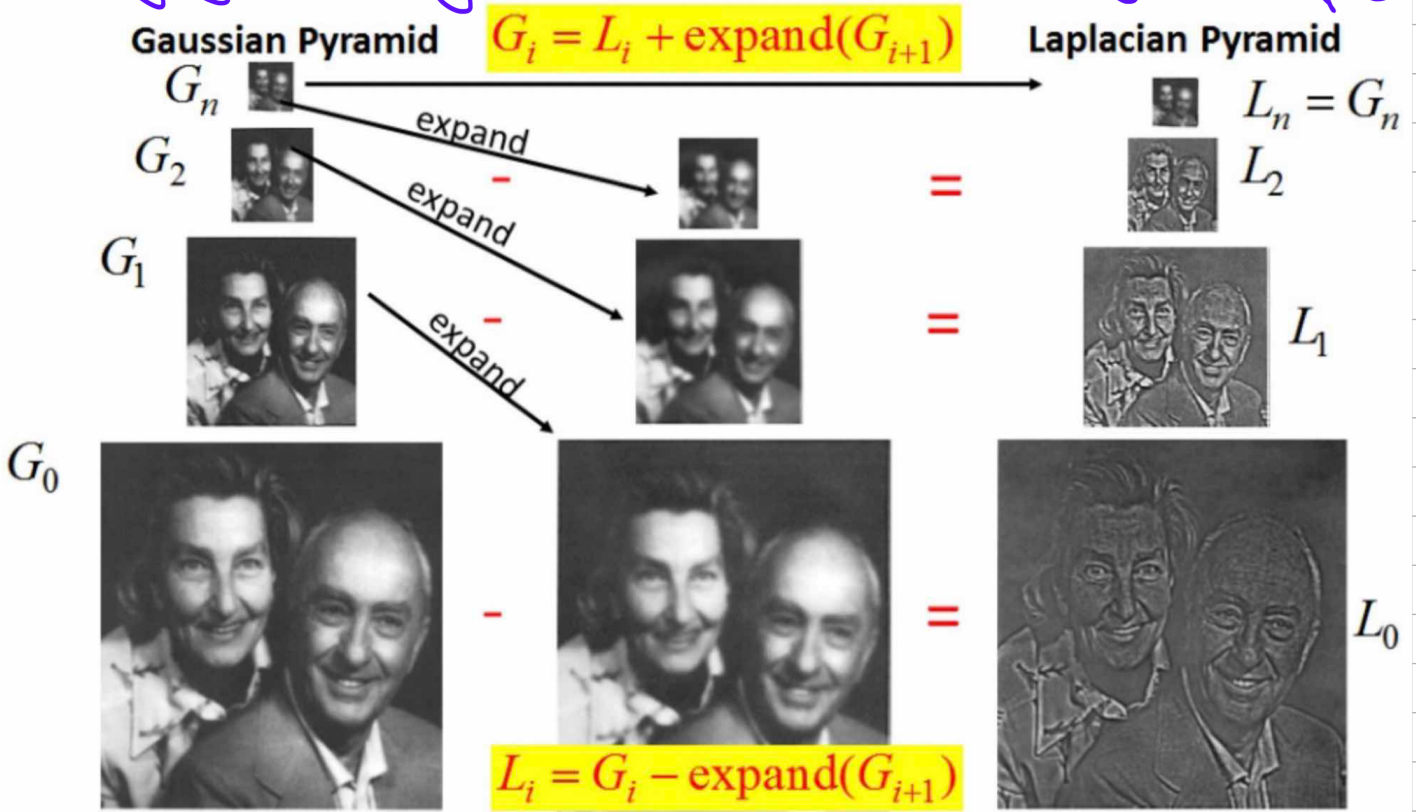
Berechnung der  $v$ -ten Ebene:  $G^{(v)} = G \cdot G^{(v-1)} = B \cdot G^{(v)}$

Gaußpyramide = Serie von tiefpass-gefilterten Bildern, dessen Frequenz immer halbiert wird

# Laplacepyramide

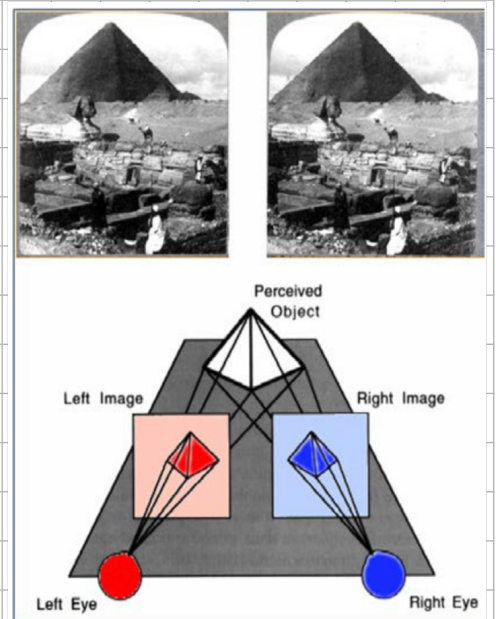
Expansionsoperator / -grad

Pyramidenerzeugung durch Bandpassfilter:  $L^{(v)} = G^{(v)} - \text{expand}(G^{(v+1)})$ ,  $L^{(n)} = G^{(n)}$   
 ⇒ Stellt Approximation der 2. Ableitung mit unterschiedlicher Glättung dar  
 ⇒ Zerlegung des Bildsignals in Bandpassbereiche versch. log. Frequenzstufung



# Stereo Vision

= Erstellung eines Tiefenbildes von zwei Bildern  
 Versatz der beiden Kameras = Disparität  
 Zuweisung von diesen auf Pixel ⇒ Disparitätsmatrix,  
 von welcher sich die Tiefe berechnen lässt  
 größere Disparität = Objekt näher zur Kamera



# Stereoskopie

= Wiedergabe von Bildern mit einem "unechten" räumlichen Tiefeneindruck



# Funktionsweise von Stereoskopen:

1. Abbildung zweier stereoskopischer Halbbilder

2. Wahrnehmung als räumliches Bild durch Parallel- oder Kreuzblick

Vereinfachte Variante: Prismenbrillen

Anaglyphenbild: übereinanderdrucken der Halbbilder in Komplementärfarben

Trennung der Farben in der Brille durch Farbfilter  
bzw. versetzte Polarisationsfilter vor den Projektionsobjektiven  
bzw. entgegengesetzt polarisiertem Licht

Shutter-3D-Systeme: Gläser mit Flüssigkeitsflächen, die steuerbar sind

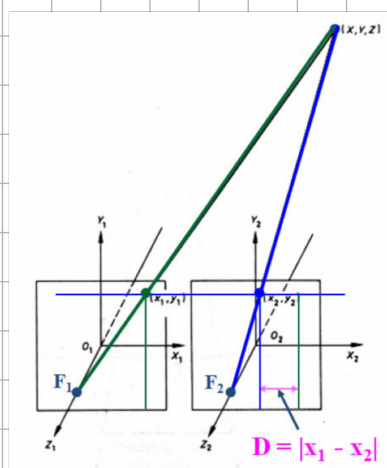
## Normalfall (achsenparalleles Stereosystem)

Basislänge  $B$ : Abstand zwischen optischen Zentren

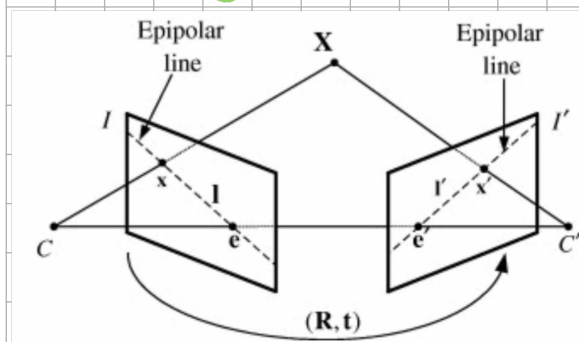
Brennweite  $f$ : Abstand der Brennpunkte zur Bildebene

Disparität  $D$ :  $D = |x_1 - x_2|$

Raumtiefe:  $Z = \frac{B \cdot f}{D}$



## Epipolargeometrie (allgemeine Stereogeometrie)



Epipole:  $e$  und  $e'$

Brennpunkte:  $C$  und  $C'$

Epipolarebenen: die beiden Vierecke

Epipolarlinien:  $l$  und  $l'$

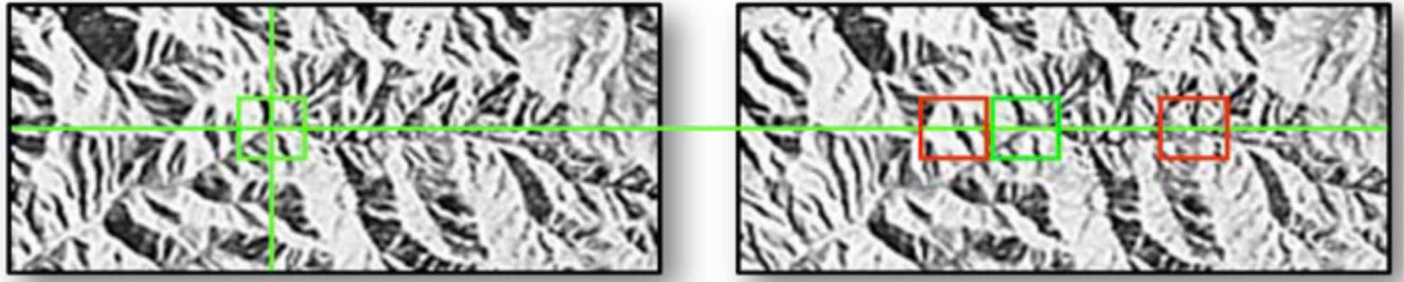
## Korrespondenzproblem

= Aufgabe für jeden linken Bildpunkt einen rechten zu finden, der

denselben Objektpunkt abbildet  $\Rightarrow$  Suchverfahren = Stereo Matching

$\Rightarrow$  Epipolarrektifizierung: Transformation von Stereobildpaaren, sodass zusammengehörende Bildbereiche auf derselben Linie liegen

## Regionsbasiertes Matching



$\Rightarrow$  zu prüfendes Bild wird entlang der Scanline bewegt & Vergleichen

durch Ähnlichkeitsmessung: - Sum of Absolute Differences (SAD)

$\hookrightarrow$  kein eindeutiges

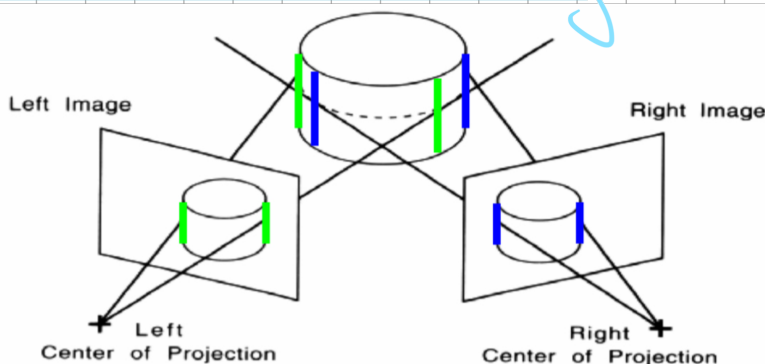
Ergebnis  $\Rightarrow$  Wert nicht vorhanden

- Sum of Squared (SSD)

- Normalized Cross Correlation (NCC)

Problem: homogene Bildbereiche = potentielle Fehlerquellen

## Merkmalsbasiertes Matching



$\Rightarrow$  Extrahierung von Merkmalen

vor dem Vergleich durch

MORAVEC oder SIFT

$\Rightarrow$  schnellerer Vergleich, wegen Daten-

reduktion, aber dafür Tiefeninformation nur für ausgewählte Bereiche vorhanden

$\Rightarrow$  Bereiche zwischen Interest Points benötigen ggf. weitere Verarbeitung

# Structure from Motion

= Gewinnung von 3D-Info durch Bildfolgen

Hauptproblem: Korrespondenzproblem, wobei Kamerageometrie = unbekannt

## Bewegungsfeld

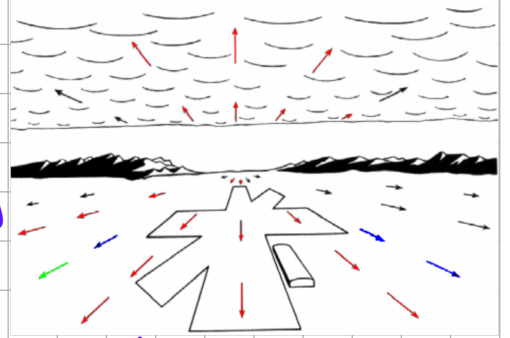
weitere Entfernung vom Betrachter = geringere Bewegung

Richtung des Bewegungsvektors = Lagenabhängig

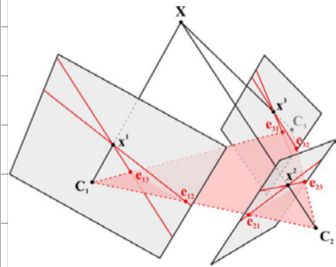
Kamerabewegung ohne Rotation  $\Rightarrow$  nach "außen/innenzeigen" aller Vektoren

zu einem Punkt, dem Focus of Expansion (FoE) / Contraction (FoC)

$\Rightarrow$  führt durch Berechnungen zur Stereokonstruktion



## Multi View Geometry



Berechnung der Geometrie durch mehr als 2 Bildern

Anschließendes Einfügen von mehr

Bündelausgleich = gleichzeitige Bestimmung in- & externer

Kameraparameter, für mögl. gute Übereinstimmung von erwarteten & gemessenen Werten

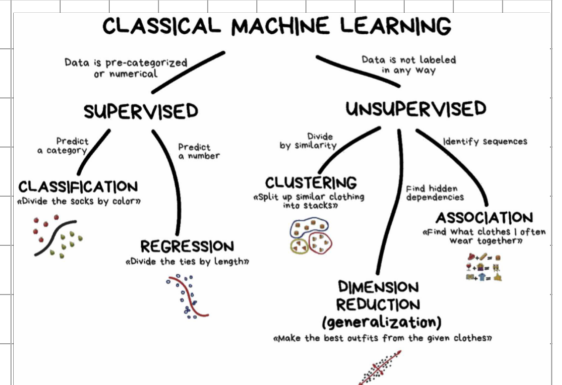
Kollinearitätsgleichung: betrachteter 3D-Punkt, zugehöriger Bildpunkt

& Projektionszentrum der Kamera müssen auf einer Gerade liegen

## Maschinelles Learning

= Methoden, die durch Lernen basierend auf Zusammenhängen Vorhersagen machen können

Ziel (Supervised): mögl. präzise Vorhersagen



Ziel (Unsupervised): Ableitung bisher unbekannter Zusammenhänge & Muster

Ziel (Reinforced): maximaler Nutzen

## Klassifikation anhand von Merkmalen

= häufigste Anwendung von Machine Learning in der Computer Vision

Merkmale = features, Klassifikatoren = classifier  
(2D, 3D, etc)

Aufgabe: bei  $n$  Merkmalen im  $nD$  Raum eine  $(n-1)D$  Hyperfläche finden, die die Klassen möglichst gut trennt, d.h. eine Zielfunktion erstellt

Neighbor Algorithmus: Zuordnung basierend auf Merkmalsentscheid der Nachbarn

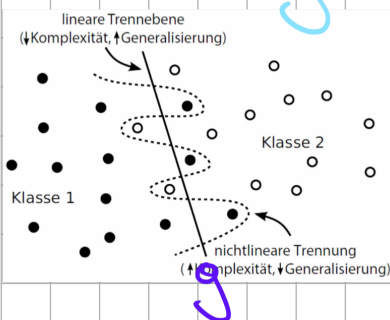
Entscheidungsbäume: Zuordnung anhand ihres Pfades von Wurzel bis Blätter

Random Forrest Algorithmus: Zuordnung durch unabhängiger Entscheidungsbäume

Support Vector Machine: Zuordnung mittels Hyperebene

⇒ Steigerung der Gesamtleistung durch Boosting-Verfahren mögl.

## Generalisierung



Modellkomplexität: mangelnde Flexibilität der Linearität

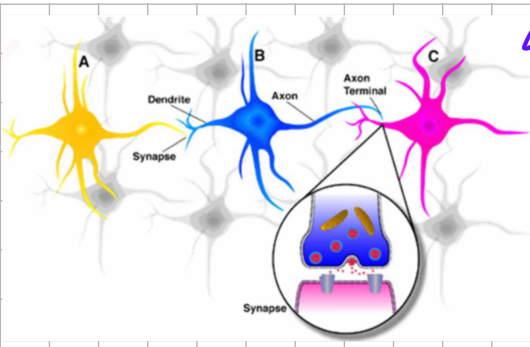
Überanpassung bei zu komplexem Modell

bias = Abweichung der durch schnittlichen Schätzung

von der echten Funktion; variance = Varianz bei unterschiedlichen Lernsätzen

bias-variance-tradeoff: Fehlerminimierung ⇒ Kompromiss zwischen verschiedenen Modellkomplexitäten

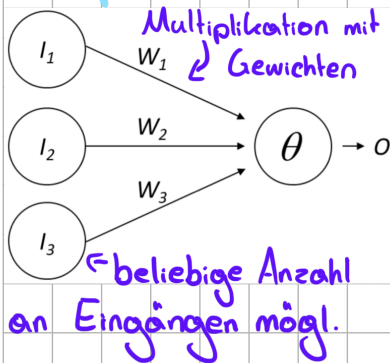
## Künstliche Neuronale Netze (NN)



Aufbau von neuronalen Netzen im Gehirn  
 A, B, C = Zellkörper - quasi ein Kondensator/Akku  
 Neuron feuert sobald Spannung groß genug  
 Synapsen = adaptiv, da gewichtete Kanten

## Perceptron

Aktivierungsfunktion



Multiplikation mit Gewichten  
 Summe der gew. Eingaben:  $\varphi(x) = \sum w_i \cdot I_i \Rightarrow O = f(\varphi)$

Schwelwertsfunktion  $\theta$  nötig, um Konvergenzprobleme zu umgehen:

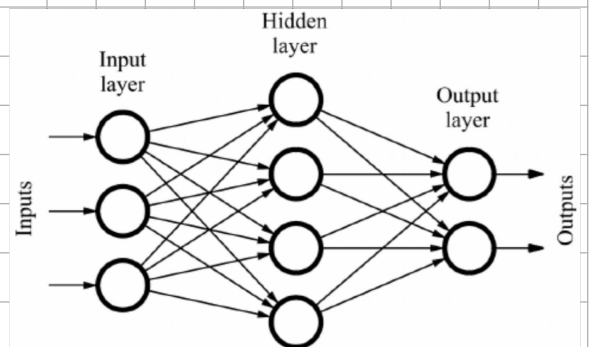
$$O = \begin{cases} 1: (\sum_i w_i \cdot I_i) + \theta > 0 \\ 0: \text{otherwise} \end{cases}$$

Gewichtanpassung:  $\delta$ -Regel / Widrow-Hoff-Regel:  $w_i(t+1) = w_i(t) + \Delta w_i(t)$   
 mit  $\Delta w_i(t) = (T - O) \cdot I_i$ ;  $T$  = gewünschte,  $O$  = tatsächliche Ausg.

Nachteil: Perceptron funktioniert nur bei linear separierbaren Funktionen

## Multilayer-Perceptron

Shortcut = direkte Verbindung zwischen Ein- und Ausgabeschicht



Vorwärtsvermittlung (= feed-forward network):

schichtweise Aktivierung von Ein- zu Ausgabeschicht ohne Rückkopplung

Trainingsverfahren: Fehler-Rückvermittlung / Backpropagation-Training

$\Rightarrow$  Gewichtanpassung in Richtung des abnehmenden Fehlers

Fehler: Distance (LMS) =  $\frac{1}{n} \sum_{p=1}^n (T_p - O_p)^2$ ;  $T_p / O_p = T / O$  des  $p$ -ten Trainingsbsp

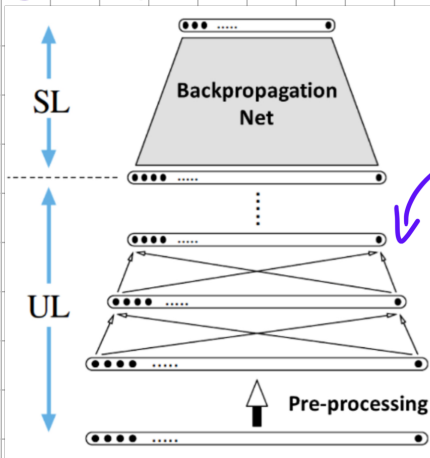
$\Rightarrow$  Fehlersuche von hinten nach vorne bei Backpropagation

# Deep Learning

Exponentielles Wachstum der Trainingszeiten

⇒ End-to-End-Learning: Erlernung von optimaler Merkmalsverarbeitung & Herleitung dieser

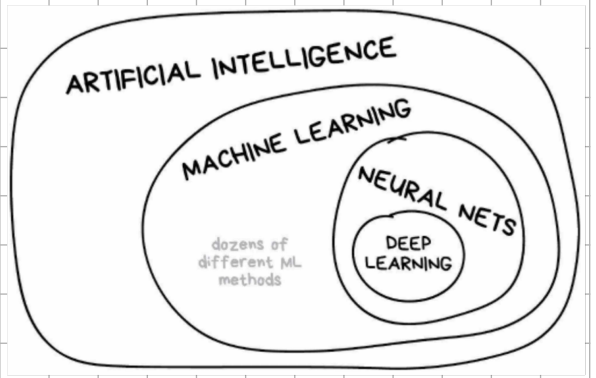
Convolutional Neural Networks (CNN): Erlernung der Filterkoeff. von \*



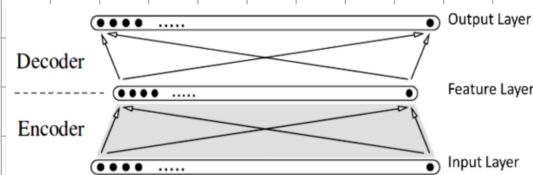
UL = Unsupervised Learning, SL = Supervised L.

tieferere Lage = einfachere Merkmale

Merkmalsextraktion bei Merkmalsbildung, um die Dimension der Eingabedaten ohne Informationsverlust zu verringern



# Auto Encoder



1. Autoencoder lernt identische Abbildung aller Eingabevektoren auf sich selbst

2. nicht benötigte Decoderlage / zweite Lage von Gewichten löschen

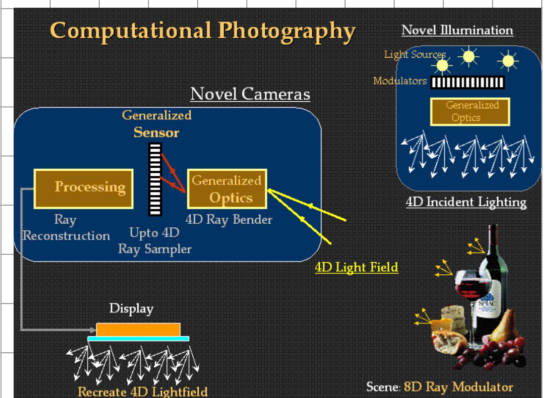
3. erste Lage einfrieren und für Merkmalsberechnung benutzen

4. wiederhole dies bis zur letzten Merkmalschicht

# Computational Photography

= Vereinigung von Computergraphik, Computer Vision & Bildverarbeitung, für neue Funktionalitäten & Anwendung der Photogr.

(rest des Unterkapitels wurde bereits behandelt, siehe vorherige Zusammenfassung)



## Lichtfeld

= Funktion, die Lichtmenge beschreibt, welche auf einen Punkt aus allen Richtungen fällt bzw. welche Strahlen dort reflektiert werden

Parametrisierung mittels 4D-Feld:  $R_i(u_i, v_i, \theta_i, \varphi_i)$  mit  $(u_i, v_i)$  = Position, auf die das Licht auftritt,  $(\theta_i, \varphi_i)$  = Richtung des Lichts

Phenoplastische Kameras: 4D-Kameras, die Ort, Intensität & Einfallswinkel von Lichtstrahlen ermitteln können

größere Winkelauflösung = geringere Ortsauflösung

## High Dynamic Range (HDR)

= Techniken, um größeren Bereich zwischen hellsten & dunklesten Regionen darzustellen, indem es Detailverlust durch Aufnahme mehrerer Bilder versch. Helligkeitsstufen aufnimmt

Tone Mapping: Kontrastreduktion von HDR-Bildern

## Fotomontage/Komposition






= Schneiden & Zusammenfügen mehrerer Fotos zu einem

## Image Inpainting

= Rekonstruktion von schlecht/nicht erhaltenen Bild-/Videoteilen

## Warping

= räumlich signifikante Veränderung jeder im Bild enthaltenen Form

Identity	$W(x) = x$	
Translation	$W(x; t) = x - t$	
Rigid	$W(x; R, t) = Rx - t$	
Similarity	$W(x; R, a, t) = aRx - t$	
Affine	$W(x; A, t) = Ax - t$	

where  $\alpha \in \mathcal{R}$ ,  $A \in \mathcal{R}^{2 \times 2}$ ,  $t \in \mathcal{R}^2$ ,  $R \in \mathcal{R}^{2 \times 2}$ ,  $|R| = 1$

Verwendung: Korrektur von Verzerrungen  
 Abbildung jedes Punktes auf anderen Punkt  
 Warpingfunktion:  $I_d(x) = I_s(W(x; p))$ , mit  
 $I_s = \text{Quell-}$ ,  $I_d = \text{Zielbild}$ ,  $W = \text{Warpingfunktion}$

## Bildmosaik/Stitching

= Kombination mehrerer sich ergänzender Bilder zu einem Panoramabild

1. Bildregistrierung: Bestimmung korrespondierender, lokaler Merkmale
2. Kalibrierung: Differenzminimierung von idealem & realem Linsensystem
3. Blending: Korrektur der im 2. Schritt gefundenen Probleme

## Morphing

= nahtlose Verwandlung von einem Bild in ein anderes durch fading (Aus- & Einblendtechniken) und Verzerrungen