

Bildverarbeitung

- HSI Farbmodell

- $c = \arccos \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}}$
- $H = c$ falls $B < G$, $360^\circ - c$ sonst
- $S = 1 - \frac{3}{R + G + B} \cdot \min(R, G, B)$
- $I = \frac{1}{3}(R + G + B)$

- Affiner Punktoperator: $f: [0..q] \rightarrow [0..q], I'(u, v) = a \cdot I(u, v) + b$

- Kontrasterhöhung: $b = 0, a > 1$
- Kontrastverminderung: $b = 0, a < 1$
- Helligkeitserhöhung: $b > 0, a = 1$
- Helligkeitsverminderung: $b < 0, a = 1$
- Invertierung: $b = q, a = -1$ (meist $q = 255$)

- Schwellwertfilter

- $I'(u, v) = \begin{cases} 255 & \text{falls } I(u, v) \geq s \\ 0 & \text{sonst} \end{cases}$

- Histogramme von Graustufenbildern

- Histogramm aufstellen
 - Histogrammfunktion: gibt die Häufigkeit eines Merkmals (meist Grauwert) an.
 - for $x := 0$ to q do
 $H(x) := 0$
 end for
 - for all pixels (u, v) in I do
 $H(I(u, v)) := H(I(u, v)) + 1$
 end for

Bei einem 8Bit Graustufenbild ist $q = 2^8 - 1 = 255$

Die Histogrammfunktion mappt also einen Wert einer Bildmatrix auf die Häufigkeit, wie oft er im Bild vorkommt.

ACHTUNG: Die Funktion mappt auf Null für alle nicht vorkommende Werte!

- p-Quantil

Bei dem p-Quantil ($p \in [0, 1]$) und n Elementen gilt $\tilde{x}_p = \begin{cases} \frac{1}{2}(x_{np} + x_{np+1}) & \text{falls } n \cdot p \text{ ganzzahlig} \\ x_{[np]} & \text{sonst} \end{cases}$

- Akkumuliertes Histogramm

- $H_a(0) := H(0)$
 for $x := 1$ to q do
 $H_a(x) := H_a(x-1) + H(x)$
 end for

- Das akkumulierte Histogramm eines Histogramms H unter dem Wert x entspricht der Summe aller Histogrammwerte einschließlich x .

- Histogrammspreizung

- Intensitätsintervall $[min, max]$ wird auf $[0, 255]$ gespreizt. Schlechtere Variante der Histogrammdehnung

- $I'(u, v) = q \cdot \frac{I(u, v) - min}{max - min}$ mit $q = 255$

- Histogrammdehnung

- Bessere Variante der Spreizung
- Verwendet Quantile statt min und max zur Spreizung. Mit alternativ berechneten min/max dann normale Spreizung durchführen.

- Histogrammausgleich

- Histogrammausgleich erhöht den Kontrast in Bereichen des Histogramms mit vielen Treffern. Stark vertretene Grauwertbereiche werden für das Auge sichtbarer gemacht.
- Histogrammausgleich ist ein homogener, nicht-affiner Punktoperator.
- $H \leftarrow \text{BerechneGrauwertHistogramm}(I)$
 $H_a \leftarrow \text{BerechneAkkumuliertesHistogramm}(H)$

```
for x := 0 to q do
    H_n(x) := round( q * (H_a(x) / H_a(q)) )
end for
```

```

for all pixels (u,v) in I do
    I'(u,v) := H_n(I(u,v))
end for

```

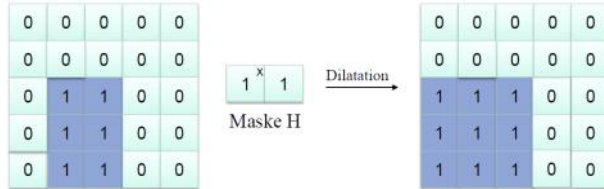
- Andere (eigene) Formulierung:

$$H_n = \text{round} \left(\frac{q}{H_{a(q)}} \cdot H_a \right) \left(= \text{round} \left(\frac{255}{H_a(255)} \cdot H_a \right) \right) \text{ (Ganze Abbildung wird modifiziert)}$$

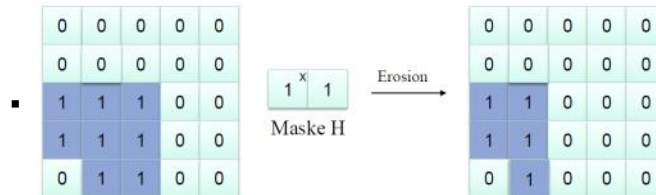
$I' = H_n(I')$ (Bild wird komplett mit H_b gemappt)

- Morphologische Operationen

- Dilatation: Objekt wird aufgeblasen, Pixel zu größeren Bereichen vergrößern. Prüfe an jeder Position, ob $H \cap B \neq \emptyset$



- Erosion: Schrumpfen eines Objektes, Schwache Pixelgruppen werden entfernt. Prüfe an jeder Position, ob $H \subseteq B$



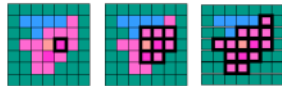
- Opening: Erosion, dann Dilatation. Dünne Strukturen werden entfernt.
- Closing: Dilatation, dann Erosion. Lücken werden aufgefüllt.
- Region Growing

Gegeben: Graustufen-Bild

Gesucht: Zusammenhängende Regionen

Algorithmus in Pseudocode:

- Wähle Saatpunkt $p_0 = (u_0, v_0)$
- Initialisiere Region $R = \{p_0\}$, wähle Schwelle ε
- while $\exists p \in R, q \notin R$ mit $\|p - q\| \leq 1$ und $|\text{Img}(p_0) - \text{Img}(q)| \leq \varepsilon$
 $R = R \cup \{q\}$



- Also: Wähle einen Startpunkt und Schwelle, und füge alle Punkte dazu, die direkt an die bisherige Region anliegen und deren Farbdifferenz zum Startpunkt kleiner als der Schwellwert ist.

- Filterarten

- Tiefpassfilter: Bandbegrenzung (Anti-Aliasing/Glättung/Rauschelimination)
 - Mittelwertfilter, Gauß-Filter
- Hochpassfilter: Kantendetektion
 - Prewitt, Sobel, Laplace, Roberts
- Kombinierte Operatoren
 - Laplacian of Gaussian

- Tiefpassfilter: Rauschunterdrückung

- Mittelwertfilter

$$\frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Gauß

- Approximation der Matrixgröße zur Filterung im Ortsbereich
 - $n = \lfloor 2\sigma \rfloor \cdot 2 + 1$, Matrix der Größe $n \times n$ wird gebraucht.

$$F_{\text{Gauß}} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

- Hochpassfilter: Kantendetektion

- Canny
 - Rauschunterdrückung mit Gauß
 - Gradient in horizontal+vertikaler Richtung (Prewitt_{xy} oder Sobel_{xy})
 - Non-Maximum Suppression
 - Hysterese-Schwellwertverfahren
- Prewitt

$$\blacksquare p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, p_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- Sobel

$$\blacksquare s_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, s_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

- Roberts

$$\blacksquare R_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, R_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, (\text{diagonale Kantendetektion})$$

- Laplace (sehr empfindlich gegen Störungen)

$$\blacksquare \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Laplace of Gaussian (LoG)

$$\blacksquare \begin{pmatrix} & & -1 & & \\ & -1 & -2 & -1 & \\ -1 & -2 & 16 & -2 & -1 \\ & -1 & -2 & -1 & \\ & & -1 & & \end{pmatrix}$$

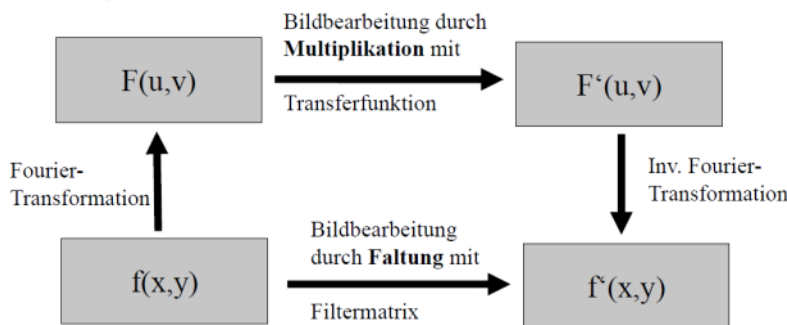
- Bildmatrix B mit Filtermatrix F falten

- F mit ihrem Mittelpunkt auf einen Punkt in B gelegt. Die Werte von F werden mit den darunterliegenden Werten von B multipliziert. Dann wird die Filtermatrix durch Addieren der Werte reduziert, und das Ergebnis überschreibt den alten Wert in B. Das wird für alle erforderlichen Werte gemacht.

- Faltung durch Filtermatrix im Frequenzbereich erreichen

- Bildmatrix und Filtermatrix fourier-transformieren
- Beide fouriertransformierten Matrixen multiplizieren
- Ergebnismatrix rücktransformieren

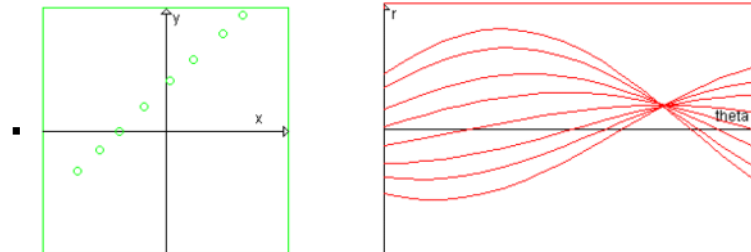
- Zusammenhänge



- Wieso ist es sinnvoll, bei Gauß-Filtern starke Glättungen im Frequenzbereich durchzuführen?

- Hough-Transformation

- Um aus einem Kantendetektierten gefilterten Bild die Kantenlinien zu finden, werden die Bildpixel nach Hough transformiert. Bildpunkte werden als Punkt in Polarkoordinaten dargestellt. Ein Kantenpixel entspricht dann einer Sinusoidfunktion im Parameterraum, ein Punkt im Parameterraum entspricht einer Geraden im Bildraum.



- Umformung eines Bildpunktes nach Hough-Transformation: Punkt darstellen als mit Winkel $\alpha = \tan^{-1} \frac{y}{x}$ (Satz des Pythagoras) und Radius $r = x \cos \alpha + y \sin \alpha$
- Die Sinusoidfunktionen, die dann im Houghraum verwendet werden, sind durch r beschrieben.

- Korrelationen zwischen Bildmatrixen

Die Formeln gelten für zwei Bildmatrixen Img_1 und Img_2 der Größe $(2n + 1) \times (2n + 1)$ mit Bildmittelpunkten (u_i, v_i)

- Sum of Absolute Differences (SAD). Minimal bei guter Übereinstimmung.

$$\blacksquare \sum_{i=-n}^n \sum_{j=-n}^n |(Img_1(u_1 + i, v_1 + j) - Img_2(u_2 + i, v_2 + j))|$$

- Sum of Squared Differences (SSD). Minimal bei guter Übereinstimmung.

- $$\sum_{i=-n}^n \sum_{j=-n}^n \left(\text{Im}g_1(u_1 + i, v_1 + j) - \text{Im}g_2(u_2 + i, v_2 + j) \right)^2$$
- Zero Mean Normalized Cross-Correlation. Maximal bei guter Übereinstimmung.
 - ≈ 1 wenn alle Werte nur um eine Konstante verschoben sind oder mit einer Konstante vervielfacht wurden.
 - Der Wert ist zwischen 0 und 1. 1 = Stimmt überein, 0 = keine Übereinstimmung.
- Kamerasystem
 - Umrechnung Bildpunkte zu Kamera-/Weltkoordinaten
 - $(X, Y, Z) = x_c$ sind Kamerakoordinaten, x_w sind Weltkoordinaten
 - Abbildung vom Kamerakoordinatensystem in das Bildkoordinatensystem:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \frac{1}{z} \cdot \begin{pmatrix} f_x \cdot X \\ f_y \cdot Y \end{pmatrix} \text{ oder } \begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = K \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$
 - Abbildung Weltkoordinatensystem in das Kamerakoordinatensystem

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = x_c = R \cdot x_w + t \text{ (Rotation und Translation anwenden)}$$
 - Punkt P bildet auf Bildpunkt (u,v) ab. Transformation aus Rotation R und Translation T beschreibt Transformation von Weltkoordinaten in Kamerakoordinaten. Beschreibe Gerade aller Punkte, die auf Bildpunkt (u,v) abbilden.
 - $g: x = T + r \cdot (P - T)$
 - Invertiere Transformationsmatrix $H = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$
 - $H^{-1} = \begin{pmatrix} R^T & -R^T t \\ 0 & 1 \end{pmatrix}$
 - Bildpunkt (u,v) ist gegeben, sowie K. Beschreibe Gerade aller Punkte im Kamerasystem, die auf den Bildpunkt abbilden.
 - $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = r \cdot K^{-1} \begin{pmatrix} u \cdot 1 \\ v \cdot 1 \\ 1 \end{pmatrix}$ (Also Formel von oben umformen, aber setze w=1)
 - Warum schneiden sich in realen Situation derartige Linien nicht?
 - Ungenauere Stereokalibrierung
 - Linsenverzerrung
 - Pixel-Diskretisierung
 - Ungenauere Lokalisierung der korrespondierenden Punkte
- Epipolargeometrie
 - Vorteile im Bezug auf das Korrespondenzproblem: +Effizienz, +Robustheit
- Fundamentalmatrix
 - Die mathematische Beschreibung der Epipolargeometrie.
 - 3x3 Matrix von Rang 2, für alle Korrespondenzen x, x' gilt $x'^T F x = 0$
 - Epipolarlinien liegen bei $l = F^T x', l' = F x$
 - Epipole liegen bei $F e = 0, F^T e' = 0$
 - Berechnung über Bildpunkt Korrespondenzen
 - $x'^T F x = 0$
 - Berechnung über Essentialmatrix
 - Essentialmatrix wird über extrinsische Parameter berechnet

$$E = [t]_x R = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} R$$
 - Für Epipole gilt $e = -K R^T t, e' = K' t$
 - $F = K'^{-T} E K^{-1}$
 - Rektifizierung: Bewirkt, dass alle Epipolarlinien horizontal mit derselben v-Koordinate wie der Bildpunkt im anderen Kamerabild verlaufen. Dann müssen nach Korrespondenzen nur noch in eine Richtung gesucht werden.
- Quaternionen
 - Quaternion
 - $q = (q_w, q_x, q_y, q_z) (1, i, j, k)^T = (q_w, q_v) (1, i, j, k)^T = q_w + i q_x + j q_y + k q_z$
 - $i^2 = j^2 = k^2 = -1; ij = -ji = k; jk = -kj = i; ki = -ik = j$
 - Rechenregeln für Quaternionen
 - $q + r = (q_w, q_v) + (r_w, r_v) = (q_w + r_w, q_v + r_v)$
 - $q \cdot r = (q_w r_w - q_v \cdot r_v, q_v \times r_v + q_w r_v + r_w q_v)$
 - $\bar{q} = (q_w, -q_v)$ für $q = (q_w, q_v)$
 - Norm: $N(q) = \sqrt{q \bar{q}} = \sqrt{\bar{q} q} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}$
 - Multiplikative Identität: $I = (1, 0)$
 - Multiplikative Inverse: $q^{-1} = \bar{q} / N^2(q)$
 - Punktrotation
 - Vektor v ist zu rotieren um Winkel ϕ um die Achse u

- Das Ergebnis ist $qa\bar{q}$ mit $a = (0, v)$ und $q = \left(\cos \frac{\phi}{2}, u \cdot \sin \frac{\phi}{2}\right)$
- Nicht vergessen Winkel in q zu halbieren!
- Quaternionen aus beliebiger Rotationsmatrix m_{ij} berechnen

$$q_w = \frac{1}{2} \sqrt{1 + \sum_{i=1}^3 m_{ii}}$$

$$q_x = \frac{m_{32} - m_{23}}{4q_w}$$

$$q_y = \frac{m_{13} - m_{31}}{4q_w}$$

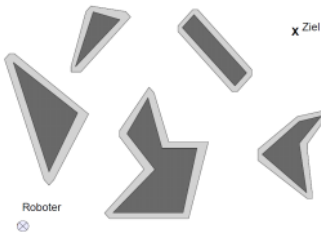
$$q_z = \frac{m_{21} - m_{12}}{4q_w}$$

Planung

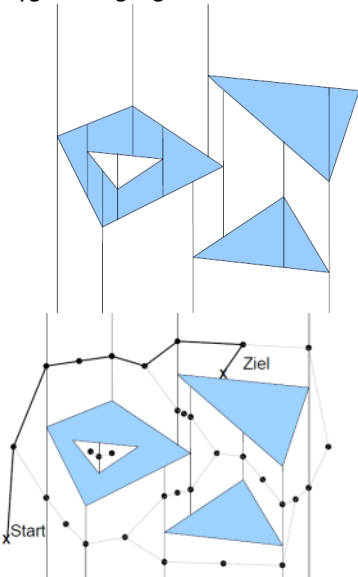
- Raumzerlegungen
 - Freiraum, Hindernisraum



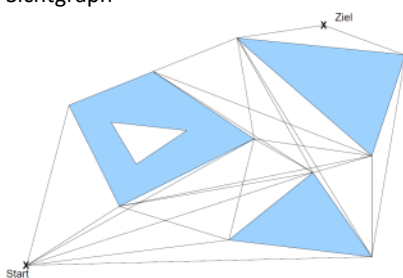
- Konfigurationsraum



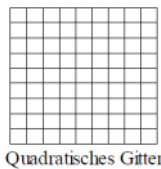
- Polygonzerlegung



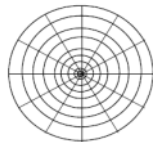
- Sichtgraph



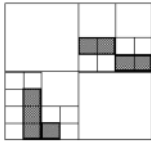
- Gitter



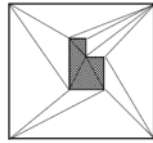
Quadratisches Gitter



Sphärisches Gitter

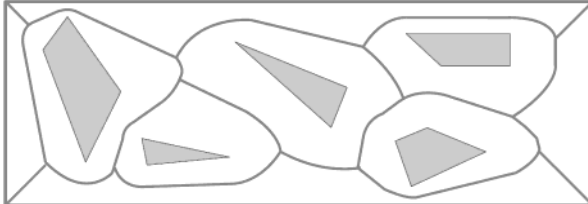


Hierarchisches Gitter

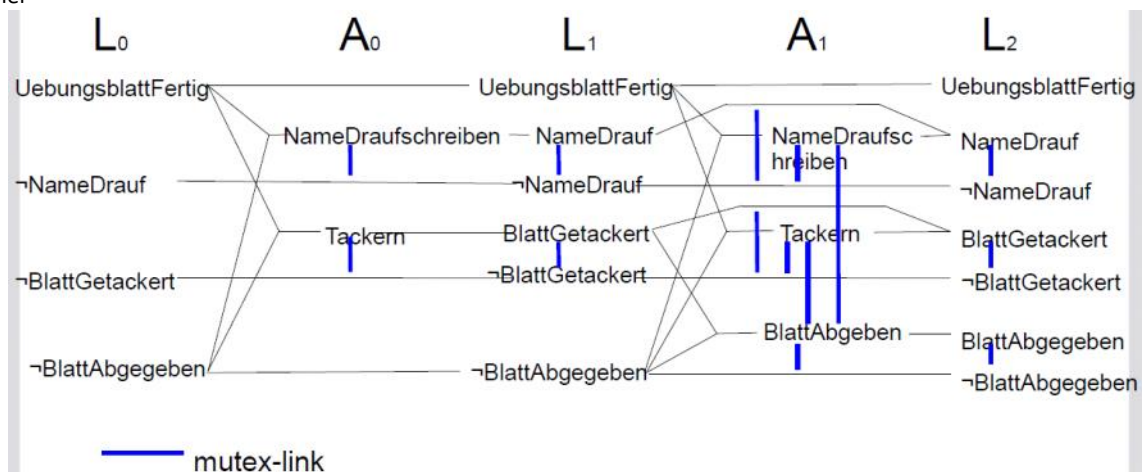


Beliebige Gitterstruktur

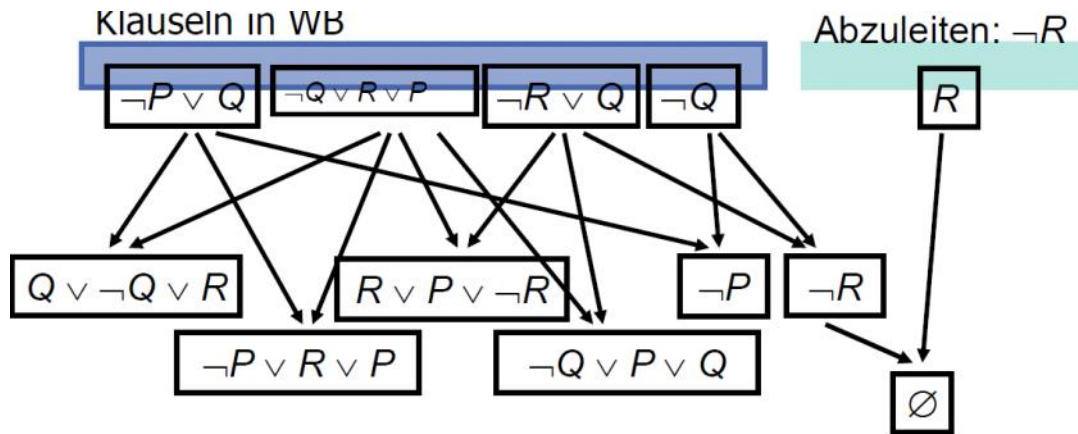
○ Voronoi-Diagramm



- Objektmodellierung: Wie können Objekte modelliert werden?
 - Kantenmodelle, Oberflächenmodelle, Volumenmodelle
- A* Suche
 - Expandiert den Knoten, für den (*geschätzte Zielentfernung* + *bisher zurückgelegte Strecke vom Start*) von allen Knoten des Baums minimal ist.
 - Heuristik aus den Übungsblättern: Geschätzte Zielentfernung ist der direkte Weg vom aktuellen Knoten zum Zielknoten.
 - Bei der Suche wird eine Baumstruktur erzeugt, die im Startknoten die Wurzel hat.
 - Tipp: An jedem Knoten, ergänze die Rechnung $\text{ZuvorEntfernung} + \text{HeuristikZielEntfernung} = \text{HeuristikGesamtEntfernung}$
 - Anforderung an die Heuristik: Sie darf Distanzen unter-, aber nicht überschätzen, damit der gefundene Pfad optimal ist.
- Planungsgraphen erstellen
 - Mutex-Links
 - Zwei Literale schließen sich aus, wenn sie komplementär sind (nicht P und P) oder nur aus sich gegenseitig ausschließenden Aktionen resultieren können.
 - Zwei Aktionen schließen sich aus, wenn ihre Effekte ein komplementäres Literal enthalten, durch den Effekt der einen eine Vorbedingung der anderen zerstört wird, oder die Vorbedingungen der einen ein Literal enthalten, das ein Literal aus den Vorbedingungen der anderen ausschließt.
 - Abschnitte
 1. Alle Literale des Startzustandes
 2. Alle Aktionen, die möglich sind (auch enthaltene Aktionen), sich gegenseitig ausschließende Aktionen durch Mutex-Links verbinden.
 3. Alle Literale, die aus den Aktionen resultieren. Sich gegenseitig ausschließende Literale durch Mutex-Links verbinden.
 4. ... bis Literale sich nicht mehr ändern
 - Beispiel



- Logik
 - Eine Logik ist definiert als $L = \{\text{Symbolmenge}, \text{Belegungsmenge}, \text{Syntax}, \text{Semantic}, \text{Folgerungsoperator}\}$
- Resolutionsalgorithmus
 - Klauseln müssen durch KNF gegeben sein
 - In jedem Resolutionsschritt kann nur eine Variable entfernt werden.
 - ACHTUNG: Die zu beweisende Klausel muss **negiert** dazugefügt werden.



- Horn-Klauseln
 - Horn-Klausel: **Disjunktion von Literalen, von denen höchstens eins positiv ist** (lässt sich zu Implikation umformen)
 - Und-Oder-Graph:

$$\cdot A \rightarrow B \quad A \Rightarrow B$$

$$A \wedge B \Rightarrow C$$

- Rückwärtsverkettung
 - Man geht von dem zu überprüfenden Literal rückwärts durch den Baum und prüft, ob alle Vorbedingungen stimmen. Diese stimmen genau dann, wenn sie als Fakt definiert sind, also eine Hornklausel aus nur positiven Literalen existiert, die das Literal der jeweiligen Vorbedingung enthält.
- Typen von Horn-Klauseln
 - *Fakt, Axiom*: Enthält keine negativen Literalen
 - *Definition*: Enthält genau ein positives Literal
 - *Integritätseinschränkung*: Enthält kein positives Literal
- DPLL
 - Rekursiv die Klauselmengende durcharbeiten. In jedem Rekursionsschritt alle Klauseln aufschreiben. In jedem Schritt die erstmögliche Operation durchführen:
 - Einheitsklausel: Finde Klausel aus nur einer Variable, entferne sie und alle Klauseln die die Variable mit korrekter Belegung enthalten (sonst nur das jeweilige Literal in der Klausel), setze Belegung entsprechend der Einheitsklausel.
 - Reines Literal: Finde Variable, die überall mit derselben Belegung vorkommt. Entferne alle Klauseln, die diese Variable enthalten und setze Belegung entsprechend der Variable.
 - Sonst: Nehme erstbeste Variable, teile den Baum in zwei Pfade auf und setze bei jedem Pfad eine andere Belegung. Jeweils Klausel entfernen.
- STRIPS, ADL
 - Aktionen bestehen aus Aktionsname, Parameter, Vorbedingungen und Effekte.
 - STRIPS basiert auf closed-world-Assumption, ADL dagegen auf open-world-assumption

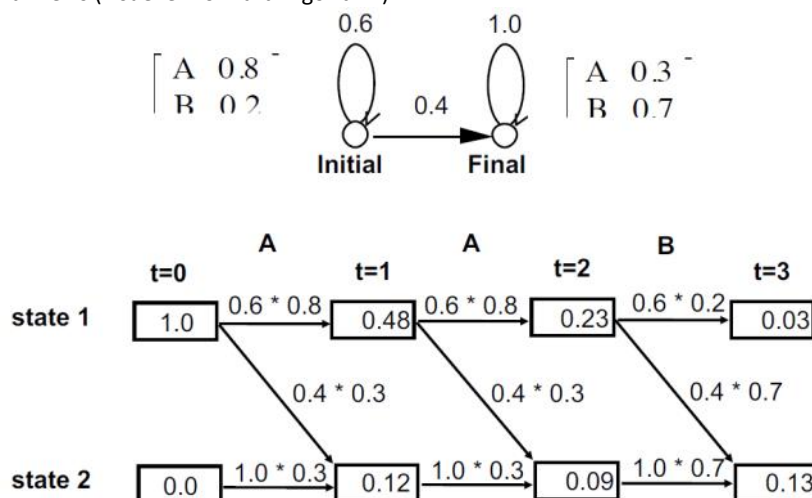
Faltung

- Analytische Faltung
 - $(f * g)(t) = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau$
- Zeichnerische Faltung
 - Eine Funktion wird gespiegelt und bewegt sich über die andere, Flächen werden Multipliziert.
 - Tipp: Wenn sich zwei Boxen falten, das Zentrum der einen Box um a auf der x-Achse verschoben ist und das Zentrum der anderen auf der x-Achse um b verschoben ist, so hat die gefaltete Funktion den Maximalwert bei a+b.
- Die Faltung zweier Funktionen entspricht dem Filtern eines Signals beschrieben durch die erste Funktion mit einem Filter, der durch die Impulsantwort der zweiten Funktion beschrieben wird.
- Abtasttheorem / Nyquist-Shannon Theorem
 - Die Abtastfrequenz muss größer als die 2-fache Grenzfrequenz (höchst vorkommende Frequenz) sein, damit das Originalsignal aus dem abgetasteten Signal wiederhergestellt werden kann. Ansonsten können sich Frequenzen überlagern oder verfälschen (Aliasing)
 - Lösungsmöglichkeiten: Erhöhung der Abtastfrequenz (aber höhere Datenmengen) oder Tiefpassfilter (aber Informationsverlust)
- Spektrogramm
 - Zeitlicher Verlauf der im Signal vorkommenden Frequenzen, bzw. zeitliche Abfolge von Spektren
 - Ein Spektrum beschreibt vorhandene Frequenzen im Signal in einem Zeitintervall.
 - Das Spektrum eines Zeitintervalls erhält man durch Abtastung, Quantisierung und diskreter Fouriertransformation eines Ausschnittes des Signals.
- Faltungstheorem
 - Die Faltung im Zeitbereich entspricht der Multiplikation im Frequenzbereich.
- Fouriertransformation
 - Gebe die Impulsantwort des Systems

- Gegeben ist ein lineares System $h(t)$ mit einem Eingangssignal $u(t)$ und einem Ausgangssignal $y(t)$ sowie der Zusammenhang:
 $a_1 y(t) + a_2 y'(t) + a_3 y''(t) + \dots = u(t)$
- Außerdem gilt: $F(x^{(n)}(t)) = (j\omega)^n X(\omega)$
- Dann lässt sich herleiten:
 - $a_1 Y(\omega) + a_2 (j\omega)Y(\omega) + a_3 (j\omega)^2 Y(\omega) + \dots = U(\omega)$
 - Impulsantwort $H(\omega) = \frac{Y(\omega)}{U(\omega)} = \frac{Y(\omega)}{a_1 Y(\omega) + a_2 (j\omega)Y(\omega) + a_3 (j\omega)^2 Y(\omega) + \dots} = \frac{1}{a_1 + a_2 (j\omega) + a_3 (j\omega)^2 + \dots}$
- Zeichne das Betragsspektrum für die Impulsantwort $H(\omega)$ für ein gegebenes a. Ist es ein Tiefpass- oder Hochpassfilter?
 - Fülle a in der Formel von H ein, bilde davon den Betrag und zeichne den Graphen.
 - Tief oder Hochpass: TODO

Klassifikation

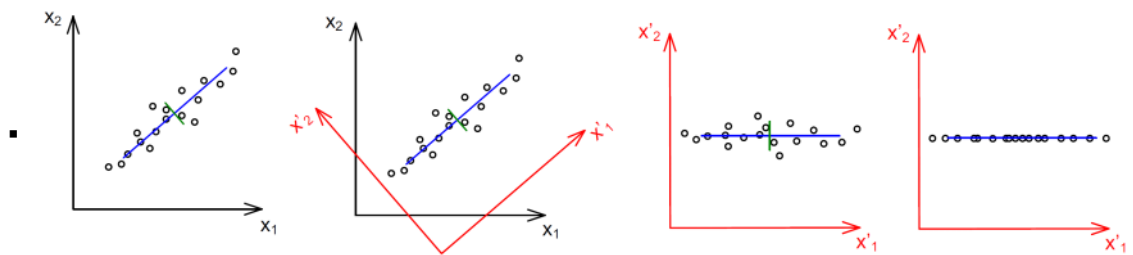
- A-Priorio Wahrscheinlichkeit
 - Wahrscheinlichkeit vor dem Test, $P(\omega_j)$
- A-Posteriori Wahrscheinlichkeit
 - Nach Bayes-Regel: $P(\omega_j | x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$, $p(x) = \sum_j p(x|\omega_j)P(\omega_j)$
 - ω_j sind die möglichen Ergebnisse, z.B. krank oder gesund. $P(\omega_j)$ sind die Apriori-Wahrscheinlichkeiten, $p(x|\omega_j)$ sind die Wahrscheinlichkeiten unter Betrachtung des Tests.
- Naiver Bayes-Klassifikator
 - $y = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k)$
 - argmax: Wo wird die Funktion f maximal?
$$x_{\max} = \operatorname{argmax}_{x \in D} f(x) \Leftrightarrow f(x_{\max}) = \max_{x \in D} f(x).$$
- Fehlerwahrscheinlichkeiten berechnen
 - Schwellwert, mit dem die Fehlerwahrscheinlichkeit bei der Klassifikation minimal wird
 - Zeichne grafisch die Funktionen $p(x|w_i)P(w_i)$ (also nach Bayes, A-Priori-Wahrscheinlichkeiten multipliziert mit Wahrscheinlichkeitsdichtefunktionen), Schnittpunkt ist die Position des Schwellwertes.
 - Minimale Fehlerwahrscheinlichkeit, die durch geeignete Wahl des Schwellwerts ermöglicht wird
 - $P(\text{Fehler}) = \sum_i P(w_i) \cdot \int_{a_i}^{b_i} p(x|w_i) dx$
 - (Wenn nur zwei Klassen existieren) Für w_1 gehen die Grenzen vom Schwellwert bis Unendlich, für w_2 gehen die Grenzen von minus Unendlich bis zum Schwellwert.
- Hidden-Markov Models
 - Markov-Annahme: Übergangswahrscheinlichkeit hängt nur vom aktuellen Zustand ab.
 - Aufgabenstellungen
 - Evaluation. Berechne Wahrscheinlichkeit einer Ausgabefolge mit gegebenen Model: Forward, Viterbi
 - Decoding. Finde Zustandsfolge, welche die Wahrscheinlichkeit einer gegebenen Ausgabefolge maximiert: Viterbi
 - Training. Passe Modellparameter an um Wahrscheinlichkeit der gegebenen Ausgabefolge zu maximieren: Forward - Backward
 - Forward Trellis (visueller Forward Algorithm)



Achtung: Forward muss i.d.R. im Endzustand enden (ist meist in der Aufgabenstellung gegeben)

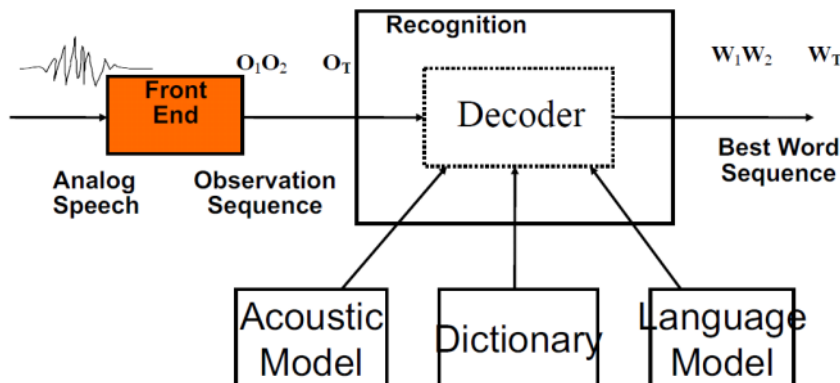
- Viterbi
 - Wie Forward, nur verwende MAX statt SUM als Reduzierer beim Eingang mehrerer Pfade.
 - Um zu sehen, welche Zustandsfolge traversiert wird, schauen welcher Eingang durch MAX verwendet wurde.
- Nicht parametrische Verfahren
 - Parzen-Windows

- Keine Annahme über Verteilung, $p(x)$ wird direkt aus den Daten geschätzt. Suche ein Fenster des Volumens V im Datenbestand, zähle die Samples darin, dann schätze $p(x) \approx \frac{k/n}{V}$ (mit k Anzahl der Samples im Fenster, n Anzahl der Gesamtsamples, V Volumen des Fensters). Je größer V , desto schwächer ist die Auflösung. Je kleiner V , desto schwächer ist die Schätzung.
- Backpropagation-Algorithmus (Zum Trainieren von Feed-Forward neuronalen Netzen)
 - Warum kann eine Funktion der Form $f(x) = 1$ falls $x > \Phi$, sonst 0 nicht als Aktivierungsfunktion verwendet werden?
 - Funktion ist nicht stetig bei $x = \Phi$ und daher dort nicht differenzierbar. Backpropagation minimiert den Fehler zwischen erwarteten und tatsächlichen Netzausgaben, zur Berechnung des Fehlergradienten muss die Ableitung der Aktivierungsfunktion berechnet werden.
 - Die Treppenfunktion wird durch eine *differenzierbare Sigmoidfunktion* angenähert.
 - Parameter beim Entwurf von neuronalen Netzen
 - (Lernalgorithmus, Aktivierungsfunktion,) Netztopologie, Lernparameter, Ein-/Ausgabecodierung, Initialgewichte
 - Problem: Es wird nur ein lokales Minimum der Fehlerfunktion gefunden (Overfitting)
- Generalisierungsfähigkeit: Fähigkeit eines Klassifikators, Daten außerhalb des Trainingsdatenbestandes korrekt zu klassifizieren.
- Fluch der Dimensionalität: Mit steigender Dimensionalität steigt die notwendige Menge an Trainingsdaten exponentiell.
- Praktische Probleme bei Klassifikatoren:
 - Begrenzte Trainingsdaten/Rechenressourcen, Teures Labeling, unbekannte Klassen, Gute Features können unbekannt sein
- Principal Component Analysis (PCA)
 - Reduziere Dimensionalität der Features durch Raumrotation an Datenbestand, dann Entfernung von Dimensionen mit geringer Varianz.



Spracherkennung

- Aufbau eines Spracherkenners



- Frontend/Vorverarbeitung
 - Anti-Aliasing Filter: Aliasing bei falscher Abtastrate umgehen
 - Analog-Digital Umwandlung: Analoges Signal wird digitalisiert
 - Fenstern: Unendliches Signal für Fouriertransformation wird simuliert durch Vervielfältigung eines Fensters des Signals.
 - FFT: Fast Fourier Transform, Spektrum aus Signal extrahieren.
- Decoder: Sucht nach der wahrscheinlichsten Wortfolge, die das gegebene Signal erzeugt hat, entsprechend der Signalformel (Fundamentalformel)
- Akustisches Modell: Meist HMMs
- Dictionaries: Wort-, phonische oder Baumstrukturierte Dictionaries
- Sprachmodell: zB. Grammatiken, N-Grams
- Fundamentalformel der Spracherkennung
 - $\hat{W} = \operatorname{argmax}_W P(W|X) = \operatorname{argmax}_W \frac{P(X|W) \cdot P(W)}{P(X)}$
 - \hat{W} ist die Hypothese, beim Spracherkenners die Best word sequence
 - X sind die Merkmalsvektoren/Beobachtungssequenz, im Sprachmodell $O_1 O_2 \dots$
 - $P(X|W)$ ist das akustische Modell
 - $P(W)$ ist das Sprachmodell
 - argmax ist der Dekodierer
- Word Error Rate (WER)

- $WER = \frac{\#Insertions + \#Deletions + \#Substitutions}{\#Words\ in\ the\ reference}$
- Es werden ganze Wörter eingefügt/gelöscht/ersetzt, nicht einzelne Buchstaben.
- Preplexität eines Sprachmodells: Durchschnittlicher Verzweigungsgrad nach jedem Wort. Kann erreicht werden durch kleineres Vokabular oder besseres Sprachmodell. Korreliert mit Wortfehlerrate.
- Dimensions of Difficulty (Problemfaktoren bei Spracherkennung)
 - Sprecher
 - Sprechstil
 - Vokabulargröße
 - Noise (Hintergrundgeräusche...)
 - Verwechselbarkeit (Buchstaben, ...)
- Vorgehensweise Gesichtsdetektion
 - **Trainingsdatenbestand** von Fotos werden aufgebaut. Gesichter darin werden manuell gelabelt, die ausgeschnittenen Gesichter werden als positive Trainingsbeispiele verwendet, zufällige verschieden-große andere Regionen ohne Gesichter werden als negative Trainingsbeispiele verwendet.
 - Eine Zuordnung von Trainingsdaten zu Annotationen ist bekannt, daher kann ein **überwachtes Lernverfahren** wie **Multilayer Perzeptron** (MLP) verwendet werden.
 - An **jedes Eingabeneuron** wird ein **Grauwertpixel geschaltet**.
 - Dazu wird das Bild auf eine feste Größe **skaliert**.
 - Für große Trainingsgenauigkeit werden Eingabebilder **vorverarbeitet**, um Umwelteinflüsse zu reduzieren (zB Histogrammegalisation) und wichtige Merkmale zu betonen (zB Kantendetektion)
 - Während dem **Training** werden Anzahl der versteckten Schichten, Anzahl Neuronen pro Schicht, Anzahl der Trainingsepisoden bestimmt. Zu langes Training kann zu Overfitting führen.
 - Um Overfitting zu vermeiden, werden die Daten in Trainings- und **Testdaten** geteilt. In jeder Trainingsperiode wird der Fehler auf den Validierungsdaten überprüft. Steigt er an, gerät Overfitting.
 - Zur Gesichtsdetektion wird der Klassifikator **über das Bild geschoben**, um überall nach einem Gesicht zu suchen.
 - Um Gesichter unterschiedlicher Größe zu finden, kann das **Eingabebild verschieden skaliert werden**, und jedes mal der Klassifikator drüber geschoben werden.