

# Computer Vision for Human-Computer Interaction

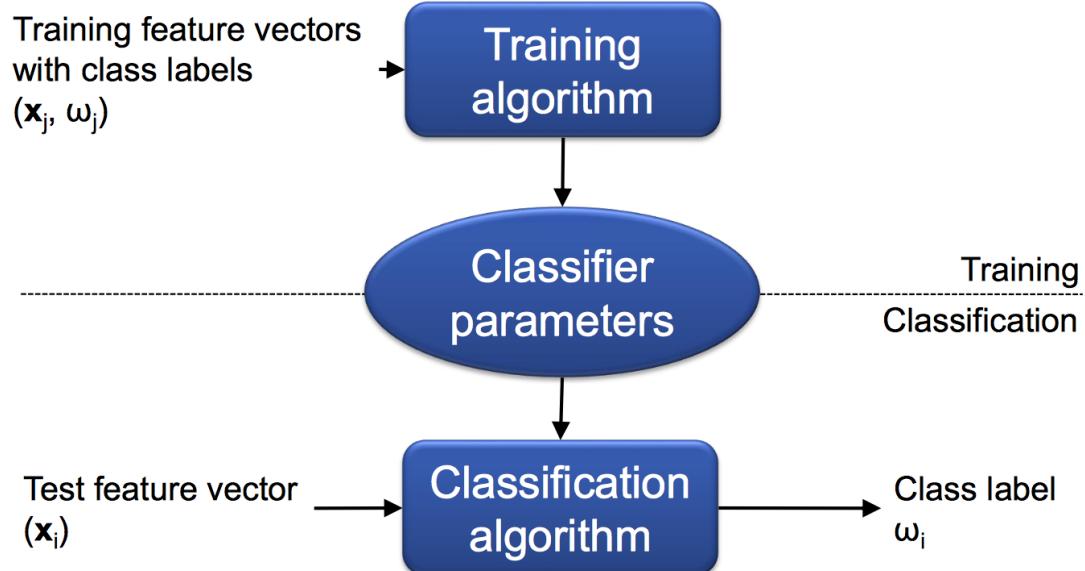
---

Manuel Lang

19. April 2017

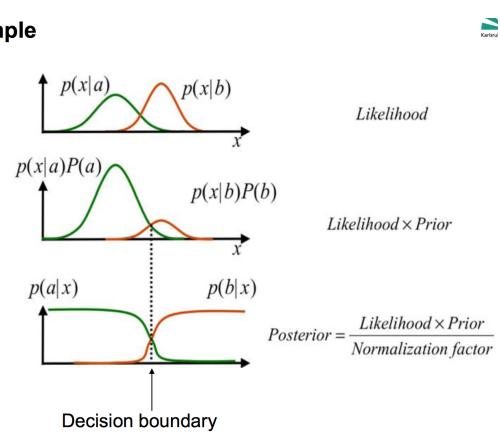
## 1 PATTERN RECOGNITION

### 1.1 CLASSIFICATION



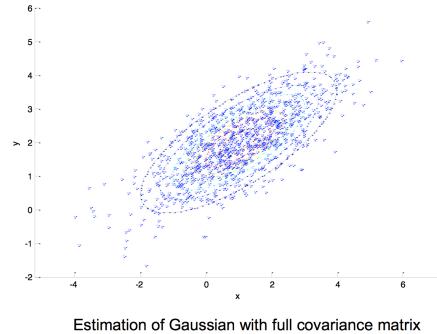
### 1.1.1 BAYESIAN CLASSIFICATION

- überwacht
- Featurevektoren  $x_1$  bis  $x_m$  mit  $x_i = \langle a_1, \dots, a_n \rangle$
- Vorhersage Klasse  $\omega_i$
- $P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$



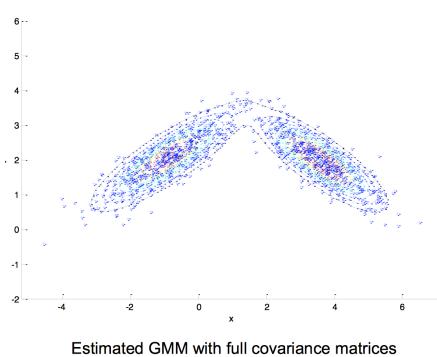
### 1.1.2 GAUSSIAN CLASSIFICATION

- Annahme  $p(x|\omega_i) \approx N(\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]$
- Nur  $\mu$  und  $\Sigma$  müssen geschätzt werden (Maximum Likelihood)



### 1.1.3 GAUSSIAN MIXTURE MODELS (GMMs)

- Gewichtete Summe von mehreren Gaussians
- $p(x) = \sum_i w_i \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]$  mit  $\sum_i = 1$
- Gewichte müssen zusätzlich geschätzt werden -> EM



#### 1.1.4 EXPECTATION MAXIMIZATION (EM)

- Problem: Zu welchem Gaussian gehört ein Punkt?
- EM Algorithmus
  - Zufällige Initialisierung der GMM
  - Wiederhole bis Konvergenz
    - \* Expectation: Berechne die Wahrscheinlichkeit  $p_{ij}$ , dass ein Punkt  $i$  zu einem Gaussian  $j$  gehört
    - \* Maximisation: Berechne neue GMM Parameter  $p_{ij}$  (Maximum Likelihood)

#### 1.1.5 PARAMETRISIERTE VS NICHT-PARAMETRISIERTE MODELLE

- Gaussian und GMM sind parametrisiert
  - Wahrscheinlichkeitsverteilung mit Parametern
  - Nur Parameter werden geschätzt
- Methoden die keine spezifische Form der Wahrscheinlichkeitsverteilung vermuten
  - nicht-parametrisiert
  - Parzen windows, k-nearest neighbors
- Parametrisierte Klassifikatoren
  - + weniger Trainingsdaten, da weniger Parameter zu schätzen
  - - funktionieren nur, wenn Model zu Daten passt
- Nicht-parametrisierte Klassifikatoren
  - + Funktionieren für alle Arten von Verteilungen
  - - mehr Trainingsdaten

#### 1.1.6 GENERATIVE VS DISKRIMINATIVE MODELLE

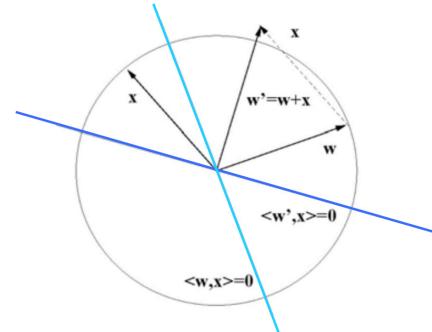
- $P(\omega_i)$  und  $p(x|\omega_i)$  explizit modelliert → generativ: Durch  $p(x|\omega_i)$  können neue Samples der Klasse  $\omega_i$  generiert werden
- direktes Modellieren von  $P(\omega_i|x)$  oder nur eine Entscheidung  $\omega_i$  abhängig von einem Muster  $x$  → diskriminativ
- diskriminativ oft leichter zu trainieren

#### 1.1.7 LINEARE DISKRIMINATIVE FUNKTIONEN

- Trenne zwei Klassen  $\omega_1, \omega_2$  mit einer linearen Hyperebene  $y(x) = w^T x + w_0$
- Wähle  $\omega_1$  für  $y(x) > 0$  und  $\omega_2$  für  $y(x) < 0$  mit dem Normalenvektor  $w^T$  der Hyperebene
- Anwendung: Perzeptron, lineare SVM

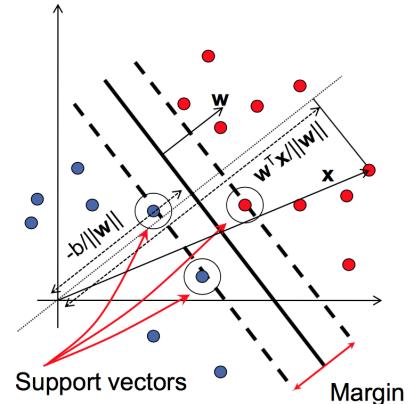
### 1.1.8 PERZEPTRON

1. Initialisiere  $w = 0$  oder zufällig
2. Klassifizierte mit  $y(x) = \text{sign}(w^T x)$
3. Falls korrekt, zu 2, sonst  $w' = w - \eta \cdot y(x) \cdot x$
4. Falls keine Fehler, done, sonst 2



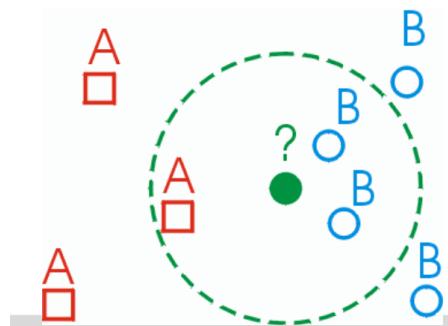
### 1.1.9 SUPPORT VECTOR MACHINES

- Hyperebene maximiert den Abstand zwischen positiven und negativen Beispielen
- Soft-margin: Minimiere  $\|w\|^2 + C \cdot \sum \xi_i$  anstatt  $\|w\|^2$  um Abweichungen zuzulassen
- nicht separierbar  $\rightarrow$  Dimension erhöhen, Kernel-Trick



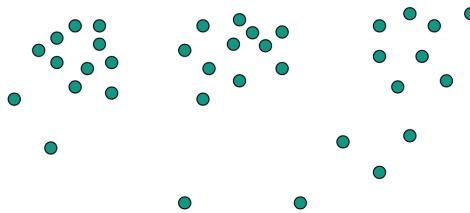
### 1.1.10 K-NEAREST-NEIGHBORS (KNN)

- Betrachte  $k$  nächste Trainingsamples und verwende meist vorkommendes Label
- Modell beinhaltet alle Trainingsdaten
  - + kein Informationsverlust
  - - viel Daten (Performanz)
  - - Distanz wird für jeden Testdatensatz neu berechnet
- Distanzmetrik benötigt
  - Wichtiger Parameter
  - $L_1, L_2, L_\infty$ , Mahalanobis, ...
  - - Wird abhängig von Problemstellung verwendet



## 1.2 CLUSTERING

- Nur Datensätze, keine Labels
- Strukturen werden erkannt

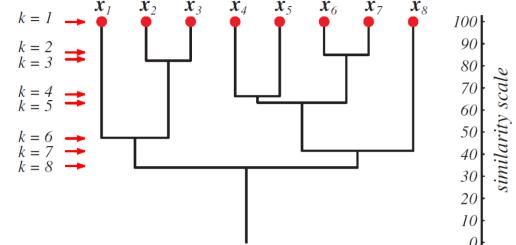


### 1.2.1 K-MEANS

- Zufällige Initialisierung von  $k$  Clustern
- Wiederhole bis Konvergenz
  - Datenpunkte zu nächstem Clusterzentrum
  - Berechne neue Clusterzentren in der Mitte der zugewiesenen Punkte
- + einfach und effizient
- - Anzahl  $k$  muss vorgegeben werden
- - Ergebnis abhängig von Initialisierung
- - funktioniert nicht für verschiedene Clustertypen (runde, überlappend)
- ähnlich zu EM, benutzt aber feste Zuweisungen anstatt probabilistischen (EM)
- EM kann für Clustering verwendet werden

### 1.2.2 AGGLOMERATIVE HIERARCHICAL CLUSTERING

- Jeder Punkt ist zu Beginn Cluster
- Vereine die zwei nächsten Cluster
- Verschiedene Distanzmaße möglich (min,max,avg,mean)
- Ergebnis ist Baumstruktur (Dendrogram)



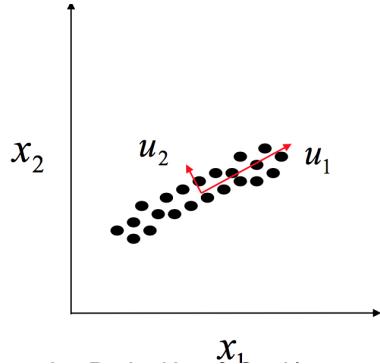
### 1.2.3 „CURSE OF DIMENSIONALITY“

- Featurevektoren haben oft sehr hohe Dimensionalität
- Operationen der linearen Algebra nicht mehr anwendbar
- Klassifikatoren arbeiten besser in niedriger Dimensionalität
- Entstehende Probleme bei hoher Dimensionalität → „Curse of dimensionality“

## 1.3 DIMENSIONALITY REDUCTION

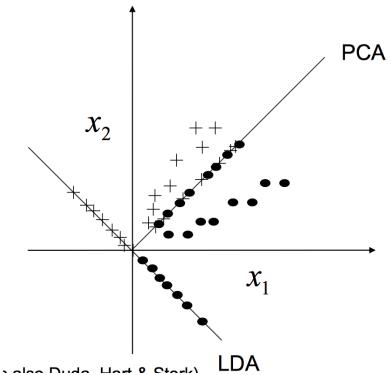
### 1.3.1 PCA

- Finde Richtungsvektoren, die den Fehlerschnitt minimieren, durch Kovarianzmatrix
- Projiziere auf den durch diese Vektoren aufgespannten Raum
- Projiziere auf  $K$  Richtungsvektoren um die Dimensionalität zu reduzieren



### 1.3.2 LDA

- TODO



## 2 GESICHTSDETEKTION

### 2.1 WIESO GESICHTSERKENNUNG?

- Personenidentifizierung
- Erkennung von Emotionen/Posen
- Mund als Quelle der Sprachen
- Lippenlesen
- Absichtserkennung
- Erkennung von Alter, Geschlecht, Hautfarbe
- Ansatz: Hautfarbe hat konsistente Farbwerte

## 2.2 FARBRÄUME

- RGB (Rot, Grün, Blau)
- HSV/HSI (Hue, Saturation, Value/Intensity)
- Y-Klasse: YCbCr (Videos), YIQ (NTSC), YUV (PAL): Trennung von Helligkeit (Y) und Chrominanz

## 2.3 MODELLIERUNG VON HAUTFARBE

- nicht-parametrisiert (Histogramme)
- parametrisierte (Gauss, Gauss-Mixturen)
- Grenzen lernen (diskriminative Modelle, ANN, SVM, ...)

## 2.4 HISTOGRAM BACKPROJECTION

- einfacher und schneller Ansatz
- Jeder Pixel in der Rückprojektion wird auf das Histogram-Bin des Farbwerts gesetzt

## 2.5 HISTOGRAM MATCHING

- Backprojection gut bei monomodaler Farbverteilung, aber schlecht bei bunten Zielen
- Lösung: Histogramm innerhalb des Suchfensters wird mit Ziel verglichen
- Verschiedene Distanzmetriken (Battacharya Distanz, Histogram intersection, Earth-movers distance, ...)

## 2.6 VERGLEICH HISTOGRAM BACKPROJECTION VS HISTOGRAM MATCHING

| Histogram Backprojection                                 | Histogram Matching                                    |
|--|---|
| • Vergleicht Farbe eines einzelnen Pixels mit Farbmodell | • Vergleicht Farbhistogramm des Bildes mit Farbmodell |
| • schnell und einfach                                    | • bessere Ergebnisse                                  |
| • nur gut bei monomodaler Verteilung                     | • anwendbar für multimodale Verteilung                |
| • genügt für Klassifizierung von Hautfarbe               | • teure Berechnungen                                  |

## 2.7 GAUSSIAN DENSITY MODELS

- Annahme: Verteilung der Hautfarben  $p(x)$  hat die Form einer parametrischen Funktion
- Gauss-Funktion  $G(x, \mu, C)$ :  $p(x|skin) = G(x; \mu, C) = (2\pi)^{-d/2} |C|^{-1/2} \exp\{-1/2(x-\mu)^T C^{-1}(x-\mu)\}$
- Mittel  $\mu$  und Kovarianz-Matrix  $C$  werden aus Trainingsdatensatz von Hautfarben  $S = \{s_1, s_2, \dots, s_N\}$  geschätzt.  $\mu = E\{x\}$ ,  $C = E\{(x - \mu)(x - \mu)^T\}$
- Klassifizierung  $p(x|skin) > \theta$  oder  $p(x|skin) > p(x|non-skin)$

## 2.8 MIXTURE OF GAUSSIANS MODELS

- Ein Gaussian kann zu wenig sein, um eine Hautfarben-Verteilung genügend zu beschreiben (z.B. HS-Raum):  $p(x) = \sum_{i=1}^K \pi_i G(x, \mu_i, C_i)$
- Parametersatz  $\Phi$  kann durch EM Algorithmus geschätzt werden:  $L = \log \prod_{i=1}^N p(x_i|\Phi)$
- Klassifizierung  $p(x|skin) > \theta$  oder  $p(x|skin) > p(x|non-skin)$

## 2.9 BAYES CLASSIFIER

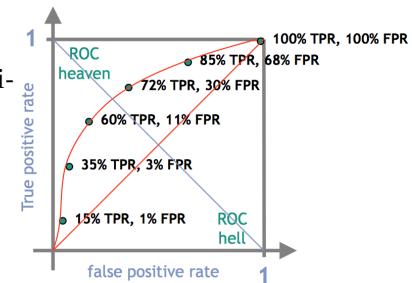
- minimale Kosten
- Klassifikation  $P(Skin|x) > P(Non-Skin|x)$
- Entscheidungsregel:  $\frac{p(x|Skin)}{p(x|Non-Skin)} \geq \frac{P(Non-Skin)}{P(Skin)}$
- Die Klassenbedingungen  $p(x|\omega)$  können über die korrespondierenden Histogramme geschätzt werden:  $p(x|\omega_i) = h_i(x) / \sum_x h_i(x)$ ,  $h_i(x)$  - Anzahl der Pixel der Klasse  $\omega_i$  mit dem Wert (x)

## 2.10 DISKRIMINATIVE MODELLE / KLAFFIKATOREN

- Künstliche Neuronale Netze (ANNs)
- Support Vector Machines
- ...

## 2.11 PERFORMANZ-METRIKEN

- ROC (Receiver-Operating-Characteristic)
  - Anwendung: Klassifikation
  - Trade-off zwischen true positive und false positive
  - true positive rate =  $TP / Pos = TP / (TP+FN)$
  - false positive rate =  $FP / Neg = FP / FP + TN$

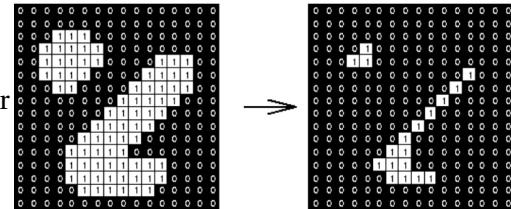


- RPC (Recall-Precision-Curve)
- DET (Detection Error Trade-Off)

## 2.12 MORPHOLOGISCHE OPERATOREN

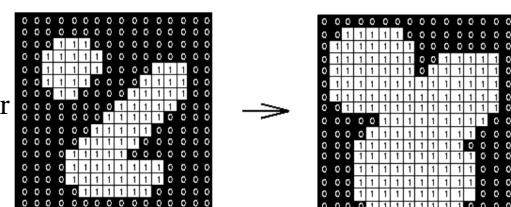
- Erosion

- entfernt Pixel an Ecken von Objekten
- setzt Pixelwerte auf das Minimum der umliegenden Pixel



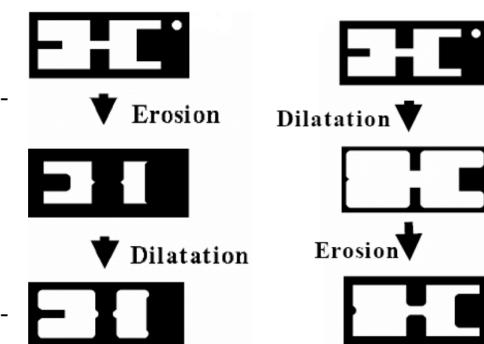
- Dilatation

- fügt Pixel an Ecken von Objekten ein
- setzt Pixelwerte auf das Maximum der umliegenden Pixel



- Opening

- erst Erosion, dann Dilatation
- Glätten, Lücken einfügen, Outliers eliminieren

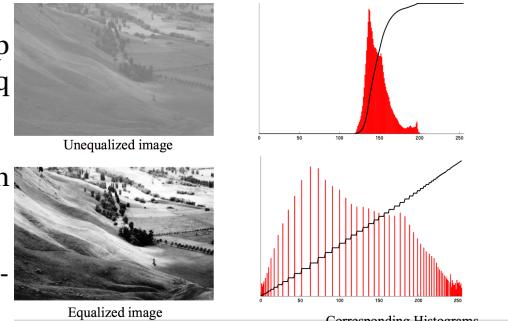


- Closing

- erst Dilatation, dann Erosion
- Glätte innere Ecken, verbinde kleine Distanzen, fülle unerwünschte Löcher

## 2.13 GESICHTSDETEKTION MIT NEURONALEN NETZEN

- Ansatz: Verwende neuronales Netz um aufrechte frontale Gesichter zu erkennen
- Input: 20x20 Pixel Bildregion
- Bereich -1 (kein Gesicht) bis 1 (Gesicht)
- Filter auf gesamtes Bild angewendet
- Bild wird skaliert, um Gesichter mit verschiedenen Größen zu erkennen
- Training
  - 1050 normalisierte Bilder von Gesichtern
  - 15 Bilder von rotierten und skalierten Gesichtsbildern
  - 1000 zufällig ausgewählte Bilder ohne Gesichter
- Vorverarbeitung
  - Beleuchtungskorrektur
  - Skalieren auf einheitliche Größe
- Histogram equalization (Histogrammspreizung)
  - definiert Mapping von Grauwerten  $p$  auf Grauwerte  $q$ , sodass die Grauwerte  $q$  einheitlich verteilt werden
  - erweitert den Bereich der Graustufen (erhöht Kontrast)
  - bildet Eingabebilder so ab, dass sie ähnliche Intensitätsverteilungen haben



- Ausgabe eines KNNs definiert einen Filter für Gesichter
- Suche: Wende Suchfenster auf Eingabebild an, wende auf Suchfenster KNN an, Eingabebild muss skaliert werden um Gesichter mit unterschiedlicher Größe zu erkennen
- Nachverarbeitung: Rauschreduktion, kombinierende sich überschneidende Ergebnisse
- Geschwindigkeits-Optimierung: erhöhe Schrittweite, mache ANN flexibler ggü Translationen, hierarchische Suche

## 2.14 FEATURE-BASIERTE GESICHTSDETEKTION

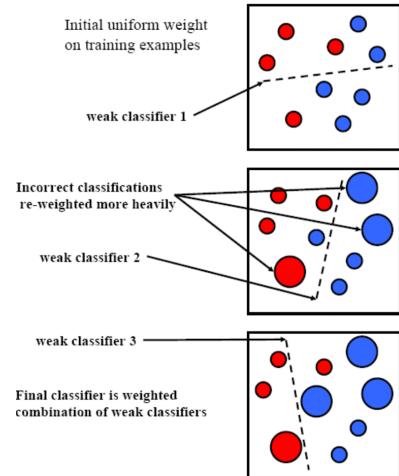
- Erkennung mit (Haar-Like) Features statt Pixeln: Features können umgebungsabhängiges Wissen beinhalten, was schwierig zu lernen ist, schneller als Pixel-basierte Ansätze, skalierungs-invariant
- Robuste Echtzeiterkennung von Features (Gesichter), Integralbildern (um Feature zu bestimmen), schwachen Klassifikatoren (Kaskaden) und Trainieren der Kaskaden
- Anwendung von Suchfenstern (z.B. 24x24px), viele Features können ausgelesen werden, daher können nicht alle Features bestimmt werden
- Integral Image:  $ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$
- Bestimmung aller Features dauert nachwievor zu lange, nur entscheidende Features sollen gelernt werden

### 2.14.1 ADABOOST

- Ziel: Erhöhen der Klassifizierungsergebnisse von einfachen Algorithmen (z.B. Perzeptron)
- Variante davon wird verwendet um Features auszuwählen und den Klassifikator zu trainieren
- kombiniert mehrere schwache Klassifikatoren um einen starken zu bilden
- schwacher Klassifikator  $h$  mit Genauigkeit geringfügig besser als Zufall
- kombiniere schwache Klassifikatoren  $h_i \in \{-1, 1\}$  linear zu einem starken Klassifikator  $H = \text{sign} \sum_{i=1}^N w_i h_i$
- Viola & Jones: Schwache Klassifikatoren  $h_j(x)$  bestehen aus Featuren  $f_j$ , einem Threshold  $\theta_j$  und einer Polarität  $p_j$ , die die Richtung des  $\text{sign}$  indiziert

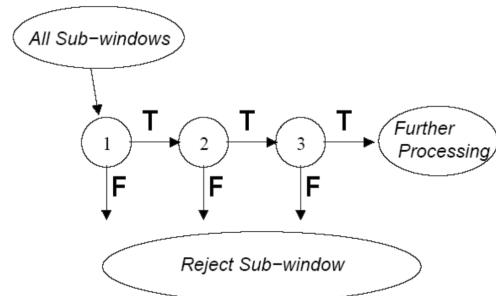
$$h_j(x) = \begin{cases} 1 & \text{für } p_j f_j(x) < p_j \theta_j \\ 0 & \text{sonst} \end{cases}$$

- selektiert die schwachen Klassifikatoren mit der geringsten Fehlerrate
- effizient für wenig gute Features mit hoher Varianz



## 2.14.2 CLASSIFIER CASCADE

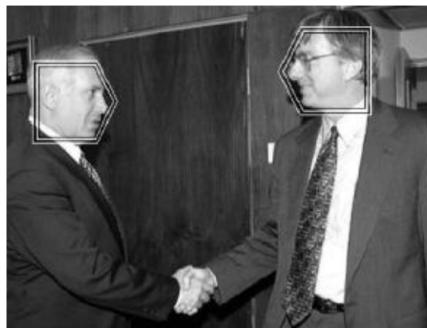
- nicht alle Fenster enthalten Gesichter, Klassifikator muss aber trotzdem für alle angewandt werden
- Ansatz: Kaskade von Klassifikatoren
  - Jeder weitere Klassifikator ist genauer: erste Schicht entdeckt alle positiven Beispiele, aber enthält viele falsche, zweite fokussiert auf false positive des ersten Layers, ...
  - Fenster ohne Gesichter werden schnell aussortiert, Gesichter werden bis zum Ende weitergereicht



## 2.14.3 CASCADE TRAINING

- senkt false positive Rate in jeder Schicht (allgemein weniger erkannte Gesichter)
- In jeder Schicht wird Ziel auf minimale Reduktion gesetzt. Ziel wird erreicht, indem weitere Features zur Schicht hinzugefügt werden.
- Schichten werden hinzugefügt, bis Ziel der Falschklassifikation erreicht wird
- Jede Schicht wird komplexer und beinhaltet mehr Features

## 2.14.4 ROTIERTE BILDER



1)



2)

- 1) out of plane rotation: → train individual classifiers for each pose ...
  - Frontal, half-profile, profile, etc.
- 2) in-plane rotation: → apply rotated detectors / rotate (sub-) images

#### 2.14.5 ZUSAMMENFASSUNG VIOLA & JONES

- Ansatz ist 15x schneller als bisherige Ansätze
- Anwendung auf verschiedene Objekte möglich
- Einfluss auf eine Vielzahl anderer Aufgaben
- entdeckt mehrere Gesichter in Bildern
- robust gegen Illuminanz

### 3 GESICHTSIDENTIFIKATION

#### 3.1 HERAUSFORDERUNGEN

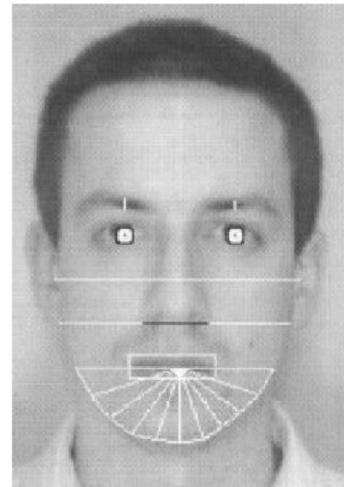
- abweichende Beleuchtung
- abweichender Blickwinkel (frontal, nicht-frontal)
- Occlusion (Sonnenbrille, Hut, Bart, Make-up)
- Gesichtsausdrücke
- Alter

#### 3.2 CLOSED SET VS. OPEN SET IDENTIFICATION

- Closed-Set: Wer ist die Person?
- Open-Set: Ist Person in Daten? Wenn ja, wer?
  1. False accept: Person nicht in Daten, aber so erkannt
  2. False reject: Person in Daten, aber nicht erkennt
  3. False classify: Person ist in Daten, so erkannt, aber die falsche Person zugeordnet

### 3.3 FEATURE-BASIERTE GESICHTSIDENTIFIKATION

- Dicke der Augenbrauen
- Position von Augen, Nase und Mund
- Breite der Nase
- Mund-Höhe und -Breite
- 11 Kanten zwischen Mund und Kinn
- Gesichtsbreite auf Nasenhöhe



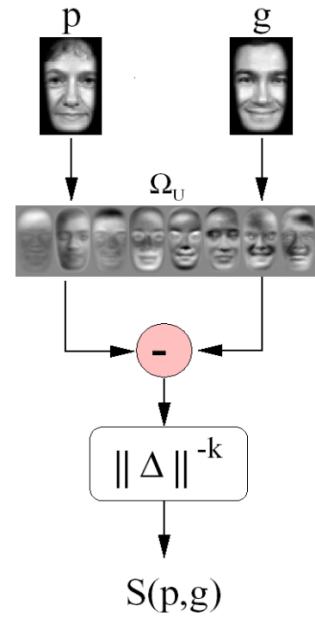
### 3.4 APPEARANCE-BASIERTE GESICHTSIDENTIFIKATION

- entweder holistisch (gesamtes Gesicht als Eingabe)
- oder lokal/fiduziell (Berechnung von Gesichtsfeatoren wie Augen, Mund, etc. getrennt)



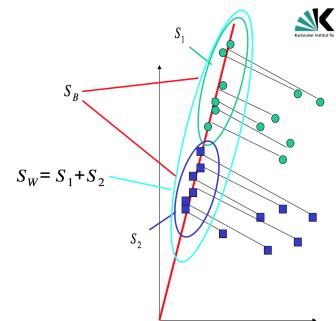
### 3.5 EIGENFACES

- Gesichtsbild definiert einen Punkt im hochdimensionalen Bildraum
- verschiedene Gesichtsbilder haben eine Reihe von Gemeinsamkeiten
- Gesichtsbilder können in einem eher geringen Unterraum beschrieben werden
- Abbildung des Gesichtsbild in einen Unterraum und berechne Ähnlichkeit
- Verwendung von PCA: Analysierte Komponenten werden Eigenfaces genannt und spannen den „face space“ auf
- Probleme:
  - Eigenfaces unterscheidet nicht zwischen Shape und Appearance -> ASM/AAM
  - PCA verwendet keine Klasseninformationen



### 3.6 FISHERFACES

- verwendet LDA (Linear Discriminant Analysis)
- wahrt Separierbarkeit der Klassen
- Maximiert Verhältnis von Streuung zwischen Klassen zu Streuung innerhalb von Klassen
 
$$W_{fId} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$
- Streuung zwischen Klassen  $S_B = \sum_{i=1}^c |x_i|(\mu_i - \mu)(\mu_i - \mu)^T$
- Streuung innerhalb von Klassen  $S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T$
- mit  $c$ : Anzahl der Klassen,  $\mu_i$ : Durchschnitt der Klasse  $X_i$ ,  $|X_i|$ : Anzahl der Samples von  $X_i$
- Fisherfaces werden unabhängig von Unterschieden innerhalb von Klassen (Beleuchtung, Gesichtsausdrücke)



### 3.7 PROBLEM: MATCHING MIT VERSCHIEDENEN POSEN

- Problem: verschiedene Blickrichtung, Kopforientierung
- verschlechtert Erkennung
- Ansätze: Pose-Normalization, 2D Pose Modelle, 2D+3D Modelle, 3D face Model fitting

### 3.8 POSE-NORMALIZATION

- Alignment mit Augenpunkten ist nicht ausreichend

- Idee

- Detektiere mehrere Gesichtsfeatures (Meshes)
- nutze Mesh um Gesicht zu normalisieren



- Rechts: 2d Active Appearance Models

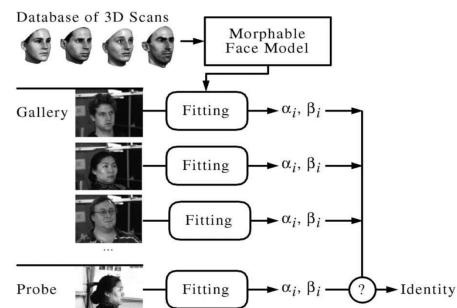
- basiert auf Textur und Forum (parametrisiert)
- Inverse compositional (IC) Algorithmus



- Modell kann verwendet werden, um Bild in frontale Pose zu transformieren

### 3.9 FACE RECOGNITION BASED ON FITTING A 3D MORPHABLE MODEL

- Methode zur Umgehung von Variationen in Pose und Beleuchtung
- Simuliert die Bild-Entstehung im 3D Raum
- Schätzt 3D Shape und Textur von Gesichtern aus einem Bild durch Anwenden eines morphable models von 3D Gesichtern auf das Bild
- Gesichter sind Modell-Parameter für 3D Shape und Textur (Shape:  $S = \sum_{i=1}^m a_i S_i$ , Textur:  $T = \sum_{i=1}^m b_i T_i$ )



### 3.10 LOCAL FEATURE BASED FACE RECOGNITION

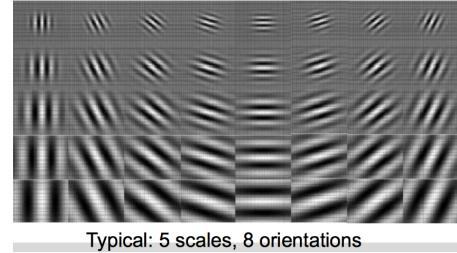
#### 3.10.1 GABOR FILTER

- 2D Sinuskurven moduliert durch Gaussglocke

$$\bullet \quad g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i(2\pi \frac{x'}{\lambda} + \psi)\right)$$

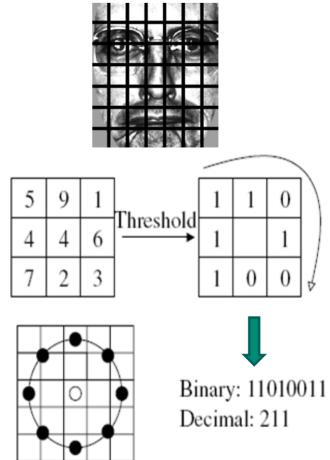
- Gabor Wavelet Transformation (GTW)

- $O_{u,v}(x, y) = I(x, y) * \psi_{u,v}(x, y)$
- Typischerweise werden mehrere Skalierungsfaktoren  $u$  und Orientierungen  $v$  verwendet
- $O_{u,v}(x, y)$  wird hochdimensionaler Vektor
- Dimensionsreduzierung (z.B. PCA)



#### 3.10.2 LOCAL BINARY PATTERNS (LBP)

- Teile Bild in Zellen
- Vergleiche jeden Pixel mit seinen Nachbarn
- Pixelwert > Threshold ? 1 : 0, liefert Binärzahl
- Konvertiere binär zu dezimal
- Berechne Histogramm über die Zelle
- Nutze Histogramm für Klassifikation (z.B. SVM/Histogramm-Entfernung)



#### 3.10.3 DENSE FEATURES

Features werden dicht berechnet, z.B. Filter auf Bild in mehreren Skalierungen.

Problem: Hohe Dimensionalität

Ansätze: Bag of Visual Words, Fisher encoding

#### 3.10.4 FISHER VECTOR ENCODING

- kompakte Repräsentation von Featurevektoren

- parametrisiertes generatives Model (z.B. GMM)

### 3.10.5 DISCRETE COSINE TRANSFORM (DCT)

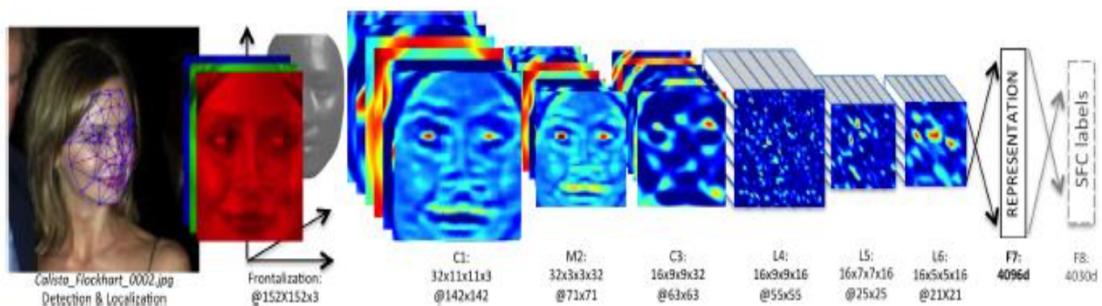
### 3.10.6 SIFT

### 3.10.7 BAG OF VISUAL WORDS (BoVW)

## 3.11 FACE RECOGNITION MIT DEEP LEARNING

### 3.11.1 DEEPFACE (FACEBOOK)

- Ansatz: Lerne ein tiefes (7 Schichten) NN (20 Millionen Parameter) mit 4 Millionen Personen direkt auf RGB-Pixeln
- Alignment: 6 Gesichtspunkte für 2D warp, 67 Punkte für 3D Modell, transformiere Gesicht in Frontal-Pose als Input für NN
- k-way softmax -> Verteilung über Klassenlabels
- Ziel: Maximiere Wahrscheinlichkeit für korrekte Klasse



### 3.11.2 FACENET (GOOGLE)

- Mappe Bilder in kompakten euklidischen Raum, sodass Entfernung  $\approx$  Ähnlichkeit der Gesichter
- Finde  $f(x) \in \mathbb{R}^d$  sodass:  
 $d^2(f(x_1), f(x_2)) \rightarrow$  klein für gleiche Identität  $x_1$  und  $x_2$   
 $d^2(f(x_1), f(x_2)) \rightarrow$  groß für verschiedene Identitäten
- Ansatz: Convolutional Neural Networks optimieren Einbettung, Triplet-based loss function -> training



### 3.11.3 DEEP FACE RECOGNITION (OXFORD)

- sehr tiefes Netzwerk
- sehr wenig conv. Kernels

## 4 FACIAL FEATURE DETECTION

- Teile von Gesichtsregionen, die entscheidende Informationen beinhalten, bspw. Augen, Augenbrauen, Nase, Mund (auch Landmarks genannt)
- Die Detektion dieser spezifischen Gesichtsbereiche wird Facial Feature Detection genannt

### 4.1 ANWENDUNGEN

- Gesichtserkennung: Geometrische Features zur Identifikation, Facial Features zur Normalisierung bei appearance based models, nötig für lokale / modulare Ansätze
- model-based head post estimation: Pose von 2D zu 3D
- Blick tracking: Lokalisierung der Augen
- facial expression recognition: Bewegung von Augen, Augenbrauen und Mund um Emotionen zu analysieren
- Modellierung von Alter: Gesichtszüge/-Strukturen unterscheiden sich im Alter

### 4.2 PROBLEME

- verschiedene Personen
- verschiedene Expressio-nen
- verschiedene Kopfdrehungen
- verschiedene Skalierun-gen
- Beleuchtung
- Occlusion

### 4.3 STATISTICAL APPEARANCE MODELS

- Idee: Verwendung von priori Wissen (Modellen um die Komplexität der Aufgabe zu reduzieren)
- muss mit Varianz umgehen können (-> deformierbare Modelle)
- verwende statische Modelle von Shape und Texturen um Landmarken zu finden

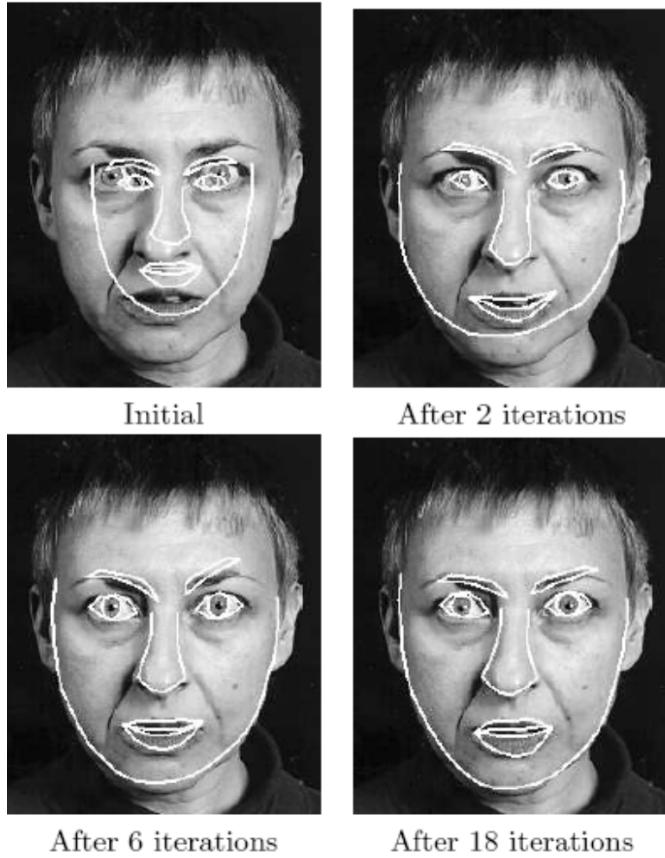
#### Appearance Models

- repräsentieren Textur und Form
- statistischen Modell aus Trainingsdaten gelernt
- Modellieren der Form-Varianz  $x = [x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T$  ergibt Modell  $x \approx \bar{x} + P_s b_s$
- Modellieren der Intensitäts-Varianz  $h = [g_1, g_2, \dots, g_k]^T$  ergibt Modell  $h \approx \bar{h} + P_i b_i$

- Konstruiere Shape-Model mit PCA ( $x = \bar{x} + P_s b_s$ )
- Konstruiere Texturmodell an jedem Punkt (image warping, texture wrapping) ( $g = \bar{g} + P_g b_g$ )
- Modelliere die Verbindung zwischen Shape und Grauwerten (Textur)

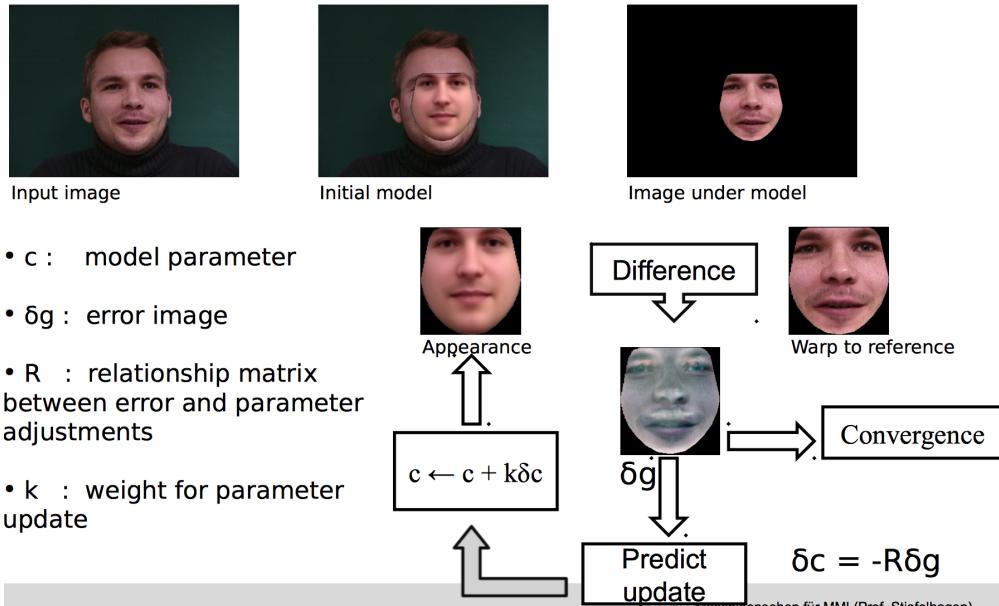
#### 4.4 ACTIVE SHAPE MODELS

- Geg. grobe Startposition, erstelle eine Instanz von  $X$  mit Shape  $b$ , Translation  $T = (X_t, Y_t)$ , Skalierung  $s$  und Rotation  $\theta$
- Iterativer Ansatz
  1. Untersuche Bildbereich im Bereich von  $X_i$  um das beste Match für den Punkt  $X'_i$  zu finden
  2. Aktualisiere Parameter  $(b, T, s, \theta)$  für den neuen Punkt
  3. Wiederhole bis Konvergenz
- Praxis: Suche entlang der Normalen des Profils



## 4.5 ACTIVE APPEARANCE MODELS

- Nachteile von ASM: verwendet hauptsächlich Form und weniger Textur
- Ansatz AAM: Optimiere Parameter (minimiere die Differenz zwischen künstlichem Bild und Ziel) mit Gradientenabstieg



### 4.5.1 ASM vs AAM

- ASM
  - Versucht ein Set von Modellpunkten auf ein Image durch ein statistischen Shape-Modell zu matchen
  - verwendet iteratives Vorgehen (Variante von EM)
  - Lokale Suche zur Bestimmung des besten Landmarks
  - Anschließend aktualisieren der Parameter um die Modellpunkte näher an die neuen Punkte des Bildes zu transformieren
- AAMs matchen sowohl Position der Modellpunkte als auch Repräsentation der Textur des Objekts auf ein Bild (verwendet Differenz des aktuell erzeugten Bildes und des Ziels um die Parameter zu aktualisieren, typischerweise weniger Landmarks)
  - statistische appearance models liefern eine kompakte Repräsentation
  - können verschiedene Identitäten, Gesichtsausdrücke, Vorkomnisse etc modellieren
  - Gelabelte Bilder benötigt
  - sehr rechenaufwendig, jedoch existieren Verfahren zur Beschleunigung (Multi-resolution search, constrained aam search, inverse compositional AAMs (CMU))

- Anwendung: Detektion von fiducial points, face recognition, pose estimation, facial expression analysis, audio-visual speech recognition

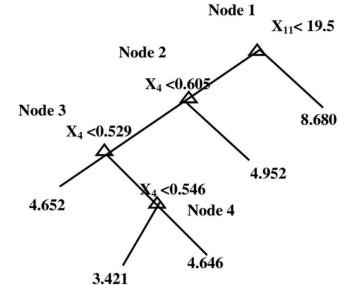
## 4.6 CONDITIONAL RANDOM FORESTS

### Conditional Random Forests For Real Time Facial Feature Detection



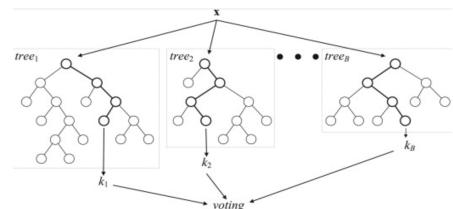
#### 4.6.1 REGRESSION TREES

- Aufbau wie Entscheidungsbaum zur Klassifizierung
- Wurzeln: Entscheidung als Vergleich von Zahlen
- Blätter: Zahlen oder Zahlenvektoren



#### 4.6.2 RANDOM REGRESSION FORESTS

- 1 Wald = viele Bäume
- Random, da verschiedene Bäume auf einem random Subset der Trainingsdaten trainiert werden, nach dem Training werden die Voraussagen der verschiedenen Bäume gemittelt (von allen regression trees)

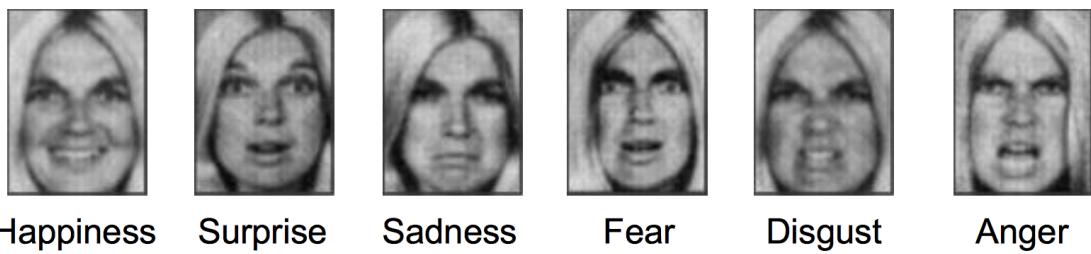


#### 4.6.3 FACIAL FEATURE DETECTION WITH CONDITIONAL REGRESSION FORESTS

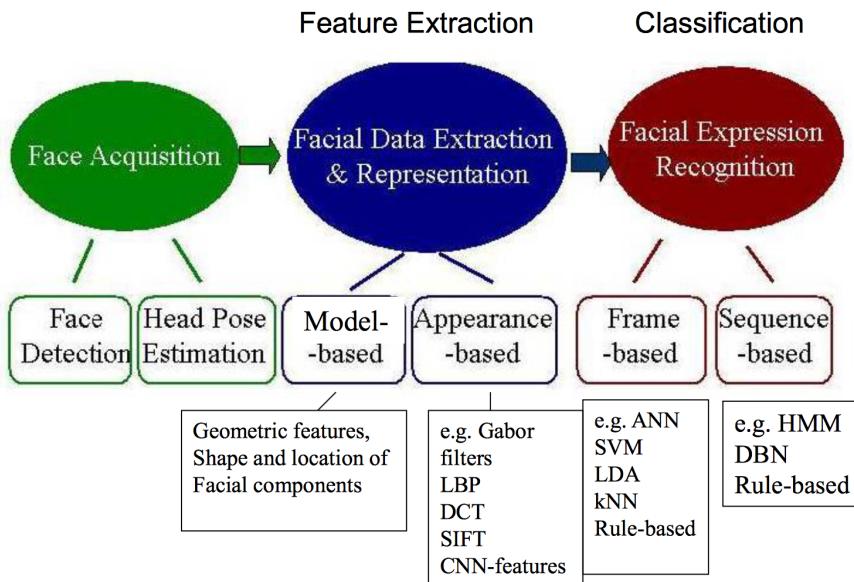
- State of the art!!!
- trainiere verschiedene Sets von Bäumen für verschiedene Posen
- Wurzelknoten akkumulieren die Votes für die verschiedenen facial difucial points auf
- jeder Baum wird mit einem random ausgewählten Set von Bildern trainiert
- extrahiert Patches in jedem Bild
- Trainingsziel: akkumuliere Wahrscheinlichkeit für einen Featurepunkt  $C_n$  gegeben einem Patch  $P$  im Blattknoten
- Training
  - Berechne Wahrscheinlichkeit für konkrete Kopfpose
  - Teile das Trainingsset für jede Pose
  - Trainiere regression forest für jedes Subset
- Testen
  - Schätze Wahrscheinlichkeiten für jede Kopfpose
  - Wähle Bäume von verschiedenen regression forests
  - Schätze die density Funktion für alle facial feature points
  - Clustering über alle Featurekandidaten für einen gegebenen facial feature point

## 5 FACIAL EXPRESSION ANALYSIS

6 grundlegende Emotionen (Ekman)



## 5.1 VORGEHENSWEISE



## 5.2 FACIAL ACTION CODING SYSTEM (FACS)

- human-observer basiertes System designt um feine Änderungen in facial features zu erkennen
- Beim Betrachten von in SlowMo aufgenommenem facial behavior kann der trainierte observer alle möglich Gesichtsausdrücke in FACS abbilden
- Diese Gesichtsausdrücke werden als action units (AU) definiert, können einzeln auftreten, aber auch kombiniert werden

| AU 1              | AU 2              | AU 4         | AU 5             | AU 6         | AU 7          |
|-------------------|-------------------|--------------|------------------|--------------|---------------|
|                   |                   |              |                  |              |               |
| Inner Brow Raiser | Outer Brow Raiser | Brow Lowerer | Upper Lid Raiser | Cheek Raiser | Lid Tightener |
| *AU 41            | *AU 42            | *AU 43       | AU 44            | AU 45        | AU 46         |
|                   |                   |              |                  |              |               |
| Lid Droop         | Slit              | Eyes Closed  | Squint           | Blink        | Wink          |

- Gesichtsausdrücke können mit psychologischen Interpretationen verknüpft werden (z.B. Emotionen)
- Gesichtsausdrücke können indizieren ob eine Person lügt

- Gesichtsausdrücke können zusätzlichen analysiert werden um Depressionen und Schmerzen zu erkennen

#### 5.2.1 AKTUELLE UMSETZUNGEN IN DER PRAXIS

- facial components models + ann
- gabor filters + svm
- cnn

## 6 HEAD POSE ESTIMATION

- Modell-basierte Ansätze
  - Lokalisieren und Tracken von facial Features
  - Bestimmte Kopfpose von 2D zu 3D
  - benötigt facial landmark tracking
- Appearance-basierte Ansätze
  - Schätzt neue Pose mit Funktionsapproximator (bsp. KNN)
  - Nutzung von Gesichtsdatenbank für Encoding
  - AAM / ASM
  - Regression forest auf 3D Tiefendaten

### 6.1 ESTIMATING HEAD POSE WITH ANNS

- Trainiere neuronales Netz zur Schätzung der Kopforientierung
- vorverarbeitetes Bild als Input
- automatische Extraktion des Gesichts (skin-color model)
- Vorverarbeitung: Histogrammnormalisierung, extrahiere Kanten, Down-Sampling
- Probleme: Beleuchtungsänderung -> Anpassung / erneutes Training

#### 6.1.1 HEAD POSE ESTIMATION USING DEPTH FROM STEREO

- Stereokameras → Entfernung können berechnet werden
- Idee: Disparity Bilder sollten weniger anfällig ggü. Beleuchtungsänderungen sein als Grauwertbilder, da beide Kameras die gleiche Änderung haben

## 6.2 REAL-TIME HEAD POSE ESTIMATION WITH RANDOM REGRESSION FORESTS

- Trainingsset: Bilder mit 3D Nasenlocation und Kopfrotationswinkeln
- Jeder Baum im forest wird durch ein Set von Patches erzeugt
- Training: Summiere die Wahrscheinlichkeit für den Posenparameter  $\theta$  unter einem Patch  $P$  auf
- Testen: Gesichtsbild → extrahiere Set von Patches → alle Patches an alle Bäume → erzeuge für jedes Patch ein korrespondierendes Set von Blattknoten → density estimator → meanshift um alle Locations zu finden

## 6.3 VERGLEICH: HEAD POSE ESTIMATION TECHNIQUES

Modell-basiert

- Getrackte Features können auch für andere Anwendungszwecke verwendet werden (z.B. Lippenlesen)
- Features sind schwierig zu finden und zu tracken
- Occlusion behindert die Range
- Gute Auflösung nötig

Appearance-basiert

- nur Kopf muss detektiert/getrackt werden
- keine Limitationen bei der Rotierung
- funktioniert auch mit geringer Auflösung
- keine Initialisierung nötig
- keine Fehler-Aufsummierung (drift)
- Beleuchtungsänderungen sind Problem → Stereo/Tiefenkameras
- viel Trainingsdaten benötigt

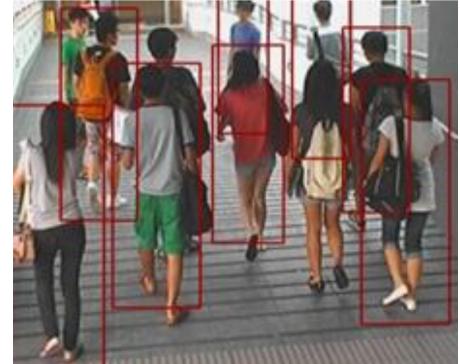
## 6.4 TRACKING FOCUS OF ATTENTION (FOA)

- Focus of Attention tracking
  - Erkenne das Interesse einer Person
  - Womit interagiert eine Person?
  - Was ist die Absicht der Person?
  - Ist sich der Anwender einer Sache bewusst?
- Human-Human Interaction
  - Wer wird betrachtet?
  - Wie ist Dynamik der Interaktion?
- Human-Robot Interaction: Hat sich der Anwender an den Roboter gerichtet?
- Smart Environments, Autos,...

## 7 PEOPLE DETECTION

### 7.1 ANWENDUNGSBEREICHE

- Person Re-Identification
- Person Tracking
- Sicherheit (z.B. Flughafenüberwachung)
- Automotive (z.B. Unfallverhinderung)
- Interaktion (z.B. Xbox Kinect)
- Medizin (z.B. Patientenüberwachung)
- Werbung (z.B. Kundenzählung)
- funktioniert auch wenn Gesichter nicht sichtbar
- funktioniert auch bei sehr schlechter Auflösung



### 7.2 PROBLEME

- Kleidung sehr vielseitig
- Occlusion durch Accessoires
- Artikulation - Formen sehr vielseitig
- Überlappung von Personen

### 7.3 RUHENDE BILDER VS VIDEOS

Modell-basiert

- Hauptsächlich Grauwertinformationen
- andere Möglichkeiten: Farbe, Infrarot, Radar, Stereo
- - oft schwieriger
- - schlechtere Ergebnisse als video-basierte Techniken
- + vielseitig anwendbar

Apearance-basiert

- Hintergrundmodellierung
- Zeitabhängige Informationen (Geschwindigkeit, Position ( $t - 1$  vs  $t$ ))
- optischer Fluss
- Kann mit Ansatz von ruhenden Bildern initialisiert werden
- - Schlecht anwendbar ohne Nebenbedingungen (bewegende Kameras / sich verändernder Hintergrund)

## 7.4 GLOBALE VS PART-BASED ANSÄTZE

### Part-based

#### Global

- holistische Modelle (1 Feature für 1 Person)
- + typischerweise sehr einfach
- + gut bei geringen Auflösungen
- - Probleme bei Occlusion
- - Probleme bei Artikulation

- Körperteile getrennt modelliert
- + besser bei Posen (sich bewegenden Körperteilen)
- + funktioniert mit Occlusion / Überlappungen
- + Trainingsdaten können geteilt werden
- - Probleme bei geringer Auflösung
- - benötigen komplexere Entscheidung

## 7.5 GENERATIVE VS DISKRIMINATIVE MODELLE

#### Generativ

- modelliert wie Daten (Personenbilder) generiert werden
- + Interpretation (wieso reject/accept) möglich
- + modelliert Objektklassen
- - oft schwierig ein gutes Modell mit wenigen Parametern zu erstellen
- - Varianz des Modells unwichtig für Klassifikation

#### Diskriminative

- nur für gegebene Daten (Person oder nicht)
- + funktioniert wenn es unmöglich ist, die Daten selbst zu modellieren
- + oft in Praxis verwendet
- - keine Genauigkeit der Vorhersage
- - keine Interpretation

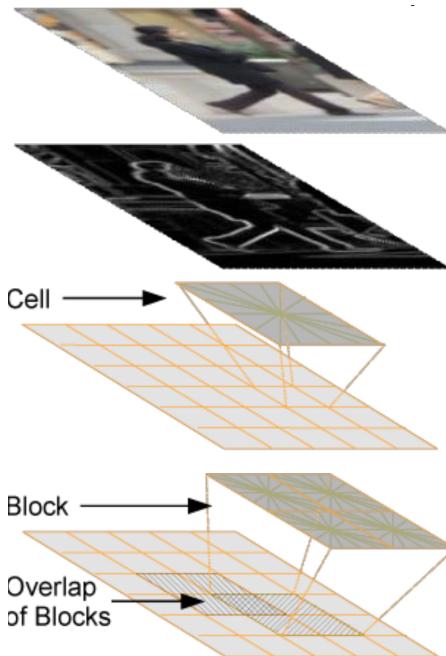
## 7.6 GRADIENT HISTOGRAMS

- extrem populär und erfolgreich
- vermeiden schwierige Entscheidungen
- Beispiele:
  - Histogram of Orientated Gradients (HOG)
  - Scale-Invariant Feature Transform (SIFT)
  - Gradient Location and Orientation Histogram (GLOH)

## 7.7 DER HOG PEOPLE DETECTOR

global, discriminative, appearance-based

1. Berechne Gradienten von Bildregionen (64x128px)
2. Berechne Orientierungshistogramm der Gradienten auf 8x8px Zellen (8x16 Zellen), typische Histogrammgröße 9 Bins
3. Normalisiere Histogramme mit überlappenden Blöcken (2x2 Zellen → 7x15 Blöcke), block descriptor hat Größe 4x9
4. Konkateniere block descriptors (7x15 (Anzahl) x 4x9 (Größe)) ergibt einen 3780-dimensionalen Featurevektor



## 7.8 SILHOUETTE MATCHING

global, discriminative, appearance-based

- Ziel: aligne Objektformen mit Bild
- Anforderungen an Alignment-Algorithmus: hohe Erkennungsrate, wenig FP, robust, einfache Berechnung (Komplexität  $O(\#positions * \#templates * \#contourpixels * \text{sizeof(searchregion)})$ )

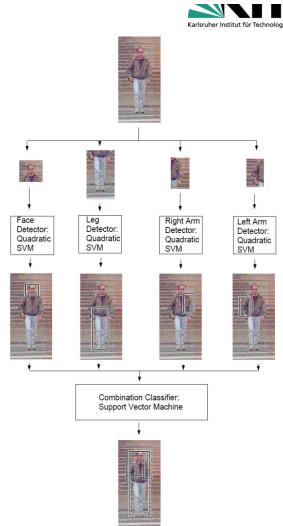
Chamfer Matching

- Berechne distance transform (DT)
- für jede mögliche Objektposition
  - positioniere bekannte Objektformen über DT
  - summiere Distanzen entlang der Kontur
  - behalte Instanzen wo die summierte Distanz unter einem Schwellwert ist

## 7.9 MOHAN PEOPLE DETECTOR

### relevant?? The Mohan Detector

- 4 parts
  - face and shoulder
  - legs
  - right arm
  - left arm
- Fixed layout
- Combination: Classifier (SVM)
- Detection
  - sliding window approach
  - 64x128 pixels
- Mohan, Object Detection in Images by Components, MIT Technical Report, 1999



## 7.10 IMPLICIT SHAPE MODEL

- Lerne viele local parts automatisch (visual vocabulary/bag of words/codebook)
- Lerne sternförmig strukturiertes Modell

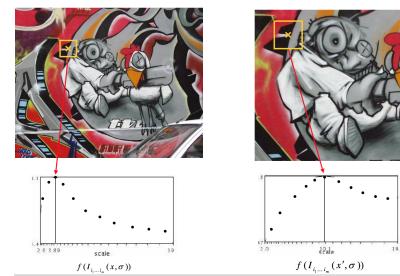
### 7.10.1 PART DETECTION

#### Keypoint Localization

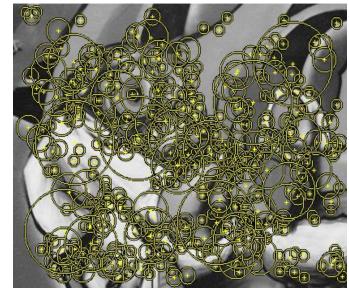
- Hessian Detector
  - suche 2D-Signaländerungen (starke 2. Ableitung)
  - $Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$
  - $det(H) = I_{xx}I_{yy} - I_{xy}^2, \ det(H) > 0 \rightarrow \text{Extremum}$



- Automatic Scale Selection
  - Funktionsantwort für Skalierungen



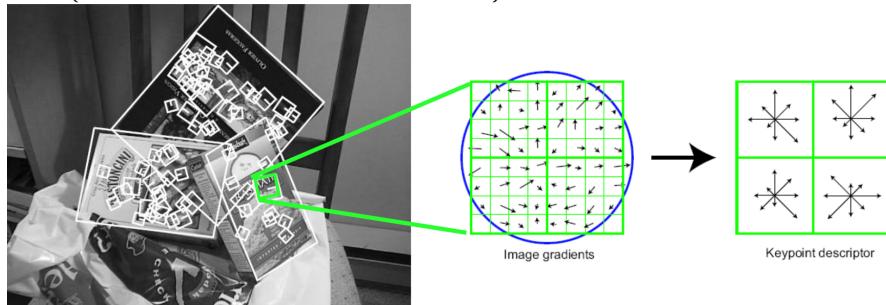
- Laplacian-of-Gaussian
  - „blob“ detector
  - suche lokale Maxima im Skalenraum des LoGs



### 7.10.2 PART DESCRIPTION

beschreiben Regionen im Umkreis von Keypoints

SIFT (Scale-Invariant Feature Transform)



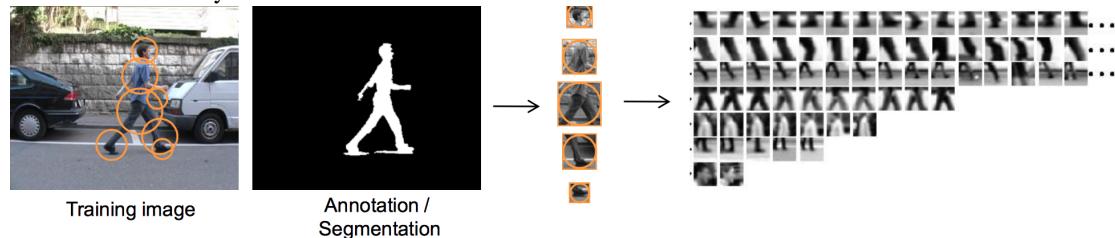
Histogramm von Gradientenorientierungen erkennt wichtige Texturinformationen und ist robust gegen geringe Verschiebungen/Verformungen

### SIFT matches      Blended image



### 7.10.3 LEARNING PART APPEARANCES

Visual Vocabulary

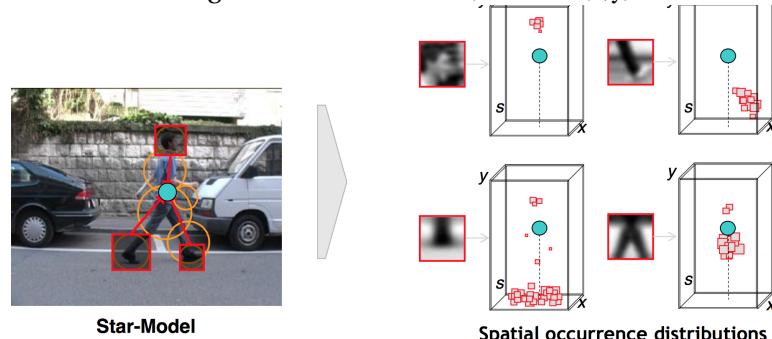


- detektiere Keypoints auf allen Samples
- bestimme lokale Deskriptoren für alle Keypoints
- gruppiere ähnliche lokale Deskriptoren (wiederkehrende Teile, missachte seltene Vorkomnisse → Deskriptor repräsentiert Teile des menschlichen Körpers)
- verwendet Clustering

### 7.10.4 LEARNING THE SPATIAL LAYOUT OF PARTS

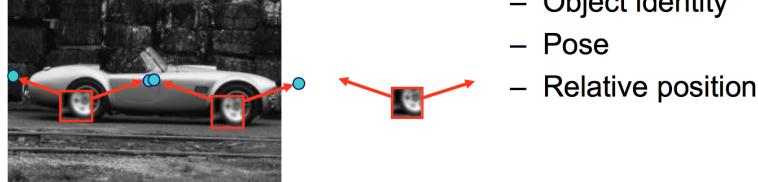
Spatial Occurrence (Star-Model)

- matche vocabulary Einträge auf Trainingsbilder
- erfasse Verteilungen der Vorkomnisse (Position ( $x,y$ ) und Skalierung)

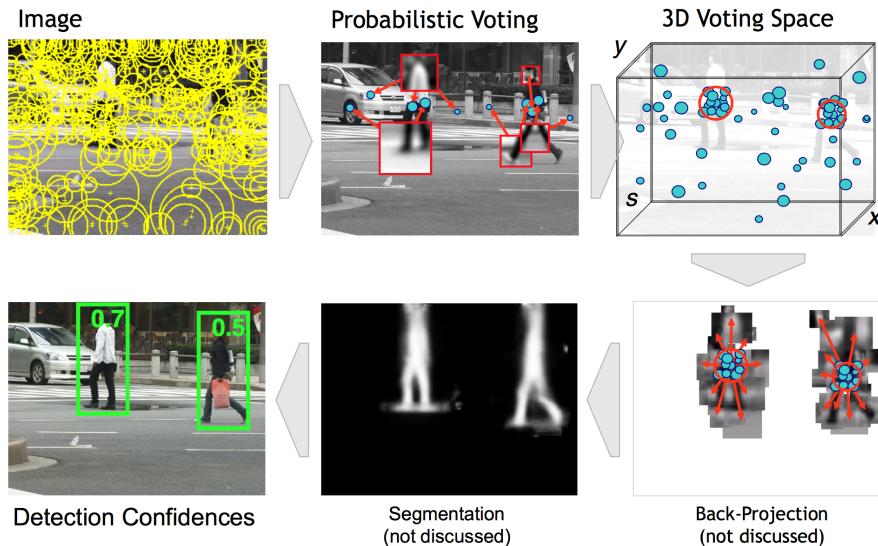


Generalized Hough Transform

- speichere mögliche Vorkomnisse für jedes Feature
- bei neuen Bildern führen die gematchten Features zu möglichen Objektpositionen
  - Object identity
  - Pose
  - Relative position



### 7.10.5 COMBINATION OF PART DETECTIONS



### 7.11 DEEP-LEARNING FOR PEOPLE DETECTION

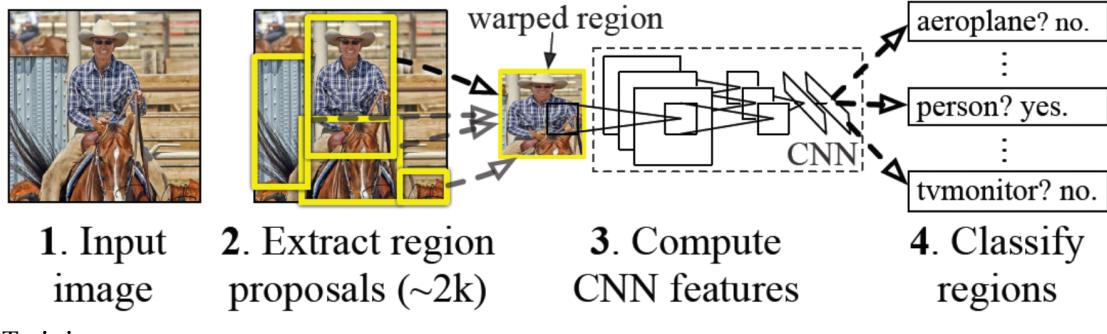
Tiefe Netzwerke sind bekannt für sehr gute Erkennungsraten, die Berechnungen sind jedoch aufwändig und machen es daher schwierig, Daten zu evaluieren. Sliding Window bspw. wäre viel zu langsam, daher Kaskaden-Ansatz (ähnlich wie Viola & Jones).

#### 7.11.1 DEEP NETWORK CASCades

- verwende bewährten Ansatz (vorgestellt: AlexNet) um gute Ergebnisse zu garantieren und beschleunige ihn mit Kaskaden
- Anforderung für frühe Kaskaden: schnell und alle TP
- AlexNet benötigt 40 Sekunden mit Sliding Window auf 640x480 Bild
- 1. Ansatz: frühere Stufen → kleinere Netze
- 2. Ansatz: VeryFast Detector als 1. Kaskadenstufe, Problem: ungenau, nicht alle TPs, Lösung: nutze nur erste 10% der Features

### 7.11.2 R-CNNs

#### R-CNN: Regions with CNN features



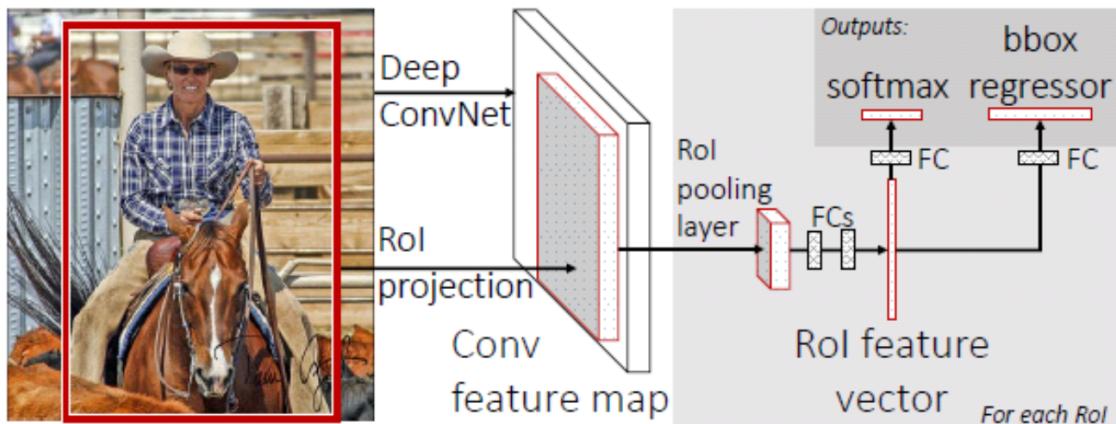
Training

1. Trainiere AlexaNet auf ImageNet-Daten (1000 Klassen)
2. Reinitialisiere die letzten Layer mit einer anderen Dimensionalität (abhängig von Klassenzahl des neuen Klassifikators) und trainiere neues Modell
3. Trainiere Klassifikator: binäre SVM für jede Klasse
4. Verbessere die vorgeschlagenen Regionen (Regressionsmodell um den geschätzten Ort zu verbessern mit den Features der vorgeschlagenen Regionen als Input und den Daten der neuen Region als Output)

Probleme

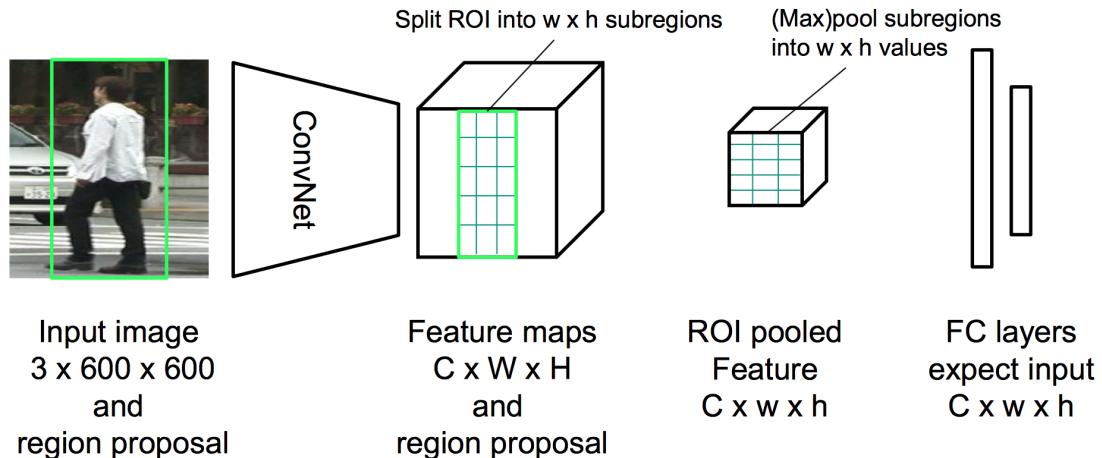
- Geschwindigkeit (jede Region durchläuft kompletten Zyklus im CNN)
- keine Adaption des CNN möglich, da SVM & BBox anschließend trainiert werden
- Komplexität

### 7.11.3 FAST R-CNNs



## ROI pooling

- Conv layers don't care about input size, FC layers do



Probleme: Zeit wird nicht für Vorhersage der Regionen verwendet, Modell deckt nicht alles ab, Vorschläge kommen von außen, sollen aber aus dem CNN kommen

### 7.11.4 FASTER R-CNNs

#### Region Proposal Network (RPN)

- Input: Feature map aus convolutional network
- Output: Liste von Vorschlägen
- Vorgehen: verschiebe kleines Netzwerk (RPN) über Feature Map
- evaluiere an jeder Position verschiedene Fenstergrößen

Training von Faster R-CNNs:

- Initialisiere conv layers (aus existierendem Modell) und trainiere RPN
- Initialisiere conv layers erneut und trainiere Detektor mit dem trainierten RPN
- Nutze conv layers aus 2 für Fine-Tuning des RPN (conv layers bleiben)
- Fine-Tuning des Detektors (conv layers bleiben)

### 7.11.5 SSD DETECTOR (CURRENT STATE OF THE ART)

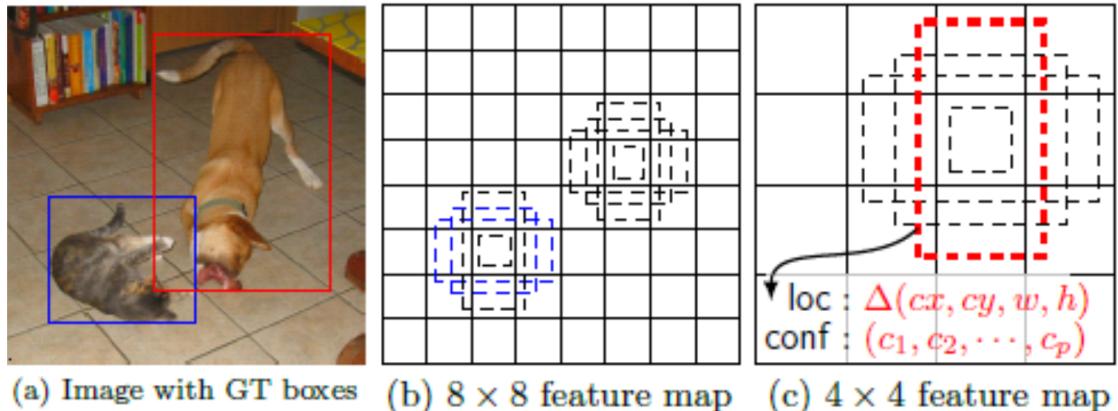
Bisher wurde folgender Ansatz verwendet:

- generiere Bounding Boxes (Vorschläge)
- resample Pixel/Features in Boxen zu einheitlicher Größe

- wende guten Klassifikator an
- → Idee: vermeide Resampling

SSD (Single Shot MultiBox Detector)

- verwende Set von Default-Boxen an jeder Position in Feature-Map
- Klassifiziere Objektklasse und Box-Regression für jede Default-Box
- Wende Boxen in verschiedenen Layern im ConvNet an (Layer verschiedener Größe machen Rescaling unnötig)



## 8 PEOPLE TRACKING

### 8.1 TRACKING VS. DETECTION

- Detektion
  - finde ein Objekt in einem Bild (Gesicht, Person, Körperteile, Landmarks)
  - Dynamik spielt keine Rolle
- Tracking
  - Bestimme Zustand (Position, Rotation, Deformierung, Pose) eines Objektes über eine Reihe von Bildern
  - Zustände in jedem Frame

### 8.2 TRACKING AS STATE ESTIMATION

- Vorhersage des Zustandes eines Systems
- Zustand kann nicht gemessen werden
- nur bestimmte Beobachtungen können gemacht werden

- Fehlerhafte Messungen
- Was ist der wahrscheinlichste Zustand  $x$ , den ein System zu einem Zeitpunkt  $t$  unter einer Sequenz von Beobachtungen  $Z_t$  haben wird?  
 $\text{argmax}_p(x_t|Z_t)$

### 8.2.1 BAYES FILTER

- Annahme: Zustand  $x$  ist Markov-Prozess  
 $p(x_t|x_{t-1}, x_{t-2}, \dots, x_0) = p(x_t, x_{t-1})$
- Zustände  $x$  generieren Beobachtungen  $z$   
 $p(z_t|x_t, x_{t-1}, \dots, x_0) = p(z_t|x_t)$
- Schätze wahrscheinlichsten Zustand  $x_t$  mit gegebener Sequenz  $Z_t$   
 $\text{argmax}_p(x_t|Z_t)$
- kann rekursiv angewendet werden
- Vorhersage:  $p(x_t|Z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1})dx_{t-1}$
- Aktualisierung  $p(x_t|Z_t) = \alpha p(z_t|x_t)p(x_t|Z_{t-1})$
- benötigt Prozessmodell  $p(x_t|x_{t-1})$  und Modell für Messung  $p(z_t|x_t)$

### 8.2.2 KALMAN FILTER

- Instanz eines Bayes-Filters
- zusätzliche Annahmen: Zustandspropagierung und Modell für Messung sind linear, Gaussian Prozess und Messungsgenauigkeit
- geschätzter Prozess:  $x_k = Ax_{k-1} + w_{k-1}$  und  $z_k = Hx_k + v_k$  mit Zustand  $x_k$  und Beobachtung  $z_k$  zur Zeit  $k$ , Transition-Matrix  $A$  und Messungsmatrix  $H$
- nur anwendbar wenn Prozess und Messungs-Prozess-Beziehung nicht linear → Extended Kalman Filter (EKF) linearisiert über Durchschnitt und Kovarianz

### 8.2.3 PARTICLE FILTER, CONDENSATION ALGORITHM

- Kalman Filter problematisch wenn Messungstiefe nicht gaussverteilt
- Partikel Filter repräsentiert und propagiert willkürliche Wahrscheinlichkeitsverteilungen repräsentiert in einem Set von gewichteten Samples
- numerisch (Kalman analytisch)
- verwendet wie auch Kalman Filter dynamisches Modell zur Beschreibung von Bewegungen

Verwendet Bayes Regel zum Tracking  $\operatorname{argmax}_{x_t} p(x_t|Z_t) = \operatorname{argmax}_{x_t} p(z_t|x_t)p(x_t|Z_{t-1})$  mit (nach Isard & Blake)  $p(x_t|Z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1})$  und der vereinfachenden Behauptung (Markov)  $p(x_t|X_{t-1}) = p(x_t|x_{t-1})$   
Partikel Filter benötigt

- Set von  $N$  gewichtetet Sampeln  $\{(s_k^{(i)}, \pi_k^{(i)}) | i = 1..N\}$
- Motion Modell  $s_k^{(i)} \leftarrow s_{k-1}^{(i)}$
- Beobachtungsmodell  $\pi_k^{(i)} \leftarrow s_k^{(i)}$

Condensation Algorithmus

1. Wähle zufällig  $N$  neue samples  $s_k^{(i)}$  aus dem alten Set  $s_{k-1}^{(i)}$  abhängig von den Gewichten  $\pi_{k-1}^{(i)}$
2. Vorhersage: Propagiere die Samples mit dem Motion Modell
3. Messung: Berechne Gewichter für die neuen Samples mit dem Beobachtungsmodell  $\pi_k^{(i)} = p(z_k|x_k = s_k^{(i)})$

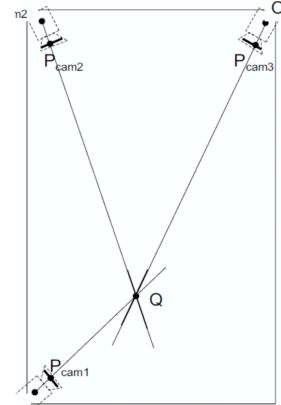
## 8.3 MULTI-CAMERA SYSTEMS

### 8.3.1 KALIBRIERUNG

1. Kalibrierungsziel mit bekannter Geometrie wird für jede Kamera aus verschiedenen Blickwinkeln aufgenommen
2. Eckpunkte werden automatisiert extrahiert
3. Positionen der Eckpunkte werden verwendet um die *intrinsics* iterativ zu schätzen
4. Anschließend wird ein festes Kalibrierungsziel von allen Kameras aufgenommen → *extrinsics*

### 8.3.2 TRIANGULATION

- Voraussetzung: Objektposition ist in verschiedenen Blickwinkeln bekannt
- idealerweise bestimmt der Schnitt der Blickwinkel die 3D-Position des Objekts
- Praxis: least-squares Approximation



## 8.4 MULTI-OBJECT TRACKING

Problem mit expliziter Triangulation: viele mögliche Treffer → Segmentiere Vordergrund und Hintergrund

### 8.4.1 AUDIO-VISUAL TRACKING OF A SPEAKER / PERSON-TRACKING FOR A ROBOT

- Ziel: Sprecher in einer Lesung
- Sensoren: Mehrere fixierte Kameras und Mikrofone
- Features: Hintergrundsubtraktion, Gesichts- und Oberkörperdetektion, GCC von Audiosignal
- Trackingverfahren: Partikel Filter

## 8.5 ARTICULATED BODY TRACKING

- Automatische Aufzeichnung und Analyse von Körperbewegungen über Zeit (Bewegung von Körperteilen im Ggs. zu Tracking von rigidem Körpern)
- Anwendungen: Sportanalyse, Biomedizinische Forschung, Computeranimationen, Filme, Gesten- und Aktionserkennung

### 8.5.1 PARTIKELFILTER FÜR TRACKING

- Körpermodell: 14 Segmente, 10 Gelenke, 32 Freiheitsgrade
- Sensoren: 3 kalibrierte Kameras
- Features: Hintergrundsubtraktion
- Tracking: Partikelfilter
- Herausforderung: großer Suchraum

### 8.5.2 VOLUME CARVING / VOXELS

- Approximiere die visuelle Hülle durch Voxel (Volumenpixel)
- Problem: sehr teure Berechnung
- Lösungen: bessere Hardware, Lookup-tables, räumliche Strukturen (z.B. Octree-based voxels)

## 9 GESTURE RECOGNITION

### 9.1 DEFINITION

Eine Bewegung meist von Körpern oder Gliedmaßen, die eine Idee, ein Sentiment oder eine Einstellung zu Ausdruck bringt

### 9.2 ANWENDUNGEN

Multimodale Interaktion

- Gesten- und Spracherkennung
- Mensch-Roboter Interaktion
- Interaktion mit Smart Environments

Verständnis der menschlichen Interaktion: Menschen verwenden viel non-verbale Kommunikation (Mimik, Gestik, ...)

### 9.3 ARTEN VON GESTEN

- Gesten mit Händen & Armen: Zeigen, Zeichensprache, Winken, Daumen hoch/runter, victory, call-me, etc.
- Gesten mit dem Kopf: Kopfschütteln, Nicken
- Gesten mit dem Körper: Don't know / shrug
- Problem: Gesten sind kulturspezifisch
- Manche Gesten werden eng mit der Sprache koordiniert

Definition, applications, types of gestures Taxonomies Building blocks

### 9.4 HIDDEN MARKOV MODELS (HMM)

- „hidden“: Beobachte und ziehe Schlussfolgerungen ohne die *hidden* Zustandsfolge zu kennen
- Markov Voraussetzung (1. Ordnung): Folgezustand hängt nur vom aktuellen Zustand ab (nicht von der kompletten bisherigen Zustandsfolge)
- 5er Tupel  $(S, \pi, A, B, V)$ 
  - Set von Zuständen  $S = \{s_1, s_2, \dots, s_n\}$
  - Initiale Wahrscheinlichkeitsverteilung  $\pi$ ,  $\pi(s_i) =$  Wahrscheinlichkeit dass  $s_i$  der erste Zustand der Sequenz ist

- Matrix der Zustandsänderungen  $A = (a_{ij})$  wobei  $a_{ij}$  die Wahrscheinlichkeit, dass Zustand  $s_j$  auf Zustand  $s_i$  folgt, ist
- $B = \{b_1, b_2, \dots, b_n\}$ , wobei  $b_i(x)$  die Wahrscheinlichkeit ist, dass  $x$  in Zustand  $s_i$  beobachtet wird
- beobachtbarer Featureraum  $V$  kann diskret  $V = \{x_1, x_2, \dots, x_v\}$  oder kontinuierlich  $V = \mathbb{R}^d$  sein

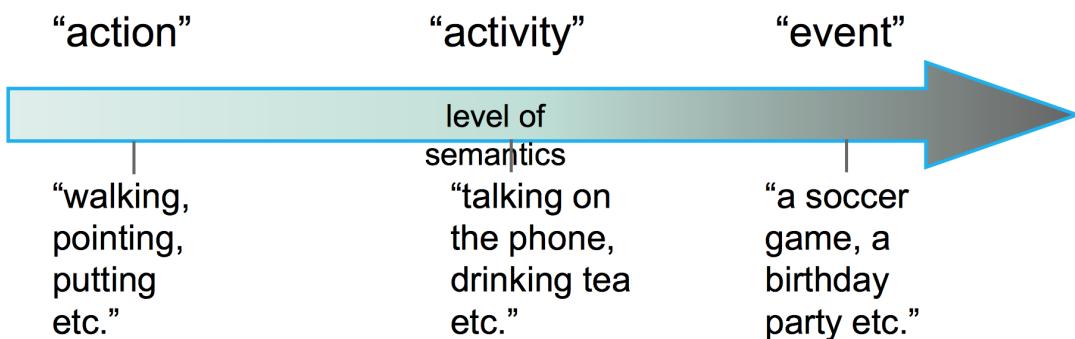
3 Hauptaufgaben von HMM (HMM  $\lambda$  und Beobachtungen  $x_1, x_2, \dots, x_t$ )

- Evaluationsproblem: Berechne die Wahrscheinlichkeit der Beobachtungen  $p(x_1, x_2, \dots, x_T | \lambda)$ ,  
→ Forward-Algorithmus
- Decodingproblem: Berechne die wahrscheinlichste Zustandssequenz  $s_{q1}, s_{q2}, \dots, s_{qT}$  d.h.  
 $\arg\max_{q1, \dots, q_T} p(q_1, \dots, q_T | x_1, x_2, \dots, x_T, \lambda)$ , → Viterbi-Algorithmus
- Learning/Optimizationproblem: Finde HMM  $\lambda'$  dass gilt:  $p(x_1, x_2, \dots, x_T | \lambda') > p(x_1, x_2, \dots, x_T | \lambda)$ ,  
→ Baum-Welch-Algo oder Viterbi-Lernen

#### 9.4.1 ANWENDUNGEN

- Sign Language Recognition (Starner et al): Auswertung von 6000 Gesten der amerikanischen Zeichensprache
- Pointing Gesture Recognition (Nickel et al): Erkenne menschliche Zeigegesten und extrahiere 3D Zeigerichtung
- Interaction with a video wall

## 10 ACTION & ACTIVITY RECOGNITION



Motivation:

- Bisher: Position von Personen, Bewegungen, Orientierungen

- Ziel: Was macht eine Person (laufen, sitzen, arbeiten, sich verstecken)?
- Wie macht sie dies?
- Was passiert in einer Szene (Meeting, Party, Telefonat, etc)?

Anwendungen:

- Video-Analyse
- Smart-Rooms
- Patienten-Überwachung
- Überwachung allg.
- Robotik

## 10.1 ANSÄTZE

### 10.1.1 LEFT-TO-RIGHT HMMs

- Ansatz inspiriert durch Spracherkennung
- Erkennt Bewegungsprimitive: berechne globale Bewegungsfeatures für jeden Frame einer Videosequenz, HMM Klassifizierung von folgenden Streams
- Aktions-/Aktivitätserkennung: Kombiniere komplexe Sequenzen von Bewegungsprimitiven mit Grammatiken und statischen Modellen

Bewegungsfeatures (motion features)

- Marker-basiert: Körpereckenwinkel erhalten durch Trackingsystem
- Video-basiert: Histogramm der Werte des optischen Flusses für alle Winkel

Erkennung von Bewegungsprimitiven:

- Ein globales Feature pro Frame (Histogramm der Bewegungswinkel)
- kontinuierlicher left-to-right HMM Klassifikator (24 Aktionsprimitive jeweils durch 1 HMM modelliert, 4 Zustände pro Bewegungsprimitiv, 16 Gaussians pro Zustand)

Ergebnisse:

- geringe Fehlerrate
- echtzeitfähig
- sehr genaue Erkennungsgenaugigkeit
- Probleme mit zyklischen Bewegungen wegen deren kurzer Dauer
- Viel Trainingsdaten mit Anmerkungen benötigt
- Generalisierungsprobleme

### 10.1.2 LAYERED HMMs

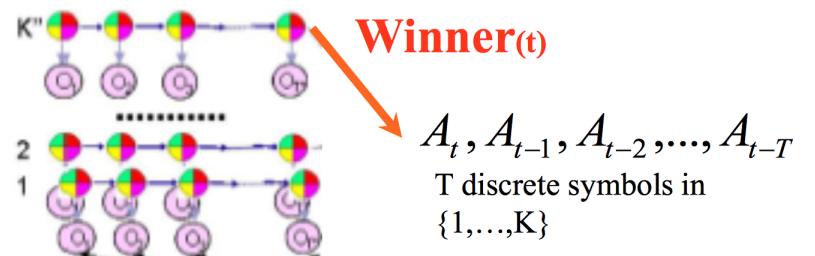
- Ziel: Erkenne komplexe menschliche Aktivitäten über längeren Zeitraum
- Typen von Kontext (Situationen, Aktivitäten, etc): Telefonate, persönliche Unterhaltungen, Präsentationen, Konversation auf Distanz, ...
- Sensoren: Mikrophone, USB Kamera, Maus, Tastatur

Hierarchischer Ansatz:

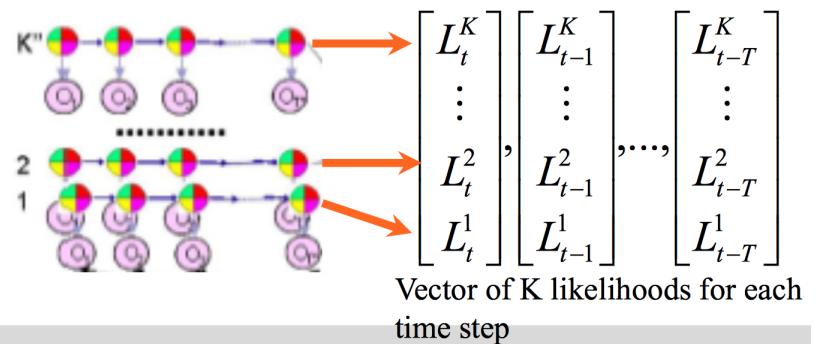
- Probleme mit normalen HMMs: fehlende Struktur, hoher Parameterraum, Overfitting auf langen Sequenzen mit wenig Trainingsdaten (schlecht Generalisierung), Fusion von verschiedenen Streams ist möglich, multipliziert aber die Anzahl an benötigten Parametern
- Lösung: Hierarchische (Layered) HMMs (LHMMs)
  - Klassen: Niemand vorhanden, eine Person, mehrere Personen, Musik, Stille, Telefonklingeln
  - Aktivitäten: Telefonat, persönliche Konversation, Präsentation, Konversation mit Distanz
- HMMs in Level L benutzen sliding windows von  $T^L$  samples
  - Daten in Zeitfenster in Level L werden analysiert
  - Likelihoods werden berechnet
  - Ergebnis wird an Level L+1 als Input übergeben
- Windowlänge variiert mit jedem Level
  - je höher das Level, desto höher die Zeitskala  $T^L$
  - höhere Level modellieren längere Aktivitäten
  - Abstraktionslevel wird bei höheren Leveln erhöht
- HMMs auf niedrigeren Ebenen werden unabhängig (separat für jeden Stream) mit Baum-Welch trainiert
- Low level HMMs erkennen feingranulierte Kontext
- Output von loweren Ebenen wird an höhere Level weitergeleitet
- 2 Ansätze:
  - Maxbelief: nur Informationen vom wahrscheinlichsten HMM werden weitergegeben

- Distributional: komplette Wahrscheinlichkeitsverteilung über Modelle wird weitergegeben

■ Maxbelief:



■ Distributional:



Vorteile:

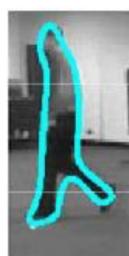
- weniger Parameter als vergleichbare herkömmliche HMMs (Overfitting ist weniger wahrscheinlich)
- Lower level HMMs können getrennt neutrainiert werden (mögliche Adaption an sich ändernde Umgebung)
- intuitivere, strukturiertere Repräsentation

## 10.2 APPROACHES INSPIRED FROM OBJECT RECOGNITION

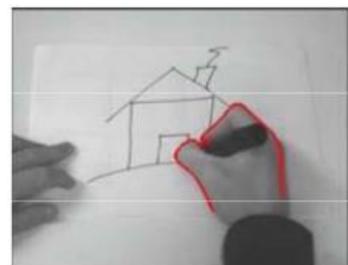
### 10.2.1 SPATIO TEMPORAL FEATURES DESCRIPTORS



**Temporal templates:**  
 + simple, fast  
 - sensitive to segmentation errors



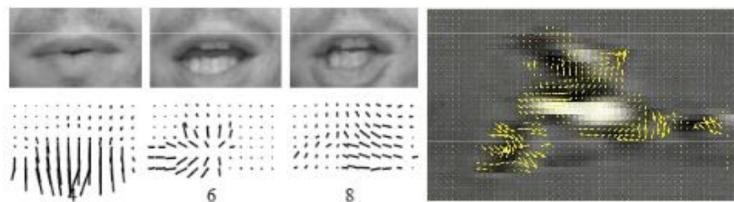
**Active shape models:**  
 + shape regularization  
 - sensitive to initialization and tracking failures



**Tracking with motion priors:**  
 + improved tracking and simultaneous action recognition  
 - sensitive to initialization and tracking failures

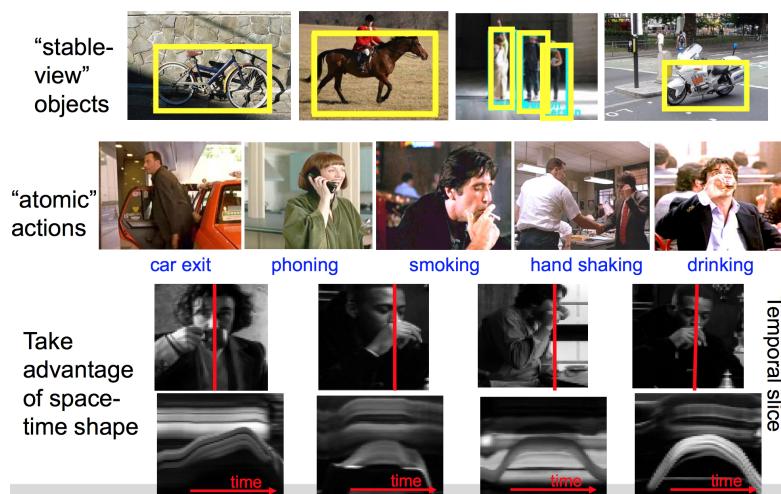
#### Motion-based recognition:

+ generic descriptors;  
 less depends on appearance  
 - sensitive to localization/tracking errors



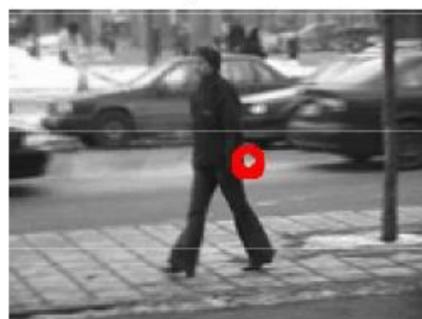
#### Actions == space-time objects?

- versuche sowohl Raum als auch Zeit zu modellieren
- Können Aktionen als Raum-Zeit Objekte betrachtet werden? → Wende Objektdetektoren auf Aktionserkennung an
- Hier werden nur atomare (= einfache) Aktionen betrachtet

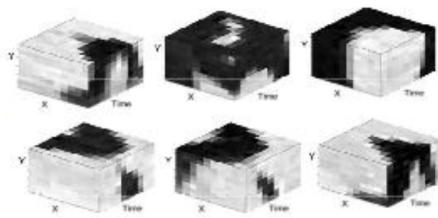


## Bag of space-time features

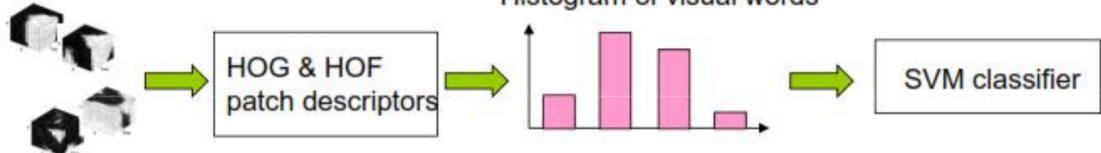
### Extraction of space-time features



### Collection of space-time patches

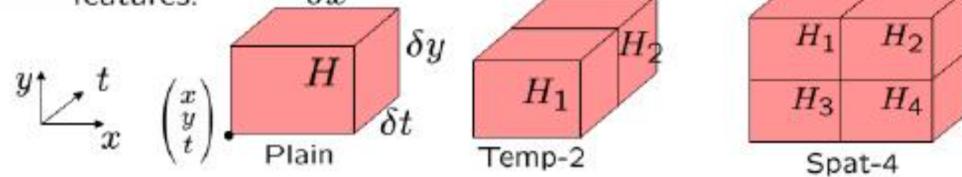


### Histogram of visual words



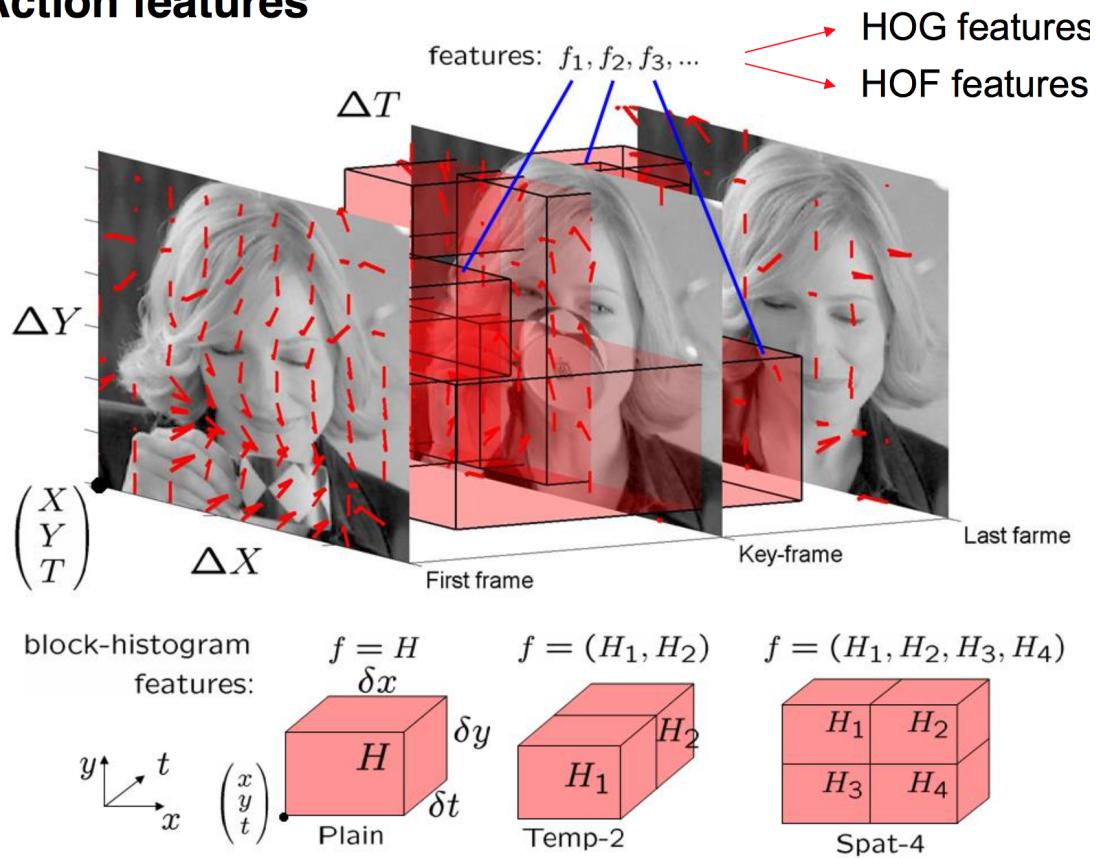
### Action features

block-histogram features:  $f = H_{\delta x}$        $f = (H_1, H_2)$        $f = (H_1, H_2, H_3, H_4)$



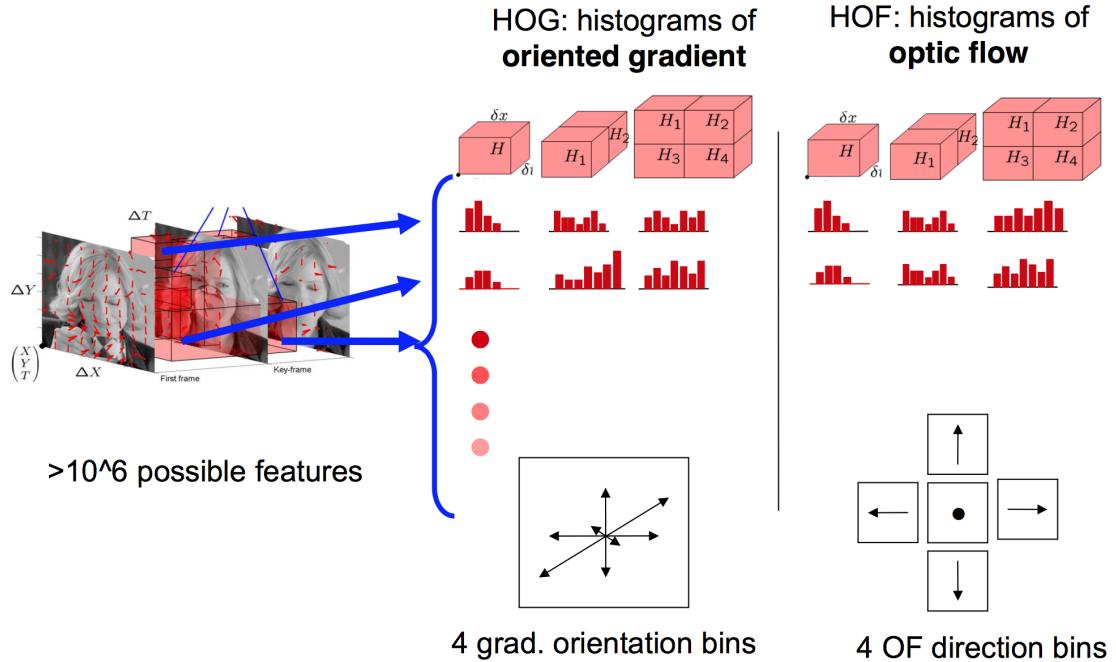
- Action volume = Space-Time kubische Region um den Kopf (Dauer der Aktion)
- codiert mit Block-Histogramm Features  $f_\theta(\cdot), \theta = (x, y, t, dx, dy, dt, \beta, \phi)$

## Action features



### Histogram features

- vereinfachter HoG: Wende Gradientenoperator (z.B. Sobel) auf jeden Frame in Sequenz an, Bin Gradienten in 4 Orientierungen
- Histogram of optical flow (Hof): berechne optischen Flow zwischen Frames, Bin OF Vektoren in 4 Richtungsbins (+1 für keine Bewegung)
- Normalisierter Aktionskubus hat Größe 14x14x8 mit Units der Größe 5x5x5 Pixel (mehr als 1 Mio. möglich Features)



### 10.2.2 BAG-OF-WORDS MODEL

- Visual Word vocabulary learning
  - Clustern lokaler Features
  - Visual Words entsprechender Clustermittelpunkten
- BoW feature Berechnung
  - Weißt jedem lokalen Feature das ähnlichste visual word zu
  - Bow Feature entspricht dem Histogramm der visual word Vorkommnisse innerhalb einer Region

### 10.2.3 DETEKTION VON SPACE-TIME FEATURES

#### Space-Time Interest Points (STIP):

- Space-Time Erweiterung des Harris-Operators (fügt Dimensionalität der Zeit zur second moment Matrix hinzu), sucht nach Maxima in erweiterter Harris corner Funktion H
- Detektion abhängig von Raum-Zeit Skalierung
- Extrahiere Features in mehreren Levels der Raum-Zeit (dense scale sampling)
- Berechne Histogramm-Deskriptoren der Space-Time Volumen in Nachbarschaft der detektierten Punkte: 4-bin HOG für jeden Kubus im 3x3x2 Raum-Zeit Grid, 4-bin HOFO für jeden Kubus im 3x3x2 Raum-Zeit Grid

#### 10.2.4 BOOSTING BASED ACTION RECOGNITION

- verwendet Boosting (z.B. AdaBoost) um Features (Block-Histogramm Features) eines Actionvolumens zu klassifizieren

#### 10.2.5 KLASIFIKATION VON AKTIONEN

- Spatio-temporal BoW
  - Erstelle visual vocabulary der lokalen Feature-Repräsentationen mittels k-means Clustering
  - Weise jedes Feature in einem Video dem nächsten vocabulary word zu
  - Berechne Histogramm von visual word Vorkommnissen über spacial time Volumen einer Video Sequenz
- SVM Klassifikation
  - Kombiniere verschiedene Featuretypen mit Multichannel Kernel
  - One-against-all approach bei Multi-Class Klassifikation

#### 10.2.6 DENSE TRAJECTORIES

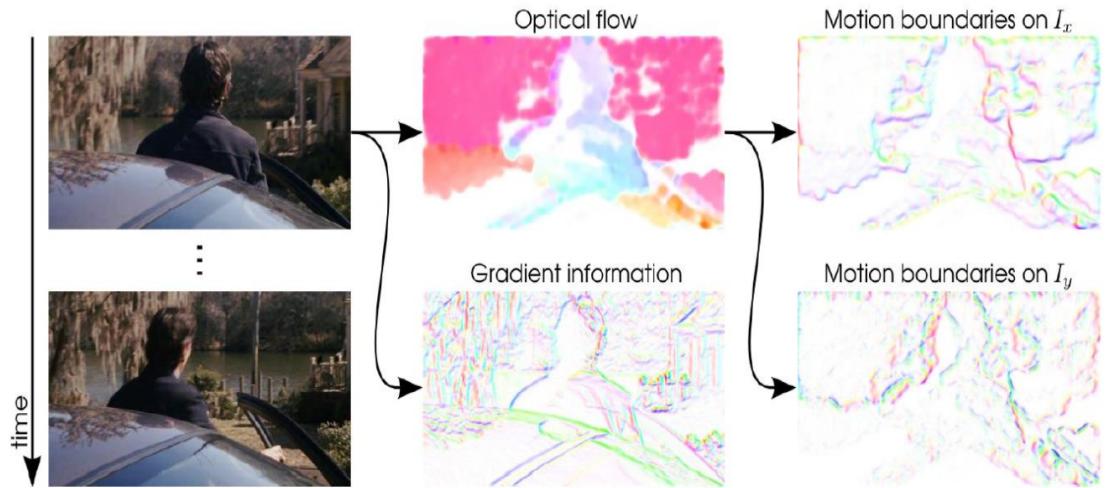
##### Featuretrajektorien

- Effiziente Videorepräsentation
- extrahiert mit KLT tracker oder matching SIFT Deskriptoren über Frames
- Qualität und Quantität ungenügend
- besser: Dense Trajectories (state of the art)
- Generiere Trajektorien durch Tracking von OF auf tief gesampelten Punkten
- Sample Features bei jedem 5. Pixel
- Entferne untrackbare Punkte (Strukturen-/Eigenwertanalyse)
- Sample Punkte auf 8 verschiedenen Skalierungen
- Tracking durch Medianfilter im OF-Feld
- Länge der Trajektorie ist fix (z.B. 15 Frames)

##### Trajektorien-Deskriptoren

- Histogram of Oriented Gradient (HOG)
- Histogram of Optical Flow (HOF)

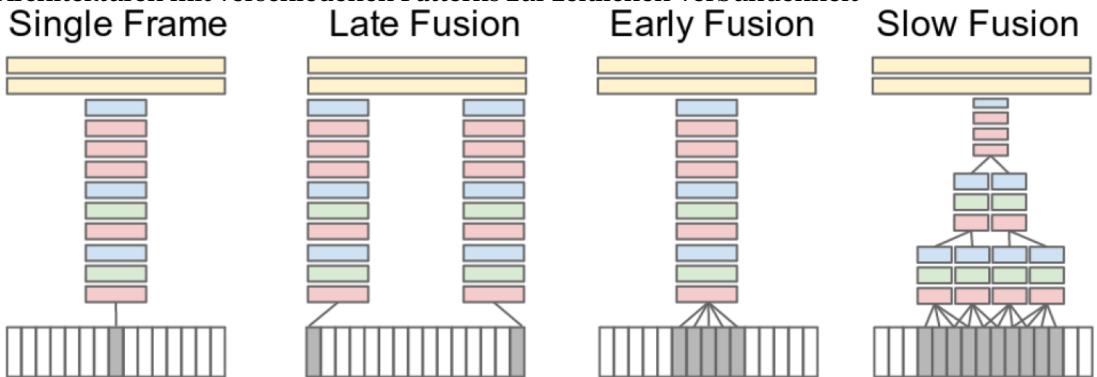
- HOGHOF
- Motion Boundary Histogram (lokale Gradienten von x- und y-Fluss und berechne HOG wie in statischem Bild)



#### 10.2.7 CNNs / USING MULTIPLE FRAMES / MULTIPLE FUSION APPROACHES

Adaption von CNNs zur Videoklassifikation abhängig von Aktivitäten?

- Architekturen mit verschiedenen Patterns zur zeitlichen Verbundenheit



- Multiresolution CNN: schnelleres Training bei geringerer Eingabegröße

