Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

GAUSSIAN MIXTURE MODELS UND DER EM-ALGORITHMUS

Fachseminar "Machine Learning"

Letztes Update: 3. Februar 2016

Marvin Suhr

Studienbereich Informatik
Hochschule RheinMain

GLIEDERUNG

1. Einleitung

2. Gaussian Mixture Models

3. Der EM-Algorithmus

4. Anwendung / Auswertung

EINLEITUNG

DAS PROBLEM

- Wir möchten aufgezeichnete Videos automatisch in Vorder- und Hintergrund einteilen

DAS PROBLEM

- Wir möchten aufgezeichnete Videos automatisch in Vorder- und Hintergrund einteilen
- Wichtig im Bereich der Videoüberwachung

DAS PROBLEM

- Wir möchten aufgezeichnete Videos automatisch in Vorder- und Hintergrund einteilen
- Wichtig im Bereich der Videoüberwachung
- Ziel: Arbeit ersparen (nur "interessante" Aufzeichnungen speichern, also Sequenzen in denen sich etwas im Vordergrund bewegt)

DAS PROBLEM

- Wir möchten aufgezeichnete Videos automatisch in Vorder- und Hintergrund einteilen
- Wichtig im Bereich der Videoüberwachung
- Ziel: Arbeit ersparen (nur "interessante" Aufzeichnungen speichern, also Sequenzen in denen sich etwas im Vordergrund bewegt)
- Falls sich nur Dinge im Hintergrund bewegen, sollen die entsprechenden Bilder automatisch aussortiert werden

ANSÄTZE

Ein solches System lässt sich auf verschiedene Weisen implementieren:

- Durchschnitt der Pixelwerte bilden (Probleme bei vielen / beweglichen Objekten)

ANSÄTZE

Ein solches System lässt sich auf verschiedene Weisen implementieren:

- Durchschnitt der Pixelwerte bilden (Probleme bei vielen / beweglichen Objekten)
- Jedes Pixel mit einem Kalman-Filter modellieren (erholt sich nur langsam wenn der Hintergrund sich verändert)

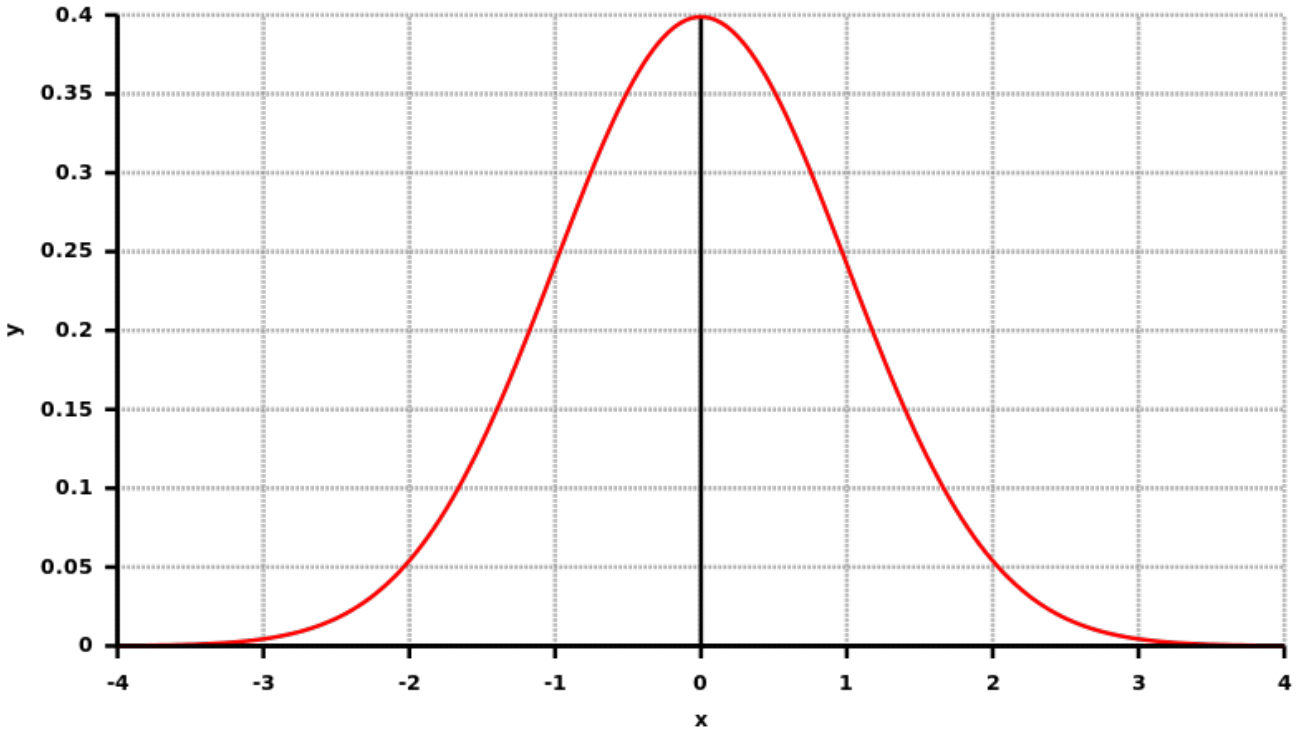
ANSÄTZE

Ein solches System lässt sich auf verschiedene Weisen implementieren:

- Durchschnitt der Pixelwerte bilden (Probleme bei vielen / beweglichen Objekten)
- Jedes Pixel mit einem Kalman-Filter modellieren (erholt sich nur langsam wenn der Hintergrund sich verändert)
- Jedes Pixel mit einer Mischung von Gauß-Funktionen modellieren (wird hier näher betrachtet)

GAUSSIAN MIXTURE MODELS

DIE GAUSSSCHE GLOCKENFUNKTION



Aus Wikipedia: Normalverteilung

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(hier mit $\mu = 0,\ \sigma = 1$)

DIE GAUSSSCHE GLOCKENFUNKTION

- Ein wichtiger und häufig verwendeter Typ von Wahrscheinlichkeitsverteilungen

DIE GAUSSSCHE GLOCKENFUNKTION

- Ein wichtiger und häufig verwendeter Typ von Wahrscheinlichkeitsverteilungen
- Wird oft im Gebiet der Statistik verwendet, in verschiedensten Kontexten (z.B. Körpergröße, Gewicht, ...)

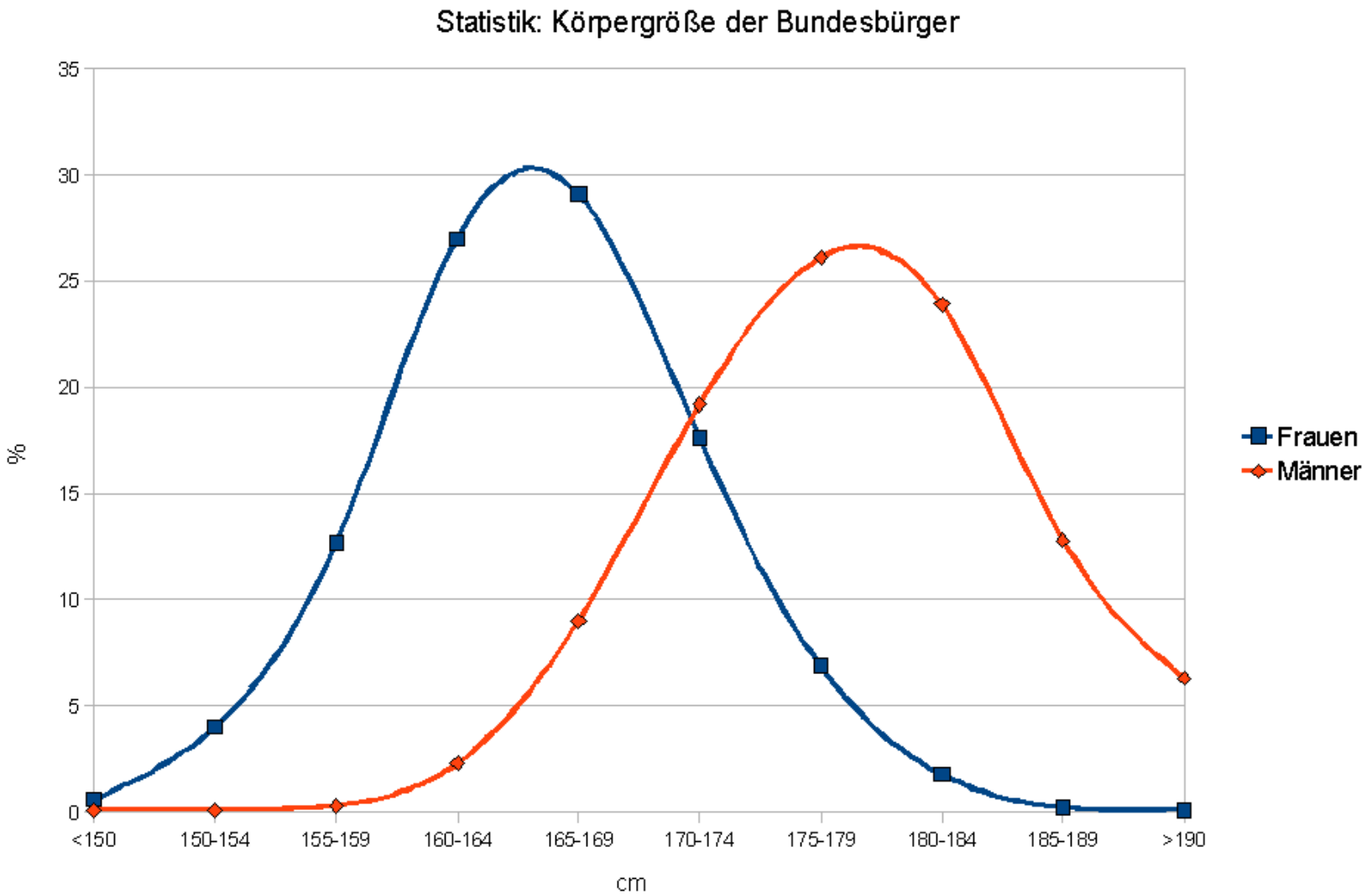
DIE GAUSSSCHE GLOCKENFUNKTION

- Ein wichtiger und häufig verwendeter Typ von Wahrscheinlichkeitsverteilungen
- Wird oft im Gebiet der Statistik verwendet, in verschiedensten Kontexten (z.B. Körpergröße, Gewicht, ...)
- Beschreibt die Abweichung von gemessenen Werten zum Mittelwert und die Wahrscheinlichkeit dazu

DIE GAUSSSCHE GLOCKENFUNKTION

- Ein wichtiger und häufig verwendeter Typ von Wahrscheinlichkeitsverteilungen
- Wird oft im Gebiet der Statistik verwendet, in verschiedensten Kontexten (z.B. Körpergröße, Gewicht, ...)
- Beschreibt die Abweichung von gemessenen Werten zum Mittelwert und die Wahrscheinlichkeit dazu
- Meistens sehr nah an den wirklichen Werten

BEISPIEL KÖRPERGRÖSSE



Aus Wikipedia: Körpergröße

GAUSSIAN MIXTURE MODELS (GMM)

Wie der Name schon vermuten lässt, besteht ein GMM aus einer Mischung von verschieden gewichteten Gauß-Funktionen (Gaussians):

$$p(x) = \sum_{i=1}^{n} w_i \cdot \mathcal{N}(x|\mu_i, \Sigma_i)$$

GAUSSIAN MIXTURE MODELS (GMM)

Wie der Name schon vermuten lässt, besteht ein GMM aus einer Mischung von verschieden gewichteten Gauß-Funktionen (Gaussians):

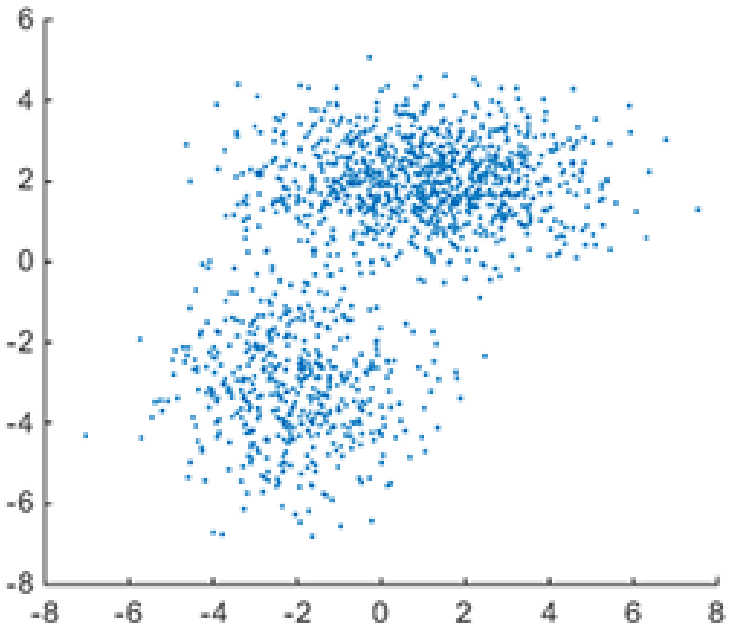
$$p(x) = \sum_{i=1}^{n} w_i \cdot \mathcal{N}(x|\mu_i, \Sigma_i)$$

n: Anzahl der Gauß-Verteilungen

w_i: Gewichtung der i-ten Verteilung ($\sum_{i=1}^{n} w_i = 1$)

$\mathcal{N}(x|\mu_i, \Sigma_i)$: Erweiterung der Normalverteilung (Kovarianzmatrix Σ statt Varianz σ^2)

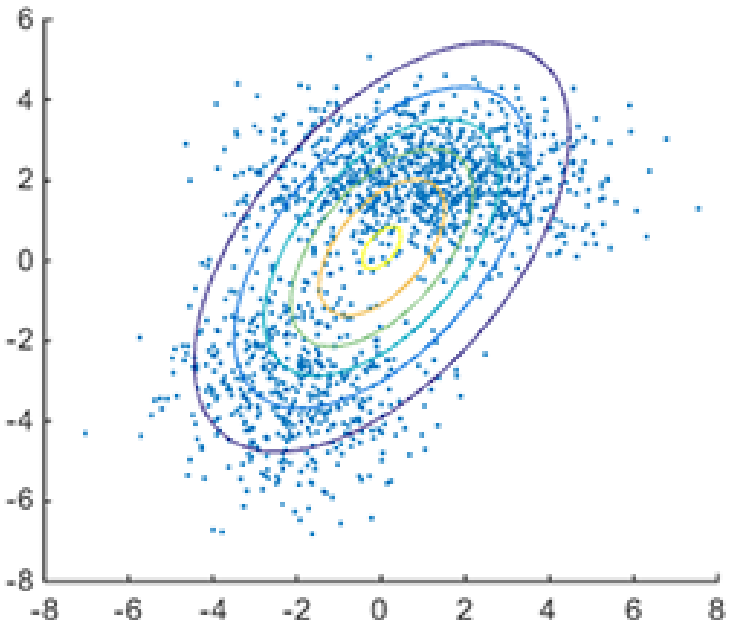
BEISPIELPLOT



Aus http://recognize-speech.com/basics/introduction-to-gaussian-mixture-models/14-basics

- Man sieht auf den ersten Blick, dass die Stichproben zwei Cluster bilden

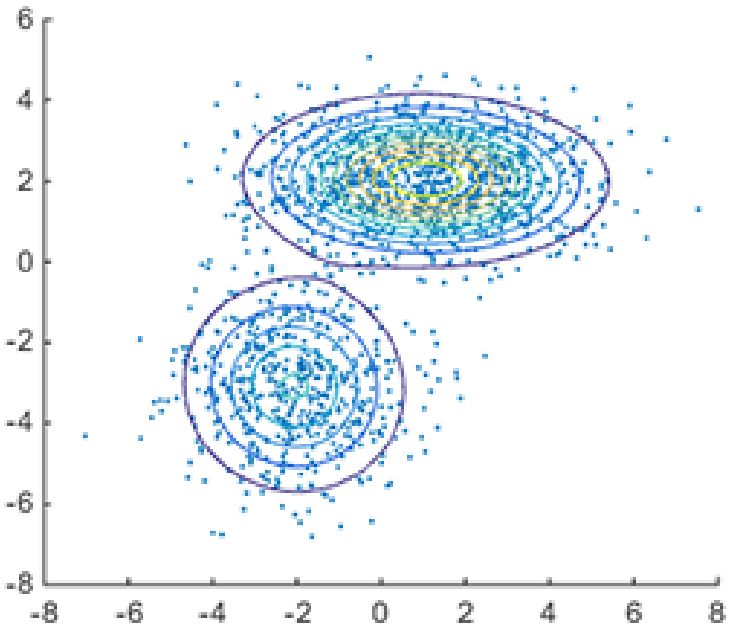
GAUSS-VERTEILUNG, AN DAS BEISPIEL ANGEPASST



Aus http://recognize-speech.com/basics/introduction-to-gaussian-mixture-models/14-basics

- Man erkennt, dass eine einzige Gauß-Funktion nicht richtig passt bzw. nicht ausreicht um beide Cluster zu modellieren

GMM, AN DAS BEISPIEL ANGEPASST



Aus http://recognize-speech.com/basics/introduction-to-gaussian-mixture-models/14-basics

- Wenn wir stattdessen ein GMM verwenden, werden die beiden Cluster problemlos erkannt

ANWENDUNG GMM

- Unser Problem: Oft sind die Parameter μ und Σ sowie die Zugehörigkeit zu den Clustern unbekannt (wir kennen nur die Stichproben)

ANWENDUNG GMM

- Unser Problem: Oft sind die Parameter μ und Σ sowie die Zugehörigkeit zu den Clustern unbekannt (wir kennen nur die Stichproben)

- Lösung: GMMs können mithilfe von verschiedenen Machine-Learning-Konzepten "trainiert" werden

ANWENDUNG GMM

- Unser Problem: Oft sind die Parameter μ und Σ sowie die Zugehörigkeit zu den Clustern unbekannt (wir kennen nur die Stichproben)
- Lösung: GMMs können mithilfe von verschiedenen Machine-Learning-Konzepten "trainiert" werden
- Das bedeutet: wir bestimmen die uns unbekannten Parameter der einzelnen Gaussians

ANWENDUNG GMM

- Unser Problem: Oft sind die Parameter μ und Σ sowie die Zugehörigkeit zu den Clustern unbekannt (wir kennen nur die Stichproben)
- Lösung: GMMs können mithilfe von verschiedenen Machine-Learning-Konzepten "trainiert" werden
- Das bedeutet: wir bestimmen die uns unbekannten Parameter der einzelnen Gaussians
- So können wir herausfinden, welche Daten zu welchem Cluster gehören (bzw. welcher Gaussian zu welchen Daten geführt hat)

DER EM-ALGORITHMUS

DER EM-ALGORITHMUS

- Der Expectation-Maximization-Algorithmus lässt sich sehr vielseitig anwenden (hauptsächlich Mischverteilungsmodelle wie z.B. GMMs)

DER EM-ALGORITHMUS

- Der Expectation-Maximization-Algorithmus lässt sich sehr vielseitig anwenden (hauptsächlich Mischverteilungsmodelle wie z.B. GMMs)
- Grundsätzlich sind uns einige Werte gegeben, andere sind uns jedoch unbekannt

DER EM-ALGORITHMUS

- Der Expectation-Maximization-Algorithmus lässt sich sehr vielseitig anwenden (hauptsächlich Mischverteilungsmodelle wie z.B. GMMs)
- Grundsätzlich sind uns einige Werte gegeben, andere sind uns jedoch unbekannt
- Uns fehlen z.B. sowohl die Parameter einer Verteilung als auch einige "versteckte Werte"

DER EM-ALGORITHMUS

- Der Expectation-Maximization-Algorithmus lässt sich sehr vielseitig anwenden (hauptsächlich Mischverteilungsmodelle wie z.B. GMMs)
- Grundsätzlich sind uns einige Werte gegeben, andere sind uns jedoch unbekannt
- Uns fehlen z.B. sowohl die Parameter einer Verteilung als auch einige "versteckte Werte"

Wir stecken in einer Zwickmühle:

- Die Parameter können nicht berechnet werden (uns fehlen die versteckten Werte)
- Die versteckten Werte können nicht berechnet werden (uns fehlen die Parameter)

DER EM-ALGORITHMUS

- Für diese Problematik existiert der EM-Algorithmus

DER EM-ALGORITHMUS

- Für diese Problematik existiert der EM-Algorithmus
- Der EM-Algorithmus besteht im Grunde aus zwei Schritten, die iterativ durchgeführt werden, bis die gesuchten Parameter konvergieren

DER EM-ALGORITHMUS

- Für diese Problematik existiert der EM-Algorithmus
- Der EM-Algorithmus besteht im Grunde aus zwei Schritten, die iterativ durchgeführt werden, bis die gesuchten Parameter konvergieren
- Expectation step und Maximization step

EXPECTATION UND MAXIMIZATION STEP

Expectation step:

- Schätze die versteckten Werte anhand der gegebenen Daten

EXPECTATION UND MAXIMIZATION STEP

Expectation step:

- Schätze die versteckten Werte anhand der gegebenen Daten

Maximization step:

- Es wird angenommen, die Werte aus dem ersten Schritt seien die wirklichen Werte
- Wir wenden ML (Maximum-Likelihood) auf diese Werte an, um die eigentlich gesuchten Parameter zu berechnen

EXPECTATION UND MAXIMIZATION STEP

Expectation step:

- Schätze die versteckten Werte anhand der gegebenen Daten

Maximization step:

- Es wird angenommen, die Werte aus dem ersten Schritt seien die wirklichen Werte
- Wir wenden ML (Maximum-Likelihood) auf diese Werte an, um die eigentlich gesuchten Parameter zu berechnen

Zu Beginn des Algorithmus werden die Parameter mit Zufallswerten initialisiert.

EM FÜR GMMS

- Nun wollen wir den EM-Algorithmus auf unsere GMMs anwenden

EM FÜR GMMS

- Nun wollen wir den EM-Algorithmus auf unsere GMMs anwenden
- Die "versteckten Werte" beziehen sich in diesem Fall darauf, welche Daten zu welchem Cluster gehören bzw. welcher Gaussian welche Daten erzeugt hat

EM FÜR GMMS

- Nun wollen wir den EM-Algorithmus auf unsere GMMs anwenden
- Die "versteckten Werte" beziehen sich in diesem Fall darauf, welche Daten zu welchem Cluster gehören bzw. welcher Gaussian welche Daten erzeugt hat
- Außerdem fehlen uns die Parameter der Verteilung

EM FÜR GMMS

- Nun wollen wir den EM-Algorithmus auf unsere GMMs anwenden
- Die "versteckten Werte" beziehen sich in diesem Fall darauf, welche Daten zu welchem Cluster gehören bzw. welcher Gaussian welche Daten erzeugt hat
- Außerdem fehlen uns die Parameter der Verteilung
- Ohne den EM-Algorithmus kämen wir nicht weiter:
  - Wenn wir die Parameter μ_i, Σ_i und w_i kennen würden, könnten wir herausfinden welche Daten zu welchem Gaussian gehören
  - Wenn wir wüssten welche Daten zu welchem Cluster gehören, könnten wir die Parameter relativ leicht berechnen

EM FÜR GMMS

- Nun wollen wir den EM-Algorithmus auf unsere GMMs anwenden
- Die "versteckten Werte" beziehen sich in diesem Fall darauf, welche Daten zu welchem Cluster gehören bzw. welcher Gaussian welche Daten erzeugt hat
- Außerdem fehlen uns die Parameter der Verteilung
- Ohne den EM-Algorithmus kämen wir nicht weiter:
  - Wenn wir die Parameter μ_i, Σ_i und w_i kennen würden, könnten wir herausfinden welche Daten zu welchem Gaussian gehören
  - Wenn wir wüssten welche Daten zu welchem Cluster gehören, könnten wir die Parameter relativ leicht berechnen
- Wir bräuchten eine der Antworten, um auf die andere Antwort zu kommen

EM FÜR GMMS

- Anstatt die Probleme einzeln zu betrachten, lösen wir beide mithilfe des EM-Algorithmus schrittweise parallel

EM FÜR GMMS

- Anstatt die Probleme einzeln zu betrachten, lösen wir beide mithilfe des EM-Algorithmus schrittweise parallel
- Vor dem ersten Schritt initialisieren wir die Parameter μ_i, Σ_i und w_i mit zufälligen Werten

EM FÜR GMMS

- Anstatt die Probleme einzeln zu betrachten, lösen wir beide mithilfe des EM-Algorithmus schrittweise parallel
- Vor dem ersten Schritt initialisieren wir die Parameter μ_i, Σ_i und w_i mit zufälligen Werten
- Es spielt keine Rolle, wie die Werte gewählt werden - der EM-Algorithmus konvergiert immer

EM FÜR GMMS

- Anstatt die Probleme einzeln zu betrachten, lösen wir beide mithilfe des EM-Algorithmus schrittweise parallel
- Vor dem ersten Schritt initialisieren wir die Parameter μ_i, Σ_i und w_i mit zufälligen Werten
- Es spielt keine Rolle, wie die Werte gewählt werden - der EM-Algorithmus konvergiert immer
- Er findet jedoch nur lokale Maximalwerte

EM FÜR GMMS - EXPECTATION STEP

- Wir initialisieren die Parameter μ_i, Σ_i und w_i mit Zufallswerten

EM FÜR GMMS - EXPECTATION STEP

- Wir initialisieren die Parameter μ_i, Σ_i und w_i mit Zufallswerten
- Wir nehmen an, unsere geratenen Werte wären die wirklichen Werte

EM FÜR GMMS - EXPECTATION STEP

- Wir initialisieren die Parameter μ_i, Σ_i und w_i mit Zufallswerten
- Wir nehmen an, unsere geratenen Werte wären die wirklichen Werte
- Zwischen jeder Stichprobe x_j und jedem Gaussian berechnen wir einen "assignment score":

$$\gamma(z_{jk}) = \frac{w_k \cdot \mathcal{N}(x_j|\mu_k, \Sigma_k)}{\sum_{i=1}^{n} w_i \cdot \mathcal{N}(x_j|\mu_i, \Sigma_i)}$$

EM FÜR GMMS - MAXIMIZATION STEP

- Nun werden die Parameter anhand der assignment scores neu berechnet (J: Anzahl der Stichproben):

$$\mu_k = \frac{1}{J_k} \cdot \sum_{j=1}^{J} \gamma(z_{jk}) \cdot x_j$$

$$\Sigma_k = \frac{1}{N_k} \cdot \sum_{j=1}^{J} \gamma(z_{jk}) \cdot (x_j - \mu_k) \cdot (x_j - \mu_k)^T$$

$$w_k = \frac{J_k}{J}$$

$$(\text{mit } J_k = \sum_{j=1}^{J} \gamma(z_{jk}))$$

WIEDERHOLEN BIS KONVERGENZ EINTRITT

- Die Parameter sind jetzt schon besser geschätzt

WIEDERHOLEN BIS KONVERGENZ EINTRITT

- Die Parameter sind jetzt schon besser geschätzt
- Im Folgenden werden die assignment scores mithilfe der neu geschätzten Parameter erneut berechnet

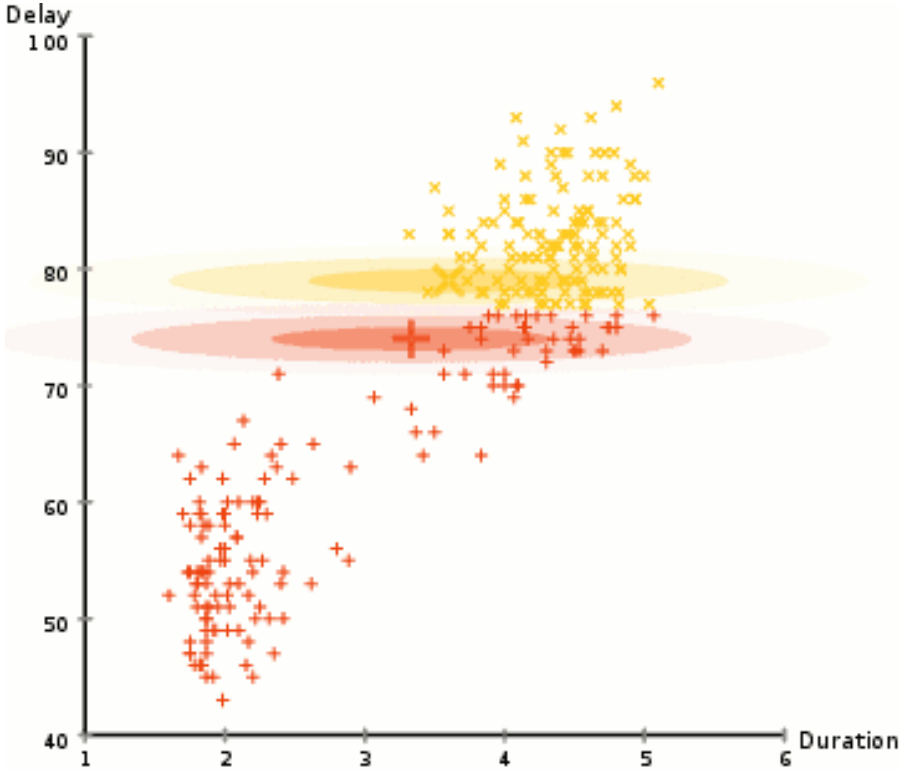
WIEDERHOLEN BIS KONVERGENZ EINTRITT

- Die Parameter sind jetzt schon besser geschätzt
- Im Folgenden werden die assignment scores mithilfe der neu geschätzten Parameter erneut berechnet
- Expectation und Maximization step werden nun wiederholt bis die Werte der Parameter konvergieren

WIEDERHOLEN BIS KONVERGENZ EINTRITT

- Die Parameter sind jetzt schon besser geschätzt
- Im Folgenden werden die assignment scores mithilfe der neu geschätzten Parameter erneut berechnet
- Expectation und Maximization step werden nun wiederholt bis die Werte der Parameter konvergieren
- Ergebnis: optimierte Parameter μ, Σ und w (lokale Maximalwerte)

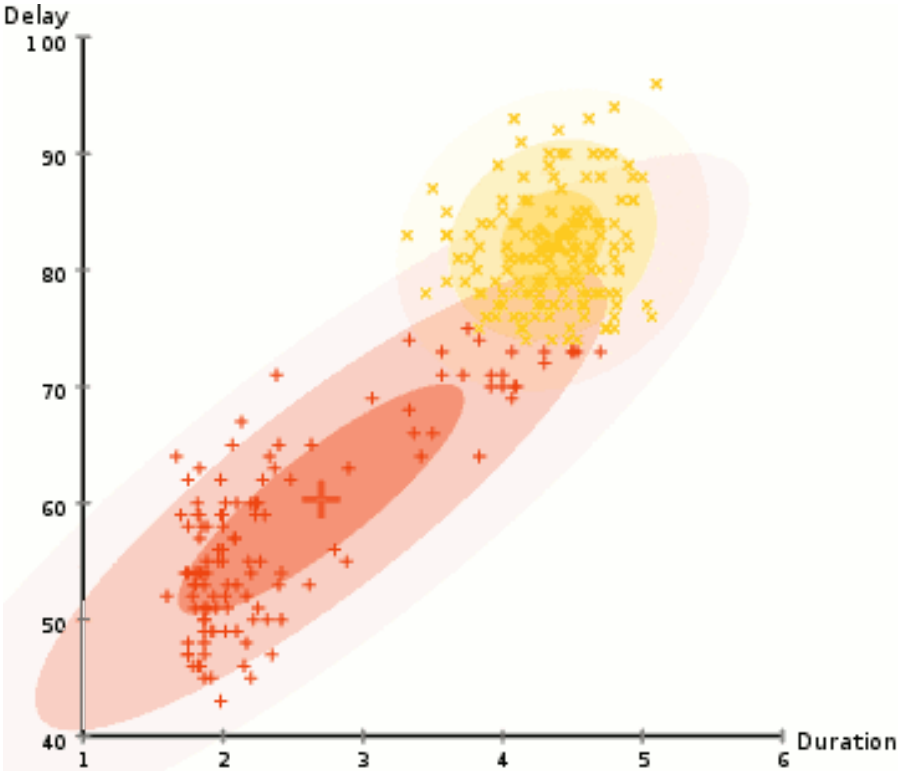
BEISPIELPLOT - ERSTE ITERATION



Aus Wikipedia: Expectation-maximization algorithm

- Parameter wurden mit Zufallswerten initialisiert

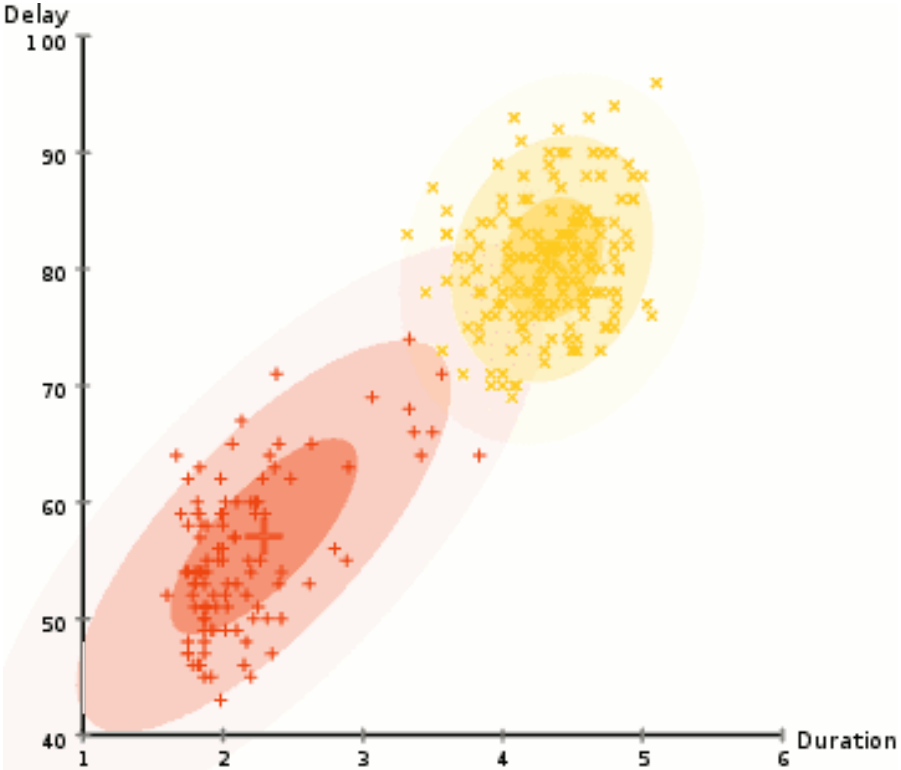
BEISPIELPLOT - ZWEITE ITERATION



Aus Wikipedia: Expectation-maximization algorithm

- Nach nur einem Schritt eine relativ gute (grobe) Schätzung

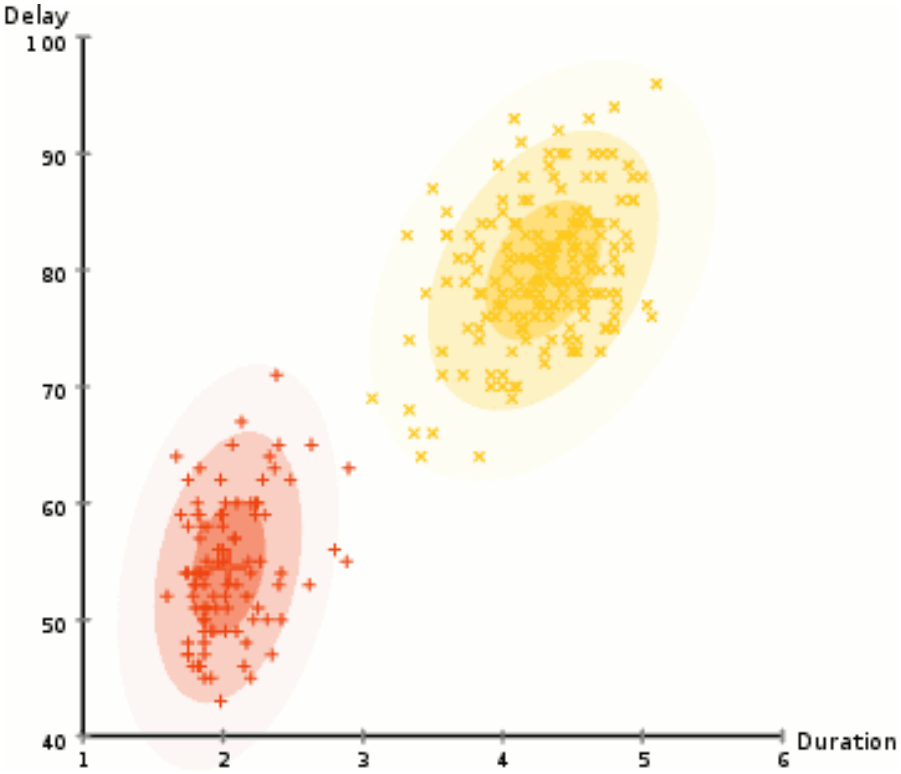
BEISPIELPLOT - FÜNFTE ITERATION



Aus Wikipedia: Expectation-maximization algorithm

- Einige Schritte weiter und die Cluster sind schon recht gut angepasst

BEISPIELPLOT - 15. ITERATION



Aus Wikipedia: Expectation-maximization algorithm

- Nach 15 Schritten erreichen wir ein nahezu perfektes Ergebnis

ANWENDUNG / AUSWERTUNG

ZURÜCK ZUR VIDEOÜBERWACHUNG

Wie können wir das Prinzip der GMMs auf die Videoüberwachung anwenden?

- Jedes Pixel wird als ein eigenes GMM modelliert

ZURÜCK ZUR VIDEOÜBERWACHUNG

Wie können wir das Prinzip der GMMs auf die Videoüberwachung anwenden?

- Jedes Pixel wird als ein eigenes GMM modelliert
- Wir versuchen jeweils zu entscheiden, ob es sich um ein Vorder- oder Hintergrundpixel handelt

ZURÜCK ZUR VIDEOÜBERWACHUNG

Wie können wir das Prinzip der GMMs auf die Videoüberwachung anwenden?

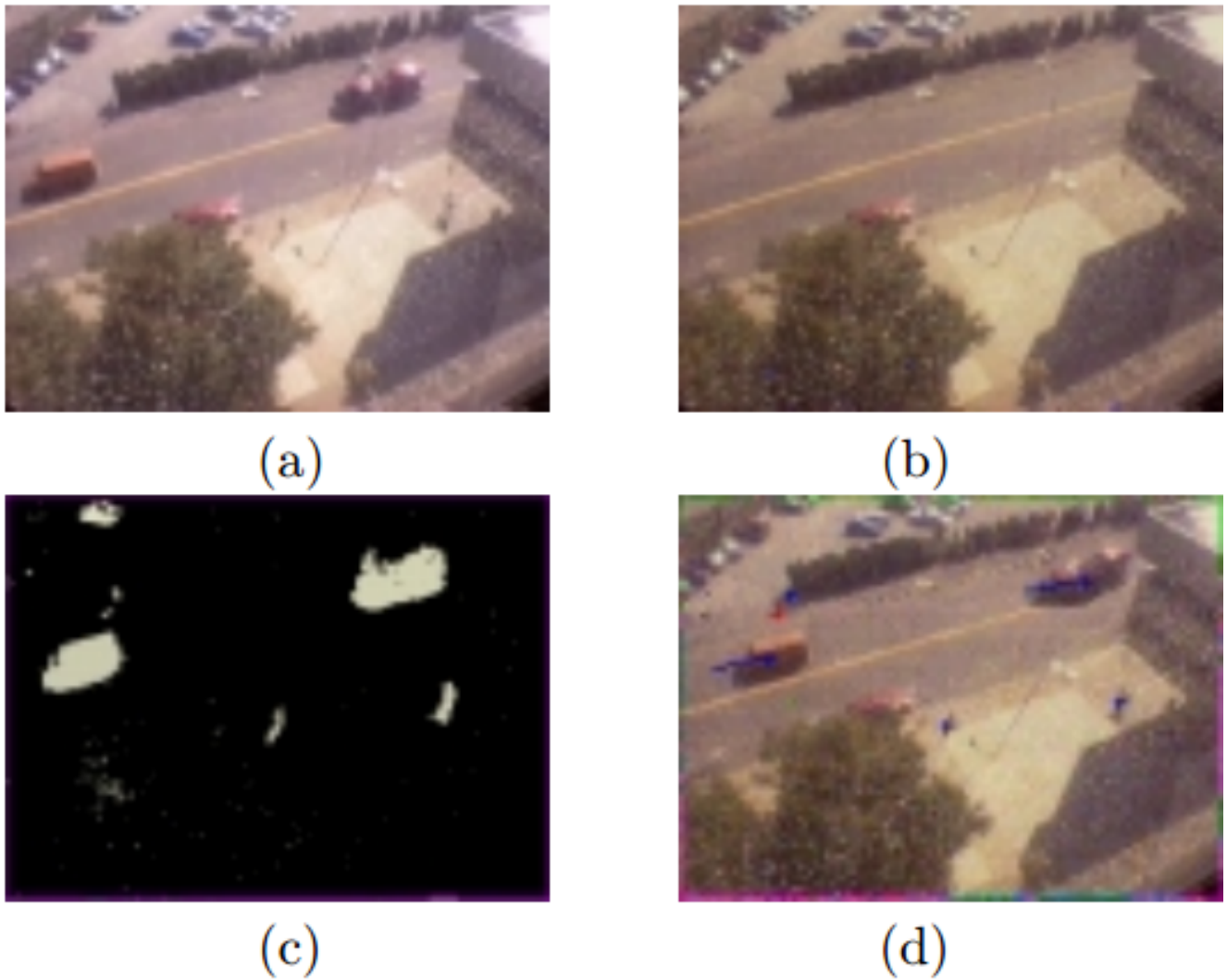
- Jedes Pixel wird als ein eigenes GMM modelliert
- Wir versuchen jeweils zu entscheiden, ob es sich um ein Vorder- oder Hintergrundpixel handelt
- Pixel, die in keine der Verteilungen passen, werden vorerst als Vordergrundpixel eingestuft

ZURÜCK ZUR VIDEOÜBERWACHUNG

Wie können wir das Prinzip der GMMs auf die Videoüberwachung anwenden?

- Jedes Pixel wird als ein eigenes GMM modelliert
- Wir versuchen jeweils zu entscheiden, ob es sich um ein Vorder- oder Hintergrundpixel handelt
- Pixel, die in keine der Verteilungen passen, werden vorerst als Vordergrundpixel eingestuft
- Es werden zu jeder Zeit mehrere Werte der Pixel gespeichert

WAS DAS PROGRAMM SIEHT



Aus “Adaptive background mixture models for real-time tracking” (Stauffer)

PIXELVERLAUF

Jedes Pixel wird über einen Zeitraum hinweg betrachtet, wir kennen also den "Verlauf" jedes Pixels:

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

Wir kennen zu jeder Zeit den Zustand jedes Pixels in den letzten t Bildern.

PIXELVERLAUF

Jedes Pixel wird über einen Zeitraum hinweg betrachtet, wir kennen also den "Verlauf" jedes Pixels:

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

Wir kennen zu jeder Zeit den Zustand jedes Pixels in den letzten t Bildern.

Nun stellen wir ein GMM auf, welches unseren Ansprüchen entspricht.

EIN GMM FÜR JEDES PIXEL

$$P(X_t) = \sum_{i=1}^{n} w_{i,t} \cdot \mathcal{N}(X_t, \mu_{i,t}, \Sigma_{i,t})$$

n: Anzahl der Verteilungen (je nach Rechenleistung)

$w_{i,t}$: Geichtung des i-ten Gaussians zur Zeit t

$\mu_{i,t}$: Mittelwert des i-ten Gaussians zur Zeit t

$\Sigma_{i,t}$: Kovarianzmatrix des i-ten Gaussians zur Zeit t

WÄHREND DER LAUFZEIT

- Jeder neu beobachtete Pixelwert X_t wird gegen die n vorhandenen Gaussians geprüft, bis eine Übereinstimmung gefunden wird

- Übereinstimmung: Wert innerhalb von 2,5 Standardabweichungen zum Mittelwert

WÄHREND DER LAUFZEIT

- Jeder neu beobachtete Pixelwert X_t wird gegen die n vorhandenen Gaussians geprüft, bis eine Übereinstimmung gefunden wird
- Übereinstimmung: Wert innerhalb von 2,5 Standardabweichungen zum Mittelwert
- Wenn eine Übereinstimmung gefunden wird, wird das Pixel (immer noch) als Hintergrund betrachtet. Außerdem wird
  - sein Mittelwert μ näher an X_t angepasst
  - seine Varianz σ^2 verringert
  - sein Gewicht w erhöht

WÄHREND DER LAUFZEIT

- Falls keine Übereinstimmung gefunden wird, wird das entsprechende Pixel als Vordergrund markiert.

WÄHREND DER LAUFZEIT

- Falls keine Übereinstimmung gefunden wird, wird das entsprechende Pixel als Vordergrund markiert.

- Der unwahrscheinlichste Gaussian im GMM wird dann folgendermaßen verändert:
  - Mittelwert $\mu = X_t$
  - Varianz σ^2: ein hoher Wert
  - Gewicht w: ein niedriger Wert

ERKENNUNG DES HINTERGRUNDES

- Die Vordergrundpixel erkennen wir, indem wir die neuen Pixel mit den Werten aller Gaussians vergleichen

Aber wie finden wir heraus, welche Pixel zum Hintergrund gehören?

ERKENNUNG DES HINTERGRUNDES

- Die Vordergrundpixel erkennen wir, indem wir die neuen Pixel mit den Werten aller Gaussians vergleichen

Aber wie finden wir heraus, welche Pixel zum Hintergrund gehören?

- Wir suchen Pixel / Verteilungen mit einem hohen Gewicht und einer niedrigen Varianz

ERKENNUNG DES HINTERGRUNDES

- Die Vordergrundpixel erkennen wir, indem wir die neuen Pixel mit den Werten aller Gaussians vergleichen

Aber wie finden wir heraus, welche Pixel zum Hintergrund gehören?

- Wir suchen Pixel / Verteilungen mit einem hohen Gewicht und einer niedrigen Varianz

- Ansatz: Wir sortieren die Verteilungen jedes Pixels nach ihrem Wert für w/σ

ERKENNUNG DES HINTERGRUNDES

- Die Vordergrundpixel erkennen wir, indem wir die neuen Pixel mit den Werten aller Gaussians vergleichen

Aber wie finden wir heraus, welche Pixel zum Hintergrund gehören?

- Wir suchen Pixel / Verteilungen mit einem hohen Gewicht und einer niedrigen Varianz
- Ansatz: Wir sortieren die Verteilungen jedes Pixels nach ihrem Wert für w/σ
- Die Verteilungen die höher auf dieser Liste sind gehören wahrscheinlicher zum Hintergrund des Bildes

EM-ALGORITHMUS?

- Der EM-Algorithmus wird hier nicht verwendet

EM-ALGORITHMUS?

- Der EM-Algorithmus wird hier nicht verwendet
- Dieser müsste in jedem Frame auf jedes einzelne Pixel angewendet werden

EM-ALGORITHMUS?

- Der EM-Algorithmus wird hier nicht verwendet
- Dieser müsste in jedem Frame auf jedes einzelne Pixel angewendet werden
- Zur Zeit des Papers (1999) zu resourcenlastig, heute allerdings durchaus möglich

EM-ALGORITHMUS?

- Der EM-Algorithmus wird hier nicht verwendet
- Dieser müsste in jedem Frame auf jedes einzelne Pixel angewendet werden
- Zur Zeit des Papers (1999) zu resourcenlastig, heute allerdings durchaus möglich
- Stattdessen wird eine "on-line K-means approximation" verwendet (wird hier nicht weiter betrachtet)

AUSWERTUNG

- Da wir zu jeder Zeit mehrere Modelle des Hintergrundes beachten, ist das System robust gegenüber Änderungen des Hintergrundes

AUSWERTUNG

- Da wir zu jeder Zeit mehrere Modelle des Hintergrundes beachten, ist das System robust gegenüber Änderungen des Hintergrundes

- Wenn ein neuer Gegenstand in den Hintergrund tritt und dort für mehrere Bilder verharrt, wird er sehr schnell in eines der Modelle übernommen

AUSWERTUNG

- Da wir zu jeder Zeit mehrere Modelle des Hintergrundes beachten, ist das System robust gegenüber Änderungen des Hintergrundes
- Wenn ein neuer Gegenstand in den Hintergrund tritt und dort für mehrere Bilder verharrt, wird er sehr schnell in eines der Modelle übernommen
- Auch Änderungen der Lichtverhätnisse oder Wetterlage sind kein Problem

AUSWERTUNG

- Auf jeden Fall ein guter Ansatz, der laut Autor des Papers auch erfolgreich funktioniert

AUSWERTUNG

- Auf jeden Fall ein guter Ansatz, der laut Autor des Papers auch erfolgreich funktioniert
- 16 Monate lang mit einer Auflösung von 120x160 Pixeln bei 11-13 Bildern pro Sekunde getestet

AUSWERTUNG

- Auf jeden Fall ein guter Ansatz, der laut Autor des Papers auch erfolgreich funktioniert
- 16 Monate lang mit einer Auflösung von 120x160 Pixeln bei 11-13 Bildern pro Sekunde getestet
- Probleme nur bei plötzlicher Änderung der Wetterlage, doch selbst dann erholt sich das System recht schnell wieder

AUSWERTUNG

- Auf jeden Fall ein guter Ansatz, der laut Autor des Papers auch erfolgreich funktioniert
- 16 Monate lang mit einer Auflösung von 120x160 Pixeln bei 11-13 Bildern pro Sekunde getestet
- Probleme nur bei plötzlicher Änderung der Wetterlage, doch selbst dann erholt sich das System recht schnell wieder
- Heuzutage durch erhöhte Rechenleistung besser umzusetzen (höhere Auflösung, mehr Bilder pro Sekunde, EM-Algorithmus)

Danke für die Aufmerksamkeit