6. Anwendungen

- Texterkennung in Videos -

Videoanalyse

Stephan Kopf

Übersicht

- Motivation
- Texterkennung in Videos
  1. Erkennung von Textregionen/Textzeilen
  2. Segmentierung einzelner Buchstaben
  3. Auswahl der Buchstabenpixel
  4. Erkennung einzelner Buchstaben (OCR)
     - Pattern matching
     - Zoning
     - Shape contexts
     - Konturprofile
     - Skelette
     - Skalenraumabbildungen
- Zusammenfassung

Motivation

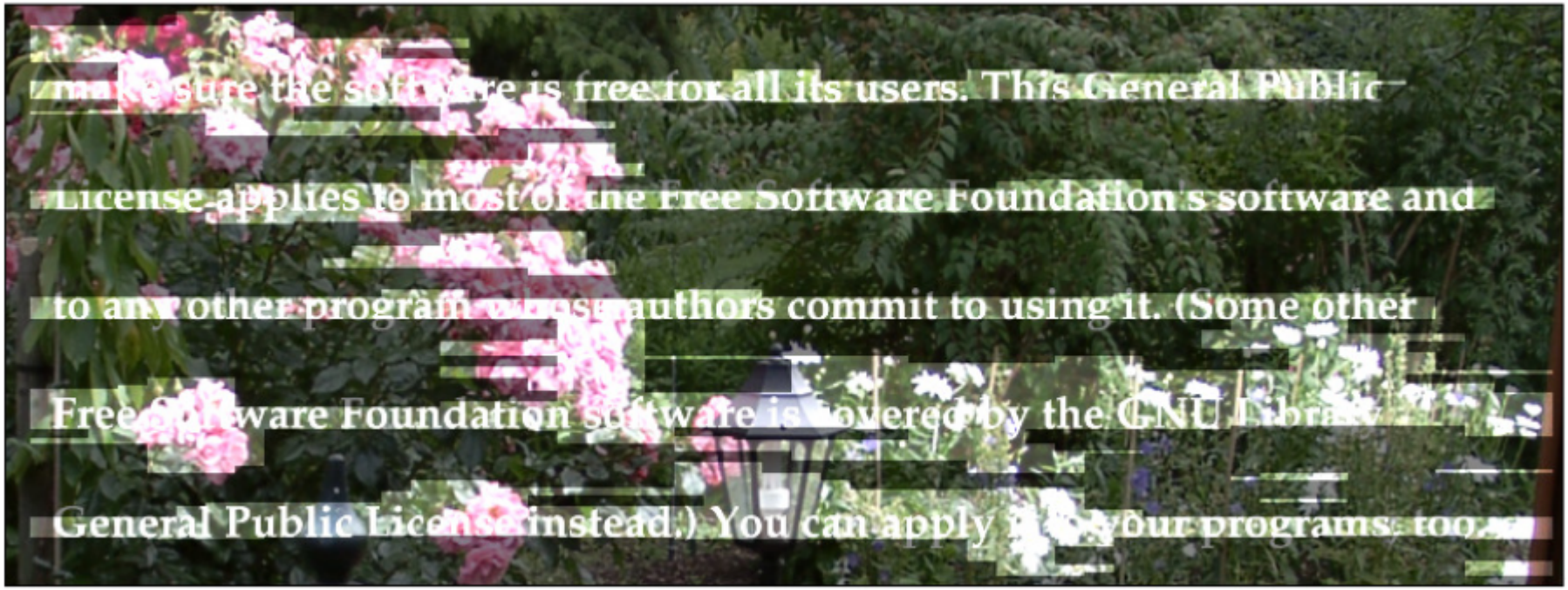
- Wichtige semantische Informationen werden in Videos durch Texte übermittelt:
  - Namen der Schauspieler in Spielfilmen
  - In Nachrichtensendungen werden die neben dem Sprecher gezeigten Bilder durch einen Text beschrieben.
  - Ort oder Zeit in einem Spielfilm
  - Fragen bei Quizshows
  - Namen/Beruf der Teilnehmer einer Diskussionsrunde
  - Titel eines Films

Erkennung von Textregionen (I)

Ablauf

2. Suche Blöcke mit starken Kanten
3. Fasse benachbarte Blöcke zu Textregionen zusammen
4. Verwende horizontale Projektionsprofile zur Erkennung einzelner Textzeilen

Suche Blöcke mit starken Kanten (Summe Kantenstärke pro Block > T)



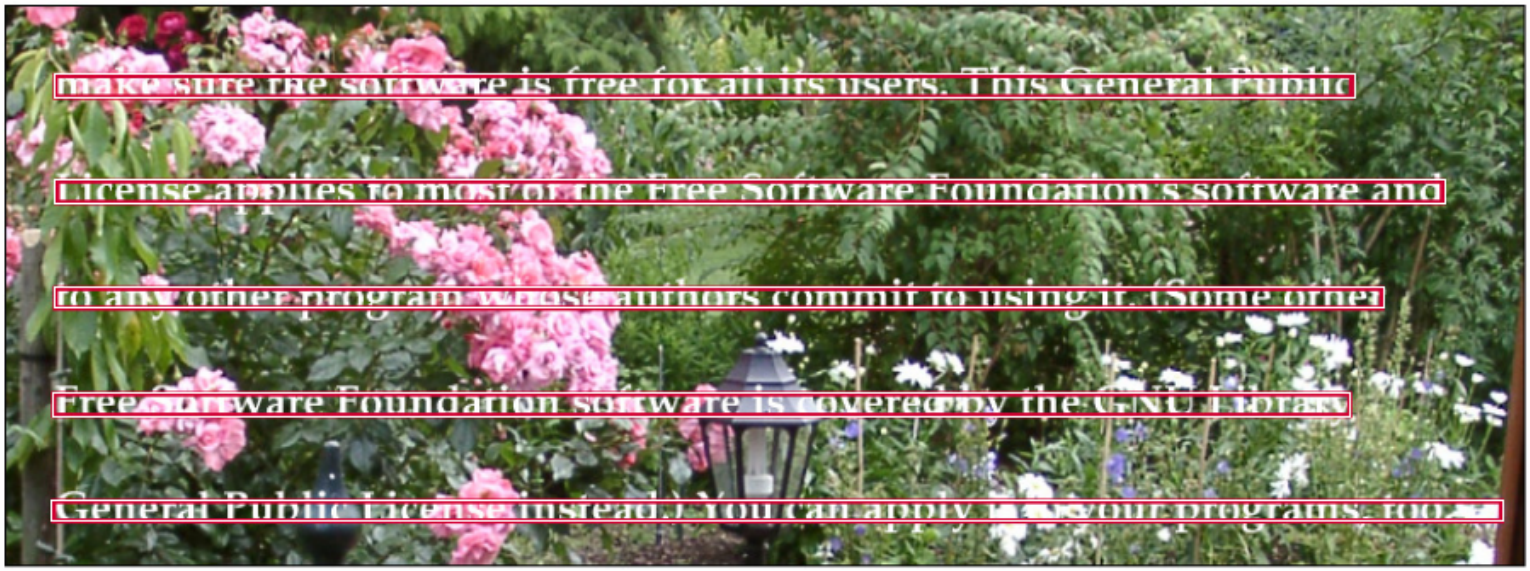
Erkennung von Textregionen (II)

Zusammenfassung von Blöcken

Erkennung von Textregionen (III)

Horizontale Projektionsprofile

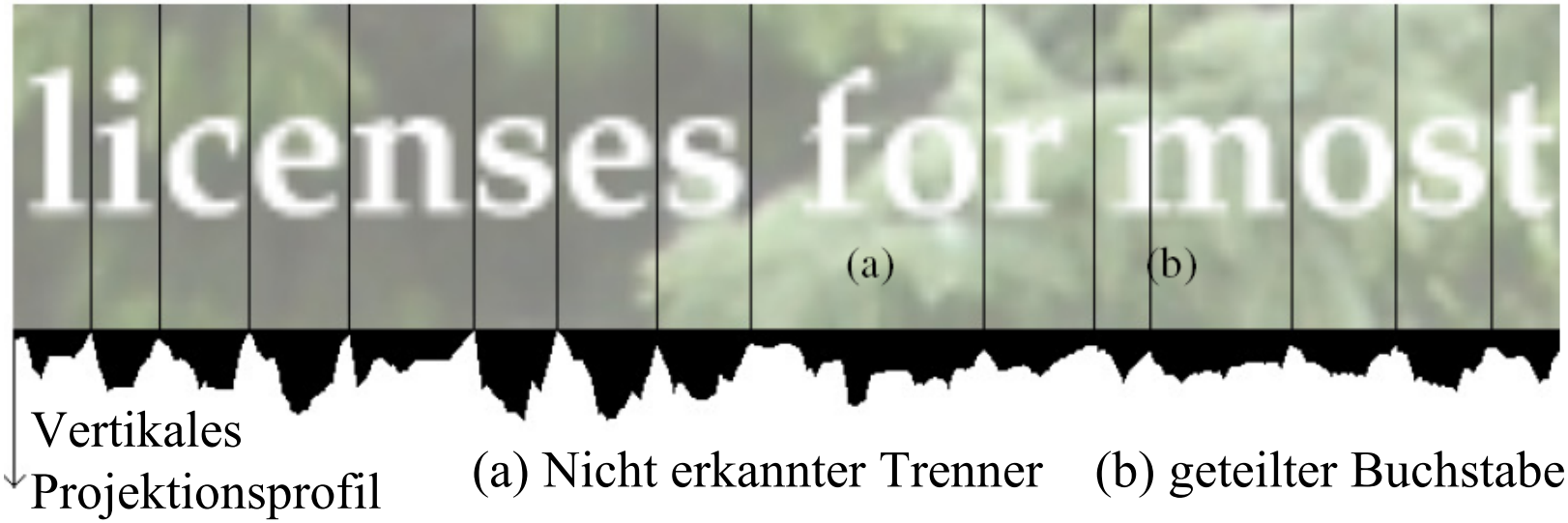
Erkannte Textzeilen



Segmentierung von Buchstaben (I)

Erster Ansatz

Analysiere vertikale Projektionsprofile um *Trenner* zwischen Buchstaben zu erkennen.



(a) Nicht erkannter Trenner (b) geteilter Buchstabe

Problem: geteilte oder verbundene Buchstaben

Segmentierung von Buchstaben (II)

Neuer Ansatz

- Analysiere *kürzeste Pfade* und nutze diese als Trenner zwischen den Buchstaben.

- Suche für jede Spalte von der obersten Zeile einen kürzesten Pfad bis zur untersten Zeile.
- Die Kosten für den Pfad sollen minimal sein.
- Die Kosten sind als Summe der absoluten Differenzen zwischen benachbarten Pfadpixeln definiert.

Segmentierung von Buchstaben (III)

Kürzester-Pfade-Algorithmus

- Der *Kürzeste-Pfade-Algorithmus* von *Dijkstra* wird verwendet, um den Pfad mit den geringsten Kosten zu finden.
- Ziel: Berechnung eines kürzesten Pfades zwischen einem Startknoten (Pixel in oberster Zeile) und einem beliebigen Knoten in einem kantengewichteten Graphen (beliebiges Pixel in unterster Textzeile).
- Initialisiere: Setze Entfernung für alle Knoten auf unendlich.
- Betrachte Knoten *u* mit geringstem Abstand zum Startknoten
  - Falls Knoten u in unterster Zeile liegt: kürzester Pfad gefunden.
  - Sonst:
    - Betrachte alle erreichbaren Pixel, d.h. die drei benachbarten (in der Zeile darunter liegenden) Pixel.
    - Prüfe, ob der Weg zu jedem Pixel über das aktuelle Pixel günstiger ist als der bisher bekannte Weg zu diesem Pixel.
    - Setze neuen Pfad, falls dieser günstiger ist.

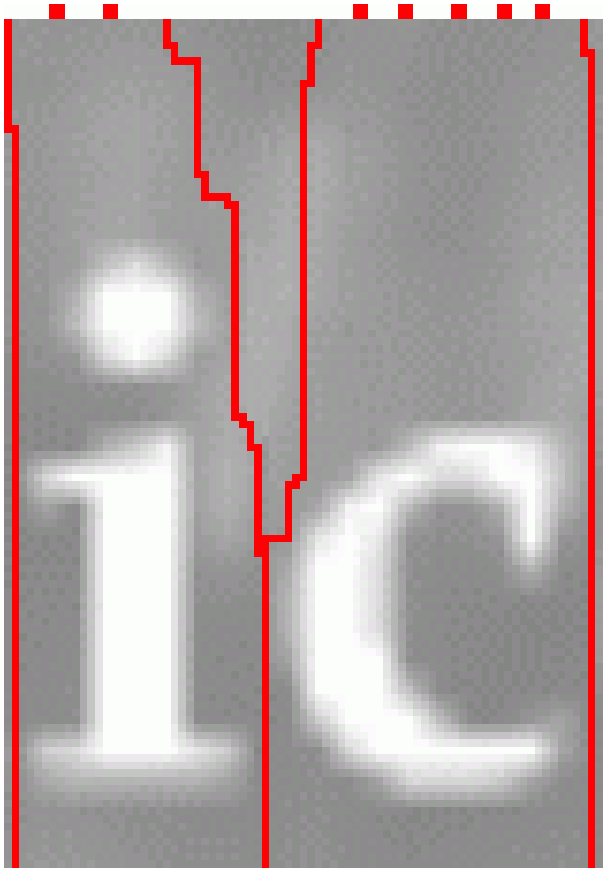
Optimierung des Kürzesten-Pfade-Algorithmus

Problem

- Der Aufwand zur Berechnung des kürzesten Pfads für jedes Pixel ist sehr aufwändig.

Optimierung

4. Initialisiere mögliche Startpixel.
5. Markiere linkes und rechtes Pixel.
6. Berechne Pfad für markierte Pixel.
7. Entferne Startpixel zwischen zwei Pfaden, falls diese zusammen laufen.
8. Wähle nächstes Startpixel.
9. Wiederhole mit 3.



Auswahl der Buchstabenpixel (I)

Identifiziere Buchstabenpixel mit Hilfe eines **modifizierten Region-Merging-Algorithmus**

- Berechne das Histogramm einer Textregion und identifiziere ein bis zwei dominante Farben.
  Annahme: Eine dieser Farben ist die Textfarbe.
- Identifiziere Regionen mit einem Region-Growing-Algorithmus.
- Jede Region kann einen der drei Zustände annehmen: *Text, Hintergrund, undefiniert.* Alle Regionen sind zunächst *undefiniert*.
- Setze alle Regionen mit Textfarbe zu *Textregionen*.
- Undefinierte Regionen am oberen oder unteren Rand der Textzeile werden als *Hintergrund* definiert.

Auswahl der Buchstabenpixel (II)

6. Berechne Distanz $\boldsymbol{D_{i,j}}$ zwischen einer undefinierten Region ***i*** und einer bekannten Region ***j*** (Text oder Hintergrund) anhand der Farben $\boldsymbol{C_i}$ und dem Schwerpunkt einer Region $\boldsymbol{G_i}$:

$$D_{i,j} = |C_i - C_j| + |G_i - G_j|.$$

7. Wähle minimale Distanz $\boldsymbol{D_{i,j}}$ und definiere Region als *Text* oder *Hintergrund*.
8. Wiederhole mit Schritt 6 bis alle undefinierten Regionen bekannt sind.

Erkennung einzelner Buchstaben (I)

Erkennung einzelner Buchstaben (OCR)

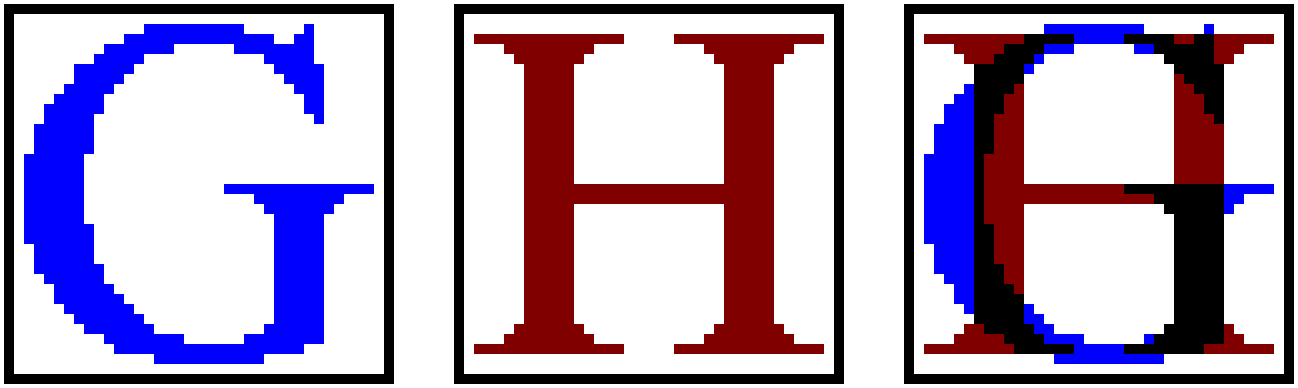
- Pattern matching
- Zoning
- Shape contexts
- Konturprofile
- Skelette
- Skalenraumabbildungen

Erkennung einzelner Buchstaben (II)

Pattern matching

Berechne die Differenz zwischen zwei Binärbildern:

$$D_{Q,J} = \frac{1}{n_x \cdot n_y} \cdot \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \begin{cases} 0 & \text{falls} \quad Q_{x,y} = J_{x,y}, \\ 1 & \text{sonst.} \end{cases}$$

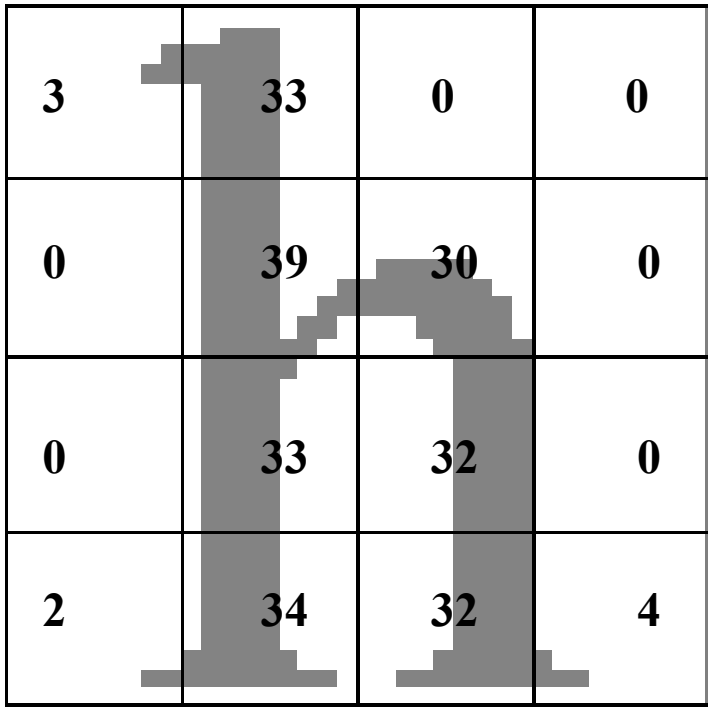


18 % der Pixel unterscheiden sich

Erkennung einzelner Buchstaben (III)

Zoning

- Definiere ein Gitter mit N x M Zellen.
- Zähle die Anzahl der Textpixel in jedes Zelle.
- Vergleiche zwei Vektoren, die durch die Anzahl der Textpixel jeder Zelle definiert sind.

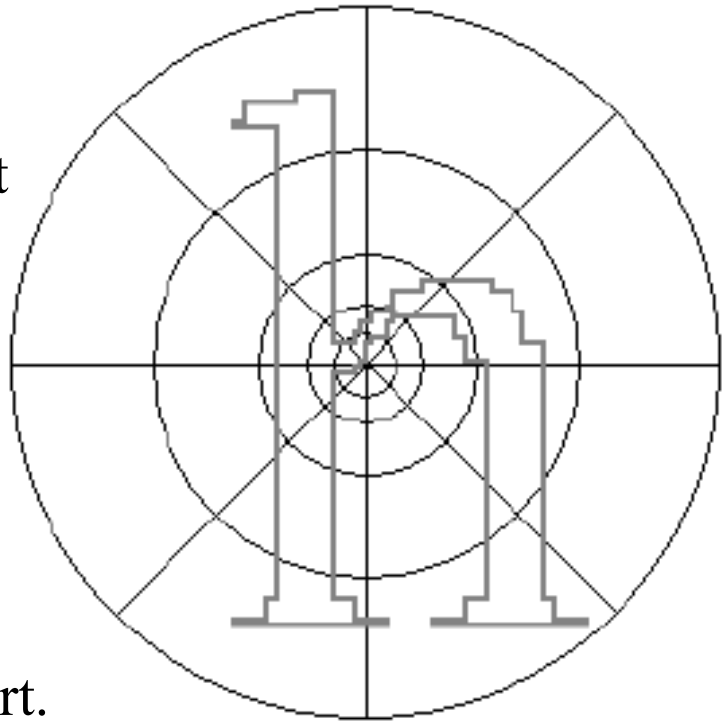


Vektor: (3,33,0,0,0,39,30,0,0,33,32,0,2,34,32,4)

Erkennung einzelner Buchstaben (IV)

Shape contexts

- Der Shape-Context-Algorithmus ist ein spezieller Zoning-Algorithmus.
- Ein rundes Raster wird zur Definition der Zellen verwendet.
- Ein Konturpixel definiert den Mittelpunkt des Rasters.
- Die Anzahl der ***Konturpixel*** (nicht der Textpixel) in jeder Zelle definieren den Merkmalsvektor.
- Als Referenzbuchstaben werden Merkmalsvektoren für ***jedes*** einzelne Konturpixel eines Buchstabens gespeichert.



Erkennung einzelner Buchstaben (V)

Konturprofile (contour profiles)

- Horizontales Profil: Analysiere obere und untere Konturpixel.
- Vertikales Profil : Analysiere linke und rechte Konturpixel.
- Aggregiere diese Profile in einen Vektor.