



# Automatische Analyse von 2D-Barcodes

Institut für Informatik  
Mustererkennung und Bioinformatik

Angewandte Bildverarbeitung, SS 2007



# Überblick

## 2D-Barcodes

### Überblick

- 1 Grundideen von 2D-Barcodes
- 2 Lokalisierung im Bild
- 3 Code-Segmentierung
- 4 Ausblick

# 2D-Barcodes



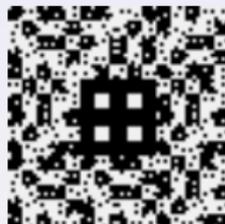
## 2D-Barcodes

### 2D-Barcodes

#### Lokalisierung

#### Segmentier.

#### Ausblick





- 2D-Barcodes sind Erweiterungen von 1D-Codes
  - Kodierung von mehr Informationen (mehrere tausend Zeichen)
  - Größe skaliert mit Informationsgehalt ⇒ kleine Fläche, zu klein für 1D-Code
  - integrierte Fehlerdetektion/-korrektur für hohe Robustheit



- es gibt zahllose Codes, die meisten sind patentiert
 

<ul style="list-style-type: none"> <li>● ArrayCode</li> <li>● AztecCode</li> <li>● Maxi-/Minicode</li> <li>● Ultracode</li> </ul>	<ul style="list-style-type: none"> <li>● DataStrip Code</li> <li>● DataMatrix</li> <li>● PDF 417</li> <li>● ...</li> </ul>
---	--



- 3 große Klassen von Codes:
  - *Multi-Row-Codes* –  
Kombination von 1D-Codes
  - Codes mit Lokalisierungsmuster –  
enthalten ein fest definiertes Symbol
  - Codes ohne Lokalisierungsmuster –  
Lokalisierung nur anhand interner Struktur
- Detektion und Dekodierung damit grundsätzlich  
codeabhängig, aber es gibt Überschneidungen



# Barcode-Dekodierung: Systemaufbau

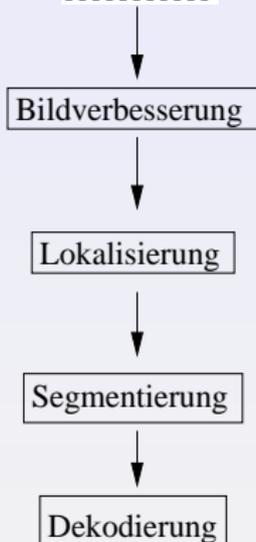
## 2D-Barcodes

## 2D-Barcodes

### Lokalisierung

### Segmentier.

### Ausblick



- Bildverbesserung:  
Ausgleich von Unschärfen, Lichtartefakten
- Lokalisierung:  
Detektion der Coderegion im Bild  
(muss robust sein gegen Rauschen, Verzerrungen, Rotationen. . .)
- Segmentierung:  
Detektion der einzelnen *Module* und ihrer Intensitäts- bzw. Farbwerte
- Dekodierung:  
Auslesen der enthaltenen Information



## 2D-Barcodes

### 2D-Barcodes

#### Lokalisierung

#### Segmentier.

#### Ausblick

- Lokalisierung und Segmentierung können verzahnt sein
- bei Schwarz-Weiss-Codes kommt Binarisierung hinzu
- Algorithmik muss mit Problemen umgehen können:
  - variierende Größe und Auflösung
  - perspektivische Verzerrungen
- Ziel ist zumeist hohe Geschwindigkeit (trotz schlechter Bildqualität!)
  - ⇒ aktuelle Systeme schaffen  $\approx 15$  Bilder/sec



- Ziel:  
Finden der Coderegion (z.B. Bounding Box) im Bild
- Problemkomplexität skaliert mit Bildstruktur. . .
  - Code vor weißem oder buntem Hintergrund?
  - Mehr als ein Code im Bild?
  - Wie groß ist der/sind die Code(s)?
- Allgemeiner Ansatz:  
"Coderegion ist gekennzeichnet durch hohen Anteil an Schwarz- bzw. Grau-Weiss-Übergängen."



# Verfahren zur Lokalisation

## 2D-Barcodes

### 2D-Barcodes

### Lokalisierung

### Segmentier.

### Ausblick

- 1 Region-Growing
- 2 Binarisierung und morphologische Analyse  
⇒ zur Lokalisation der Coderegion
- 3 Auswertung des Gradientenbildes  
⇒ Lokalisation und Bestimmung der Orientierung  
⇒ rudimentäre Unterscheidung verschiedener Codes
- 4 Hough-Transformation



# Region-Growing

2D-Barcodes

2D-Barcodes

Lokalisierung

Segmentier.

Ausblick

- Ziel: Finde den Umriss der Region
- Achtung: die Kontur existiert nur virtuell!!!
- (einfache) Ansätze:
  - a) falls Code in normalisierter, achsenparalleler Lage  
 → Projektionshistogramme
  - b) sonst Region-Growing:
    - 1 finde einen Startpixel ("Seed") innerhalb des Codes
    - 2 suche in definierter Nachbarschaft weitere Codepixel, die mit dem Startpixel verbunden sind
    - 3 berechne konvexe Hülle aller gefundenen Pixel
    - 4 suche Pixel in der Hülle, die noch nicht zur Region gehören und mache mit 2. weiter
    - 5 wenn es keine weiteren Pixel mehr gibt oder Hülle stabil ist: berechne 4-Seiten-Polygon, das konvexe Hülle approximiert



# Morphologische Analyse

2D-Barcodes

2D-Barcodes

Lokalisierung

Segmentier.

Ausblick

- Extraktion (des Umrisses) der Coderegion
- Algorithmus:
  - binarisiere das Bild (stückweise), z.B. mit Otsu:
 

”Maximiere Varianz zwischen den Gruppen bei minimaler Varianz innerhalb der Gruppen.”

$$\sigma_{intra} = P(g_1) \sum_{i=1}^t (i - \mu_1)^2 + P(g_2) \sum_{i=t+1}^T (i - \mu_2)^2$$

$$\sigma_{inter} = P(g_1)(\mu_1 - \mu)^2 + P(g_2)(\mu_2 - \mu)^2$$

$$\Rightarrow \max \frac{\sigma_{inter}}{\sigma_{intra}}$$

- Dilatation/Closing bis alle Lücken geschlossen sind
- berechne Kontur der Region



# Gradientenbild-Analyse I

2D-Barcodes

2D-Barcodes

Lokalisierung

Segmentier.

Ausblick

- 2D-Barcodes weisen spezifische Vorzugsrichtungen für die Gradienten im Bild auf

- Algorithmus:

- berechne lokale Gradienten mit Sobel-Filter

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- berechne lokale Gradientenstärke  $G$  und -richtung  $\Phi$

$$G = \sqrt{S_x(x, y)^2 + S_y(x, y)^2}, \quad \Phi = \arctan \frac{S_y}{S_x}$$

- zerlege das Bild in disjunkte Blöcke
  - bestimme pro Block  $B$  Pixel  $P_G$  mit hohem Gradienten  $G > \theta_B$
  - berechne pro Block das Histogramm der Gradientenrichtungen für die  $P_G$
  - gruppierere Blöcke mit ähnlichen Histogrammen zu Regionen



# Gradientenbild-Analyse II

## 2D-Barcodes

## 2D-Barcodes

## Lokalisierung

## Segmentier.

## Ausblick

- Histogramm-Analyse:

- unimodal: 1D-Code, PDF



- bimodal: Datamatrix-Code



- sonst: Maxicode



- aus bimodalen Histogrammen kann Orientierung (grob) geschätzt werden

⇒ Hauptgradientenrichtungen entsprechen Zeilen-/Spaltenausrichtung

# Hough-Transformation



## 2D-Barcodes

### 2D-Barcodes

### Lokalisierung

### Segmentier.

### Ausblick

- Ziel: Extraktion der Vorzugsgeraden im Code  
 ⇒ integrierte Segmentierung der "Module"
- Algorithmus:
  - berechne Gradientenbild
  - suche Pixel mit großem Gradientenbetrag
  - Houghtransformation dieser Pixel
    - ⇒ Maxima im Hough-Array geben Geraden an
    - ⇒ max. Ausdehnung pro Gerade legt Region fest
- falls Bild des Codes fronto-parallel aufgenommen wurde  
 gibt es zwei Scharen paralleler Geraden
- bei perspektivischen Verzerrungen Fluchtpunkt-Kriterium



# Segmentierung der Module

## 2D-Barcodes

### 2D-Barcodes

### Lokalisierung

### Segmentier.

### Ausblick

- Ziel: Detektion der Module und ihrer Farbwerte
- Segmentierung ist stärker Code-abhängig
  - Maxicode:  
 Zerlegung des Codes in Strukturprimitiva wie Punkte,  
 Liniensegmente und Bögen



- Datamatrix:  
 finde orthogonales Raster



# Maxicode-Segmentierung



## 2D-Barcodes

### 2D-Barcodes

### Lokalisierung

### Segmentier.

### Ausblick

- Algorithmus:
  - detektiere Kantenpixel (Sobel, etc.)
  - Chaining mit Hystereseschwellwert und Non-Maximum-Suppression
    - ⇒ ein Pixel breite, markante Konturverläufe
  - Extraktion von Segmenten und Bögen
  
- Anwendung von Regeln wie...
  - ”Suche Punkt im Zentrum konzentrischer Bögen.”



- gesucht ist das Raster, d.h. die Breite und Höhe der einzelnen Module
- Ansätze:
  - Analyse der Verteilung der Schwarz-Weiß-Übergänge und Suche nach passender Modulgröße  
 ⇒ gemeinsamer Teiler, der äquidistante Zerlegung erlaubt
  - explizite *Corner-Detection* kombiniert mit Hough-Transformation (auch für Lokalisierung)  
 ⇒ finde Eckpunkte des Rasters und leite Modulgröße ab
- Quasi-Standard für Corner-Detection: Harris-Detektor



# Harris-Corners I

## 2D-Barcodes

## 2D-Barcodes

## Lokalisierung

## Segmentier.

## Ausblick

- der Corner-Detektor detektiert nicht nur "Corners"  
 ⇒ "Corners" sind alle lokal markanten Punkte
- Auswertung der lokalen Autokorrelations-Funktion
- Ausgangspunkt: Moravec's Detektor
  - gegeben ein Operatorfenster im Bild
  - analysiere Grauwertänderungen unter Verschiebung
    - a) falls Umgebung homogen, nur geringe Änderungen
    - b) falls Kante, geringe Änderungen entlang der Kante und große Änderungen orthogonal dazu
    - c) falls (isolierter) Eckpunkt, große Änderungen in allen Richtungen

$$\Rightarrow E(x, y) = \sum_{u,v} w_{u,v} \cdot |I_{x+u,y+v} - I_{u,v}|^2$$

- Corner-Detection durch Suche nach lokalen Maxima in E (oberhalb eines Schwellwertes)



# Harris-Corners II

2D-Barcodes

2D-Barcodes

Lokalisierung

Segmentier.

Ausblick

- Nachteile von Moravec:
    - Ansatz ist anisotropisch, d.h. richtungsabhängig
    - Berechnung ist rauschempfindlich durch Rechteckfenster
    - zu viele Punkte werden gefunden, weil nur Absolutwert von E zählt, aber nicht Varianz
- ⇒ Harris-Detector behebt diese Probleme:
- kontinuierliche Ableitungen statt richtungsabhängiger
  - Gauss-Glocke als Operatorfenster
  - Berücksichtigung der lokalen Varianz von E

$$\Rightarrow E(x, y) = (x, y) \begin{bmatrix} A & C \\ C & B \end{bmatrix} (x, y)^T$$

$$A = \frac{\delta^2 I}{\delta x^2}, \quad B = \frac{\delta^2 I}{\delta y^2}, \quad C = \frac{\delta^2 I}{\delta x \delta y}$$

# Harris-Corners III



2D-Barcodes

2D-Barcodes

Lokalisierung

Segmentier.

Ausblick

- Detektion durch Auswertung der Eigenwerte:
  - wenn beide annähernd 0, ist Punkt uninteressant
  - wenn einer groß und der andere klein, dann Kante
  - wenn beide groß, dann Ecke
- vereinfachte Berechnung:

$$\begin{aligned} \text{Tr}(M) &= A + B \\ \text{Det}(M) &= AB - C^2 \\ \Rightarrow R &= \text{Det} - k \cdot \text{Tr}^2 \end{aligned}$$

- für  $R$  gilt:
  - $R \gg 0$  nahe Ecken
  - $R \ll 0$  nahe Kanten
  - $R \approx 0$  in flachen Regionen
- Ecken und Kanten resultieren aus lokalen Minima/Maxima



- Einfluss der Kamera bislang unberücksichtigt
- alle Ansätze bislang auf Fronto-Parallelität ausgelegt  
 ⇒ Was passiert, wenn das nicht gilt?
- daher demnächst
  - Kamerageometrie und Kalibrierung
  - Linsenverzerrungen
  - Bildtransformationen
  - Korrektur der Perspektive