

# Formelsammlung Echtzeit Bildverarbeitung

Mario Felder

15. April 2014



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Rasterung . . . . .	1
1.2	Nachbarschaftsrelationen und Abstand . . . . .	3
1.3	Globale Charakterisierung von Bildern . . . . .	4
1.3.1	Histogramm . . . . .	4
1.3.2	Mittelwert . . . . .	4
1.3.3	Varianz . . . . .	4
<b>2</b>	<b>Punktoperationen und Bildverknüpfungen</b>	<b>7</b>
<b>3</b>	<b>Filteroperatoren im Ortsraum</b>	<b>9</b>
<b>4</b>	<b>Fourier-Transformation</b>	<b>11</b>
<b>5</b>	<b>Segmentierung und Merkmalsextraktion</b>	<b>13</b>
<b>6</b>	<b>Linien-Segmentierung und Merkmalsextraktion</b>	<b>15</b>
<b>7</b>	<b>Farbe</b>	<b>17</b>
<b>8</b>	<b>Anwendungen</b>	<b>19</b>

# Kapitel 1

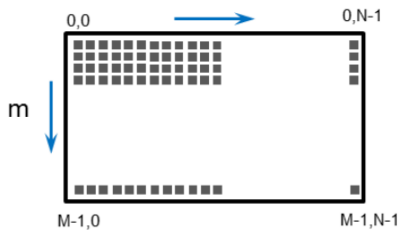
## Einführung

Die Bilddaten können mathematisch als Matrix beschrieben werden:

$$I = [I_{m,n}]$$

mit  $0 \leq m \leq M - 1$  und  $0 \leq n \leq N - 1$

**Achtung:** MATLAB verwendet das Intervall  $[1, N]$  bzw.  $[1, M]$



### 1.1 Rasterung

Die Rasterung ist ein Mass für die Detailtreue eines Bildes. Bei gegebener CCD- bzw. CMOS-Sensorgrösse ( $M \times N$  Pixel) wird die Auflösung

durch die geometrische Abbildung bestimmt.

Dabei gelten folgende Zusammenhänge:

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f} \quad \Leftrightarrow \quad b = \frac{f \cdot g}{g - f}$$

$f$ : Brennweite

$b$ : Bildweite

$g$ : Gegenstandsweite

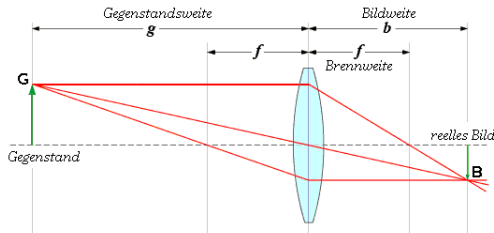
Häufig ist  $g \gg b$  und es kann somit  $b = f$  gesetzt werden.

Weiter ergibt sich daraus:

$$\frac{B}{G} = \frac{b}{g}$$

$B$ : Bildgrösse

$G$ : Gegenstandsgrösse



Auflösung bei einem Sensor mit  $N \times M$  Pixel Auflösung:

$$G_b = \frac{g \cdot W}{b \cdot N}$$
$$G_h = \frac{g \cdot H}{b \cdot M}$$

$G_b$  : Auflösung in  $\frac{mm}{Pixel}$  (Breite)

$G_h$  : Auflösung in  $\frac{mm}{Pixel}$  (Höhe)

$W$  : Breite des Sensors [mm]

$H$  : Höhe des Sensors [mm]

$N$  : Horizontale Pixel

$M$  : Vertikale Pixel

## 1.2 Nachbarschaftsrelationen und Abstand

Vierer-Nachbarschaft:

	$m - 1, n$	
$m, n - 1$	$m, n$	$m, n + 1$
	$m + 1, n$	

Achter-Nachbarschaft:

$m - 1, n - 1$	$m - 1, n$	$m - 1, n + 1$
$m, n - 1$	$m, n$	$m, n + 1$
$m + 1, n - 1$	$m + 1, n$	$m + 1, n + 1$

Euklidische Abstandsnorm:

$$d(I_{m,n}, I_{p,q}) = \sqrt{(m - p)^2 + (n - q)^2}$$

Maximum Abstandsnorm:

$$d(I_{m,n}, I_{p,q}) = \max(|m - p|, |n - q|)$$

## 1.3 Globale Charakterisierung von Bildern

### 1.3.1 Histogramm

Das Histogramm gibt die absolute oder relative  $p_I(g)$  Häufigkeit aller Grauwerte  $g \in [0, 255]$  eines Bildes an.

Relative Häufigkeit:

$$0 \leq p_I(g) \leq 1 \quad , \forall g$$
$$\sum_g p_I(g) = 1$$

Kumulative Häufigkeit:

$$h_I(g) = \sum_{g' \leq g} p_I(g') \quad , h_I(255) = 1$$

### 1.3.2 Mittelwert

$$\mu_I = \frac{1}{M \cdot N} \sum_{m,n} I_{m,n} = \sum_g g \cdot p_I(g)$$

### 1.3.3 Varianz

$$\sigma_I^2 = \frac{1}{M \cdot N} \sum_{m,n} (I_{m,n} - \mu_I)^2 = \sum_g (g - \mu_I)^2 \cdot p_I(g)$$

Lösung in MATLAB:

```
1 %read image
2 Image = imread( 'sample.png' );
3
4 %using formula
5 [M, N] = size(Image);
6 mu = sum(Image(:))/(M*N)
7
8 sd = sum((double(Image(:)) - mu).^2)/(M*N);
9 sd = sqrt(sd)
10
```

```
11 %with MATLAB function
12 mu = mean2(Image)
13 sd = std2(Image)
14
15 %using the histogram
16 [Count, Value] = imhist(Image);
17 RelCount = Count/sum(Count);
18 mu = sum(Value .* RelCount)
19
20 sd = sum((Value - mu).^2 .* RelCount);
21 sd = sqrt(sd)
```

Lösung in C:

```
1  for(int m = 0; m < M; m++){
2      for(int n = 0; n < N; n++){
3          sum += Image[m][n];
4      }
5  }
6  mu_I = sum/(M*N);
7
8  /**
9   *
10  * TODO
11  *
12  **/
```





## Kapitel 2

# Punktoperationen und Bildverknüpfungen



## Kapitel 3

# Filteroperatoren im Ortsraum



## Kapitel 4

# Fourier-Transformation



## Kapitel 5

# Segmentierung und Merkmalsextraktion





## Kapitel 6

# Linien-Segmentierung und Merkmalsextraktion



# Kapitel 7

## Farbe



## Kapitel 8

# Anwendungen