

CPII - V10

2D Faltung

[`scipy.signal.convolve2d()`]

besser: $F^{-1}(F(f) * F(g))$

speziell: `scipy.ndimage.filters.gaussian_filter()`

Filter: `gaussian, blur, comet, belichtung`

`scipy.ndimage.filters :`

`uniform_filter ()`

`median_filter ()`

Kantenerkennung

Faltung mit $[1, -1]$ oder $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ (Kanten in x bzw. y finden)

Kanten finden: $\sqrt{I_x^2 + I_y^2}$

Sobel-Filter `scipy.ndimage.sobel ()`

Bewegungskorrektur (Belichtungszeit)

* Einfache Entfaltung

Idee : Suche PSF um das Bild zu Entfalten

$F(S) = F(B) * F(K)$

$B = F^{-1} \left(\frac{F(S)}{F(K)} \right)$

`scipy.signal.deconvolve () : nur 1D`

* Wiener Entfaltung

Wiener Filter: reduziert Rauschen

`scipy.signal.wiener()`

$B = F^{-1} \left(\frac{F(Gl\ ttung(S))}{F(K)} \right)$

Allgemein : Bildrauschen entfernen

Gauss-Filter, Median-Filter, Wien-Filter

Bildanalyse

A) Sementation (Graustufen -> schwarz/weiss)

-> Histogramm

B) Morphologie

1. "binary open"

2. "binary close"

`scipy.ndimage.binary_opening ()`, `scipy.ndimage.binary_closing ()`

`grey_opening ()`, `grey_closing ()`

`morphology_gradient ()`

`scipy.ndimage.label ()`

Granulometry : Größe von Gebieten bestimmen :

binary_open mit verschieden großen Kreisen

Pattern - Matching : Suche nach Muster -> Korrelation

$$z(t) = \int_{-\infty}^{\infty} f(\tau) g(t+\tau) d\tau$$

$$z_n = \sum f_k g_{k+n}^*$$

Autokorrelation: $f = g$

$$\text{Kreuzkorrelation: } (f \times g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t+\tau) d\tau$$

Es ist: $f \times g = f(t) * g(-t)$