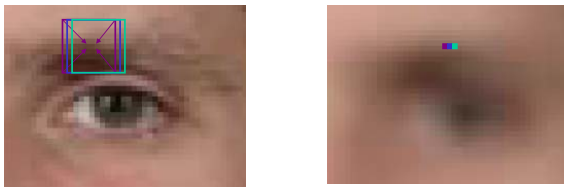
T3 – Linear Filtering
Edge Detection, Wiener Filter

Christian Vögeli
cvoegeli@inf.ethz.ch

Aufgabe 1 – Filtering in Matlab

Aufgabe 1 – Filtering in Matlab



combination rule defines filter type
=> Matlab: `fspecial ('rule', fsize)`

Matlab & Aufgabe 1b)

- Warning:
  - Matlab is powerful
  - Takes getting used to
  - More or less elegant ways
- Aufgabe 1b):
  - straight forward linear filtering
- Displaying images in Matlab:
  - `figure;`
  - `subplot(2,3,1); imshow(im1);`
  - `subplot(2,3,3); imshow(im3);`

divides figure into 2x3 array

selects subplot position

Aufgabe 1 c)

- MipMaps:
  - Filtering: aliasing caused by texture subsampling
  - Gaussian filtering is one way to obtain the pyramid
  - Often: tri-linear interpolation
- Filter:
  - `fspecial('gaussian',fsize,sigma);`
- Sampling:
  - Array access on image:
  - `imaf(1:2:size(gp{i-1},1),1:2:size(gp{i-1},2));`

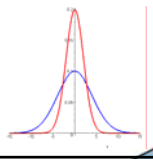
entire task e.g. 3

e.g. 0.5

Aufgabe 1 d)

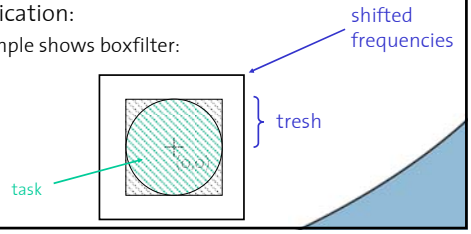
- Laplacian of Gaussian (Idea):
  - Large brightness changes
    - => steep slope of intensity function
      - => large/extremal first derivative
        - => zero crossings of second derivative
- Filter size depending on σ:
  - Larger σ needs larger filter

Filtering for noise reduction



Aufgabe 1 e)

- Lowpass/highpass filtering:
  - Here: box-filter in frequency domain
  - Artifacts in image domain? …
- Clearification:
  - Example shows boxfilter:



Aufgabe 1 f)

- Wiener Filter:
  - Aufgabe 2)
- Add noise
- Restore original image using different window sizes

Aufgabe 2 – Wiener Filter

- Image reconstruction problem:

f(x) ⟶ (f*h)(x) ⟶ (f*h)(x) + η(x)

II ? -> 2a)

o(x) ⟵ $\tilde{h}(x)$

- Given the optimality criterion, we can „simply" find $\tilde{h}(x)$:

$$\Phi_{bd}(x) = (\Phi_{bb} * \tilde{h})(x)$$ Wiener-Hopf equation

d: desired f, often d = f

Aufgabe 2 b)

$$\Phi_{bf}(x) = (\Phi_{bb} * \tilde{h})(x)$$

- Convolution -> integral
  -> solve in Fourier domain -> multiplication
- lecture slides…

$\mathcal{F}[\tilde{h}](u)$ = …

Aufgabe 2 c)

- 1/SNR = 0
  => $\mathcal{F}[\tilde{h}](u) = \frac{1}{\hat{h}(u)}$
- Restoration w/o noise:

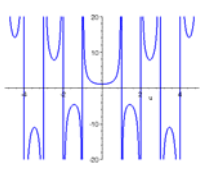
f(x) ⟶ g(x) ⟶ f(x)

h(x) ↓ $\mathcal{F}[h](u)$; h(x) ↓ $\mathcal{F}[h](u)$

=> $\mathcal{F}[h](u) \cdot \mathcal{F}[h](u) = 1$

Aufgabe 2c) – e)

- Maple: define functions
- plot(1/h(u), u=-5..5, -20..20, numpoints=5000);



limitation because 1/0

noise amplification

- interpret