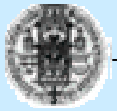


ALBERT-LUDWIGS-
UNIVERSITY FREIBURG

Institute for Pattern Recognition
and Image Processing
Computer Science Department

SOMMERCAMPUS 2004

MATLAB-KURS

3. Teil Toolboxes und Open-Source

Veranstalter:

GERD BRUNNER, Bernard Haasdonk, Klaus Peschke

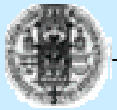
Übersicht:

- Toolboxes allgemein
- Speziell: Image Processing Toolbox
- Open Source Toolboxes
- Was gibt es sonst noch?

Toolboxes

Prinzip der Toolboxes

- Modular aufgebautes System
- Funktionssammlungen (erweiterte Bibliotheken)
- Graphische Tools
- Demos
- Meisten Funktionen editierbar à Benutzerspezifische Modifikationen möglich
- Einzeln benutzbar à allerdings starke Einschränkung
- Viele Open-Source Programme und Toolboxes



ALBERT-LUDWIGS-
UNIVERSITY FREIBURG

Institute for Pattern Recognition and Image Processing
Computer Science Department

Toolboxes und Blocksets

Aerospace Blockset
Bioinformatics Toolbox
CDMA Reference Blockset
Communications Blockset
Communications Toolbox
Control System Toolbox
Curve Fitting Toolbox Data Acquisition Toolbox
Database Toolbox Datafeed Toolbox Embedded Target for Infineon C166® Microcontrollers Embedded Target for Motorola® HC12
Embedded Target for Motorola® MPC555
Embedded Target for OSEK/VDX® Embedded Target for TI C2000™ DSP
Embedded Target for TI C6000™ DSP Excel Link
Extended Symbolic Math Toolbox Filter Design HDL Coder Filter Design Toolbox
STATISTIC TOOLBOX
Financial Derivatives Toolbox Financial Time Series Toolbox Financial Toolbox Fixed-Income Toolbox Fixed-Point Toolbox
Fuzzy Logic Toolbox GARCH Toolbox Gauges Blockset
Genetic Algorithm and Direct Search Toolbox
Image Acquisition Toolbox
IMAGE PROCESSING TOOLBOX
Instrument Control Toolbox
Link for Code Composer Studio™ Link for ModelSim®
Mapping Toolbox MATLAB®
MATLAB Builder for COM
MATLAB Builder for Excel
MATLAB Compiler MATLAB Report Generator MATLAB Web Server
Model Predictive Control Toolbox
WAVELET TOOLBOX
SPLINE TOOLBOX
Model-Based Calibration Toolbox
NEURAL NETWORK TOOLBOX
OPC Toolbox
OPTIMIZATION TOOLBOX
Partial Differential Equation Toolbox Real-Time Windows Target Real-Time Workshop®
Real-Time Workshop Embedded Coder RF Blockset RF Toolbox
Robust Control Toolbox Signal Processing Blockset Signal Processing Toolbox SimDriveline SimMechanics SimPowerSystems
Simulink® Simulink Accelerator
Simulink Control Design Simulink Fixed Point Simulink Parameter Estimation Simulink Report Generator Simulink Response Optimization Simulink Verification and Validation Spline Toolbox Stateflow® Stateflow Coder Statistics Toolbox
Symbolic Math Toolbox System Identification Toolbox Video and Image Processing Blockset
Virtual Reality Toolbox Wavelet Toolbox xPC Target xPC Target Embedded Option xPC TargetBox™

The MathWorks - MATLAB Technical Computing

IMAGE PROCESSING TOOLBOX

IPT bietet:

- Ein unfangreiches Angebot an Standard Algorithmen
- Graphische Tools für Bildverarbeitung, -analyse, -visualisierung
  und zur Algorithmenentwicklung
- Wiederherstellen von verrauschten Bildern (Filter)
- Bildqualitätsverbesserung, Merkmalsextraktion
- Analyse von Texturen und Formen
- Räumliche Transformationen von Bildern
- Alignment (Angleichen von 2 Bildern)
- Visualisierung von Bilder
- FFT, DCT, Radon, ROI-Extraktion, Faltung, Farbraumkonvertierung,
  Kantendetektion, Morphologie, …

IPT – Wichtige und hilfreiche Funktionen

```
>> info = imfinfo('44001.jpg')

info =

        Filename: '44001.jpg'
        FileModDate: '17-Apr-1999 17:45:08'
        FileSize: 33048
        Format: 'jpg'
        FormatVersion: ''
        Width: 256
        Height: 384
        BitDepth: 24
        ColorType: 'truecolor'
        FormatSignature: ''
        NumberOfSamples: 3
        CodingMethod: 'Huffman'
        CodingProcess: 'Sequential'
        Comment: {}
```

MATLAB-HILFE!

Ein Bild laden:

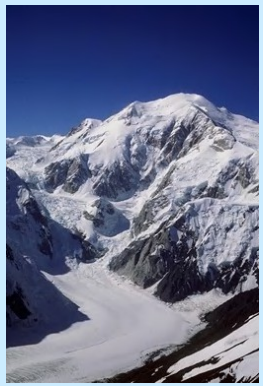
```
>> image_1 = imread(filename, fmt);   % für Grauwert- und Farbbilder,
                                      % mögliche Formate: jpg. gif, tiff, png, hdf, bmp….
>> size(image_1)                      % gibt die Größe aus,  das Format ist wichtig!!
ans =
384   256     3
```

Ein Bild speichern:

```
>> imwrite(image_1,filename,fmt);     % einfachster Fall, jedes Format hat zusätzliche
                                      %  Parameter, z.B. Kompression, Metadaten, Farbsystem….
```

Bildtypen konvertieren:

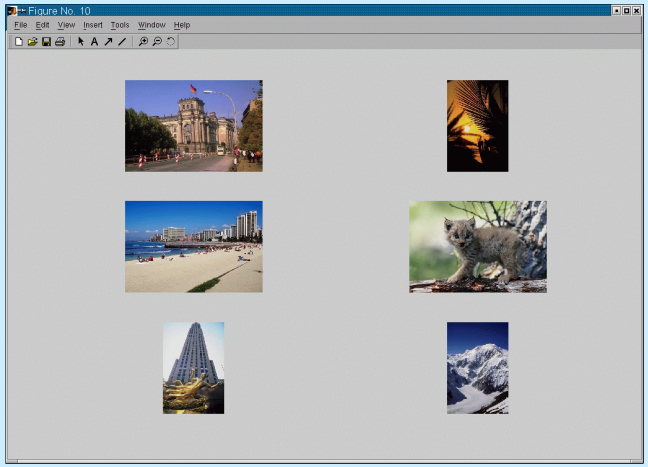
- *im2double;*
- *rgb2gray;*
- *rgb2ind;*
- *im2bw;*
- *label2rgb;*



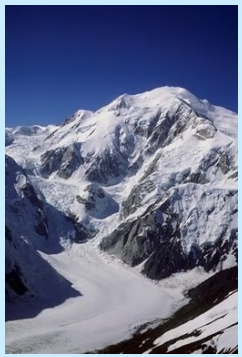
à

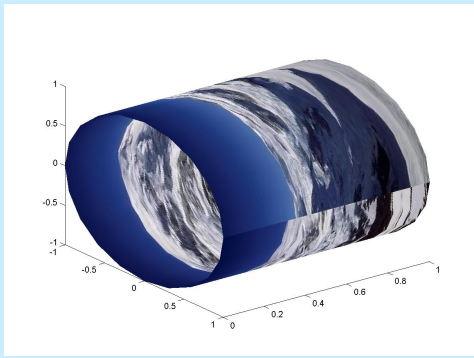
Bilddarstellung:

- colorbar % Farbbalken
- imshow % Bild wird gezeigt
- subplot % mehrere Bilder
- figure % Fenster
- zoom
- warp % Abbildung eines
  % Bildes auf ein Objekt



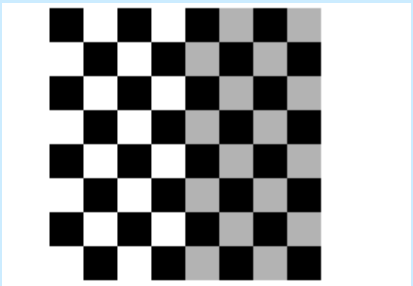

```
>> [x,y,z] = cylinder;
>> warp(z,x,y,image_1);
```

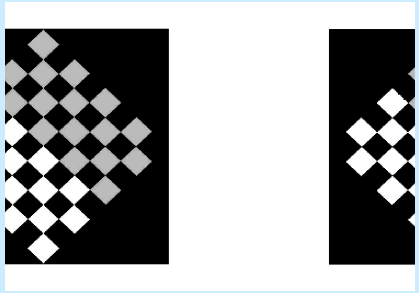




RÄUMLICHE BILDTRANSFORMATIONEN

Aliasing Ani-Aliasing Interpolation (pixel-basiert) Geometrische Transf.	Wenn Bildgröße reduziert wird; à Pixel weniger Entsprechende Korrektionsalgorithmen Bicubic (4x4 nächste Nachbarn), Bilinear (2x2) Rotieren, verkleinern, größe ändern,…
I = checkerboard(20); imshow(I)	Erstellt ein Schachbrettmuster



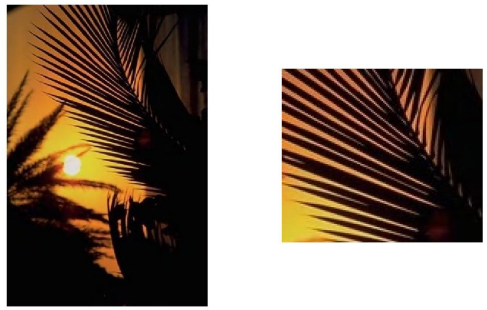


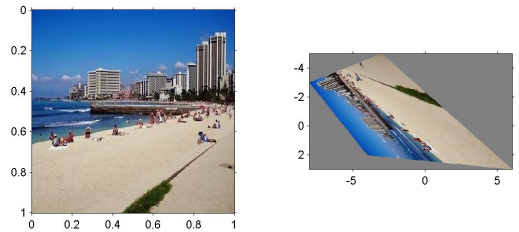
RÄUMLICHE BILDTRANSFORMATIONEN

```
>> image_crop = imcrop(image,[75 68 130 112]);
```

..weitere hilfreiche Funktionen:
- imresize
- imrotate
- imtransform

```
>> I = imread(,image');
>> udata = [0 1];  vdata = [0 1];  % input coordinate
                                   % system
>> tform = maketform('projective',[ 0 0;  1  0;  1  1; 0 1],...
                                  [-4 2; -8 -3; -3 -5; 6 3]);
>> [B,xdata,ydata] = imtransform(I, tform, 'bicubic', ...
     'udata', udata,  'vdata', vdata,  'size', size(I),  'fill', 128);
>> subplot(1,2,1), imshow(udata,vdata,I), axis on
>> subplot(1,2,2), imshow(xdata,ydata,B), axis on
```



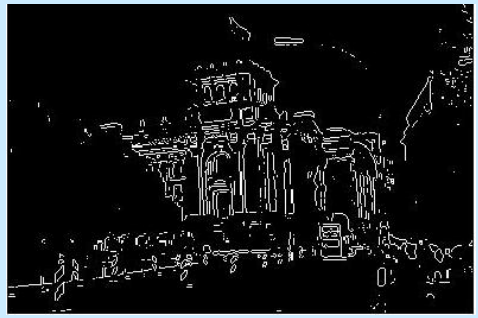


BILDANALYSE- UND STATISTIK

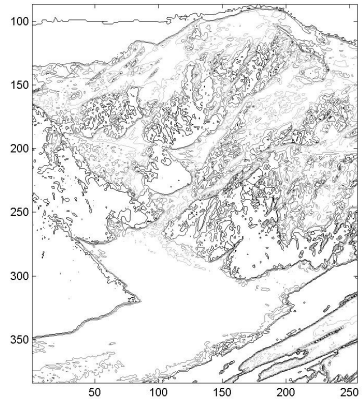
Bildanalyse:	
Adaptive Filter	Ein Filter mit räumlich variierenden Eigenschaften
Konturen	Linien konstanter Intensität
Kantendetektion	Bereiche mit großen Intensitätsänderungen
Eigenschaften	Bild- oder regionenbasiert (z.B. Fläche, Bounding-Box)
Histogramm	Verteilung der Bildintensität
Quadtree Dekomposition	Bild wird in homogene Blöcke partitioniert

- edge — Kantendetektion: Sobel, Prewitt, Canny, …
- hough — Berechnet die Hough-Transformation

>> edgedemo



BILDSTATISTIK	
pixval	Pixelwert (Region)
imcontour	Contourplot eines Bildes
imhist	Bildhistogramm
mean2, std2, corr2	Durchschnitt, Std, Korrelation
regionprops	Mißt verschiedenste Eigenschaften einer ausgewählten Region z.B. Konvexe Hülle,



Nachbarschaft und Blockoperationen

Blockoperation Boarder padding	Bild in Blöcken verschiedenster Größe verarbeitet Es können zusätzliche Spalten und Zeilen hinzugefügt werden
Spaltenoperationen Nachbarschaftsberechnungen	Vor der Verarbeitung wird ein Bild in Zeilenform gebracht Berechnungen wie Faltung, Filter…

```
I2 = blkproc(I,[4 6],@myfun);
% Man kann eine Inline Funktion definieren, z.B.
f =  inline('mean2(x)*ones(size(x))');
I2 = blkproc(I,[4 6],f);
```

LINEARE FILTER UND FILTERDESIGN

Filtern ist ein Technik, welche Bildverbesserungen oder Bildveränderungen erlaubt. Matlab bietet eine Vielzahl an vordefinierten Filteroperationen

Faltung	Operation basierend auf Nachbarpixel Viele Bildverarbeitungsoperationen als Faltung implementiert – z.B. Glättung, Bildverbesserung….
Korrelation	Eine Nachbarschaftsoperation, wo jedes Ausgabepixel einer gewichteten Summe von benachbarten Eingabepixel entspricht. Die Gewichte sind durch den Korrelationskern definiert.
FIR Filter (Finite impuls filter)	Ein Filter mit endlicher Impulsantwort

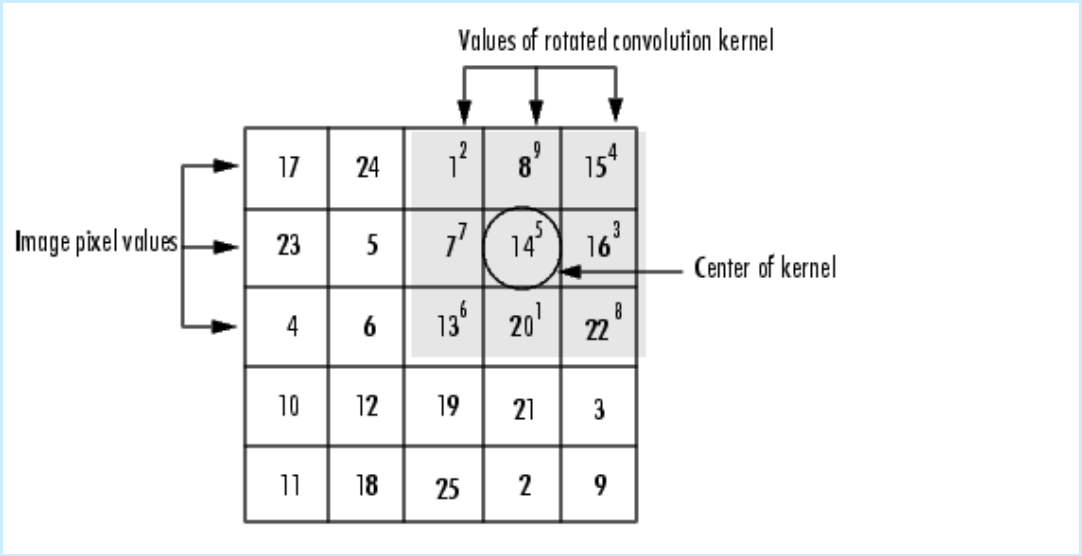
Faltung

Berechnung des Pixels (2,4)

$$1 \cdot 2 + 8 \cdot 9 + 15 \cdot 4 + 7 \cdot 7 + 14 \cdot 5 + 16 \cdot 3 + 13 \cdot 6 + 20 \cdot 1 + 22 \cdot 8 = 575$$

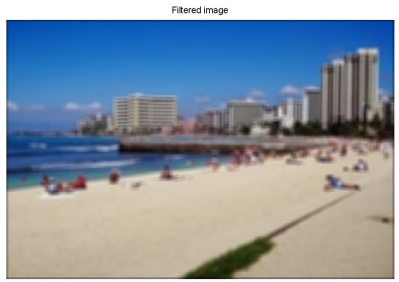
```
% Bild A
A = [17  24   1   8  15
      23   5   7  14  16
       4   6  13  20  22
      10  12  19  21   3
      11  18  25   2   9]

% Faltungskern
h = [8   1   6
      3   5   7
      4   9   2]
```

à Kern um 180 rotiert....



Beispiel: Durchschnittsfilter

```
>> I = imread(‘image’);
>> h = ones(5,5) / 25;
>> I2 = imfilter(I,h);
>> imshow(I), title('Original image')
>> figure, imshow(I2), title('Filtered image')
```



Boundary padding Option:

Wenn man ein Pixel am Bildrand mittels Faltung berechnen will, wird ein Teil des Faltungskernes außerhalb des Bildes sein.

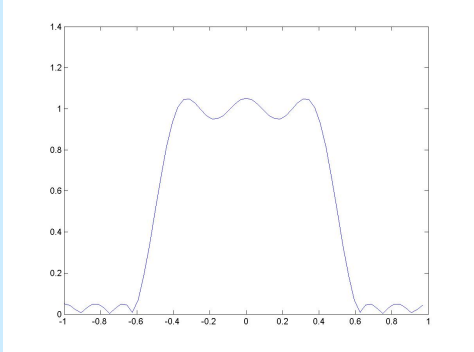
- Zero padding

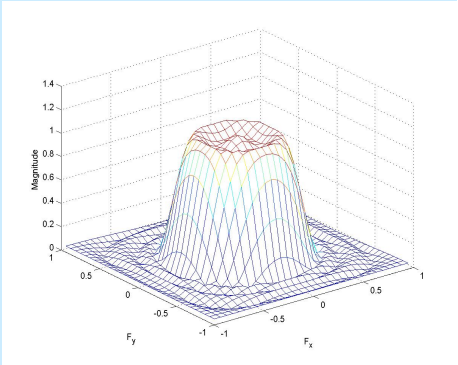
- Boarder replication

```
>> I2 = imfilter(I,h,’replicate’);
```

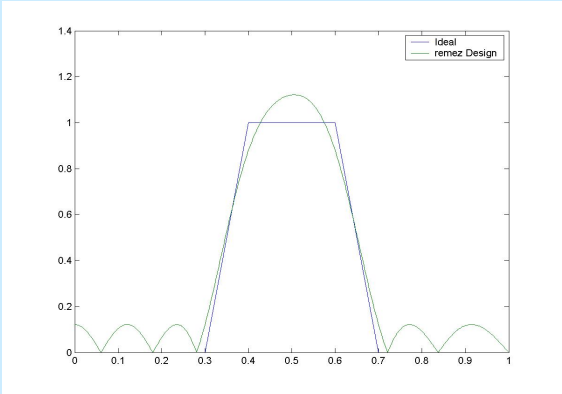
FILTERDESIGN

Frequenz-transformationsmethode	1-dim à 2 dim FIR Filter Es werden die meisten Charakteristika des 1-dim Filers erhalten.
>> b = remez(10,[0 0.4 0.6 1],[1 1 0 0]); >> h = ftrans2(b); >> [H,w] = freqz(b,1,64,'whole'); >> colormap(jet(64)) >> plot(w/pi-1,fftshift(abs(H))) >> figure, freqz2(h,[32 32])	Linearer Filter nach dem Parks-McClellan Alg. Der Filterfehler zw. optimalem und approx. Filter ist minimiert!



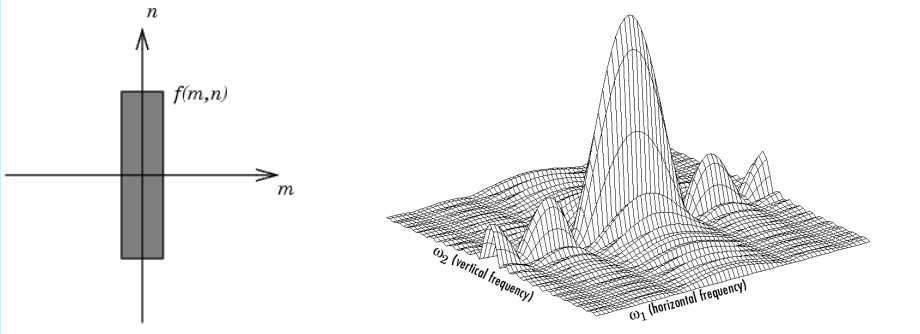


>> firdemo % DEMO



TRANSFORMATIONEN

Diskrete Fourier Trans. (FFT)	fft, fft2, ifft2, ifft
Diskrete Kosinus Trans (DCT)	dct, dct2, idct2,idct
Radon Transformation	radon, iradon



>> dctdemo % Demo zeigen

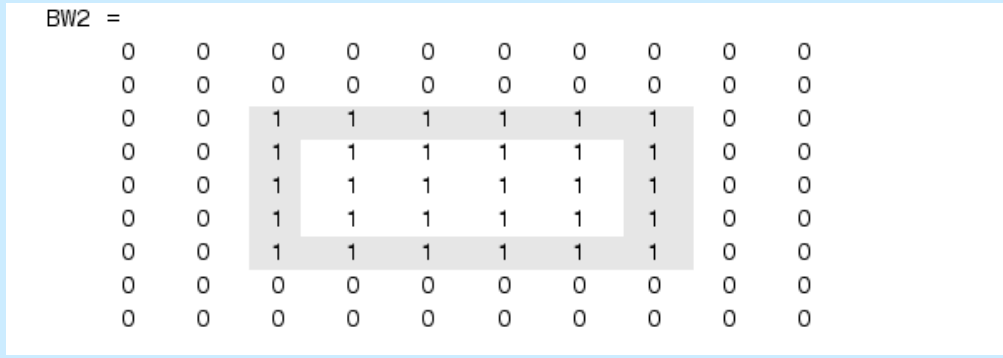
Morphologische Operationen

Morphologie ist eine auf Form basierte Bildverarbeitungstechnik. Pixelwerte im Ausgabebild hängen von den korrespondierenden Pixel des Eingabebildes und deren Nachbarn ab.

Dilation	Fügt Pixel zu Objekträndern hinzu
Erosion	Entfernt Pixel von Objekträndern
Strukturelement	Meistens das zu „verarbeitende“ Objekt bzw. Muster (typischerweise kleiner als das Bild)

```
>> BW = zeros(9,10);   % Dilatation eines binären
                       % Bildes (Rechteck)
>> BW(4:6,4:7) = 1
BW =
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     1     1     1     0     0     0
     0     0     0     1     1     1     1     0     0     0
     0     0     0     1     1     1     1     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

```
>> SE = strel('square',3)
SE =

Flat STREL object containing 3
neighbors.

Neighborhood:
     1     1     1
     1     1     1
     1     1     1
>> BW2 = imdilate(BW,SE)
```



```
BW2 =
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     1     1     1     1     1     1     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Weitere Funktionalität der IPT

Abstandstransformationen - Euklidisch - City Block - Chessboard	Bestimmen Entfernung von 2 Punkten
Regionenbasierte Berechnungen >> BW = roipoly >> BW2 = roifill	 Region of interest definieren ROI mit best. Werten auffüllen

Dekonvolution

Dekonvolution	Inverser Prozess der Faltung
Distortion Operator	Prozess der für die Veränderung des Bildes verantwortlich ist. z.B. Distortion verursacht durch eine PSF (Point spread function)
Verschiedenste Verfahren: deconvwnr	Dekonvolution mittels Wienerfilter (ohne Rauschen, perfekter inverser Filter)

Open Source Toolboxes

- Mathtools.net - Link Exchange for the Technical Computing Community
- http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html (Bayes Toolbox)
- http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html (HMM)
- http://www.vision.caltech.edu/bouguetj/calib_doc/ (Kamera Kalibrierungs TB)
- http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox/ (SVM TB)
- http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html (Speech Processing TB)
- …. und viele mehr…

Hinweise zu den Aufgaben:

Faltung von Bilder: conv2 (2-dim. Faltung)

Erzeugung von .avi Dokument: avifile (Filename angeben)
getframe (gibt movieframe zurück)
movie (spielt ein movie ab)
addframe (fügt frame dem movie hinzu)
aviinfo, frame2im (konvertiert einen Frame in ein Bild)

Weitere Faltungsfunktionen: fspecial (Erzeugung von Filter)
deconvwnr (Wiener Filter)

Hilfreiche Funktionen: dir (Infos übe ein Verzeichnis)
strcat (verbindet Strings), imread, uint8, imfilter

Aufgabe 1:

Schreiben sie eine Funktion, welche auf dem Grünkanal eines beliebigen RGB Bildes den nachstehenden Tiefpassfilter anwendet:

Filter = ones(9,9) /81;

(Verwenden sie das Bild: 44025.jpg)

Aufgabe 2:

„Warpen“ (verzerren) sie auf einer Kugle ein beliebiges Bild. Verwenden sie dazu die Funktion warp.

Aufgabe 3

Die Datei blurred.avi beinhaltet eine Bildsequenz von einem unklaren (blurred) Objekt. Alle Bilder sind im selben Format (.png) mit einer Größe von 128x128x3 abgespeichert. Sehen sie sich das .avi unter Matlab an. Benutzen sie dazu die zuvor aufgelisteten Funktionen.
Nachdem sie das Movie visualisiert haben, soll eine geeignete Dekonvolution (Entfaltung) für jeden Movieframe gefunden werden. Für die Entfaltung ist die Kenntnis des ursprünglichen Filters notwendig.

PSF = fspecial(„motion“,LEN,THETA); LEN = 31; THETA = 11;

Führen sie nun die geeignete Dekonvolution für jeden Frame durch.
(Hinweis: Wiener Filter, deconvwnr).
Speichern sie die Bildsequenz erneut als .avi ab und sehen sich das Ergebnis an.
Optional können sie auch noch ein AVI des originalen Bildmaterials erstellen.