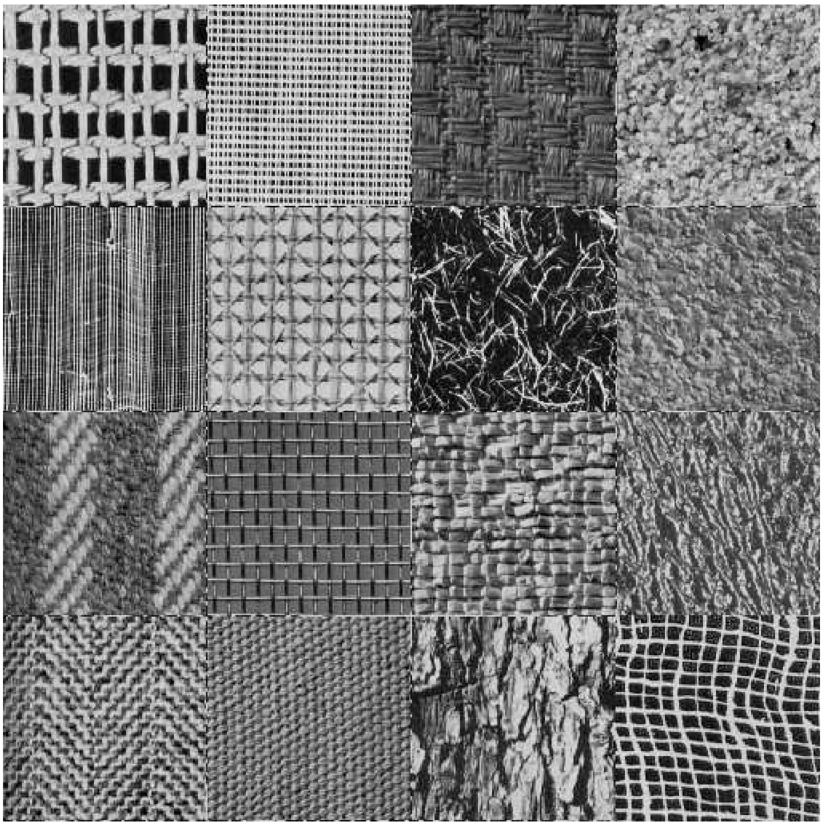
Lecture 20:
Texture Analysis

Contents

What is Texture? (1)



16 different textures. Author: P. Brodatz.

What is Texture?

- no general formal definition
- intuitive notion as a repeating pattern
- often with local stochastic fluctuations:
  size, shape, orientation, and colour of the underlying pattern *(textons)* may vary
- Texture analysis often combines empirically successful statistical expressions.
- Texture is a scale phenomenon.
- Our visual system recognises a textured region as a single object.
  It can distinguish between differently textured regions.
  This is called *texture discrimination.*

M I A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Problems

- How can one discriminate two textures?
  (Traditional edge detectors cannot separate two regions with different textures.)
- Is it possible to describe the complex structure of a texture with a few parameters?
- Can one even synthesise texture for inpainting purposes?

Texture Analysis without Neighbourhood Context

Often the average grey value is not sufficient to distinguish different textures. Then better descriptors such as statistical moments may be useful.

Statistical Moments

- Consider an image with N pixels with grey values f_1,...,f_N and mean μ. Then its *central moment of order k* is defined as

$$M_k \;:=\; \frac{1}{N}\sum_{i=1}^{N}(f_i-\mu)^k.$$

- Equivalent alternative characterisation with the image histogram: Consider a histogram $p(z_1)$,...,$p(z_L)$: z_1,...,z_L are the different grey values, and $p(z_i)$ is the probability that z_i occurs. Then the central moment of order k is given by

$$M_k \;:=\; \sum_{i=1}^{L} p(z_i)\,(z_i-\mu)^k.$$

Important Central Moments

- $k=2$: *Variance (Varianz)*

$$\sigma^2 \;:=\; M_2 \;=\; \frac{1}{N}\sum_{i=1}^{N}(f_i-\mu)^2$$

  measures the quadratic fluctuations around the mean

- $k=3$: *Skewness (Schiefe)*

$$V \;:=\; \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i-\mu}{\sigma}\right)^3$$

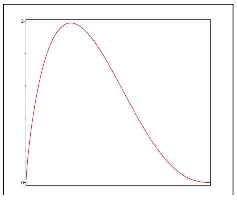
  measures the asymmetry:
  $V<0$: skewness towards left side.
  $V>0$: skewness towards right side.
  The normalisation by σ makes different distributions comparable.

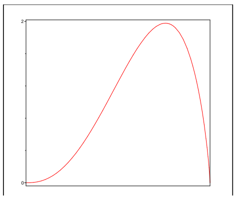
Texture Analysis without Neighbourhood Context (3)

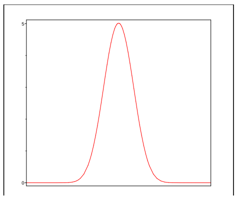
- $k = 4$: *Excess, Kurtosis (Exzess, Kurtosis)*

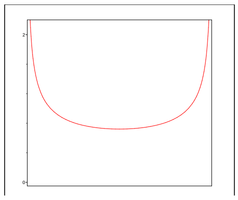
$$\varepsilon := \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i-\mu}{\sigma}\right)^4 - 3$$

measures the flatness of the distribution:
Subtraction of 3 gives $\varepsilon = 0$ for a Gaussian distribution.


positive skewness

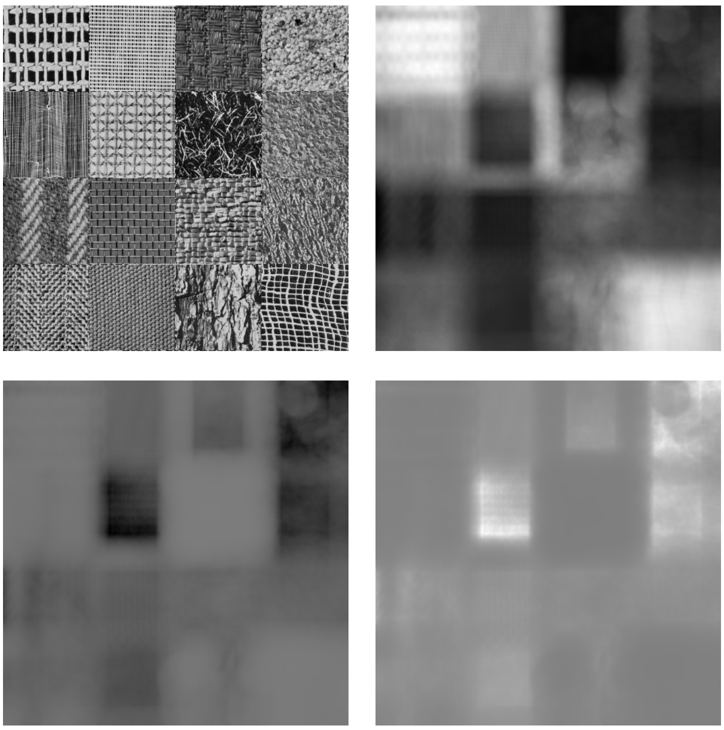

negative skewness


positive kurtosis


negative kurtosis

Examples of distributions. Author: B. Burgeth.

Texture Analysis without Neighbourhood Context (4)



Top left: Original image, 512×512 pixels. **Top right:** Variance, averaged within a disk-shaped neighbourhood of radius 30, and affinely rescaled to [0,255]. **Bottom left:** Skewness, affinely rescaled. **Bottom right:** Kurtosis, affinely rescaled. Author: J. Weickert.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Texture Analysis without Neighbourhood Context (5)

How to Discriminate Textures

- Replace every grey value by one or more texture attributes (e.g. statistical moments) within some window.
- Apply a classical edge detector to the resulting scalar- or vector-valued feature image (Lecture 13).
- Edge detection of a vector-valued image $\boldsymbol{f} = (f_1, ..., f_m)^\top$ proceeds in a similar way as for the scalar-valued case:
  For instance, instead of the gradient magnitude one uses

$$\left(\sum_{k=1}^{m} |\boldsymbol{\nabla} f_k|^2 \right)^{1/2}$$

Texture Analysis without Neighbourhood Context (6)

Disadvantage of All Texture Descriptors So Far

- If one uses them for a *global* description of an image or a region, then the spatial ordering of the pixels does not matter:
  A checkerboard image has the same texture descriptor as an image with a black and a white half.
- Statistics with features that ignore the spatial context is called *first-order statistics.*
- Incorporating the neighbourhoud context would be desirable.
  This leads to *second-order statistics.*

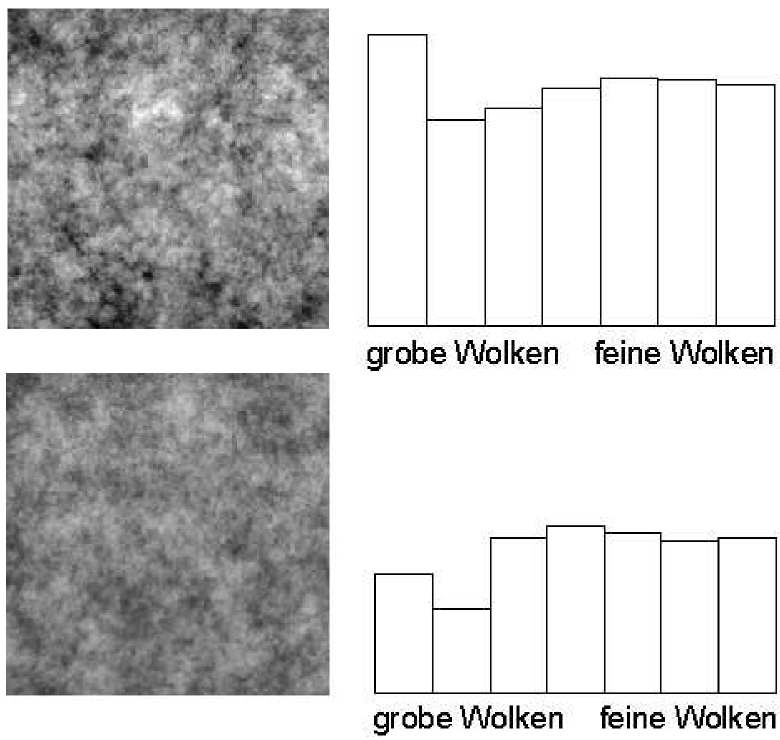
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Texture Analysis with Neighbourhood Context

Possibility 1: Multiscale Representation

- Characterise the image on different scales, e.g. using
  - pyramids (Lecture 6)
    Example: Analysing the cloudiness in fabrics
  - scale-space concepts (Lecture 12)
    Example: Convolving the image with Gaussians of increasing standard deviation (Gaussian scale-space)
- Afterwards use a texture descriptor with first-order statistics on each scale.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33



Top: Fabric image with standard deviation σ at different scales of a Laplacian pyramid representation.
Bottom: The same representation for another fabric image. Authors: J. Weickert, R. Rösch.

Possibility 2: Cooccurrence Matrices (Grauwertübergangsmatrizen, Kookkurenzmatrizen)

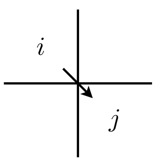
(Haralick et al. 1973)

- Basic idea: histogram of neighbourhood relations
- Specify some displacement vector $\boldsymbol{d} = (d_1, d_2)^\top$.
  It identifies pairs of points that are to be compared.
  Example: Using $\boldsymbol{d} = (1, 0)^\top$ compares every pixel with its right neighbour.
- Create a bivariate grey value histogram $p_{i,j}$.
  It specifies the probability that some grey value i occurs together with some grey value j in direction $\boldsymbol{d}$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

2	1	2	0	1
0	2	1	1	2
0	1	2	2	0
1	2	2	0	1
2	0	1	0	1

5×5 image



$d = (1, 1)^\top$

$$\begin{array}{cc|ccc|c} & & & j & & \\ & & 0 & 1 & 2 & \\ \hline & 0 & 0 & 2 & 2 & \\ i & 1 & 2 & 1 & 2 & \cdot \frac{1}{16} \\ & 2 & 2 & 3 & 2 & \end{array}$$

p(i,j)

Left: A 5×5 image with three greyscales 0, 1 and 2. **Middle:** The displacement vector $\boldsymbol{d} = (1, 1)^\top$ allows to analyse the lower right neighbour (x axis points to the right, y axis points downwards). **Right:** Cooccurrence matrix corresponding to $\boldsymbol{d} = (1, 1)^\top$. Example: A grey value 2 having a grey value 1 as lower right neighbour occurs in 3 out of 16 cases. Therefore, $p_{2,1} = \frac{3}{16}$. Author: T. Schneevoigt, adapted from R. Jain, R. Kasturi, B. G. Schunck.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Characteristic Expressions for Cooccurrence Matrices

The information in coocurrence matrices is very rich.
For the purpose of texture discrimination, it should be condensed.
Many heuristically motivated expressions have been proposed:

- *Largest Probability*

  The maximal $p_{i,j}$ gives the most frequent grey value configuration with respect to the displacement vector $\boldsymbol{d}$.
  Example: In the previous image with $\boldsymbol{d} = (1,1)^\top$, the configuration $(2,1)$ was most frequent ($p_{2,1} = \frac{3}{16}$).

- *Contrast*

  $$\sum_{i,j} (i-j)^2 \, p_{i,j}$$

  small, if similar grey values in the $\boldsymbol{d}$-neighbourhood are more frequent than dissimilar ones

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

- *Homogeneity*

  $$\sum_{i,j} \frac{p_{i,j}}{1 + |i-j|}.$$

  large, if many similar grey values exist in the $\boldsymbol{d}$-neigbourhood

- *Entropy*

  $$-\sum_{i,j} p_{i,j} \, \log \, p_{i,j}$$

  measures minimally required code length (cf. Lecture 8);
  maximal, if all possible configurations are equally probable;
  (thus, it also characterises the randomness)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Possibility 3: Texture Analysis in the Fourier Domain

- well-suited for (almost) periodic textures
- goal: characterise rich information in Fourier domain with a few parameters
- Consider Fourier spectrum of textured image f in polar coordinates: $|\hat{f}(r,\phi)|$
- By integrating in circular direction,

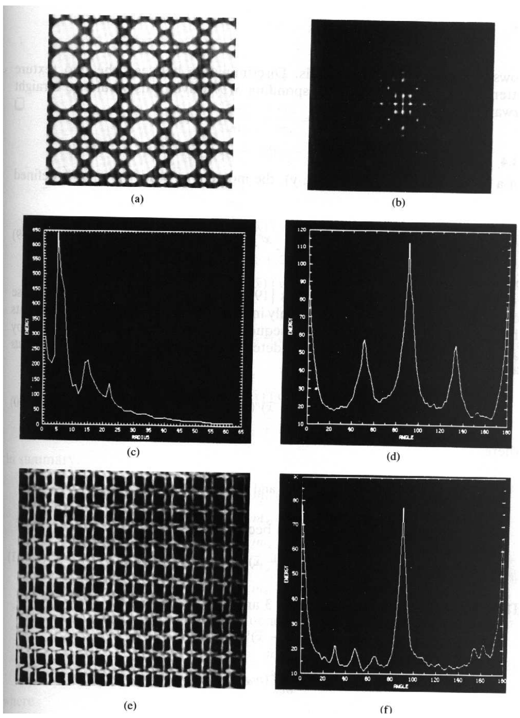
$$g(r) := \int_0^{2\pi} |\hat{f}(r,\phi)|\, d\phi\,,$$

  and in radial direction,

$$h(\phi) := \int_0^{R} |\hat{f}(r,\phi)|\, dr\,,$$

  one obtains two functions in one variable: $g(r)$ and $h(\phi)$.
  They characterise the radial and the angular behaviour in the Fourier spectrum.
- By computing simple expressions (e.g. maximum, mean, variance) for these functions, many textures can be discriminated.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33



(a) Image with periodic texture. (b) Fourier spectrum. (c) Plot of $g(r)$. It is less conclusive. (d) Plot of $h(\phi)$. One sees structures at 0, 45, 90, and 135 degrees. (e) Another textured image. (f) Its $h(\phi)$ plot. Here one cannot find characteristic diagonal structures. Authors: R. C. Gonzalez, R. E. Woods.

Texture Inpainting

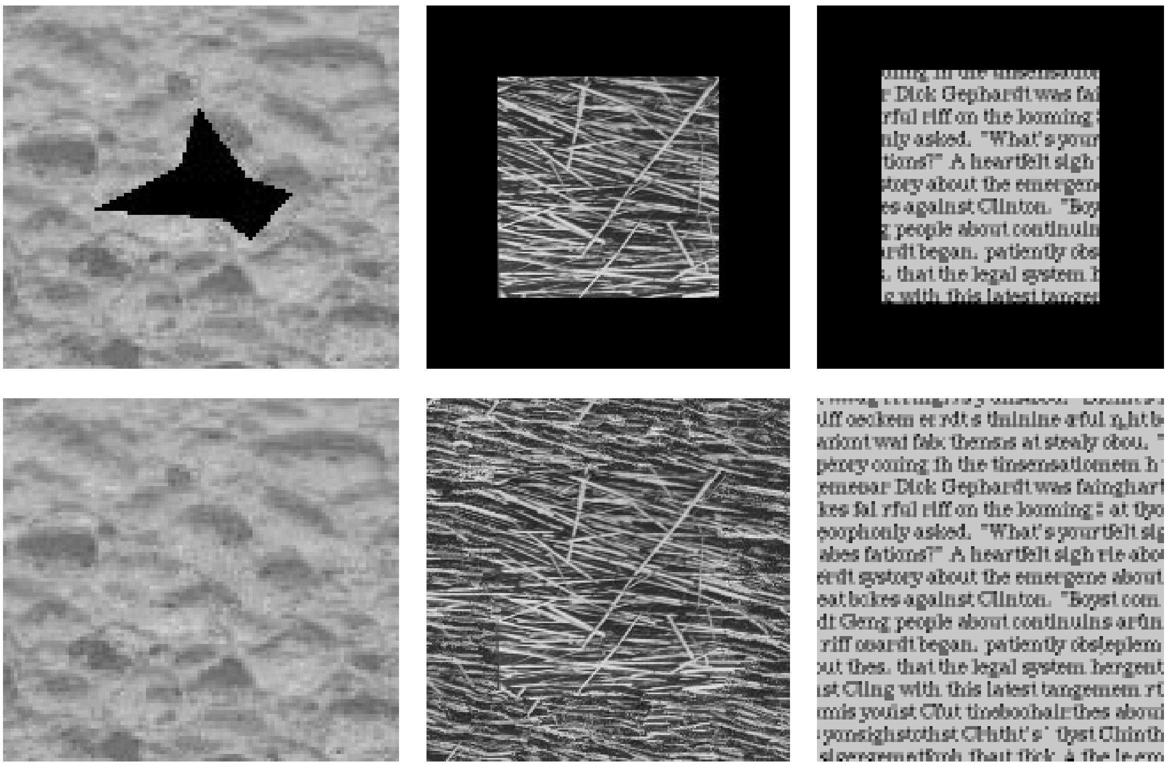
Problem

- In Lecture 18, we have inpainted missing image structures by solving the Laplace equation $\Delta u = 0$.
- This restores smooth regions, but does not work well for inpainting textures.

Remedy: Exemplar-Based Inpainting (Efros / Leung 1999)

- clever copy and paste
- proceeds pixelwise from known region into the inpainting area
- searches for the most similar patch in the known region
- copies only the central pixel in the patch
- close in spirit to NL means (Lecture 15)

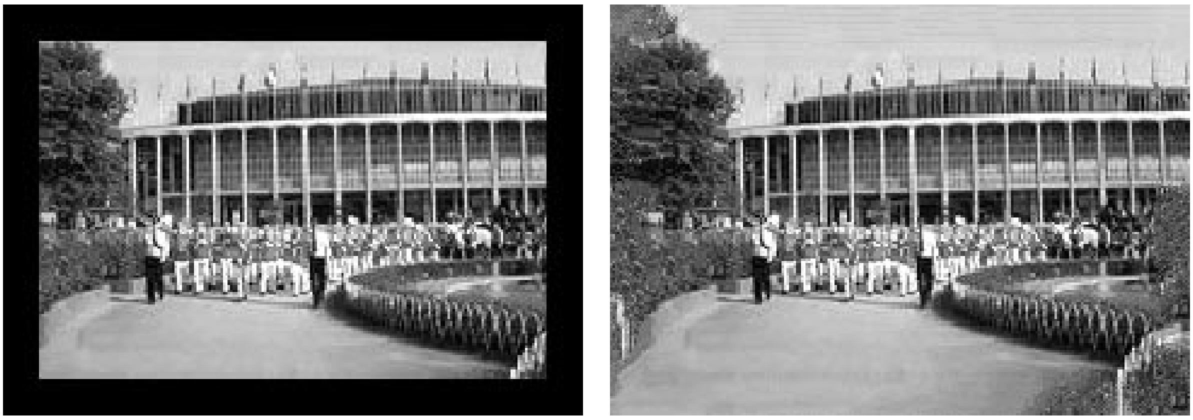
M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33



Top: Original images, where texture is missing in the black areas. **Bottom:** After exemplar-based inpainting. Authors: A. Efros, T. Leung.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Texture Inpainting (3)



Left: Original image. **Right:** Exemplar-based texture inpainting allows to extend the image in a surprisingly realistic way. Authors: A. Efros, T. Leung.

Texture Inpainting (4)

Left: Original image. **Middle:** The inpainting region is selected manually. **Right:** After exemplar-based inpainting. Source: `http://www.cc.gatech.edu/~sooraj/inpainting/`.

Texture Inpainting (5)

Exemplar-based Inpainting of Highly Sparsified Textures

original image

optimised data, 5 %

inpainted

There are variants of exemplar-based inpainting that can also handle very sparse data (Facciolo et al. 2009). If these data are carefully optimised, reconstructions of high quality are possible. This is used in our research on inpainting-based compression. Authors: L. Karos, P. Bheed, P. Peter, J. Weickert.

Summary

Summary

- Texture characterises reoccurring patterns, often with stochastic fluctuations.
- There is no fundamental theory for analysing textures.
  Heuristically motivated strategies dominate the field.
- First-order statistics (e.g. moments) may not be sufficient:
  Neighbourhood structures are important in texture analysis.
- Simple remedy: embedding into a multiscale representation
- Cooccurrence matrices are grey value histograms of neighbourhood structures.
  They can be condensed to useful texture descriptors.
- For periodic structures, texture descriptors in the Fourier domain can be useful.
- Exemplar-based inpainting performs pixelwise copy and paste into the inpainting region by comparing texture patches over the entire image.

References

- R. Jain, R. Kasturi, B. G. Schunck: *Machine Vision*, Mc Graw-Hill, New York, 1995.
  (Chapter 7)
- R. C. Gonzalez, R. E. Woods: *Digital Image Processing*, Pearson, Upper Saddle River, Third Edition, 2008.
  (Section 11.3.3)
- W. K. Pratt: *Digital Image Processing: PIKS Inside*. Wiley, New York, Third Edition, 2001.
  (Section 16.4–16.6)
- R. M. Haralick, K. Shanmugam, and I. Dinstein: Textural features for image classification. *IEEE Transactions on Systems Man and Cybernetics*, Vol. 3 No. 6, pp. 610–621, Nov. 1973.
  (introduced cooccurence matrices)
- M. Tuceyran, A. K. Jain: Texture analysis. In C. H. Chen, L. F. Pau, P. S. P. Wang (Eds.): *The Handbook of Pattern Recognition and Computer Vision*. Second Edition, pp. 204–248, World Scientifc, Singapore, 1998.
  (http://www.cs.iupui.edu/~tuceryan/research/ComputerVision/texture-review.pdf)
  (survey chapter)
- V. Randle, O. Engler: *Introduction to Texture Analysis: Macrotexture, Microtexture and Orientation Mapping*. CRC Press, Boca Raton, 2000.
  (entire monograph on texture analysis)
- A. A. Efros, T. K. Leung: Texture synthesis by non-parametric sampling. *Proc. Seventh International Conference on Computer Vision* (Corfu, Sept. 20–25, 1999), pages 1033–1038.
  (texture inpainting)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

- G. Facciolo, P. Arias, V. Caselles, G. Sapiro: Exemplar-based interpolation of sparsely sampled images. In D. Cremers, Y. Boykov, A. Blake, F.R. Schmidt (Eds.): *Energy Minimisation Methods in Computer Vision and Pattern Recognition*. Springer LNCS Vol. 5681, pp. 331–344, 2009.
  (sparse exemplar-based inpainting)
- L. Karos, P. Bheed, P. Peter, J. Weickert: Optimising data for exemplar-based inpainting. In J. Blanc-Talon, D. Helbert, W. Philips, D. Popescu, P. Scheunders (Eds.): Advanced Concepts for Intelligent Vision Systems. Springer LNCS Vol. 11182, pp. 547.-558, 2018.
  (data optimisation for sparse exemplar-based inpainting)

Assignment C10 – Classroom Work

Problem 1 (Inverse Filtering)

Let $\boldsymbol{f}$ be a blurred 1-D signal, $\boldsymbol{u}$ be the unknown ideal signal, and $\boldsymbol{h}$ be a known shift-invariant convolution kernel. Consider deblurring with inverse filtering.

(a) Sketch the function that is multiplied with $\hat{f}$ in dependence of $|\hat{h}|$.

(b) Explain the behaviour of this deblurring method when h is a lowpass filter. You can use the results obtained in (a).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Problem 2 (Cooccurrence Matrices)

(a) Compute the cooccurrence matrix of the 5×5 image

3	0	2	1	3
0	2	3	2	2
2	1	0	2	1
3	2	3	1	0
3	2	0	1	0

with $\boldsymbol{d} = (-1, -1)^\top$.
Assume that the x axis points to the right and the y-axis points downwards.

(b) Determine the highest probability as well as the contrast of the cooccurrence matrix.

(c) What happens if we use $\boldsymbol{d} = (1, 1)^\top$ instead?
Would it yield the same highest probability and contrast?

(d) How does the image have to look like, such that the contrast in the direction $\boldsymbol{d}$ is minimal?

(e) What is the dimension of the coocurrence matrix of a $m \times n$-image which grey values can be represented by 3 bits?

Assignment H10 – Homework

Problem 1 (Discrete Deconvolution) (3+3+2 points)

Let the following one dimensional discrete energy function be given:

$$E(\boldsymbol{u}) = \frac{1}{2}\sum_{i=1}^{N}((\boldsymbol{h} * \boldsymbol{u})_i - f_i)^2 + \frac{\alpha}{2}\sum_{i=1}^{N-1}\left(\frac{u_{i+1}-u_i}{h}\right)^2$$

where $\boldsymbol{f} \in \mathbb{R}^N$ is the blurred initial signal, and where the blurring kernel $\boldsymbol{h}$ is given by the binomial filter $\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)$.

Assume that the signal is mirrored at the boundaries.

(a) Compute the linear system of equations that a minimiser of $E(\boldsymbol{u})$ necessarily has to satisfy.

(b) Show that the corresponding system matrix is symmetric. What is the maximal number of nonvanishing entries in the rows of this matrix?

(c) How many nonvanishing entries per row would you expect for a general blurring kernel of size $2m+1$ with $m \geq 1$?

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Problem 2 (Deconvolution Methods) (2+2 points)

Let $\boldsymbol{f}$ be a blurred 1-D signal, $\boldsymbol{u}$ be the unknown ideal signal, and $\boldsymbol{h}$ be a known shift-invariant convolution kernel. Consider pseudoinverse filtering and Wiener filtering.

(a) Sketch the function that is multiplied with $\hat{f}$ in dependence of $|\hat{h}|$ for both deblurring methods. Do not forget to include ϵ or K in the corresponding drawings.

(b) Explain the behaviour of the two deblurring methods when h is a lowpass filter. You can use the results obtained in (a). What is the meaning of ϵ and K, respectively? How should the values for these parameters be chosen? Justify your answers.

Assignment H10 (2)

Please download the required files from the webpage

`http://www.mia.uni-saarland.de/Teaching/ipcv19.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex10.tar.gz`.

Problem 3 (Deconvolution with Wiener Filtering) (3+1+1+2 points)

The file `deconv.c` contains an almost complete deblurring program based on manipulations in the Fourier domain. It only requires to specify the deblurring function in the subroutine `deconv-filter`.

(a) Supplement the missing code for Wiener filtering. However, do not use C/C++ libraries such as `complex.h` that offer computations with complex data types. You can compile the program via
`gcc -O2 -o deconv deconv.c -lm` .

(b) The image `bus1.pgm` has been blurred with the small Gaussian `k-gauss1.pgm`. Use the compiled program to perform a deblurring. What is a good value for K?

(c) Do the same experiment with the image `bus2.pgm` and its corresponding blurring kernel `k-gauss2.pgm`. What is a good value of K in this case?

(d) Finally, consider the image `bridge-blur.pgm` which is a digital photo with simulated motion blur corresponding to the kernel `k-motion.pgm`. Try to compensate for the motion blur by finding a suitable value for K. Why does deblurring take so long in the case of this image?

Assignment H10 (4)

Problem 4 (Texture Inpainting) (2+3 points)

The program `texture.c` provides a full implementation of a texture inpainting algorithm in the spirit of exemplar-based inpainting. You can compile the program via
`gcc -O2 -o texture texture.c -lm` .

(a) Apply texture inpainting to `circles.pgm` with the mask `circlemask.pgm` and patch sizes $m = 2, 3, 6, 8, 10$. Describe the influence of m on the inpainting result.

(b) Create extensions of the images `stone.pgm`, `cyrus.pgm` and `boat.pgm` with the corresponding masks. (They do not have to be perfect.) Try at least two choices for the patch size m per image.

Assignment H10 (5)

Submission

Please submit the theoretical Problems 1 and 2 in handwritten form before the lecture. For the practical Problems 3 and 4 submit the files as follows: Rename the main directory `Ex10` to Ex10_<your_name> and use the command

```
tar czvf Ex10_<your_name>.tar.gz Ex10_<your_name>
```

to pack the data. The directory that you pack and submit should contain the following files:

- the source code for the Wiener filter in `deconv-filter.c`,
- the three deblurred images from Problems 3(b), (c), and (d),
- The six extended images from Problem 4(b),
- and a text file README that
  - answers the questions from 3(d) and 4(a),
  - states the used values for the parameter K in the Problems 3(b), (c), and (d), and
  - contains information on all people working together for this assignment.

Please make sure that only your final version of the programs and images are included.
Submit the file via e-mail to your tutor via the address:

```
ipcv-xx@mia.uni-saarland.de
```

where `xx` is either `t1`, `t2`, `t3`, `t4`, `t5`, `w1`, `w2`, `w3` or `w4` depending on your tutorial group.

Deadline for submission: Friday, June 21, 10 am (before the lecture)