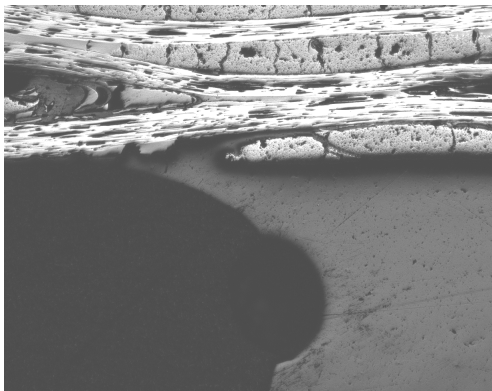


Segmentace textury

Případové studie

Jan Kybic

Zadání



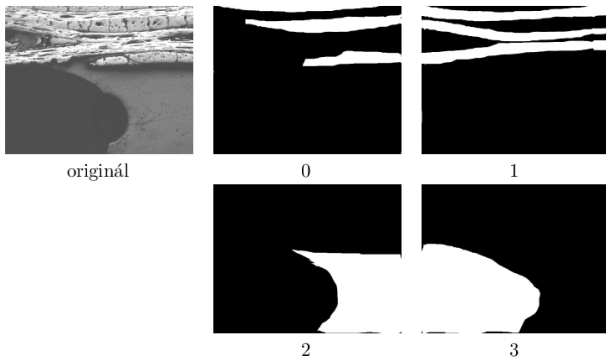
Mikroskopický obrázek segmentujte do tříd:

- ▶ Příčná vlákna
- ▶ Podélná vlákna
- ▶ Matrice
- ▶ Trhliny

Zvolená metoda

- ▶ Deskriptorový popis
- ▶ Učení s učitelem
- ▶ ML klasifikátor

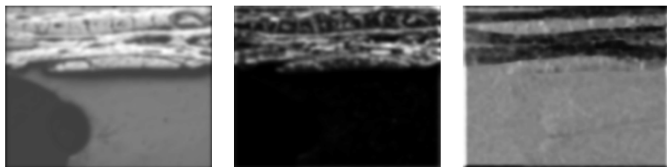
Masky tříd



Obrázek 1: Shora a zleva: Původní obrázek a masky pro třídy 0,1,2,3.

- ▶ Ground truth
- ▶ Ruční segmentace

Deskriptory



Obrázek 2: Deskriptory. Zleva hodnota šedé, lokální energie, směrnost

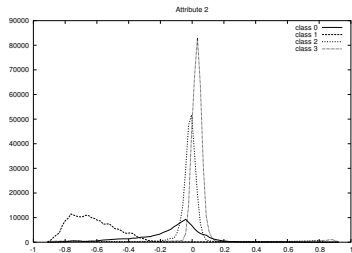
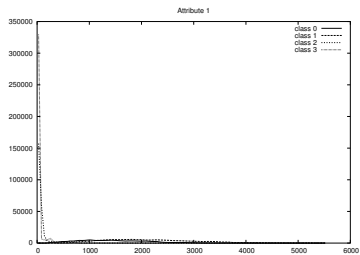
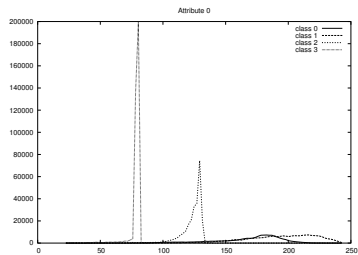
- ▶ $P1$: Hodnota šedé $\mu = f * G_\sigma$ $\sigma = 15$
- ▶ $P2$: Lokální energie $E = (f - \mu)^2 * G_\sigma$
- ▶ $P3$: Směrnost

$$\frac{E_h - E_v}{E_h + E_v} \quad E_h = (\partial_x f)^2 * G_\sigma$$
$$E_v = (\partial_y f)^2 * G_\sigma$$

Deskriptory (2) — motivace

- ▶ $P1$ (hodnota šedé): odliší matice od trhlin
- ▶ $P2$ (lokální energie): odliší vlákna od ostatních tříd
- ▶ $P3$ (směrnost): odliší orientace vláken

Deskripty (3) — histogramy



Statistický popis

- ▶ Pravděpodobnost $p(\mathbf{x}|T_i)$ vektoru deskriptorů v závislosti na třídě považujeme za normální.
- ▶ Složky vektoru deskriptorů považujeme za nezávislé
- ▶ \implies jednoduchý popis průměrem $\bar{\mathbf{x}}_i$ a standardní odchylkou σ_i

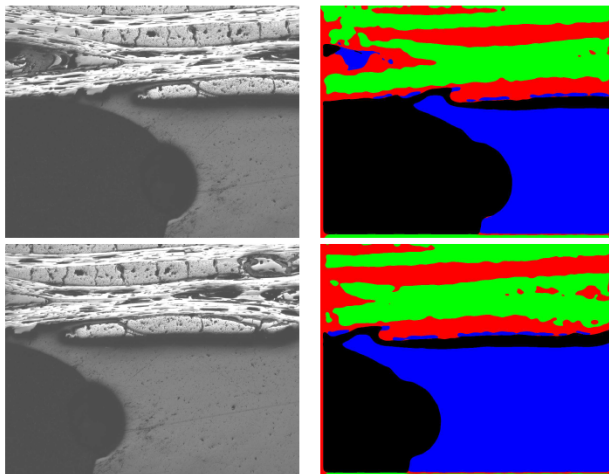
Klasifikace

Předpokládáme stejnou apriorní pravděpodobnost tříd $p(T_j)$. Pak je ML klasifikace neznámého pixelu s deskriptorovým vektorem \mathbf{x} vyjádřitelná jako:

$$j_{\text{ML}} = \arg \min_j \|(\mathbf{x} - \bar{\mathbf{x}}_j) / \sigma_j\|$$

kde operace $/$ je brána po elementech.

Výsledky



Obrázek 4: Obrázek (vlevo) a jeho klasifikace (vpravo). Horní obrázek byl použit při trénování, spodní nikoli. Třídám 0,1,2,3 odpovídá červená, zelená, modrá a černá barva.

Možná vylepšení

- ▶ Větší trénovací množina.
- ▶ Větší počet deskriptorů a optimalizace jejich parametrů na základě výsledků klasifikace.
- ▶ Úpravy výpočtu deskriptorů, aby postačovalo menší okolí a výsledky nebyly ovlivněny kraji obrazu a okraji textur.
- ▶ Zavedení třídy pro *nevím*.
- ▶ Lepší modelování sdružených pravděpodobností, opuštění požadavku nezávislosti a normálnosti. Obecnější návrh klasifikátoru, neparametrické klasifikátory.
- ▶ Použití apriorní informace ze sousedství pixelů pomocí např. markovských náhodných polí.
- ▶ Iterativní segmentace, kdy výpočet deskriptorů je řízen výsledky klasifikace z předchozí iterace.

Texture classification

using Haralick descriptors

- ▶ **Task:** for a texture patch, find a class
- ▶ Training and test images
- ▶ Calculate descriptors (features) for a patch
- ▶ Feature selection
- ▶ Classification

Co-occurrence matrices

$$P_{0^\circ,d}(a,b) = \left| \left\{ [(k,l), (m,n)] \in D : \right. \right. \\ \left. \left. k - m = 0, |l - n| = d, f(k,l) = a, f(m,n) = b \right\} \right|$$

$$P_{45^\circ,d}(a,b) = \left| \left\{ [(k,l), (m,n)] \in D : \right. \right. \\ \left. \left. (k - m = d, l - n = -d) \vee (k - m = -d, l - n = d), \right. \right. \\ \left. \left. f(k,l) = a, f(m,n) = b \right\} \right|$$

$$P_{90^\circ,d}(a,b) = \left| \left\{ [(k,l), (m,n)] \in D : \right. \right. \\ \left. \left. |k - m| = d, l - n = 0, f(k,l) = a, f(m,n) = b \right\} \right|$$

$$P_{135^\circ,d}(a,b) = \left| \left\{ [(k,l), (m,n)] \in D : \right. \right. \\ \left. \left. (k - m = d, l - n = d) \vee (k - m = -d, l - n = -d), \right. \right. \\ \left. \left. f(k,l) = a, f(m,n) = b \right\} \right| ,$$

Haralick descriptors

- ▶ **Energy**

$$\sum_{a,b} P_{\phi,d}^2(a,b).$$

- ▶ **Entropy**

$$\sum_{a,b} P_{\phi,d}(a,b) \log_2 P_{\phi,d}(a,b).$$

- ▶ **Contrast** (typically $\kappa = 2, \lambda = 1$)

$$\sum_{a,b} |a - b|^\kappa P_{\phi,d}^\lambda(a,b).$$

- ▶ **Inverse difference moment (homogeneity)**

$$\sum_{a,b;a \neq b} \frac{P_{\phi,d}^\lambda(a,b)}{|a - b|^\kappa}.$$

- ▶ **Correlation**

$$\frac{\sum_{a,b} [(ab)P_{\phi,d}(a,b)] - \mu_x \mu_y}{\sigma_x \sigma_y},$$

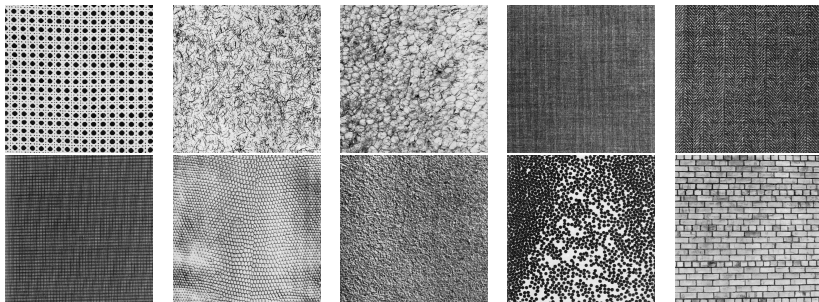
where μ_x, μ_y are means and σ_x, σ_y are standard deviations

Preprocessing and classification

- ▶ Distances $d \in \{1 \dots 10\}$
- ▶ Angles $\phi \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ (consider symmetry)
- ▶ 5 Haralick descriptors
- ▶ \rightarrow 202 features ($d = 0$ is treated specially)
- ▶ Normalized to zero mean and unit standard deviation
- ▶ Select 10 best features (ratio of intra-class / inter-class variance)
- ▶ Find Gaussian parameters (μ, Σ) for each class
- ▶ Maximum probability normal classifier

Brodatz texture

10 classes



Cut into 36 non-overlapping patches 100×100 pixels.
Randomly divide into training & test sets.

Results — Confusion matrix

Row i , column j — number of patches from class j classified as i .

	1	2	3	4	5	6	7	8	9	10
1	21	0	0	0	0	0	0	0	2	0
2	0	16	0	0	0	0	4	0	0	0
3	0	0	11	0	0	0	0	0	0	0
4	0	0	0	18	0	0	0	0	0	0
5	0	0	0	0	12	0	0	0	0	0
6	0	0	0	0	0	15	0	0	0	0
7	0	1	0	0	0	0	16	0	0	0
8	0	0	0	0	0	0	0	22	0	0
9	0	0	0	0	0	0	2	0	17	0
10	0	0	4	0	0	0	0	0	0	19

Total classification accuracy 94%.

Texture classification

using Wavelet descriptors

- ▶ Non-decimated wavelet coefficients (Unser, 1989), discrete wavelet frame (DWF)
- ▶ Descriptors = Wavelet energy signature

$$f = [\|s_L\|^2, \|d_L\|^2, \|d_{L-1}\|^2, \dots, \|d_1\|^2]$$

$$s_i(l) = 2^{-i} h(2^{-i} l) * x(l)$$

$$d_i(l) = 2^{-i} g(2^{-i} l) * x(l)$$

- ▶ 10 descriptors, feature selection not needed

Calculating wavelet frames

- ▶ Two-scale relation \rightarrow recursive calculation (fast)

$$s_{i+1}(k) = [h]_{\uparrow 2^i} * s_i(k)$$

$$d_{i+1}(k) = [g]_{\uparrow 2^i} * s_i(k)$$

- ▶ Haar filters

$$H(z) = (1 + z)/2 \qquad h = \frac{1}{2}[1 \ 1]$$

$$G(z) = (z - 1)/2 \qquad g = \frac{1}{2}[1 \ -1]$$

- ▶ 2D extension
 - ▶ At each level, create 4 sub-bands by filtering with $H_x H_y$, $H_x G_y$, $G_x H_y$, $G_x G_y$.
 - ▶ Separability \rightarrow only 6 1D filterings
 - ▶ Low-pass band $H_x H_y$ used as input to the next level

Results — Confusion matrix

Row i , column j — number of patches from class j classified as i .

	1	2	3	4	5	6	7	8	9	10
1	20	0	0	0	0	0	0	0	1	0
2	0	17	0	0	0	0	0	0	0	0
3	0	0	15	0	0	0	0	0	0	0
4	0	0	0	18	0	0	0	0	0	0
5	0	0	0	0	12	0	0	0	0	0
6	0	0	0	0	0	15	0	0	0	0
7	0	0	0	0	0	0	22	0	0	0
8	0	0	0	0	0	0	0	22	3	0
9	1	0	0	0	0	0	0	0	15	0
10	0	0	0	0	0	0	0	0	0	19

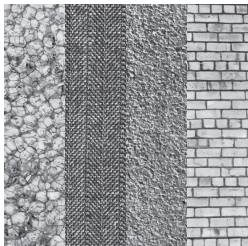
Total classification accuracy 97%.

Texture segmentation

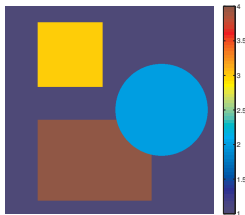
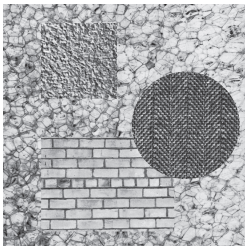
using Wavelet descriptors

- ▶ **Task:** segment image into regions based on texture
- ▶ Wavelet signature over a small window → descriptor vector for each pixel
- ▶ Training image → pdf for each class
- ▶ Segmentation algorithm (region growing, GraphCut)
— most likely class + spatial smoothness

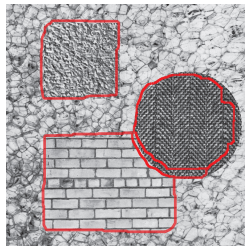
Example



training image



test image



result