Texture-Based Segmentation

Internship at the project team PULSAR

A. Schnaars[1]

[1]University of Magdeburg
Supervisor: Guillaume Charpiat PhD
Pulsar - INRIA Sophia-Antipolis

INRIA Sophia-Antipolis, 15th march 2010

Outline

1. Motivation
2. Method
   - Creating a Texture Classifier
   - Creating an Edge Classifier
   - Global Optimization
3. Results
4. Future Work

Definition

No unique definition of what a texture is:

Human Vision System[3]

- Information of texture to distinguish between things where edge of objects in the enviroment are not defined by clear boundaries
- Large number of symbols or simple shapes -> individual objects
- V1 neurons in the primary visual cortex: Texture element orientation, size, contrast and color

→Consisting of repetition or quasi repetition of some fundamental image elements

State of the Art & Goal

- There is no “optimal” texture feature
- Structrual, statistical, model-based approach
- SIFT and SURF most widly used
- Sparsly application in REAL-time processing like video-analysis

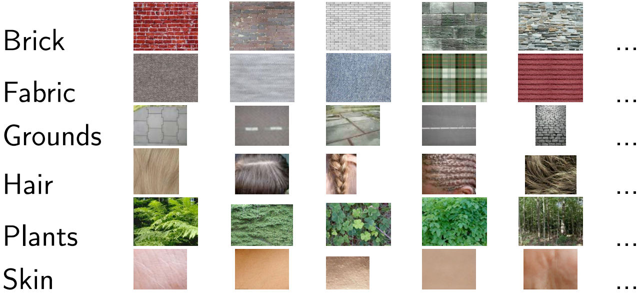
Goal

- Combination of several informations (scale, boundaries, ...)
- Fast classification
- Robust classifier
- Basis for object segmentation

Outline

Ground Truth

- Problem: Mostly segmentation of objects but not of single texture regions and no annotations information
- Database: Texture patterns from free texture libraries $\rightarrow$17 different texture classes, each pattern in 3 different resolutions



Texture Descriptor

Requirements:

- Fast processing
- Scale invariance
- Introduction of color information
- Illumination invariance

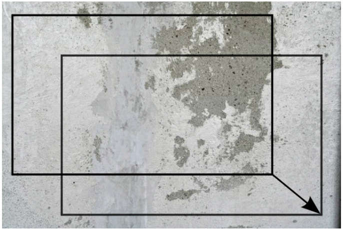
Integrative method: Combination of color and texture information

- CIE L*a*b* color space
- Unser Sum- and Difference-Histogram Features

Unser Sum- and Difference-Histogram Features

Texture Descriptor

- Approximation for the two-dimensional Haralick texture features on the co-occurrence matrix
- Frequencies of sums, respectively differences of pixel color levels with a certain displacement (d_x, d_y) within a region of interest D



Unser Sum- and Difference-Histogram Features

Texture Descriptor

$$
\begin{aligned}
h_s^c(i) &= Card\left\{(x_1, y_1) \in D^c \middle| g_{x_1,y_1} + g_{x_2,y_2}\right| = i\right\}, i \in [0; 2(G^c - 1)],\\
h_d^c(j) &= Card\left\{(x_1, y_1) \in D^c \middle| g_{x_1,y_1} - g_{x_2,y_2}\right| = j\right\},\\
&\quad j \in [-G^c + 1; G^c - 1], c \in L * a * b*\\
h_{s/d}^3(i) &= \{h_{s/d}^L(i), h_{s/d}^a(i), h_{s/d}^b(i)\}
\end{aligned}
$$

Unser Sum- and Difference-Histogram Features

Texture Descriptor

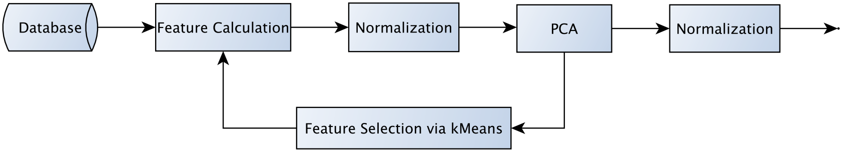
Features

- Sum/ difference mean: $f_{cj} = \frac{1}{N} \sum_i i \, h^c_{s/d}(i)$
- Sum/ difference contrast: $f_{cj} = \frac{1}{N} \sum_i (i - \mu)^2 \, h^c_{s/d}(i)$
- Sum/ difference angular second momentum: $f_i = \sum_i \left[h^3_{s/d}(i) \right]^2$
- Sum/ difference entropy: $f_j = \sum_i -h^3_{s/d}(i) log(h^3_{s/d}(i))$
- Plus color values of each color channel

- Regions of interest: 5x5, 7x7, 9x9
- 8 directions x displacement 1, 2, 4

$\approx$1000 Features

Feature Selection and Reduction



- Normalisation via a sigmodial function to keep outliers

$$y' = \frac{y - \mu}{std} \qquad y'' = \frac{1 - e^{-y'}}{1 + e^{-y'}}$$

PCA
Feature Selection and Reduction

- Assumption: Correlated features
- PCA: Uncorrelating data by mapping the data into a most significant subspace
- Goal: Feature selection through analysis of the most important principle components
  $\rightarrow$Features with same loadings have the same character

Feature selection (as defined in [2])

- kMeans on the reduced principle components
- Choosing feature as representants whose loadings have the smallest distance to a cluster's center

Interim Results

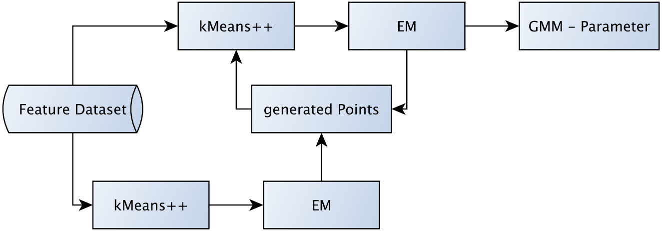
Clustering of the Reduced Texture Descriptor

Problems at:

- Boundaries
- Coarse textures
- Subtle color shifts (small value differences)

Textures Classes Training

- Assumption: Each class could be described by a mixture of gaussians
- Problem: Big amount of data
  →Online learning of the parameters



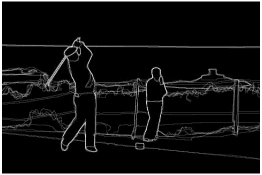
$$P_{Texture}(x) = \sum_k p(x|\mu_k, \theta_k)\pi_k$$

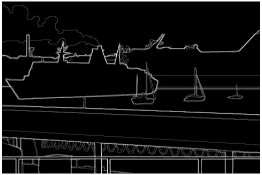
Outline

1. Motivation
2. Method
   - Creating a Texture Classifier
   - Creating an Edge Classifier
   - Global Optimization
3. Results
4. Future Work

Ground Truth

Berkeley Segmentation Dataset [1]: Hand-labeled segmentations from 30 human subjects, public benchmark contains 300 images and its segmentations

Edge Descriptor

For each channel of the CIE L*a*b* color space:
Sobel Operator

- First and second derivatives
- Blocksizes 1, 2, 3, 5, 7
- Magnitude and orientation

Edge Descriptor

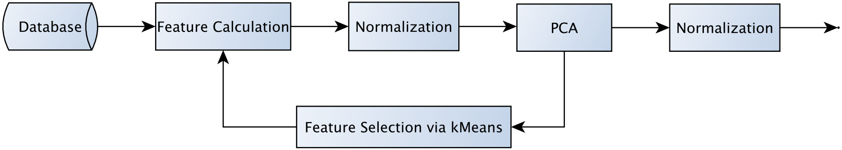
Autocorrelation Matrix

$$A(x) = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2(x,y) & I_x I_y(x,y) \\ I_x I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

- Block sizes of 3, 9, 23
- The biggest Eigenvalue

Plus color values
≈70 Features

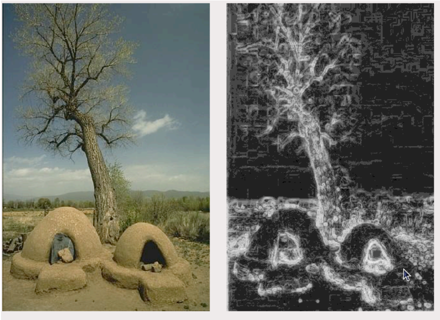
Feature Selection and Reduction



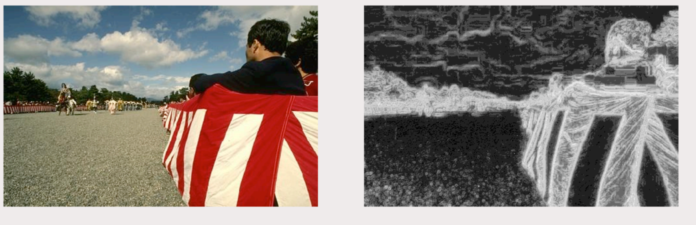
Edge Classifier Training

- Assumption: Edges can be very noisy
- Classifier that seperates boundaries clearly
  $\rightarrow$ SVM
- Issue: Big amount of data
  $\rightarrow$Online learning of the parameters : Neural Network

Interim Results



Interim Results

- 
- Results are moderate compared to the effort
- Eventually overfit

Outline

1. Motivation
2. Method
    - Creating a Texture Classifier
    - Creating an Edge Classifier
    - Global Optimization
3. Results
4. Future Work

Combination of all learned informations in a (minimised) cost function:

Graph Cut

$$E(p) = \sum_{p \in I} L(p) + \alpha \sum_{p,q \in I} V(L(p), L(q))$$

$$L(p) = -log(P_{Texture}(p))$$

$$V(L(p), L(q)) = -log(min(P_{Edge}(q), P_{Edge}(p)))$$

Extension

- Measurement of the interactions between different texture classes
- Location of some texture

- Frequency of certain neighboring textures in certain directions

Problems:

- Annotation of the data, e.g. Berkeley tool only segments
- Optimisation tool (GraphCut): No different interactions between two classes in different directions

Further Applications

Object recognition

- Learning shape parameters of certain textures in certain objects
- Learning locations of objects
- Probability to find an object in an area based on its location and texture components

Thank you for your attention!

Questions?

For Further Reading I

Pablo Arbelaez, Charless Fowlkes, and David Martin.
The berkeley segmentation dataset and benchmark, 2007.

Yijuan Lu, Ira Cohen, Xiang Sean Zhou, and Qi Tian.
Feature selection using principal feature analysis.
In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 301–304, New York, NY, USA, 2007. ACM.

Colin Ware.
Visual Thinking for Design.
Morgan Kaufmann, July 2008.