Texture analysis

Team 5

Alexandra Bulgaru
Justyna Jastrzebska
Ulrich Leischner
Vjekoslav Levacic
Güray Tonguç

Contents

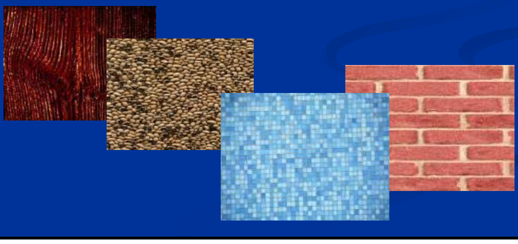
- Project goal
- Definition of texture
- Features used in texture analysis
- Example of application for texture based image query
- Results
- Conclusions

Project goal

- Defining a set of features which would help in identifying the textures in the image
- Examining the relation between features and the textures
- Defining a simple set of features to identify similar textures in texture database
- Possibility of using texture classification and segmentation in later applications

Definition of a texture

- Texture is used to describe two dimensional arrays of variation.
- The elements and rules of spacing or arrangement in texture may be arbitrarily manipulated, provided a characteristic repetitiveness remains.



Features used in texture analysis

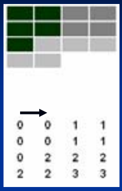
Problem of feature selection depends on:

- Type of application (medical, aerial, etc.)
- Need of invariances (rotational, shifting, scaling, lightning, etc.)

Examples of features we used:

- **Statistical** (for example derived from co-occurrence matrix like entropy, contrast, correlation)
- **High level** (derived from the watershed algorithm)
- **Frequency domain** (energy bands)

Co – occurrence matrix



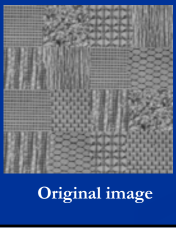
→

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

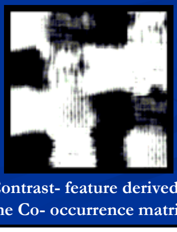
neighbour pixel value

Reference pixel value:

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



Original image



Contrast- feature derived from the Co- occurrence matrix

Calculation of the feature value

$$\begin{bmatrix} 0.16 & 0.08 & 0.04 & 0 \\ 0.08 & 0.16 & 0 & 0 \\ 0.04 & 0 & 0.24 & 0.04 \\ 0 & 0 & 0.04 & 0.08 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 4 & 9 \\ 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 \\ 9 & 4 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0.08 & 0.16 & 0 \\ 0.08 & 0 & 0 & 0 \\ 0.16 & 0 & 0 & 0.04 \\ 0 & 0 & 0.04 & 0 \end{bmatrix}$$

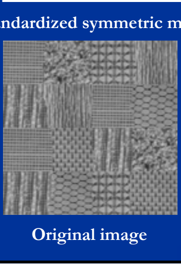
Standardized symmetric matrix

↓

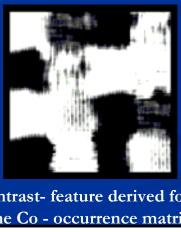
Sum of all cells

=

0.586



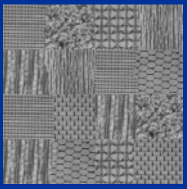
Original image



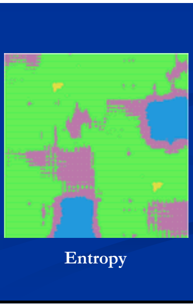
Contrast- feature derived form the Co - occurrence matrix

Statistical features - Entropy

$$Entropy = -\sum_i p_i \cdot \ln(p_i) \qquad p_i = \frac{\text{number of pixels with intensity value i}}{\text{total number of pixels}}$$



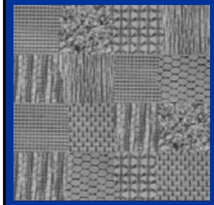
Original image



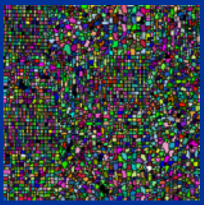
Entropy

Watershed segmentation

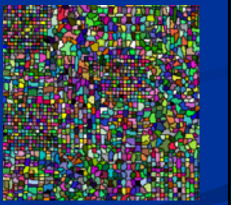
- Average area of components
- Number of components in specific region
- Ratio between circumference and components number



Original image

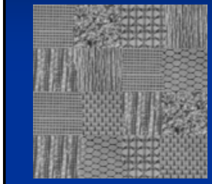


Watershed from original image

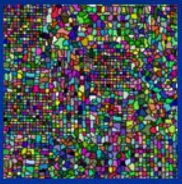


Watershed form the median filtered image (smoothing of nois to avoid oversegmentation)

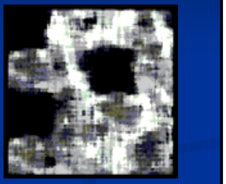
Watershed analysis – Average area of components



Original image

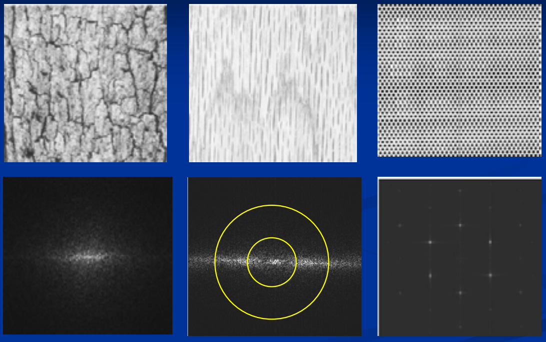


Watershed form the median filtered image



With a big filter size: Better features inside but borders are imprecise

Frequency domain feature

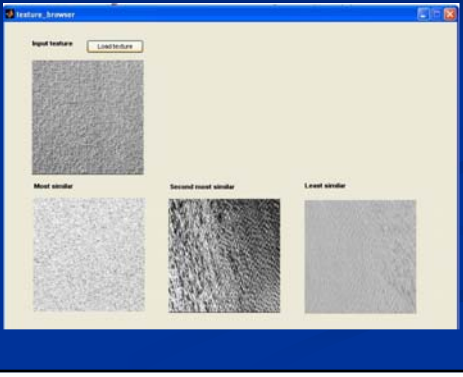


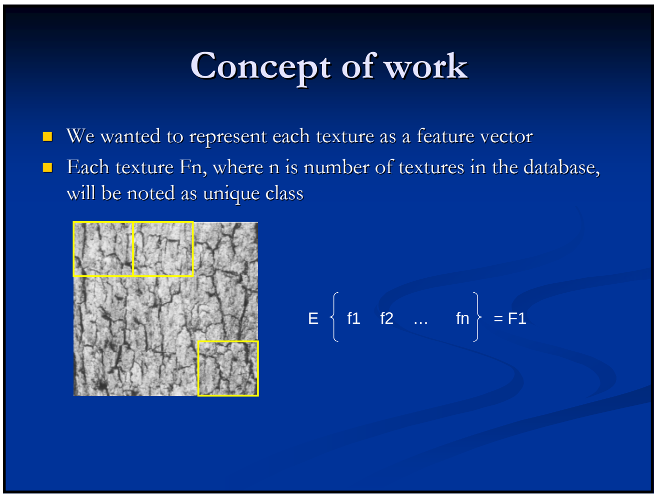
Low spectrum

Low to high spectrum

Periodicity of image

Example of application for texture based image query

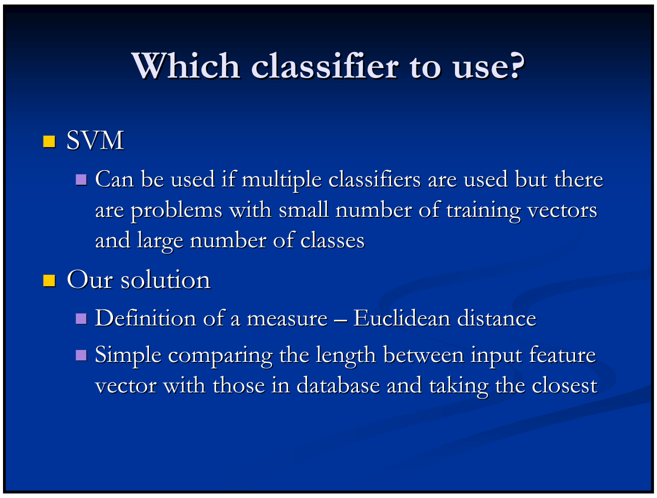




Concept of work

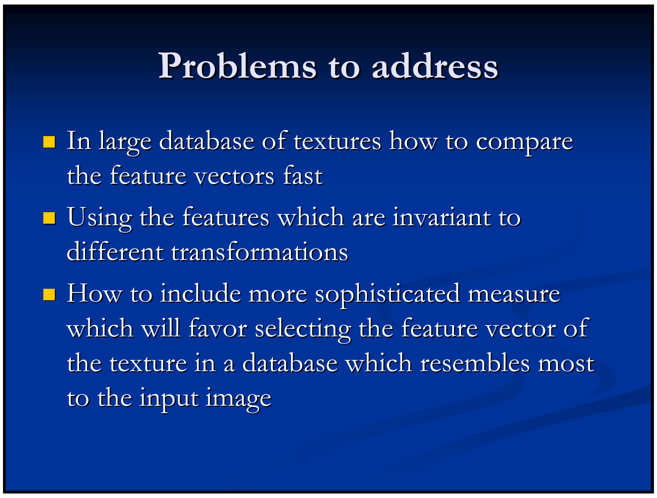
- We wanted to represent each texture as a feature vector
- Each texture Fn, where n is number of textures in the database, will be noted as unique class

E { f1 f2 … fn } = F1



Which classifier to use?

- SVM
  - Can be used if multiple classifiers are used but there are problems with small number of training vectors and large number of classes
- Our solution
  - Definition of a measure – Euclidean distance
  - Simple comparing the length between input feature vector with those in database and taking the closest



Problems to address

- In large database of textures how to compare the feature vectors fast
- Using the features which are invariant to different transformations
- How to include more sophisticated measure which will favor selecting the feature vector of the texture in a database which resembles most to the input image