

University of Konstanz
Data Analysis and Visualization Group
Prof. Dr. Daniel A. Keim
Michael Behrisch, Robert Gregor, Dominik Sacha

SS 14

Submission deadline:
So, 29.06.2012, 12pm.

Practical Assignment (90 + 35 points) *-- Image Retrieval --*

General Information:

- You have to form **groups of 2-3 persons**. Individual submissions are only accepted in special cases.
- You will (more) get detailed descriptions of the feature vectors and their intuition in the **exercise slot 11.06.2014**. It will be beneficial for you to be there.
- Due to technical difficulties we decided against iBrowse. You can come up with your own code. The tasks are adapted likewise.
- You may **only use standard Java functionality** in your code. Please write your own code for implementing the data structures and algorithms.
(Exception: You may use a Math library, such as Apache Commons Math, for statistical calculations)

The same accounts if you are using another programming language than Java (implement histograms, comparison functions, feature extraction by hand, but you are allowed to use statistical packages/libraries).

Remark: In case of some programming languages we might not be able to give support. Java is accepted in the Information Engineering course of studies

- Submission deadline: **So, 29.06.2012, 12pm.**

Description of the Data Sets:

1. TreeMap Data: 100 randomly created Treemap images in the png format.
2. Matrix Data: 700 sorted matrices of size 17x17 cells in the png format.

The two data sets are included in the Datasets.zip

Exercise Definition:

Data Import (0 Points - required)

- Import the test image data files into an appropriate color images data structure in main memory.¹

Color Conversion (5 Points)

- Write a routine for converting color images into their black/white/gray-value representatives (required for some of the feature extractors)

Feature Extraction (10 + 10 + 20 Points)

- Implement at least two of the following image descriptors. The image descriptors will be presented in the next exercise session (11.06.2014)
 - a. Color histogram (global, local). (10 Points)
(Parameters: number of bins, number of cells)
 - b. Global Edge histogram (10 Points)
(Convolute the image with Sobel-Operators before computing a histogram similar to the previous task²)
 - c. Texture Haralick Features (20 Points)
*(Paper is given Online in Ilias;
Implement the Gray-Tone Spatial Dependence Matrix and the
14 Global Textual Features given in the Appendix I; The Feature Vector is then
comprised of these 14 statistical measures)*

Similarity Functions (10 + 10 + 20 Points)

- Implement two of the following similarity functions:
 - a. Euclidian Distance (10 Points)
 - b. Cosine Distance (10 Points)
 - c. Quadratic Form Distance (20 Points)
(Hard coding of matrix values is accepted)

Evaluation (20 Points)

- Apply all your implemented descriptors on the two datasets.
Test at least five different input images as query objects on each of the two datasets.

Vary the parameters of the feature extraction methods (bins for the histogram; number of the local cells, amount of gray values) and compare the results.

Evaluate the usefulness of the feature vectors. Show some findings and reason on the advantages and disadvantages of the approaches. Document your results visually (make screenshots or show the rankings by inserting the respective images in a

¹ <http://docs.oracle.com/javase/tutorial/2d/images/index.html>

² Use the built-in Java `ConvolveOp` class to compute the Convolution.

ranking order).

Surprise us (optional 20 Points)

- Try to improve the retrieval performance, this could e.g. be achieved by one or several of the following:
 - a. By combining/weighting feature descriptors
 - b. By combining/weighting distance functions
 - c. By implementing a more advanced image feature descriptor³
 - d. By implementing a multi-scale approach that takes into account differently scaled version of the input image to compute the descriptors.
- Develop your own ideas and be able to justify them.

Deliverables:

1. **ZIP File of your Eclipse/Netbeans project source code**
2. **A PDF describing the content, start procedure/parameters, output**
3. **A PDF for the evaluation with screenshots/images from your program or output**

The filenames must incorporate your names (i.e. behrisch.jar, behrisch.zip)

Send the files to mmdbsl4@dbvis.inf.uni-konstanz.de

³ e.g. similar in style to one of the global MPEG7 image descriptors discussed in: <http://www.cacs.louisiana.edu/~cice/cacs508/descriptors.pdf> ; On the other hand Gabor-Filter based texture descriptors or SIFT-like local descriptors are very likely to provide better results than the edge histogram or the Haralick features.