Introduction

Motivation
Mass-digitization of historic prints has recently become a popular issue for libraries. But modern OCR-programs often achieve poor recognition rates for historic prints. There may be two reasons for this:
1. Inadequate scan quality (resolution, compression)
2. Common algorithmic design (e.g. support vector machines)

Goal
A new OCR-concept specifically adjustable to out of the ordinary fonts.

Solution
Glyph classification as pattern matching with on-the-fly generated patterns.

Pattern Recognition vs. Pattern Matching

Pattern Recognition
- fuzzy glyph recognition based on supervised learning
- generally applicable but inflexible for unexpected input data
- limited accuracy due to the optimization of average cases
- "this is an R in Helvetica at 32pt"
- "this is an R in Alte Schwabacher at 45pt"

Pattern Matching
- algorithmically simple (e.g. counting of common pixels)
- works for arbitrary patterns without learning
- no conceptual limits for the accuracy
- drawbacks:
  - patterns have to be known to the software
  - approach depends on scan quality

Approach
- mass-digitization: many pages in the same fonts and layout
- operator (user) can specify these fonts and pt-sizes to the OCR-software, as usual in a desktop publishing print job
- rasterization engine generates glyph patterns on-the-fly
- font technology from desktop publishing (widely established and highly efficient)

Pattern Recognition
- "this is an R in Helvetica at 32pt"
- "this is an R in Alte Schwabacher at 45pt"

Pattern Matching
- comparison of a sample with a set of glyph patterns
- classification by highest similarity

Implementation

Data Structure: Run-Length-Encoding
Instead of storing every pixel, group them together to intervals. Store start and end points in interval lists.

Store additional information per segment:
- # pixels, # intervals, bounding box size and coordinates, center of gravity, etc.

Test Setup

2 testfile types: 1. lowercase letters & numbers, 2. uppercase letters
- Each file consists of 12 lines in different font sizes from 6 to 27 pt.
- is generated in 15 different fonts including 3 fraktur fonts
- and is scanned at 300 ppi and 600 ppi.

Results

Recognition rate by point size

Recognition rate (%) by font in 300 and 600 ppi

Font
- Times New Roman
- Times New Roman bold
- Myriad Pro
- Helvetica
- Gill Sans
- Verdana
- Stone Serif bold
- Breitkopf Fraktur
- Unger Fraktur
- 'Alte Schwabacher'
- 'Helveticat'
- Arial

Conclusion

With the integration of our glyph classification into an OCR-software, it seems possible to reach the usual 1-2% error rates of modern office applications also for out of the ordinary fonts. We have shown that our concept can be implemented with standard components from desktop publishing. The necessary user input - specification of fonts and font sizes - is identical as for print devices. However, the concept can still be improved in many ways.