Similarity Measures for Pattern Matching On-the-fly

Ursina Caluori and Klaus Simon

Swiss Federal Laboratories for Materials Testing and Research (EMPA)
Dübendorf, Switzerland

ABSTRACT

Recently, we presented a new OCR-concept [1] for historic prints. The core part is the glyph recognition based on pattern matching with patterns that are derived from computer font glyphs and are generated on-the-fly. The classification of a sample is organized as a search process for the most similar glyph pattern. In this paper, we investigate several similarity measures which are of vital importance for this concept.

Keywords: similarity measures, OCR, pattern matching, glyph recognition

1. INTRODUCTION

Modern software for *optical character recognition (OCR)* like *neural networks* or *support vector machines* [2, 3, 4, 5] follow the *machine learning* approach. The graphical appearance of a character is allowed to vary in a certain range where its common or typical shape has to be learned in a supervised training process. Its main advantage is generality, i.e. the software can give reasonable answers in many cases. On the other hand, the recognition accuracy depends on the used training set and, therefore, is subject to principal restrictions. Furthermore, the training process itself is demanding and has to be done by the software developer. Consequently, modern OCR-software supports especially complex situations like handwritings or highly varying office applications.

The old fashioned way of pattern recognition is *pattern matching* where a sample image is compared with a pattern pixel by pixel. If all or almost all pixels of the sample match a specific glyph pattern, then the sample is identified. This algorithmically simple method is highly accurate and works for arbitrary patterns without training or other preparations. The drawback is the involved image processing which overstrained the early computer technology such that pattern matching was replaced by more efficient concepts in the 70s.

In the meantime computer performance has improved by a factor of more than a 1000 both in running time and storage capacity. This can be used for a renewal of pattern matching in mass-digitizations of historic prints. Typical for such applications is a constant layout for many thousands of scans showing unusual metal fonts*.

The stability of the layout allows a professional operator to adjust the OCR-software to a specific task by assigning the involved fonts as input to the program. Please note that most of the metal fonts designed during the last 500 years also exist as computer font replications [6] and treating fonts as input parameters is common standard in desktop publishing. A modern raster image processor can generate glyph patterns from such a computer font very efficiently.

Recently, at the Archiving 2013 conference [1] we have shown that the described pattern matching is functional but only a small set of 15 fonts was considered and no special questions could be investigated. So we present here a new enlarged test with 112 fonts addressing the best possible similarity measure for this kind of pattern matching.

In chapter 2 we describe our concept in greater detail including several plausible similarity measures. Next, we consider the realized recognition test for glyphs. Finally, we analyze our results and draw conclusions.

2. MOTIVATION AND ALGORITHM

The usual result of a scanning process is a gray image which has to be binarized to 0 (black) for textual parts and 1 (white) for the background. Black pixels are called *dots* and adjacent dots are summarized into *segments*. A *pattern*—or more exactly, a *glyph pattern*—is formed by the segments found in a glyph image where a glyph is a specific graphical representation of a character. In our context, the considered glyphs are described as vector graphics and can be rendered to a glyph image with a raster image processor. The associated font, the corresponding character, the used pt-size and the induced bounding box are stored together with the glyph image in each pattern's representation. The set of currently considered fonts is assigned by the user.

* often black letter (fraktur) fonts

One or more[†] segments of the input file belong to a character. Such a group is denoted a *sample* and the main task of an OCR-system is the assignment of a fitting glyph to it. Segments are stored in a run length encoding both in horizontal and vertical directions. Accordingly, consecutive dots are gathered in distinct intervals which are organized in linear lists and sorted in relation to their endpoints. In each case, the used coordinates are relative to the corresponding bounding box.

Our classification of a sample is a search for the most similar glyph pattern. So we have to describe the comparison between a sample S and pattern P . First, we calculate the projection profiles of S and P where a *projection profile* is given by the number of pixels per line in horizontal or vertical direction. The best sample-pattern overlapping is determined by the maximum cross correlation between corresponding projection profiles. The horizontal (vertical) projection profile of S is denoted as X_S (Y_S) and X_P (Y_P) analog for P . The index i in $X_S(i)$ indicates the corresponding value in the i -th row or column, respectively, of the bounding box. In the same way we define the projection profiles for white pixels between the intervals $\bar{X}_S$, $\bar{Y}_S$, $\bar{X}_P$ and $\bar{Y}_P$. Then L_S (R_S , T_S , B_S) stand for left (right, top, bottom) outline of S where $L_S(i)$ means the number of white pixels between the left rim of the bounding box and the first dot in the opposite direction. For P analog notation is used.

Finally, we have to define the considered similarity measures $M_\ell = M_\ell(S, P)$, $\ell = 1, \dots, 20$. We are interested in algorithmically simple concepts matching our run length encoding. Especially, we are looking for robust versions which perform well for different scalings and noisy data. In the following, we assume without loss of generality that the overlapping between S and P is fixed and the relative coordinates for the row and column lists have been adapted to the common bounding box CBB . Accordingly, we have to compare the pairs of rows (A_i, B_i) , $i = 0, \dots, t$, and the pairs of columns (C_j, D_j) , $j = 0, \dots, k$, where $t \cdot k$ is the size of CBB and the interval endpoints in the lists of A_i, B_i, C_j, D_j are sorted in increasing order left-to-right or top-to-bottom. Then we define the following similarity measures.

- The measure traditionally associated with pattern matching is the percentage of common pixels denoted as M_{16} . Another obvious choice is the number of common dots CD . Two of the interval lists can be compared by a *merge step* as in the well-known *mergesort* algorithm, see for instance [7, p. 12–15], which can be done in a running time proportional to the number of involved interval endpoints. In order to get a number between 0 and 1 we define $M_1 = CD / \sqrt{|S| \cdot |P|}$ where $|S|$ ($|P|$) stands for the number of dots in S and P , respectively. In other words, M_1 is the cross correlation function when a pixel in S and P is considered as a vector with value 1 for a dot and 0 for a white pixel[‡].
- Let us consider the comparison of a pair (A_i, B_i) of interval lists in greater detail. The merge step divides the intervals into maximal sections belonging to both lists and those occurring only in one. Therefore, the intersection of all $A_i \cap B_i$, $i = 0, \dots, t$ induces a set of maximal common subintervals $\{D_1, \dots, D_d\}$ and the symmetrical difference $(A_i \setminus B_i) \cup (B_i \setminus A_i)$ leads to the corresponding interval set $\{U_1, \dots, U_u\}$. Then, the measure M_2 is given by

$$M_2 = \frac{\sum_{i=1}^d (2 \cdot |D_i|)^2}{\sum_{i=1}^d (2 \cdot |D_i|)^2 + \sum_{i=1}^u |U_i|^2} \quad (1)$$

where $|D_i|$ stands for the lengths of D_i . Please note that M_2 is related to row lists only. It is some kind of minimal square method. Longer sequences of common pixels are weighted higher. The third measure M_3 is the same as M_2 but related to the column lists. The next variant M_4 refers to both row and column lists. In M_5 we additionally consider maximal common sections of white pixels which are added to D -intervals.

- The measures M_6 and M_7 correspond to the values of the maximum cross correlation of projection profiles, M_6 for the x - and M_7 for the y -direction. The version M_8 is the weighted sum of last two where M_6 is multiplied with the standard deviation of X_S and M_7 analogously with the one of Y_S .
- The next measures are correlated with the differences in outlines and projection profiles. Please note that the outlines

[†]like in an “i”

[‡]i.e. with inverted pixel color

have to be related to *CBB* and not the bounding boxes of *S* or *P*. Then, M_9 is defined as:

$$M_9 = \frac{1}{t \cdot k} \sum_{i=0}^t \left(k - |X_S(i) - X_P(i)| - |\bar{X}_S(i) - \bar{X}_P(i)| - |L_S(i) - L_P(i)| - |R_S(i) - R_P(i)| \right) + \frac{1}{k \cdot t} \sum_{j=0}^k \left(t - |Y_S(j) - Y_P(j)| - |\bar{Y}_S(j) - \bar{Y}_P(j)| - |T_S(j) - T_P(j)| - |B_S(j) - B_T(j)| \right) \quad (2)$$

The measure M_{10} is similar to M_9 but only the outlines are considered. M_{11} is another variant of M_9 where the average percentage of differences in the involved outlines and profiles is replaced by their average contrast. The contrast of the real numbers x and y is defined as $|x - y|/(|x| + |y|)$.

- The average contrast between the number of intervals in the rows and columns is counted in M_{12} . In M_{13} , M_{14} and M_{15} we consider the average contrast between the expected interval lengths where the probability p_I of interval I is defined as the fraction of its length and the corresponding row or column lengths. M_{13} is related to dot intervals, M_{14} to intervals of white pixels and M_{15} considers dots again but with the inverse quadratic interval length as weighting function. The remaining measures M_{17} to M_{19} are combinations of others, at first $M_{17} = \alpha \cdot M_2 + \beta \cdot M_1$ where α is the standard deviation of Y_S and β the one of X_S . Similarly, we get $M_{18} = M_2 \cdot (0.9 + 0.1 \cdot M_1)$ and

$$M_{19} = ((M_2 \cdot (0.2 + 0.8 \cdot M_5)) \cdot (0.97 + 0.03 \cdot M_{13})) \cdot (0.99 + 0.01 \cdot M_{12}). \quad (3)$$

- Finally, in contrast to the others a majority measure M_{20} is defined. Thus, it classifies a sample not by a search process but as the character for which most of the four measures M_{10} , M_{16} , M_{18} and M_{19} vote for. These measures were chosen in a greedy empirical estimate in our preparations.

3. TEST OF THE GLYPH RECOGNITION

For each of the font sizes 6 pt, 7 pt, . . . , 12 pt we designed a synthetic test file (target) containing 112 text lines with the character combination a, b, . . . , z, 0, . . . , 9, A, . . . , Z. Each text line is typeset in one of our 112 considered fonts. The resulting pages are printed as “image” with a Xerox Phaser 7500 where the mode “image” ensures through the involved halftoning process that the following recognition task is a challenge. The printouts are scanned at 300 and 600 ppi with an Epson Perfection V700 Photo. Then, the scans are binarized with Niblack’s method [8]. Finally, we get 7 input images (6–12 pt) for 300 and 600 ppi each.

The test was carried out with a reduced version of our software. Our font database contains 112 fonts currently (see table 3 in appendix B for a list). However, the encoding differs a bit from font to font. For that reason, we only used the 184 characters that are common to all fonts. Hence, we compared $112 \times 62 = 6944$ samples with each of the total 20608 patterns in the search for the most similar glyph pattern. In order to speed up the comparison somewhat, we performed a rough size check of the involved bounding boxes. Figures 1(a) and 1(b) show an example of our visualizations where the black samples stand for correct and gray ones for wrong classification. Apart from the gray colorization, the visualization is exactly the same as the input file.

4. ANALYSIS

Our approach of searching for the most similar glyph pattern implies the necessity of a highly efficient implementation of the involved basic procedures, first of all the used similarity measures. For that reason, we consider only very simple similarity measures[§] which can immediately be derived from the used run length encoding. Consequently, all considered measures can be computed in a running time proportional to the number of the involved interval endpoints. Hence, our running time scales with the scan resolution and not with its square.

The starting point of our study is the classical understanding of pattern matching, namely the *percentage of common pixels* as similarity measure. Of course, for ideal data without noise and distortions this kind of pattern matching shows perfect results, in other words, with increasing scan quality and font sizes the hit rate will converge to 100 %. Therefore,

[§]in contrast to the similarity measures considered in [9]

the challenge for pattern matching arises from noisy data in connection with small font sizes. Accordingly, we carried out our performance test for the most popular font sizes in connection with a relatively high noise level. Our central question was: *Can we find a single measure which performs well in all circumstances or do we need specific solutions for different situations?*

Figure 1. Detection Visualization

Figure 2. Character hit rates by measure.

The main result can be found in figure 2 (see tables 1 and 2 in appendix A for exact numbers). For 300 ppi each of the best performing measures $M_1, M_2, M_4, M_{17}, \dots, M_{20}$ achieves a character hit rate over 99 % at 12 pt. Please note that the glyph recognition considered here is just a part of an OCR-software and an error rate of 1–2 % is widely accepted as quality standard for the entire OCR-process. Not unexpectedly, we see that error rates increase with decreasing font sizes. However, the monotony of this tendency for declining hit rates over all measures is rather surprising. The transitions from 8 to 7 pt and further down to 6 pt may be the most significant ones. At 8 pt each of the best performing measures holds at least a hit rate of 98 % but then lost more than 0.5 % on the way to 7 pt and additionally more than 2 % for 6 pt. This fact can also be observed for the scan resolution 600 ppi but in the reduced amount of 50 %.

A closer look at our input data explains the problem. In figure 3 some scanned glyphs from the font Parisian, our worst case, are shown. This font has a very small x-height[¶] in relation to its H-height^{||} and, additionally, extreme differences in the line thickness. As a consequence, the structural details of the letters begin to crumble, thin lines break into pieces or simply disappear, in other words, we are approaching the “frontier” of OCR-technology, see [5] for more.

[¶]height of a small x in a given font
^{||}height of the letter H in a given font used as standard height of capitals

Figure 3. Parisian in line 2 and 5 (6 pt, 300 ppi, scale factor 2).

Let us now turn to single measures. The classical “common pixels” measure M_{16} performs well (98 %) for 12 pt and 300 ppi but declines more rapidly than the “common dots” M_1 with decreasing font sizes. Obviously, it is more prone to noise. The quadratic weighted version M_2 performs best among the elementary measures. Surprisingly, the same concept applied to columns or to rows and columns instead of to rows alone declines the hit rate. Moreover, the additional consideration of white pixels in M_5 reduces the hit rate significantly.

Unexpected is also the result of the measures $M_6, \dots, M_{15}$. These variants around profiles, outlines and interval lengths perform well at higher pt-levels. However, for the critical sizes 6 and 7 pt they give no new perspectives. All our combination measures $M_{17}, \dots, M_{20}$ perform better than the champion of the rest M_2 . The best one is M_{20} which achieves at 6 pt 95.95 % for 300 ppi and 97.42 % for 600 ppi. Possibly, this approach can be improved a bit further.

In order to make our test statistically relevant we considered 112 commercially available, more or less popular fonts. As long as the structural details of fonts survive in the scan process we see no significant differences in relation to the hit rates between them. This is especially true for black letter (fraktur) fonts which is of vital importance for the mass-digitization of historic prints.

5. CONCLUSION

We investigated several simple similarity measures as an alternative to the classical “common pixels” measure CPM for glyph recognition by pattern matching in connection with noisy data. We found a couple of measures which are less prone to noise than CPM . Each of these measures achieves a character hit rate over 98 % for font sizes ≥ 8 pt. For the lowest considered font size 6 pt we observe for our best measure 95.95 % (300 ppi) and 97.42 % (600 ppi), whereas CPM trails behind with 92.93 % (300 ppi) and 93.63 % (600 ppi). It seems natural to ask if it is possible to design new measures which also avoid the decline in hit rates for small font sizes.

REFERENCES

- [1] U. Caluori and K. Simon, “An OCR-Concept for Historic Prints,” in *ARCHIVING 2013*, ISBN: 978-0-89208-304-6, 137–142, IS&T (2–5 April 2013).
- [2] H. Bunke and P. Wang, *Handbook of Character Recognition and Document Image Analysis*, World Scientific, Singapore/New Jersey/London/Hong Kong (1997).
- [3] M. Cheriet, N. Kharma, C. Liu, and C. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*, Wiley-Interscience (2007).
- [4] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, John Wiley, New York (2001).
- [5] S. Rice, G. Nagy, and T. Nartker, *Optical Character Recognition: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers, Boston (1999).
- [6] Haralambous, Y., *Fonts & Encodings*, O’Reilly, Cambridge (2007).
- [7] T. Corman, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, ISBN 0-262-03141-8 (1990).
- [8] W. Niblack, *An Introduction to Image Processing*, Prentice-Hall, Englewood Cliffs, NJ (1986).
- [9] M. Pelillo and E. Hancock(Eds.), *Similarity-Based Pattern Recognition*, Springer, 978-3-642-24470-4 (2011).

APPENDIX A. HIT RATES

pt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	95.22	95.36	94.61	95.28	89.13	64.76	71.01	85.70	92.55	86.52	85.44	65.11	79.82	71.33	75.95	92.93	95.35	95.42	95.64	95.95
7	97.39	97.34	96.76	97.51	93.84	73.17	78.89	91.65	95.95	91.33	89.65	72.64	87.11	79.35	83.27	96.13	97.35	97.49	97.48	97.57
8	98.13	98.30	97.48	97.98	96.18	79.64	84.12	93.92	97.12	94.02	92.35	77.52	90.29	84.75	86.84	97.44	98.26	98.37	98.24	98.43
9	98.60	98.53	97.84	98.17	96.93	85.77	87.44	96.66	97.96	95.35	95.39	83.12	91.92	87.72	90.91	98.01	98.46	98.57	98.63	98.72
10	98.75	98.88	98.19	98.69	98.16	89.73	90.15	97.48	98.46	96.70	96.79	86.84	93.84	89.33	93.56	98.42	98.88	98.92	98.73	98.88
11	98.11	98.30	97.75	98.21	97.29	88.49	89.66	96.82	98.26	96.40	96.44	87.67	92.37	89.67	91.75	98.00	98.19	98.40	98.27	98.56
12	99.11	99.08	98.68	99.01	98.52	93.00	92.55	98.08	98.89	97.70	97.87	90.05	95.36	91.58	95.25	98.99	99.05	99.15	99.05	99.14

Table 1. Character hit rates (%) per measure for scan resolution 300 ppi.

pt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	96.88	96.85	96.37	96.92	91.58	64.49	72.18	88.12	94.24	89.52	88.26	57.60	81.16	80.92	74.83	93.63	96.86	97.02	97.22	97.42
7	98.16	98.47	97.78	98.27	96.04	72.84	81.02	93.55	96.95	93.27	92.25	66.49	87.83	87.44	82.82	96.49	98.50	98.47	98.52	98.47
8	98.70	98.76	98.47	98.65	97.83	79.23	88.05	96.18	98.16	96.26	94.97	73.72	92.40	92.63	88.18	98.16	98.76	98.70	98.83	98.86
9	99.16	99.04	98.98	99.06	98.39	86.22	88.98	97.19	98.53	95.98	96.49	72.19	93.68	90.64	92.05	98.70	99.05	99.09	99.06	99.15
10	99.32	99.21	99.35	99.31	98.91	91.52	91.55	98.06	98.91	97.18	97.49	76.77	94.84	92.47	93.66	99.18	99.21	99.28	99.22	99.28
11	99.40	99.31	99.38	99.41	99.16	92.50	94.83	98.79	99.24	98.11	98.24	81.18	95.65	94.57	95.16	99.41	99.31	99.40	99.35	99.40
12	99.44	99.38	99.29	99.41	99.35	94.23	95.79	98.89	99.29	98.60	98.57	83.76	96.18	94.53	96.00	99.41	99.40	99.41	99.37	99.38

Table 2. Character hit rates (%) per measure for scan resolution 600 ppi.

APPENDIX B. USED FONTS

font name	No.	font name	No.	font name	No.	font name	No.
Arial	0	Bauer Bodoni	28	Bitstream Vera Sans bold-italic	56	Univers Com light	84
Arial italic	1	Bauer Bodoni italic	29	Century Schoolbook	57	Univers Com light-italic	85
Arial bold	2	Bauer Bodoni bold	30	Century Schoolbook bold	58	Univers Com black	86
Arial bold-italic	3	Bauer Bodoni bold-italic	31	Century Schoolbook italic	59	Univers Com black-italic	87
Times New Roman	4	Stone Sans medium	32	Century Schoolbook bold-italic	60	Unger Fraktur	88
Times New Roman italic	5	Stone Sans medium-italic	33	Palatino	61	Unger Fraktur bold	89
Times New Roman bold	6	Stone Sans bold	34	Palatino bold	62	Breitkopf	90
Times New Roman bold-italic	7	Stone Sans bold-italic	35	Palatino italic	63	Walbaum Fraktur	91
Verdana	8	Stone Sans semibold	36	Palatino bold-italic	64	Alte Schwabacher	92
Verdana italic	9	Stone Sans semibold-italic	37	GillSans	65	Arnold Boecklin	93
Verdana bold	10	Stone Serif medium	38	GillSans bold	66	Duc De Berry	94
Verdana bold-italic	11	Stone Serif medium-italic	39	GillSans italic	67	Duc De Berry DFR	95
Helvetica	12	Stone Serif bold	40	GillSans bold-italic	68	Eckmann	96
Helvetica italic	13	Stone Serif bold-italic	41	GillSans light	69	Fette Fraktur	97
Helvetica bold	14	Poppl Residenz	42	GillSans light-italic	70	Fette Kanzlei	98
Helvetica bold-italic	15	Poppl Residenz light	43	Futura medium	71	Linotext	99
Courier New	16	Futura-Bold	44	Futura bold	72	Linotext DFR	100
Courier New italic	17	Present	45	Futura medium-italic	73	Luthersche Fraktur	101
Courier New bold	18	Parisian	46	Futura condensed-medium	74	LutherscheFraktur DFR	102
Courier New bold-italic	19	Computer Modern Serif	47	Futura condensed-extrabold	75	Old English	103
Garamond3	20	Computer Modern Serif italic	48	Avant Garde	76	Weiss Rundgotisch	104
Garamond3 italic	21	Computer Modern Serif bold	49	Avant Garde italic	77	Wilhelm Klingspor	105
Garamond3 bold	22	Computer Modern Serif bold-italic	50	Avant Garde demibold	78	Wittenberger Fraktur	106
Garamond3 bold-italic	23	MS Sans Serif	51	Avant Garde demibold-italic	79	Wittenberger Fraktur bold	107
Myriad Pro	24	MS Reference Sans Serif	52	Univers Com	80	Wittenberger Fraktur DFR	108
Myriad Pro italic	25	Bitstream Vera Sans	53	Univers Com bold	81	Wittenberger Fraktur DFR bold	109
Myriad Pro bold	26	Bitstream Vera Sans bold	54	Univers Com italic	82	1883 Fraktur	110
Myriad Pro bold-italic	27	Bitstream Vera Sans italic	55	Univers Com bold-italic	83	1883 Fraktur bold	111

Table 3. List of fonts and their indices.