Collecting Historical Font Metrics from Google Books

Robert LiVolsi, Richard Zanibbi, and Charles Bigelow
Rochester Institute of Technology
rjl3050@rit.edu, rlaz@cs.rit.edu, cabppr@rit.edu

Abstract

A system is presented for extracting key metrics from fonts used in historical documents. The system identifies important landmarks on a page, such as margins, paragraphs, and lines, and applies frequency analysis techniques to identify relevant sizes. The system was validated by comparing its measurements to the measurements of a human expert on randomly selected samples, and differed on average from the expert by less than 5% for x-height, body size, and line spacing metrics.

1. Introduction

Character size is a major determinant of the legibility of text and has been studied from several disciplinary perspectives, including psychophysics [3], typographic history [12], and combinations of the two [4]. The current migration of reading from print to digital display raises questions of optimal character size, which analysis of font sizes in historical books may help answer.

Measurement of type sizes in historical and modern printed books has relied mainly on visual determinations made with optical or digital magnifiers, but such studies have usually been limited to a few hundred books or, more rarely, a few thousand [4, 12]. Recent digitization of large numbers of books dating back to the early era of European printing, e.g. Google Books, makes possible the automatic metrical analysis of type sizes in many thousands of books, and, potentially, many millions.

A pioneering culturomic study of five million books by Michel et. al. [6] analyzes centuries of lexical and grammatical usage to identify cultural trends. Reading is a visual activity as well as a symbolic one, and in this preliminary study, we focus on form rather than content, analyzing quantitative features of typographical elements to better understand trends in visual size optimization over centuries of typographic literacy.

This study determines metrics of x-height, body size, and line spacing, in accord with standard typographic-historical studies [12] and typical font usage. The x-height is defined as the distance from the text baseline (the imaginary horizontal line on which letters sit) to the x-line (the imaginary horizontal line tangent to the top of the lower-case x). The x-height is the major determinant of the perceived size of text [4]. Non-ascending or descending lower-case letters that are x-height include a, c, e, n, o, v, and x, and are also referred to as minims. Body size is defined as the distance between the descender line (the imaginary line tangent to the bottoms of the descending strokes) and the ascender line (imaginary line tangent to the tops of the ascending strokes), and is the standard metric for identifying font sizes. Lastly, line spacing is defined as the distance between subsequent baselines. These metrics rank among the most important factors influencing print cost as well as text legibility. The main objective of this paper is to present the system developed for this task and to assess whether the system can identify dominant font metrics reliably when compared with a human expert.

Surprisingly, while estimating font metrics [13] and identifying ascender, descender and minim characters (e.g. for word shape coding [11]) is pervasive in document image analysis [7], we have been unable to locate references concerned with compiling font metric statistics for their own sake. In our approach, we randomly sample pages from a book, and use the largest detected paragraph from each page to estimate font metrics. Our system also needs to be fast: we wish to collect metrics from thousands, even millions of books.

Collecting font metrics from historical documents is challenging, as pages are often skewed and/or warped, and noisy due to ink spread, bleedthrough of ink from the opposite side of a page, and dirt and damage accumulated from use over time. Before estimating metrics for the dominant font on a page, we go through the following steps: 1) deskew the page using a Hough transform, 2) segment text lines through Fourier analysis of vertical pixel projections, 3) merge textlines into paragraphs, selecting the largest paragraph, 4) re-estimate

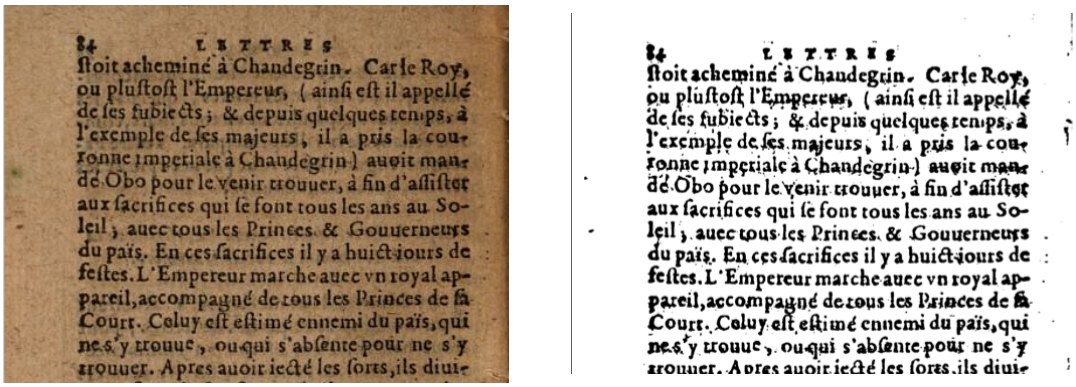

Figure 1: Historical document before and after processing by Google Books. While the majority of noise reduction is accomplished before inputting the document, the system must still be robust to any remaining noise, handwritten annotations, page deformations, and complex layout.

text line locations within the selected paragraph, 5) extract connected components, and 6) estimate the location for the baseline, x-line, and x-height, using connected component bounding boxes, and finally 7) classify connected components as ascenders, descenders, or minims. We then collect our metrics. We describe these steps in Sections 2 through 5 of the paper.

The methods we employ are simple; we use scanned book images that have been pre-processed and binarized by Google Books (see Figure 1). Preprocessing steps seem to include denoising, approximate page alignment, removal of page bleedthrough, and binarization. While the results presented in this paper are encouraging, we readily acknowledge that the system may benefit from incorporating more sophisticated methods for deskewing and page segmentation [1], and text line and baseline detection [2, 5, 9].

2. Paragraph Selection

The page is deskewed by first sub-sampling the original image by one quarter to decrease processing time and smudge text lines. The Hough transform is applied as in [5, 10], which decomposes the page into lines by their distance and angle from the origin. The peak value in the Hough domain is selected as the primary direction of the page, and the page is rotated at full resolution so that the dominant line is horizontal.

Paragraphs are segmented using XY segmentation, [8], resulting in a segment tree. Pure XY segmentation, however, will not always divide consistently depending on the spacing between lines. In order to account for this issue, a stopping condition is introduced. The typical line height is determined by taking the vertical projection of the entire page, normalizing the projection around its mean value, and applying a Fourier transform to provide a 1-d frequency analysis. The frequency with the greatest magnitude tends to correspond to the height of an individual line. However, the results may be inconsistent because individual lines themselves have two to four peaks: the baseline and the x-line, and possibly the ascender line and descender line. The Harmonic Product Spectrum is used to reduce this noise by repeatedly down-sampling the signal and multiplying the frequency contents together:

$$Y(\omega) = \prod_{r=1}^{R} |X(\omega r)| \tag{1}$$

$$\hat{Y}(\omega) = \max\{Y(\omega)\} \tag{2}$$

X is the original frequency spectrum, R is the number of harmonics to consider, Y is the harmonic product, and $\hat{Y}$ is the fundamental frequency. Note that in a discrete space, the more harmonics considered, the more the space is compressed. $R = 3$ is used in the system.

The fundamental frequency is taken as the typical line height, L_{typ}. $L_{min} = \frac{1}{2}L_{typ}$, and $L_{max} = 2L_{typ}$, defining a range of valid line heights. After segmenting the page, recombination rules are used to group adjacent lines together. Recombination occurs after a region is divided into vertical strips and it is determined that the height of at least one of those strips is within the range of line heights. All of the following rules must be satisfied:

1. The candidate region (for recombination) is aligned with the existing region on the left side.
2. The candidate region is within the range of valid line heights $[L_{min}, L_{max}]$.
3. The vertical distance between the candidate region and the existing region is less than L_{typ}.

4. The candidate region is at least half the horizontal length of the existing region.

Small lines, such as the end of paragraphs, are ignored. After paragraph segmentation is complete, only the largest region on the page is kept for further examination. This speeds up page processing and filters out unwanted areas of text, such as headers, footers, and margin notes.

3. Text Line Segmentation

The line segmenter is provided an individual paragraph. Edges are then detected by taking the absolute value of the derivative in the horizontal direction, as follows:

$$\forall(i,j \in I), \hat{I}_{(i,j)} = |I_{(i,j+1)} - I_{(i,j)}| \qquad (3)$$

where I is the original image of the paragraph and i and j are row and column indices, respectively. This helps emphasize character transitions and guards against minima of the vertical projection, occurring in the middle of an ascender or descender stem due to serifs.

Frequency analysis, i.e., calculating the FFT and the harmonic product spectrum, is performed again on the vertical projection of $\hat{I}$ to obtain a more accurate estimate of the line height. A sliding window equal in size to the estimated line height is then used across the vertical projection, alternating between identifying local maxima and local minima. When a maximum is found, the window start is placed at that maximum to find the next minimum, and vice versa.

Each local minimum corresponds to a split point between lines. The paragraph is divided into lines and the result is passed on to the character segmenter.

4. Connected Component Analysis

Once individual lines are identified, connected component analysis is performed to identify potential characters. Margins $\frac{1}{4}$ the height of the line region are added to the top and bottom of each line to account for ascenders and descenders that may have been split. Any components that reside entirely in the margin area are immediately discarded. The line height is also used to approximate the character size. Minims are approximately half of the line height, and character widths are approximately equal to the minim height (which is the same as x-height).

Connected components wider than the measured line height are divided to account for both noise and close character spacing. A sliding window approach is used again to identify local maxima and minima in the horizonal projection. The size of the window is equal to the height of the current line.

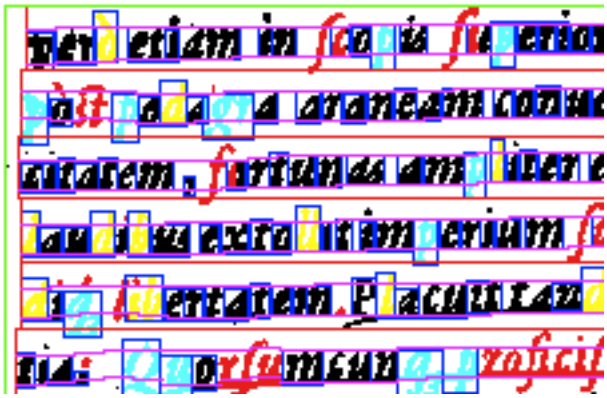

Figure 2: Detected text lines are outlined in red, and detected characters are outlined in blue. The detected baselines and x-lines are shown in purple. Cyan characters are descenders, yellow characters are ascenders, and red characters are rejected.

Although characters are still occasionally grouped together, this is of little concern because individual characters do not need to be classified. Likewise, split characters, such as the letter i, do not need to be rejoined. The connected components are sufficient for estimating the metrics of interest.

5. Font Metric Estimation

Our text line analysis has four steps: 1) the baseline, x-line, and x-height are roughly estimated, 2) minims are classified based on the rough estimate, 3) the baseline and the x-line are estimated again based on the location of the minims, and 4) the ascenders and descenders are identified.

For each line, the baseline, x-line, and x-height are estimated first by taking a histogram of the character bottom edges, the character top edges, and the character heights. The histograms are convolved by a Gaussian filter to account for the low number of samples. The peaks are then taken as the estimated metric values [13].

Characters with a height within one standard deviation of the estimated x-height are identified as minims. Characters less than $\frac{1}{2}$ of the x-height are classified as noise. To account for line skew, which typically occurs because of the book binding, polylines are used to represent the baseline and the x-line. Each line segment is calculated by averaging the estimated baseline or x-line and the bottom or top edge of any minim present at that segment. The polylines are then convolved with

Table 1: Error distributions by sample and by book, as well as comparison to synthetic documents with known ground truth, for the three metrics. Error is the percent difference between the system measurement and the human measurement. The mean error is shown with the standard deviation in parentheses ($\mu(\sigma)$).

	X Height	Body Size	Line Spacing
$Sample$	4.1%(4.3%)	1.8%(6.4%)	-0.4%(3.4%)
$Book$	3.7%(3.3%)	2.9%(5.4%)	-0.6%(4.1%)
$Synthetic$	0.1%(1.5%)	-1.4%(1.6%)	-0.1%(0.2%)

a Gaussian filter from left to right along the textline to produce a smooth curve. The remaining characters are then classified as ascenders if more than $\frac{1}{4}$ of the height exists above the x-line polyline, descenders if more than $\frac{1}{4}$ exists below the baseline polyline, and full if they are in both regions (i.e., characters that span the ascender and descender regions).

Histograms with 4 pixel resolution are taken of all minim, ascender, and descender heights on each page. A histogram for the line spacing is also taken by finding the difference between baseline estimates for successive lines. The body size is estimated as the difference between the sum of the ascender and descender heights and the minim height. The peaks are taken as the estimated statistics.

The final system output is shown in Figure 2. Despite the curvature of the page and handwritten annotations, the system can still identify a majority of the character types correctly.

6. Validation

The document analysis system processed 230 books from the 16^{th} and 17^{th} centuries. Pages were stored as binary images at 600x600 dpi. The largest paragraph by area was selected from each page using the segmentation method described above, and the x-height, body size, and line spacing were measured.

In order to verify the accuracy of these results, 30 books were randomly sampled from the 230 books, and 7 pages were randomly sampled from each book. To ease human measurement, a program was made to display four lines from the middle of each paragraph selected by the system. The user makes 3 baseline to x-line, 3 descender to ascender, and 3 baseline to baseline measurements with a mouse. This sampling process closely emulates how an expert in typography would determine the desired font measurements on a historical text by hand. These measurements were made by one of the authors, a professional typographer, without looking at the results produced by the system.

Human measurements and system measurements were compared directly for each sample. The error was calculated as the percent difference between the system measurements and the three human measurements averaged together for each metric.

Comparisons were also made at the book level. Spurious measurements were removed to account for multiple font sizes in the samples taken from the same book, which occurs often in the historical texts. This was achieved by looking for clusters across all of the measurements for the same metric with a threshold of 1 pt. size and retaining the cluster with the most samples. Thus, only the metrics for the dominant font are considered. This analysis was automated and applied to the system samples independently without looking at the human measurements. Error was calculated as the percent difference between the average system and average human measurements for each book.

For both the sample level and book level comparisons, the error distribution is biased positively for the x-height and body size metrics, as shown in Table 1. The algorithm aquires these metrics by finding the very top and bottom of each connected component. Any noise around the edge of the character will make it appear larger, whereas a human would know to ignore this noise. Furthermore, type designers make the round tops and bottoms of letters like o and c overhang the baseline and x-line, and similarly, make the peaks of the x-line serifs of letters like n and m, and the baseline angles of v and w overhang the baseline. and x-line, so the automatic measures will be slightly greater than those of a typographer measuring the traditional flat or non-overhanging x-height. Similar overhangs can be found at the ascender and descender lines, which will increase the automatic body size measurement compared to the typographer's measurement.

Additionally, the system processed synthetic, single column, single page documents generated using 22 modern fonts, one font per document, and compared to ground truth, with body size taken to be the vertical distance spanned by the letters p and h. Synthetic comparisons have improved error distributions over historical comparisons, as shown in Table 1. The pages processed were free of noise, and because digital fonts were used, there was no ambiguity in the ground truth. The x-height again tends to be slightly over-estimated, but the body size is actually underestimated in this case. This is most likely because the letter t, which is taller than a minim but shorter than an ascender, is confused as an ascender in some fonts and thus brings the entire average down for the body size. Lastly, the line spacing estimates are near exact, being off by only a single pixel

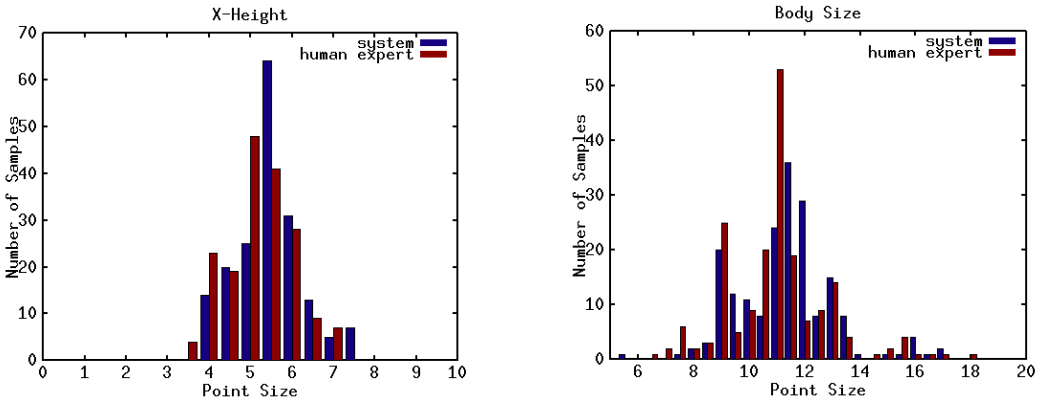

Figure 3: A histogram of x-height measurements (left) and body size measurements (right).

for one synthetic sample.

The distributions of system and human measurements across all samples are shown in Figure 3 for x-height and body size. Deviations occur around expected font sizes, such as 10, 12, and 14 pt., because historical texts are significantly noisier than modern documents. The system produces a comparable distribution to the human expert, although its measurements tend to be slightly larger.

7. Conclusion

A system was presented for efficiently analyzing historical documents and extracting key typographical metrics, specifically x-height, body size, and line spacing. This was achieved mostly using standard approaches in document analysis. The results were then validated by comparing samples analyzed by the algorithm against samples analyzed by a human expert. For all three metrics, error was found to be within 5% on average. Future work will focus on analyzing thousands of documents and searching for cultural trends in the metrics.

Acknowledgements

The authors would like to acknowledge Rob Pike and Jon Orwant for their assistance, and Diana Watkins and Andis Reks for gathering and organizing hundreds of historical documents. This research was funded by a Google Faculty Research Awards Grant.

References

[1] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. Historical document layout analysis competition. In *ICDAR*, pages 1516–1520. IEEE, 2011.

[2] A. Asi, R. Saabni, and J. El-Sana. Text line segmentation for gray scale historical document images. In *Proc. Historical Document Imaging and Processing*, pages 120–126, New York, NY, USA, 2011.

[3] G. Legge. *Psychophysics of Reading in Normal and Low Vision*. Lawrence Erlbaum Associates, 2007.

[4] G. Legge and C. Bigelow. Does print size matter for reading? a review of findings from vision science and typography. *J. of Vision*, 11(5), 2011.

[5] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *IJDAR*, 9(2-4):123–138, 2007.

[6] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[7] G. Nagy. Twenty years of document image analysis in pami. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):38–62, 2000.

[8] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. *ICPR*, pages 347–349, 1984.

[9] S. S. Bukhari, F. Shafait, T. M. Breuel. Text-Line Extraction using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters. In *ICDAR*, 2011.

[10] V. Shapiro, G. Gluhchev, and V. Sgurev. Handwritten document image segmentation and analysis. *Pattern Recognition Letters*, 14(1):71 – 78, 1993.

[11] A. L. Spitz. Shape-based word recognition. *IJDAR*, 1(4):178–190, 1999.

[12] H. Vervliet. *French Renaissance Printing Types: A Conspectus*. Oak Knoll Press, 2010.

[13] A. Zramdini and R. Ingold. Optical font recognition using typographical features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):877–882, Aug. 1998.