National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

COMPUTER AIDED DESIGN OF DIGITIZED FONTS

DAINI XIE

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

OCTOBER 1995

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN IRREVOCABLE NON-EXCLUSIVE LICENCE ALLOWING THE NATIONAL LIBRARY OF CANADA TO REPRODUCE, LOAN, DISTRIBUTE OR SELL COPIES OF HIS/HER THESIS BY ANY MEANS AND IN ANY FORM OR FORMAT, MAKING THIS THESIS AVAILABLE TO INTERESTED PERSONS.

L'AUTEUR A ACCORDE UNE LICENCE IRREVOCABLE ET NON EXCLUSIVE PERMETTANT A LA BIBLIOTHEQUE NATIONALE DU CANADA DE REPRODUIRE, PRETER, DISTRIBUER OU VENDRE DES COPIES DE SA THESE DE QUELQUE MANIERE ET SOUS QUELQUE FORME QUE CE SOIT POUR METTRE DES EXEMPLAIRES DE CETTE THESE A LA DISPOSITION DES PERSONNE INTERESSEES

THE AUTHOR RETAINS OWNERSHIP OF THE COPYRIGHT IN HIS/HER THESIS. NEITHER THE THESIS NOR SUBSTANTIAL EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT HIS/HER PERMISSION.

L'AUTEUR CONSERVE LA PROPRIETE DU DROIT D'AUTEUR QUI PROTEGE SA THESE. NI LA THESE NI DES EXTRAITS SUBSTANTIELS DE CELLE-CI NE DOIVENT ETRE IMPRIMES OU AUTREMENT REPRODUITS SANS SON AUTORISATION.

ISBN 0-612-05142-0

Canada

Abstract

Computer Aided Design of Digitized Fonts

Daini Xie

To make presentation of texts look good and legible is the main goal in digital font design and document typesetting. In this thesis, the principles in traditional font designs and typesetting to produce legible fonts have been studied and summarized.

A hypothetic visual model is proposed. In this model, the human visual system decodes visual information vertically and horizontally. After studying 9 most widely used and visually distinct typefaces (serif and san-serif), we found that each letterform has its own primary feature encoding directions. We also found that as long as the vital information of the letter is preserved, nonlinear distortion and misinformation in the non-primary direction will not seriously affect its legibility. However, the legibility of the letterforms will be reduced seriously if the primary features are removed. An objective method for comparing typeface quality is proposed. Font design methods and software support for designing digital fonts are compared.

The proposed font quality comparison method gives the following quality rating on the 9 tested typefaces from high to low: Palatino, New Century School Book, Times New Roman, Helvetica, Arial, Courier, Galliard BT, Swis721 Th BT, and Impact.

The results of comparing three font design methods such as bitmap method, outline method and component method, favour the last one due to its capability of combining different shape design methods and effectively managing large component databases.

Acknowledgements

I want to thank my supervisor, Dr. C. Y. Suen for providing this research topic, many references, ideas and the Natural Sciences and Engineering Research Council of Canada and the Ministry of Education of Quebec for financial support. I feel very lucky to have the opportunity to work at CENPARMI — a result of many years of hardworking in pattern recognition and machine intelligence by Dr. Suen and many of his colleagues.

I would like to thank all my friends and colleagues in CENPARMI for providing a friendly and fun working environment and all kinds of help during my study here. I would like especially to thank Nick Strathy and his parents Mr. William Strathy and Mrs. Dawn Strathy for their support and lots of prayer for the finishing of this work.

Finally, I want to take this special opportunity to thank my dear parents for all their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Today, computer technology is more and more involved in our daily work. It has changed considerably people's way of living and working. One of the examples is its involvement in information processing and electronic publishing.

In the early days, information was printed on the paper and its presentation was done by professional typographers. Now, this situation has changed. The presentation of information is done mostly by people who do not have much typographic training. Most interactive typesetting programs such as MS WORD, WordPefect, Ventura and Pagemaker are trying to give maximum freedom to the users. For professional typographers, freedom of design is wonderful because they know the principles of setting legible text. For most of the average users who do not have much experience in typesetting, designing legible documents is a challenge.

The whole issue of designing legible documents is a Man-machine communication problem. The information exchanged between men and the computer must be encoded in a way that both ends can understand and process quickly. A human retrieves information directly or indirectly from computers mostly through the visual channel. That is why the information presented by the computer must be designed in a way that can be read easily.

For over a century, many researchers have studied the factors that affect the speed of reading. In recent years, more effort has been put on the study of computer display design.

How to measure the legibility of typefaces is the problem this thesis is trying to solve. Most of the previous research was done through psychological experiments which involved thousands of subjects. The objective measurements are lacking. The objective measurements is important because it will help us to understand the visual system better. It will also provide a cheaper way for us to compare the quality of various typefaces. Once the measurement of legibility is defined, then we will be able to concentrate on the design of typefaces.

The study on the readability of typefaces and printed documents is very closely related to the characteristics of the human visual system. In this thesis, we first surveyed various studies related to the human visual system, especially in the area of contour integration and the legibility of prints. Based on the previous psychological experiments and a proposed hypothetic visual model, a method of measuring the legibility of letterforms has been designed.

Chapter 2

The Contour Integration

It is known that the human visual system is more sensitive to the outside contour of an object than to its inner contour. The outside contour defines the overall shape of the object. The human visual system has outstanding contour integration skill so that the shape of an object can be distinguished from a very noisy background.

A perceived contour is integrated by small line segments of various orientations. Several studies have demonstrated that curved lines composed of oriented segments can "pop out" under appropriate conditions. In the research conducted by David J. Field[1], some of the rules behind the contour integration process have been revealed. In their experiments, small oriented line segments are used to compose a curve. Under different orientation relationship among the segments, the subjects of this experiment are asked to identify the curve embedded in an array of randomly oriented elements(See Fig. 1).

- The detection of the path is related to the relative orientation of the successive elements which constitute the path

  The first experiment by them shows that the performance of path detection is dropping when the orientation difference (α in Fig. 2) between successive path elements is increasing. And when the orientation difference reaches 90°, there are only about 50% correction rate(See Fig. 3).

- The existence of selectivity in path integration

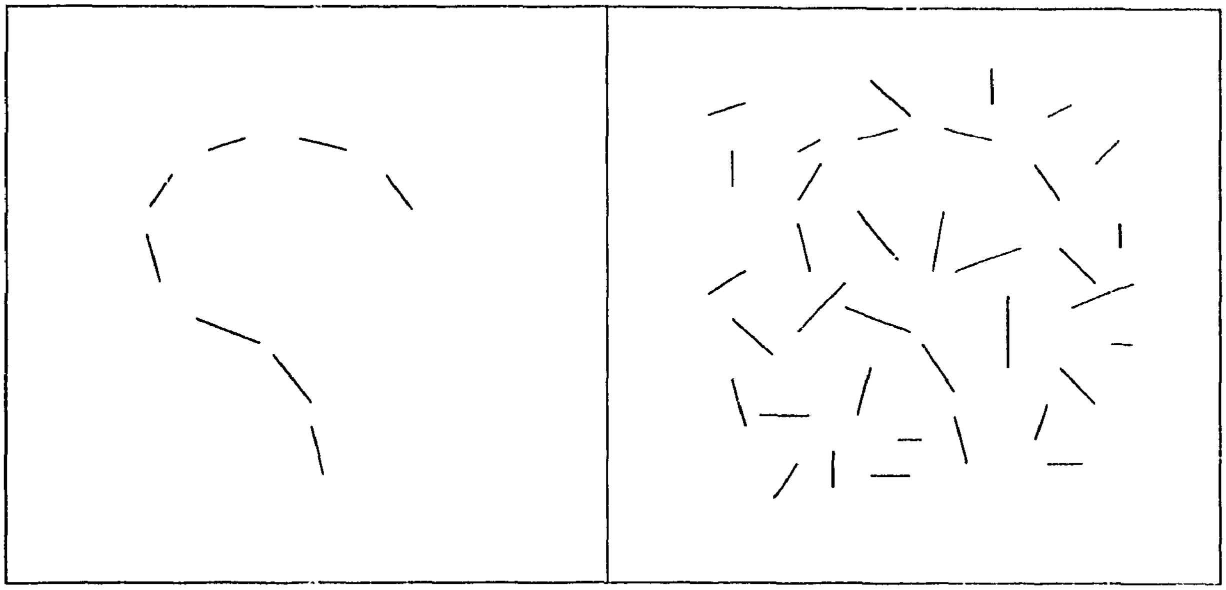


Figure 1: The visual task in Field's experiment : identify the provided curve in an array of randomly oriented elements

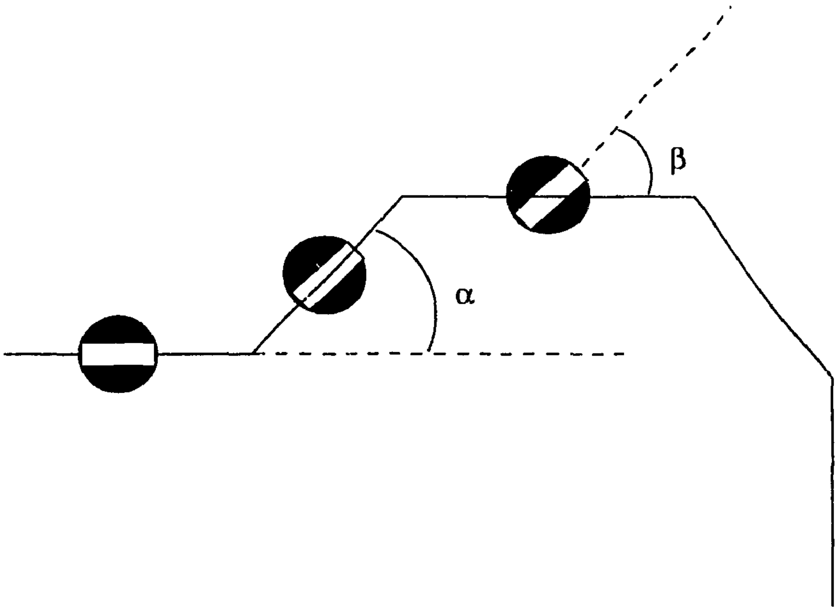


Figure 2: Adjustable parameters in Field's experiment

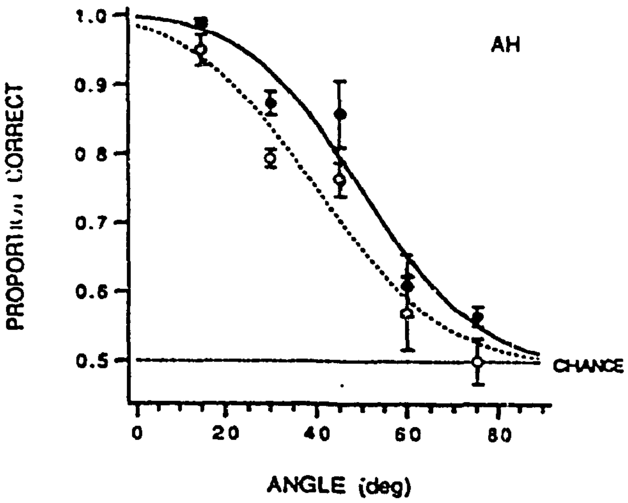


Figure 3: Path detection correct ratio as function of orientation difference of successive path elements

Since the first experiment shows the possibility of detecting path even when orientation differences of successive path elements reach 60° and the maximum orientation difference of the path elements presented in the random field is 90°, there must exist certain selecting process that picks the right elements to integrate them into a contour.

The experiment to verify above hypothesis has been designed so that the orientation of the elements is varied relative to the orientation of the path. The result is shown in Fig.4. The performance reduces substantially when the relative orientation between the elements and the path (β in Fig. 2) increases by 15°. The result shows that there is a constraint between the relative position and the orientation of the elements. The result suggests that the visual system can integrate large differences in orientation only when such differences lie along a smooth path.

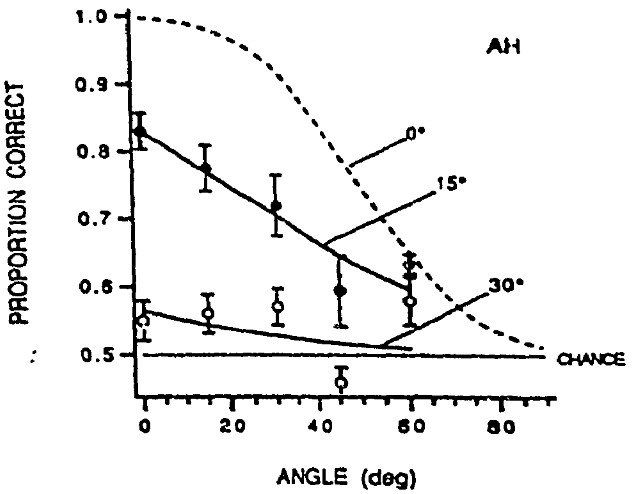


Figure 4: Performance comparison of relative orientation differences between the elements and the path

- No significant performance difference when there is moderate increase in inter-element distance

  The result is depicted in Fig. 5. Although the observers' performance decreases with distance, it is clear that the human visual system is capable of performing path integration over a wide range of distances.

The following chapter is the direct application of the rules described above to find the dominant points on digitized contours.

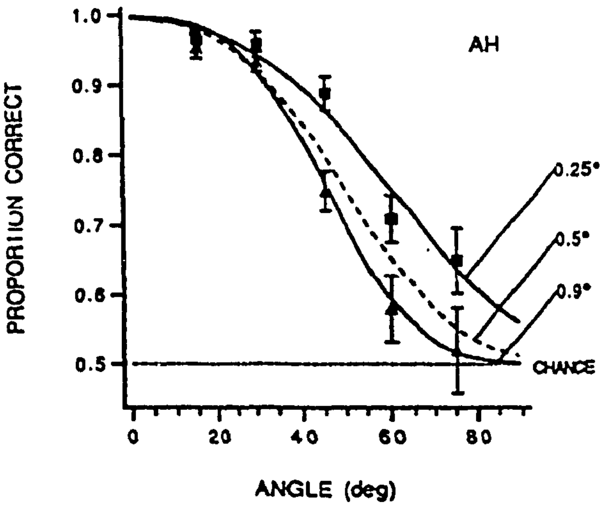


Figure 5: Performace comparison when there is moderate increase in inter-element distance

Chapter 3

Determining Dominant Points of Contours

Determination of dominant points can simplify the analysis of the shapes by drastically reducing the amount of data to be processed while at the same time preserving the important information about the shape.

Many methods have been used in shape representation. They comprise the following approaches:

- **Polygon Approximation Methods**

  In these methods, usually functions are defined for measuring fitting errors, and then methods are designed to minimize the fitting error.

  Wu [18] gives a summary on some of the commonly used polygon approximation methods in terms of their error function, optimization scheme and fitting strategies. The characteristic of these methods is that they are computational intensive.

- **Determination based on perceptual organization**

  Unlike the polygonal approximation methods, determination methods based on dominant points apply heuristics to determine important points of a contour.

  The human visual system is known for its efficient processing of low level visual information such as grouping elements of a perceived scene into meaningful or coherent clusters without prior knowledge of its contents. Comprehensively, incorporating perceptual organization into machine vision system may yield many advantages.

The heuristics of dominant points detection come from the famous observations by researchers such as Attneave, Rock, etc, that human subjects partition 2D curves based on three different objectives. They tend to choose a set of contour points that:

- best mark those locations at which distinctive curve segments are glued together;
- best allow the reconstruction of the complete curves;
- best allow a viewer to distinguish a given curve from the others.

Methods of determining dominant points usually use straight lines as primitives, and curvature maxima as dominant points ([7], [13]). In [5], Lowe presented a method that extracted the longest segments possible so that the curves are segmented at their most natural scale.

In this thesis, a simple method to extract polygon from an arbitrary shape has been designed. This method combines the advantage of the available methods and produces very good partition results. In this new method, corner points and end points of line segments are the main target points; then by partitioning smooth curve segments into smaller line segments and filtering colinear points, the contour is partitioned at its most natural scale. The following presents a more detailed description of the design.

3.0.1 Finding the target points

Corner points and end points of straight line segments can be found by evaluating the curvature on the given contour. Corner points are curvature extremas on the contour. The end points of straight line segments can be detected by finding the zero curvature segments on the contour. The challenge is how to determine the points used to describe a smooth and continuous contour.

In the previous chapter, we have shown that the perception of continuity must be derived from a process which integrates along the length of the path. Based on the contour integration characteristics we studied earlier, the line segments are used as primitives, and the points which form consecutive line segments with variation of 15deg in orientation between each other are selected as dominant points on a smooth contour.

Suppose $\mathcal{K}$ is the curvature function of a digitized curve $\mathcal{C}$. $\mathcal{K} = k_0, k_1, k_2, \ldots k_i, \ldots k_n$, and j is a known dominant point, then the closest dominant points to j on the contour j' must satisfy

$$\sum_{i=j+1}^{j'} k_i \geq 0 \qquad (1)$$

The following decides the steps used in the algorithm that selects the dominant points on a contour.

- Step 1 : Estimating the curvature of each point on the digitized contour;
- Step 2 : Extracting the corner points, i.e., curvature extremas;
- Step 3 : Extracting the end points of line segments from the contour;
- Step 4 : Partitioning the smooth contour into smaller line segments;
- Step 5 : Eliminating the collinear points obtained from previous steps;

3.1 Scale-based filtering and curvature estimation

Curvature estimation is the key in the proposed method. This topic has been intensively studied because the curvature is an important feature used in pattern recognition. Some of the research studies[3] demonstrate the effectiveness of local orientation on the global perception of curvature. Rosenfeld-Johnston's angle detection procedure[7], and Freeman-Davis algorithm[2] are some of the most popular ones. All these methods provide various schemes for determining the local region of support and the significance of points. It has been pointed out in [6] that the region of support of a point can be determined independently and is based on the local properties. However, in these methods, incorrect curvature measure maybe assigned to a point if the input smoothing parameter is not chosen correctly, and hence dominant points maybe omitted.

In the proposed method of estimating curvature, the scale-based filter to provide precise way of estimating the tangent of each point on the contour, and then we derive the curvature of each point from its tangent.

3.1.1 Scale-based filtering

Due to its capability in describing shapes at different scales, and its invariance to scaling, rotation and translation, scale-based filters have been used widely in computer vision applications [17], and [15].

A scale-based filter has the following form:

$$g(t,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-t^2/2\sigma^2}. \tag{2}$$

Suppose

$$C = (x(t), y(t))$$

is the parametric description of contour C, where t is a linear function of the path length ranging over the closed interval [0,N], where N is the number of points on the contour. If the curve is closed, x(t) and y(t) will be periodic functions.

Convolve x(t), y(t) with Gaussian curvature filter at different σ values, new contours

$$C' = (X(t), Y(t))$$

at different scales will be obtained.

$$X(t,\sigma) = x(t)\bigotimes g(t,\sigma)$$

$$= \int_{-\infty}^{\infty} x(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-u)^2/2\sigma^2} du$$

$$\dot{x} = \frac{dx}{dt} \tag{3}$$

$$\ddot{x} = \frac{d^2x}{dt_2} \tag{4}$$

$$t = \dot{y}/\dot{x}; \tag{5}$$

$$k = \frac{d^2y}{dx^2} \tag{6}$$

$$k = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \tag{7}$$

$$k = d\alpha/ds \tag{8}$$

Some of the interesting properties of the filtered curve are listed below:

The filtered $\dot{X}(t,\sigma)$ and $\ddot{X}(t,\sigma)$ can be computed from x(t), y(t) by using:

$$\dot{X}(t,\sigma) = x(t)\bigotimes(\frac{\partial g(t,\sigma)}{\partial t}) \tag{9}$$

$$\ddot{X}(t,\sigma) = x(t)\bigotimes(\frac{\partial^2 g(t,\sigma)}{\partial t}) \tag{10}$$

Fig. 6 shows the effect of filtering at various scales.

3.1.2 Curvature estimation

Now the problem of curvature estimation has an obvious solution as indicated in Equation 9. The first order derivatives of a scale-filtered contour can be calculated by convolving the parametric contour function with the first order derivative of the

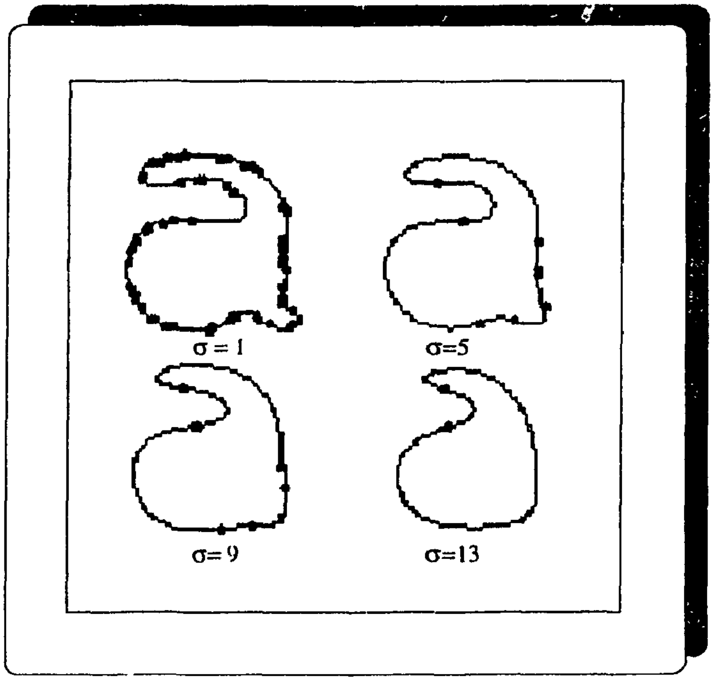


Figure 6: Contour and its inflexion points after Gaussian filtering

Gaussian kernel. In order to get the first order derivatives through Equation 9, we must choose a small σ value so that our estimation can be as close as possible to the original curve.

A small σ value will quickly decrease the Gaussian kernel, we can chop the function to a smaller filter mask as shown in Equation 13 before we convolve Equation 13 with x(t) and y(t) to make the computation more efficient.

$$F[12] = \tag{11}$$

$$[-0.00000001523, -0.0000037267, -0.00033546, -0.0111, -0.1353, -0.6065, \tag{12}$$

$$0.6065, 0.1353, 0.0111, 0.00033546, 0.0000037267, 0.00000001523] \tag{13}$$

The suitable smoothing parameter σ is dependent on the image size. In this research, $\sigma = 1$ is used for images of size 32x32.

Once dy and dx are found, the tangent can be estimated by Equation 5. The curvature which is the angular velocity of the tangent vectors along the contour can

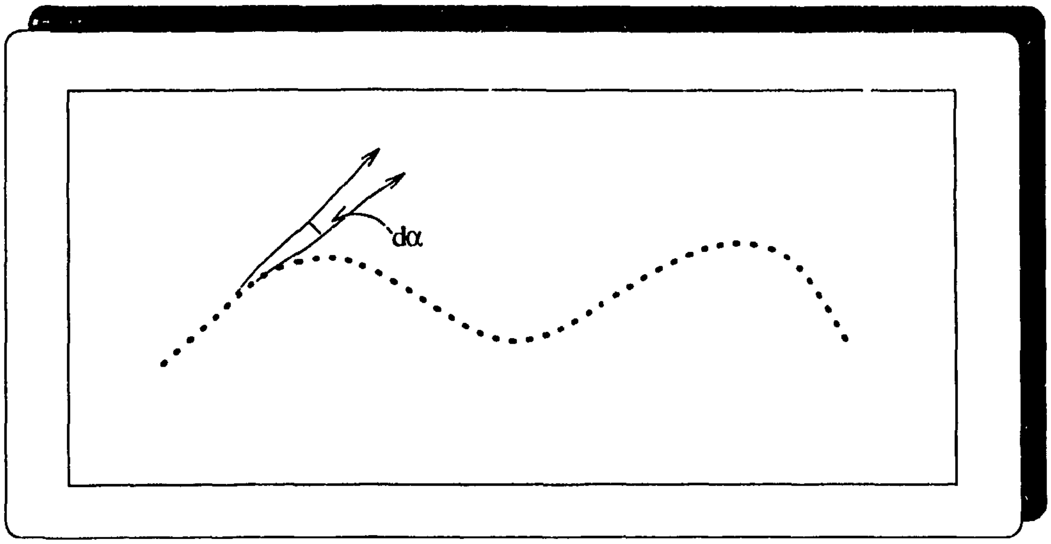


Figure 7: Relationship Between Curvature and Tangent Vector

be computed by Equation 8.

Fig. 7 shows the relationship between estimated curvature and tangent vector on the contour.

Fig. 8 and Fig. 10 show the tangent functions and curvature function of a test contour.

3.2 Filtering noisy curvature function by using binomial filter

The curvature function estimated from the proposed method contains a lot of noise due to the noisy input boundary and small σ value of the Gaussian filter. Since the noise in the curvature function has a high frequency, we can use any low pass filter to get rid of it. We choose a binomial filter mask(equation 14) here to filter the noise in the curvature function. See the result in Fig. 11 compared with the unfiltered one in Fig. 10.

$$F = [1,6,15,20,15,6,1]/64; \tag{14}$$

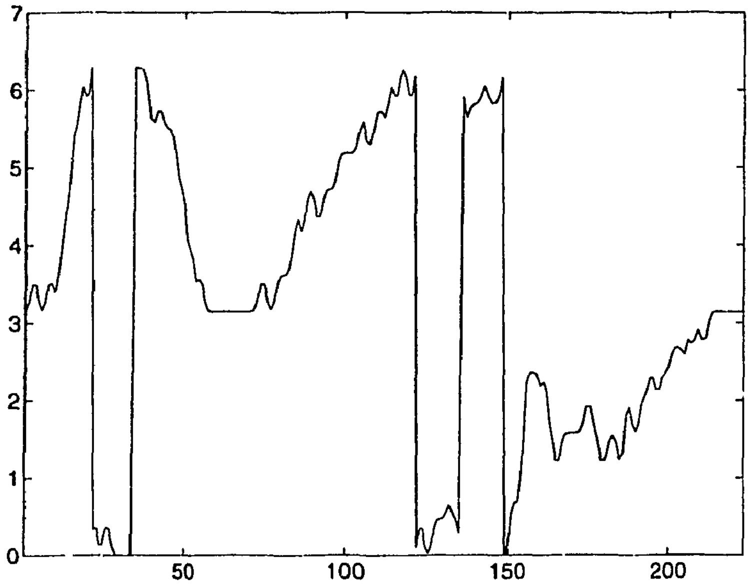


Figure 8: The Tangent Function of a Test Contour

Fig. 11 gives a sample of filtered curvature function of Fig. 10.

After finding the corners, we extract all the possible line segments from the input contour. However, the filtered curvature function sometimes has a lot of small fluctuations across the X-axis, hence a threshold value is necessary to suppress all these zero crossings. After this process, small zero curvature segments are merged into bigger pieces. Since line segments are constituted from the consecutive points on a digitized contour with the same tangent, we can expect that the second order derivatives will be zero for all the points on the same line. Therefore the zero curvature segments on the curvature function of a contour correspond to the line segments on the contour. Figs. 13 and 14 show an exact correspondence of curvature function zero segments and their contour line segments. Unfilled rectangle pixels in the figure correspond to the points on line segments.

Fig. 15 gives examples of approximation polygons after corners and line segments have been selected. We can see that all the corners and line segments are correctly extracted, and the basic shape of the object is defined by just these points alone.

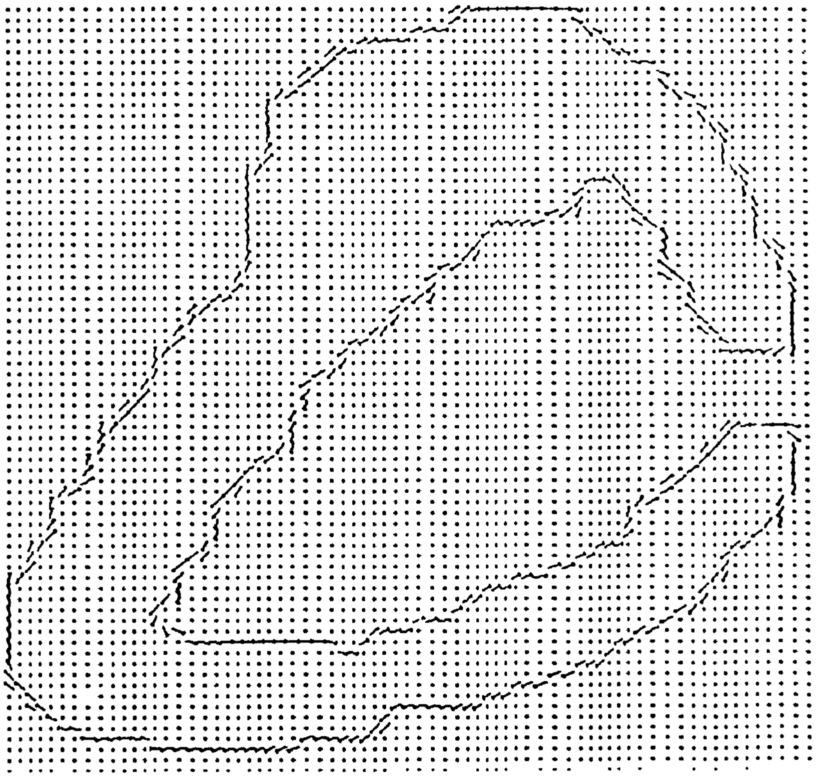


Figure 9: The estimated tangent vector on a contour

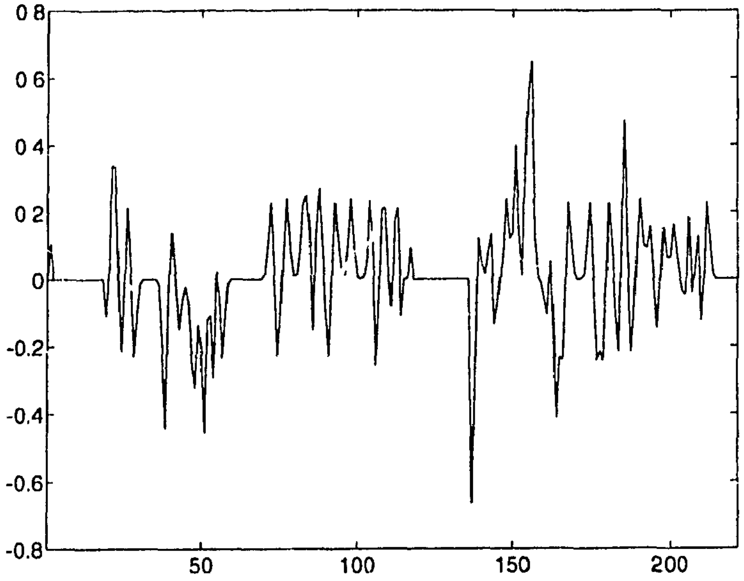


Figure 10: The Curvature Function of a Test Contour

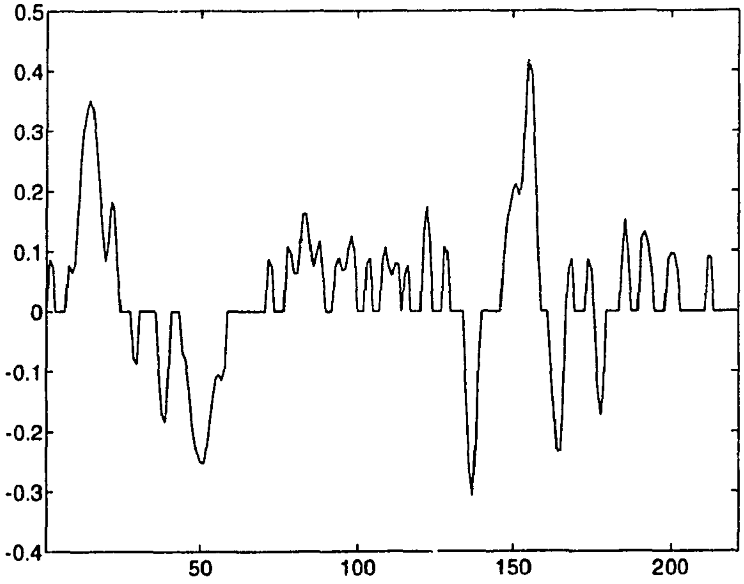


Figure 11: Curvature Function After Binomial Filtering with Filter Size=7

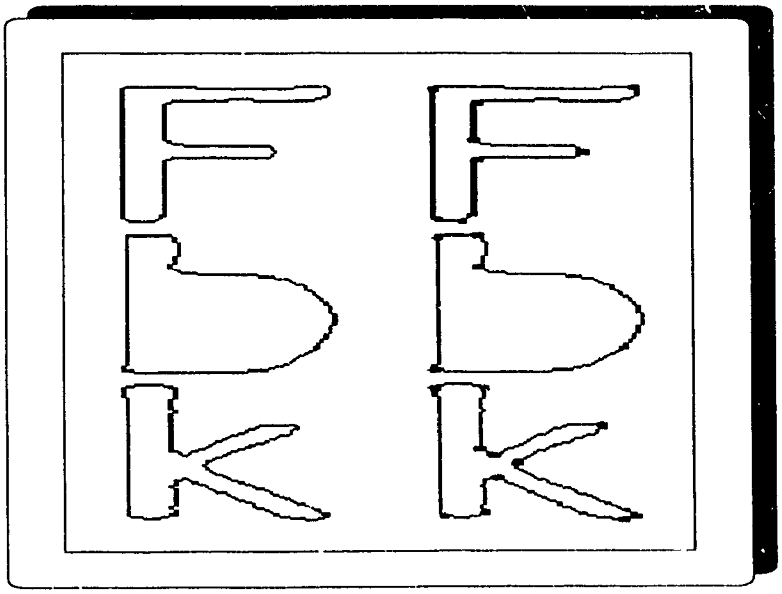


Figure 12: Examples of corner detection, threshold=0.2

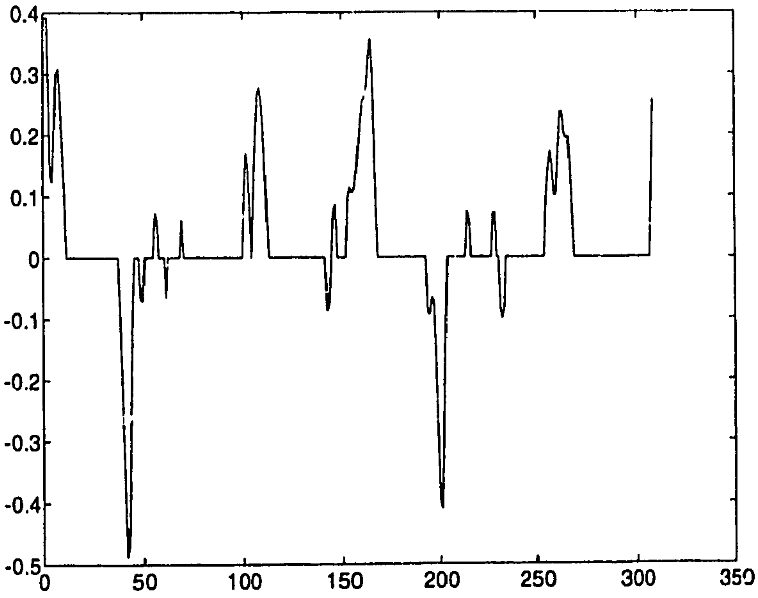


Figure 13: Curvature plot of a test contour

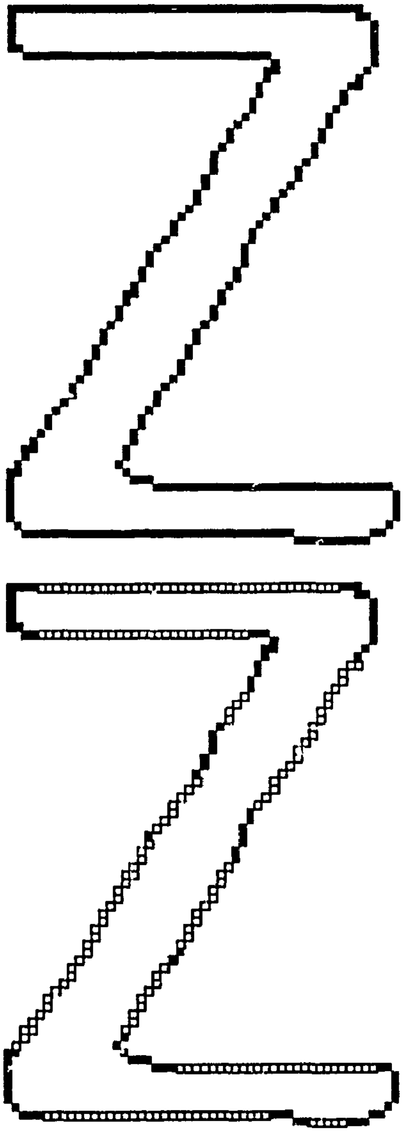


Figure 14: Line segments of a test contour

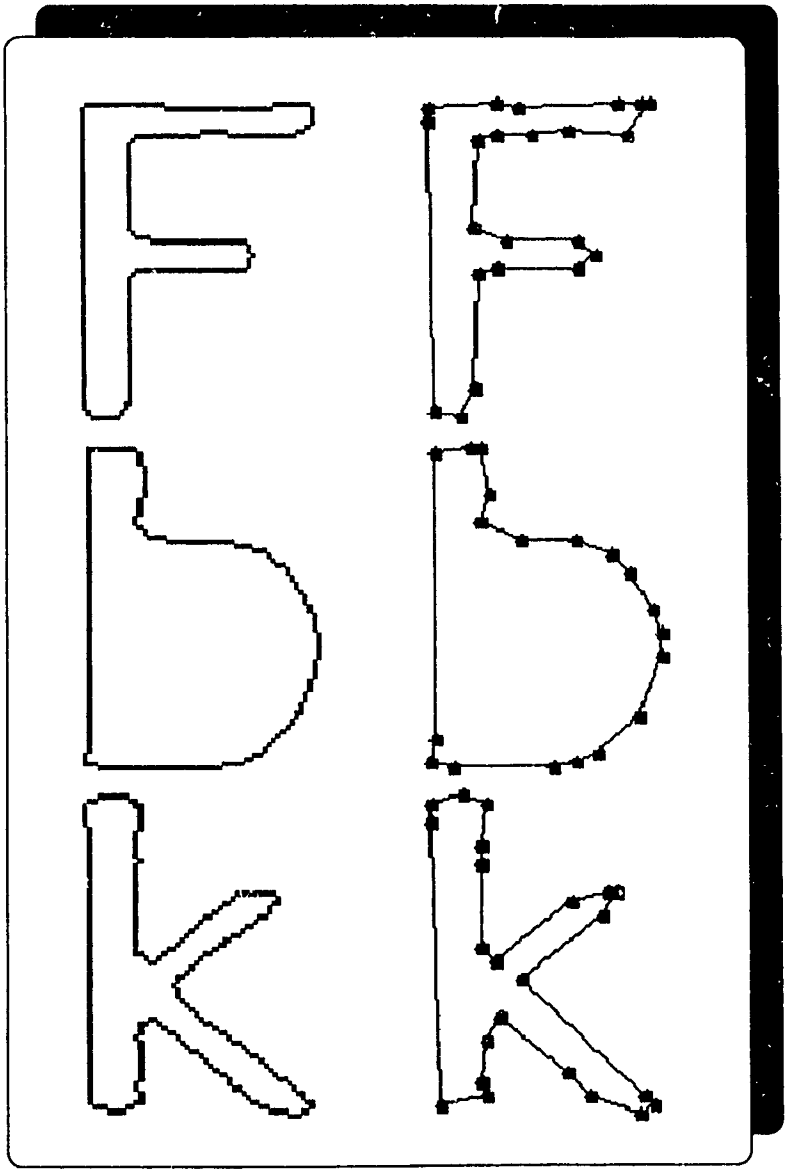


Figure 15: The pclygon after corners and line segments are determined

Fig. 16 gives the approximate polygon after smooth curves are segmented into pieces. It shows a much closer shape to the original when a few points are added than the result from the previous stage.

Fig. 17 shows that by removing colinear points, points for describing the shape can be reduced further without sacrificing the quality of the approximation.

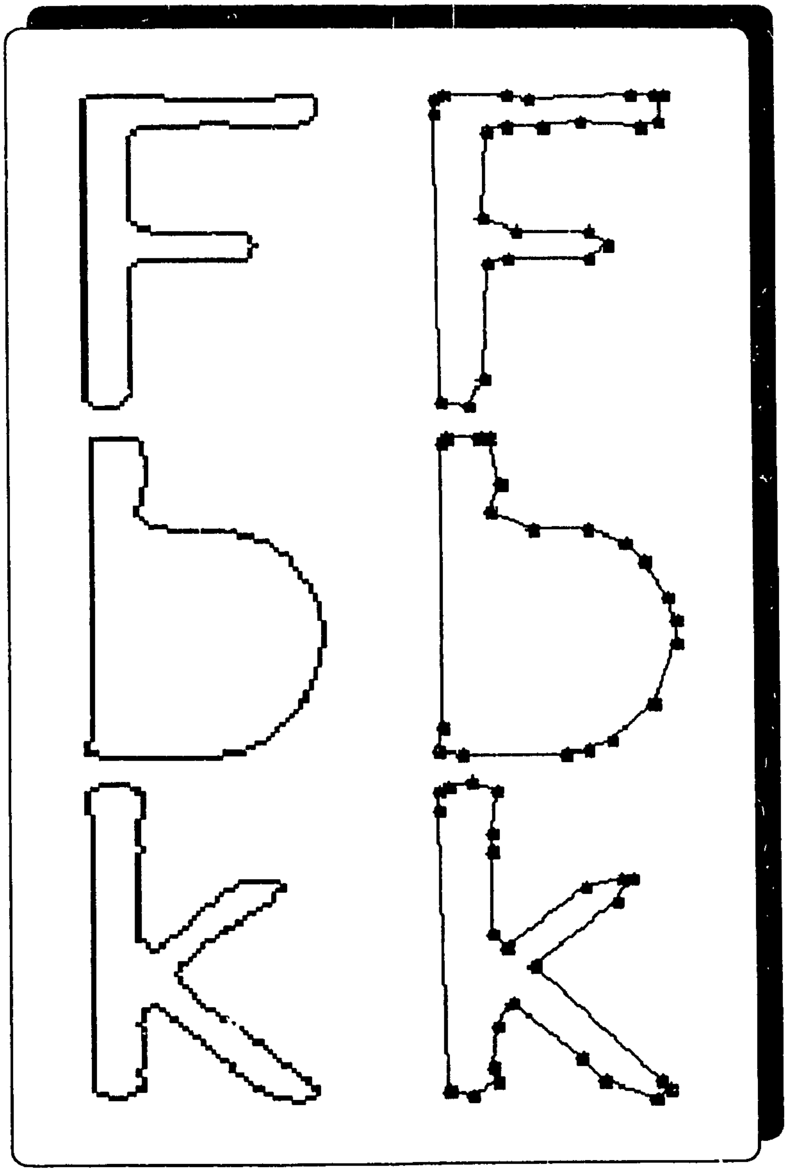


Figure 16: The approximated polygon after Step 4

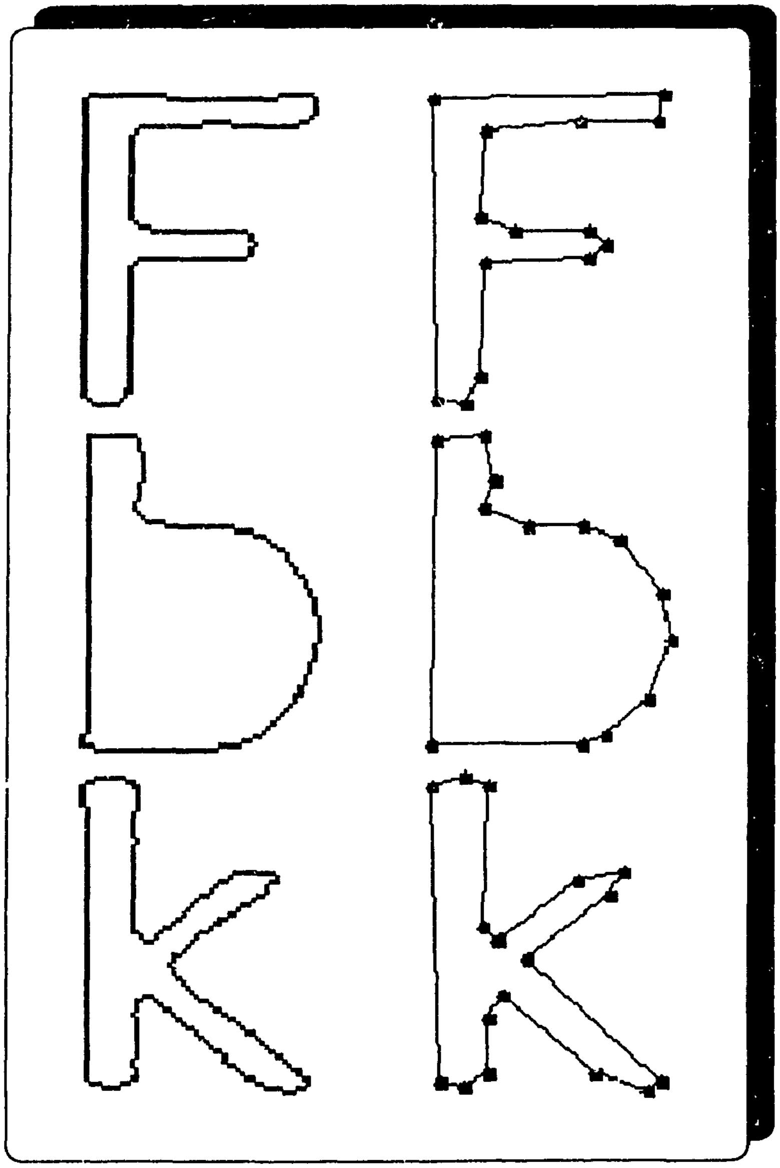


Figure 17: Final approximation after elimination of colinear points

Chapter 4

Legibility

Unlike studies related to perceptual organization, research on legibility examines how the higher level visual composition affects the efficiency of reading, such as the arrangement of the shape elements of the letterforms, characteristic word forms and all the other typographical factors such as type size, line width, leading, etc.. When people read a piece of continuous text, they are actually reading for comprehension rather than recognition of individual letters or words. When text is considered at this level, the full complexity of the reading process — comes into play [8].

Before starting a review of the research activities on legibility, some terminologies on typography will be introduced to make following text easier to read. The knowledge of typography is not only useful in designing fonts and documents, it also reflects intriguingly on the nature of human reading abilities.

4.1 Terminologies related to typography

The letterforms are letters, numbers and other symbols that people can read.

A typeface is an abstract design idea on how letters are to be presented. It is a distinctive design for a set of visually related symbols. Times New Roman, Optima, and Courier are all typefaces; they are designs for the shapes of the symbols for which they are defined. A typeface can be realized in various sizes.

A character set defines the set of symbols.

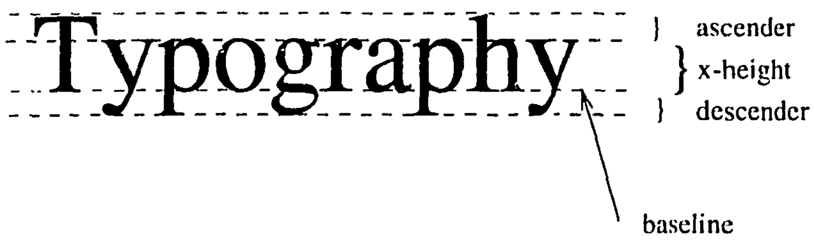


Figure 18: Parts of letters

A font is a particular example of a typeface, often in a particular size, with symbols for each element of a particular character set. Many distinct fonts may be associated with a particular typeface. For example, for Times New Roman fonts may different in size, as well as in weight,and styles such as italic, horizontally expanded and condensed.

The individual symbols in a font are called glyphs.

The x-height is the basic height of the lower-case letters. It is the distance that most of the lower-case letters extend above the baseline, upon which they rest. In a character set, the height of lower-case "x" is usually considered as the x-height (See Fig. 18).

The ascender is the part of the letter that reaches above the x-height.

The descender is the part of the letter that falls below the baseline.

The cap-height is the distance that the upper-case letters extend above the baseline.

Serifs are small lateral extensions at the end of strokes. Some typefaces have serifs. and some do not. Typefaces which do not have serifs are called san serif typefaces.

The weight of a typeface is the heaviness or blackness of the letters.

The contrast of a typeface refers to the relationship between vertical and horizontal strokes. The greater the difference between the thickness of horizontal and vertical strokes, the greater the contrast will be.

4.2 Previous research on legibility

Studies on the legibility of alphanumeric symbols can be found in [14],[9], [12], [10] and [11]. It has been shown that some typefaces are more legible than the others and good typesetting can compensate the legibility of certain typefaces. Generally, letters and typesettings which are visually pleasing to the eye are considered to be legible by human subjects and result in faster perception and more efficient reading.

Legible typefaces have the following characteristics:

- Familiarity to the reader

  Typefaces that respect history and the reader's familiarity is generally considered legible. Most typefaces used today are evolved to their current shape after many typographic studies and research. A good typeface design will conform to the range of letter shapes that is familiar to the reader. Intentional distortion or deviations from the norm result in low quality, regardless of how well the print result may be [8].

- Simple outline

  Fonts with simple outlines are easier to perceive and cause less confusion. Heavy serif is not suggested for designing legible fonts. However, it is a very debatable issue on whether serif typefaces are more legible than san serif ones. Most typographers and readers prefer to read running text by serif typefaces despite the fact that many experimental result confirms san serif fonts cause less error in reading. Since a very simple outline will also cause assimilation between shapes, it will result in decreasing readability to a certain degree. The difference between serif and san serif typefaces is that serif typefaces tend to break the assimilation between letters and san serif ones are on the contrary. For example, the letter 'bd' and 'pq'are mirror-image shapes in san serif typefaces, but not in most serif typefaces. Since the shape elements such as bow and stems are assimilated, when different letters are put together, it is easier to cause confusion and ambiguity. For example, the combination 'rn' in a serif typeface is usually less like an 'm' than in a san serif face:'rn'.

Most of the legibility experiments based on reading errors have been designed by asking subjects to identify single letters of different typefaces. Many people think that the experimental results contradict with the typographers' preference. Since human reading is more based on word reading and context of the reading material than individual letters, it is not surprising that the typefaces such as those with serifs which provide more distinct wordform in the running text are more preferable from the reader's point of view.

- Aspect ratio between 0.7 and 0.9

  It is also worth to note that numerals with height-width ratio of 10:7.5 and a stroke width-to-height ratio of 1:10 are the most readily perceived. Numeral such as 6, 9, 3 and 5 especially need to have optimal height-width proportion for easier perception.

- Even heaviness of vertical strokes and horizontal strokes

  It is recommended by typographers that the shape elements of a typeface should always conform certain design constraints. In most cases, designers specify the parameters for all the shape elements of a typeface ahead of the design, and then modifying individual glyphs slightly to make the typeface look at its best. Usually, vertical stem thickness and horizontal stroke thickness are made to be constant for all the letterforms of one typeface. This results in consistent appearance when the typeface is used in a piece of running text, which in turn, causes less surprises to distract the reader.

- Good balance

  This is another aesthetic issue involved in font design. Balanced design shows a very regular division of space by the shape elements. It meets the expectation of the human visual system, and that is why it was found to be visually pleasing and contribute to comfortable reading.

- Ample white space between strokes

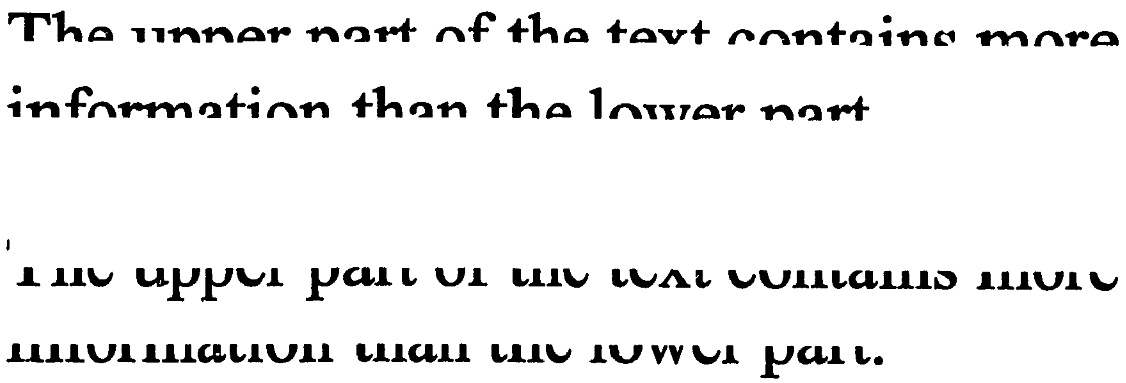


Figure 19: The upper part of the text contains more information than the lower

The white space between strokes makes them easier to identify and then, result in more accurate recognition. When there are enough spacing between strokes, the font information is less likely to be destroyed by noise.

- x-height, ascender and descender sizes

  The ratio of x-height with respect to font body size is an important design parameter that affects the overall look of the running text. The valid range is between 40% and 60%. Too small x-height percentage results in difficult reading because most lower case letters will become very small, and their features become unnoticeable. On the other hand, too large an x-height will make characteristic wordform disappear when all the characters have a similar height. The right pencentage of x-height with respect to the font body size depends also on different display devices. Low resolution display devices may require fonts with larger x-height percentage for efficient use of the available space.

  The sizes of the ascender and descender are not required to be equal. Since more information is contained in the upper part of the letterforms (See Fig. 19), people usually make ascender a little bigger than descender when they have to make a trade-off.

Besides the choice of typefaces, there are also typesetting factors that influence the legibility of printed text.

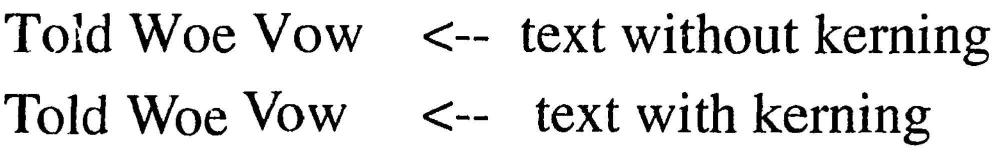


Figure 20: Text before kerning and after kerning

- Inter-letter spacing

  The inter-letter spacing of words greatly affects the legibility of a piece of text. Irregularities in spacing attract the reader's attention and reduce the quality of text presentation. In most of the electronics publishing systems, kerning is a basic functionality to provide constant inter-letter space for the running text(See Fig. 20).

- Inter-word space

  The inter-word spacing is less critical as inter-letter spacing because the eye is not decoding shape information in the spaces between words, but only recognizes the breaks between words. The rule in the traditional typesetting practice is that inter-word spaces should not be bigger than interline white space. So when the interline space increases, the interword space should also increase to keep the text readable.

- Text line width

  Studies by Wiggins[16] and Tinker [14] provide some interesting data about line width and legibility. It has been found that slow reading is caused by more lost eye fixations in order to track errors and to find the beginning of the next line. There exists an optimum range of line width, above and below this range will result in reading performance drop off. In one of the studies by Tinker, the optimum range of line width for 10-Point typefaces is between 3 inches and 4.5 inches.

  Listed in Table. 1 are the data from a study by Tinker which involved 6,000 subjects on the determination of optimal line widths and the leading for each type size he used.

10-Point Type	Line Width	Set Solid	1-Point Leading	2-Point Leading	4-Point Leading
	9	-9.3	-6.0	-5.3	-7.1
	14	-4.5	-0.6	-0.3	-1.7
	19	-6.0	-5.1	0.0	-2.0
	31	-3.7	-3.8	-2.4	3.6
	43	-9.1	-9.0	-5.9	-8.8
11-Point Type	**Line Width**	**Set Solid**	**1-Point Leading**	**2-Point Leading**	**4-Point Leading**
	7	-11.2	-9.0	-12.2	-10.2
	16	4.7	-0.6	0.8	-3.3
	25	-0.7	0.7	0.0	-1.4
	34	-2.5	-0.1	-1.6	-2.6
	43	-6.4	-4.7	-3.5	2.8
12-Point Type	**Line Width**	**Set Solid**	**1-Point Leading**	**2-Point Leading**	**4-Point Leading**
	9	-7.4	-6.0	-5.8	-5.0
	17	-2.6	-0.9	0.8	-0.9
	25	-0.8	-2.5	0.0	2.4
	33	-2.7	-0.7	0.0	2.1
	41	-8.1	-3.7	-3.5	-3.5

Table 1: Optimal combinations of line width (in picas) and leading for various text type sizes. Figures are the percent reduction in speed relative to a chosen setting(shown in bold face). "Set solid" indicates no added leading

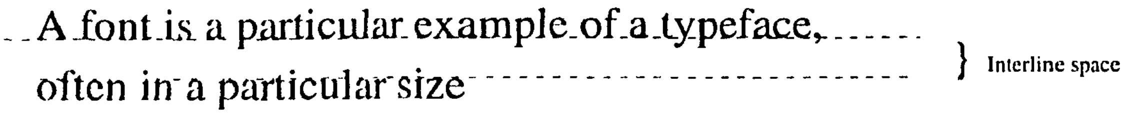


Figure 21: The interline space between 2 lines of text

- Justification of lines

  Justification of lines is to put lines of text in some special relation to form the margins. There are several kinds of justification:

  - Flush Left Text abuts the left margin.
  - Flush Right Text abuts the right margin
  - Centered Text is centered, same amount of margin on the left and right
  - Justified Text abuts both margins.

  Most typographers agree that the text should be at least left justified to make text legible. But there are divided opinions on whether a text should be justified on the right side as well as the left. The reason that justification is necessary for fast reading is determined by the nature of human reading process. When a person is reading, his or her eye scans through the text line by line with many times of eye fixation. Whenever a fixation occurs, the muscles have to rotate the eye and refocus again. Most of the reading time is spent on the move from one fixation to the next one. Left justified text helps the eye to search for the new line to start.

- Inter-line white space, line width, type size

  The combination of inter-line white space, line width, and type size also affect the legibility of text.

  The interline white space normally is defined as the amount of white space between the base line and the x-height on the line below as shown in Fig. 21.

  The appropriate interline space is determined by other factors such as type size, the design of the type such as the x height, and character set and the length of

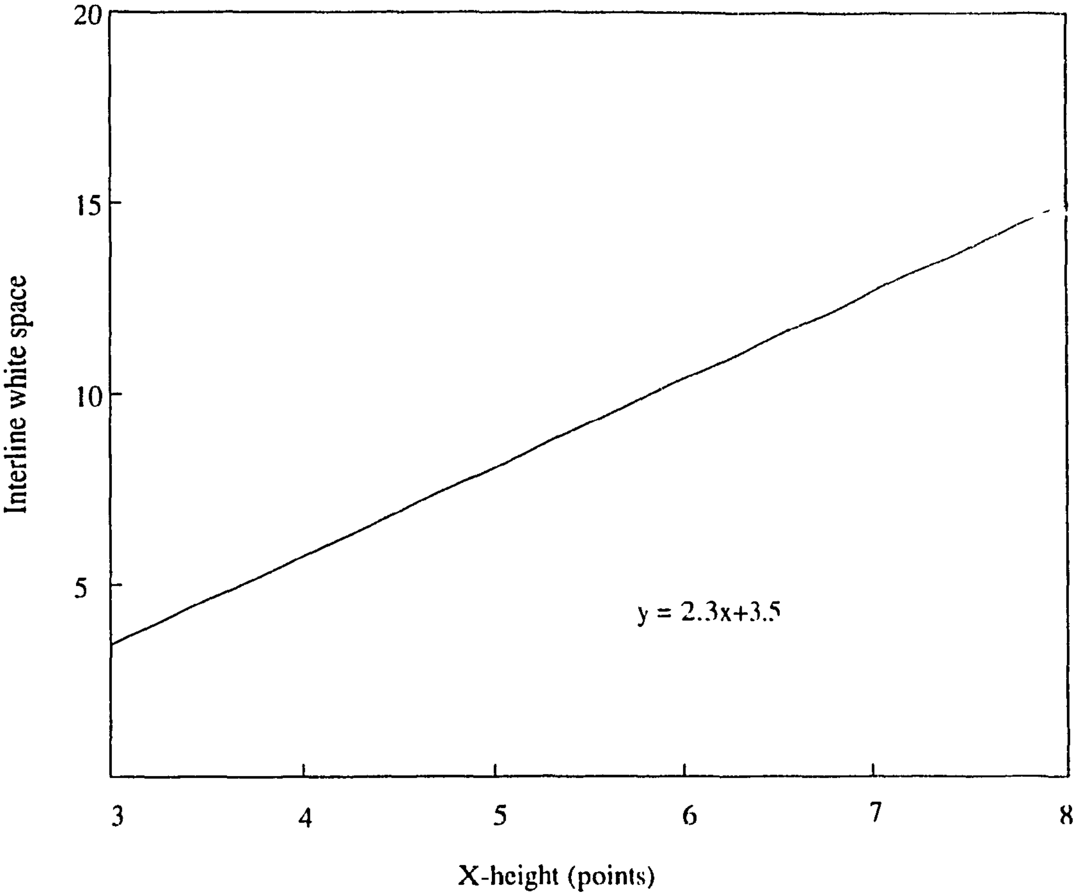


Figure 22: Interline white space as a function of measured x-height

the line. By experience, the relationship between interline white space and the factors listed above can be described as function relationship. The functions shown in the following figures are based on experience and experiments, and are very useful for electronics typesetting system during the formatting. It can also be used to evaluate the quality of the document typesetting.

Fig. 23 shows that when the line width increases, the interline white space has to increase correspondingly to make texts readable. This is very useful in designing text browser interfaces, so that the software system can adjust the display of the text automatically according to the size of the viewing area.

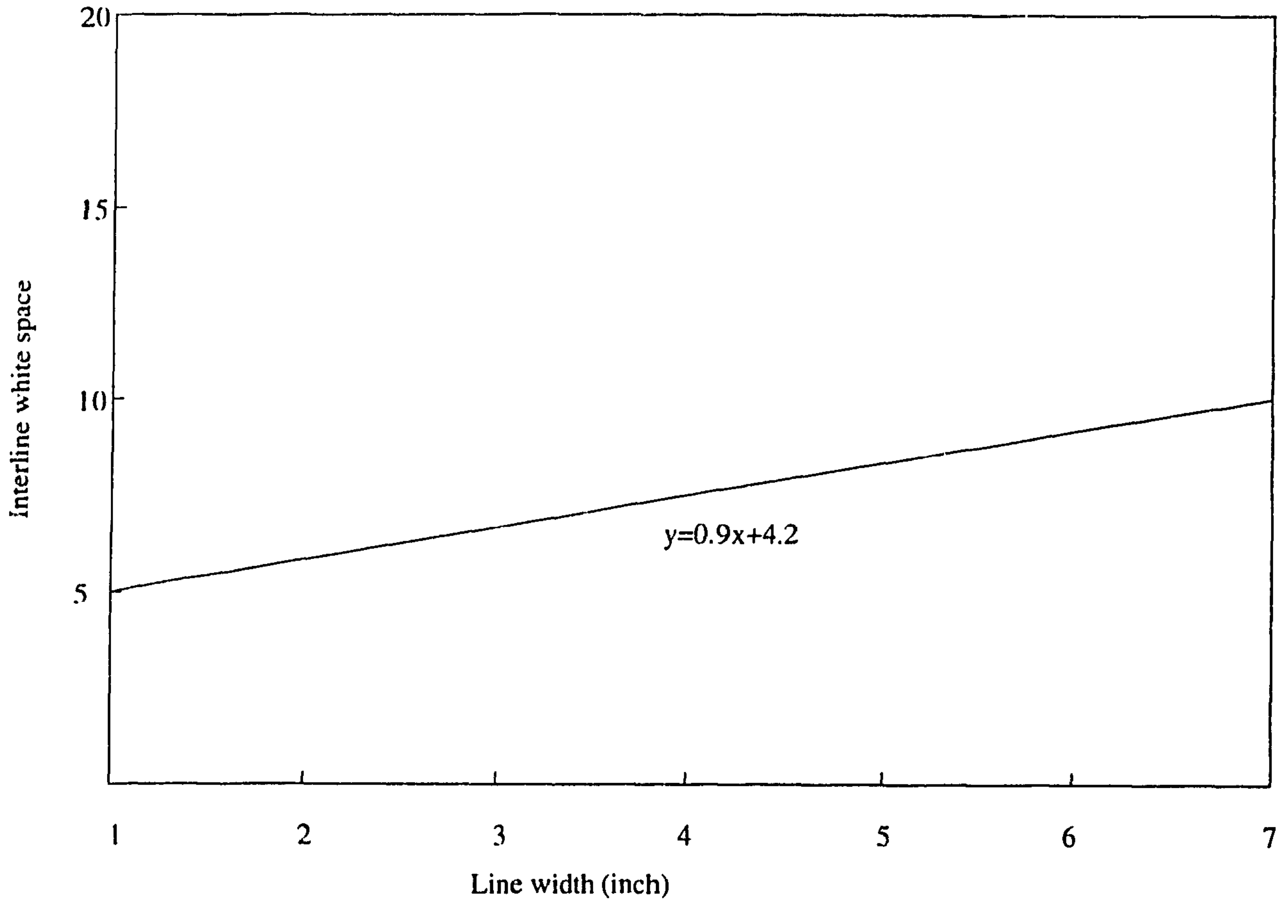


Figure 23: Interline white space as a function of line width

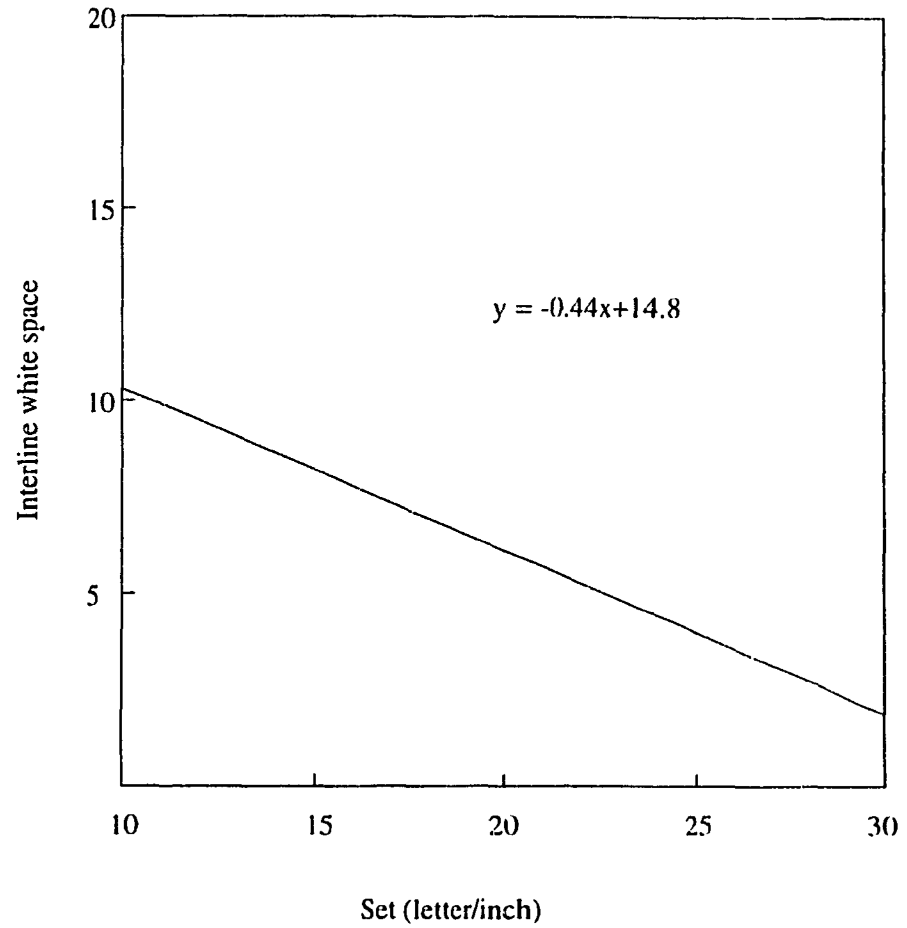


Figure 24: Interline white space as a function of measured average width of characters

Chapter 5

The Information Content of Letterforms

The recognition of letterforms and reading of documents involves the conversion of information from images on papers or screens to symbolic information in the brain. Previous studies show that our visual system has limitation in processing visual information gathered by the eyes. To make typeface or document typesetting legible becomes the problem of designing the letterforms and document typesettings in such a way that the information expressed by them can be absorbed by the visual system most efficiently.

A visual system model is needed in order to understand how the visual information of the letterforms is decoded. The most widely adopted visual system model is the information processing model which will be introduced in the following section.

5.1 The information processing model

The information processing model considers the visual system as a multi-channel filter which detects spatial frequency features of different frequency bands in parallel. Fig. 25 shows how the overall sensitivities of the eyes as the sum of the sensitivities of many overlapping filters. In fact, the existence of these filters has been verified using biological data [8].

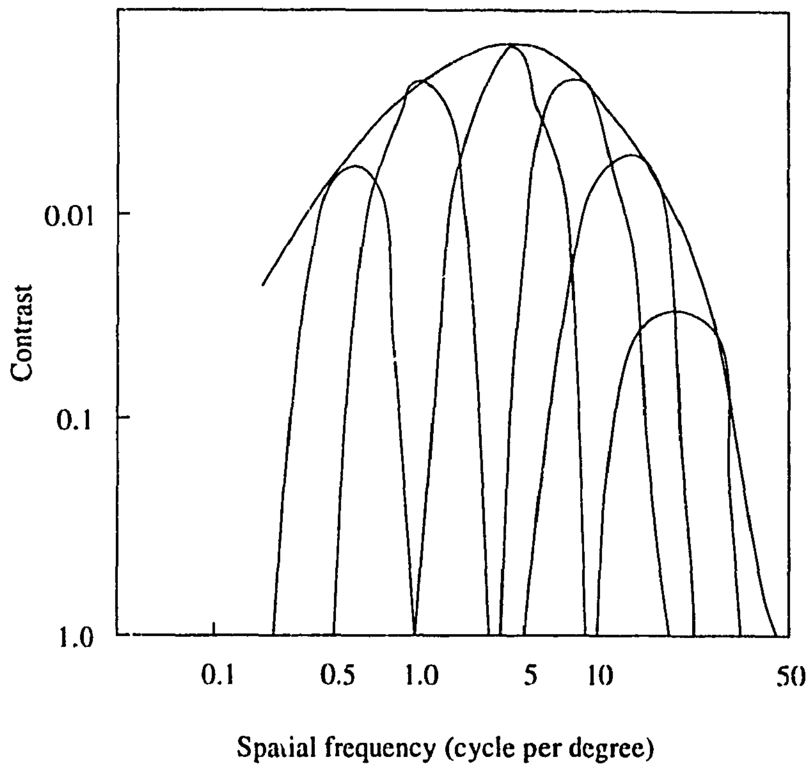


Figure 25: Spatial sensitivity of the eye as the sum of the sensitivity of many overlapping filters

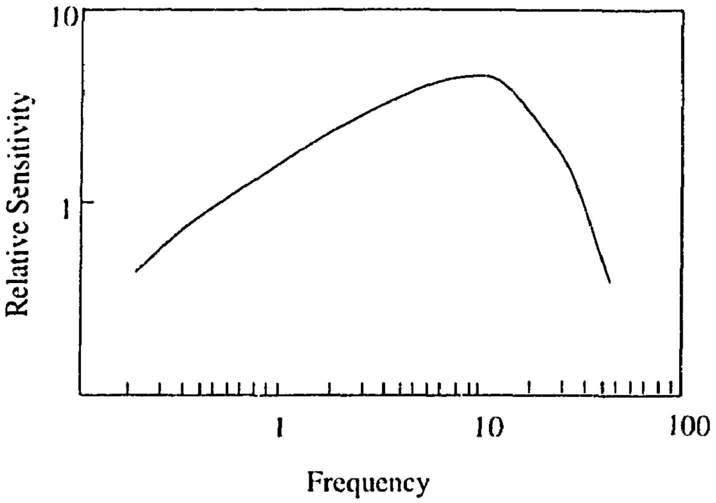


Figure 26: Spatial sensitivity of the eye.

The spatial frequency in Fig. 25 known as the number of cycles per degree of visual angle is related to the sensitivities of the eye. Fig. 26 shows that the human visual system reaches its peak sensitivity when spatial frequency is around 6 - 8 cycles per degree. The sensitivity becomes very low around 50 or 60 cycle per degree, only uniform gray is perceived by most people.

The visual angle is defined as the viewing angle extended by the object. Fig. 27 shows the relationship between visual angle ϕ, distance d and spatial frequency.

Spatial frequency is defined as cycles per degree of visual angle, that is the number of intensity changes from bright to dark and dark to bright within one degree of the visual angle. When the viewing distance is determined, one degree of visual angle can be converted to the absolute length by the following relationship:

$$x = 2d(\tan \phi/2) \tag{15}$$

So cycles per degree of the visual angle can be converted to cycles per inch at certain viewing distance. For example, 60 cycles per degree is about 283 cycles per inch at a viewing distance of 12 inches.

The information processing model is reasonable in several aspects of the letterform recognition:

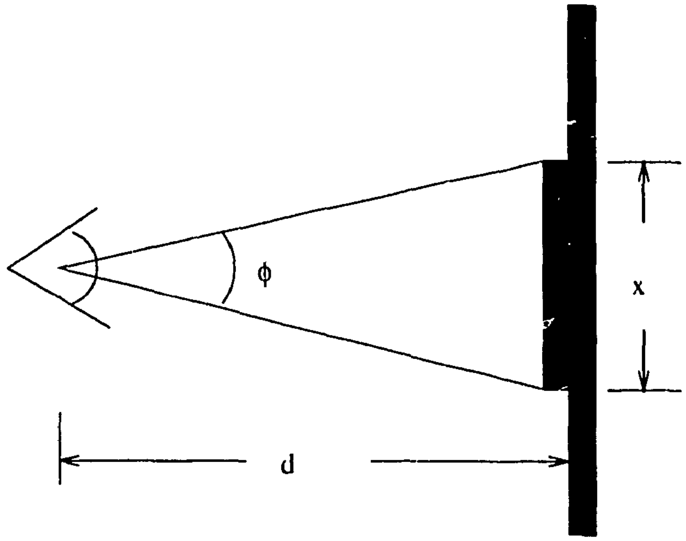


Figure 27: The relationship between visual angle ϕ, viewing distance d and spatial frequency

- The model is consistent with the idea that redundant information makes it easier, not harder to recognize letters. Low frequency information will be enough for recognizing the letter.

- The model also suggests if misinformation happens in one of the visual channels, the recognition may become more difficult even though the information necessary for recognition is present within the signal. However, it does not tell us how the visual system evaluates the importance of the information in each channel and to what extent the information can be modified without seriously disturbing the correct recognition. The human visual system has already demonstrated its outstanding classification skill even when the letterforms are severely distorted. We need a more elaborate model for the human visual system to understand this aspect.

5.2 A proposed visual model for character recognition

We all know that the real difficulties in recognizing letterforms are not caused by the high frequency disturbance on the letterform such as noises introduced by image acquisition, but the low frequency disturbance which causes the shape variations of the letterforms.

There must be fundamental information within each letterform that is vital to its recognition so that as long as this information is present, the human visual system can withstand lots of other distortions if they do not disturb the vital information seriously. A legible typeface or document are those important features which are presented in the way that will attract readers' attention.

In order to know what kind of feature composition of the letterforms attracts the visual attention, a hypothetic visual model is established according to the known characteristics of the human visual system. The proposed model regards the human visual system as a biased feature selector which first evaluates the amount of information presented in the horizontal and vertical visual channels. The visual attention is attracted to the direction which contains more information. The feature selection process is governed by the visual attention, i.e., the features selected from the direction that attracts the attention are more important than those from the direction that does not. The lower-case letters from the alphabet are the research subjects.

The goal is first to determine the primary feature encoding directions of each letterform. Then, the difference between the amount of features presented in the primary encoding directions and the non-primary encoding direction will be evaluated as the parameter which indicates how salient the significant features are presented in the letterform. This parameter will be used with all the other parameters developed in the following chapter to compare the legibility of various typefaces.

The features of interest here have very simple forms. They are white space segments embedded in the letterforms from the vertical direction and the horizontal direction; the profile information from the left, right, top and bottom direction(See Fig. 28). The reasons for selecting the above features are: first, the eyes are most

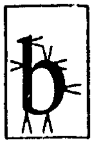


Figure 28: Interesting features in a letterform

Figure 29: The partial images of the lower case letterforms constructed from their primary profiles

sensitive to the outside contour of any shape; and second, the white space embedded in the character is used as the feature separator.

5.2.1 Finding the primary encoding directions

By comparing the complexity of the left, right, top and bottom profiles of each letterform, a maximum of 2 directions will be selected as the primary directions for the its profile information. Fig. 29 shows that the partial images generated from the 2 primary profiles are sufficient for distinguishing them from one another.

The complexity of each profile from the four directions is measured by the following equation:

$$c = -\Sigma p_i * \log p_i \qquad (16)$$

where p is the percentage of the size of the profile segment with respect to the sum of the size of all profile segments. Each profile is broken into increasing, decreasing and flat segments according to the changes of the tangent along the profile path.

The algorithm used to decompose the profile into segments is the same as described in the previous chapter. Table. 2 shows the profile information distribution on lowercase letters of one of the san-serif typeface -Arial.

The l, r, t, and b in the table stand for left, right, top and bottom profiles respectively. For example, for the letter "b", the table indicates that most of the information is in the left and bottom directions. If the profile is a line, for example the left profile

letter	l	r	t	b
a	1.584501	0.890584	1.022952	1.531764
b	0.000000	1.169845	0.662654	1.340133
c	1.313560	1.539106	1.375525	0.968005
d	1.181378	0.000000	1.036459	1.613981
e	1.650062	1.334688	1.244079	1.496726
f	1.078229	1.149610	0.692242	1.096819
g	1.091698	0.811837	1.613896	1.067854
h	0.000000	0.859583	0.928005	1.133060
i	0.000000	0.000000	0.000000	0.000000
j	0.996448	0.260637	0.682792	0.000000
k	0.000000	1.049605	1.144896	1.291450
l	0.000000	0.000000	0.000000	0.000000
m	0.000000	0.000000	1.795133	1.443444
n	0.000000	0.154076	1.394888	1.232093
o	1.235945	1.102992	0.958643	0.988266
p	0.000000	0.973770	1.210854	0.989174
q	1.104008	0.000000	1.403425	1.224224
r	0.000000	0.261160	0.980990	0.442585
s	1.222978	1.576964	1.191185	1.327160
t	1.162458	1.417647	1.355043	0.814925
u	1.097032	0.000000	1.183489	1.540011
v	0.503502	0.465999	1.546414	1.096044
w	1.205344	1.331424	1.981801	1.718206
x	1.075157	1.449438	1.125588	1.454238
y	0.854165	0.223718	1.435721	0.725926
z	1.182396	1.820275	0.000000	0.000000

Table 2: Information distribution of typeface Arial

of the letter 'b', then there is no variation in that direction, hence the amount of information is zero, just as shown in the above table.

The result is consistent with many experiments conducted by typographers. The letters i,j,l,x which are constantly difficult for human to recognize in many experiments conducted by different people either contain little information in any direction or the amount of information on each direction is similar. On the other hand, letters b and d which are easiest for humans to recognize show distinctive profiles in its top and left or right directions.

By investigating 9 most popular typefaces(serif and san serif), we found that each letter in the alphabet has its primary feature encoding directions that do not vary across all these typefaces. It has also been found that as long as the information from the primary encoding direction is preserved, severe distortion such as missing information and nonlinear distortion in the non-primary direction will not affect the recognition of that character. From the testing data, we also find that letters with more distinct profile information, i.e the profile complexity difference between the primary encoding directions and non-primary directions is bigger, the more legible the letter turned out to be. This is understandable because when the shape complexity is greatly different between the primary information and non-primary direction, the feature for recognizing the letter becomes more salient. The following equation is used to measure the salient level for letter's primary profile information.

$$s = 1 - 2\Sigma_{i=0}^{i=np} c_i / \Sigma_{i=0}^{i=nn} c_i$$

where np is the number of primary profiles, nn is the number of all profiles and c_i is the amount of information of a profile.

For each letter, by keeping the information mostly undisturbed in its primary decoding direction. various low frequency disturbances can be added without losing its readability(See Fig. 30).

However, if the font style changes the information in the primary encoding directions of a letterform, it will reduce typeface legibility significantly even though

Figure 30: Characters with distortions

all the necessary information for recognizing the letterform is present. For example, Fonts with too large or too small x-height will all result in a reduction of the salient parameter of the letterform, which in turn reduces the legibility of the letterform. Legibility for letters with ascender, descender and the primary feature directions on the right or left are very sensitive to the size of the x-height.

Chapter 6

The design of digital typefaces

The design of digital typefaces is a highly specialized subject. The typefaces we are using today were designed by a small number of people. Even though we are not going to design typefaces ourselves, it is necessary for us to learn the rules of making legible typefaces so that we can make a good choice for our computer systems.

6.1 Issues in font design

One goal in design is to control the characteristics of letters, such as their weight and x-height, as size and resolution vary. Fonts for low-resolution devices are particularly hard to control because very few pixels are available for creating the shapes of letters. This limits the range of choices for letterform features.

The perception of the size of the font depends mostly on its x-height because most of the letters in the running text are lower-case. In the design, the vertical space is a precious resource because only 40% to 60% percent of the font body size as x-height will be used to design most lower-case letters. For example, a 15 inch computer monitor with 1048x768 resolution actually operates at 70 pixels per inch, or roughly 1 point per pixel. A 12-point font with a 50% x-height left only 6 pixels vertically to design the lower-case letter shapes(Fig.31). And in fact there is also a very limited range to choose the x height, which can only be 5 pixels at the minimum and 7 at the maximum in this particular case.

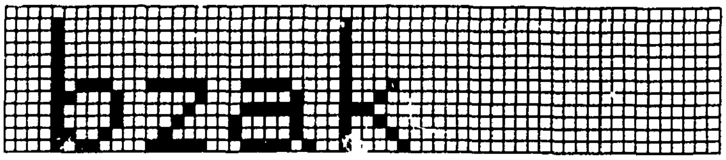


Figure 31: Only a few pixels left to express 12 point fonts at 72 pixels per inch

Design with relatively large x-height are attractive at lower resolution because they make better use of the available space, and appear larger on the screen with the same line height than designs of smaller x-height Fig. 32, [8].

The issues related to the control of the overall appearance of a typeface are listed as the following:

- The color

  The typographic color of a typeface is a measure of its overall blackness. It can be assessed by the following relationship:

$$color = S_{a-z}/T_v \tag{17}$$

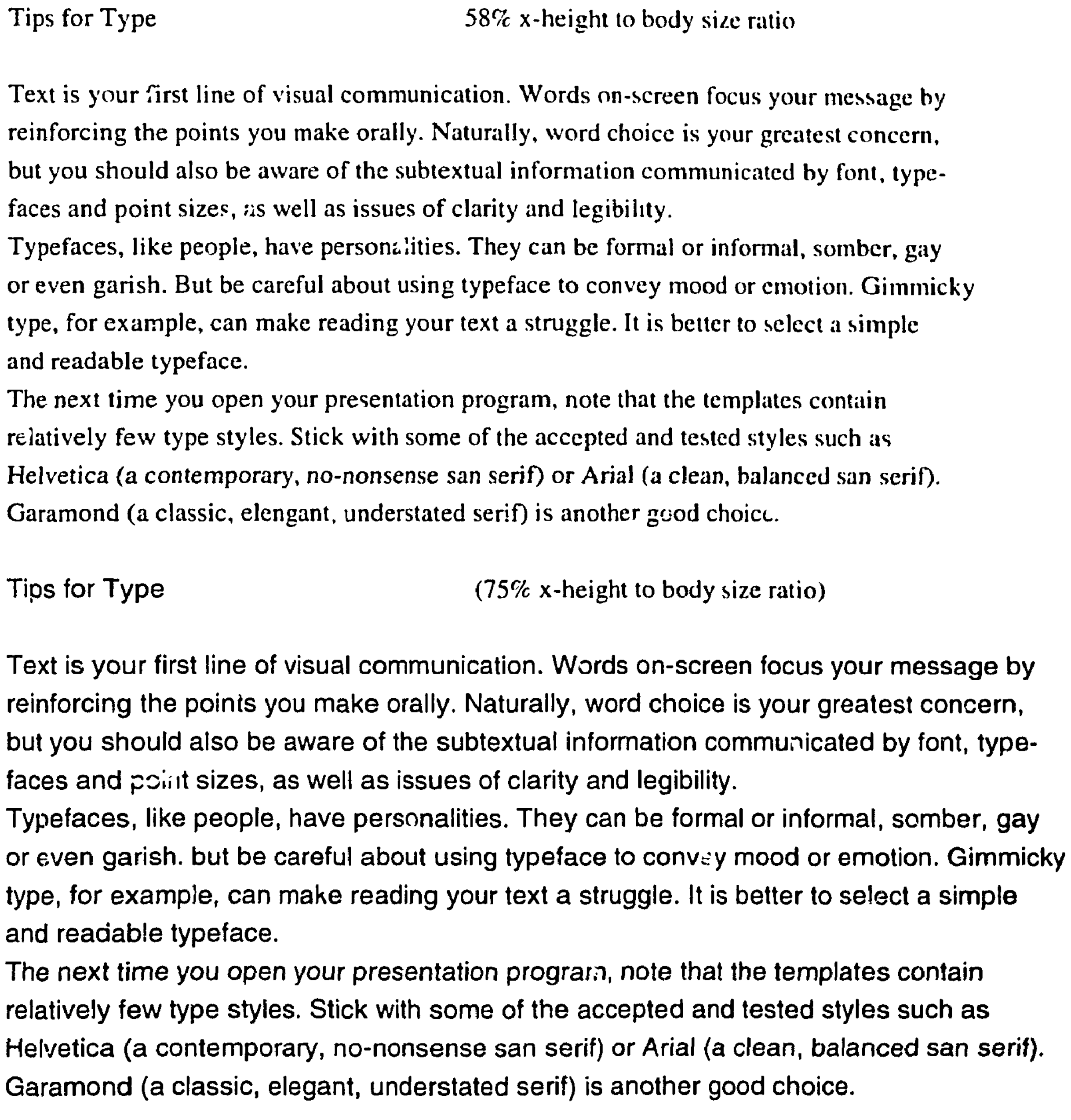
  where S_{a-z} is the horizontal space occupied by a lower-case alphabet and T_v is the vertical stem thickness.

- The weight

  The weight of a typeface is a measure of its heaviness or boldness. It can be expressed as:

$$W = T_v/x \tag{18}$$

  where T_v is the vertical stem thickness and x is the x-height of the typeface. Apparently, the larger the W is, the darker the typeface appears. It is also true that narrow typefaces appear darker than wide type faces. Weights higher than about 1:5 result in darker typefaces that are not generally easy to read. If W drops below about 1:7, the face will appear too light.



Tips for Type 58% x-height to body size ratio

Text is your first line of visual communication. Words on-screen focus your message by reinforcing the points you make orally. Naturally, word choice is your greatest concern, but you should also be aware of the subtextual information communicated by font, typefaces and point sizes, as well as issues of clarity and legibility.
Typefaces, like people, have personalities. They can be formal or informal, somber, gay or even garish. But be careful about using typeface to convey mood or emotion. Gimmicky type, for example, can make reading your text a struggle. It is better to select a simple and readable typeface.
The next time you open your presentation program, note that the templates contain relatively few type styles. Stick with some of the accepted and tested styles such as Helvetica (a contemporary, no-nonsense san serif) or Arial (a clean, balanced san serif). Garamond (a classic, elengant, understated serif) is another good choice.

Tips for Type (75% x-height to body size ratio)

Text is your first line of visual communication. Words on-screen focus your message by reinforcing the points you make orally. Naturally, word choice is your greatest concern, but you should also be aware of the subtextual information communicated by font, typefaces and point sizes, as well as issues of clarity and legibility.
Typefaces, like people, have personalities. They can be formal or informal, somber, gay or even garish. but be careful about using typeface to convey mood or emotion. Gimmicky type, for example, can make reading your text a struggle. It is better to select a simple and readable typeface.
The next time you open your presentation program, note that the templates contain relatively few type styles. Stick with some of the accepted and tested styles such as Helvetica (a contemporary, no-nonsense san serif) or Arial (a clean, balanced san serif). Garamond (a classic, elegant, understated serif) is another good choice.

Figure 32: The appearance of the text using fonts with different x-heights

- The contrast

  Typographic contrast is the ratio of the weights of vertical to horizontal letter elements. It can be expressed as:

  $$C_T = T_v/T_h \tag{19}$$

  where T_v is the vertical stem thickness of lower-case letters and T_h is the horizontal stroke thickness.

  The high legibility of traditional fonts is partly due to the contrast. Extremely high contrast typefaces look striking, but less legible. Lower contrast typefaces such as those with contrast close to unity appears flat all over (Fig. 33).

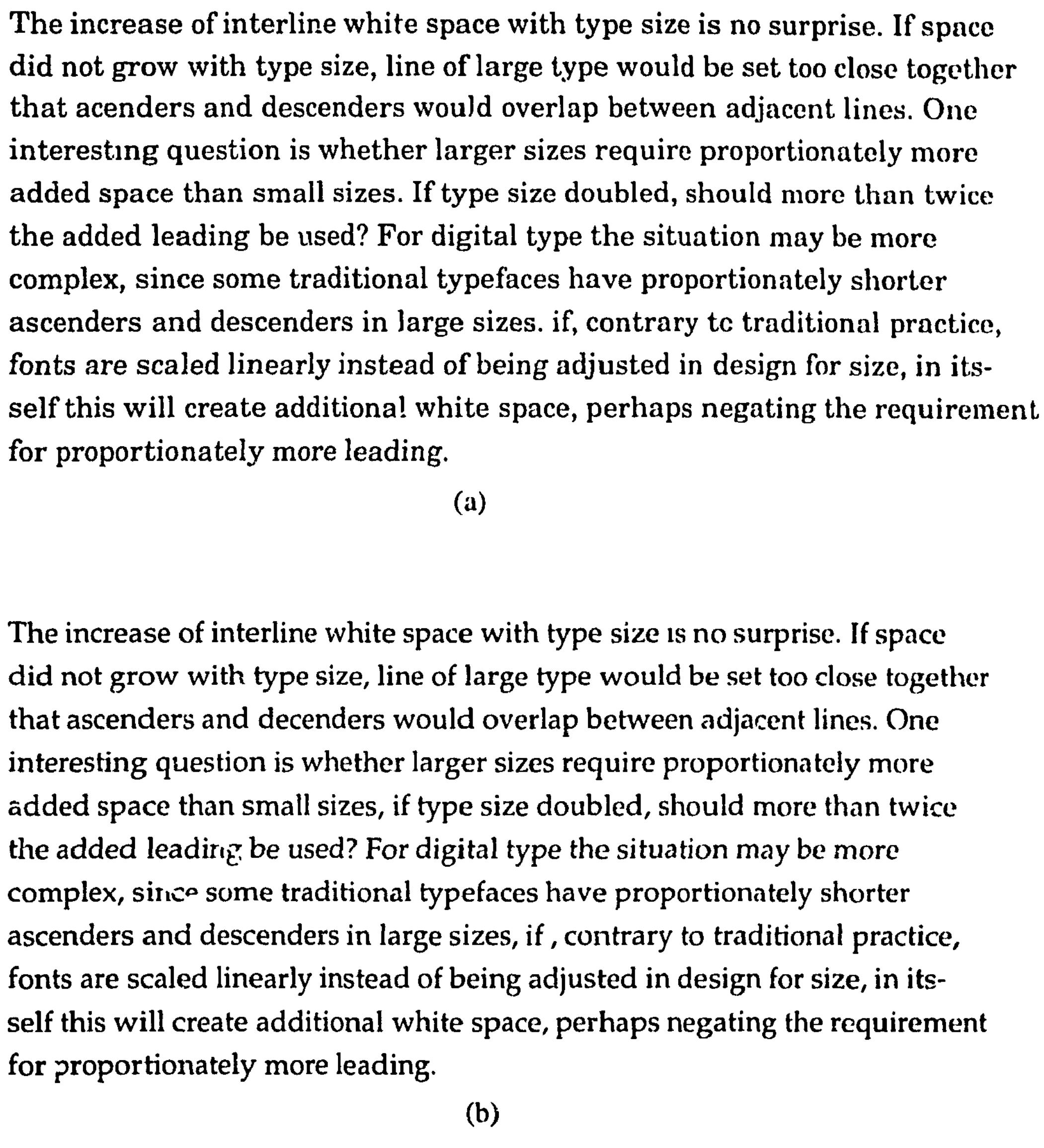
6.2 Determination of the design parameters

There are limited choices of possible combinations of T_v, T_h, and x-height when designing fonts of a particular size.

The following is a table of possible combinations of x-height, T_v and T_h for 10-point font for a 72-pixel per inch display device. From Table. 6.2, the combinations for designing legible 10-point font for a 72-pixel per inch display device are only 2, i.e. the second and the third row which the weight and contrast are close to the desired range.

It is obvious that higher resolution devices are able to provide more legible fonts with more style than low resolution devices even though the absolute size of the letter is the same.

Fig. 34 gives the possible weight and contrast plots for various font sizes. The shaded area represents the ideal range for text typefaces. From these figures, we can see that the number of good combination increases with the number of pixels available to describe the fonts.



The increase of interline white space with type size is no surprise. If space did not grow with type size, line of large type would be set too close together that acenders and descenders would overlap between adjacent lines. One interesting question is whether larger sizes require proportionately more added space than small sizes. If type size doubled, should more than twice the added leading be used? For digital type the situation may be more complex, since some traditional typefaces have proportionately shorter ascenders and descenders in large sizes. if, contrary tc traditional practice, fonts are scaled linearly instead of being adjusted in design for size, in itsself this will create additional white space, perhaps negating the requirement for proportionately more leading.

(a)

The increase of interline white space with type size is no surprise. If space did not grow with type size, line of large type would be set too close together that ascenders and decenders would overlap between adjacent lines. One interesting question is whether larger sizes require proportionately more added space than small sizes, if type size doubled, should more than twice the added leading be used? For digital type the situation may be more complex, since some traditional typefaces have proportionately shorter ascenders and descenders in large sizes, if , contrary to traditional practice, fonts are scaled linearly instead of being adjusted in design for size, in itsself this will create additional white space, perhaps negating the requirement for proportionately more leading.

(b)

Figure 33: Comparison of 2 fonts which have the major difference in contrast (a) set in New Century Schoolbook, and (b) in Palatino. The faces are distinct in style and contrast, but have roughly the same weight, set, and x-height.

T_v	T_h	x-height	W	C_T
1	1	4	0.250	1.000
1	1	5	0.200	1.000
1	1	6	0.167	1.000
1	2	4	0.250	0.500
1	2	5	0.200	0.500
1	2	6	0.167	0.167
1	3	4	0.250	0.333
1	3	5	0.200	0.333
1	3	6	0.167	0.333

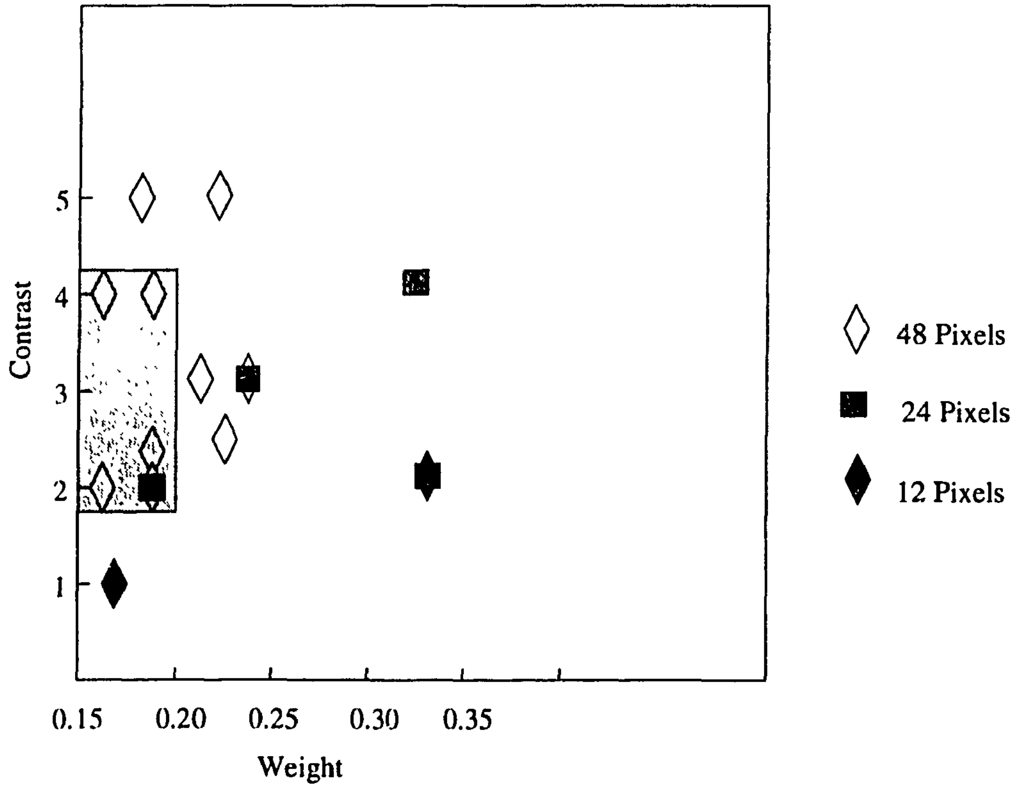


Figure 34: Possible weights and contrasts with 3 cell height.

Face Name	Color	Contrast	Weight	Salient Parm
Arial	202	1	0.139	0.158895
Courier	252	1	0.107	0.428359
Galliard BT	118	5	0.178	0.564321
Helvetica	122	1	0.139	0.458895
Impact	41	14	0.35	0.500210
New Century Sch. Bk.	129	1.66	0.156	0.486565
Palatino	129	2.5	0.156	0.564930
Swis721 Th BT	288	1	0.0625	0.491508
Times New Roman	115	2.5	0.178	0.163159

Table 3: Comparison on 9 typefaces

Contrast, color, weight comparison and feature legibility of nine commonly used typefaces are shown in Table. 3.

Consider the contrast, weight and salient parameters of above 9 typefaces, the order of typefaces according to their legibility is as following: Palatino, New Century Sch. Bk., Times New Roman, Helvetica, Arial, Courier, Galliard BT, Swis721 Th BT, and Impact. The criteria used are: (1) the font has to satisfy the optimum weight and contrast range, and (2) they will be further divided by their salient parameters.

6.3 Font description methods

There are lots of digitized typefaces available. They may be described in different ways. All these methods have to weigh trade-offs among precision, scalability and storage requirement. The followings presents some of the existing methods have been developed.

6.3.1 Bitmap method

Fonts are described and manipulated as explicit bitmaps(Fig. 31). This is the most straight forward method. Digital fonts can be generated easily through digitizing printed fonts. The disadvantage is that storage requirement increases as the square of the font size. Aliasing may arise from scaling(Fig. 35).

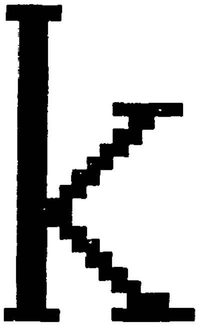


Figure 35: The bitmap font after scaling

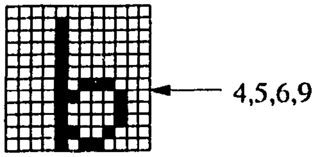


Figure 36: The bitmap of a font and its run-length code

6.3.2 Run-length code

This is essentially a bitmap method. However, the bitmap is encoded as run-length code, so that the storage requirement grows only proportionally to the font size. In run-length encoding, the computer calculates and stores counts of ones and zeros, rather than the actual bits themselves(See Fig. 36). The method works well with font images because there are only a few black and white segments per scan line in a font image and the number does not increase with font size. Fonts encoded this way need to be decoded before they are displayed. Since this method is just an encoding method for bitmaps, so it has the same aliasing problem after scaling.

6.3.3 Vector

The vector method stores fonts as a list of lines which form the font shape. This technique originated in the earlier days when most output devices drew only lines and points. For example, the Hershey fonts [8] are public-domain fonts designed for

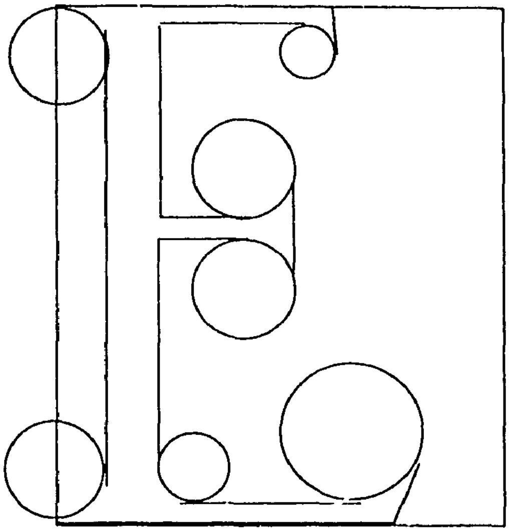


Figure 37: An example of a font expressed as vectors and arcs

vector output devices. When these fonts are displayed on a bitmap output device, they must be rasterized, that is, the lines must be rendered in a bitmap array and then presented on a screen or on paper.

6.3.4 Arc and vector

Arc and vector method uses circular arcs and straight lines to describe the outlines of fonts(See Fig. 37). Because only the points on the outlines of the fonts are recorded, less storage is required to encode the shape accurately. This method needs a good approximation algorithm to select the boundary points to describe the font. It also needs to rasterize the boundary points and fill the enclosed shapes into bitmap forms before the font can be displayed. It exhibits satisfactory result for scaling down fonts. Aliasing is still a problem when scaling up because sometimes the joints between vectors and arcs can not maintain continuity after scaling.

6.3.5 Spline

The most compact way to store font data is to use splines to describe a given font. Spline method also generates so far the nicest looking fonts. Fonts used in most

Figure 38: The scaling of New Times Roman(True Type)

desktop publishing system are described by spline method, because it can provide high quality fonts of many different sizes by scaling. In addition, spline curves can also be manipulated interactively, so that systems that produce these fonts can be made.

Mathematically, splines are piecewise polynomial parametric curves when they are linked together under certain continuity constraints. The general form of the piecewise polynomial can be expressed as:

$$x(t) = a_0 + a_1 * t + a_2 * t^2 + ... + a_{n-1}t^{n-1} + a_n t^n \tag{20}$$

$$y(t) = b_0 + b_1 * t + b_2 * t^2 + ... + b_{n-1}t^{n-1} + b_n t^n \tag{21}$$

where a_i and b_i are coefficients and n is the order of the polynomials.

The coefficients are determined so that the piecewise polynomials are continuous at the joint. In general, a spline of order n has continuity in the (n-1) derivative at each point.

Bezier splines are one important class of cubic splines. Because Bezier curves can be used to describe font shape efficiently, relatively fast to compute and offer both kinds of user interaction – on-curve and off-curve control points – they become very popular in computer-aided designing systems.

A simple formula can be expressed as:

$$z_t = (1-t)^3 z_1 + 3(1-t)^2 t z_2 + 3(1-t)^2 z_3 + t^4 z_4 \tag{22}$$

where z_i are the control points (x_i, y_i) and t is the parameter ranging from 0 to 1.

Both True Type fonts and Adobe's Type I fonts use cubic splines. The font looks very nice after scaling Fig. 38.

6.4 Supporting software systems for designing typefaces

Many systems have been developed for designing fonts. However, they are mostly used for in-house design by typeface design companies such as Adobe. Commercial font design tools are scarcely available mostly because font design is such a highly specialized and difficult task that it needs many years of expertise and a good aesthetic sense. Only a few of commercial font editors such as Fontastic on Mac and Microsoft's Font Editor provide a number of tools for editing font bitmaps, or Metafont an algorithm based font designer are available.

6.4.1 Bitmap tools

This type of tools support creation and edition of font bitmaps. It is useful for all methods of typeface design.

There are small sets of utilities for the design purposes, e.g. editing commands such as:

- Draw a pixel or pixels by different pen.
- Draw a line between 2 points by different pen.
- Draw a circle.
- Draw a curve specified by a set of control points.
- Erase a symbol or part of a symbol.
- Fill a closed region.
- Adjust side-bearings.
- Cut, copy and paste a region of a symbol.
- Merge two symbols.

- Copy a symbol or a set of symbols.

Global Operations:

- Scale a font up or down.
- Create and adjust guide marks.
- Set a background symbol or image.
- Change the weight of a font.
- Slant a font to obtain an oblique version
- Set sample text.

6.4.2 Outline tools

Outline tools provide very convenient ways to create shapes independent of size and resolution. Designers are freed from the constraint of designing within certain space so that they can concentrate on designing font shapes. Specific size and resolution can be obtained by scaling.

In addition to interactive curve editor, there are also other systems such as Karow's Ikarus [4] which create outlines from digitizer input, or rastering programs that convert an outline to its equivalent bitmap form.

Although the outline tools are much better than bitmap tools in making individual shapes, they do have shortcomings. One difficulty with editing outlines is that the inside and outside boundaries must be coordinated. Although in most parts of letters, the inside and outside boundaries are parallel or symmetric, it is not always true(See Fig. 39). Besides, it is also hard to copy parts of a letter to another because the shape is defined by control points. For instance, it is tough to make a consistent serif design for the whole character set.

Figure 39: The inner and outside contours of letter 'b' are not parallel to each other

6.4.3 Component tools

The component tools are based on the ideas of decomposing a letter into generic components such as those shown in Fig. 40. The font shape then is composed by assembling its components. The design of the components can be created by either bitmap oriented tools or outline tools. The font design system I had worked on is based on this idea.

The component tools have been proven to be very effect in making typefaces on symbol sets which have lots of duplicated components, for instance, the oriental fonts. They are also very convenient tools for creating fonts with similar serif features, or derive creative styles by cross combining serif features and font structure features of many different fonts.

The font design system I designed has the following main features:

- Bitmaps tools

  The bitmap tools provide the full range of command for bitmap editing listed above and with the following additional commands to make shape editing easier.

  - Thinning of the shape on the left side or the right side
  - Thinning the shape on the left side or the right side between specified range of rows of pixels
  - Thickening of the shape on the left side or the right side
  - Thickening the shape on the left side or the right side between specified range of rows of pixels
  - Smoothly shortening the shape
  - Smoothly expanding the shape

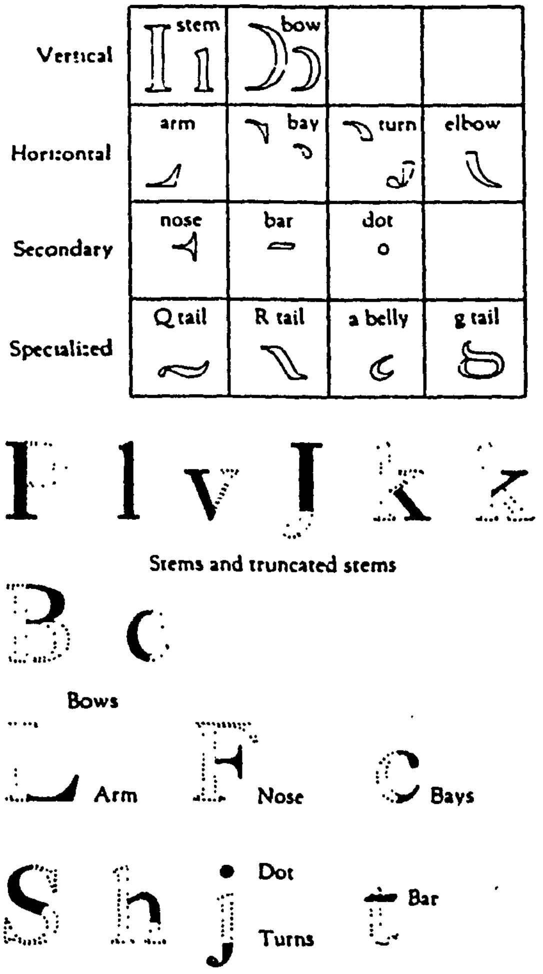


Figure 40: Font components of the letterforms in the alphabet

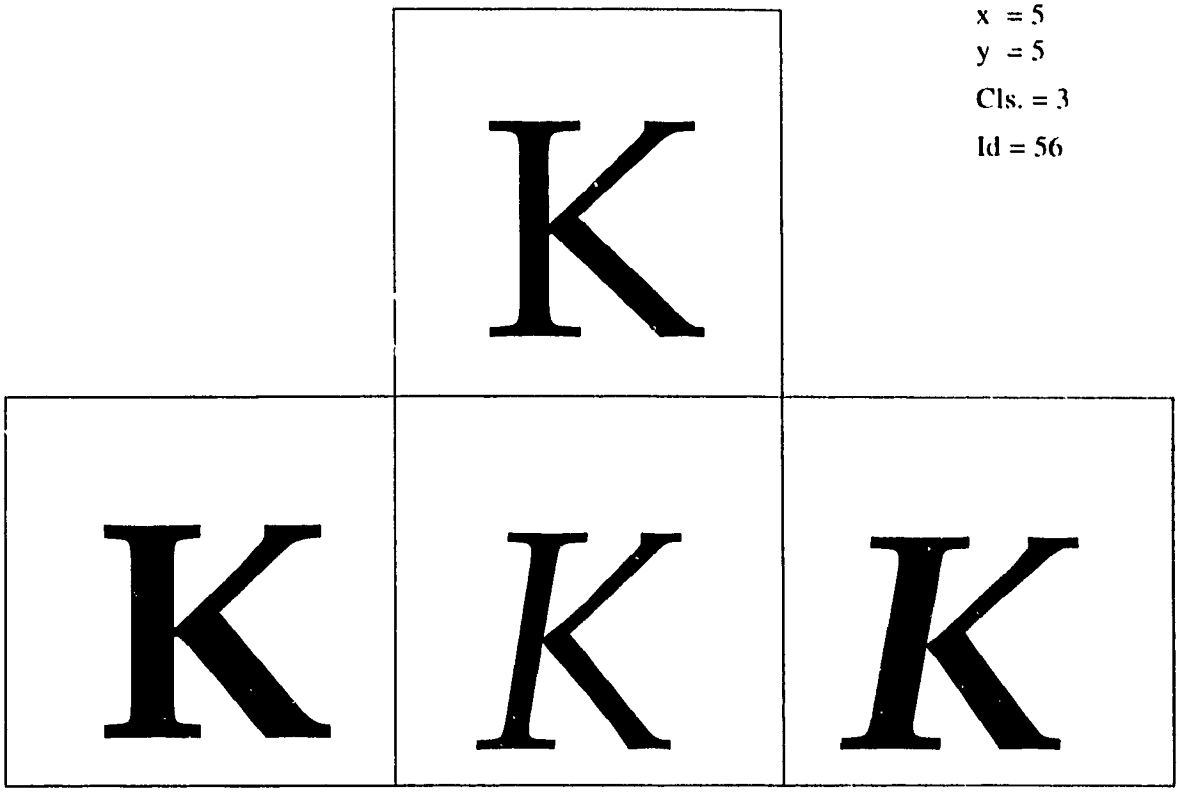


Figure 41: The composition window

- Main composition screen

  The main composition screen contains 4 small windows each of which contains the character of different weight that is under editing (see Fig. 41).

  The components can be selected by using the page-up and page-down keys. When the component is selected, the user can use up, down, left and right key to position the selected component. The complete set of commands consist of the following:

  - Select a component
  - Select a number of components
  - Moving the selected components
  - Moving rm the selected components by entering the distance to be moved

- Component database

  There could be thousands of components in the database, especially when designing oriental fonts. It will be very tiring if the user has to go through all the available components in order to pick the right one. There are small sets of commands in the system used to manage the component database.

  - Query for the component classes

    The designer can enter the id of a component class to retrieve a list of all the components in that class and they will be displayed on the screen graphically.

  - Query for a group of components

    The designer can also retrieve a group of components by specifying the class id of the component and the size of the component.

  - Query for all the characters which use a particular component

    Another very useful query is to retrieve all the characters which use a particular component. This is very helpful when the designer is modifying an existing component in the database because he can check the effect of the change on all the other characters.

  - Insert a new component

    After being created, a new shape can be inserted to the database by giving it an id to identify the class of the shape. The index of the component in that class will be automatically provided.

  - Delete an existing component

    Deleting a component is a little tricky, because there is a possibility that the component has been referenced more than once. In fact, each component is reference counted so that only when no other characters is using that component, it will be deleted from the database.

- Selecting the right components

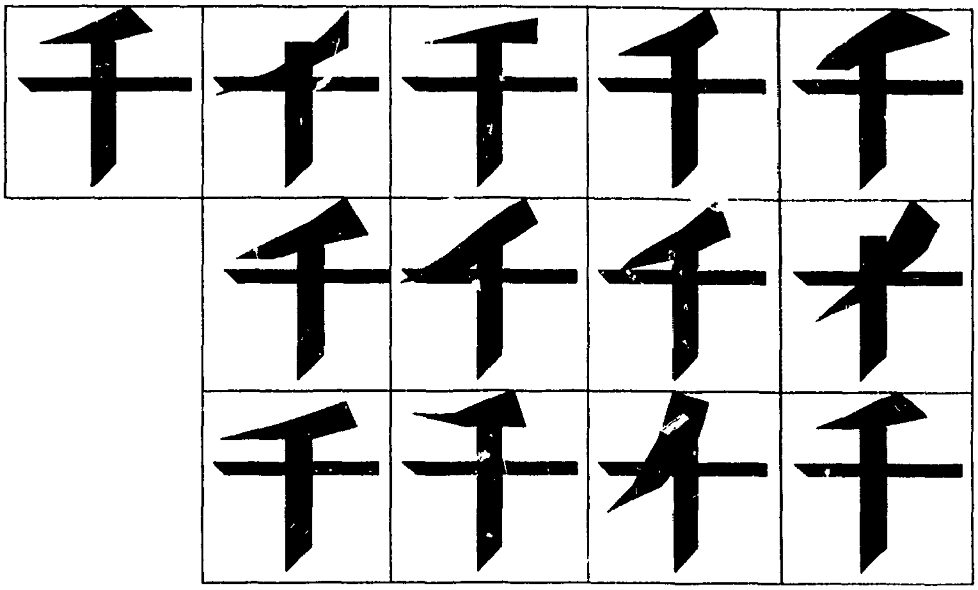


Figure 42: The component selection window

The selection of the right components for the character under construction is done by comparing the visual effect of putting the components on the character. The components can be filtered by specifying the size of their bounding box. Fig. 42 shows the screen display designed to help the selection. The stroke selected by the user has been picked and shown in the window on the left.

Chapter 7

Conclusion

In this thesis, the major parameters which control the visual appearance of the typefaces have been summarized. The relationships among x-height, horizontal stroke width, vertical stroke width, weight and contrast will help designer to choose valid combinations in new designs. The visual model developed in the thesis and the measurement of salient features provide useful tools for comparing the legibility of various typefaces and guide the designer to make the right decisions on the letter shape. This study only investigated lower case letters of 12 point size for 300 dpi devices, the result is still useful because the majority of running texts are composed by fonts of this size.

From the research, we found that each letter had its primary directions in which most of its features are encoded. This knowledge can be applied to select the distinctive features for OCR systems and help to speed up the recognition program. For example, according to the primary encoding directions of each pattern such as left/right, top/bottom, left/bottom, right/bottom, left/top, right/bottom and unknown, the patterns can be classified into subgroups. Then, a more elaborate classification scheme can start from here on a smaller set of patterns. This method makes sense because that is how we recognize letters when reading a piece of text, not by resolving all the details of each letter, but through rough assessment of the location of the information that catches the eye and the overall word form. It is also inevitable that the future pattern systems have to become OWR(Optical Word Recognition) that combines word recognition with current character recognition in order to break through the current speed and accuracy limits.

Chapter 8

The Information Distribution of Letter Forms

letter	l	r	t	b
a	1.584501	0.890584	1.022952	1.531764
b	0.000000	1.169845	0.662654	1.340133
c	1.313560	1.539106	1.375525	0.968005
d	1.181378	0.000000	1.036459	1.613981
e	1.650062	1.334688	1.244079	1.496726
f	1.078229	1.149610	0.692242	1.096819
g	1.091698	0.811837	1.613896	1.067854
h	0.000000	0.859583	0.928005	1.133060
i	0.000000	0.000000	0.000000	0.000000
j	0.996448	0.260637	0.682792	0.000000
k	0.000000	1.049605	1.144896	1.291450
l	0.000000	0.000000	0.000000	0.000000
m	0.000000	0.000000	1.795133	1.443444
n	0.000000	0.154076	1.394888	1.232093
o	1.235945	1.102992	0.958643	0.988266
p	0.000000	0.973770	1.210854	0.989174
q	1.104008	0.000000	1.403425	1.224224
r	0.000000	0.261160	0.980990	0.442585
s	1.222978	1.576964	1.191185	1.327160
t	1.162458	1.417647	1.355043	0.814925
u	1.097032	0.000000	1.183489	1.540011
v	0.503502	0.465999	1.546414	1.096044
w	1.205344	1.331424	1.981801	1.718206
x	1.075157	1.449438	1.125588	1.454238
y	0.854165	0.223718	1.435721	0.725926
z	1.182396	1.820275	0.000000	0.000000

Table 4: Letter information distribution of typeface Arial

letter	l	r	t	b
a	1.396401	0.674699	1.191566	1.373898
b	0.682908	0.892113	0.780756	1.585397
c	1.122372	1.034215	1.219813	0.994927
d	1.288396	0.000000	0.727282	1.477414
e	1.120911	1.552201	0.925601	1.234245
f	1.376966	1.389557	1.039565	0.000000
g	1.082927	0.614989	1.594325	1.162936
h	0.585953	1.007931	1.060745	1.078288
i	1.204564	0.692659	0.685210	0.000000
j	1.163925	0.328791	1.032523	0.257319
k	0.855706	1.076777	1.107723	0.917547
l	0.867938	0.000000	0.673012	0.000000
m	0.968196	0.000000	1.444911	1.379629
n	0.968196	0.693147	0.993285	1.078288
o	1.189925	0.986728	0.986728	1.189925
p	0.855706	1.480777	1.275747	0.944106
q	1.458661	0.562335	1.627731	1.073894
r	0.894337	0.759724	0.927257	0.666278
s	1.106505	1.523665	1.254854	0.687092
t	1.149281	1.236525	1.162184	0.962255
u	0.654670	0.000000	1.352576	1.185909
v	0.796576	0.835743	1.067668	1.539502
w	1.496875	1.763974	1.410246	1.985285
x	1.231167	1.251603	0.811005	0.723245
y	1.255832	0.955636	1.173053	0.992950
z	1.004006	1.222389	0.000000	0.000000

Table 5: Letter information distribution of typeface Courier

letter	l	r	t	b
a	1.726952	0.856998	0.950971	1.517887
b	0.000000	1.822230	1.604080	1.145973
c	1.459870	1.314704	1.060839	1.347168
d	1.743233	0.000000	0.961733	1.483738
e	1.334451	1.484769	1.189247	1.211258
f	0.000000	1.823233	1.165212	0.417388
g	1.922307	1.624774	1.069309	0.785404
h	0.000000	2.223784	2.013978	1.075712
i	1.201816	0.651757	1.059196	0.000000
j	1.396487	1.232544	0.981077	0.673012
k	0.000000	1.845632	1.727021	1.366071
l	0.000000	0.000000	1.509599	0.000000
m	0.843587	0.773150	1.994900	1.563405
n	0.882383	0.716141	1.610973	1.062869
o	1.001864	0.941595	1.161196	1.082790
p	0.573639	1.160623	1.393952	0.798477
q	1.368806	0.682908	1.183498	0.816496
r	0.730167	0.979872	1.452281	1.041561
s	1.656391	1.523703	0.997716	0.801819
t	0.690457	1.561651	1.023547	0.992784
u	0.721464	0.721464	1.732835	1.612435
v	0.654741	0.585953	1.041219	1.253915
w	0.651757	0.596852	1.305550	1.980808
x	0.921212	1.395252	1.065509	1.020853
y	1.061476	0.226638	1.036112	0.802865
z	1.097543	1.337780	0.000000	0.000000

Table 6: Letter information distribution of typeface Galliard

letter	l	r	t	b
a	1.584501	0.890584	1.022952	1.531764
b	0.000000	1.169845	0.662654	1.340133
c	1.313560	1.539106	1.375525	0.968005
d	1.181378	0.000000	1.036459	1.613981
e	1.650062	1.334688	1.244079	1.496726
f	1.078229	1.149610	0.692242	1.096819
g	1.091698	0.811837	1.613896	1.067854
h	0.000000	0.859583	0.928005	1.133060
i	0.000000	0.000000	0.000000	0.000000
j	0.996448	0.260637	0.682792	0.000000
k	0.000000	1.049605	1.144896	1.291450
l	0.000000	0.000000	0.000000	0.000000
m	0.000000	0.000000	1.795133	1.443444
n	0.000000	0.154076	1.394888	1.232093
o	1.235945	1.102992	0.958643	0.988266
p	0.000000	0.973770	1.210854	0.989174
q	1.104008	0.000000	1.403425	1.224224
r	0.000000	0.261160	0.980990	0.442585
s	1.222978	1.576964	1.191185	1.327160
t	1.162458	1.417647	1.355043	0.814925
u	1.097032	0.000000	1.183489	1.540011
v	0.503502	0.465999	1.546414	1.096044
w	1.205344	1.331424	1.981801	1.718206
x	1.075157	1.449438	1.125588	1.454238
y	0.854165	0.223718	1.435721	0.725926
z	1.182396	1.820275	0.000000	0.000000

Table 7: Letter information distribution of typeface Helvetica

letter	l	r	t	b
a	1.500943	0.154076	0.870780	1.401277
b	0.000000	0.802283	0.823970	0.907535
c	1.470817	1.407260	1.176818	1.290952
d	0.785558	0.000000	0.970648	1.354885
e	0.788197	1.211272	1.171363	0.890777
f	0.764754	0.986933	0.669328	0.594489
g	0.802944	0.440852	1.300990	0.876199
h	0.000000	0.801489	1.048286	0.578534
i	0.000000	0.000000	0.000000	0.000000
j	0.538681	0.297229	0.000000	0.000000
k	0.000000	1.385592	1.144327	1.322124
l	0.000000	0.000000	0.000000	0.000000
m	0.000000	0.000000	1.493725	1.113791
n	0.000000	0.000000	0.908616	0.587827
o	0.759547	0.661613	0.940744	0.955247
p	0.000000	0.677361	0.958667	1.074490
q	0.758561	0.000000	1.201350	1.264378
r	0.000000	0.706991	0.977230	0.883533
s	1.613265	1.511308	0.961286	1.278646
t	1.303947	1.348355	0.616385	0.684722
u	0.666278	0.000000	1.079596	1.374776
v	1.804927	1.804927	1.326718	1.082353
w	1.598398	1.565014	1.495086	1.487228
x	1.806758	1.806758	1.083768	1.042109
y	2.021077	1.797194	1.149983	0.807001
z	0.988603	0.722887	0.000000	0.000000

Table 8: Letter information distribution of typeface Impact

letter	l	r	t	b
a	1.657372	0.753442	1.123629	1.742327
b	0.562335	1.026763	1.378151	1.038652
c	1.068263	1.370140	1.260690	1.126013
d	1.368957	1.060857	1.522918	1.311704
e	1.257929	1.614021	1.082583	1.401803
f	0.994924	1.012590	1.121752	0.265797
g	2.349259	1.747687	1.060725	1.242664
h	0.445144	0.973409	1.321432	1.065624
i	1.015183	0.500402	0.719304	0.000000
j	0.942007	0.385682	1.161884	0.900256
k	0.445144	1.393983	1.540837	1.019769
l	0.445144	0.500402	1.091246	0.000000
m	0.603797	0.697075	2.185479	1.357556
n	0.603797	0.697075	1.679585	1.065624
o	1.349088	1.126885	0.959615	1.342113
p	0.271189	1.257318	1.769521	0.869799
q	1.360686	0.500402	1.523794	0.905287
r	0.366925	0.509459	1.362447	0.903262
s	1.427426	1.310747	0.970959	0.683739
t	1.238177	1.208734	0.464357	0.697684
u	0.666278	1.060857	1.415383	1.736028
v	0.432714	0.527651	1.083409	1.286480
w	0.642657	0.947057	1.642377	1.916480
x	0.859689	1.272143	0.000000	0.000000
y	1.082296	0.891784	0.842149	1.476547
z	0.859689	1.272143	0.000000	0.000000

Table 9: Letter information distribution of typeface New Century School Book

letter	l	r	t	b
a	1.657372	0.753442	1.123629	1.742327
b	0.562335	1.026763	1.378151	1.038652
c	1.068263	1.370140	1.260690	1.126013
d	1.368957	1.060857	1.522918	1.311704
e	1.257929	1.614021	1.082583	1.401803
f	0.994924	1.012590	1.121752	0.265797
g	2.349259	1.747687	1.060725	1.242664
h	0.445144	0.973409	1.321432	1.065624
i	1.015183	0.500402	0.719304	0.000000
j	0.942007	0.385682	1.161884	0.900256
k	0.445144	1.393983	1.540837	1.019769
l	0.445144	0.500402	1.091246	0.000000
m	0.603797	0.697075	2.185479	1.357556
n	0.603797	0.697075	1.679585	1.065624
o	1.349088	1.126885	0.959615	1.342113
p	0.271189	1.257318	1.769521	0.869799
q	1.360686	0.500402	1.523794	0.905287
r	0.366925	0.509459	1.362447	0.903262
s	1.427426	1.310747	0.970959	0.683739
t	1.238177	1.208734	0.464357	0.697684
u	0.666278	1.060857	1.415383	1.736028
v	0.432714	0.527651	1.083409	1.286480
w	0.642657	0.947057	1.642377	1.916480
x	1.767378	1.596126	0.804447	1.172348
y	1.215352	0.836526	0.842149	1.341237
z	0.859689	1.272143	0.000000	0.000000

Table 10: Letter information distribution of typeface Times New Roman

letter	l	r	t	b
a	1.257673	0.636514	1.211405	1.175205
b	0.000000	1.923000	1.860870	1.389410
c	1.170601	1.339551	1.191760	0.859967
d	1.916532	0.000000	2.062913	1.217010
e	1.218195	1.040927	0.886464	0.972481
f	0.000000	0.000000	0.912307	0.660298
g	2.162539	1.785177	1.095486	0.858231
h	0.000000	0.000000	1.728812	1.231308
i	1.320199	0.450561	0.815029	0.000000
j	1.414046	1.312530	0.443987	0.825791
k	0.000000	1.875389	1.947710	1.336140
l	0.000000	0.000000	1.331861	0.000000
m	0.740066	0.661563	1.446578	1.565213
n	0.740066	0.661563	1.212077	1.231308
o	1.260046	1.261426	0.923208	1.322068
p	0.577992	1.269181	1.539369	0.790129
q	1.495892	0.687092	1.228717	1.168929
r	0.752792	1.214524	1.280882	0.755945
s	1.161345	1.171771	1.036106	0.325083
t	1.380109	1.381482	1.106474	1.017456
u	1.067090	0.304636	1.807233	1.205614
v	0.458144	0.583414	1.079752	1.418265
w	0.795684	0.433399	1.617925	1.982310
x	1.952704	1.339532	1.895122	1.578355
y	0.585953	0.579196	1.075959	1.135189
z	1.074820	1.927340	0.421581	0.000000

Table 11: Letter information distribution of typeface Palatino

letter	l	r	t	b
a	1.610053	0.598270	1.137884	1.566707
b	0.000000	1.039250	0.917402	1.451734
c	0.948267	1.601589	0.981087	1.096361
d	0.980301	0.000000	1.084432	1.220047
e	1.567236	1.379930	1.219681	1.018047
f	0.905985	0.693147	0.690028	1.022342
g	1.110251	0.541848	1.294978	1.411760
h	0.000000	0.836611	1.056655	1.045335
i	0.000000	0.000000	0.000000	0.000000
j	0.353169	0.099623	0.664064	0.000000
k	0.000000	1.107714	1.089688	1.026250
l	0.000000	0.000000	0.000000	0.000000
m	0.000000	0.154076	1.763498	1.345187
n	0.000000	0.154076	1.077573	1.045335
o	1.567236	1.170274	0.942216	0.960086
p	0.000000	1.067508	1.096264	0.808905
q	1.015109	0.000000	1.294978	0.901330
r	0.000000	0.278769	0.917402	0.676727
s	1.168956	1.906783	1.018722	1.451291
t	0.926334	0.833365	0.983961	0.682908
u	0.682908	0.000000	1.011721	1.110934
v	0.576334	0.531183	1.159728	0.877572
w	0.555857	0.497162	1.893163	1.463414
x	1.319484	1.678580	1.465571	1.370000
y	0.783222	0.227268	1.255774	0.718195
z	2.073554	2.052619	0.000000	0.000000

Table 12: Letter information distribution of typeface Swiss721 Th BT

Times New Roman:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Palatino:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Arial:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Courier:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Helvetica:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Impact:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

New Century School Book:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Swis721 Th BT:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Galliard BT:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Figure 43: Sample text of the 9 most commonly used typefaces

Bibliography

[1] David J. Field and Anthony Hayes. Contour integration by the human visual system: Evidence for a local "associate field". *Vision Res.*, 33(2):173–193, 1992.

[2] H. Freeman and L.S. Davis. Angle detection on digital curves. *IEEE Trans. Comput.*, 26:297–303, March 1977.

[3] L. Glass. Moire effect from random dot. *Nature*, 243:578–580, 1969.

[4] P. Karow. *Ikarus-System: computer-controlled font production for CRT and Lasercomp.* Hamburg, Germany: Karow Rubow Weber GmbH, 1979.

[5] D. G. Lowe. Organization of smooth image curves at multiple scales. In *Proc. 2nd Int. Conf. Computer Vision*, pages 558–567, 1988.

[6] T. Pavlidis. *Structural Pattern Recognition.* Springer-Verlag, 1977.

[7] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Trans. Comput.*, 22:875–878, September 1973.

[8] R. Rubinstein. *Digital Typography.* Addison-Wesley Publishing Company, 1988.

[9] M. A Spencer. *The Visible World.* Iowa: Iowa State University Press, 1963.

[10] C. Y. Suen. Character recognition by computer and applications. In K. S. Fu T. Y. Young, editor, *Handbook of Pattern Recognition and Image Processing.* Orlando: Academic Press, 1986.

[11] C. Y. Suen and M. K. Komoda. Legibility of digital type-fonts and comprehension in reading. In J. C. van VLIET, editor, *Proc. of the International Conference on Text Processing and Document Manipulation.* Cambridge University Press, 1986.

[12] C. Y. Suen and C. Shiau. An iterative technique of selecting an optimal 5x7 matrix character set for display in computer output systems. In *Proc. soc. Infor. Display*, pages 85 88, 1986.

[13] Cho-Huak Teh and Roland T. Chin. On detection of dominant points on digital curves. *IEEE Trans. on Pat. Anal. and Mach. Intell.*, 11(8):859–872, 1989.

[14] M. A Tinker. *Legibility of Print.* Iowa: Iowa State University Press, 1963.

[15] N. Ueda and S. Suzuki. Learning visual models from shape contours using multiscale convex/concave structure matching. *IEEE Trans. on Pat. Anal. and Mach. Intell.*, 15(4):337-352, 1993.

[16] R. Wiggins. Effects of three typographical variables on speed of reading. *Journal of Typographic Research*, 1:5–18, January 1967.

[17] A. P. Witkin. Scale space filtering. In *Proc. 8th IJCAI*, pages 1019–1022, Karlsruhe, West Germany, 1983.

[18] Jiann-Shing Wu and Jin-Jang Leou. A new approximation schemes for object shape representation. *Pattern Recognition*, 26(4):471–474, 1993.