



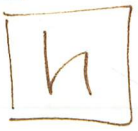
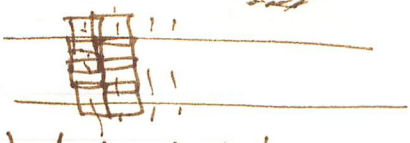
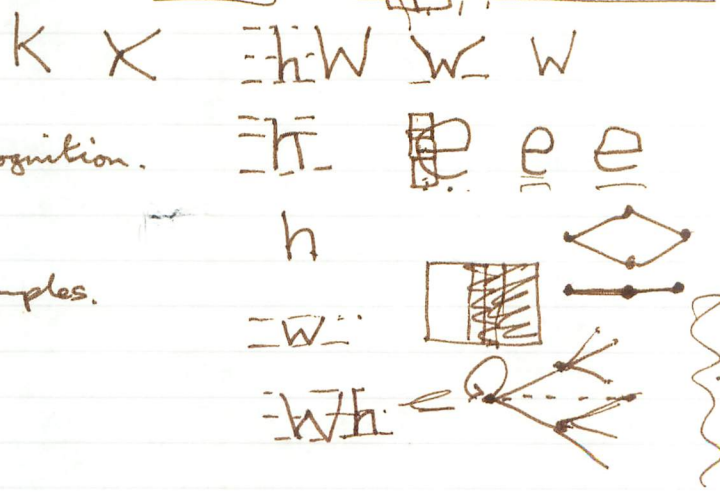

initial ideas ~~for~~ (tried in mark 1)

1. use grey-scale
2. don't segment before attempting recognition.
3. use Walsh functions.
4. learn discrimination net from examples.

problems with mark 1.

1. recognition no better than 50%.
2. could not understand why recognition failed.
   Discrimination net did not encompass enormous variation, and though possible to extend it, since the net was automatically produced by the program & since it was so complex, it wasn't clear ~~how~~ what was going wrong & how to remedy it (relevant to development).
3. not clear whether net could be extended to cope with variations across typefaces (e.g. width of characters) without becoming too large for practical purposes.
4. practical problems with grey scale:
   (a) lack of readily available scanners.
   (b) size of digitization — slows processing and requires more memory.
   (c) requires much use of multiplication — slow on microprocessors.
5. learning: how to distinguish variations which are significant for recognition (requires extra node(s) and arc(s)) from those which are not (requires generalization of description at node).

ideas for mark 2.

1. use binary images. (to save on memory and increase speed). therefore, Walsh functions abandoned.
2. generate net from manually designed ideal characters with pre-defined set of horizontal variations — helps understanding of recognition failures, since it is known just what variations are encompassed by net. (relevant to problems 2&3).
3. handle vertical variations by non-linear scaling of image - e.g. to compress tall ascenders or expand short ascenders.

problems with mark 2.

1. not clear how to constrain stretching & compressing variations of ideal characters to encompass normal, but only normal, characters. adding constraints would make network far too large (not mathematically predictable).

Aims

History
- comercial vs. academic
- hardware constraints of early comercial.

Scanning
- binary vs grey-scale

Variability
- across typefaces
  - width relative to height
  - length of ascenders and descenders
  - thickness of strokes — constrast
  - serifs
  - shape - e.g. height and slope of 'e' crossbar
- print quality
  - density - affects width of strokes
  - broken characters
  - joined characters
  - smudging
  - extra blobs of ink
- presentation
  - position
  - orientation
  - distortion - e.g. page of book not lying flat on scanner.
- scanning
  - resolution + thresholding → loss of thin strokes
  - spreading → loss of hole in 'e', for example.
  - ragged edges

Classification methods.

Distinguish descriptive capability from implementation aspects.

Linear discriminants

Boolean logic

Decision trees and nets – equivalent in recognition capability to Boolean logic, but allows sharing of ~~finite-state machines~~ logic. Also, since sequential, only those features which are required for discrimination need be extracted.

Finite-state machine – equivalent to decision tree, but with addition of repetition. Possible way to handle variability in e.g. width of character – but no way to put limits ~~on~~ on number of repetitions.

Syntactic

regular grammar – equivalent to finite-state machine

context-free grammar – allows nesting – not of much relevance to character recognition.

more powerful grammars – e.g. transformational – equivalent in ~~power~~ expressive power to general programming language, but more cumbersome than general rule-based systems.

General rule-based

Prolog is best known system of this kind – but its backtrack implementation is ~~very~~ inefficient.

1-8-82 Typereader contract.

13-12-82 letter from Peter – had sent schedule to SDA.

83 mark 1: used Walsh functions, developed on Z80
grey scale images, discrimination net built automatically from examples

84 mark 2: template matching, binary images, developed on Microframe.
discrimination net built from hand-designed templates.

1-85 mark2 demonstrated to Peter & Ben.
1-2-85 letter from Peter – had just demo of V2. enclosed part of 2nd payment.

27-3-85 went to Kaller Infotech

16-4-85 letter from Muirhead to Peter about mods to FAX.
9/10-85 mark 3: higher level descriptions of characters, use of epilogue.
general purpose high-level language which provides extra flexibility

3-12-85 letter from Peter – I had told him that TR was ready for porting & demonstration

29-1-86 press launch.