Chapter 1

Handwriting Recognition Engine

1.1 Radical Recognition Process

The recognition of a Radical is a search process. The data format of Radicals is described in section **??**. Essentially, a Radical is an ordered collection of strokes. In order to recognise a radical, the sequence of strokes from the database is compared to an input sequence. The measure of similarity is a feature vector that consists of the number of strokes and the similarities between them. Furthermore, permutations of the stroke sequence will be penalised.

1.2 Character Recognition Process

In the usual case, the strokes do not interfere in their sequence, but there are a view exceptions.

In order to scale to the normalised size, the length of an edge of the bounding box is compared to the length of the normalised character.

1.3 Error Handling

see section **??** in chapter **??** for possible sources of error

1.3.1 Error Recognition

why this section? to demonstrate own achievements of error recognition. the reader should know how it is done technically.

what goes into this section? the aspects of finding errors. finding errors is not a straightforward trivial task - whenever something does not match it is an error - doesn't work like that. instead, firstly, it needs to be made sure that it actually is an error. meaning - not a recognition error, but a user error. secondly, the type of error needs be identified. see section **??** (or handwritten page 58) for sources of error.

how will this section be written? technical - first describe how the error recognition integrates into the recognition process, then how errors are identified.

1.3.2 Error Processing

why this section? actually the 'handling' or 'processing' aspect could be described in the recognition section 1.3.1 as well. so this section is only for a better overview, for document structure, thematically they are the same section. thus they are put together under Error Handling 1.3.

what goes into this section?