Chapter 1

Handwriting Recognition Engine

1.0.1 Curve Handling

1.0.1.1 Detecting Curves in Strokes

Different curve matching techniques are described in section **??**. The technique used here refers to is a vector-based technique, that aims at the extraction of features to describe the curved stroke. In this prototype the direction feature plays a crucial role. It is generally follwing several approaches in on-line handwriting recognition literature. The direction feature as it is used in other applications has been discussed in section **??**). The approach for curve recognition used here defines direction vectors.

The general concept is to define a direction vector between a point within the point sequence and a successor point. However, since the measured points are usually very close together, the direction vector is defined between a point and another point that is further away, defined by a dynamic threshold, considering distance, total number of points and the total length of the stroke. If non of these vectors deviates from the general direction of the stroke, a dynamic threshold, considering the total stroke length, the stroke is considered straight. Nevertheless, for matching straight strokes, a feature vector for straight strokes is used. If the slope of the curve changes, i.e. the direction of the vectors changes, a curve has been detected. There is a threshold T_{min} for the minimal size of a *recognisable angle* between two vectors. Any angle smaller than that will be ignored completely. A second threshold T_{max} describes the minimal size for an angle γ that is taken into account. Any angle β_i between T_{min} and T_{max} will be stored. When the sum of the β_i exceeds T_{max} the position will be treated as a curve.

$\alpha < T_{min} \Rightarrow \alpha$ will be ignored

$T_{min} < \beta_i < T_{max} \Rightarrow \beta_i$ will be stored

$T_{max} < \gamma \Rightarrow \gamma$ will be interpreted as an angle in the stroke

$T_{max} < \sum_{i=k}^{l} \beta_i \Rightarrow \sum_{i=k}^{l} \beta_i$ will be interpreted as an angle in the stroke

1.0.1.2 Feature Extraction for Curved Strokes

The feature vector for curved strokes contains the following elements:

$$F := \begin{pmatrix} \text{Length} & & l \\ \text{Initial point} & & p_I \\ \text{Endpoint} & & p_E \\ \text{Velocity} & & v \\ \text{Corner points} & c_i & i := 1, \ldots, m \\ \text{Corner angles} & \gamma_i & i := 1, \ldots, m \\ \text{Direction sequence} & \vec{d_j} & j := 1, \ldots, q \end{pmatrix}$$

S 14, 16, 17 how is curver handling done? show requirements. what alternatives were there to consider? stroke matching with angles instead of point position. s. 24

1.0.2 Dynamic Time Warping

what's the similarity measure for points and strokes? show requirements. what alternatives were there to consider?

s. 51 how is dynamic time warping done here? pointer to papers or hwr - chapter, don't explain DTW here. show requirements why DTW? what alternatives were there to consider? none - it is the alternative. to all the other stuff I've been doing. however, what about 3D time warping?

1.1 Radical Recognition Process

1.2 Character Recognition Process

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?