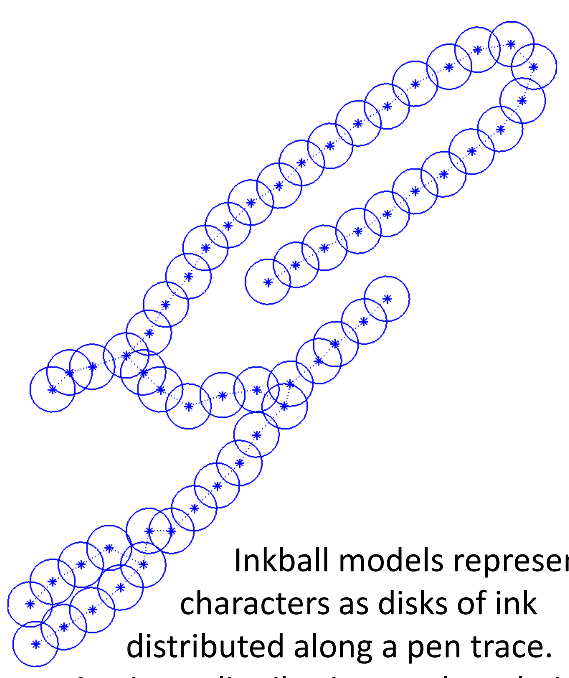
Inkball Models as Features for Handwriting Recognition

Nicholas Howe, Smith College
Andreas Fischer, University of Fribourg & HES-SO
Baptiste Wicht, University of Fribourg & HES-SO

ABSTRACT Inkball models provide a tool for matching and comparison of spatially structured markings such as handwritten characters and words. Hidden Markov models offer a framework for decoding a stream of text in terms of the most likely sequence of causal states. Prior work with HMM has relied on observation of features that are correlated with underlying characters, without modeling them directly. This paper proposes to use the results of inkball-based character matching as a feature set input directly to the HMM. Experiments indicate that this technique outperforms other tested methods at handwritten word recognition on a common benchmark when applied without normalization or text deslanting.



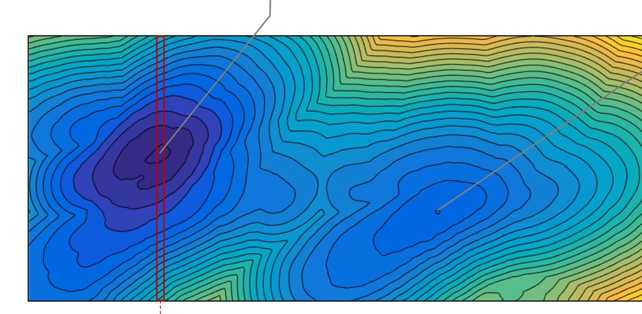
Inkball models represent characters as disks of ink distributed along a pen trace. Setting a distribution on the relative locations of neighboring disks yields a generative model. The likelihood of an observation under the model corresponds to the configuration with minimal collective displacement.

IDEA: Use the inkball model fit as input to a standard HMM. The model fit captures in one number a host of complicated structural details. While a single character model on its own is not discriminating enough to serve for character recognition, the collection of responses to a full set of character models carries meaningful patterns of information that the HMM can utilize.

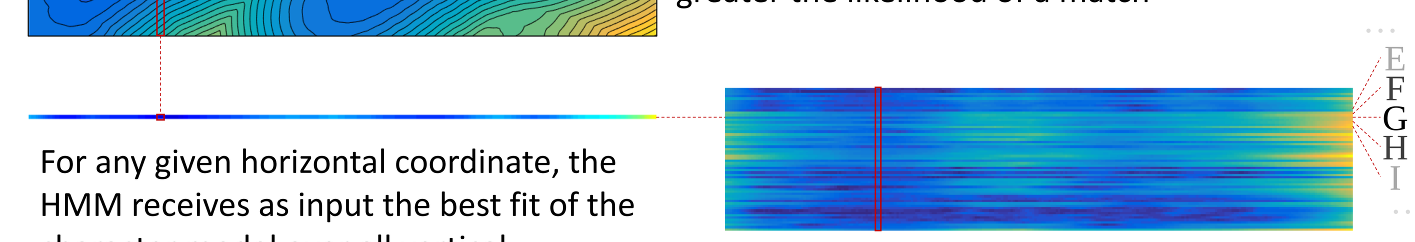
Hidden Markov Models (HMM) are a standard statistical framework for sequence recognition tasks. Originally introduced for speech, they have also been widely used for handwriting recognition based on different handwriting features. Well-known examples include Marti & Bunke's geometric features and the SIFT-like gradient features proposed by Rodriguez & Perronin and Terasawa & Tanaka, respectively. None of these directly model the character structure.

Baseline System

The fit score at any point corresponds to a particular configuration of the model in response to observations.

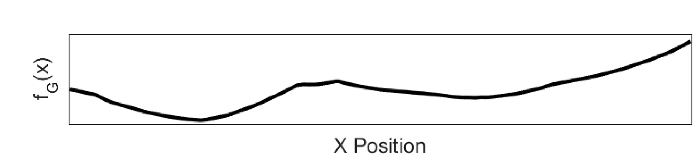


An inkball model placed anywhere in the image will deform to match the observed ink distribution. The lower the deformation, the greater the likelihood of a match



Each row corresponds to an individual character prototype

For any given horizontal coordinate, the HMM receives as input the best fit of the character model over all vertical positions.



HMM states correspond to horizontal positions. At each state, the HMM sees the best fit of every character model.

Experiments with Feature Design

Maybe feature values are sensitive to scale variations?

Experiment **S3** triples the number of features, with prototypes at 80%, 100%, and 125% of the original size

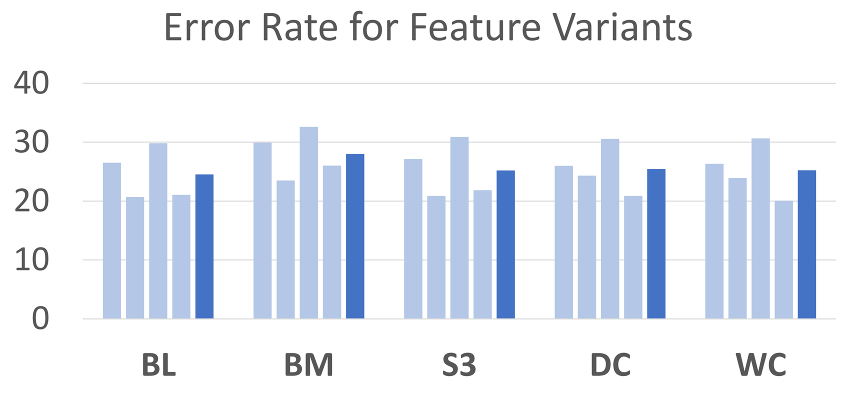
Maybe fitting a part-structured boundary model gives better features than an inkball model?

Experiment **BM** uses part-structured boundary models [Howe, HIP 2015]

Maybe the HMM needs more information to distinguish its position in each character?

Experiment **DC** adds a set of derivative features to the originals

Experiment **WC** adds a set of features comparing the local feature value to the minimum in a character-sized window



Do any of the variant feature selections improve on the baseline method (**BL**)?

No. Even though all except **BM** include the baseline features as a subset, none outperform the baseline. There seems to be a penalty for adding features.

Further experiments will stick to the baseline method.

Experiments with Prototype Selection

How are the prototypes selected for the baseline algorithm?

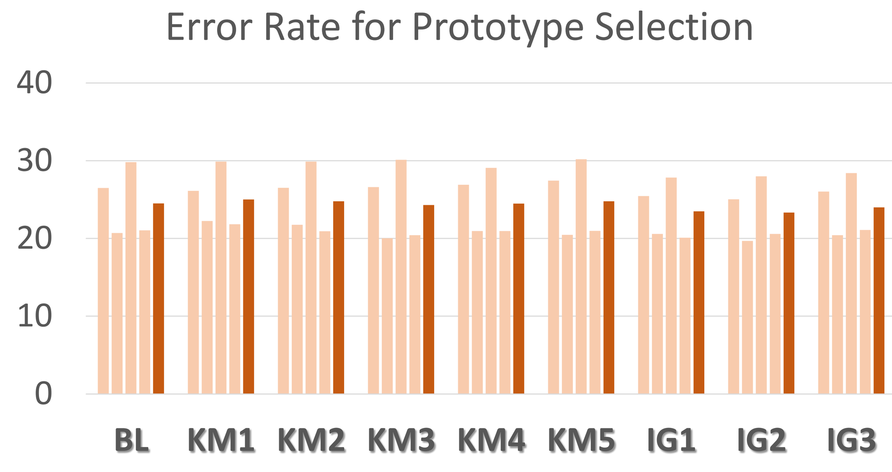
They are picked arbitrarily.

That doesn't sound very rigorous. Maybe choosing more representative character prototypes will improve the results?

Experiments **KM1-KM5** cluster character samples using *k*-medoids and take the cluster centers as prototypes.

Are there any other ways to select good prototypes?

Experiments **IG1-IG3** greedily select prototypes for information gain versus simple character recognition.



Do any of the prototype selection methods improve on the baseline?

Not significantly. The IG variants do the best, with **IG2** showing the lowest mean error. But the improvement is so small it may not be worth the extra effort required for the selection process.

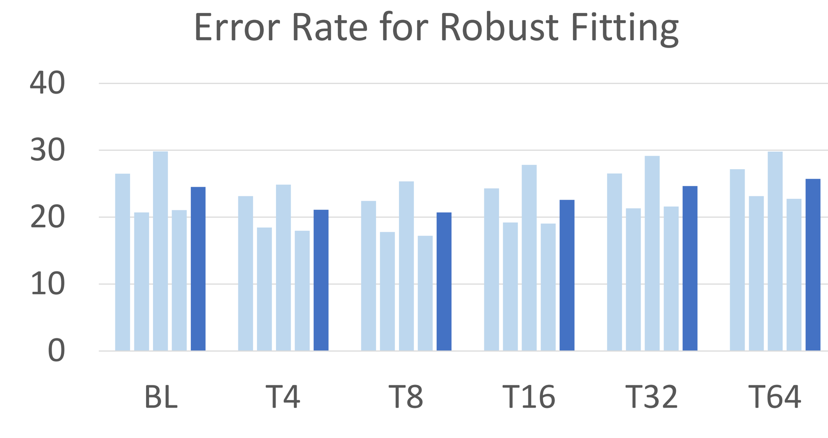
The IG2 method may be useful in some situations. But the simplicity of the baseline method is still attractive.

Experiments with Modified Fitting

Is there any other way to lower the error rate?

Maybe. Modifying the deformation score to use truncated Gaussians has shown promising results before.

Experiments **T4** through **T64** test varying truncation levels.



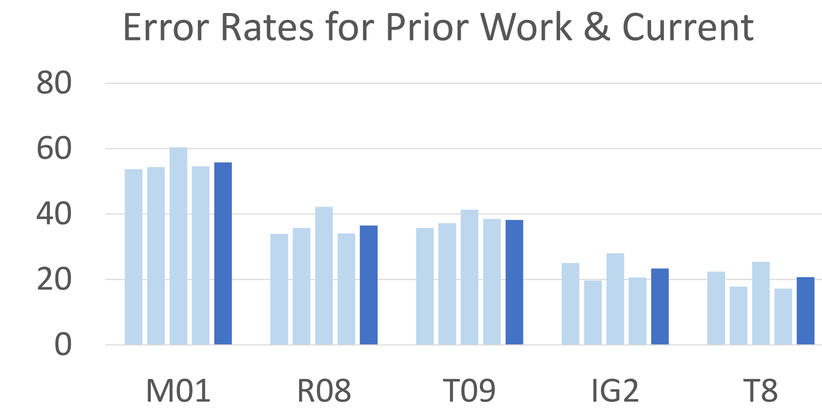
Does robust fitting lower the error?

Yes! The best results appear at **T8**.

Comparison with Prior Work

How does this method compare with prior work?

We compare to Marti & Bunke (**M01**), Rodriguez & Perronin (**R08**), and Terasawa & Tanaka (**T09**)



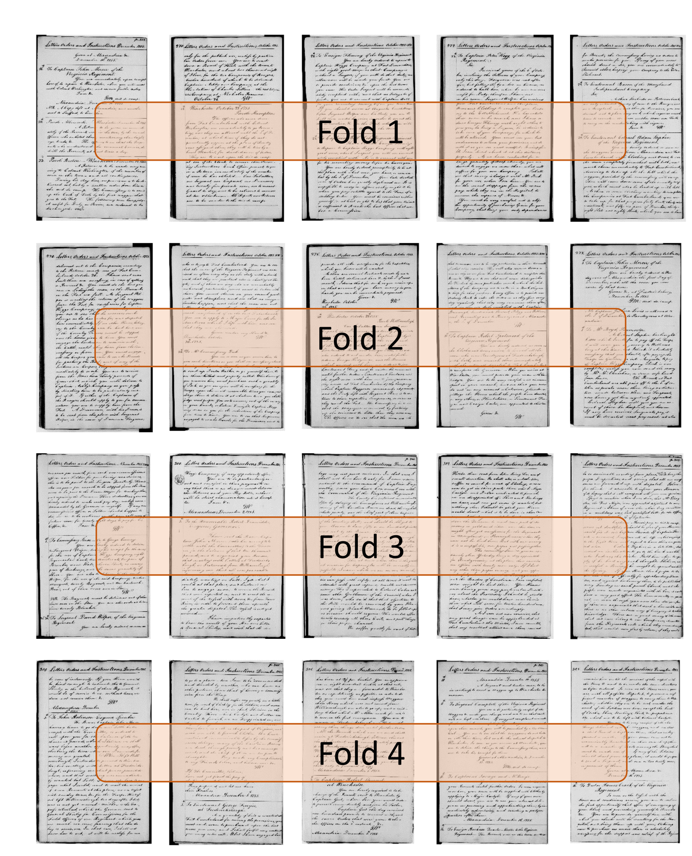
The results look suspiciously good.

All methods are run here without preprocessing & cleaning, which handicaps the prior work.

Experimental Conditions

How were the methods tested?

All use the 20-page George Washington dataset, divided into four folds.



Are the images preprocessed in any way?

Page images are binarized and split into lines.

Don't you need to deslant the lines?

No. With this method neither deslanting nor noise cleanup is required.

CONCLUSION Inkball character models fitted to observations serve as excellent input to a Hidden Markov Model for the character recognition task. These results must be seen as preliminary since all experiments are carried out for just one data set (GW20). Future work should look at performance on additional standard data sets, including multiwriter text with a diversity of handwriting styles. A more strict comparison with prior work would include image deslanting and cleaning for the algorithms that depend upon it.