Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

Downs and Acrosses: Textual Markup on a Stroke Level

Melissa Terras

School of Library, Archive, and Information Studies, University College London

Paul Robertson

MIT Computer Science and Artificial Intelligence Laboratory (CSAIL)

Abstract

Textual encoding is one of the main focuses of Humanities Computing. However, existing encoding schemes and initiatives focus on ‘text’ from the character level upwards, and are of little use to scholars, such as papyrologists and palaeographers, who study the constituent strokes of individual characters. This paper discusses the development of a markup system used to annotate a corpus of images of Roman texts, resulting in an XML representation of each character on a stroke by stroke basis. The XML data generated allows further interrogation of the palaeographic data, increasing the knowledge available regarding the palaeography of the documentation produced by the Roman Army. Additionally, the corpus was used to train an Artificial Intelligence system to effectively ‘read’ in stroke data of unknown text and output possible, reliable, interpretations of that text: the next step in aiding historians in the reading of ancient texts. The development and implementation of the markup scheme is introduced, the results of our initial encoding effort are presented, and it is demonstrated that textual markup on a stroke level can extend the remit of marked up digital texts in the humanities.

1 Introduction

Humanities Computing scholars are more than familiar with the techniques and technologies used to mark up textual data: the markup of electronic text, and the development of

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

associated tools and technologies remains the main thrust of the discipline.[1] However, encoding schemes and initiatives such as the Textual Encoding Initiative focus on 'text' from the character level upwards in order to 'represent all kinds of literary and linguistic texts for online research and teaching, using an encoding scheme that is maximally expressive and minimally obsolescent'.[2] Whilst this scheme does provide excellent standards for textual encoding, and thereby the means to digitally manipulate text to enable further linguistic, stylistic, and historical study, it is of little use for scholars such as palaeographers and papyrologists who study literary and linguistic texts on a more basic level than that of the character: their focus being on the individual character *strokes* themselves.

Building palaeographical databases which capture data regarding texts at the stroke level is also imperative in order to train Artificial Intelligence systems aimed at supporting the reading of damaged and deteriorated texts (Terras, 2002). Whilst there has been some interest in the development of encoding schemes to capture such textual data, there has been little work done to date in developing systematic techniques that can be used to annotate texts on this level for use in the humanities.[3] However, annotating corpuses of images to

---

Correspondence:

Melissa Terras, School of Library, Archive, and Information Studies, University College London, Gower Street, London WC1E 6BT.

E-mail:

m.terras@ucl.ac.uk

Paul Robertson, MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT Building NE43, 200 Technology Square, Cambridge, MA 02139.

E-mail:

paulr@ai.mit.edu

[1] Willard McCarty's preliminary definition, and history, of Humanities Computing demonstrates that the established and successful projects and initiatives in Humanities Computing have centred, to date, around textual data (McCarty, 2003).

[2] http://www.tei-c.org/

[3] There have been many handwriting recognition systems constructed that identify and use elementary strokes and graphical primitives, such as loops, lines, and crosses, to identify text.

develop data on which to train cognitive visual systems is a frequent practice in the field of Artificial Intelligence (Charniak, 1993, Lawler and Dry, 1998, Robertson, 1999). By adopting and adapting tools and techniques developed for the markup of aerial images (Robertson, 2001), we can show that it is possible to markup *images* of text that result in an XML representation of each individual character on a stroke by stroke basis.

The development of a markup system to annotate texts in this manner was imperative to aid in the construction of an intelligent system to effectively 'read' Roman stylus tablets. The ink and stylus tablets from Vindolanda[4] are an unparalleled resource of the ancient world, giving an immediate, personal, and detailed account of the Roman occupation of Britain from around AD 92 (Bowman and Thomas, 1994). Although the handwriting on the ink texts can be made visible through the use of infra-red photography, the physical condition of the stylus tablets renders them illegible to the human eye. Text in the stylus tablets was incised into wax held within a recess in the tablet, and in nearly all surviving stylus tablets the wax has perished, leaving a deteriorated recessed surface showing the scratches made by the stylus as it penetrated the wax. This is devilishly difficult to decipher, with a skilled reader taking several weeks, if not months, to read a single tablet, and readings are complicated further by the fact that the tablets could have been reused many times by melting the wax and inscribing new text over the old.

(Figure 1 about here)

Novel imaging techniques were developed at the Department of Engineering Science, University of Oxford, to analyse these texts (Bowman *et al.* 1997, Brady *et al.*, 2004). However, it was necessary to develop a computer program to aid the historians in interrogating the images of the Vindolanda texts, to speed up the process of reading such

---

Defining letters as combinations of elementary strokes has also been used for online processing of complex characters in Chinese or Korean alphabets (Liu *et al.*, 2001). However, this paper describes the first attempt to generate a comprehensive XML based encoding scheme that captures the type of knowledge humanities experts use when attempting to read texts, and so builds on the techniques that they have already developed to decipher unknown characters.

[4] A Roman fort on the Stanegate near Hadrian's Wall and modern day Chesterholm. See Bidwell (1985) for further information regarding Vindolanda, Birley (1999) for an account of the discovery of the tablets, and Bowman (2003) for an introduction to the content of the Vindolanda texts.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

damaged and deteriorated documents. In order to utilize Artificial Intelligence techniques which could aid further in their reading, it was necessary to collect as much information as possible regarding the individual strokes which constitute these texts, in order to train a system to identify possible and probable configurations of strokes.

An encoding scheme was developed which encapsulates all the information used by experts as they attempt to read these documents. A suitable program was identified, adopted, and adapted to enable images of the document to be encoded and tagged, which generated XML files containing representations of the texts on a stroke by stroke basis. A series of images of Vindolanda texts were tagged, and the resulting corpus of annotated images and their XML files were used to train an Artificial Intelligence system to effectively 'read' in stroke data of unknown text and output possible, reliable, interpretations of that text: the future for aiding historians in generating readings of the Vindolanda texts. Additionally, the corpus of tagged images is the largest available dataset regarding Old Roman Cursive (the form of handwriting found on the Vindolanda texts), and provides a means for further study of the palaeography of the Roman Army's documentation at this period.

2 Knowledge Elicitation

The encoding scheme was developed after a process of Knowledge Elicitation, where the behaviour of experts is analysed to understand the knowledge and procedures they use whilst carrying out expert tasks (Diaper, 1989; McGraw and Harbisson-Briggs, 1989). All available information regarding Old Roman Cursive (ORC) was gathered, to identify the type of information noted when discussing the strokes used in contemporaneous texts. There is a paucity of documents from this period of the Roman Empire, and so the development of the character forms in ORC are the subject of much academic debate (see Bowman and Thomas 1983, 1994, and Bowman, 2003).

(Figure 2 about here)

To gain a deeper understanding of these letter forms, interviews were undertaken with three expert papyrologists working in the field of Old Roman Cursive palaeography. The experts were asked to discuss particular topics such as individual letter forms, standard forms and deviating forms of characters, the physical relationship of characters to each other (ligatures, serifs, etc.), specific features such as descenders, ascenders, and junctions, characters which are often confused with another, and the identification of similar hands in

different documents. These interviews were transcribed, and Automated Knowledge Elicitation techniques were used to help resolve this information into a structure that served as the basis of a markup scheme to annotate images of the texts on a stroke level.

(Figure 3 about here)

3 The Markup Scheme

The markup scheme consists of over thirty different elements which can be used when annotating individual character strokes. The markup scheme was developed by restructuring the information shown above to a more linear format, and can be summarized as follows:

Each region in the image of the text is expected to correspond to an area occupied by either a:

- Character Box (the area surrounding a collection of strokes which make up an individual character)
- Space Character (indicating the space between words)
- Paragraph Character (indicating areas of indentation in the text)
- Interpunct (indicating the mark sometimes used to differentiate words in this period of Old Roman Cursive)

Each character is comprised of strokes:

- Traced and numbered individually (this numbering being used to identify separate strokes, and not necessarily reflecting the order in which the strokes were made, as this is very difficult to determine in such abraded texts. Identifying strokes in this manner involves some interpretation by the annotator regarding what is actually present, and what can be identified as an individual stroke.)

Each stroke has end points, which are either:

- Blunt (i.e. with no complex formatting to the end point)
- hooked (with direction specified, i.e. up left, up right etc. The direction specified was relative to the base line of the text in the document.)

- ligatured (with direction specified)
- or has a serif (with direction specified)

Each character may also have stroke meetings, where strokes combine. These are either:

- end to end (where the ends of strokes combine, such as in our letter ‘N’)
    - exact meet (where the strokes meet exactly)
    - close meet (where there is a small gap between the two strokes)
    - crossing (where there is a small overlap between the two strokes)
- or middle to end (where one stroke meets the middle of another, such as in the junction in our capital letter ‘T’)
    - exact
    - close
    - crossing
- or crossing (middle to middle, such as with the letter ‘X’)

Each stroke was assigned additional tags, having:

- A direction (‘Direction’), giving the stroke orientation and type, which included:
    - Straight Strokes (‘DirectionStraight’)
    - Simple Curved Strokes (‘DirectionCurved’)
    - Complex curved Strokes (‘DirectionCurvedWave’)
    - Loops (‘DirectionLoop’)
        - Each of these tags included further orientation tags to indicate left, right, etc. (orientation being dictated by taking the centre point of the Character Box on the base line of writing, and deriving from that up, down, left, right, etc. Describing orientation in this manner replicates the language the papyrologists use when discussing texts, and also includes some level of abstraction in the modelling of

the characters, which is useful for comparing descriptions of the characters later. If the direction had been measured absolutely, for example using co-ordinates, it would be very difficult to compare individual instances of the characters automatically, as the information captured would have been too specific.) For example, DirectionStraightDownLeft represents a stroke which is straight, and veers to the left, down from the centre of the character box, away from the centre.

- A Length ('Length'), being either comparatively short, average, or long. (Again, this replicated the language used by the papyrologists when discussing stroke length.)
- A Width ('Width') being either comparatively thin, average, or wide. (Replicating the variables used in the papyrologists discussion of stroke width.)
- A Place ('Place') on the writing line, being either within the average, descending below the nominal writing line, or ascending above the average height of a character.

Each stroke meeting, or junction, had:

- an Angle ('Angle'), with note taken of the orientation of the meeting and whether the angle was acute, right, obtuse, or parallel.

4 Building the Data Set

4.1 Annotation Program

To be able to annotate the images using this encoding scheme, an annotation program was used. This was a slightly revised version of the GRAVA Annotator Program, a Motif[5] program developed by Paul Robertson (Computer Science and Artificial Intelligence Laboratory, MIT) originally developed to manually segment and label a corpus of aerial satellite images[6] (see Robertson, 2001, Appendix C). The results of the annotation are

[5] Motif is the industry standard graphical user interface environment for standardizing application presentation on a wide range of platforms. Developed by the Open Software Foundation (OSF, see www.opengroup.org) it is the leading user interface for the Unix system. In this case, we utilized Exceed, an X-Server for the PC, to run our Motif application on the Windows platform.

[6] The corpus produced in this case was subsequently used to train a system to segment, label, and

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

written to a file in Extensible Markup Language[7] (XML). Robertson's software was adapted to incorporate the encoding scheme, detailed above, that was relevant to the Vindolanda texts, to allow these to be labelled in the same manner.

(Figures 4 &5 about here)

4.2 Corpus selection

The only documents contemporaneous with the Vindolanda stylus tablets are the Vindolanda ink tablets. Although these are very different mediums, psychological studies have shown that writing by the same scribe on different mediums is overwhelmingly similar, as writing is a higher level cognitive process unaffected by change in writing implements or surfaces (Mathyer, 1969). There is no reason to think that the letter forms in the Vindolanda stylus tablets will differ substantially from those contained in the ink tablets, and for this reason, annotating images of the text contained on the ink tablets would provide palaeographic data regarding the letter forms expected within the stylus tablets.

Seven ink tablets were identified, with the help of one of the experts, that would provide good, clear images of text to annotate, and have enough textual content to provide a suitable set of data. These texts were mostly personal correspondence. Additionally, two stylus tablets were chosen—simply because they were the only stylus texts that had been successfully read by the experts. One of the experts was provided with large scale images of these nine texts and asked to trace the individual strokes of each letter form as they appeared on the tablets. These images were then married up with the published texts regarding the documents to confirm the readings of the letters (letters such as R and A are easily confused). This was combined with the information gleaned from the Knowledge Elicitation exercises regarding character forms and formation to provide the information with which to annotate the images.

Although this type of annotation does involve some form of interpretation by the knowledge engineer, it was done in as systematic and methodological a fashion as possible. Care was taken to annotate the images without the addition of personal bias by retaining a distanced stance, annotating what was present rather than what *should* be expected. The

---

parse aerial images so as to produce an image description similar to that produced by a human expert.

[7] http://www.w3.org/XML/

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

annotated texts were cross referenced with the published texts after annotation, to ensure that the texts contained the same characters in the same order, as a means of quality control.

4.3 Encoding

Characters were annotated firstly by drawing around the outline of a character using a computer mouse, and assigning it a character label. Individual strokes were then traced, and numbered by selecting the option from a drop down menu. Stroke ends were then identified and labelled. Finally, stroke junctions were noted and assigned labels. An example of this is shown below, using the letter S from the start of ink tablet 311 as an example.

(**Figure 6 about here)**

These annotations are preserved in an XML file which textually describes the annotated image. The placing of the strokes is preserved by noting the co-ordinates of each feature. For example, this is the XML which describes the bounding character box of the letter ‘S’ in the above example:

```
<GTRegion author="Melissa Terras" regionType="ADDCHARACTER"
regionUID="RGN0" regiondate="04/04/02 18:07:16" coordinates="112,
142, 106, 223, 87, 317, 52, 377, 56, 420, 126, 423, 180, 349, 203,
244, 199, 108, 269, 71, 323, 28, 298, 7, 190, 23, 118, 69, 112,
142"></GTRegion>
```

The region type tag ‘ADDCHARACTER’ defines this region as a character box which surrounds a collection of strokes, the region ID ‘RGNO’ gives the annotated region a unique identification number, and the co-ordinates preserve the shape of the character box. Each individual region that is annotated (characters, strokes, stroke endings, and stroke meetings) has its own similar line in the XML file, with each having a unique identifying Region ID, and a region type which specifies what type of annotation has been made. This file is structured hierarchically; all strokes, stroke endings, and stroke meetings ‘belong’ to an individual character.

4.3.1 Additional Encoding

The additional directional information described above in section 3 was included into the annotation by assigning a textual code for each region in the ‘comments’ field of the GRAVA annotator. The letter S from 311, above, was deemed to be of large height and

width, and so the additional comments added to the XML file regarding the character box were:

```
comments="*S, SizeHeightLarge, SizeWidthLarge"
```

'*S' identifies the character as the letter S, SizeHeightLarge indicates that the height is large, and SizeWidthLarge indicates that the width is large. This gives the final resulting XML output for this region as

```
<GTRegion author="Melissa Terras" regionType="ADDCHARACTER"
regionUID="RGN0" regiondate="04/04/02 18:07:16" coordinates="112,
142, 106, 223, 87, 317, 52, 377, 56, 420, 126, 423, 180, 349, 203,
244, 199, 108, 269, 71, 323, 28, 298, 7, 190, 23, 118, 69, 112, 142"
comments="*S, SizeHeightLarge, SizeWidthLarge"></GTRegion>
```

An example of a full sample XML file is given below. This file describes the letter S, as shown in Fig. 6, firstly defining a character box (ADDCHARACTER), then the two individual strokes (StrokeO1, StrokeO2). Stroke ends are then identified (StrokeEndHookUpLeft, StrokeEndBlunt), and the junction is then identified (StrokeMeetingJunctionCrossEnds). Comments added to the regions give additional directional information regarding the strokes within the file, as described above in section 3. The hierarchical nature of the file ensures that all strokes, stroke endings, and junctions are associated with the ADDCHARACTER region they belong to: in this case, the letter S. All the files contain a reference to the original image they were generated from. (The GT tag represents 'GRAVA Tools'.)

```
<GTAnnotations imageName="C:\grava\311.tif" author="Melissa Terras"
creationDate="09/10/03 15:23:58" modificationDate="09/10/02 15:27:30">
<GTRegion author="Melissa Terras" regionType="ADDCHARACTER"
regionUID="RGN0"

regiondate="09/10/03 15:24:36" coordinates="338, 22, 298, 4, 186, 30, 106,
62, 94,

196, 36, 356, 46, 408, 140, 420, 184, 288, 198, 106, 282, 68, 338, 22"
comments="*S,

SizeHeightLarge, SizeWidthLarge">
```

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

```
        <GTRegion regionType="StrokeO1" regionUID="RGN1" coordinates="170,
        76, 162, 184, 152, 276, 124, 350, 102, 382, 74, 372"
        comments="DirectionStraightDownLeft, LengthLong, WidthAverage,
        PlaceDescendingLeft"></GTRegion>

        <GTRegion regionType="StrokeO2" regionUID="RGN2" coordinates="146,
        94, 286, 18" comments="DirectionStraightUpRight, LengthAverage,
        WidthAverage, PlaceAscendingRight"></GTRegion>

        <GTRegion author regionType="StrokeEndHookUpLeft " regionUID="RGN3"
        coordinates="62, 354, 94, 354, 94, 386, 62, 386, 62,
        354"></GTRegion>

        <GTRegion regionType="StrokeEndBlunt" regionUID="RGN4"
        coordinates="162, 62, 178, 62, 178, 88, 162, 88, 162,
        62"></GTRegion>

        <GTRegion regionType="StrokeEndBlunt" regionUID="RGN5"
        coordinates="134, 80, 154, 80, 154, 104, 134, 104, 134,
        80"></GTRegion>

        <GTRegion regionType="StrokeEndBlunt" regionUID="RGN6"
        coordinates="270, 14, 294, 14, 294, 36, 270, 36, 270,
        14"></GTRegion>

        <GTRegion regionType="StrokeMeetingJunctionCrossEnds"
        regionUID="RGN7" coordinates="154, 70, 184, 70, 184, 98, 154, 98,
        154, 70"></GTRegion>

</GTRegion>

</GTAnnotations>
```

4.4 Results

Each image of the Vindolanda texts was annotated in the way described above. Care was taken to be as methodological as possible whilst undertaking this task. In total, 1506 individual characters from the ink tablets were annotated, and 180 characters from the stylus tablets. The nine completed sets of annotated images represented approximately 300 hours of

work, with an average of around six or seven characters being completed in an hour. The images were annotated and checked over a period of three months.

5 Use of the Corpus

5.1 Palaeographic use

The corpus was firstly used to export an example of every single letter it contained, which provides a useful reference to Old Roman Cursive for palaeographers.

(Figures 7 & 8 about here)

It has not previously been possible to undertake a comprehensive comparison of the differences between Old Roman Cursive as found on the Vindolanda stylus tablets and the ink texts, due to the nature of the documents and the difficulty in making out the letter forms on the stylus tablets. Although this research was based on the assumption that the writing would take similar forms (see 4.2) the fact that this corpus made explicit features of every character meant it was possible to make a direct comparison of the letter forms found on the stylus tablets with those found on the ink texts (Terras, 2002, 3.10). Interesting observations were noted, for example: the stylus 'I' shows less use of ligatures and serifs than that found in the ink texts, and the ink 'I' tends to slope from bottom left to upper right, whilst the stylus 'I' inclines slightly from upper left to bottom right. Using the corpus in this manner furthers the understanding of Old Roman Cursive, and so increases the chances of being able to read further documents successfully.

Additionally, making every aspect of each character explicit means that the data can be used to propagate probabilities that would help in the identification of unknown characters. For example, see the table, below, which displays a few attributes associated with every single letter F in the corpus:

This table indicates that, usually, the letter 'F' in ORC is of a large height, i.e. is significantly taller than the average character. Additionally, it is usually comprised of three strokes and two junctions, with the first stroke being of longer than average length which descends below the writing line to the left. If a papyrologist comes across a large letter with a long descending first stroke to the left, there is a good chance that this may be the letter F. This is hardly news to a skilled papyrologist! However, the corpus makes such information *explicit*, instead of being held implicitly within the papyrologists' expertise. The interesting

use of the data will lie in its future interrogation: future research will look at probabilities *across* the corpus: for example, if there is an unknown character with a long descending first stroke to the left, what is the numerical probability that this is an F, or an S, or a Q, or an R, etc.? Making such information explicit is the first way to begin to develop systems that can help interrogate the information presented within the texts automatically.

5.2 Artificial Intelligence and the corpus

The primary aim of building such a corpus was to provide data with which to drive an Artificial Intelligence system, which could input unknown text and output probable interpretations. To match unknown to known characters, it is necessary to have models of those known characters to compare the unknown data to. Annotating the character forms in the way described above meant that vector co-ordinates for all the characters had been captured, and so character models for each type of character could be generated.

A character model is defined as a probability field that indicates the likely placing of one or more strokes of a two-dimensional character, producing a general representation of a character type. Unknown characters can then be compared to a series of these models, and the probability that they are an instance of each one calculated, the highest probability indicating a match.

On a conceptual level, the (stroke-based) character model is constructed by taking an image of an individual character, finding its bounding box (identifying the rightmost x co-ordinate, and the leftmost x co-ordinate, and the highest and lowest y co-ordinates), and transforming this into a standardized (21 by 21 pixel) grid. The stroke data is blurred slightly to produce a generalized model which allows specific instances to be compared to it more successfully.[8] Each standardized representation is accumulated onto a generalized matrix for each character type: resulting in a generalized representation of each type of character. These are subsequently used as the models to which unknown characters are compared. An example of how these steps combine to generate a character model is given below, where a small corpus which contains three ‘S’ characters is used to generate a character model of an S.

(Figure 9 about here)

[8] i.e. the stroke data is convolved with a Gaussian Blur operator to reduce over-fitting.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

Character models of every character contained within the dataset were generated, as shown below:

(Figure 10 about here)

These models were used to train an Artificial Intelligence system to effectively 'read' in stroke data of unknown text and output possible, reliable, interpretations of that text. By combining these stroke models with other linguistic models and results from further knowledge elicitation exercises, and advanced Artificial Intelligence techniques, a system was constructed that was able to replicate the process the historians went through to output possible interpretations of the Vindolanda texts (Terras, 2002). More development needs to be undertaken before this system becomes a stand-alone application, but its success is grounded on the fact that it has such a rich dataset to draw information from. Only by capturing palaeographic data in the level of detail displayed in the corpus can it be used to generate useful models and statistics for use in Artificial Intelligence applications.

6 Conclusion

Annotating images of text on a stroke by stroke basis provides a rich palaeographic dataset which can be manipulated to generate novel representations and further understanding of letter forms and systems. By adapting techniques from Artificial Intelligence, it has been possible to encode a series of Old Roman Cursive texts, generating new tools to analyse and interrogate further documents along the way. Borrowing techniques from other fields in Computing and Engineering Science means it is possible to extend the remit of 'textual markup' in the humanities.

Whilst the developed encoding scheme described here primarily regards Old Roman Cursive text, there is no reason why it could not be expanded to cover other types of hands (indeed, it is suspected that the markup scheme is fairly comprehensive for most Latinate languages). The markup scheme is not, at present, compliant with any standard apart from the XML standard itself: initiatives such as the TEI encoding scheme stop at the character level. One of the questions raised by this research is where does 'textual' encoding stop? At what level is text no longer text, but graphical information? Is there a need to include palaeographical markup within established initiatives, or is such markup only relevant to the few experts who would have use from the system and data described here? Nevertheless, the tools described and developed which allow XML representations to be derived from image

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414
data could have many uses with the sphere of humanities' digital resources.

It has been demonstrated here that it is possible to develop and implement tools which allow the capturing of complex palaeographic data through marking up images of text to generate XML representations of letter forms. Continuing the development of such schemes and tools will increase the usefulness of computing tools for the palaeographic community, and may well provide the means to train systems to read previously illegible ancient texts.

Acknowledgements

Thanks go to the supervisors of this research: Professor Alan Bowman (Centre for the Study of Ancient Documents, University of Oxford), and Professor Mike Brady (Department of Engineering Science, University of Oxford). Thanks also go to the anonymous papyrologist guinea pigs, who allowed access to their expertise. Dr Xiao-Bo Pan, formerly of the Robots Group, Department of Engineering Science, University of Oxford, aided in the preparation of images exported from the corpus.

References

Bidwell, P. T. (1985). *The Roman Fort of Vindolanda at Chesterholm, Northumberland*. London, Historic Building and Monuments Commission for England.

Birley, R. (1999). *Writing Materials*. Greenhead, Roman Army Museum Publications.

Bowman, A. K. (2003). Life *and Letters on the Roman Frontier*. London, British Museum Press.

Bowman, A. K., Brady, J. M., *et al*. (1997). Imaging incised documents. *Literary and Linguistic Computing*, **12**(3): 169–76.

Bowman, A. K. and Thomas, J. D. (1983). *Vindolanda: The Latin Writing Tablets*. London, Society for Promotion of Roman Studies.

Bowman, A. K., and Thomas, J. D. (1994). *The Vindolanda Writing-Tablets (Tabulae Vindolandenses II)*. London, British Museum Press.

Brady, M., Pan, X., Terras, M., and Schenk, V. (forthcoming (2004)). *Shadow Stereo, Image Filtering and Constraint Propagation*. Images and Artefacts of the Ancient World, London, British Academy.

Charniak, E. (1993). *Statistical Language Learning*. Cambridge, MA: MIT Press.

Diaper, D. (1989). *Knowledge Elicitation: Principles, Techniques and Applications*. Chichester:

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

Ellis Horwood.

Lawler, J. M. and Dry, H. A. (eds). (1998). *Using Computers in Linguistics, A Practical Guide*. London: Routledge.

Liu, C-L., Kim, I-J., and Kim, J. H. (2001). Model-based stroke extraction and matching for handwritten Chinese character recognition. *Pattern Recognition*, **34**: 2339–52.

McCarty, W. (2003). Humanities computing. Preliminary draft entry for *The Encyclopedia of Library and Information Science*. New York: Dekker.

http://www.kcl.ac.uk/humanities/cch/wlm/essays/encyc/

McGraw, K. L. and Harbisson-Briggs, K. (1989). *Knowledge Acquisition: Principles and Guidelines*. London: Prentice-Hall International Editions.

Mathyer, J. (1969). Influence on writing instruments on handwriting and signatures. *Journal of Criminal Law, Criminology, and Police Science*, **60**: 102–12.

Robertson, P. (1999). *A Corpus Based Approach to the Interpretation of Aerial Images*. IEE IPA99, Manchester.

Robertson, P. (2001). *A Self Adaptive Architecture for Image Understanding*. Ph.D. Thesis Department of Engineering Science, Oxford, University of Oxford.

Terras, M. (2002). *Image to Interpretation: Towards an Intelligent System to Aid Historians in the Reading of the Vindolanda Texts*. Ph.D. Thesis, Department of Engineering Science, University of Oxford.

Table 1 Data extracted from the corpus relating to every single example of the letter F

Tablet	Letter	Overall Height	Overall Width	No of Strokes	No of Junctions	
	Stroke 1 Length	Stroke 1 Place				
225f	F	Large	Large	3	2	Long
	Descending, Left					
248f	F	Large	Large	3	2	Long
	Descending, Left					
248f	F	Large	Large	3	2	Long
	Descending, Left					
291f	F	Large	Large	3	2	Long

	Descending, Left					
311f	F Descending, Left	Large	Average	3	2	Long
311f	F Descending, Straight	Large	Average	3	2	Long
225f	F Within Line	Average	Average	3	2	Average
225b	F Descending, Left	Large	Average	2	1	Long
248f	F Descending, Left	Large	Large	2	1	Long
311f	F Descending, Left	Large	Average	2	1	Long
311f	F? Within Line	Average	Average	2	1	Average
291f	F? Within Line	Small	Average	1	0	Average

(Figure captions)

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

Fig. 1 Stylus tablet 836, one of the most complete stylus tablets unearthed at Vindolanda. This text is a letter from Albanus to Bellus, containing a receipt and further demand for payment of transp ort costs. The incisions on the surface can be seen to be complex, whilst the woodgrain, surface discoloration, warping, and cracking of the physical object demonstrate the difficulty papyrologists have in reading such texts.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

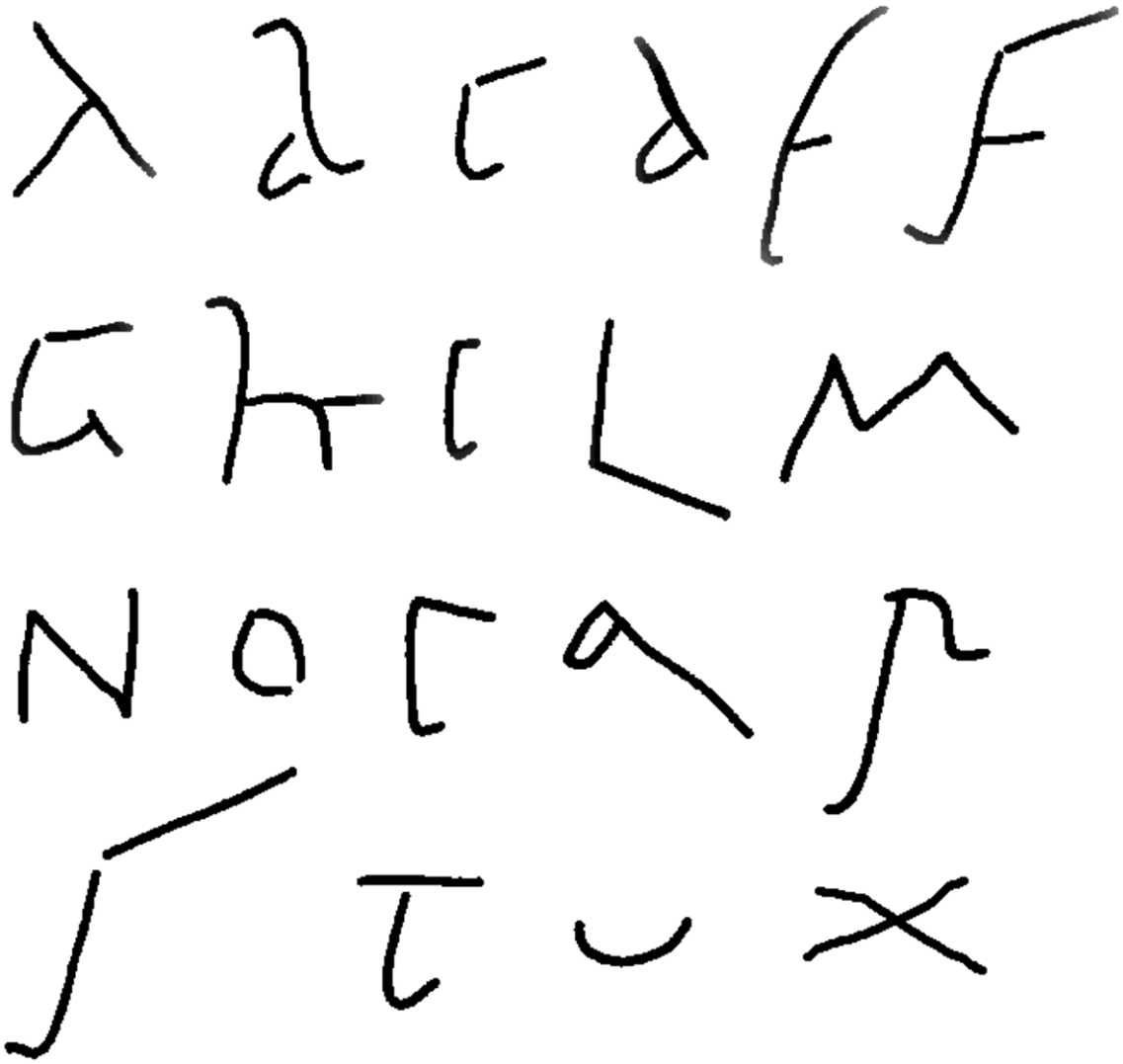

Fig. 2 Characteristic letter forms in ORC, based on Bowman and Thomas (1983, p. 54). The alphabet consists of 20 characters (not having the letters j, k, v, w, y, and z present in modern day English script).

Fig. 3 An ontology of information used by experts when discussing letter forms, presented in the form of a semantic network. The character information can be grouped into five main subheadings: strokes, stroke combinations, context, basic forms and similar characters, with each subheading being discussed in various ways.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

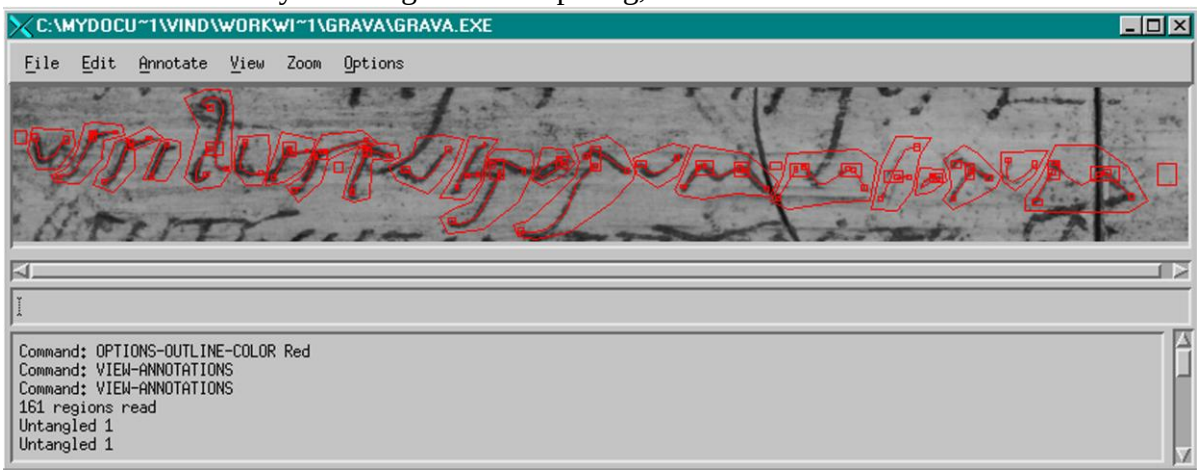

Fig. 4 The GRAVA annotation program. The facility allows regions to be hand traced and assigned a label. The application screen is divided into a menu bar, image display pane, command input area, and message area. Most interaction with the annotator is performed using the mouse.

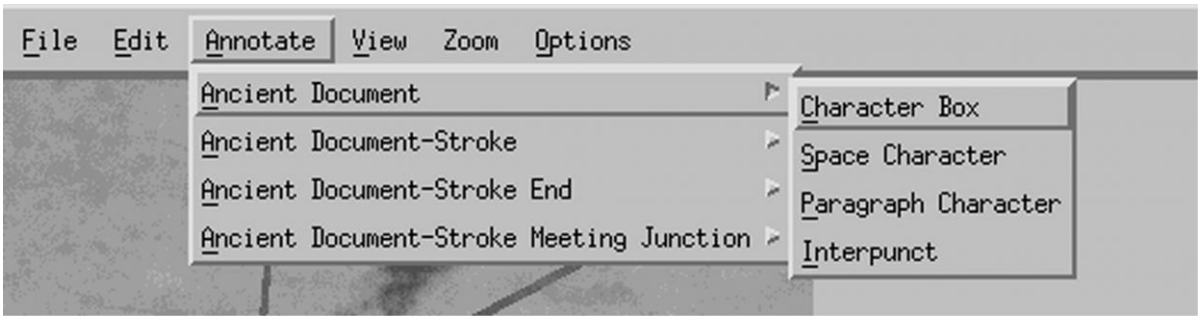

Fig. 5 Drop down menus in the GRAVA Annotator, incorporating the encoding scheme determined from the knowledge elicitation exercises.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

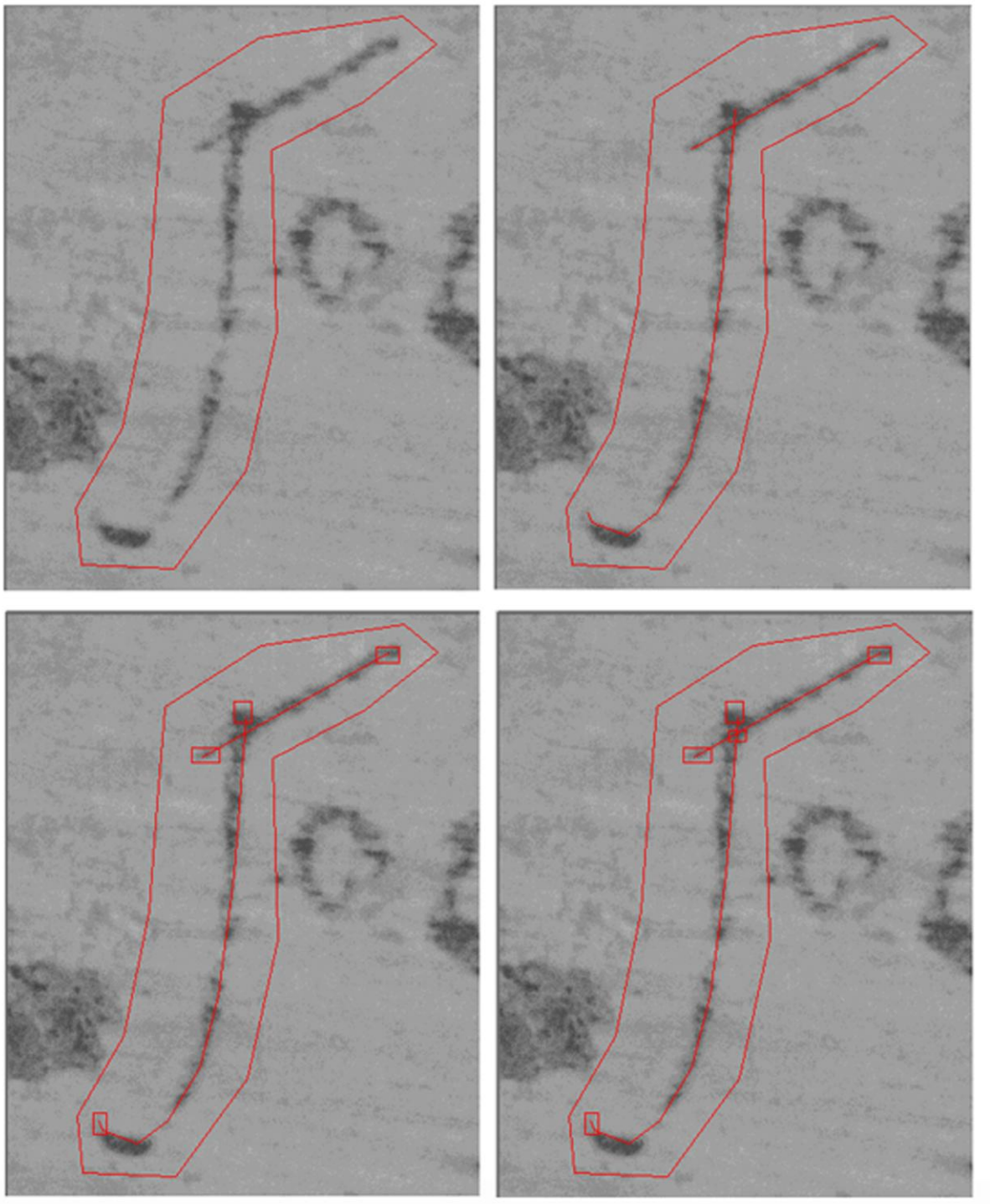

Fig. 6 Steps taken in annotating a letter. The outline is traced at first, followed by individual strokes, stroke endings, and stroke meeting junctions. All are assigned labels from the drop down menus in the GRAVA program.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

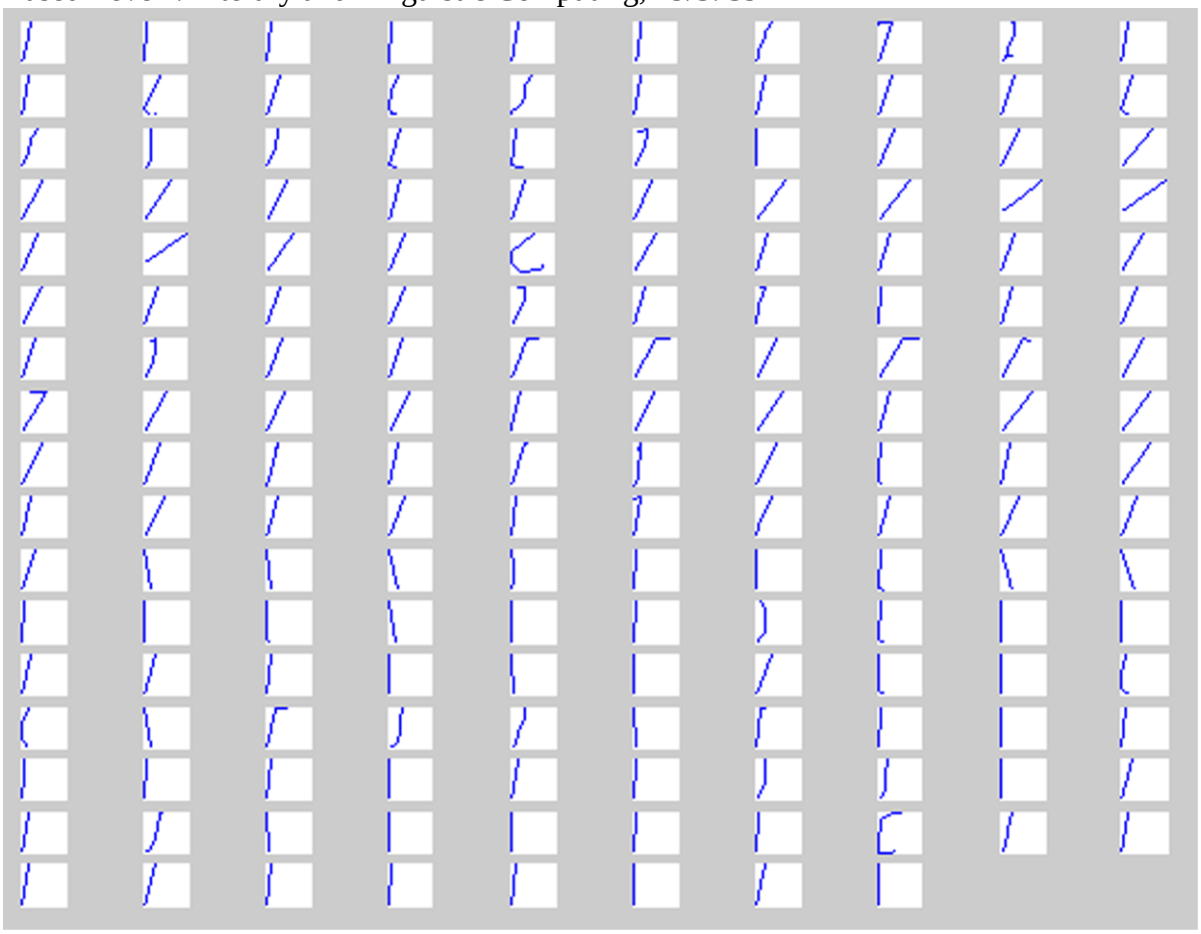

Fig. 7 Every instance of the character I in the ink tablet annotated corpus.

Fig. 8 Every instance of the character I in the stylus tablet annotated corpus.

Fig. 9 The generation of a character model from a small corpus. Three letter S's are identified, and bounding boxes drawn around them (A). The stroke data is then transformed into a 21 by 21 pixel grid (B). A Gaussian Blur is applied (C). The composite images generate a character model (D). The darker the area, the higher the probability of the stroke passing through that pixel. In this way, the probabilities of the stroke data occurring are preserved implicitly in the character models.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414

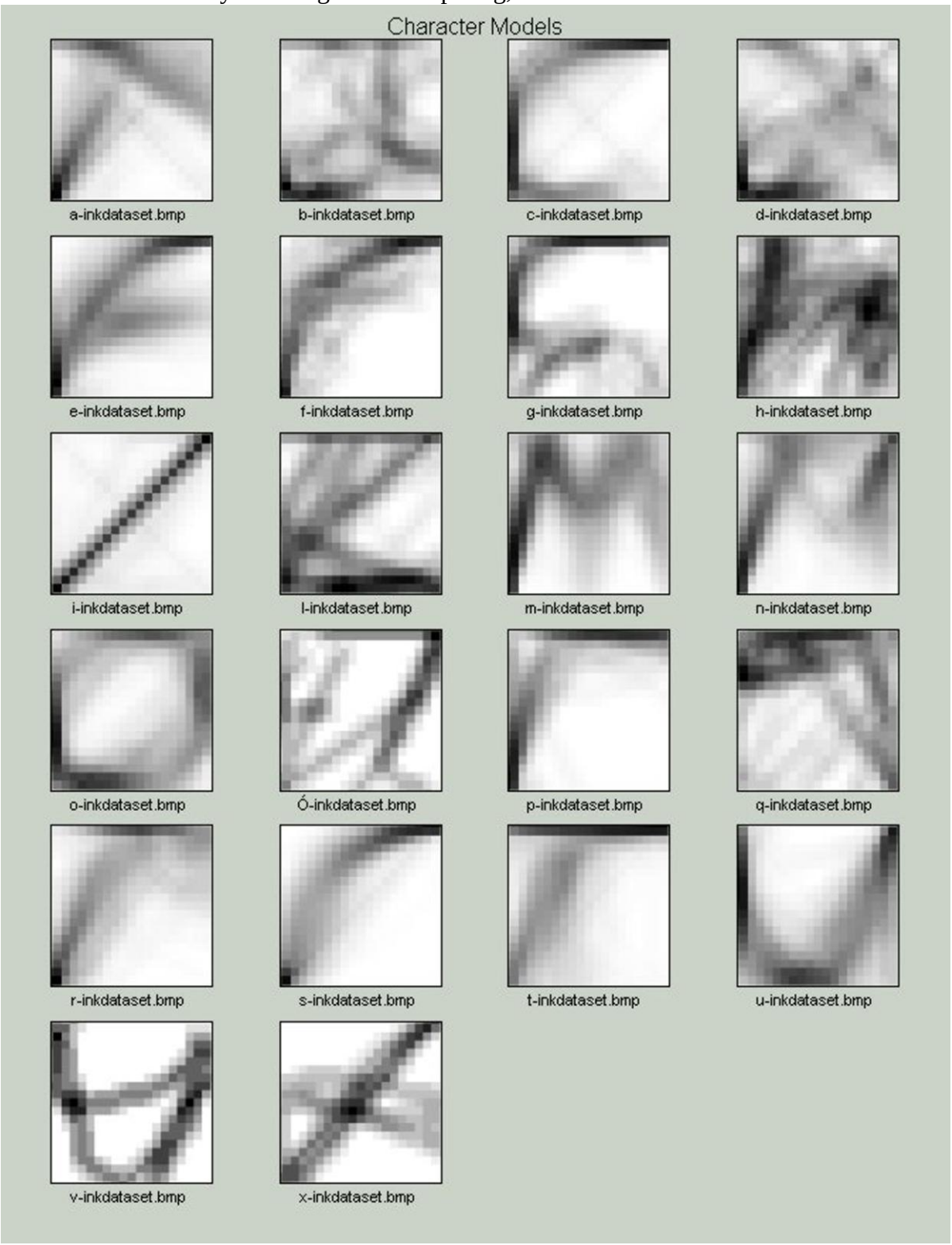

Fig. 10 Character models generated from the training set of the ink text corpus. The darker the area, the higher the certainty that a stroke will pass through that individual section of the character box.

Terras, M. and Robertson, P. (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level". Literary and Linguistic Computing, 19/3. 397 - 414