Image and Interpretation

Using Artificial Intelligence to Read Ancient Roman Texts

Melissa Terras & Paul Robertson

The ink and stylus tablets discovered at the Roman Fort of Vindolanda are a unique resource for scholars of ancient history. However, the stylus tablets have proved particularly difficult to read. This paper describes a system that assists expert papyrologists in the interpretation of the Vindolanda writing tablets. A model-based approach is taken that relies on models of the written form of characters, and statistical modelling of language, to produce plausible interpretations of the documents. Fusion of the contributions from the language, character, and image feature models is achieved by utilizing the GRAVA agent architecture that uses Minimum Description Length as the basis for information fusion across semantic levels. A system is developed that reads in image data and outputs plausible interpretations of the Vindolanda tablets.

The ink and stylus texts from Vindolanda are an unparalleled source of information regarding the Roman Army and Roman occupation of Britain for historians, linguists, palaeographers, and archaeologists.[1] The visibility and legibility of the handwriting on the ink texts can be improved through the use of infrared photography. However, due to their physical state, the stylus tablets (one of the forms of official documentation of the Roman Army) have proved almost impossible to read.

This paper describes a system designed to aid historians in reading the stylus texts: in the process developing what appears to be the first system developed to aid experts in reading an ancient document. This system brings together years of research in image analysis, AI architectures, pa-

Terras, Melissa & Robertson, Paul. "Image and Interpretation: Using Artificial Intelligence to Read Ancient Roman Texts." *HUMAN IT* 7.3(2005): 1–56

pyrology, and palaeography undertaken in the Department of Engineering Science,[2] and the Centre for the Study of Ancient Documents,[3] University of Oxford, to construct a complete AI system that goes from signal to symbol. We describe knowledge elicited from experts working on the texts, and the stochastic Minimum Description Length (MDL) agent architecture used to fuse knowledge of image features, knowledge of Latin texts including character models, and statistical linguistic models of word and character frequency, to arrive at plausible interpretations of the Vindolanda texts. The system described in this paper can read in image data and output textual interpretations of the writing that appears on the documents.

The GRAVA architecture (Robertson 2001) was developed to solve interpretation problems. An interpretation problem is one in which an input, for example an image or signal, must be interpreted in the context of domain knowledge to produce a symbolic representation. Such problems usually have multiple plausible interpretations, and the task involves finding the most likely interpretation. The GRAVA architecture has previously been applied successfully to the interpretation of satellite aerial images, and is adapted here to provide plausible solutions to the interpretation problem of constructing a reading of the Vindolanda texts.

The system is not an "expert system" that automatically "reads" and provides a transcription of the texts; rather it is a papyrologist's assistant that mobilizes disparate knowledge, such as linguistic and visual clues, and uses these to speed up the process by which an expert can arrive at the most likely interpretation of a text.

This paper is divided into sections. The first describes the background of the project, providing information about the Vindolanda texts, and discusses the knowledge elicitation exercises undertaken to make the process the papyrologists use to read ancient texts explicit. The next sketches the parts of the GRAVA architecture used in this system and illustrates the approach with an illustrative example. In the section following that, we discuss how the linguistic and character shape knowledge is modelled within the GRAVA framework: the success of the system is achieved by mobilizing knowledge of the characters and words along with the stroke data from image analysis routines. Then we present results of the running system applied to ink and stylus tablets, and the

final section provides a conclusion, sketching out the successes and possible applications of this research.

The Vindolanda Texts

The discovery of the tablets in Vindolanda,[4] a Roman Fort built in the late 80s AD near Hadrian's Wall at modern day Chesterholm, has provided an unparalleled resource regarding the Roman occupation of northern Britain and the use and development of Latin around the turn of the first century AD. Textual sources for the period in British history from AD 90 to AD 120 are rare, and the ink and stylus tablets are a unique and extensive group of documents providing a personal, immediate, detailed record of the Roman Fort at Vindolanda from around AD 92 onwards (Bowman and Thomas 1983, 1994, 2003; Bowman 1997).

The ink tablets, carbon ink written on thin leaves of wood cut from the sapwood of young trees, have proved the easiest to decipher. In most cases, the faded ink can be seen clearly against the wood surface by the use of infrared photography, a technique used frequently in deciphering ancient documents (Bearman and Spiro 1996). The majority of the six hundred writing tablets that have been transcribed so far contain personal correspondence, accounts and lists, and military documents (Bowman and Thomas 1983, 1994, 2003).

The two hundred stylus tablets found at Vindolanda appear to follow the form of official documentation of the Roman Army found throughout the Empire (Turner 1968; Fink 1971; Renner 1992). It is suspected that the subject and textual form of the stylus tablets will differ from the writing tablets as similar finds indicate that stylus tablets tended to be used for documentation of a more permanent nature, such as legal papers, records of loans, marriages, contracts of work, sales of slaves, etc (Renner 1992), although the linguistic aspects of the tablets will be similar as they are contemporaneous documents from the same source, probably written by the same scribes.

Figure 1. Stylus tablet 836, one of the most complete stylus tablets unearthed at Vindolanda. The incisions on the surface can be seen to be complex, whilst the wood grain, surface discoloration, warping, and cracking of the physical object demonstrate the difficulty papyrologists have in reading such texts.

Manufactured from softwood with a recessed central surface, the hollow panel of the stylus tablets was filled with coloured beeswax. Text was recorded by incising this wax with a metal stylus, and tablets could be re-used by melting the wax to form a smooth surface. Unfortunately, in nearly all surviving stylus tablets the wax has perished,[5] leaving a recessed surface showing the scratches made by the stylus as it penetrated the wax.[6] In general, the small incisions are extremely difficult to decipher. Worse, the pronounced wood grain of the fir wood used to make the stylus tablets, staining and damage over the past two thousand years, and the palimpsestic nature of the re-used tablets further complicate the problem; a skilled reader can take several weeks to transcribe one of the more legible tablets, whilst some of the texts defy reading altogether. Prior to the current project, the only way for the papyrologists to detect incisions in the texts was to move the text around in a bright, low raking light in the hope that indentations would be

highlighted and candidate writing strokes become apparent through the movement of shadows, although this proved frustrating, time consuming, and insufficient in the transcription of the texts.

In 1998 the Department of Engineering Science and the Centre for the Study of Ancient Documents at the University of Oxford were jointly awarded a research grant by the Engineering and Physical Sciences Research Council (EPSRC) to develop techniques for the detection, enhancement and measurement of narrow, variable depth features inscribed on low contrast, textured surfaces (such as the Vindolanda stylus tablets). To date, the project has developed a wavelet filtering technique that enables the removal of woodgrain from images of the tablets to aid in their transcription (Bowman, Brady & Tomlin 1997). In addition, a technique called "Shadow Stereo" or "Phase Congruency" has been developed, in which the camera position and the tablet are kept fixed, but a number of images are taken where the tablet is illuminated by a strongly orientated light source.[7] If the azimuthal direction of the light sources (that is, the direction to the light source if the light were projected directly down on to the table) is held fixed, but the light is alternated between two elevations, the shadows cast by incisions will move but stains on the surface of the tablet remain fixed. This strongly resembles the technique used by some papyrologists who use low raking light to help them read the incisions on the tablet (Terras 2000; Molton et al. 2003). Edge detection is accomplished by noting the movement of shadow to highlight transitions in two images of the same tablet, and so candidate incised strokes can be identified by finding shadows adjacent to highlights which move in the way that incised strokes would be expected (Schenk 2001; Schenk & Brady 2003). Although this is not a standard technique in image processing, encouraging results have been achieved so far, and a mathematical model has been developed to investigate which are the best angles to position the light sources (Molton et al. 2003). Work currently being undertaken is extending the performance and scope of the algorithms (Pan et al. 2004), and the papyrologists are beginning to trust the results and suggestions which are being made about possible incisions on the tablets (Bowman & Tomlin forthcoming). Future work will be done in relating the

parameters of analysis to the depth profile of the incisions to try to identify different overlapping writing on the more complex texts.

Whilst these have had some success in analysing the surfaces of the tablets, it was essential to develop a way of aiding the papyrologists in utilising generated results. Although the developed algorithms could be easily added to readily available image manipulation software,[8] allowing others to apply the algorithms themselves, it would do little to actually provide a tool that would actively help the papyrologists in the transcription of texts; a complex process which has been described as "teasing information out of material which is all too often barely legible, fragmentary, or obscure (or all three at once)," (Bowman 1994, 10). The focus of this paper is to present the possibility of developing an intelligent system that can aid the papyrologists in their task, which will, in the future, dovetail with the image processing work to provide a complete "signal to symbol" program which can produce realistic interpretations of the Vindolanda stylus texts within a reasonable timeframe.

The Process of Reading an Ancient Document

In order to identify the tools that could be built to aid the papyrologists in their transcription of the Vindolanda tablets, it was first necessary to try and gain an understanding of what the papyrology process actually entails. Little investigation has been done so far to ascertain how experts read such damaged and debraded documents (Terras 2002).[9] Techniques borrowed from the field of Knowledge Elicitation (McGraw & Harbison-Briggs 1989; Waterman 1986) were used to gather quantitative and qualitative information about how papyrologists work, resulting in an in-depth understanding of the ways different experts approach and reason about damaged and abraded texts. Firstly, as with all knowledge acquisition tasks, the domain literature was researched, and any associated literature was collated. Three experts were then identified who were working on the ink and stylus texts, and who were willing to take part in this investigation. The experts were observed whilst going about their tasks, and unstructured interviews were undertaken, where the experts described their domain, and the individual processes and techniques that they preferred. More structured interviews were then

undertaken, when the experts were asked to describe particular facets of their task, such as the identification of letter forms and the role of grammar, word lists, and external historical and archaeological resources in the reading of the documents. A series of Think Aloud Protocols (TAPs) were then undertaken (a technique adopted from experimental psychology, where the expert is urged to utter every thought that comes to mind whilst undertaking a specified task) and the experts were given structured tasks to complete. These sessions were recorded, transcribed, and analyzed using Content Analysis techniques (Terras 2002). These exercises demonstrated that the experts use a recursive reading mechanism which oscillates between different levels, or modules, of reading, and the process was rationalized into defined units, to develop a connectionist model of how papyrologists approach and start to understand ancient texts.

An expert reads an ancient document by identifying visual features, and then incrementally builds up knowledge about the document's characters, combinations of characters, words, grammar, phrases, and meaning, continually proposing hypotheses, and checking those against other information, until s/he finds that this process is exhausted. At this point a representation of the text is prepared in the standard publication format. At each level, external resources may be consulted, or be unconsciously compared to the characteristics of the document.

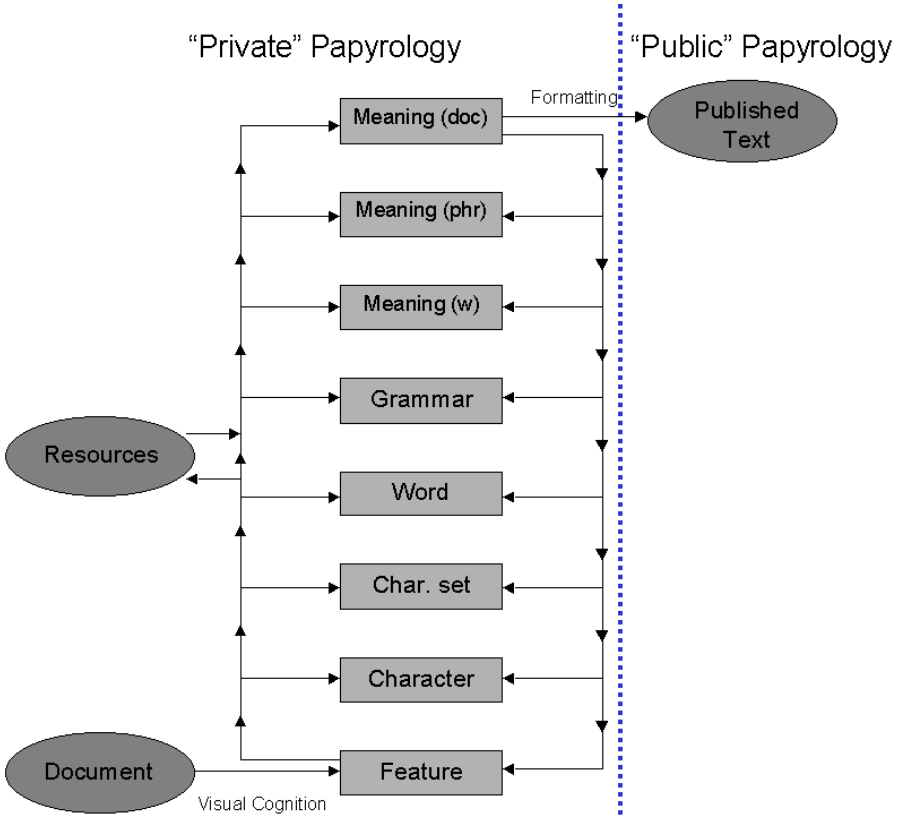


Figure 2. The proposed model of how experts read an ancient text. Public papyrology (Youtie 1963) refers to the published reading of texts in a common format, after the reading has taken place. Private papyrology is the implicit process the experts undertake when reading a text.

Replicating the Process Using Artificial Intelligence

The field of automated handwriting recognition is expansive and complex (see Impedovo (1993) for an introduction), but although there has been a great deal of work on pattern recognition algorithms aimed at recognising text, the vast majority is irrelevant to this research. Most techniques are aimed at recognising printed text, making them incompatible with the hand-written, cursive text found on the Vindolanda tablets.

Work done on hand-written non-cursive text primarily approaches the problem with pattern class or neural network learning: where the system learns individual characters from thousands of examples. There are simply insufficient examples to be able to train such a neural net, even from the corpus of Vindolanda ink texts. Other work, largely from the late 1960s and early 1970s, emphasised "syntactic pattern classification": the idea that a character is composed of strokes that have a certain relationship to each other (see Connell & Brady's approach to shape representation (1987), and Fu & Swain's introduction to Syntactic Pattern Recognition (1969)). The attempts to teach a machine to "read" text in this manner were hampered by the problem of stroke detection: image processing techniques were not developed enough to provide the necessary data. There have been previous attempts to use connectionist inspired models of human reading (see Figure 3) as the basis on which to build systems to "read" cursive handwriting (Dodel & Shinghal 1995; Parisse 1996; Côté et al. 1998). These systems have had limited success: they are dependent on oversimplification of word shape and contour, the models rely on strong contextual constraints, and they are only successful with small lexicons of 25-35 words. Appropriating these systems in an attempt to build a tool to aid the stylus tablets would be unsuccessful for the same reason that existing image processing techniques could not be used to analyse the surface data: the data is too fragmentary, too noisy, and too complex. It had been suggested that using Minimum Description Length may provide a way to successfully model systems when only sparse data exists (Robertson 2001), and this is the technique adopted here.

AI has repeatedly shown that interpreting our world requires bringing to bear a great deal of world knowledge. The process of reading an ancient document, especially documents that are in a very bad state of disrepair such as the Vindolanda stylus tablets, is an especially good example of the need to mobilize a great deal of *a-priori* knowledge. It is clear that the remnants of writing on the tablets themselves contain insufficient information to recover the original written text, but by applying sufficient linguistic knowledge of character shapes, word and letter frequency, and grammatical information, the tablets can sometimes be read. Implementing a system that can automate the process requires

an architecture that can fuse the many different kinds of knowledge in order to arrive at an estimate of the most probable interpretation. Numerous architectures have been proposed to solve interpretation problems: the HEARSAY system (Erman et al. 1980) was developed for speech recognition and more recently Hidden Markov models (HMM) have been used in natural language processing (NLP) (Charniak 1993) but these approaches have drawbacks. Blackboard systems can be hard to control, and HMMs can be too restrictive to incorporate complex and diverse kinds of knowledge (Robertson 2001, Robertson & Laddaga 2003). We developed GRAVA to address these concerns and provide a flexible backbone upon which image interpretations problems can be solved. In the following section we describe the features of GRAVA used in this project.

Solving Interpretation problems with GRAVA

GRAVA (Grounded Reflective Adaptive Vision Architecture), developed by Robertson (2001) provides a way to implement different modules, or agents, of computer programs which can compare and pass different types of information to each other. This agent based system attempts to find interpretations of input data (usually image data) by fitting models to the data. The *best* interpretation is sought by trying to find the simplest explanation of the input data. The motivating concept behind GRAVA is the idea that the input data can be thought of as a message that has been encoded by some process and communicated through a noisy channel.

GRAVA utilises an architecture where the atomic elements are implemented as agents (in Yolambda, a dialect of LISP), using familiar programming practices. The primary purpose of an agent "is to fit a model to its input and produce a description element that captures the model and any parameterisation of the model" (Robertson 2001, 59). The GRAVA system manipulates agents, and builds programs from them. Agent co-operation can span semantic levels, allowing hierarchical stacking. This enables the building of systems that exhibit semantic interaction, a well understood hierarchical concept (McClelland & Rumelhart 1986) that allows the behaviour and performance of systems to be closely monitored and understood (using techniques such as

convergence analysis). Such an architecture is well suited to interpretation problems, which can be solved by fitting an input to a series of models and computing the likelihood of these matching. Many interpretation problems have more than one possible solution, and by using such a system many solutions can be propagated, the best solution being the ultimate target. Of most relevance to *this* paper, Robertson shows how his architecture is an effective way of rendering hierarchical systems by demonstrating how his software can "read" a hand-written phrase. The text in this case was a nursery rhyme. Given that the GRAVA system was shown to be a robust and easily adaptable architecture, it was chosen as the basis on which to develop a system to read the Vindolanda ink, and eventually, stylus tablets.

In this section we provide an overview of the GRAVA architecture by introducing the concepts behind the GRAVA architecture, the objects that make up the architecture, and an overview of their protocols. We conclude the section by developing an illustrative example of how GRAVA agents compete using Minimum Description Length (MDL), showing how GRAVA was used to "read" a handwritten text, and thus provides the basis for the system developed to propagate interpretations of the Vindolanda texts.

Agent Selection Paradigms

Autonomous agents are expected to operate without the intervention of a central control mechanism (such as a focus of control database). One approach to the agent selection problem that has been the focus of considerable attention is the notion of a market based approach. The idea is that an agent wishes to farm out a subtask to another agent capable of performing the subtask. Agents that are candidates to perform the subtask compete by bidding a price. This often works well, producing efficient solutions. However, two problems arise in such systems:

1. Selecting an appropriate basis for cost computations so that the bidding is fair (so that different types of information can be compared across different semantic levels).

2. Because the bidding is piecewise local, such systems are prone to find local minima and miss the global minima (i.e. the best match on one semantic level may not be the correct match overall).

Our approach addresses these two problems as follows: the basis for cost computation used is description length, and the use of a Monte Carlo sampling technique avoids the problem of being restricted to the best local solution. These concepts are described more fully here, before discussing the MDL GRAVA architecture in detail.

Minimum Description Length

A major puzzle in perceptual psychology has been how the brain reconciles different sorts of information, for example colour, motion, boundaries of regions, or textures, to yield a percept (Eysenck & Keane 1997). For example, in image segmentation, by changing the texture model it is possible to increase or decrease the number of regions that are identified. Similarly, the number of boundary shapes that are found can be increased or decreased by changing the search parameters. The reading of ancient documents is just one (complex) example of an interpretation problem, where the individual is faced with visual ambiguity and competing information, which has to be reconciled and resolved with other types of information (in this case linguistic data) to generate a plausible solution. There has been substantial consideration, by psychologists and image processing experts, of how these different processes are combined. One suggestion is that there is a common value that can be used to calculate the "least cost" solution when comparing different types of information. This has been adopted by the field of Artificial Intelligence, in the concept of Minimum Description Length: (MDL).

First introduced in the late 1960s, and developed in the 1970s, (Wallace & Boulton 1968; Rissanen 1978), MDL applies the intuition that the simplest theory which explains the data is the best theory.[10] MDL can be used as a means of comparing data in coding theory, in which the goal is to communicate a given message through a given communication channel in the least time or with the least power. MDL is a very powerful and general approach which can be applied to any

inductive learning task, and can be used as a criterion for comparing competing theories about, or inductive inferences from, a given body of data. It is well suited to interpretation problems: where solutions can be generated by comparing unknown data to a series of models, when the most likely fit can generate plausible solutions to the problem.

Description length is the correct measurement in an interpretation problem because it captures the notion of likelihood directly. Description length is calculated from the probability of an input, such as the data describing the shape of an unknown character, matching a known model. By adding the description lengths of the outputs of different agents, it is possible to generate a global description length for the output of the whole system. The smallest global description length generated from many runs of a system is most likely to be the correct solution to the problem: the Minimum Description Length, or MDL. This is the approach adopted in the GRAVA system.

Monte Carlo Select Methods

The MDL Agent architecture described in this paper addresses the need to integrate knowledge at different semantic levels. To understand an image that consists of high-level components such as words, intermediate level features such as characters, and of low-level features such as strokes and endpoints, we need to integrate different levels of processing.

Single thread of control solutions, such as the blackboard and forward chaining approaches, depend upon taking a path towards a solution and backtracking past failures until a solution is found (Erman et al. 1980). The same is true of subsumption (Brooks 1986). These are essentially depth first searches for a solution – not searches for the *best* solution. In a robot navigation problem, we may be happy if the robot negotiates the obstacles in the environment and finally ends up at the destination. In interpretation problems, just finding a solution is not good enough. A Latin sentence can have many plausible parses. Most of them do not make sense.

Markov Chain/Monte Carlo methods (MCMC) have become popular recently in computer vision (Geman & Geman 1984; Hammersley & Handscomb 1964; Lau & Ho 1997) but have been limited to modelling low-level phenomenon such as textures (Karssemeijer 1992).

In natural language understanding, use of hidden Markov models has been successful as an optimisation technique with certain restricted kinds of grammar. Problems that can be described as Hidden Markov Models (HMM) (Baum 1972) can yield efficient algorithms. For example, in natural language understanding, some grammars permit efficient algorithms for finding the most probable parse. Stochastic Context Free Grammars (SCFGs) can be parsed so as to find the most probable parse in cubic time using the Viterbi algorithm (Viterbi 1967). Only the simplest grammars and problems can be solved efficiently in this way, however, and for the more interesting grammars and for more complex problems in general, other techniques must be used. Certainly something as loosely defined as an agent system incorporating semantics from multiple levels would rarely fit into the HMM straitjacket.

Even for natural language processing, finding the best solution can be prohibitively expensive when the Viterbi algorithm cannot be employed. In visual processing, with images whose complexity greatly exceeds that of sentences, and which are three dimensional (as opposed to the linear arrangement of words in a sentence), finding the best solution is infeasible. Approximate methods are therefore necessary: Monte-Carlo techniques are very attractive in these situations.

In an ambiguous situation, such as parsing a sentence, in which many (perhaps thousands) legal parses exist, the problem is to find the parse that is the most probable. If the problem can be defined in such a way that parses are produced at random and the probability of producing a given parse is proportional to the probability that the parse would result from a correct interpretation, the problem of finding the most probable parse can be solved by sampling. If P is a random variable for a parse, the probability distribution function (PDF) for P can be estimated by sampling many parses drawn at random. If sufficiently many samples are taken, the most probable parse emerges as the parse that occurs the most frequently. Monte Carlo techniques use sampling to estimate PDF's.

Monte Carlo methods are attractive for a number of reasons:

1. They provide an approximate solution to the search of combinatorially prohibitive search spaces.

2. By adjusting the number of samples, the solution can be computed to an arbitrary accuracy.
3. Whereas the best solution cannot be guaranteed by sampling methods, measuring standard error during the sampling phase allows the number of samples to be adjusted to yield a desired level of accuracy.

GRAVA employs a Monte Carlo method, a means of providing approximate solutions by performing statistical sampling, to randomly choose which data is passed upwards between levels. This is a "weighted random" selection process which picks likely data much more frequently than less likely examples (see Robertson 2001, 48). If only the data with the lowest Description Length was passed up between levels, the correct answer may never be found: the data with the locally Minimum Description Length may not be the correct selection on the global level. The stochastic nature of this method of sampling ensures the generation of different results, and also means that the system rapidly generates possible solutions without relying on exhaustive search (cutting search time). The system generates possible solutions on each iteration; the more iterations, the better the chance that a match to the solution is generated. Convergence on an ideal solution is then asymptotic: the system finds approximate solutions, and the more iterations that occur, the better the approximation that is reached. In practice, the system tends to find the exact solution in a short number of iterations, meaning that performance times are acceptable (this is demonstrated in the "Results" section below).

MDL Agent Architecture

Interpretation problems represent an interesting and important class of problems. Many important problems in AI can be characterized in this way. Our architecture is designed to allow complex programs to be structured around the notion of finding the most likely interpretation. Frequently our interpretations at one level affect interpretations at other levels of processing. Sound and image fragments taken in isolation are often unintelligible because without context there is not enough information to construct a reasonably likely interpretation of the frag-

ment. This is particularly evident in the problem of interpreting Vindolanda writing tablets.

A recurring issue in multi-agent systems is the basis for cooperation among the agents. Some systems assume benevolent agents where an agent will always help if it can. Some systems implement selfish agents that only help if there is something in it for them. As a direct result of the use of Monte Carlo sampling, agents in GRAVA appear to cooperate in order to achieve a global MDL. Agent cooperation is an emergent property of the architecture since the agents do not directly engage in cooperative behaviour. Rather agents are selected by Monte Carlo sampling that happen to cooperate to produce the global MDL. Our approach to agent cooperation involves having agents compete to assert their interpretation. If one agent produces a description that allows another agent to further reduce the description length so that the global description length is minimized, the agents *appear* to have cooperated. Locally, the agents compete to reduce the description length of the image description. The algorithm used to resolve agent conflicts guarantees convergence towards a global MDL thus ensuring that agent cooperation "emerges" from agent competition. The MDL approach guarantees convergence towards the most probable interpretation.

In the following subsection, we describe the architectural objects that implement these concepts in the GRAVA architecture.

Objects in the GRAVA Architecture

The architecture is built from a small number of objects: Models, Agents, Interpreters, and Descriptions.

All of these terms are commonly used in AI literature to mean a wide range of things. In the GRAVA architecture they have very specific meanings. Below, we describe what these objects are and how they cooperate to solve an interpretation problem.

The architecture takes an input description Δ_{in} and produces an output description Δ_{out} as its result. A description consists of a collection of description elements $<\varepsilon_1, \varepsilon_2,..., \varepsilon_n>$. The output description is an interpretation $(I \in Q(\Delta_{in}))$ of the input where $Q(x)$ is the set of all possible interpretations of x.

$$\Delta_{out} = I(\Delta_{in})$$

The goal of the architecture is to find the best interpretation I_{best} that is defined as the interpretation that minimizes the global description length.

$$\arg\min_{I_{best}} DL(I_{best})(\Delta_{in})$$

Descriptions and Description Elements

A description Δ consists of a set of description elements ε.

$$\Delta = < \varepsilon_1, \varepsilon_2, ..., \varepsilon_n >$$

Agents produce descriptions that consist of a number of descriptive elements. The descriptive elements provide access to the model, parameters, and the description length of the descriptive element. For example, a description element for a character might include a representation of the strokes that make up the character. A description element is a pair consisting of a model and its parameters.

Description elements are produced by *agents* that *fit models* to the input. Description elements may be implemented in any way that is convenient or natural for the problem domain.

Models

Fitting a model to the input can involve a direct match but usually involves a set of parameters. Consider as input, the string "t h r e e b l i n d m i c e". We can interpret the string as words. In order to do so, the interpreter must apply word models to the input in order to produce the description. If we have word models for "three", "blind", and "mice" the interpreter can use those models to produce the output description:

```
((three) (blind) (mice))
```

The models are parameterless in this example. Alternatively we could have had a model called "word" that is parameterized by the word in question:

```
((word three) (word blind) (word mice))
```

In the first case there is one model for each word. In the case of "three" there is an agent that contains code that looks for "t", "h", "r", "e", and "e" and returns the description element "(three)". In the second case there is one model for words that is parameterized by the actual word. The agent may have a database of words and try to match the input to words in its database.

Agents

The primary purpose of an agent is to fit a model to its input and produce a description element that captures the model and any parameterization of the model. An agent is a computational unit that has the following properties:

1. It contains code that is the implementation of an algorithm that fits its model to the input in order to produce its output description.
2. It contains one or more models (explicitly or implicitly) that it attempts to fit to the input.
3. It contains support for a variety of services required of agents such as the ability to estimate description lengths for the descriptions that it produces.

The "fit" method returns a (possibly null) list of description elements that the agent has managed to fit to the data. The interpreter may apply many agents to the same data. The list of possible model fits from all applicable agents is concatenated to produce the candidate list from which a Monte Carlo selection is performed.

An Illustrative Example of MDL Agents

A key idea of this paper is that of agent cooperation that spans semantic levels. The idea of semantics affecting lower level processes is not new. In the late 1970's there was much interest within AI concerning heter-archical-programming approaches (Fahlman 1973; McDermott & Sussman 1973). These systems suffered from behaviour that defied understanding and, equally importantly, analysis. Marr noted that in the human visual system that there was no evidence that higher level

processes had any control over lower level vision and that we could profitably study low-level vision without addressing such issues. Computer vision has largely followed that path for the past 20 years. The issue has not gone away, however, and the notion of how differing levels of semantic interpretation are made to interact is at the very heart of the AI problem. One particularly interesting approach is the parallel distributed processing (PDP) work of Rumelhart and McClelland (McClelland & Rumelhart 1986). The PDP approach is to develop neural network architectures that implement useful processes such as associative memory. Of particular relevance to the MDL agents described in this paper is an Interactive activation model for word perception developed by Rumelhart (McClelland & Rumelhart 1986b).

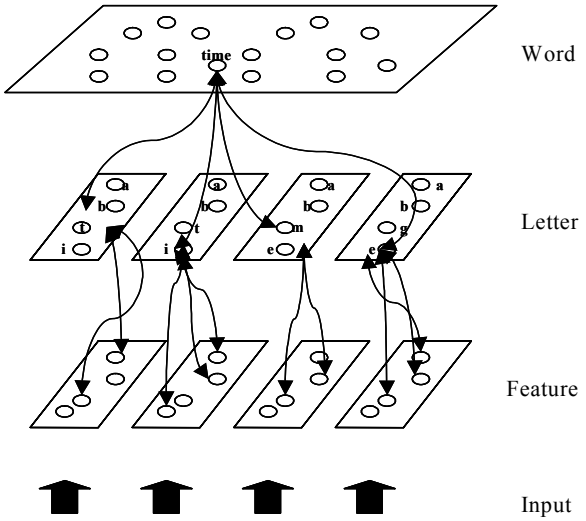


Figure 3. Interactive activation model for word perception

Figure 3 shows the multi-level word recognition architecture. Each plane contains mutually exclusive possibilities for an interpretation. So for example the word "Time" is exclusive of the word "Trim'" and the letter "T" is exclusive of the letter "J". If the word is "trim" it is not "time" so the more we believe the word to be "trim" the less we believe it

is "time". This is modelled in the McClelland and Rumelhart architecture by forming inhibitory links between words on the same level. Similarly on the character level the more that we believe a character is "T" the less we believe that it is "J" (or any character other than "T") and this, too, they implement by building bi-directional inhibitory links between all characters. Between levels the connections are excitory because believing that the First character is "T" improves the likelihood that the word is "Time" or "Trim" so there is an excitory link between the letter "T" and all words that start with "T". The excitory links are also bidirectional since believing that the word is "Trim" increases the likelihood that the character is "T".

The PDP approach depicted in Figure 3 has a number of problems, including how neural systems of this form can be mapped over an input string. Even if these issues were to be successfully addressed, however, the approach throws out the programming model completely and leaves us to develop a completely new computation model based on neuronal systems.

A goal of the GRAVA architecture is to enable systems to be built that exhibit the kind of semantic interaction sought by these earlier systems that:

1. is based on a well understood concept that permits the behaviour to be reasoned about (unlike the heterarchical approaches); and
2. retains a more conventional programming model (than the PDP approach).

The former is achieved by appealing to well-understood ideas from communication theory and Monte-Carlo methods (Shannon 1949, Hammersley & Handscomb 1964), and the latter is achieved by providing an architecture in which the modules of competence are implemented as agents using familiar programming practices.

In this subsection we demonstrate this by solving the problem that McClelland and Rumelhart solved using PDP but within the GRAVA architecture. This serves to clearly illustrate the power of the architecture

on a simple problem and provides a sketch of an approach to interpreting Vindolanda tablets.

Recognizing a Hand Written Phrase

Figure 4. Nursery Rhyme Test Data

The experiment described here was trained on a very small corpus and tested on the hand-written phrase "MARY HAD A LITTLE LAMB" which is shown in figure 4. Although the test data is simple the system illustrates well the approach described in this paper and its robustness. In this example, illustrated in figure 5, there are three levels of agents. The lowest level agents detect low-level features of the written text. There are four feature detectors at this level. Each agent reports on features discovered within a character position.

1. Top stroke endpoints. This agent reports on the number of stroke endpoints at the top of the character. For example the letter "N" has one stroke endpoint on the top and "W" has two.
2. Bottom stroke endpoints. This agent reports on the number of stroke endpoints at the bottom of the character. For example the letter "A" has two endpoints at the bottom of the character and the letter "I" has one.
3. Stroke Junctions. This agent reports on the number of line junctions formed from three or more lines. For example the letter "A" has two such junctions. The letter "N" has none.
4. Character present. This agent detects whether the character position contains anything at all. Everything but the space character contains something.

The character boxes are divided into "top" and "bottom" so that the top stroke and bottom stroke features can be determined from a simple line endpoint filter. The system contains two very simple filters that detect line endpoints and complex junctions.

In order to simulate the dearth of low-level cues typical of vision problems, we designed this experiment to utilize feature detectors that are by themselves incapable of uniquely identifying the characters in the handwritten text. Even if there is no noise and the feature detectors detect their features with 100% accuracy there is insufficient information using only the features described above to unambiguously identify a character. For example 'S', 'C', 'I', 'L' and 'N' all have one endpoint at the top, one at the bottom, and no junctions.

The next level of agents attempts to build character descriptions from the evidence collected by the feature detectors. The character agent contains a database that has information in the form of a set of models. The models represent evidence from low-level sensors and the frequency of the letters in the nursery rhyme. In this case the nursery rhyme acts as a corpus from which statistical evidence is collected and from which the models are constructed.

The word agent attempts to find evidence for words in the input by looking at evidence from the character finding agents. The universe of possible words is derived from the words found in the corpus. In this case, the full corpus of words is restricted to that found in the nursery rhyme. The correct interpretation of any test data will be expected to match words found in the corpus.

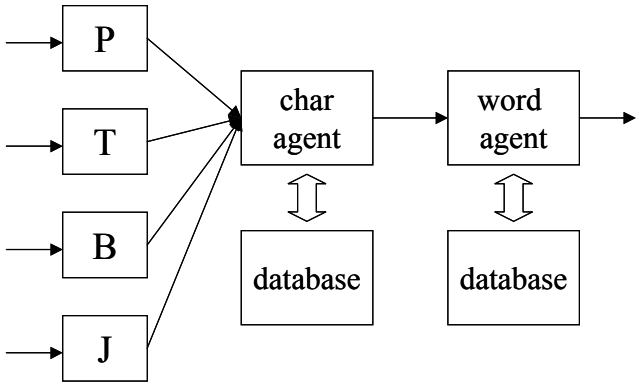


Figure 5. Hand Written Phrase Recognizer Program

All together the following full corpus of symbols can be used to build a description of the input:

- *Low level features*: t=0, t=1, t=2, b=0, b=1, b=2, j=0, j=1, j=2, p=0, p=1.
- *Character level symbols*: A, B, C, D, E, F, G, H, I, L, M, N, O, R, S, T, U, V, W, Y, Space, and
- *Word level symbols*: A, And, As, Everywhere, Fleece, Go, Had, Its, Lamb, Little, Mary, Snow, Sure, That, The, To, Was, Went, White.

Performance of the Illustrative Example

The following run shows the result of running this simple example with the six agents described above (as shown in Figure 5).

```
=> (runCycles #t)

Description Length=306.979919
Description=(t=0 b=0 j=0 p=0 t=0 b=2 j=0 p=1 t=0 b=2 j=2 p=1
             t=0 b=2 j=2 p=1 t=2 b=1 j=1 p=1 t=0 b=0 j=0 p=0
             t=2 b=2 j=2 p=1 t=0 b=2 j=2 p=1 t=0 b=0 j=0 p=1
             t=0 b=0 j=0 p=0 t=0 b=2 j=2 p=1 t=0 b=0 j=0 p=0
             t=1 b=1 j=0 p=1 t=1 b=1 j=0 p=1 t=2 b=1 j=1 p=1
             t=2 b=1 j=1 p=1 t=1 b=1 j=0 p=1 t=2 b=1 j=1 p=1
```

```
             t=0 b=0 j=0 p=0 t=1 b=1 j=0 p=1 t=0 b=2 j=2 p=1
             t=0 b=2 j=0 p=1 t=0 b=0 j=2 p=1)

Description Length=116.690002
Description=(M A A E HAD A   I L T E S T   I R M B)

Description Length=65.699996
Description=(M R A Y HAD   R LITTLE LAMB)

Description Length=61.749996
Description=(M R A E HAD A LITTLE LAMB)


Description Length=41.649997
Description=(MARY HAD A LITTLE LAMB)
```

The agents start out with a description based on the identifications of the low level agents that detect tops, bottoms, junctions, and non-space. The description length is calculated as the sum of the description lengths of each of the symbols in the description and in this case comes to 306.979919. By the second result the description length has improved to 116.690002. This has involved identifying characters from the low level cues. As can be seen from the description there are a number of errors in the interpretation due to ambiguity. The word 'HAD' is correctly identified but in the remaining description there are 9 character identification errors out of 18. By the third result matters have improved so that the description length is 65.699996, three words are correctly identified and 3 character identification errors remain. The next improvement brings the number of correct words to four and still has three character errors. Finally the sentence is correctly interpreted with a description length of 41.649997 and no errors.

This example, while simple, illustrates two important properties of the architecture:

1. *MDL formulation leads to the most probable interpretation.* The final interpretation was indeed the correct one.
2. *Global MDL mobilizes knowledge to address ambiguities.* In the second result of this example fully half of the characters were 'guessed' wrongly by the system. By the fifth result all of the ambiguous choices had been made correctly because the correct

choices were what lead to the globally minimum description length.

While the input data is inadequate for the correct and unambiguous identification of the characters, the semantic constraints allow correct identifications to be made. The semantic constraints are propagated by letting competing agents 'win' in proportion to the reduction in global description length. The Monte Carlo sampling used in this algorithm prevents the system from getting wedged in local minima and ensures that the globally minimum description length (i.e. the most probable interpretation) is converged upon.

Applying the GRAVA System to the Vindolanda Writing Tablets

In order to be able to utilize the GRAVA system to aid in the reading of the Vindolanda Texts, information regarding the character forms and language used at Vindolanda had to be collated, and a more sophisticated character agent had to be developed. This section details the development of the character and word agents that were applied to images of the Vindolanda tablets.

Collating Corpus Data from Vindolanda

Palaeographic and linguistic data regarding the characters and language used at Vindolanda was collated. A markup scheme was developed to annotate a corpus of images of the ink and stylus texts (using the GRAVA annotation tool), resulting in an XML representation of each character on a stroke by stroke basis (Terras & Robertson 2004). In total, 1506 individual characters from the ink tablets were annotated, and 180 characters from the stylus tablets. Word lists were compiled, and lexicostatistics regarding the frequency of words, characters, and combinations of characters used at Vindolanda were generated (Terras 2002). This data provided the necessary corpus data to build the character and word models for use in this application of the GRAVA system.

System Development and Architecture

The character database formed by annotating images of the ink tablets is the main source of information regarding the character forms on the

Vindolanda Texts available. The models derived from this data are used to compare the unknown characters in the test set (which may be annotated by hand, or generated automatically by image processing algorithms that allow feature extraction (see the section "Analyzing Automated Data" below, as well as Robertson et al. 2005). The difference between the original system (presented earlier) and the new system lies in the way that character models are developed, and how the test data is compared to this set of character models. Whereas the original system relied on endpoints, this final version relies on data regarding the strokes themselves. A stroke detection agent replaced the endpoint agents in the feature level of the original system. This results in models of characters that are less sensitive to the feature detection process (i.e. the generation of end points, which is problematic when dealing with noisy images such as those of the stylus tablets). It also means that the feature level agents depend on information that is much more easily propagated from automatic feature detection, allowing for easier amalgamation with the stroke detection system. Most importantly, stroke information is much more stable than endpoint data. Small changes in image quality cause only small changes in the stroke features detected. A schematic of the final system that was developed is shown below (figure 6), incorporating all elements of the resulting process.

Before being used by the character agent, both the test set of data and the annotated corpus must be prepared in order to analyze their stroke data. This is done by extracting the strokes, drawing a bounding box around each of the characters to preserve groupings of strokes, and transforming these strokes onto a canonical sized grid to allow easy comparison.

Character models are built from the annotated set of images from the corpus by applying a Gaussian blur operator, and summing every instance of each individual character to build up a character model. The character agent then compares the unknown characters from the test data with those in the character models, also utilizing frequency information about each character to generate a description length for each model. One of these likely characters is then selected using the Monte Carlo sampling method and passed up to the Word agent. This ensures that,

over successive iterations, a fair, representative amount of each candidate character is selected and passed onto the next level.

The Word level takes in the data from the Character agent, combines them to form words, and compares them with the words from the corpus – the word "models" in this case being the words found in the corpus. A description length for this comparison is noted. A selection is made from the possible words utilizing Monte Carlo sampling methods, and the final word output is generated.

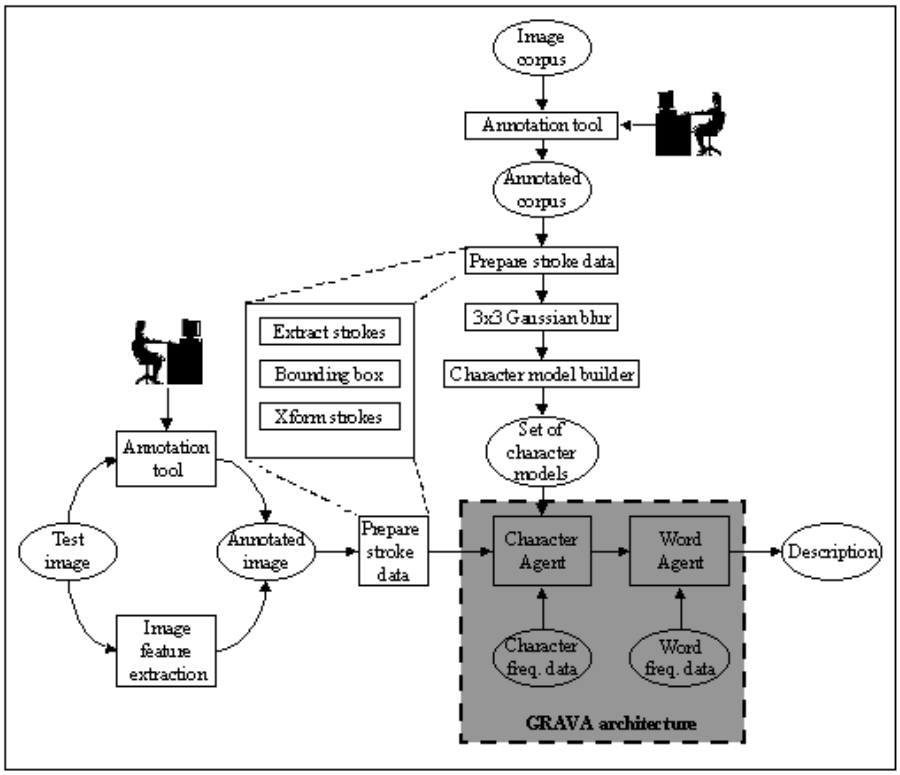


Figure 6. Schematic of final system. GRAVA architecture is highlighted to indicate the processes which are carried out as part of the final run of the system. The "Annotated image" can be generated manually by an "Annotation tool", or automatically by "Image feature extraction".

The system then adds the description lengths for all the words in the phrase (or string of words) together, giving a global description length for that combination of characters and words.

The system repeats this process as often as the user dictates, and keeps track of the lowest global description length generated by each successive run. The Minimum Description Length produced corresponds with the most likely answer: or the best fit answer available.

The preparation of the character models, and the way that both the character and word agents work, is discussed in detail below, before demonstrating results from this system.

The Construction of Character Models

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. Whilst the first implementation of the system relied on an end point agent, this was replaced by a stroke detection agent that builds up character models based on the actual strokes of the character, in order for this system to work with data generated from image processing techniques, which would allow the system to become automated in the future.

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 convolved with a Gaussian Blur operator to reduce over-fitting. 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.

Calculating the Final Model

The first character is drawn onto a "blank" grid. A second instance of the character (if there is one available) is drawn over this, the values of this being added to the first instance. Additional instances of the character are laid over the grid, and the values summed as they go. This results in a composite model of all available character instances from the corpus, showing the path the strokes are most likely to make.

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.

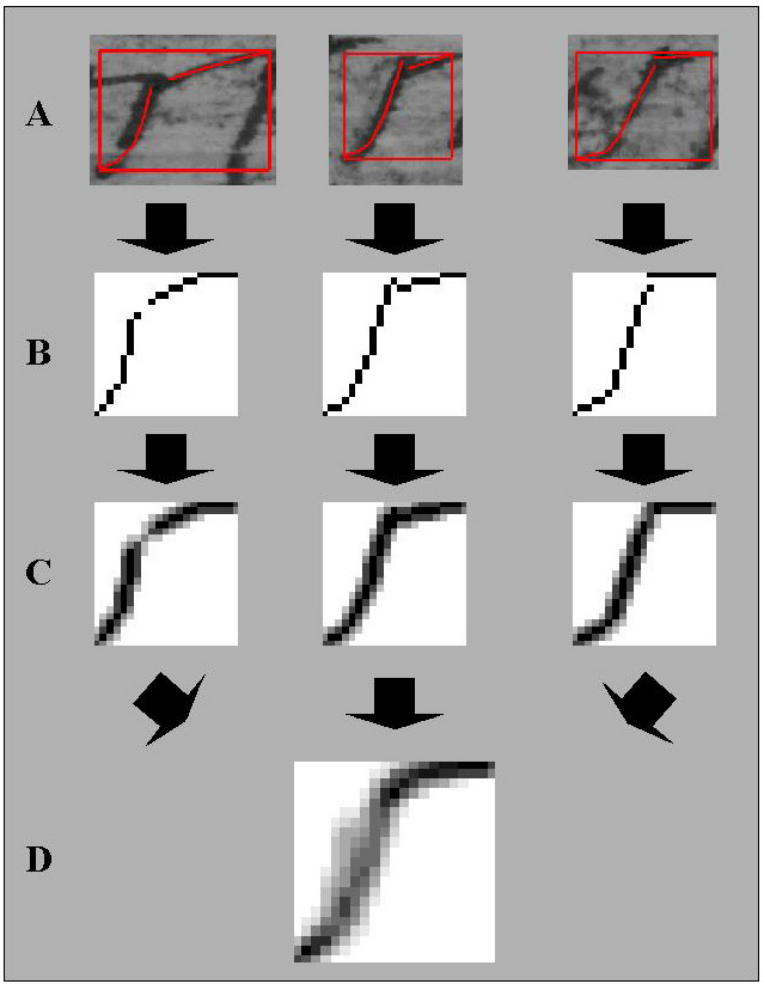


Figure 7. The generation of a character model from a small corpus. Three letter S's are identified, and bounding boxes drawn around them (A). The stroke data is then transformed into a 21 by 21 pixel grid (B). A Gaussian Blur is applied (C). The composite images generate a character model (D). The darker 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.

Learning Models from the Corpus

The corpus of annotated images was randomly divided into a test set and two training sets: that of the ink and the stylus tablets. Keeping the test and training data separate allows fair results to be generated. Two sets (ink and stylus) of character models were generated from the training set; the ink text character models are shown below.

The character models generated from the ink data show that some of the letter forms, such as A, C, I, M, N, and T are fairly standardized throughout the test data. Other characters are more problematic. H, L, and V have a more confused appearance, indicating greater variability in the appearances of instances of the characters. D and Q are problematic, as the long strokes can either slope diagonally left to right, or right to left. A bi-product of this research is that the letterforms generated from this data can be compared to the letter forms generated from the stylus tablet data, generating new information regarding the palaeography of the Vindolanda texts, which is of great interest to papyrologists and palaeographers (Terras 2002).

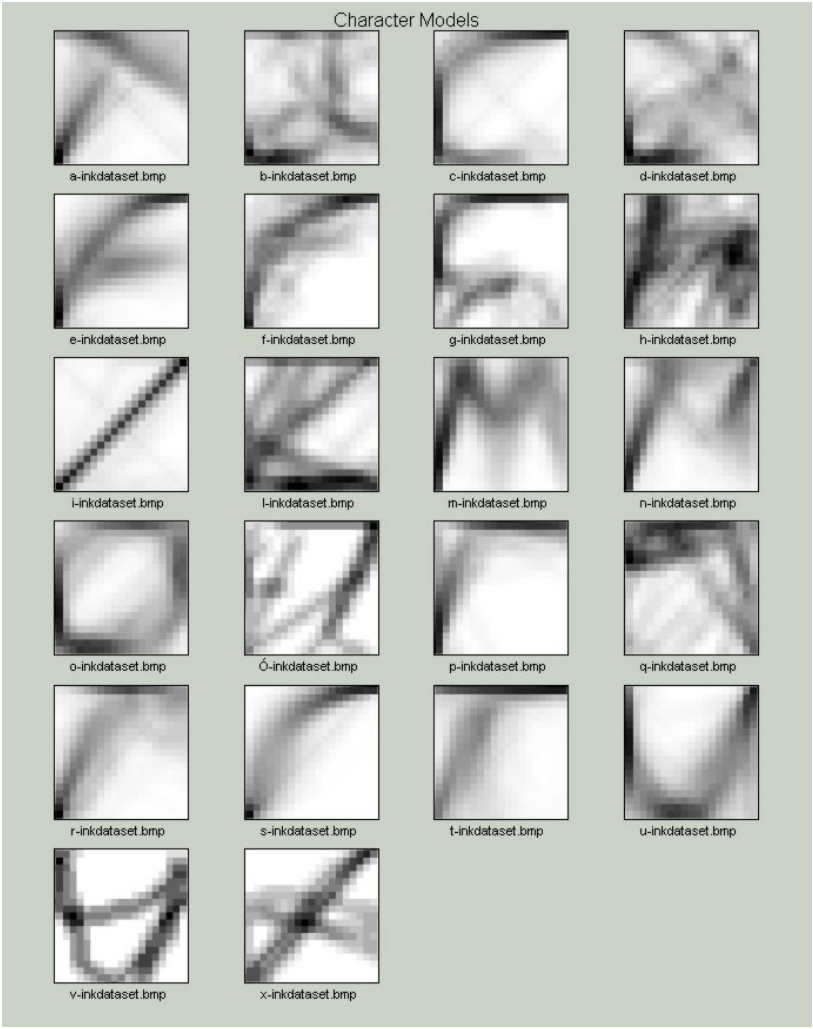


Figure 8. Character models generated from the training set of the ink text corpus. The darker the area, the higher the certainty that a stroke will pass through that box.

Not all of the letterforms are present in stylus text character models, as the data set of the stylus tablets was much smaller than that of the ink tablets and some characters are missing from the sample set. Neverthe-

less, both sets of models provide adequate representations for comparison with unknown characters: missing characters from the stylus set can be replaced with those from the ink set.

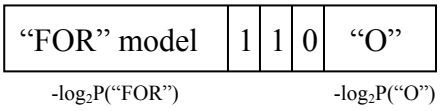
The Character Agent: Comparing Unknown Characters to the Character Models

The character agent's function is to compare unknown characters to the character models composed from the training set. This is achieved by extracting the strokes from the test data, transforming them to the standard size, then calculating the description length for matching the unknown character to each model in the data set. The character agent also utilizes statistical information about the likelihood of a character being present, derived from the letter frequency analysis of the corpus. After the description length has been calculated (from a combination of the MDL frequency and MDL comparison of stroke evidence), one of the likely characters identified is selected, using the Monte Carlo sampling methods, and passed up to the word level.

Comparing Unknown Words to the Word Corpus

The Word agent's function is to compare strings of possible characters to the word models in the corpus, in much the same way as the character agent compares stroke data to models generated from the corpus. However, the word agent's task is considerably simpler than that of the character agent, as there is no need to represent stroke data. The word "models" are the words in the corpus themselves, and the probability of them occurring is known from the word frequency data.

This is best understood by considering a possible message representing the fit of a word model to a character sequence. Consider fitting the word "foo" to the word model "for". The message begins with a representation of the model for the word "for", then for each character there is either a "1" bit, indicating that the character in that position matched the model, or a "0" bit indicating that the character didn't match, followed by a representation of the character that failed to match. For this example the message stream will look like this:



The description length of the match is therefore the code length for the model used, one bit for each matching character, and one bit plus the code length for each non-matching character.

The system compares the string of characters to each individual word model in the corpus. The comparison with the lowest DL generates the most likely word identification.

However, what if the string of letters represents a word that is *not* in the corpus? The system compares the string to all available models, and also generates the total description length from the sum of all of the characters' description lengths. If there is no match between the sequence and the existing models, this DL will be the lowest, and the solution is presented as a string of characters. However, due to the fact that bi-graph information is included in the word agent, the string with the minimum DL will be statistically the most likely combination of characters, even though a word has not been matched to the input. This can be seen in the results section below.

Performance of the Word Agent

The string of characters "U S S I B U S" was fitted to all 342 of the word models that had a string length of 7 characters. Only the results with a description length less than 36 bits are presented here (these being the lowest 8 description lengths computed.) It is clear that the string USSI-BUS, present in the corpus, is the closest match, as this produces the lowest description length. Four other words (or word fragments present in the corpus) are identified as being as probable as the string of characters itself, SCRIBUS, SCRIBAS, VEHIMUS, UIDEDUN. This indicates how the system propagates best-fit solutions, which are approximate to the correct solution.

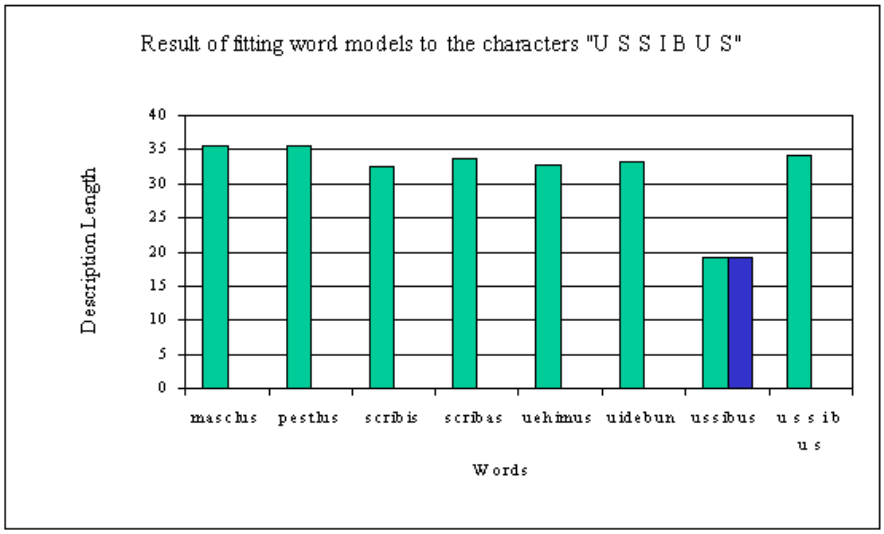


Figure 9. Result of fitting word models to the characters USSIBUS, most likely fit. The word USSIBUS is by far the most likely contender.

Again, the system chooses which word to select as a solution using a Monte Carlo selection algorithm. The system does not incorporate any other data regarding word sequencing or grammar, at this stage. The MDL for a string of words is calculated by simply summing the description length for each word. Subsequent iterations produce different sequences of characters and words. The most probable solution is that with the lowest MDL after a number of successive runs.

Results

The adapted version of the GRAVA system was used to analyze various sets of test data, to see how effective it could be in producing the correct "reading" of a text. Firstly, a section of tablet 255 was analyzed, using the character models ascertained from the ink tablet corpus as the basis of comparison. This gave encouraging results, and also shows the asymptotic nature of the system's convergence on a solution. This experiment was then repeated with the same section of tablet, which in this case had been annotated in a different (wrong) manner, to see how

the system coped with more difficult data. A section of stylus tablet was then analyzed, using firstly the set of character models derived from the ink corpus, and secondly the set of character models derived from the stylus corpus, to indicate how successfully the system operates on the stylus texts. Finally, a section of ink tablet which had been automatically annotated was analyzed, to indicate if this implementation of the system provided a possible solution to the problem of incorporating data generated from the image processing algorithms into a knowledge-based system.

Using Ink Data for Ink Tablets

The first complete run was done on a section of ink tablet 255.

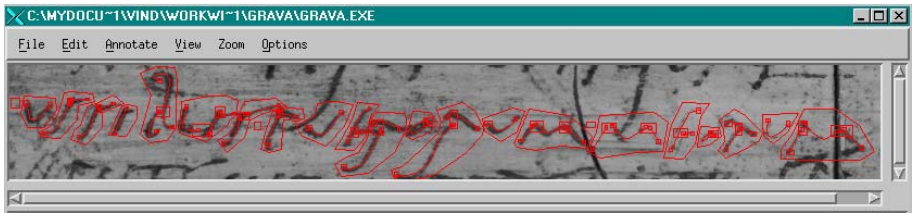


Figure 10. A section of ink tablet 255, annotated with "ussibus puerorum meorum".[11]

The set of character models used in this run was that derived from the ink tablet corpus. The output of the system is shown below.

```
Grounded Reflective Agent Vision Architecture (GRAVA) Version 2.0.
Yolambda listener pushed. Type :exit to return to GRAVA.

=> ... load the system and the data ...

=> (runCycles 25)
iteration 0 DL=440.220794 interpretation=( ... ((2482 252) (2517 250))) ... )
iteration 1 DL=64.085075 interpretation=(  u r s i b u s puerorum  m n o a u m  )
iteration 2 DL=49.374412 interpretation=(ussibus puerorum  m n o r u m  )
iteration 3 DL=48.831413 interpretation=(ussibus puerorum  m n o a u m  )
iteration 5 DL=47.816696 interpretation=(ussibus puerorum  m e o a u m  )
iteration 8 DL=36.863136 interpretation=(ussibus puerorum meorum  )
iteration 25

=> :exit
```

Figure 11. Output from first successful run on the section of 255, using the ink tablet character models.

In this run of the system, 8 iterations took place before the correct answer was generated. (Although the system carried out 25 iterations on this data, the Minimum Description Length generated occurred in iteration 8, and so this is the last data shown. It should also be stressed that, with all of these results, the experiment was run a number of times, and the results presented are the best case, where the system generates the correct result in the fewest iterations.) Previous outputs from iteration 1, 2, 3, and 5, had been possible solutions, but that from iteration 8 proved the best, given the data provided. This was also the correct solution. The GRAVA system successfully reconstructs the correct reading of the text in a short time, on this occasion.

System Performance

Although, above, the correct output was generated in merely 8 iterations, because of the stochastic nature of the process there is a possibility that the correct answer may never be found. If it is generated (in practice the correct output is usually generated within a few iterations) the number of iterations taken to reach this answer will be different on each run. This can be shown by determining the average description length that is generated over a variety of runs on the same data. The example, 255front7,[12] as used above, was run 200 times, with 25 iterations specified in each run. By plotting the average description length gene-

rated from each of the 25 iterations over the 200 runs, it becomes obvious that the system converges on a result. The MDL of the "correct" result in this case was 36.863136, whilst after 25 iterations the system averaged 37.08221. This is due to the fact that the average asymptotically approaches the perfect value because of the Monte Carlo sampling methods utilized (see the previous section "Monte Carlo Select Methods").

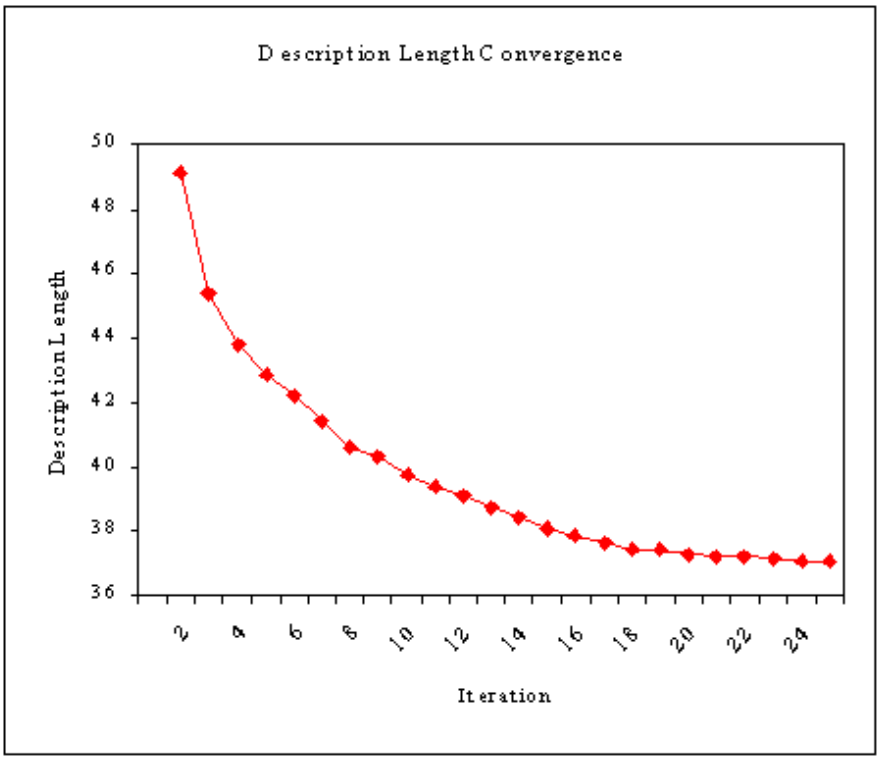


Figure 12. Description Length Convergence Over Iterations

Of course, this example only pertains to the section of 255 used as the test set. The more complex the input data, the longer the system will take to converge on a solution (which will have a different MDL). This example, however, demonstrates that the system is effective at generating

likely solutions within a relatively short time frame: an important consideration when building systems to aid human experts as they cannot be expected to wait for hours until exhaustive searches complete.

Using Ink Models for an Unknown Phrase

A second section of ink tablet was analyzed, this time to see how the system coped with a more confusing section of strokes, and also how it could identify a word that was not in the corpus. The image was annotated USSIBUSS, whilst the eventual, correct, annotation is USSIBUS.

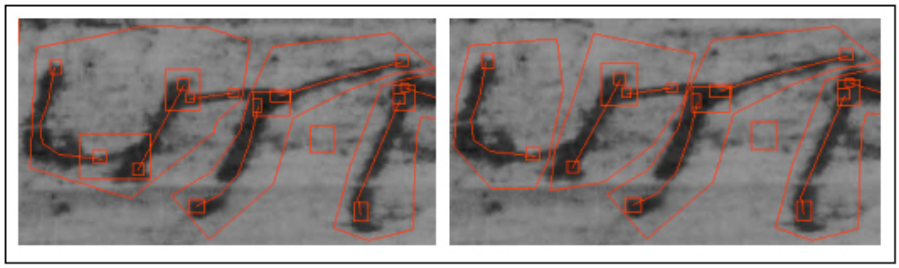


Figure 13. On the left, the section of 255 correctly annotated as US. On the right, the same section annotated incorrectly as USS.

Steps were taken to ensure that the word fragment USSIBUSS did not occur in the word corpus for this iteration, to see how the GRAVA system coped with this difficult input.

```
Grounded Reflective Agent Vision Architecture (GRAVA) Version 2.0.
Yolambda listener pushed. Type :exit to return to GRAVA.

=> ... load the system and the data ...

=>(runCycles 200)
iteration 0 DL=440.220794 interpretation=( ... ((2482 252) (2517 250))) ... )
iteration 1 DL=69.437759 interpretation=(  u r s i b l n s puerorum  m n o a u m  )
iteration 3 DL=69.077354 interpretation=(  u s s i b l n s puerorum  m n o a u m  )
iteration 5 DL=68.062644 interpretation=(  u s s i b l n s puerorum  m e o a u m  )
iteration 6 DL=57.469482 interpretation=(  u r s i b l n s puerorum meorum  )
iteration 18 DL=57.109081 interpretation=(  u s s i b l n s puerorum meorum  )
iteration 25 DL=56.903217 interpretation=(  u s s i b l t s puerorum meorum  )
iteration 199
#f

=>
```

Figure 14. Output from the system, analysis of awkwardly annotated segment of 255, using the ink tablet character models.

These results are interesting on a number of levels. Firstly, the system is confused by the last few characters in USSIBUSS, indicating that it is having problems identifying them. The first problem character is identified (not unreasonably) as an L, the second as either an N or a T. This shows how an unclear character can be assigned a number of possible solutions. Secondly, although the sequence of characters (USSIBUSS) is not in the word corpus, the system does a good job at reconstructing a possible string of characters, resulting in USSIB**S. This is partly due to the use of the character models, and also the use of letter frequencies and bi-graph frequencies. This approximate solution should be enough to give some indication to a human user of what the correct word may be (the system will eventually have to interact with experts in this manner). Finally, the MDL generated from this solution to the problem is 56.903217. The MDL generated from the alternative (correct) annotation of the characters in 5.1 was 36.863143. This shows that the most likely solution to identifying the letters will have the lowest MDL, and also that there is some need, in the future, to encapsulate the opportunity to re-annotate difficult characters in a way that will eventually produce the lowest MDL to generate possible solutions.

Using Ink Models for Stylus Tablets

It was suggested in Terras (2002) that the letter forms from the ink tablets should correspond to those from the stylus tablets closely enough to allow models from the ink tablets to aid in readings of the letters of the stylus tablets. In this experiment, a section of stylus tablet 797 was analyzed, firstly using models derived from the ink tablets, and in the subsequent section, using the small set of models derived from the stylus tablet corpus. This section of tablet contained fairly common words, NUNC QUID (although it had taken the experts a substantial length of time to come up with this reading.)

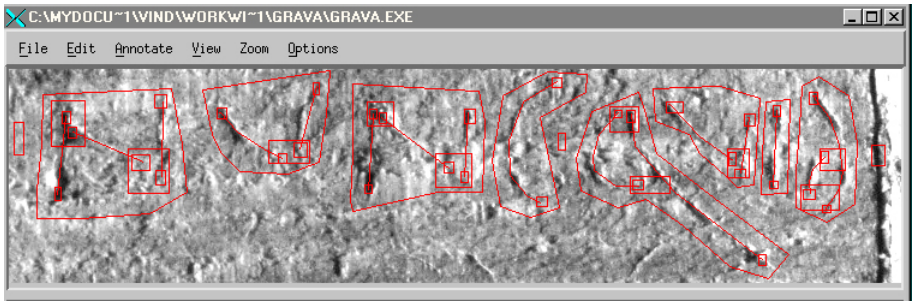


Figure 15. Section of stylus tablet 797, annotations showing NUNC QUID.

Although a fairly small sample of text, it contains a few difficult characters (the curvy letter D, for example, and the large C): the results demonstrating how the system will cope with the difference in character forms between the ink and the stylus texts.

```
Grounded Reflective Agent Vision Architecture (GRAVA) Version 2.0.
Yolambda listener pushed. Type :exit to return to GRAVA.

=> ... load the system and the data ... (Ink Models)

=> (runCycles 200 #t)
iteration 0 DL=34.567626 interpretation=(  n n n c   d u i d  )
iteration 1 DL=31.070133 interpretation=(  n n u i   d u i m  )
iteration 2 DL=31.070131 interpretation=(  n u u i   d n i m  )
iteration 5 DL=22.481197 interpretation=(  u u u i quid  )
iteration 82 DL=21.051862 interpretation=(nunc quid  )
iteration 199
#f

=>
```

Figure 16. Output of GRAVA system, section of 797 utilizing ink character models.

The system took 82 iterations to reach the correct interpretation, quite a high number of run cycles, probably due to the differences in forms between the character sets. The first few iterations, as suspected, show that the system had problems with the letter D, interpreting it as an M, and the large letter C, interpreting it as an I. However, the correct reading was eventually generated when enough run cycles were allowed to sort through the various hypotheses thrown up by the data.

Using Stylus Models for the Stylus Tablets

The same section of tablet 797 was analyzed, this time using the small selection of character models generated from the annotated stylus tablets.

```
Grounded Reflective Agent Vision Architecture (GRAVA) Version 2.0.
Yolambda listener pushed. Type :exit to return to GRAVA.

=> ... load the system and the data ... (Stylus Models)

=> (runCycles 25 #t)
iteration 0 DL=35.304573 interpretation=(  u u n c   q n i d  )
iteration 1 DL=34.447456 interpretation=(  u u u c   q u l m  )
iteration 5 DL=30.377746 interpretation=(nunc   q n i m  )
iteration 12 DL=24.857675 interpretation=(  u u n c quid  )
iteration 16 DL=24.145236 interpretation=(  u u u c quid  )
iteration 18 DL=21.051862 interpretation=(nunc quid  )
iteration 24
#f

=>
```

Figure 17. Output from section of 797, utilizing the stylus character models set.

This run of the system identified the correct response in only 18 iterations, making it much more successful than the run, above, where data from the ink character models were used. There are a number of reasons why this is the case. Firstly, the character models generated from the ink and stylus tablets are *slightly* different, and this small difference must have a large effect on the comparisons. Secondly, although the stylus models character set is impoverished due to the lack of available data, it contains almost all of the characters based in this sample (save for the letter U. The model for the character U was borrowed from the ink tablet models in order to be able to try this test. The letter U seems to be preferred by the recognizer over other characters in this run. This is probably because the model for U borrowed from the ink tablet models was based on a large number of samples whereas the models for the stylus characters were based on a small sample of characters). However, the stylus character model set does not contain all the available characters, which would make it difficult to identify further examples where less common letters predominantly feature. Although comparisons with the stylus character models appear better than the ink character models, there is still a place for the ink character models. Future implementations of the system could have more than one model for each type of letter.

Analyzing Automated Data

It has been shown, above, that the system can correctly interpret annotated images from the corpus which were generated by hand. This is all well and good, but in essence the system needs to work effectively alongside image processing algorithms that have been developed in tandem to this research which can automatically detect possible strokes on the stylus tablets (Bowman, Brady & Tomlin 1997; Terras 2000; Schenk 2001; Molton et al. 2003; Pan et al. 2004; Brady et al. forthcoming). The section of 255 used as test data was analyzed and annotated automatically.

Figure 18. Section of 255 analyzed using the automatic feature extraction algorithms.

This was then used in the system as test data. Due to the image data of the third word being incomprehensibly faint, this was elided from this test. The results are presented below.

```
Grounded Reflective Agent Vision Architecture (GRAVA) Version 2.0.
Yolambda listener pushed. Type :exit to return to GRAVA.

=>... load the system and the data ...

=> (runCycles 200 #t)

DL=326.142322 interpretation=( …  )
iteration 0 DL=24.575424 interpretation=(ussibus puerorum  )
iteration 2 DL=23.990461 interpretation=(ussibus solearum  )
iteration 199
#f

=>
```

Figure 19. Output from an analysis of 255, using automatically annotated images and the character models from the ink tablets.

Although the system initially finds the correct result, it goes onto find a "better" result that was wrong. This is because in the second word it identifies **ER*RUM, where * is an error. Eventually, the Monte Carlo search finds characters that match SOLEARUM well enough to get a lower description length than PUERORUM. However, the system did get the first word 100% correct at the character level, and the second word was almost recovered despite a rather poor interpretation at the character level. This shows promise towards generating possible interpretations of images through utilizing this architecture. Further work needs to be done on the stroke extraction algorithms used in the automated annotation process, to encapsulate stroke data more fluidly, to enable this to be utilized by the character agent. The current algorithms have problems with the grouping of strokes, and identifying the continuation of strokes where the traces of them are faint. When these problems are overcome, and the automated output is in a clearer format, it will be easier to generate possible interpretations of the images using the GRAVA system. Nevertheless, these results show promise towards a possible solution to the problem of how to segue the image processing algorithms into an information based system to aid the papyrologists in the reading of the texts.

Conclusion

This paper has shown how an MDL agent architecture such as GRAVA provides the infrastructure to model the reasoning process the papyrologists go through when reading an ancient document. The system described has an emergent behaviour similar to that of human experts and generates possible, reasonable, interpretations. We have shown that when stroke knowledge is fused with other constraining linguistic knowledge that plausible readings of tablets can be generated.

Because the foundations upon which the architecture is built (MDL and Monte Carlo sampling) are well understood, the behaviour and performance of the architecture can be estimated by means such as convergence analysis. While this application has no time critical aspect, in other applications where time may be an issue, such as real-time vision applications, we can adjust the number of samples to fit the computa-

tional budged. We can in essence trade off interpretation accuracy against time.

The agents in the architecture appear to cooperate, but this cooperation is an emergent property of the architecture. The agents are explicitly *competitive* but the effects of Monte Carlo sampling makes their aggregate behaviour appear to be *cooperative*. This emergent cooperative behaviour is perhaps the most important aspect of the architecture because it provides an implicit information fusion model that depends upon the effect that local decisions have upon the global interpretation. In the case of reading the Vindolanda tablets, information from character strokes is fused with statistical language information including character frequency, character bi-graph frequency, and word frequency knowledge. Additional knowledge such as grammatical knowledge or even subject context knowledge could easily be added, incrementally, to the described system.

It can be objected that our approach is computationally expensive but upon closer examination we find that the comparison with competing techniques is encouraging. Most AI architectures, as we discussed above, search for the first solution that could be found in a depth first manner. Our goal is to find approximations to the *best* solution so comparison with those architectures is not appropriate. Speech Recognition and statistical NLP often use variants of hidden Markov models to find the best solution. These methods are comparable and are faster than the Monte Carlo approach advocated in this paper but they cannot easily be adapted to work with the disparate kinds of knowledge that GRAVA can handle. The extra price we pay for the Monte Carlo implementation more than pays for itself by providing a flexible knowledge fusion model.

Although this research did not deliver a stand-alone application for the papyrologists to use to aid in reading the stylus tablets, this was not the primary aim of the project. It has provided an understanding of the type of tools required by the experts, as well as implementing a system that can analyse image data and propagate useful interpretations. Further testing and development is necessary before a completed application can be made available, at present the system requires manual input from the engineering scientists to output results, but the findings presented here provide the basis for the construction of a standalone system: a fruitful

culmination of varied, interdisciplinary research. Research continues in the Department of Engineering Science and the Centre for the Study of Ancient Documents at the University of Oxford, to see how the system can be developed to aid in the propagation of readings of not only the stylus tablets, but other types of ancient document, such as Greek inscriptions.

The research presented here presents many opportunities for future work. From a humanities angle, this type of computer tool could prove to be instrumental in reading various types of documents that were illegible to the human eye: the joining of image processing and linguistic information allowing many possible interpretations of data to be generated to aid experts in their task. It would not be difficult to adapt this system to other linguistic systems given that the necessary statistics were available, and the primary sources made available for digitization. Just how useful such a tool actually is to those who read such primary sources remains to be seen, but papyrology as a field has so far embraced computer based tools and resources rapidly.

MDL based architectures could be used on any number of image processing tasks, where complex information from other semantic levels needs to be used to interpret images. It has been shown here that MDL provides the common currency to relate different types of information, and this could be investigated further. An architecture such as the one described in this paper could be used to read other types of handwriting, but the architecture could be expanded much further, to incorporate other semantic levels, such as grammar and contextual information. MDL architectures could also be used for entirely different image interpretation problems, such as facial recognition, sign language interpretation, or medical image analysis, as long as procedural information from different semantic levels was available or obtainable, to allow complex hierarchical systems to be implemented. So far as to say, MDL architectures could provide the basis for the development of any type of computer based interpretation system, for example: speech recognition (or production), the analysis and interpretation of physical processes (the monitoring of weather, water flow and direction, and the many indicative signs of volcanic activity), predicting the outcome in war gaming systems, etc. The scope for the appropriation and development

of this type of architecture is almost limitless: the important point being it provides a way of comparing and contrasting semantically different types of information fairly and efficiently to generate the best probable outcome from available data. It is an interesting aside that this research, aimed at solving a Humanities based problem, has aided in the development of novel techniques in computing and engineering science: a relatively rare occurrence.

This paper has presented a novel approach to a complex problem, delivering a system that can generate plausible interpretations from images, in the same way that human experts appear to do, to aid them in their task. It effectively draws together disparate research in image processing, ancient history, and artificial intelligence to demonstrate a complete signal to symbol system. In doing so, this research offers further opportunities to develop intelligent systems that can interpret image data effectively, to aid human beings in complex perceptual tasks.

Acknowledgements

The authors would like to thank Professor Sir Michael Brady and Professor Alan K. Bowman for their guidance on this research. Dr Xiao-Bo Pan aided in the preparation of images.

Melissa Terras is a lecturer in Electronic Communication in the School of Library, Archive, and Information Studies, University College London. She has an MA in History of Art and English Literature, an MSc in IT (Software and Systems), both from the University of Glasgow, and a D.Phil in Engineering Science from the University of Oxford. Her research interests include Ancient Documents, Artificial Intelligence, Image Processing, Virtual Reality, and Advanced Internet Technologies.
E-mail: m.terras@ucl.ac.uk
Web page: http://www.ucl.ac.uk/slais/melissa-terras/

Paul Robertson is a Research Scientist at Massachusetts Institute of Technology's Computing Science and Artificial Intelligence Laboratory (CSAIL). He holds a BA in Computer Science from the University of Essex and a D.Phil in Engineering Science from the University of Oxford. His

research interests include Artificial Intelligence, Self-Adaptive Software, Computer Vision and Advanced Computer Languages.
E-mail: paulr@csail.mit.edu
Web page: http://people.csail.mit.edu/~paulr

Notes

1. This paper is a version of Robertson et al (2005) which has been written with Humanities based computing scholars in mind, as opposed to the original paper which is much more mathematical in nature and serves to explain in depth the technical aspects of the system to the Artificial Intelligence and Image Processing community. Given the two different audiences of the respective journals, it was felt appropriate to submit papers regarding the research project to both, as the focus, scope, and interest of readers are diverse, and are unlikely to have much intersection.
2. <http://www.eng.ox.ac.uk/>
3. <http://www.csad.ox.ac.uk/>
4. Further examples of the tablets, and contextual information regarding Vindolanda, can be found at the Vindolanda Tablets Online site <http://vindolanda.csad.ox.ac.uk/>.
5. It is suspected that around 2000 of such tablets exist outside Egypt (Renner 1992).
6. Only one stylus tablet, 836, has been found so far with its wax intact. Unfortunately this deteriorated during conservation, but a photographic record of the waxed tablet remains to compare the visible text with that on the re-used tablet.
7. These techniques were developed with the permission and guidance of the British Museum – where the tablets are now housed. As the documents are fragmentary and fragile, it was important to use non-invasive techniques which would not cause them to deteriorate any further.
8. For example using the Visual C++ plugin with PhotoShop.
9. Aalto (1945), Youtie (1963), Youtie (1966), and Bowman & Tomlin (forthcoming) are the only discussions found (as yet) which try to describe what the papyrology process actually entails, with some higher level discussion available in Turner (1968).
10. The basic concept behind MDL is an operational form of Occam's razor, "Pluralitas non est ponenda sine necessitate" or "plurality should not be posited without necessity." Formulated by the medieval English philosopher and Franciscan monk William of Ockham (ca. 1285-1349), the phrase has been adopted by Communication Theorists to suggest that one should not increase, beyond what is necessary, the number of entities required to explain anything (Forster 1999). The shorter an explanation, the better.

11. Meaning "for the use of my boys", referring to some cloaks and tunics for Clodius Super's *pueri* (Bowman and Thomas 1994).
12. The example 255front7 is the seventh line from the image of the front of tablet 255. Some tablets have writing on both sides, and are routinely labelled "front" and "back". Because of the size of image files used to capture the data, and the demands on computer memory presented by manipulation of these, the files were split into individual lines of text for annotation. This also gave easily usable small test sets of data.

References

AALTO, PENTII (1945). "Notes on Methods of Decipherment of Unknown Writings and Languages." *Studia Orientalia, Edidat Societas Orientalis Fennica* XI.4.

BAUM, LEONARD E. (1972). "An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process." *Inequalities* 3: 1-8.

BEARMAN, GREGORY H. & SHEILA I. SPIRO (1996). "Archaeological Applications of Advanced Imaging Techniques." *Biblical Archaeologist* 59.1: 56-66.

BOWMAN, ALAN K. (1997). "The Vindolanda Writing Tablets." *XI Congresso Internazionale di Epigrafia Greca e Latina*. Roma.

BOWMAN, ALAN K., MICHAEL J. BRADY & ROGER S. O. TOMLIN (1997). "Imaging Incised Documents." *Literary and Linguistic Computing* 12.3: 169- 176.

BOWMAN, ALAN K. & J. DAVID THOMAS (1983). *Vindolanda: The Latin Writing Tablets*. London: Society for Promotion of Roman Studies.

BOWMAN, ALAN K. & J. DAVID THOMAS (1994). *The Vindolanda Writing-Tablets*. (Tabulae Vindolandenses, II). London: British Museum Press.

BOWMAN, ALAN K. & J. DAVID THOMAS (2003). *The Vindolanda Writing-Tablets.* (Tabulae Vindolandenses, III). London: British Museum Press.

BOWMAN, ALAN K. & ROGER S. O. TOMLIN (forthcoming 2005). "Wooden Stylus Tablets from Roman Britain." *Images and Artefacts of the Ancient World*. Eds. Alan K. Bowman & Michael Brady. Oxford: Oxford University Press.

BRADY, MICHAEL, XIAO-PO PAN, MELISSA TERRAS & VEIT SCHENK. (forthcoming 2005). "Shadow Stereo, Image Filtering and Constraint Propagation." *Images and Artefacts of the Ancient World*. Eds. Alan K. Bowman & Michael Brady. Oxford: Oxford University Press.

BROOKS, RODNEY A. (1986). "A Robust Layered Control System for a Mobile Robot." *IEEE Journal of Robotics and Automation* 2: 14-23.

CHARNIAK, EUGENE (1993). *Statistical Language Learning*. Cambridge, MA: MIT Press.

CONNELL, JONATHAN H. & MICHAEL BRADY (1987). "Generating and Generalizing Models of Visual Objects." *Artificial Intelligence* 31.2: 159-183.

CÔTÉ, MYRIAM ET AL. (1998). "Automatic Reading Of Cursive Scripts Using A Reading Model And Perceptual Concepts: The PERCEPTO System." *International Journal of Document Analysis and Recognition* 1.1: 3-17.

DODEL, JEAN-PIERRE & RAJJAN SHINGHAL (1995). "Symbolic/ Neural Recognition of Cursive Amounts on Bank Cheques." *Proceedings of the Third International Conference on Document Analysis and Recognition*. Vol. 1. Los Alamitos, CA: IEEE Computer Society Press. 15-18.

ERMAN, LEE D. ET AL. (1980). "The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty." *Computing Surveys* 12.2: 213-253.

EYSENCK, MICHAEL W. & MARK T. KEANE (1997). *Cognitive Psychology, A Student's Handbook*. Hove, Psychology Press.

FAHLMAN, SCOTT E. (1973). *A Planning System for Robot Construction Tasks*. (AI Technical Reports, 283), Cambridge: MA: MIT AI Lab.

FINK, ROBERT O. (1971). *Roman Military Records on Papyrus*. Cleveland, OH: Case Western Reserve University.

FORSTER, MALCOLM R. (1999). "The New Science of Simplicity." *Simplicity, Inference and Econometric Modelling*. Eds. Hugo A. Keuzenkamp, Michael McAleer & Arnold Zellner . Cambridge: Cambridge University Press. 83-119.

FU, KING SUN & PHILIP H. SWAIN (1969). "On Syntactic Pattern Recognition." *Software Engineering* 2: 155-182.

GEMAN, STUART & DONALD GEMAN (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6: 721-741.

HAMMERSLEY, JOHN M. & DAVID C. HANDSCOMB (1964). *Monte Carlo Methods*. London: Chapman and Hall.

IMPEDOVO, SEBASTIANO (1993). *Fundamentals in Handwriting Recognition*. (NATO Advanced Study Workshop on Fundamentals in Handwriting Recognition). Château de Bonas: Springer.

KARSSEMEIJER, NICO (1992). "Stochastic Model for Automated Detection of Calcifications in Digital Mammograms." *Image and Vision Computing* 10.6: 369-375.

LAU, T. W. E. & Y. C. HO (1997). "Universal Alignment Probabilities and Subset Selection for Ordinal Optimization." *Journal of Optimization Theory and Applications* 93: 455-489.

MCCLELLAND JAMES L. & DAVID E. RUMELHART (1986). *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*. 2 vols. Cambridge. MA: MIT Press.

MCCLELLAND JAMES L. & DAVID E. RUMELHART (1986b). "The Programmable Blackboard Model of Reading." *Parallel distributed processing : Explorations in the Microstructure of Cognition. Vol. 2, Psychological and biological Models*. Cambridge. MA: MIT Press. 122-169.

MCDERMOTT DREW V. & GERALD L. SUSSMAN (1973). *The Conniver Reference Manual*. (AI Memo 259). Cambridge. MA: MIT AI Lab.

MCGRAW, KAREN L. & KAREN HARBISON-BRIGGS (1989). *Knowledge Acquisition: Principles and Guidelines*. London: Prentice-Hall.

MOLTON, NICK ET AL. (2003). "Visual Enhancement of Incised Text." *Pattern Recognition* 36: 1031-1043.

PAN, XIAO-BO ET AL. (2004). "Enhancement and Feature Extraction for Images of Incised and Ink Texts." *Image and Vision Computing*. 22.6: 443-451.

PARISSE, CHRISTOPHE (1996). "Global Word Shape Processing in Off-Line Recognition of Handwriting." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.4: 460-464.

RENNER, TIMOTHY (1992). "The Finds of Wooden Tablets from Campania and Dacia as Parallels to Archives of Documentary Papyri from Roman Egypt." *Copenhagen Congress paper*.

RISSANEN, JORMA (1978). "Modelling by Shortest Data Description." *Automatica* 14: 465-471.

ROBERTSON, PAUL & ROBERT LADDAGA (2003). "An Agent Architecture for Information Fusion and its Application to Robust Face Identification." *Proceedings of the 21st International Conference on Applied Informatics*, Innsbruck, Austria.

ROBERTSON, PAUL (2001). *A Self-Adaptive Architecture for Image Understanding*. Diss. Oxford: Department of Engineering Science, University of Oxford.

ROBERTSON, PAUL ET AL. (forthcoming 2005). "Image to Interpretation: An MDL Agent Architecture to Read Ancient Roman Texts." Submitted.

SCHENK, VEIT U. B. (2001). *Visual Identification of Fine Surface Incisions*. Diss. Oxford: Department of Engineering Science, University of Oxford.

SCHENK, VEIT U. B & MICHAEL BRADY (2003). "Visual Identification of Fine Surface Incisions in Incised Roman Stylus Tablets." Paper presented at *ICAPR 2003: International Conference in Advances in Pattern Recognition, Calcutta, December 10-13 2003*.

SHANNON, CLAUDE E. (1949). "Communicating in the Presence of Noise." *Proceedings of the Intsitute of Radio Engineers* 37: 10-21.

TERRAS, MELISSA (2000). "Towards a Reading of the Vindolanda Stylus Tablets: Engineering Science and the Papyrologist." *Human IT* 4.2: 255-272.

TERRAS, MELISSA (2002). "Image to Interpretation: Towards an Intelligent System to Aid Historians in the Reading of the Vindolanda Texts." Diss. Oxford: Department of Engineering Science, University of Oxford.

TERRAS, MELISSA & PAUL ROBERTSON (2004). "Downs and Acrosses, Textual Markup on a Stroke Based Level." *Literary and Linguistic Computing* 19.3: 397-414.

TURNER, ERIC G. (1968). *Greek Papyri: An Introduction*. Oxford, Clarendon Press.

VITERBI ANDREW J. (1967). "Error Bounds for Convolution Codes and an Asymptotically Optimal Decoding Algorithm." *IEEE Transactions on Information Theory* 13: 260-269.

WALLACE, CHRISTOPHER S. & DAVID M. BOULTON (1968). "An Information Measure for Classification." *Computer Journal* 11: 185-195.

WATERMAN, DONALD A. (1986). *A Guide to Expert Systems*. Reading, MA: Addison-Wesley.

YOUTIE, HERBERT C. (1963). "The Papyrologist: Artificer of Fact." *Greek Roman and Byzantine Studies* 4: 19-32.

YOUTIE, HERBERT C. (1966). "Text and Context in Transcribing Papyri." *Greek Roman and Byzantine Studies* 7: 251-8.