

ABSTRACT

STANLEY, JEFFREY CHRISTOPHER. To Read Images Not Words: Computer-Aided Analysis of the Handwriting in the *Codex Seraphinianus*. (Under the direction of Robert Rodman.)

The purpose of this study has been to present a new method of investigating an undeciphered writing system using a computer, by analyzing the written images as images instead of relying on a transcription scheme to map written characters to numbers. The computer can extract and organize hundreds of written images more quickly and reliably than a human can, while proper human supervision can turn this data into insights about the writing system, avoiding the problems potentially introduced by the intermediate layer of a transcription scheme. The study implements several applications that demonstrate this principle, using the *Codex Seraphinianus* as a corpus, including a type classifier and a search engine. The implementations are able to identify recurring sequences of tokens in the corpus and to propose a classification of tokens into types, two main sub-problems in decipherment. Specifically, the header tokens in the first three chapters are classified into 53 types. Examining the results yields findings that support Serafini's recent statements that the writing is artistic, not linguistic. The automatic nature of the writing is briefly examined in light of the findings, and future directions are encouraged.

To Read Images Not Words: Computer-Aided Analysis
of the Handwriting in the *Codex Seraphinianus*

by
Jeffrey Christopher Stanley

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2010

APPROVED BY:

Dr. James Lester

Dr. Jon Doyle

Dr. Robert Rodman
Chair of Advisory Committee

BIOGRAPHY

Jeff Stanley graduated from Duke University in 2003 with high honors. His undergraduate degrees are in Linguistics and Computer Science, with a minor in Ancient Greek. His honors thesis in Linguistics, *Tongues of Malevolence: A Linguistic Analysis of Constructed Fictional Languages with Emphasis on Languages Constructed for "The Other"*, is a multilayered linguistic analysis of made-up languages in movies, books, and television. He also participated in the development of the Speech Accent Archive, an undertaking at George Mason University concerned with documenting online the phonological inventories and speech qualities of the world's languages and speakers.

Since graduation, Jeff has worked as a software engineer at the Supreme Court of the United States, the Research Triangle Institute, and two video game companies. He remains a student of language and culture, with special interests in:

- undeciphered languages
- ancient languages
- made-up languages
- writing systems
- phonology and sound symbolism
- language universals
- the development of the oldest languages and their role in early cultures

ACKNOWLEDGMENTS

I would like to thank my advisor, Robert Rodman, for bringing the *Codex Seraphinianus* to my attention and for being willing to advise me on the topic of mysterious languages even though it is not an area of study for him or for the department. I would also like to extend thanks to Enrico Prodi, who gave me his personal notes from Serafini's talk to the Oxford Bibliophiles, and to the others in that society who were so helpful.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES.....	vii
DEFINITION OF TERMS.....	1
A NOTE ON APPROACH.....	1
INTRODUCTION.....	1
Decipherment and Transcription Schemes.....	1
Problems with Transcription Schemes.....	2
Computers and Transcription.....	2
The Role of Computers in Decipherment.....	4
A Problem of Computer-Aided Decipherment.....	4
Proposal to Solve the Problem.....	4
The Codex Seraphinianus: A Mysterious Writing System.....	6
Asemic Writing.....	8
Automatic Writing.....	9
Statement of Thesis.....	11
Primary Thesis.....	11
Secondary Aims.....	12
HISTORY OF SCHOLARSHIP.....	13
Computer-Aided Analysis of Undeciphered Writing Systems.....	13
Indus Script.....	13
Rongo-Rongo.....	15
Linear A.....	17
Developing Methodology Using a Known Language.....	18
Computer Recognition of Written Images.....	21
Printed Character Recognition.....	21
Handwriting Recognition.....	23
Epigraphy.....	24
Automatic Handwriting.....	24
Glossolalia and Speaking in Tongues.....	26
Other Related Processes.....	27
Summary.....	27
MATERIALS.....	28
Hardware and Operating System.....	28
Development Environment.....	28
Third-Party Software.....	28
Corpus.....	29
METHODS AND RESULTS.....	30
Header Text (“Capital Letters”).....	30
Preparation for Extraction.....	30

Extraction	31
Identification of Similar Tokens	32
Study 1: Consolidation into Types	34
Study 2: Search Engine	41
Body Text (“Lower Case Letters”)	48
Preparation for Extraction	48
Extraction	48
Study 1: Words Ordered by Length	50
Study 2: Search Engine	53
DISCUSSION OF RESULTS	58
Linguistic Observations on the Writing	58
Formulaic Headers	58
Free Variation within a Type	59
Non-repetition of Words	61
Summary	63
Methodological Observations on the Studies	63
Successes	63
Problems	64
CONCLUSION	65
Computer-assisted Decipherment	65
The Codex Seraphinianus	65
Areas for Further Study	66
REFERENCES	68

LIST OF TABLES

Table 1: Patterns in the codex headers.....	37
Table 2: Types found in chapters 1-3, with suggested names and transcriptions.....	37
Table 3: Words or parts of words made of common types in the headers which are nevertheless unique.....	47
Table 4: Table showing search engine results for a target word, pages 10-65.....	56
Table 5: Kinds of variation in three of the header types.....	60

LIST OF FIGURES

Figure 1: The Codex Seraphinianus.....	6
Figure 2: The Indus script.....	13
Figure 3: The Rongo-rongo script.....	15
Figure 4: Linear A.....	17
Figure 5: Mlle. Smith’s Martian graphemes as collected by Flournoy.....	25
Figure 6: Comparison of header tokens.....	33
Figure 7: Applying a whitespace matrix to the last example from the previous figure....	34
Figure 8: Typing app showing results for the first three chapters of the codex.....	36
Figure 9: Search results for the common sequence SFA.....	43
Figure 10: As above but with Max Dist set to 1.....	44
Figure 11: Example of an illustration page (13).....	46
Figure 12: Unique pre-dash tokens appearing on illustration pages in chapter one dealing with similar plants.....	46
Figure 13: Samples from page 12 of the codex showing text mixed with (a) diagrams, (b) tables, (c) two-line compartments, and (d) non-word strokes.....	49
Figure 14: An application that displays all word instances on a page sorted by length, applied to page 13 of the codex.....	51
Figure 15: Sorting the word instances on page 13 by length.....	52
Figure 16: Near the top of the word results are extraction errors.....	53
Figure 17: Demonstration of search engine for body text and close-up (bottom) of best matches, pages 10-48.....	54

Figure 18: Two tokens near the top of page 52 that are similar but are not detected by the search algorithm because they vary in length.	57
Figure 19: Tokens in the codex headers that appear to differ only by a loop near an endpoint.....	60
Figure 20: Ornamented and unornamented S tokens near each other on the same page. .	61
Figure 21: Four possible sentences from page 52.....	62

DEFINITION OF TERMS

In this paper, a symbol when it can be identified as a reusable and consistent unit (as the letter A in the Latin alphabet) will be called a type, while an individual instance appearing in a written sample will be called a token. These terms are roughly equivalent to the idea of a grapheme and an allograph.

A NOTE ON APPROACH

It can be assumed with confidence at this point, to take Serafini at his word, that the writing in the *Codex Seraphinianus* does not encode a language. (This will be discussed below.) However, in methodology this study will treat the writing as though the question is still unresolved, in order to remain relevant to the real-world undeciphered writings that pose this issue, such as the Indus and Rongo scripts. On the other hand, when discussing the codex specifically, the study will consider its asemy.

INTRODUCTION

Decipherment and Transcription Schemes

The world abounds with writing systems left as mysteries for the scholars that come after, writing systems representing lost cultures like Linear A and B from Crete, Rongo-Rongo from Easter Island, the Egyptian and Mayan hieroglyphs; and writing systems that were designed as mysteries from conception like that in the Voynich Manuscript. Computer studies on undeciphered writing have grown in popularity, evidenced for example by recent debates about the Indus script and Pictish stones (Rao 2009a; 2009b; Lee 2010; Sproat 2010) and by an apparent success at MIT (Snyder 2010; Hardesty 2010). All of these cases and others will be visited in this paper. To organize these writing systems with characters so unlike any known script, scholars create transcription schemes that map each apparently unique character to a number.

Scholars can then easily keep track of these characters and share their ideas about the writing without describing or drawing the actual features of the writing.

Problems with Transcription Schemes

A transcription scheme, while useful, introduces a level of indirection between the writing and scholarship. When the tokens to be documented fall cleanly into a manageable set of types, like the twenty-six letters of the alphabet used for English, the transcription scheme provides a fast and easy way to represent them. In other cases, because it is created by people who do not yet understand what they are trying to represent, this intermediate layer can actually hinder research by providing an incorrect or limiting framework for studying the original script. In some cases the transcription scheme becomes a constant subject of debate, taking time away from the study of the original script. The transcription scheme for Rongo-Rongo, for example, is fraught with problems because of the fluid nature of the characters, which appear to morph and combine in ways that are not understood, hardly compatible with the rigid nature of a numeric mapping. Finally, sometimes transcriptions are made incorrectly, but these become widely used instead of the original text, because the original text is not widely available in good quality or simply because it is easier to manage, leading to misunderstandings that affect any scholars working with the transcribed material (cf. Pozdniakov 2007, Guy 2005, Sproat 2003).

Computers and Transcription

Unicode has emerged as the standard for representing the world's scripts on the computer. In the past, scholars had to create their own fonts or find special-case fonts for the scripts they wanted to study, resulting in many small, isolated, non-standardized fonts. Unicode provides a single standard for representing a hundred

different scripts, including the Latin alphabet, Devanagari, Runic, and even Linear B, an ancient script of Crete that is only partially understood (Allen, 2009: 181).

Basically, Unicode provides a standard for representing the characters, but the end user must have a font capable of displaying the characters according to these standards. It is the presence of the character linguistically, not the visual qualities of the original writing, that is important to applications that benefit from Unicode.

To illustrate this, consider Sanskrit (the Devanagari script), which is apparently composed of a finite set of distinct alphasyllabic types, but it is made graphically more complex by the connections between them. The vowels of Sanskrit are attached to the consonant they follow. When two or more consonants in Sanskrit are placed in sequence, they can combine to form a new graphical representation. Doubling of a character also modifies the representation (Coulson 1992: 4-19). These are only a few examples. However, none of these considerations present special problems in computerized transcription. This is because it is the content that matters, not the actual graphical tokens that were written in some original manuscript. The software that interprets the transcription decides how to display it. Indeed, some fonts capable of displaying the script are able to display combined consonants (called ligatures) while others cannot (Chopde 2001).

Linear B can be represented by Unicode today because the signs of Linear B are distinct and the types are identifiable; it is no longer vital to communicate the original tokens in all their graphical variation in the Linear B corpus. This is not feasible for an unknown writing system. Only the graphical surface is observable. The underlying structure is not known. This particularly applies to a writing system that does not easily break into a small subset of distinct tokens.

The Role of Computers in Decipherment

Recently, computers are playing a significant role in the study of mysterious writing systems. Many scholars have approached the Indus script with computers; most recently Rao et al. performed extensive analysis of the Indus script to show that it shares some features with known languages (Rao 2009a). Sproat (2003) used a computer to identify parallel sequences of tokens in the Rongo-Rongo corpus. See the section on previous scholarship for details on these studies. Corpora for undeciphered and semi-deciphered scripts are appearing online both in photographic and transcribed form, ready for computer analysis (Guy 2005; Younger 2010), so it may be expected that computers will continue to grow in this role. Furthermore, it can be expected that not all languages capable of being discovered have been yet, as a recently found Olmec script demonstrates (Rodríguez Martínez 2006).

A Problem of Computer-Aided Decipherment

With the advance of the computer in this field, the imperfections of the transcription scheme are potentially exploded many times through the myriad calculations made possible by technology, with no human supervisor; or if there is a human supervisor, the effort of checking each calculation is laborious. This may well explain why computers have not been embraced by more researchers.

Proposal to Solve the Problem

Considering the respect that computers have in digital imaging, there may be an obvious solution to this problem: Remove the transcription scheme. Optical character recognition and handwriting recognition are enjoying unprecedented accuracy, with archaeologists able to teach a computer to recognize ancient Roman inscriptions

unreadable by humans. See the section on previous scholarship for details on these studies. This same technology should be applied to advance computers the next step in decipherment, leading to systems that are not more prone to error but far less prone to error than traditional systems. In summary, transcription schemes for unknown writings are problematic, but computers are fully capable of organizing written images without a transcription scheme, while a human would have to resort to a transcription scheme to keep track of them all.

The Codex Seraphinianus: A Mysterious Writing System

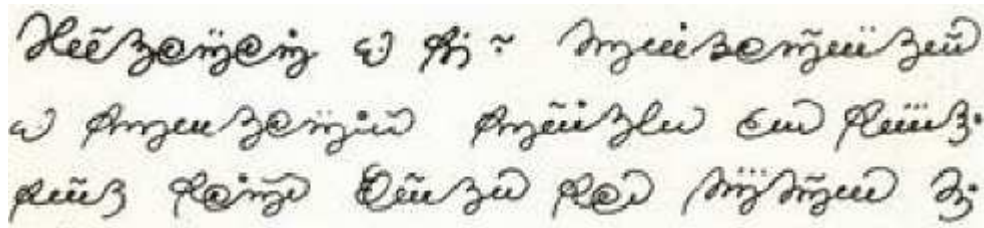


Figure 1: The Codex Seraphinianus.
 Top: pages 12 and 13. Bottom: detail page 12.

Created in the 1970's by Luigi Serafini, an Italian architect, the *Codex Seraphinianus* has been fancied by some a guide to another world. It bears over 300 pages of fantastic color illustrations and handwriting not resembling any known writing. The pictures depict all kinds of strange things, from a flower that can be blown up into a balloon, to trees that uproot themselves and swim upstream to spawn, to strange vehicles, even

strange physical and chemical reactions. On the cover of one popular edition, a copulating man and woman morph into an alligator.

The *Codex* is organized into eleven chapters. At the beginning of each chapter is a table of contents page. There are two types of content pages, one consisting mainly of text sometimes with small images or diagrams and one consisting mainly of illustrations with captions, which this paper will refer to as text pages and illustration pages respectively. At the top of every page is a header written in the mysterious script. This header text is made of large disconnected tokens that are different from the connected, cursive-like text found in the body of the pages. However, some of the header tokens resemble some of the body tokens, especially the initial body tokens, leading people to refer to them as miniscule and majuscule letters (Derzhanski 2004).

Wechsler (1987) and Derzhanski (2004) independently deciphered the writing system used to number pages in the *Codex*, but studies of the header and body text remain speculative, with only a few linguistic observations possible, such as the classification into the capital and lower-case letters.

The writing in the *Codex* has been chosen for this study for several reasons, as a compromise between raw undeciphered artifacts and familiar, accessible written formats.

1. The symbols in the codex are not easily classified and appear to be formed in a composite manner reminiscent of Rongo-Rongo and some of the symbols in the Indus script and Linear A.
2. The *Codex* mixes text with illustrations and presents text of varying sizes and juxtapositions, so it provides a challenge in the vein of raw source material.

3. The text in the *Codex* is all compiled together in an organized format so that it can be referred to in a familiar way, by page number, chapter, and location on the page. This allows focus on the thesis without excessive discussion of the corpus.
4. The *Codex* can be obtained as a compilation of high-resolution color electronic scans of the pages. Therefore it allows study of the original source material in a way that is organized, coherent and ready for computer analysis.

At the time of starting this research, the *Codex* presented a good example of a data set because of its organization, its length, and because, since its author is tight-lipped but still alive, there was a non-zero chance of one day verifying the results.

Asemic Writing

Expectations for this study changed in May 2009, when Serafini announced clearly at a talk to the Oxford Bibliophiles that the writing in the *Codex Seraphinianus* does not represent any language, known or secret. It is asemic. Rather than an undeciphered writing system, the study's focus became an asemic writing system. This development actually revealed new reasons for using the *Codex* as a corpus.

1. By widening its range of interest, it has the potential to demonstrate the technology's value in areas other than linguistics: art or psychology for instance.
2. Because the writing encodes no language, the *Codex* effectively embodies a worst case scenario. If the study can organize and analyze this pseudo-language, it can be expected to work for writings that are known to encode languages, as well as disputed writing systems like Rongo-Rongo and the Indus Script.

Recently, asemic writing has gained a small following as an avant-garde art form, but it is still an esoteric phenomenon. Tim Gaze, editor of *Asemic Magazine*, stated in an interview that asemic writing is “something which looks like a form of writing, but

which you can't read" (Alexander 2009(1)). Technically, no one can read true asemic writing, because it does not represent a language. However, while it cannot be read in a traditional sense, it can provide a similar experience: ". . . I'm trying to create things which are totally open in meaning, suggestive to a viewer, but without a privileged meaning" (Alexander 2009(1)). Another leader in the field, Michael Jacobson, known for creating an 89-page asemic work called *The Giant's Fence*, similarly calls it "a code that is open to interpretation, with no fixed meaning." He explains that one emerging poetic movement focuses on breaking writing down and that asemic writing breaks it down into the rudimentary features of texture and line. Jacobson believes that asemic writing could draw people back to literacy by contributing to a "multi-media" experience (Alexander 2009(2)).

To the Oxford Bibliophiles, Serafini expressed this sentiment. Since his talk was not recorded or published, the following is from notes by an attendee.

The book creates a feeling of illiteracy which, in turn, encourages imagination, like children seeing a book: they cannot yet read it, but they realise that it must make sense (and that it does in fact make sense to grown-ups) and imagine what its meaning must be . . . The writing of the Codex is a writing, not a language, although it conveys the impression of being one. It looks like it means something, but it does not; it is free from the cage of a language and a syntax. It involves a visual process, not a linguistic process (Prodi 2009).

Automatic Writing

Additionally, Serafini stated that the experience of composing the writing in the Codex was similar to automatic writing (Prodi 2009). To create over 300 pages of purely artistic text carefully deliberated to appear to encode a language seems insurmountable, so it makes sense that the task would need to be or to become automated; but what does Serafini really mean?

Of *The Giant's Fence*, Jacobson commented: “The main difficulty with writing [*The Giant's Fence*] though, was to keep the style consistent over the 2 years it took to write the book. I don't think I have the ability or the stamina to recreate a work like it” (Alexander 2009a). The Codex Seraphinianus is about four times as large (though the comparison is an inexact one), and was finished in a little over two years (Manetto 2007). In a 2007 phone interview for *El País*, Serafini described the Codex as water that gushed out of him. He made several claims about the language, implying not only that it was primarily artistic but also that “I realized I was leaving the pencil alone . . . I made it up suddenly. It is a vision, a dream language. The mystery, for me, is simply in the artistic act” (Manetto 2007, translated). It may be accepted, then, that he was able to manage such a work because composing the writing, for him, was some sort of automated task, requiring little or no conscious effort. The idea of writing resulting from a vision or dream is not new and will be addressed below.

The section on previous scholarship provides a discussion of scientific research on automatic writing and a related phenomenon, glossolalia or speaking in tongues. At this point it will be enough to cover its non-scientific side. Automatic writing enjoys a history of mysticism. Often it is claimed by a spiritualist as a gift, as a communication or inspiration from some other entity. A famous case is that of H el ene Smith, known as the “Muse of Automatic Writing”, who would write in what she claimed was a Martian language and then translate it to French. She would also speak and create art automatically (Rosenberg 2000). William Stainton Moses, a leader in the Church of England, became fascinated with automatic writing after experiencing it himself. In his book on the subject he gives many first-hand, corroborated accounts as evidence for an “Intelligence outside of a human brain,” including cases in which the language was unknown to the psychic and in which writing occurred apparently without any human intervention (Moses 1882). Two religious texts that claim to have been written

automatically (channeled by spirit guides) are the *Oahspe* by Newbrough and *The Aquarian Gospel of Jesus the Christ* by Dowling. *A Dweller on Two Planets* by Oliver is only one of the many 19th century books supposedly written by former inhabitants of Atlantis seeking to impart knowledge of their destroyed civilization through the pens of the living.

However, there are also less lofty manifestations of automatic writing. Along with other unmediated processes, some Freudian psychologists use automatic writing to reveal the unconscious. It was first proposed in 1878 that what is ascribed to spiritual beings is really another self, dissociated. In a seminal case, Pierre Janet realized that, while his patient Lucie could not feel any sensation on her body, a personality that would write through her automatically when she was distracted, signing “Adrienne,” felt everything (Van der Hart 1989).

The third movement in which automatic writing finds its home is an artistic one. In his Surrealist Manifesto, André Breton emphasizes all forms of automatism, especially automatic writing, as the “. . . true functioning of thought . . . in the absence of all control by reason . . .” (Breton 1972). Other well-known surrealists such as Alexander Brotchie and Benjamin Péret practice and recommend writing in the absence of controlled thought as a valuable exercise (Brotchie 2003). Today “free writing” is a popular writing exercise not just among surrealists.

Statement of Thesis

Primary Thesis

By taking advantage of a computer’s ability to deal directly with images, simple and effective applications can be made for answering questions that linguists ask about

undeciphered writing systems without the potential problems of a transcription scheme.

Secondary Aims

In this case, the applications produced may shed light on the writing system of the *Codex Seraphinianus*, leading to new insights about the nature of the writing and its relation to known writing systems. In turn, this could shed light on the nature of asemic and automatic writing in general.

HISTORY OF SCHOLARSHIP

Computer-Aided Analysis of Undeciphered Writing Systems

Indus Script

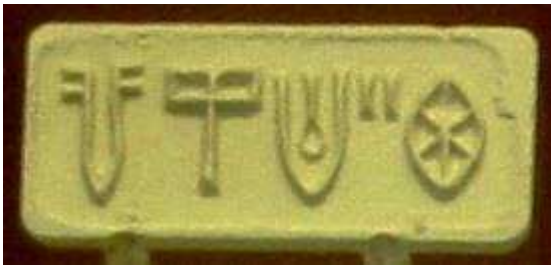


Figure 2: The Indus script.
Seal impression from the British Museum.

Of all the undeciphered languages, the Indus script (also known as the Harappan script) has received the most computer-aided attention. Long-lasting attempts have been made to use computers to analyze the script, including Knorosov and Parpola independently (Knorosov 1965; Parpola 1994), and recently Yadav et al. and Rao et al. (e.g. Yadav 2008; Rao 2009). Such studies use transcription schemes based on the Mahadevan concordance (1977), which identifies potentially 417 unique types, of which 40 only appear in damaged or uncertain contexts. The result of mitigating these damaged and ambiguous texts is a final corpus of 1548 short texts (single lines from seals mostly) and 7000 tokens (Yadav 2008; Rao 2009). Many signs appear to be combinations of other signs, but in the scheme they are treated as independent signs (Yadav 2007). Yadav et al. used computer techniques to show that the ordering of the signs is not random, that there are significant sequences in the texts, and used these sequences to break down longer texts into components of not more than four tokens each (Yadav 2007).

Rao et al. compared the Indus script to known scripts (most transcribed though Unicode), particularly with regard to conditional entropy, which is a measure of the flexibility in ordering of tokens. For instance, a system in which type B must always follow type A is rigid and has low entropy, whereas a system in which any type may follow type A with equal probability is flexible and has higher entropy. They calculate conditional entropy values for known linguistic and nonlinguistic systems to compare against the Indus script, and they find that the conditional entropy for the Indus script is similar to the conditional entropy of writing systems used to represent natural human languages (Rao 2009a). This agrees with the findings of Yadav et al. that the ordering of the signs is not random. This study was met with intense criticism on several counts, from bad source data to leaps in logic; the debate surrounding the Indus script is an acrimonious one. A slightly later study using a Markov model has been more successful. This study avoids comparing the Indus script to other known systems, as the claims involving these comparisons is what drew the most criticism (Patel 2010). From the Markov model, Rao et al. are able to confirm once more that the sequencing is not random and claim that the symbols appear to fall into functional equivalence classes. They suggest unpreserved signs from damaged texts using the model (Rao 2009b).

Recently another study has used entropy as in Rao (2009a) to propose that symbols on Pictish stones encode a language (Lee 2010). An insightful criticism of these entropy-based studies can be found in Sproat (2010).

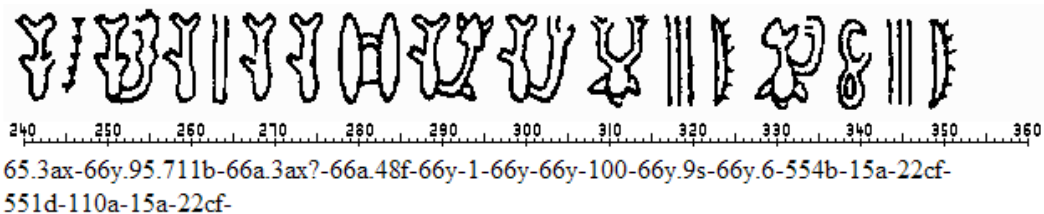
Rongo-Rongo

Figure 3: The Rongo-rongo script.

Top: detail from Tablet Q. Bottom: part of Barthel's tracings of the tablet, with transcription.

Rongo-Rongo refers to the mysterious glyphs found on artifacts from the island of Rapa Nui (Easter Island). The Rapa Nui spoken language is known in a modern form and through a few colonial-era documents, but the writing continues to perplex. The corpus is sparse and consists mostly of tablets marked on both sides in boustrophedon fashion. The glyphs resemble animals, beads, water, and many other items; and they are flexible in the sense that, while there seem to be sets of compositional features like heads, arm positions, and tails, the glyphs are the result of myriad configurations of these traits. A few scholars noted what they called parallel texts across multiple tablets

or within the same tablet, where the same sequence or nearly the same sequence is repeated. In 2003, Richard Sproat, then at the University of Illinois at Urbana-Champaign, used a computer to seek out these parallels, drawing from work on suffix arrays for string matching. Though this work was never published formally, it is freely available on his website and well known to Rongo-Rongo scholars. For input, Sproat used standard transcriptions of the tablets, based on Barthel and modified by the Rongorongo commission. Barthel's transcription allocates 799 mappings for signs, with similar signs grouped together, though some of the mappings are left empty. Diacritics indicate combinations or concatenations of signs, and lower case letters denote certain sign-specific variations (Pozdniakov 2007; Guy 2005). Sproat drops these decorations to obtain an input composed strictly of numbers. In the results, he presents several sequences that are similar, though not necessarily the same, because of the distance of his input from the actual text. In his search for "approximate matches" this flexibility is desirable, as it leads to more results that can then be examined by humans. He does note, though, that in some places his research suffered without him even knowing about it at first because of gaps in the source data where transcription simply had not been completed.

The most recent notable study of Rongo-Rongo using a computer (Pozdniakov 2007) exceeds previous work in a few ways. It identifies many more parallel sequences; it proposes allographs based on these, it present a 52-glyph inventory (instead of the original hundreds), and investigates the word structure and glyphs statistically based on this proposal. Unfortunately the primary conclusion of the paper is that the possibilities cannot be restricted enough to form any kind of decipherment. Other results are useful, however. Particularly the small glyph inventory arrived at statistically invites further research. Also, because of the high number of parallel sequences, the study authors suggest that the content of the corpus is unlikely to include anything other than record-keeping or similar repetitive tasks. The

identification of glyphs and their allographs in this paper has gathered some scholarly criticism. Because the method is not properly described, the counterexamples and arguments that critics are presenting cannot be checked against it (cf. Miller 2008).

Linear A



Figure 4: Linear A.
Detail on a tablet from Pylos.

One of the earliest languages of ancient Crete and the Greek mainland, Linear A has been easier to address than other unknown scripts. Most of the values for the phonetic symbols have been projected backward from the later Linear B script, deciphered in 1955, and scholars generally accept the result as a proper transliteration. In addition, the format of Linear A tablets is similar to Linear B commercial tablets, so scholars

are able to identify pictographic symbols for wine, grain, etc., as well as identify which symbols represent numeric values. However, Linear A does have unknown symbols (ones that do not match Linear B) as well as composite symbols (ligatures) that are not all fully understood (though many easily yield their components, others do not). These unknown symbols are referred to by standard enumerations just as in other unknown scripts. The problem with Linear A is not so much the values of the symbols as identifying what language it is supposed to represent. Scholars can sound out much of Linear A, but they do not know what the sounds mean. Most Linear A research has been concerned with comparing it to other early languages.

Packard (1974) was the first to publish a comprehensive computer analysis of Linear A, by calculating frequencies for each symbol, with one of his aims being to verify that the Linear B projections are reasonable. As in other decipherment studies, he uses the numeric transcription as input; but because of the unique situation of Linear A he is also able to analyze and present his results organized by phonetic sound. While frustrated by an undeveloped corpus at the time, his procedure was praised, and his results seem to favor the accepted assignments (Packard 1974; Chadwick 1977). Packard also identified parallel sequences and performed some other analyses. Recently, Brent Davis has applied Packard's procedure to a new aim: by calculating frequencies for symbols in the context of word position, to identify morphological suffixes. These morphological elements could be compared to other languages to try to identify Linear A's language family (Davis 2010).

Developing Methodology Using a Known Language

To develop methodologies that can be shown to be effective, one option is to use a known language, treating it as though it were unknown. In a recent successful attempt, Ugaritic was treated as an unknown language. Using statistics gathered from Hebrew,

a closely related language and writing system, the computer identified 29 of 30 letters and 60% of word meanings correctly (60% of those words that have cognates in Hebrew). For this study, rather than use an alphanumeric transcription scheme, a special system was developed to input Ugaritic symbols using a modern keyboard. This system is given only a mention in a news article; no details appear in the paper (Snyder 2010; Hardesty 2010).

Summary

While not many undeciphered languages still exist, those that do are receiving computer attention. Existing undeciphered languages have large symbol inventories and scattered corpora, and a computer can help organize and mine these extensive data sets. In addition, it can be expected that not all languages have even been discovered, as a recently found Olmec script demonstrates (Rodríguez Martínez 2006), as well as claims of a Pictish language (Lee 2010).

Different undeciphered languages have different problems. Rongo-Rongo scholars know some things about the Rapa Nui language but cannot read the writing, while Linear A scholars can read the writing somewhat without knowing its language. However there are problems they share. Common themes in computer decipherment include the dependence on a transcription scheme to provide manageable inputs and the search for parallel sequences of symbols to try to identify significant linguistic structures. Other questions address the classification of the language: Is it definitely a language? What other languages does it resemble? Computers allow the statistical analysis that can help answer these questions. Enumerated for the benefit of clarity, here are some of the questions that are asked about undeciphered writing systems. Some overlap. To show the importance of parallel sequences, questions that such analysis can help answer are marked with an asterisk (*). Where works are cited, they

are merely examples to help illuminate the nature of the question. Many other references are available on these subjects.

1. Is it a language? (Sproat 2010; Rao 2009a; Rao 2009b; Lee 2010)
 - a. Do its symbols follow rules found in known languages?
 - Type number and frequency
 - Zipf's Law
 - Entropy
 - Statistical models
2. What other languages is it most closely related to? (Packard 1974)
 - a. What writing systems does it resemble graphically?
 - b. Can values be assigned to symbols in a way that appears to create words or structures found in some other language?
 - c. *Can rules or structures be identified that parallel rules or structures in some other language?
3. What are the types of the writing system? (Coe 1999)
 - a. *Do the types vary by context or are they consistent?
 - b. *Are the types distinct or can they be combined to create composites or ligatures? If composites exist, how do they break down?
4. Is it primarily alphabetic, syllabic, or ideographic? Which symbols are phonetic and which are ideographic? (Chadwick 2004)
 - a. How many types are found in the writing system?
 - b. What ordering and placement rules do the types follow?
5. What is the morphological structure of the language?
 - a. Are any types most prominent in a particular position in words (usually beginning or end)? (Davis 2010; Rao 2009 b)
 - b. Are any types found commonly in conjunction with other types?
 - c. *Are any common sequences found that differ by only one or two symbols? (Chadwick 2004)

6. Is the transcription scheme helpful?
 - a. Does it cover all of the types in the writing system?
 - b. Does it cover variation in types?
 - c. Does it represent relationships between types that might be significant?
 - d. Does it organize and communicate these features effectively?

Computer Recognition of Written Images

Printed Character Recognition

The first serious studies in optical character recognition took place in the 1960's, and today the technology has reached such a level that a typed or printed English document can be scanned and its linguistic content digitized with claimed accuracy of 97-99% (characters recognized correctly) (Mori 1999: xiii, 6, 7). Character recognition today generally follows three steps (Mori 1999; Cheriet 2007):

1. Preprocessing: The image is prepared for analysis.
2. Feature extraction: Identifying features of the character image are collected. One of the most important decisions in character recognition is which features to collect; researchers have explored optimizing the selection through automated processes.
3. Pattern matching: The features are used to classify the image. This can be by a decision tree, a neural network, or any other decision-making algorithm.

The features used for character recognition can be anything that helps describe the token. In image-domain methods each pixel is used; one token image is compared to another by measuring the differences between their pixels (Manmatha 2003). An affine transformation can be applied to allow some flexibility, though this is slow (Manmatha 1995; Barmpoutis 2009). These approaches save on extraction since no additional calculation is needed, but comparison can be long since the features are

numerous. The crossing method, drawing lines at intervals through the image and recording how many times they cross the foreground, is a simple feature extraction technique. It involves fewer matches that are fast to perform, and it overcomes certain kinds of variation, but it encodes less information. The decision on which technique to use is therefore one that should be made carefully (Mori 1999: 21). The aim is not only efficiency but correct classification, and to this end all kinds of features have been tried to allow recognition despite token variation, to capture the essential shape of a type, especially contour and stream following, orientation matrices, and mathematical descriptors. For more information see Mori (1999) and Cheriet (2007). Printed character recognition deals with basic problems found in all cases of computer recognition of writing, particularly extraction of words and characters and separation of text from graphics (cf. Fletcher 1988); but the source documents are generally cleaner and more regular than those used in handwriting recognition.

In font-dependent character recognition, the character image can be expected to appear exactly or almost exactly in a known configuration, so it requires only simple image analysis. A pixel-by-pixel comparison will do. Font-independent character recognition requires more complex systems, as the actual image being analyzed could be in any one of a vast number of configurations and still needs to be recognized as the correct type. Contour direction, partitioned into a spatial matrix, is one popular feature choice in these cases (Cheriet 2007: 59). However, character recognition researchers routinely improve accuracy by considering the entire word or even beyond the word. For small domains, the word can be matched against a lexicon. Otherwise, the lexicon can be re-organized to allow faster search, or a Markov model can help anticipate the type of the character given the context (Cheriet 2007: 234-237).

Most work to date on character recognition has been conducted on the English alphabet and Mandarin Chinese, although significant progress has been made on Indian scripts (Bhardwaj 2009; cf. Bruel 2009, Kompalli 2007).

Handwriting Recognition

The problem of handwriting recognition shares more features with the problem of undeciphered writings than does simple character recognition because it is more concerned with the original images on the manuscript, and of course most undeciphered scripts are handwritten rather than typed. Handwriting is not necessarily consistent, meaning representations of the same type could vary substantially within the same document. Segmentation is of course more complicated than in printed character recognition, as tokens can connect or overlap, also in inconsistent ways (cf. Nicchiotti 2000, Yanikoglu 1998). A Markov model can be used to segment handwriting at the same time as recognition (Cheriet 2007: 237). There are also approaches that avoid segmentation, such as matching against a lexicon, feasible for a limited domain (Cheriet 2007: 251).

Manmatha et al. show how to group handwritten words into an index without worrying about segmentation or recognition, a process they call word spotting. Once the words have been classified, the end-user can browse all the images of a particular word in a document or series of documents. Automated recognition is not performed; rather the words can be identified by a human after indexing. Manmatha and related scholars have tested many image comparison algorithms for word spotting. Euclidean distance mapping, their first and baseline attempt, takes as input the XOR result of the two images to be compared and measures the distance from each foreground pixel to a background pixel (Manmatha 1995; 1996). A recent success uses dynamic time warping, which models the two word images along the horizontal axis as distortions of

each other (Rath 2003). However, an even more recent attempt using corner detections is faster and almost as accurate (Rothfeder 2003). Up-to-date research on word spotting can be found here: http://orange.cs.umass.edu/irdemo/hw-demo/wordspot_retr.html.

Epigraphy

Computers have been used for a long time in storing and propagating representations of ancient inscriptions, whether from photographs or from squeezes, which are popular tools for making impressions of inscriptions on paper. Techniques even exist for capturing three-dimensional information, from photographs taken at multiple angles (Meyer 2004), from squeezes scanned under multiple lightings (Barmpoutis 2009), and from more expensive laser scanning procedures (Bodard 2009). The computer can then be used to identify the separate elements. In a study of Greek inscriptions from Epidauros, the computer successfully segmented and clustered the tokens according to their graphical similarity, effectively creating an atlas of all the tokens by type. A high accuracy was achieved by making use of three-dimensional data from the squeezes. The clustering was then used to propose a dating scheme for the inscriptions (Barmpoutis 2009). Another study has shown how Egyptian hieroglyphs can be indexed by their constituent Bezier curves (Meyer 2006), and it seems only a small step to automated recognition of hieroglyphs through pattern matching.

Automatic Handwriting

As there has been no direct scientific treatment of automatic handwriting, the author uses research in related phenomena to form a theory. Therefore this section is necessarily long.

While there has been no scientific treatment of automatic writing, there have been intellectual inquiries into the mystic claims of automatic writing. These mystic writings may be in a language known to the subject or may be in a mysterious language. In a famous case, Professor Théodore Flournoy tracked the notable Hélène Smith, who claimed to be receiving correspondences in a Martian language. As she would translate them into French, he was able to conduct a study. He found certain resemblances to French; despite this, or even because of this, he remained convinced that it was not a purposeful ruse but a true unconscious phenomenon. The Martian phonemes are completely contained within the French phoneme inventory, for example, and the Martian graphemes also are mapped to the same sounds as French graphemes. Syntax is exactly the same as in French (Flournoy 1900: 246-251). Flournoy describes other instances of Mlle. Smith’s automatic writing in familiar (not Martian) characters (cf. Flournoy 1900: 288), but the Martian is relevant because it appears in a strange writing system, akin to the Codex Seraphinianus. It should be stressed that Serafini makes no mystic claims about the codex. Starting with Pierre Janet, psychologists have confirmed that the ability to write unconsciously does exist (Van der Hart 1989).



Figure 5: Mlle. Smith’s Martian graphemes as collected by Flournoy. (Flournoy 1900: 208)

Glossolalia and Speaking in Tongues

There is a process closely related to automatic writing that has received attention in some circles: It is the phenomenon called “speaking in tongues” exhibited by some Christian spiritualists. It falls into a class of behaviors called glossolalia. Like automatic writing, it is given to mysticism. (Mlle. Smith not only wrote in her Martian language but also spoke it.) The subject, often in the middle of prayer or worship, spontaneously babbles. In some cases, the babbling is claimed to be in a language unknown to the speaker but is recognized and understood by someone else. In other cases it is accepted as a “spiritual” language, unknown to anyone on earth (Cutten 1927: 116, 164, 165). Glossolalia occurs in other cultures and has occurred throughout history. Specifically with regard to the Christian practice, Lombard identified four types of glossolalia that could occur independently or blended: pre-vocalizations like groans, verbalization, pseudo-language, and speaking in foreign languages (“xenoglossie”), and May later adapted these to apply to non-Christian instances (May 1956). As in automatic writing, something novel is produced with minimal conscious effort. Especially in the early twentieth century when it was becoming widespread, people studied speaking in tongues in a psychological context. The conclusions are brief and necessarily untested: that it results when the conscious self disintegrates, allowing the unconscious to control motor functions. In response to speaking languages not known to the subject, these scholars bring up case studies to show that this could be enhanced (“exalted”) memory allowed to come forth in the absence of the conscious self: For instance a chambermaid in a fit spoke Latin and Greek, and it was found that in a past job she had swept while her master recited long passages in those languages (Cutten 1927: 176). While in the early twentieth century terms were used such as ecstasy and hysteria, by the 1970’s this kind of disintegration of consciousness had become known as a dissociative episode (Arkin 1981: 63, 64).

As a side note, it might be worth considering the cooing and babbling of human babies as a spontaneous and creative automatic process, based on the observations that all babies engage in at least the cooing stage, even babies of deaf parents, and that progress from one stage to the next does not require practice (Lenneberg 1967). If so, perhaps the mechanisms that produce glossolalia are left over from this early development.

Other Related Processes

Similar processes have been studied more scientifically. Like automatic writing and glossolalia, musical improvisation creates something new, sometimes without conscious control (Nettl and Russell 1998: 82, 171, 339). Dreaming, as well, can be seen as the brain creating something new in the absence of conscious control. Brain scans and lesion studies of these processes suggest that a prefrontal area is inhibited while another prefrontal area, the area responsible for interest and curiosity, is excited (Limb 2008; Dietrich 2003; Solms 1997). This theory has been extended to scribbling as well (Sheridan 2002). The main idea is that the brain enters a state in which it is interested but not regulated. It then uses whatever resources are available to it (memory and symbolic rules) to manifest something concrete (Solms 1997).

Summary

Based on previous research, it can be expected that if the writing of the Codex Seraphinianus was generated automatically then it is rooted in Serafini's acquired experience with language. This idea can be kept in mind while viewing the results of the following analysis.

MATERIALS

Hardware and Operating System

All of the procedures described in this paper were performed with a HP Pavilion (model dv6119us) with a 15 inch display, running Microsoft Windows XP Media Center Edition, Version 2002, Service Pack 3. The processor is AMD Turion(tm) 64 X2 Mobile, 803 MHz (960 MB RAM). This can be considered a low end machine compared to machines being offered today.

Development Environment

Applications were developed in Microsoft Visual Studio 2005 Professional Edition, Service Pack 2, using the C# language.

Third-Party Software

To save the time and effort involved in developing complex graphical analysis tools, AForge.NET (version 1.7.0) was used as an image manipulation toolkit, added as a reference to the project. AForge.NET is a software development kit with a full suite of artificial intelligence utilities, including imaging, machine learning, and robotics algorithms. Only the AForge.Imaging namespace was used in this study. The entire framework is freely available at <http://code.google.com/p/aforge/>. It will be clearly stated when the procedures make use of AForge.NET.

Corpus

The Codex Seraphinianus was acquired in Adobe Acrobat Reader format (PDF) and converted into 369 individual images, one per page, using PDF to Image Converter 1.0 by PDF-TIFF-Tools.com. The output images measure 1573 by 2169 pixels, with a resolution of 150 dpi and a bit depth of 24.

To avoid the very first pages of the codex which have non-standard formatting, the first page used for the study is page 10.

METHODS AND RESULTS

Header Text (“Capital Letters”)

For these implementations, the header tokens were extracted (“blobbed”) using the third-party software. A function was devised to measure the graphical similarity between tokens. In one implementation, the tokens in the first three chapters of the codex were then classified into types using a simple automated algorithm combined with human supervision. In another implementation, a search engine demonstrates the possibility of locating parallel sequences, a common task for analyzing undeciphered languages, based on graphical similarity alone.

Preparation for Extraction

When an image of one of the pages of the codex is first processed by the application suite developed in this study (described in the rest of the paper), three AForge.NET filters are applied to it. A GrayscaleRMY filter is used to convert it to 8bpp, the format that is required by the AForge.NET blobbing algorithms (see below for a description of blobbing). A Threshold filter increases the definition of the token lines to a black and white contrast. Thirdly, an Invert filter is used to get the tokens to be white on a black background so they can be properly blobbed. The AForge.NET default values were used for each of these filters.

To save a significant amount of time during the blobbing process, the image is cropped before extraction to an area that can be reasonably anticipated to encompass the header text and not much extraneous content, using an AForge.NET Crop filter. The rectangle used for cropping is location (0, 0) and size (1572, 270).

Extraction

A blobbing algorithm is meant to separate foreground images from a background by identifying contiguous shapes. The result of blobbing is a collection of blobs that represent these shapes. In the case of the codex headers, a blob is produced for each token.

The AForge.NET BlobCounter class is used to determine the location and extent of actual tokens on the page. To extract the majuscule letters, the BlobCounter is set up with the following values:

```
FilterBlobs = true  
MinHeight = 20  
MinWidth = 20  
MaxHeight = 300  
MaxWidth = 300
```

These settings yield only blobs that fall into the correct size range for the capital letters. Additionally, the implementation sorts the blobs by topmost first and considers only blobs that fall within a twenty pixel threshold of the topmost blob. This ensures that only the blobs in the header are considered (not blobs that might appear farther down the page in illustrations for example). Any blob that passes all of these filters is considered a valid capital letter token. Its page number, location, extents, and visual data are stored in an array that can be accessed by the application suite.

This technique does not work on pages in which the header is not near the top, including the title page and the landscape-oriented pages. These pages were discarded from the study.

Rarely, because of the integrity of the processed page, extraction failed on particular tokens. For instance, out of the 264 tokens in chapter one (pages 10-43), five

were not properly extracted, with only part of the token being extracted. These failed extractions were disregarded.

Identification of Similar Tokens

The application suite compares two tokens using a simple whitespace count with jitter as described below. First it scales both images to the same size (70 x 70) using an AForge.NET ResizeBilinear filter (a practice called normalization). More information about the normalized size is below. It then takes the ratio of white pixels in the XORed image to the white pixels in both original images combined (the tokens are white on a black background because of the filters). Because the images are not necessarily lined up to begin with, jitter is used as described in Manmatha (1995) to find the best alignment between them. Specifically, one of the images is tested with x offsets of -4, -2, 0, 2, 4 and y offsets of -4, -2, 0, 2, 4, resulting in 25 different tests instead of one. (The AForge.NET CanvasMove filter was used to create the offset images.) The resulting float value is used as a measure of the similarity between the two images. This value can range from zero (if the two images match exactly) to one (if the images do not overlap at all). An alternative comparison algorithm was also tried, Euclidean distance mapping as described in Manmatha (1995). While it is demonstrated to give good results for handwritten words, it appeared to produce no significantly better results in this application, probably because of the relative shortness of these images (letters instead of words).

Reducing the normalized size, of course, reduces the number of pixels and therefore lowers processing time. (When this is done the jitter values also need to be adjusted.) However, there is a tradeoff because the resolution of the image is also reduced. Probably, a more intricate comparison algorithm can be applied in future research to allow much smaller images, saving processing time. The images and

following caption below show the XOR comparison algorithm at work for two different tokens at large and small normalized sizes.

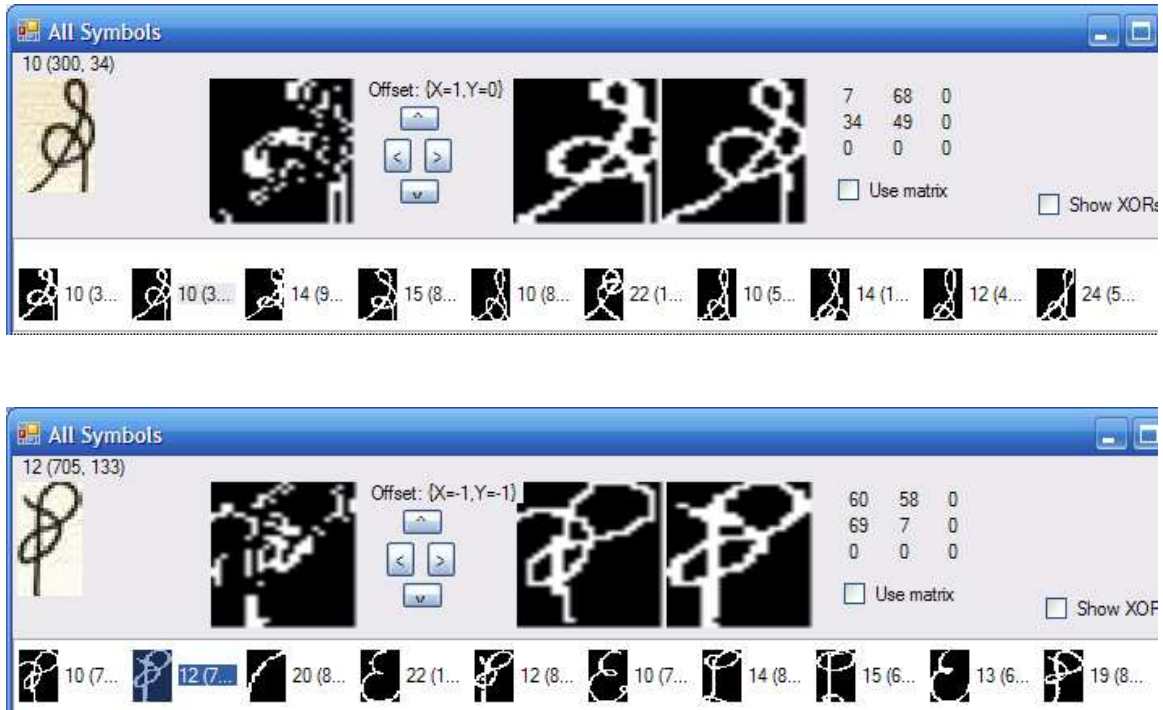


Figure 6: Comparison of header tokens.

The first token in the bottom row of each image is the target. In each case, the second token has been selected to show some comparison features of the tool. The leftmost box shows the selected token as it appears in the codex. The rightmost boxes show the target (on the left, stretched to a large square) and the selected (on the right, stretched to a large square) tokens side by side. The remaining box shows the result of the XOR operation. The tool also reports the most successful offset and allows the user to jitter the XORed image using the arrow buttons to verify this. In order: Target page 10 token 3, size 70x70; target page 10 token 3, size 20x30; target page 10 token 7, size 70x70; target page 10 token 7, size 20x30. (The isolated curve that appears in the results is an extraction error.)

One attempt to patch the algorithm is the incorporation of a whitespace matrix, which splits the image into nine sections and considers the white-count difference in each when making its decision. While this worked as expected, it did not help the results and even degraded them when two tokens of the same type varied too much.

The technique ultimately used to proceed with the study uses the 70x70 size with no additional considerations.

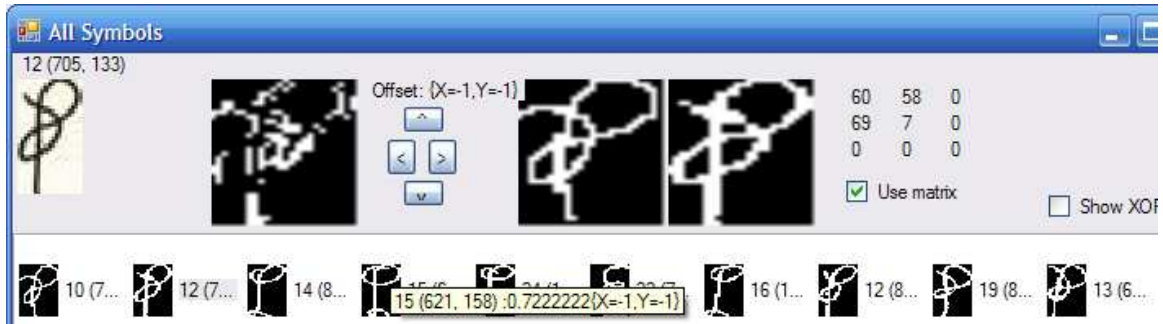


Figure 7: Applying a whitespace matrix to the last example from the previous figure. The rightmost numbers represent whitespace counts in each section of the matrix for the selected token. Notice that, while the results match the target closely with regard to whitespace balance, most of the results in the middle of the row appear to be of a different type.

Study 1: Consolidation into Types

Rationale: If the tokens of the writing system can be separated into a finite number of types, this number might indicate whether it is primarily alphabetic, syllabic, or ideographic. In addition, the identification of types is necessary for many forms of linguistic analysis including n-gram analysis and Markov modeling. However, it is important to note that to identify types is to enter into the limitations of a transcription scheme described in the introduction.

Similar tokens are automatically grouped together by the value explained above in a simple, fast, flat clustering scheme: For each token, if it meets a similarity threshold (< 0.65) with the first added member of an existing type, it is added to that type; otherwise it creates a new type. No attempt is made to maintain a medoid or centroid. Single-link clustering was also tried; as expected, it produced somewhat better results but took a much longer time. This is an area for further research.

Because of the nature of the writing in the codex, no attempt was made to measure the accuracy of the clustering scheme. Though some of the tokens definitely appear to be of the same type, the classification of many of the tokens is ambiguous. Since the writing is asemantic, this is no surprise. It is clear that these decisions about the tokens cannot be made without human supervision.

Once the types are tentatively identified by the computer, the process of human supervision begins. An interface was developed to allow the human user to review the types, combine types, and easily move tokens among existing types or use them to create new types mainly by dragging and dropping. In addition, each token can be viewed in its original context on the page. This is an important stage during which the human user is at liberty to examine the tokens in these two contexts: similarity with other tokens and usage on the page.

The header tokens of the first three chapters of the codex (pages 10-123) were classified into types using this application. Because the amount of human interaction was substantial, as some of the classification decisions were time-consuming, saving and loading functionality was devised so that each chapter could be processed and saved separately and then combined with the others later. This allows the user to choose to work with a small chunk of the data at a time. In addition, limiting the amount of working data keeps automatic comparison time manageable, since each new token must in the worst case be compared to every other token. While not large, the resulting collection could be used as training data to aid in the classification of the rest of the codex. It is described below.

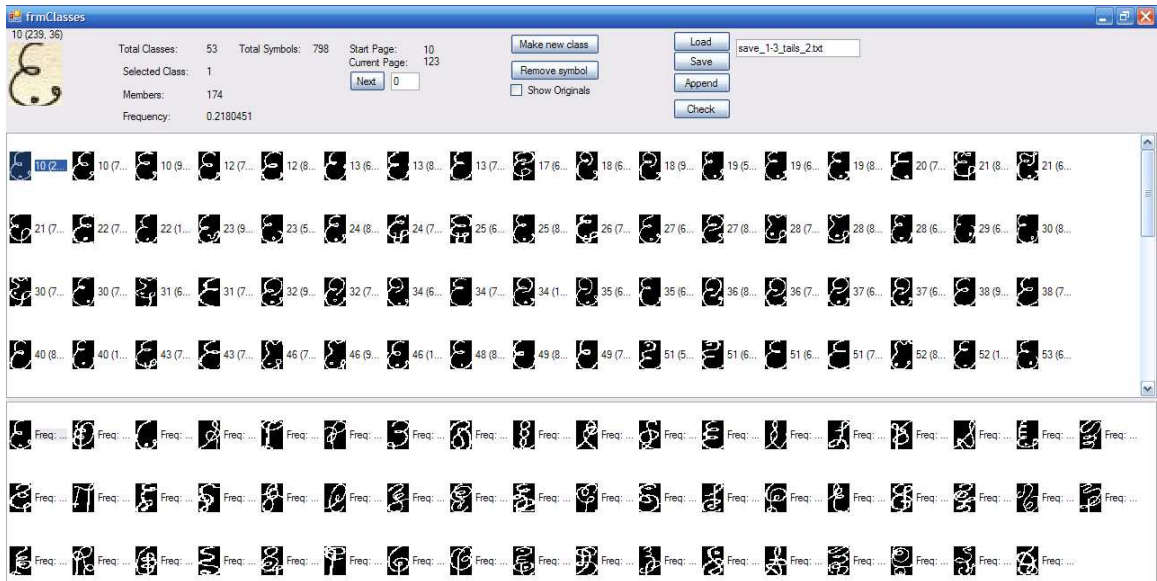


Figure 8: Typing app showing results for the first three chapters of the codex.
The bottom pane shows the final 53 types. The top pane is showing all of the tokens of the first type.

Several observations surface just from the first few chapters. Chapter 1 (pages 10-43) shows dominance of a few types, and there are many types that only appear once. Chapter 2 (pages 46-103) provides by far the most tokens of the three chapters and introduces much more diversity of tokens. The 438 tokens could not be classified comfortably by the author at first glance into any fewer than 70 types, 34 of which contained only a single token. Chapter 3, with the fewest tokens, mostly fit into the classes already created from the previous chapters. When all three chapters were combined, there were 798 tokens classified into around 70 types. This limited number of types is the result of the author ignoring small variations between the tokens, such as a dot appearing inside the convexity, a loop appearing at an endpoint, and certain differences in curve contours. Otherwise there would be substantially more types.

However, this initial classification led to a discovery. After examining the large number of types that only occur once, the author realized that most of these precede the dash on an illustration page and hence are of no linguistic value. They effectively

would mean something like: “This is a unique identifier for this illustration page and serves no other purpose.” This relationship will be visited further in the discussion section. Discarding these unique tokens provided a final classification of 53 types.

It is useful to point out here the templates or patterns that recur frequently in the page headers. This will make it easier to talk about where the types are found. It does not take a computer to notice these, but the typing application certainly helps highlight them since certain types appear mainly in these patterns. The usages for these patterns were solidified by the search engine application described in the next section. The transcriptions are based on the suggestions in the table for the types, which immediately follows it. More will be said about the patterns later.

Table 1: Patterns in the codex headers.

First Occurrence	Trans.	Usage
Page 10	DSD	Ends header on the first page of a chapter
Page 10	ESSH3	Starts header on the first page of a chapter
Page 11	DE2C	Table of contents header
Page 12	DS	Starts header
Page 12	SFA	Ends header
Page 88	L2F	Starts header
Page 88	TE	Ends header

Table 2: Types found in chapters 1-3, with suggested names and transcriptions.



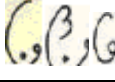

	Frequency	Possible Name	Possible Trans.	Examples	Found
1	174 (.218)	E type	E		Throughout In the pattern TE
2	103 (.129)	D type	D		Throughout In the pattern DS
3	84 (.105)	C type	C		Throughout
4	68 (.085)	S type	S		Throughout In the patterns DS and SFA

Table 2 (continued)










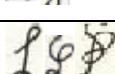







5	49 (.061)	Phi type	F		Throughout In the patterns L2F and SFA
6	48 (.060)	P type	P		Throughout
7	29 (.036)	3 type	A		Mainly in the pattern SFA
8	28 (.035)	8 type	B		Mainly chapters 2-3
9	26 (.033)	3-knot type	H3		Throughout
10	21 (.026)	3-leaf (trefoil) type	T		Mainly in the pattern TE
11	18 (.023)	Hook type	L2		Only in the pattern L2F
12	18 (.023)	Double-E type	E2		Throughout
13	16 (.020)	2-knot type	H2		Throughout chapter 2
14	12 (.015)	L type	L		Throughout chapters 1-2
15	10 (.013)	PS type (or Harp type)	P2		Throughout chapters 2-3
16	9 (.011)	SC type (or Half Harp)	S2		Throughout chapters 2-3
17	6 (.008)	Triple-E type	E3		Chapters 2-3
18	5 (.006)	Dragon type	J		Chapter 2
19	5 (.006)	Double-3 type	A2		Chapter 2 and distorted on page 25
20	5 (.006)	Pi type	N		All chapters
21	4 (.005)	Snail type	G		Chapters 2-3, only three words

Table 2 (continued)





















22	4 (.005)	(Inverted) Treble type	Bo		Chapter 2
23	4 (.005)	E8 type	Eb		Pages 108 and 109
24	4 (.005)	Double-Y type	Y2		Chapter 2
25	3 (.004)	Snail-8 type	Gb		Pages 62 and 67
26	3 (.004)	Y type	Y		Chapter 2
27	3 (.004)	Pinwheel type	X		Page 18 and distorted on page 71
28	3 (.004)	L-Hook type	Ll2		Pages 50 and 55
29	3 (.004)	E-stick type	Ei		Pages 16 and 46
30	2 (.003)	Y-8 type	Yb		Page 96
31	2 (.003)	Curl-E type	We		Pages 122 and 123 (one word)
32	2 (.003)	Spring type or Shrub type	I		Pages 26 and 27
33	2 (.003)	3-bar type	Ai		Pages 70 and 71
34	2 (.003)	R type	R		Pages 64 and 66
35	2 (.003)	Dragon-L type	Jl		Page 87
36	2 (.003)	L-8 type	Lb		Pages 19 and 93
37	2 (.003)	Triple-C type	C3		Pages 12 and 13

Table 2 (continued)

38	2 (.003)	E-L type	El		Pages 122 and 123 (one word)
39	1 (.001)	Trefoil-stick type	Ti		Page 120
40	1 (.001)	C-D type	Cd		Page 96
41	1 (.001)	Smoke type	W		Page 89
42	1 (.001)	2-knot-E type	H2e		Page 101
43	1 (.001)	Left-8 type	B2		Page 23
44	1 (.001)	S-bar type	Si		Page 67
45	1 (.001)	6's type	O		Page 57
46	1 (.001)	3-8 type	Ab		Page 19
47	1 (.001)	Left 3-8 type	Ab2		Page 69
48	1 (.001)	Gears type	Go		Page 65
49	1 (.001)	Phi-3 (Tree) type	Fa		Page 40
50	1 (.001)	<none>	Sy?		Page 53
51	1 (.001)	Spiral-stick type	Oi		Page 89
52	1 (.001)	Phi-H type	Fh		Page 27
53	1 (.001)	E-Phi type	Ef		Page 51

Study 2: Search Engine

Rationale: A large portion of effort on undeciphered languages is spent identifying parallel sequences of tokens throughout the corpus, on the expectation that the more contexts in which a sequence can be found the better understood it becomes. This application is meant as a proof of concept that these sequences can reasonably be found by searching the images.

Independent from the interface for identifying types, a search engine interface was produced, based on the same similarity value explained above. This is a fully visual search engine, not based on any sort of transcription or typing scheme. The user places image data on the Microsoft Windows clipboard (by copying a token or sequence of tokens from one of the pages of the *Codex* in any imaging program) and then starts the search. The application attempts to find token sequences that match the token sequences on the clipboard by taking the average similarity between the sequence on the clipboard and each sequence in the page header. Any sequence satisfying the similarity threshold (< 0.65) is returned in the search results. For instance, if the sequence on the clipboard is two tokens long, the application will test this against each bigram in the header. It ignores empty space so that it does consider two tokens separated by any distance of empty space as a valid bigram to test against. The user can easily recognize which results fit the search criteria best and click on these to view them in context on the page, ignoring the ones that do not.

The search runs on a separate thread so that the user can view the search results while the program is still performing analysis on remaining pages.

When specifying the search criteria, the user might not want to scale all token images to the same size as described above. This is because the size of the image is an

important clue for matching. When a small dot is scaled to 70 pixels square, for example, it may appear to the algorithm to be similar to a long vertical bar also scaled to 70 pixels square. On the other hand, it would be undesirable to miss valid matches simply because of small variations in size. For this reason, the search interface provides the option to the user in the form of a “do not scale” checkbox to scale or not to scale the images used for comparison. Instead of scaling, the application creates a new image of the desired size and writes the token onto it, effectively padding it. (A bug to note is that tokens exceeding 70 pixels are clipped according to this padding scheme.) The user can fetch matches both with the box checked and then unchecked to obtain thorough results.

The search engine allows adjustment of two threshold values to control results: individual threshold and overall threshold. Each token must individually meet the individual threshold, and the average of all the tokens must meet the overall threshold. Setting the individual threshold slightly lower than the overall threshold yields the best results, because it allows any single token to fail the match slightly as long as the other tokens exceed the match enough to recover.

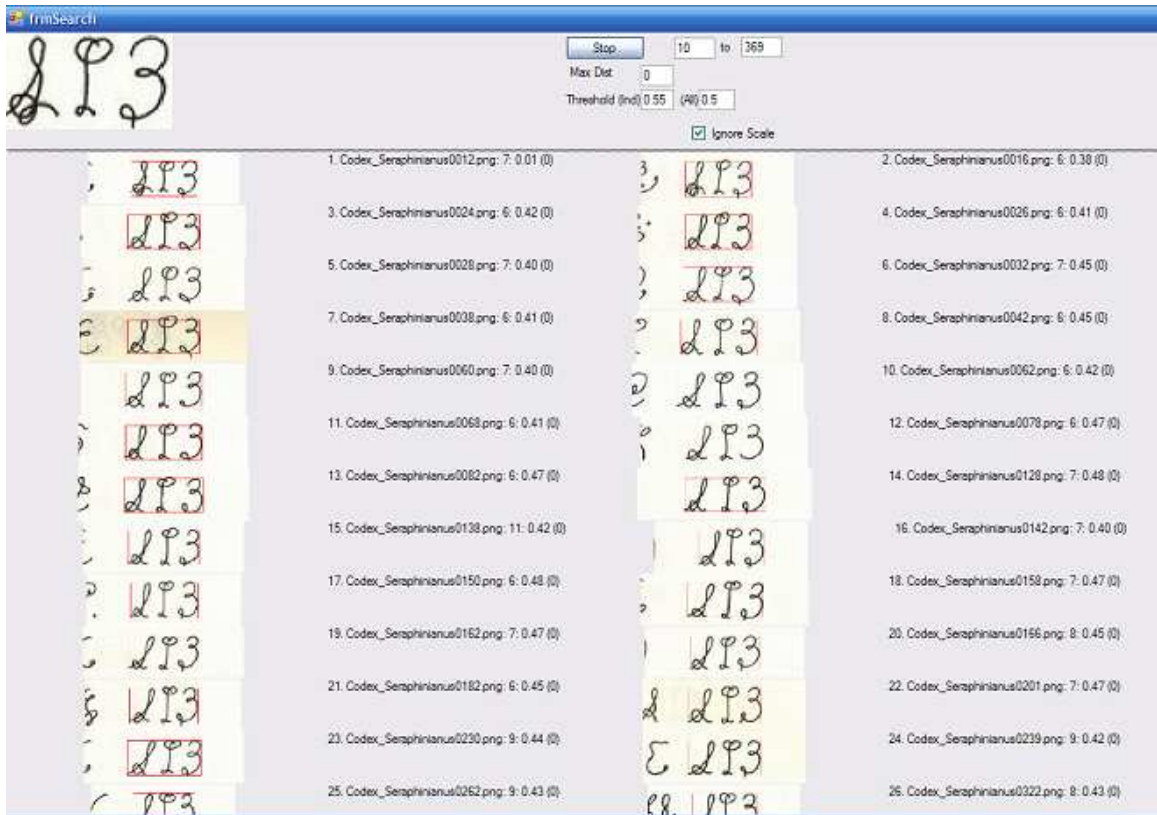


Figure 9: Search results for the common sequence SFA.

Thresholds: Individual = .55, All = .5. Max Dist (see below) is set to 0. The first three results are from pages 12, 16, and 24. The last result fully visible is from page 239.

Finally, for a multi-token sequence, the user can specify the maximum number of tokens that may not match (“max dist”). Setting this value to 1 allows results for which the algorithm may have given a false negative on one token but correctly identified the others.

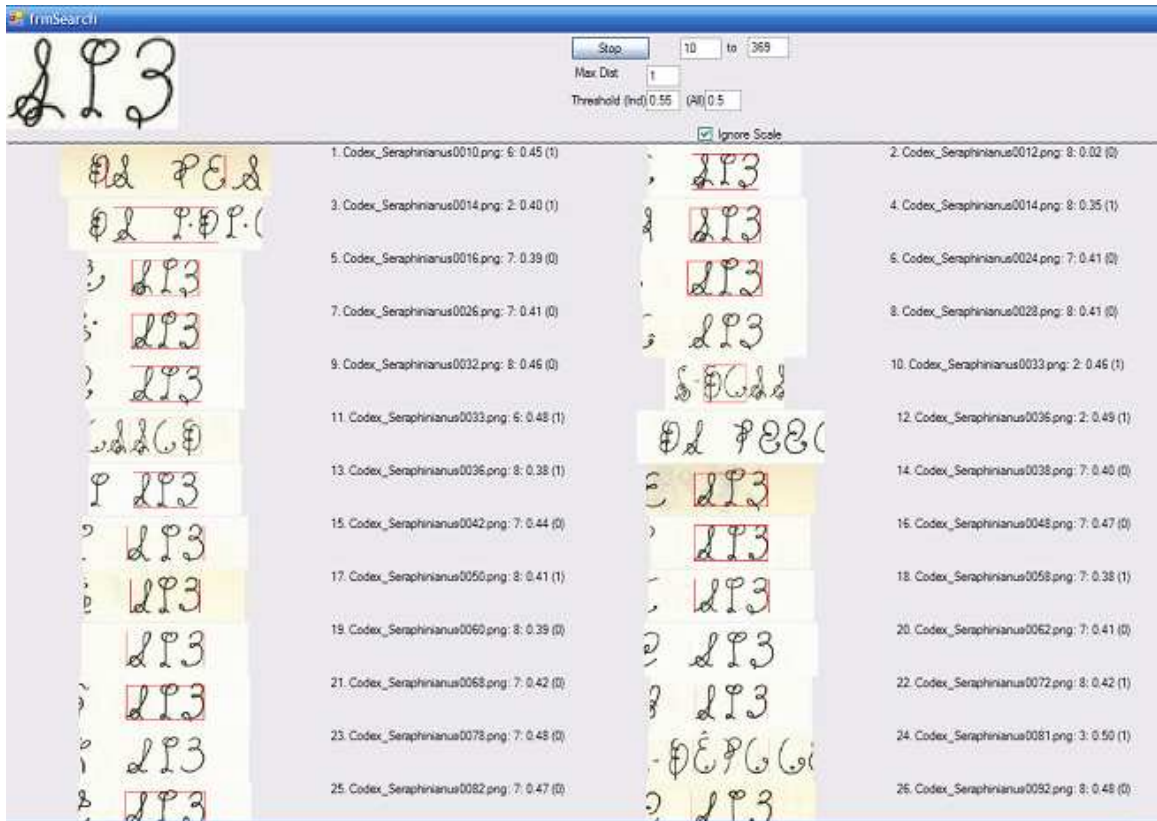


Figure 10: As above but with Max Dist set to 1. Many more results are returned, most of which are instances of the search string. The first three instances of SFE found are from pages 12, 14, and 16. The last fully visible instance of SFE is from page 78.

Additionally, this “max dist” feature allows matches that differ in one or more tokens from the target. Scholars seek such matches because they provide clues about which tokens might be interchangeable, either because they are orthographic variants of the same type or because of the morphological rules of the language. This is demonstrated through the search results shown above. The first three occurrences of SFE clearly begin with S-like tokens with two large overlapping loops at their endpoints. The rest of the matches begin with tokens that lack these loops. Because the inclusion of additional loops three times in a row accidentally is unlikely, especially right at the beginning of the book, and because all of these matches occur in

apparently exactly the same context, it would seem that these tokens are freely interchangeable.

We can also see variation in the token E. In almost every case it ends in a small loop, with one exception, page 78. Because the omission of a loop accidentally is not unlikely, and because it only occurs in one isolated case, this could be dismissed as a mistake.

It should also be noted that the search engine makes immediately evident the repeated use of the word throughout the codex headers, always in the same position. While this could easily be noticed by a human researcher, it would take longer to document each case.

In addition to exploring similarities among the headers, the search engine can explore uniqueness in the headers. When the type classifier yielded several types of only one token, it was found that these one-shot tokens usually appear before the dash on the image pages. Further exploring this phenomenon, using the search engine to find all dashes in the codex is an easy way to browse all of the pre-dash tokens. The results show that the pre-dash tokens are almost all unique. Even between pages that seem to show closely related illustrations, the tokens do not resemble each other.

Initial results from the search engine suggested that apart from the obvious repeated words, recurrence in the header sequences is rare. This was briefly explored by searching for sequences composed of common types in the headers that might be expected to recur. Results can be seen below. Apart from the target and its corresponding illustration page, the appearance of the word or sequence is unique.

Table 3: Words or parts of words made of common types in the headers which are nevertheless unique.
The left column shows the target and associated illustration page where it is often repeated. The right side shows some other search results. None of the search results are exactly the same sequence as the target. To the left are the page-numbers on which the headers occur.

Target sequence and illustration page	Other finds
<p>12 </p> <p>13 </p>	<p>10 </p> <p>48 </p> <p>60 </p> <p>147 </p> <p>269 </p>
<p>112 </p> <p><Not on illustration page></p>	<p>10 </p> <p>177 </p> <p>204 </p> <p>295 </p>
<p>52 </p> <p><Not on illustration page></p>	<p>66 </p> <p>80 </p> <p>210 </p> <p>320 </p>

Body Text (“Lower Case Letters”)

For the body text, whole words were extracted (“blobbed”). A first implementation orders all words on a single page by length to allow for some initial impressions. Then a function was devised to measure the graphical similarity between words. A search engine demonstrates the validity and shortcomings of this function and allows for further observations concerning the distribution of words in the codex.

Preparation for Extraction

The body text was prepared for blobbing in the same way as the header text, with minor differences by trial and error. First an AForge.NET GrayscaleBT709 filter was applied, then a Threshold filter with a threshold value of 170, and lastly the Invert filter.

Extraction

Body text words were blobbed by AForge.NET in the same way as the header text characters. Table of contents and landscape orientation pages were discarded. Several techniques had to be implemented to mitigate the unpredictable layout of the body text. The complicating factors are:

1. words too close vertically
2. words squeezed in between lines
3. words running into the page margin or gutter shadow
4. words accidentally broken during preprocessing
5. dots and other diacritics appearing above words
6. numbers, strokes, and isolated symbols appearing to serve a mathematical purpose
7. charts, diagrams, and other unconventional configurations of tokens

8. illustrations

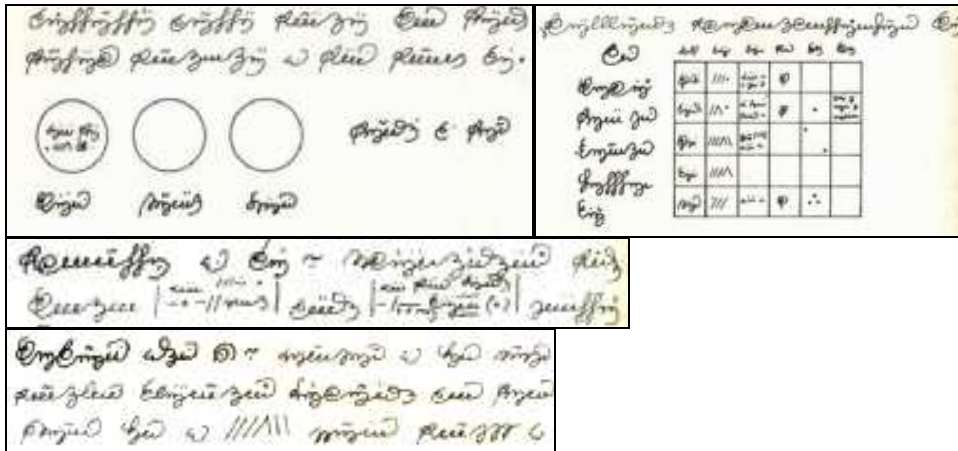


Figure 13: Samples from page 12 of the codex showing text mixed with (a) diagrams, (b) tables, (c) two-line compartments, and (d) non-word strokes.

To deal with these factors the following method was used.

1. Any blob that is taller than expected (> 40) and overlaps a blob below is reduced so that it does not overlap the blob below.
2. Any blob that is still taller than expected is a candidate for being split in half horizontally. The blackest row is found, and if the whitespace above and below this row is approximately equal (within 300) the blob is split on the row. This is to fix words that are too close vertically.
3. Blobs that are very close horizontally are merged. This is to fix words broken during preprocessing and to attempt to attach the extra blobs formed in step 2 to their proper owners. To do this, a box the length of the blob and 6 pixels high is drawn through the middle of the blob. It is then inflated 6 pixels on either side, and any intersection with any other blob results in a merge with that blob. Any blob taller than 40 pixels cannot be merged.
4. Blobs that are contained fully inside other blobs are merged. This is to make sure dots and diacritics are included. To save time, this is combined with the next step.

5. Blobs that are very close vertically are merged. This re-merges any extra blobs remaining from the split in step 2 and also helps combine the compartmentalized blobs so that they can be filtered out. The blob is compared against all other blobs; any blobs that intersect its bottom are merged if either (a) the blobs have exactly the same x position on the page or (b) if both blobs are thin ($H < 15$). As with the horizontal merge, any blob taller than 40 pixels cannot be merged.
6. Step 1 is repeated since new blobs have potentially been formed.
7. Any blobs that cannot be words due to height (< 20 or > 60), total size ($H+W < 30$), or aspect ratio ($H/W > 1.2$) are discarded. This hopefully eliminates most illustrations and stray page markings from consideration.

The end result is all words from portrait-oriented pages that do not intersect the page margin or gutter shadow, including single tokens, image captions and words found inside charts, and not including inline compartments as these are generally merged together and filtered out. The author is not prepared to verify its completeness. Certainly the stated words are missing, but there could easily be other losses as well. For this research it is not essential that every single word was captured.

Study 1: Words Ordered by Length

Rationale: Observation reveals that within a page of the *Codex* the same sorts of word shapes are repeated often, as though the words are cognates. Comparing them could lead to identification of morphological features.

The words are too small and intricate to submit to the same similarity metrics used on the headers. An application to order all body words on a page by length is an attempt to quickly and easily identify these similar words to determine next steps.

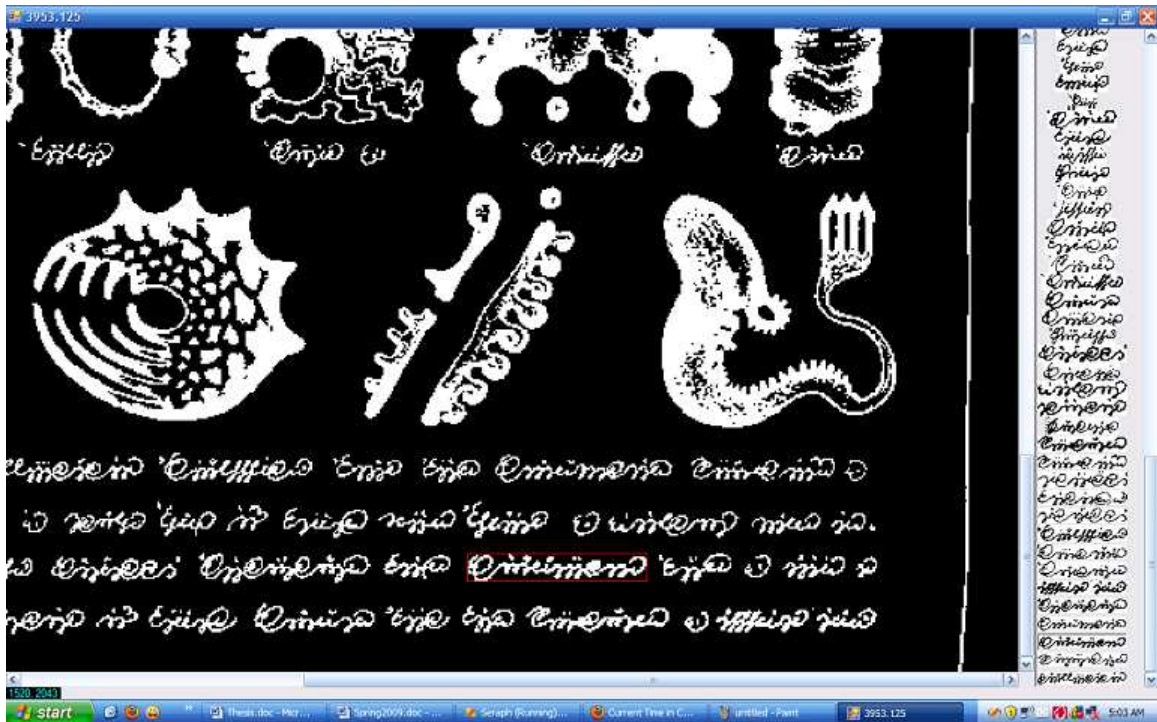


Figure 14: An application that displays all word instances on a page sorted by length, applied to page 13 of the codex.
 Selecting a word boxes its appearance in red. (Timed trials 3828, 3828, 3844ms.)



Figure 16: Near the top of the word results are extraction errors. *These are not actually words but rather pieces of illustrations. Further research could find a way to filter these out, possibly by black to white ratio.*

Study 2: Search Engine

Ordering words by length provides a good initial look at word similarities on a page. Indexing the words across multiple pages should yield more exact results on a larger area of the corpus. As an initial step toward this word spotting, a corner matching technique was used, similar to, but simpler and faster than, Rothfeder (2003), to build a search engine. Gray intensities were not used, though there is certainly room for such an endeavor in future research. Rather the evaluation considers only the positions of the corner points. For this study the Susan corner detector algorithm was used because it is included in AForge.NET. Each corner point in the target image is matched to the closest corner point in the candidate, if there is one within 6 pixels. The total match value is computed as follows: The total difference in distance for all match pairs is divided by the total number of matches to get an average distance. As in

Rothfeder, this is multiplied by the total number of corner points in the target over the number of matches, to favor candidates with more matches. A potential problem was found in that if only one match pair was found but its distance was zero, this would result in a zero value. This is prevented by forcing the distance for match pairs to always at least .01.

In a search engine application, this algorithm demonstrated validity. As in the search engine for header tokens, it reads the Windows clipboard for a target. A target word on page 13 was chosen by the author because he knew of a very similar word nearby, differing mainly in line thickness and diacritic marks. The search engine found the target word and the similar word as the first and second matches respectively. Matches share the same general word shape as the target. This word shape includes diacritic marks. Future research could be conducted to remove the diacritic marks from the comparison or to compare them separately.





















Figure 17: Demonstration of search engine for body text and close-up (bottom) of best matches, pages 10-48.

The target word is in the second-to-last row of page 13, chosen because there is known by the author to be a very similar word nearby. The red dots on the target image show the corner points. The target word and similar word appear as expected as the first two matches. Notice that matches 3 and 4, while not as similar, do share similar word shapes with the target, especially when considering the positioning of diacritic marks.

It is significant that there are only two instances of this word in the codex. The uniqueness of words is not limited to this case. While particular sequences clearly occur again and again even on the same page, and there are very short tokens which appear throughout the codex, recurrences of words seem rare at first glance.

To further explore word uniqueness in the codex and the effectiveness of the search algorithm, a search was conducted for a short word that is graphically simple compared to the other words in the codex, on the hypothesis that even this word would be rare or unique. The results can be seen on the next page. For reference, there are over 200 words on page 12 of the codex. In pages 10-65, the search engine gathered 131 hits. Out of these, 18 resemble the target word. (Several of these results were ranked below those that do not resemble the target word. This indicates that the algorithm is too simple; in the future, it should be tried with grayscale intensities as in Rothfeder.) The search engine shows that this word appears throughout the codex, debunking the hypothesis.

Table 4: Table showing search engine results for a target word, pages 10-65.
The number preceding the image is the rank assigned by the search engine. Following the image is the page number and in parentheses the coordinates on the page. Only results resembling the target word are shown. Out of 131 results found by the search engine, these 18 resemble the target. Page 60 has 6 occurrences, while 36 and 52 each have 3; 62 has 2.

1.  52 (310, 472) (target)	85.  60 (480, 1076)
2.  52 (1474, 428)	86.  36 (715, 1367)
10.  60 (273, 461)	93.  24 (856, 455)
11.  62 (560, 407)	109.  60 (1019, 660)
12.  36 (122, 507)	
36.  60 (117, 415)	
39.  32 (1118, 144)	
41.  40 (333, 105)	
50.  62(1023, 711)	
53.  52 (468, 1132)	
59.  22 (1445, 402)	
65.  60 (416, 1426)	
70.  36 (682, 1566)	
80.  60 (505, 1019)	

Based on the results above, the algorithm clearly could benefit from techniques researched by Rothfeder's group. It should especially be noted that this algorithm does not seem to sort matches well in its present state for short words. Also it does not work for words that are similar but have been drawn to different lengths, such as the following pair near the top of page 52. Further application of the algorithms explored in the word spotting papers could certainly help these problems.



Figure 18: Two tokens near the top of page 52 that are similar but are not detected by the search algorithm because they vary in length.

DISCUSSION OF RESULTS

Linguistic Observations on the Writing

Formulaic Headers

The search engine makes it easy to realize that the headers follow formulae. Since this was noted previously by Derzhanski and can be observed simply by flipping through the headers, it is not a remarkable success of the study. However, the application suite was able to add some insights. Specifically, when the type classifier yielded several types of only one token, it was found that these one-shot tokens always appear before the dash on the image pages. This could be hypothesized by a human scholar, but it would take a long time to confirm that the pre-dash tokens are unique. The classifier made this evident. Further exploring this phenomenon, using the search engine to find all dashes in the codex is an easy way to browse all of the pre-dash tokens. The results confirm that the pre-dash tokens are unique. Even between pages that seem to show closely related illustrations, the tokens do not resemble each other. One possibility is that these particular tokens are ideographs. Ideographic writing systems represent an idea with a single token and therefore require a vast number of tokens. There are languages that mix ideographic tokens with phonetic tokens (Packard 1974). However, a writing system that does not correlate orthographic representation with semantic meaning would be unmanageable. Chinese speakers recognize thousands of different ideographs, but they learn to recognize and recall them through their constituent elements (Shu 1997). Since Serafini is the only user of the language in the codex, this is not a problem for him. He can make up a new ideograph for each illustration page and assign it meaning without worrying about it being used by a wider community. It is impossible that he could have thought out all of these ideographs before beginning the book. By improvising them, even if he does consciously assign them meaning as

he writes them, Serafini is creating a huge collection of unique tokens that draw their meaning, if they have one, from their particular usage. They had might as well not be there at all. He is establishing a practice for his writing system that results in more work without anything gained, that of making up a new token for any new set of items, even if that set of items resembles one that has already been assigned a token. This would make sense if the token were used somewhere else to refer to the items, but it is not. It should also be noted that the practice of making them unique is not sustainable, unless it is confined to the codex. Eventually the writing community would begin reusing tokens, either knowingly or unknowingly. For this reason it must be scoped to the current linguistic act (book, article, discussion, etc.).

This observation shows two things. First, if he intended a language, Serafini certainly made up the language or parts of the language as he wrote. Secondly, he was more concerned with the appearance or novelty of the language than its utility; otherwise he would not have created more work for himself without any functional benefit.

Free Variation within a Type

It has been shown by tracking the sequence SFE through the codex that the type S can be written with ornate loops or without these loops. This distinction does not appear to be contextual, or it would be found in more than just the first three occurrences of this word in the codex. Rather it seems accidental (freely interchangeable). This observation is important because it suggests that other tokens that differ only by a loop ornament near an endpoint are likely to be accidental variations of the same type. This happens often in the codex.



Figure 19: Tokens in the codex headers that appear to differ only by a loop near an endpoint.

Especially since the looped versions are rare, these could be freely interchangeable variations. While classifying, the author decided to make them of the same type, leading ultimately to 53 types.

This observation is interesting because the set of variations allowed seems well defined, so it is not completely unrestricted. If an ornament is added, it looks like it can only be of a few kinds, even across tokens. If the variations were completely interchangeable, there would be no need to restrict them in such a way.

Table 5: Kinds of variation in three of the header types.

None			
Loop			
Overhang			
Extension			
Dot inside			
Dot above			
Hook			
Other combinations			

This observation is also interesting because in several cases the ornamented token and the unornamented token are found in the same header near each other. This inconsistency would seem to imply that the variation is purposeful and therefore not accidental.

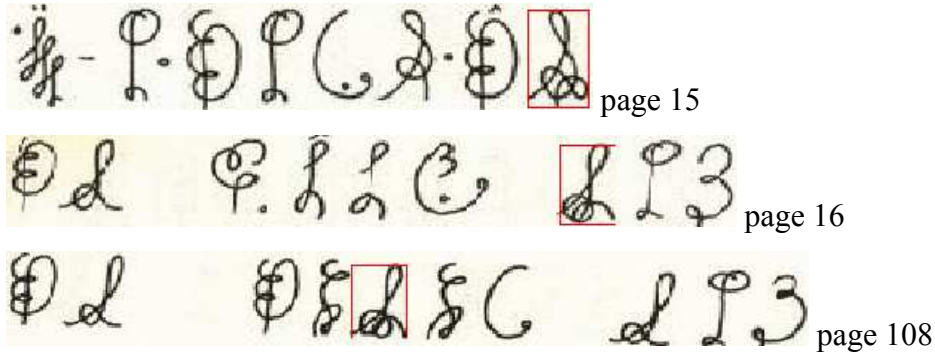


Figure 20: Ornamented and unornamented S tokens near each other on the same page.
Similar examples can be found for the other types.

Of course it is possible that the variations are both freely interchangeable and strictly defined and that the author is choosing to use what he considers a healthy mix of all variations. If so, it simply emphasizes Serafini's concern with the appearance of the language: By allowing free variation he enlivens the visual appearance, but by restricting the kinds of variation he maintains the appearance of linguistic rules.

Non-repetition of Words

While the headers observe formulae and there are certainly repeated short words (at first sight supposedly function words), there is a remarkable uniqueness of words in the codex, both in the headers and the body text. As far as the body text, it has been suggested and shown in a very limited sample that there may be several words that have similar features on a page but that are not exactly the same. This could easily be further verified by a lengthier exploration. A similar situation has been suggested for the headers: The words that are not fixed by the formulae appear not more than twice, once on a text page and once on the accompanying illustration page. No counterexamples have been found. In fact, most of the words in the headers, in addition to containing common types, also contain very uncommon types, a

phenomenon that suggests that their recurrence elsewhere is unlikely. Even those composed entirely of common types do not appear to recur.

That words in the headers are not repeated is conceivable. It would be the equivalent of a book on animals in which each page has a completely different title, such as “bears”, “cats”, “dogs”, “squid”, etc. That long words in the body text are not repeated on the same page while pieces of them are is harder to explain. The match in figure 17 could be two instances of the same word even though the diacritic is off, but this situation for words of more than four characters seems rare. Sequences recur but, as far as this study has found, not long words. They could be different forms of the word. This indicates either an inflectional language with so many inflections that they are not reused, which seems unlikely, or an agglutinative language in which one word contains a substantial amount of information conveyed through combination of several roots and morphemes. It is this combinatorial trait that keeps the words unique. If the dots following some words in the codex are taken as periods, the sentences they delineate sometimes contain many long words, and even within a sentence these long words sometimes resemble each other. Serafini appears to be reduplicating the word in a slightly different form to emphasize it or for some other grammatical or semantic purpose.

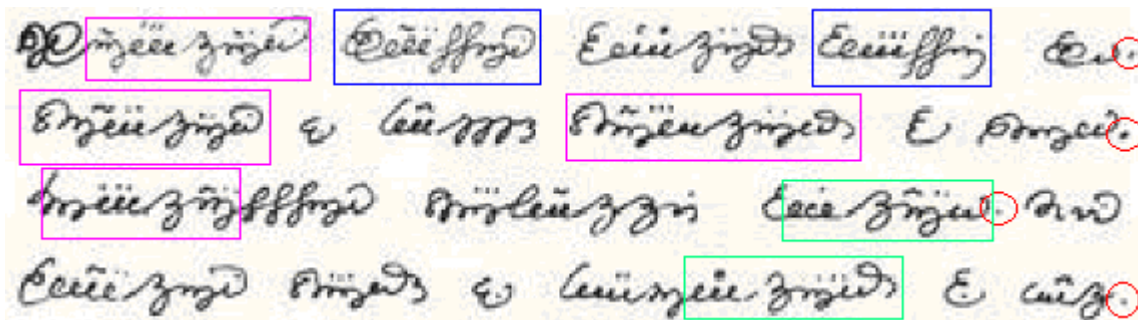


Figure 21: Four possible sentences from page 52.

Markup done in MS Paint. Periods are circled in red. Words that appear to correspond with each other are boxed in corresponding colors. Note that even within a sentence the same word may correspond with a very similar word.

Summary

None of the results discussed demonstrates that the language of the codex is not a language, but the results do demonstrate that Serafini's concern for visuals was stronger than his concern for building a functional language. Firstly, he creates a unique token for each illustration page. Secondly, he defines restricted sets of type variations that seem to fill no linguistic purpose but are freely interspersed. These techniques add no practical value but do add visual value to the writing. They do require more work on the author's part. Serafini must have felt that the decline in utility was worth an increase in visual interest. Thirdly, Serafini repeats sequences of characters without repeating exact words. Similar to the variation within character types, this is a variation within word types: it allows him to carry visual motifs throughout portions of the codex without repeating them exactly.

Methodological Observations on the Studies*Successes*

These implementations show that it is possible to study a language using a computer visually, without resorting to a transcription scheme. This finding is most significant as it accomplishes the primary aim of the research. Not only is it possible to extract and classify tokens when they are clearly written, consistent, and uninterrupted, as most of the header tokens are in the codex, but also to extract tokens plagued by the typographical challenges of the body text in just a few seconds. In this case, the algorithms used are elementary, implemented by someone with no prior experience in image recognition or extraction. Surely much more could be accomplished by a scholar more advanced in these fields.

Problems

The insufficiency of the algorithms has already been noted, particularly the searching and sorting of words in the body text, as well as certain extraction errors with the body text. While the implementations met success with the header text, other undeciphered writings will pose additional problems. On the Rongo tablets, for instance, the lines are not straight, and photos of the Linear A tablets are not always clear. Fortunately, clear, straight drawings of some of these corpora are available online, but not all. Other issues will be addressed in the conclusion.

CONCLUSION

Computer-assisted Decipherment

This study demonstrates that it is possible to study an unfamiliar writing system by a computer graphically, without using a transcription scheme. Character extraction and recognition techniques have reached a level of success that allows computer-assisted classification of unknown symbols and identification of recurring sequences, two primary sub-problems in decipherment. (It is important to remember that the algorithms used in this study are only the simplest to implement; there are many more refined alternatives that can be investigated.) Since no transcription needs to be produced by the human user, this kind of system promises less effort and less potential for error than existing systems. Because it examines the writing in its original state, it also opens the door for new insights about the writing.

The Codex Seraphinianus

As stated in the discussion of results, the only clear finding about the writing in the codex is that it is intended to be visually interesting. Serafini put time and effort into visual features that appear to serve no linguist purpose. These visual features are balanced in a way that creates patterns without continual repetition of words and symbols. Being an artist and not a linguist, Serafini may have thought that this visual balance would be the best way to create a pseudo-language. He may have wanted to spend his time working on the artistic aspect, not wasting time on the linguistic aspect which he did not feel qualified to capture. This conjecture is supported by Serafini's statement that reading the codex is meant to be a visual process, not a linguistic one (Prodi 2009). While he states that visually he thinks it mixes elements of Arabic, cuneiform and some dead languages (Manetto 2007), he never gives any opinions

about the linguistic appearance of the writing, even though clearly some could be made, such as whether he thinks it looks like an alphabet, or that words seem to recur in an inflected form. He does not seem concerned with the linguistic (or pseudo-linguistic) aspect of the writing.

The insights gained from this computer analysis could be a valuable addition to an interdisciplinary study of the codex. For instance, the features of the writing that serve no linguistic purpose or are even counter to what would be expected linguistically could be an intriguing topic for an artistic or psychological inquiry. Also, now that types of header tokens and types of variation in the header tokens have been identified, an investigation into the unconscious influences of the writing can take advantage of this information.

Areas for Further Study

There are several parts of this research that could benefit from further study, especially under the guidance of someone skilled in image analysis, particularly document and handwriting analysis. Some of them have been mentioned already and will be reiterated here along with others.

This study has relied on low cost extraction methods in regard to both time and money. The quality of extracted images, especially of the body text, suffered because of this. The gutter shadow blotted out many words. Even with extensive pre-processing, the layouts of some of the words and illustrations prevented extraction. These problems could be addressed by someone skilled in document analysis that could make in-house segmentation and extraction tools rather than rely on third-party software.

The word spotting techniques used in this study are simplistic. While the algorithms performed well on the header tokens, better results could probably be obtained for both the header and body text by considering recent advances including dynamic time warping (Rath 2003) and gray-level corner matching (Rothfeder 2003).

Because of its connected nature, the body text was studied as whole words. Without knowing the components the words cannot be segmented properly. Therefore this study seems to show that when an unfamiliar text is connected in this way the tokens must be examined, at least initially, at the word level. Once the search engine is improved to the state that words can be successfully classified (if desired), additional research could investigate some sort of search heuristic that would look at all word types and propose possible segmentations.

Only the first three chapters have been used to determine the header types. The 53 resulting types should be used as training data so that the rest of the codex headers can be classified. If future studies agree with the classification, analyses should be performed on these types. Because it is a false language, studies could begin immediately with the classification proposed here for the first three chapters, then later the next chapters, and so on, to gauge Serafini's consistency. Statistical modeling can help clarify how the false language in the codex relates to other languages. In particular, a Markov model could be used to discover the rules Serafini used to produce the headers. It is thought that during automatic processes the parietal lobe converts symbolic rules into manifest product, and the Markov model could show what those rules are like.

REFERENCES

- Alexander, Lynn (2009a). Michael Jacobson. Interview. Aug. 2009. PRATE: Full of Crow Interview Series. New York, USA: Full of Crow and FC Press.
<http://fullofcrow.com/prate/2009/08/michael-jacobson/>.
- Alexander, Lynn (2009b). Tim Gaze, *Asemic Magazine*. Interview. PRATE: Full of Crow Interview Series. New York, USA: Full of Crow and FC Press.
<http://fullofcrow.com/prate/2009/11/tim-gaze/>.
- Allen, Julie D., ed. (2009) *The Unicode standard / The Unicode consortium, version 5.2*. Mountain View, CA: Unicode, Inc.
- Arkin, Arthur M. (1981). *Sleep-talking: Psychology and psychophysiology*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Barpoutis, Angelos, Eleni Bozia and Robert Wagman (2009). A novel framework for 3D reconstruction and analysis of ancient Roman inscriptions. *Machine Vision and Applications*. Published online.
- Bekedam, D. J., G. H. A. Visser, J. J. de Vries and H. F. R. Prechtl (1985). Motor behaviour in the growth retarded fetus. *Early Human Development* 12(2): 155-165.
- Bhardwaj, Anurag, Sriranguraj Setlur and Venu Govindaraju (2009). Keyword spotting techniques for Sanskrit documents. In G. Huet, A. Kulkarni, and P. Scharf (eds.), *Sanskrit CL 2007/2008, LNAI 5402* (403-416). Berlin and Heidelberg: Springer-Verlag.
- Bodard, Gabriel, and Ryan Baumann (2009). Opportunities for epigraphy in the context of 3D digitization. Presentation at AIA 110th Annual Meeting.
- Breton, André (1972). *Manifestos of surrealism* (Richard Seaver and Helen R. Lane, trans.). Ann Arbor: The University of Michigan Press.
- Brotchie, Alastair, and Mel Gooding (2003). *Surrealist games*. Boston and London: Shambhala Press.
- Bruel, Thomas M (2009). Applying the OCRopus OCR system to scholarly Sanskrit literature. In G. Huet, A. Kulkarni, and P. Scharf (eds.), *Sanskrit CL 2007/2008, LNAI 5402* (391-402). Berlin and Heidelberg: Springer-Verlag.

- Chadwick, John (1977). Review of *Minoan Linear A* by David W. Packard. *Computers and the Humanities* 11(1): 51.
- Chadwick, John (2004). Linear B. In J. T. Hooker (eds.), *Reading the Past*. London: The British Museum Press.
- Cheriet, Mohamed, Nawwaf Kharmah, Cheng-Lin Liu, and Ching Y. Suen (2007). *Character recognition systems: A guide for students and practitioners*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Chopde, Avinash (2001). *Online interface to ITRANS*. <http://www.aczoom.com/itrans/online/>. Last updated June 12, 2001.
- Coe, Michael (1999). *Breaking the Maya code*. New York, New York: Thames & Hudson Inc.
- Cohen, Jonathan D., James L. McClelland and Kevin Dunbar (1990). On the control of automatic processes: A parallel distributed processing account of the Stroop effect. *Psychological Review* 97(3): 332-361.
- Coulson, Michael (1992). *Sanskrit: A complete course for beginners*. Chicago, Illinois: Contemporary Books.
- Cutten, George Barton (1927). *Speaking with tongues: Historically and psychologically considered*. USA: Yale University Press.
- Danielsson, P.E. (1980). Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227-248, 1980.
- Davis, Brent (2010). Linear A: hints of Minoan inflectional morphology. Presentation at AIA Annual Meeting, 2010.
- Derzhanski, I. A. (2004). Codex Seraphinianus: some observations. <http://www.math.bas.bg/~iad/serafin.html>. Last updated 2004.
- Dietrich, Arne (2003). Functional neuroanatomy of altered states of consciousness: The transient hypofrontality hypothesis. *Consciousness and Cognition* 12: 231-256.
- Fletcher, Lloyd A., and Rangachar Katsuri (1988). A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10(6): 910-918.

Flournoy, Théodore (1900). *From India to the planet Mars: A study of a case of somnambulism with glossolalia* (Daniel B. Vermilye, trans.). New York, NY: Harper & Brothers.

Guy, Jacques (2005). *The Rongorongo of Easter Island*. <http://www.rongorongo.org/>. Last updated February 8, 2005.

Hardesty, Larry (2010, June 30). Computer automatically deciphers ancient language. *MITnews*. <http://web.mit.edu/newsoffice/2010/ugaritic-barzilay-0630.html>.

Hobson, J. Allan (1995). *Sleep*. New York, New York: Scientific American Library.

Knorozov, Yuri (ed.) (1965). *Predvaritel'noe soobshchenie ob issledovanii protoindijskih tekstov*. Moscow: Institut Etnografii, Akademiya Nauk SSSR.

Kompalli, Suryaprakash (2007). A stochastic framework for font-independent Devanagari OCR. Dissertation, State University of New York at Buffalo. UMI number: 3244288.

Laberge, S. (2000). Lucid dreaming: evidence and methodology. *Behavioral and Brain Sciences* 23(6), 962-3.

Laberge, S. and H. Rheingold (1990). *Exploring the world of lucid dreaming*. New York: Ballantine.

Lee, Rob, Philip Jonathan, and Pauline Ziman (2010). Pictish symbols revealed as a written language through application of Shannon entropy. *Proceedings of the Royal Society A: Mathematical, Physical & Engineering Sciences*, March 31: 1-16.

Lenneberg, Eric H (1967). *Biological foundations of language*. New York: J. Wiley and sons.

Leymarie, F. and M. D. Levine (1992). A note on "Fast raster scan distance propagation on the discrete rectangular lattice." *CVGIP: Image Understanding*, 55: 85-94.

Limb, Charles J., and Allen R. Baum (2008). Neural substrates of spontaneous musical performance: An fMRI study of jazz improvisation. *PLoS ONE* 3(2): e1679.

Manetto, Francesco (2007). Historia de un libro raro. *El País.com*. http://www.elpais.com/psp/index.php?module=elp_pdapsp&page=elp_pda_noticia&idNoticia=20071111elpepspag_13.Tes&seccion=cul.

- Manmatha, R., Chengfeng Han and E. M. Riseman (1995). Word spotting: a new approach to indexing handwriting. Technical Report CS-UM-95-105, Computer Science Dept, University of Massachusetts at Amherst, MA.
- Manmatha, R., Chengfeng Han, E. M. Riseman and W. B. Croft (1996). Indexing handwriting using word matching. *Digital Libraries '96: 1st ACM International Conference on Digital Libraries*.
- Marslen-Wilson, William (1973). Linguistic structure and speech shadowing at very short latencies. *Nature* 244(5417): 522-523.
- Madura, Patrice Dawn (1996). Relationships among vocal jazz improvisation achievement, jazz theory knowledge, imitative ability, musical experience, creativity, and gender. *Journal of Research in Music Education* 44(3): 252-267.
- May, L. Carlyle (1956). A survey of glossolalia and related phenomena in non-Christian religions. *American Anthropologist* 58(1): 75-96.
- Meyer, Élise, Pierre Grussenmeyer, Temy Tidafi, Claude Parisel and Jean Revez (2004). Photogrammetry for the epigraphic survey in the Great Hypostyle Hall of Karnak Temple: A new approach. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 35(5): 377-382.
- Meyer, Élise, Pierre Grussenmeyer, Claude Parisel, Jean Revez and Temy Tidafi (2006). A computerized solution for the epigraphic survey in Egyptian Temples. *Journal of Archaeological Science*. 33(11): 1605-1616.
- Miller, Kirk and Jacques Guy (2008). Something wrong with Pozdniakov's basic inventory of rongorongo. Forum discussion in KRR_Study Yahoo Group, started June 25, 2008.
- Mori, Shunji, Hirobumi Nishida and Hiromitsu Yamada (1999). *Optical character recognition*. New York, NY: John Wiley & Sons, Inc.
- Moses, William Stainton (1882). *Psychography: A treatise on one of the objective forms of psychic or spiritual phenomena* (2nd ed.). London: The Psychological Press Association.
- Nettl, Bruno and Russell, Melinda (eds.) (1998). *In the course of performance: studies in the world of musical improvisation*. Chicago: University of Chicago Press.

- Nicchiotti, G. and C. Scagliola (2000). A simple and effective cursive word segmentation method. *Proc. 7th IWFHR*, Amsterdam.
- O'Neill, Daniel (2005). Etiology of the dancing plague. *InterCulture: an Interdisciplinary Journal* 2(3): 1, 7–12.
<http://interculture.fsu.edu/pdfs/oneill%20dancing%20plague.pdf>.
- Packard, David W. (1974). *Minoan Linear A*. Berkeley, California: University of California Press.
- Parkes, J. D. (1985). *Sleep and its disorders*. London: W. B. Saunders.
- Parpola, A. (1994). *Deciphering the Indus script*. Cambridge University Press, Cambridge.
- Patel, Samir S. (2010). The Indus enigma. *Archaeology* 63(2): 18, 58, 60, 65, 66.
- Pozdniakov, Konstantin and Igor Pozdniakov (2007). Rapanui writing and the Rapanui language: Preliminary results of a statistical analysis. *Forum for Anthropology and Culture* 3: 3-36.
- Prechtl, Heinz F. R. and Brian Hopkins (1986). Developmental transformations of spontaneous movements in early infancy. *Early Human Development* 14(3-4): 233-238.
- Prodi, Enrico (2009). Personal email to author. Sep. 10, 2009.
- Rao, R. P. N., N. Yadav, M. N. Vahia, H. Joglekar, R. Adhikari and I. Mahadevan (2009a). Entropic evidence for linguistic structure in the Indus script. *Science* 324(5931): 1165. Supporting online material at www.sciencemag.org/cgi/content/full/1170391/DC1.
- Rao, R. P. N., N. Yadav, M. N. Vahia, H. Joglekar, R. Adhikari and I. Mahadevan (2009b). A Markov model of the Indus script. *PNAS* 106(33): 13685-13690.
- Rodríguez Martínez, Ma. del Carmen, Ponciano Ortíz Ceballos, Michael D. Coe, Richard A. Diehl, Stephen D. Houston, Karl A. Taube, and Alfredo Delgado Calderón (2006). Oldest writing in the New World. *Science* 313(5793):1610-1614
- Rosenberg, Daniel (2000). Speaking Martian. *Cabinet Magazine 1: Invented Languages*. Winter 2000/01.

Rath, T. M. and R. Manmatha (2003). Word image matching using dynamic time warping. *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR) 2*: 521-527. Madison, WI, June 18-20, 2003.

Rosenzweig, Mark R., Arnold L. Leiman and S. Marc Breedlove (1999). *Biological psychology: An introduction to behavioral, cognitive, and clinical neuroscience*. Sunderland, Massachusetts: Sinauer Associates, Inc.

Rothfeder, J. L., S. Feng and T. M. Rath (2003). Using corner feature correspondences to rank word images by similarity. *Proc. of the Workshop on Document Image Analysis and Retrieval (DIAR)*, Madison, WI, June 21, 2003.

Saver, Jeffrey L., and John Rabin (1997). The Neural substrates of religious experience. *The Journal of Neuropsychiatry and Clinical Neurosciences* 9(3): 498-510.

Serafini, Luigi (1983). *Codex Seraphinianus* (1st American edition). New York: Abbeville Press.

Sheridan, Susan (2002). The neurological significance of children's drawing: The Scribble Hypothesis. <http://www.marksandmind.org/scribble.html>.

Sheridan, Susan (2010). Scribbles: The missing link in a theory of human language in which mothers and children play major roles. Submitted to *Medical Hypotheses Journal*. <http://www.marksandmind.org/scribbs.html>.

Shu, Hua and Richard C. Anderson (1997). Role of radical awareness in the character and word acquisition of Chinese children. *Reading Research Quarterly* 32(1): 78-89.

Silvestri, Rosalia (2007). Somnambulation, somniloquy, and sleep terrors. In Culebras, Antonio (ed.), *Sleep disorders and neurologic diseases*. New York, NY: Informa Healthcare USA, Inc.

Snyder, Benjamin, Regina Barzilay and Kevin Knight (2010). A statistic model for lost language decipherment. Presented at *Annual Meeting of the Association for Computational Linguistics*.

Solms, Mark (1997). *The neuropsychology of dreams*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Sproat, Richard (2003). Approximate string matches in the *Rongorongo* corpus. <http://compling.ai.uiuc.edu/rws/ror/>. Last updated January 11, 2003.

Sproat, Richard (2010). Ancient symbols, computational linguistics, and the reviewing practices of the general science journals. *Computational Linguistics*, 36(3).

Van der Hart, Onno, and Rutger Horst (1989). The dissociation theory of Pierre Janet. *Journal of Traumatic Stress* 2(4): 397-412.

Van Lancker, Diana (1979). Idiomatic versus literal interpretations of ditropically ambiguous sentences. Manuscript.

Visser, G. H. A., R. N. Laurini, J. I. P. de Vries, D. J. Bekedam, and H. F. R. Prechtl (1985). Abnormal motor behavior in anencephalic fetuses. *Early Human Development* 12(2): 173-182.

Wechsler, Allan (1987, July 23). SF-LOVERS digest V12 #334. Posting on SF-Lovers mailing list. <http://groups.google.com/group/rec.arts.books/msg/25e55b7771903c1d>.

Williams, J. Mark G., Andrew Mathews and Colin MacLeod (1996). The emotional Stroop task and psychopathology. *Psychological Bulletin* 120(1): 1, 3-24.

Yadav, N. (2007). Indus script: Search for grammar. Presentation first given at *The Indus Script: Problems and Prospects*, 2006. Updated Dec. 23, 2007. Available at <http://www.harappa.com/script/tata-writing/tata-indus-script-research.html>.

Yadav, N., M. N. Vahia, I. Mahadevan and H. Jogelkar (2008). A statistical approach for pattern search in Indus writing. *International Journal of Dravidian Linguistics*, 37(1): 39-52.

Yanikoglu, B. and P. A. Sandon (1998). Segmentation of off-line cursive handwriting using linear programming. *Patt. Recog.* 31 (12): 1825-1833.

Younger, John (2000). Linear A texts in phonetic transcription. <http://people.ku.edu/~jyounger/LinearA/>. Last updated 14 March 2009.