The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

Table of Contents

The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

List of Participants

Name	Institutional Affiliation
Bolt, Jon	University of South Carolina
Cream, Randall	University of South Carolina
Kumar, Ekshita	University of South Carolina
Miller, David	University of South Carolina
Waggoner, Jarrell	University of South Carolina
Wang, Song	University of South Carolina
Zhou, Jun	University of South Carolina

The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

Abstract

Our proposal for a Level-II Start-Up grant for the Sapheos project seeks to develop innovative software to analyze, represent, and collate images in the humanities. While there are an array of text-based digital projects underway that offer increasingly powerful tools for marking up, analyzing, and visualizing textual data in the humanities, image-based analysis has not received similar attention. From the project's inception, our aim has been to develop extensible open-source software that researchers across the humanities can use to link image to text in a discrete, granular fashion. Working with the NEH-funded Spenser Project, a multi-institutional Scholarly Editions project, we're developing two significant image-based software tools: (a) digital collation software that builds on and extends the work of optical methods, using transparency to "stack" and collate multiple copies simultaneously, and (b) software for automatically sectioning and identifying (x,y) coordinate pairs for parts of an image, on the paragraph, line, and word level, linking an image to the xml markup of a text in a powerful and transformative way.

The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

For many in the humanities, the book is irreplaceable as both a work of art and a material aspect of human culture. Already a hopelessly outdated technology when Augustine wrote his *Confessions* in the 4th century—tied to the metaphor of the scroll and purposely built to mimic the mechanical act of scrolling from left to right, top to bottom—the codex has survived as bearer of accident, meaning, and metaphor. More recently, digital humanists have worked to incorporate the metaphor of the book into digital devices for reading, from the British Library's Silverlight-based page turning application to Kindle's digital ink. Reading as thumbing through a codex, scrolling through page-images sequentially, remains a key interface requirement for many users, writing the metaphor into a new array of technologies.

Central to the book, then, is the ordering of a collection of pages, each separate and distinct, yet linked by the logic of the fold or the gathering. The digital humanities remains relatively under-developed in responding to this aspect of reading culture. Much of our recent focus has been concerned with developing efficient and accurate methods of text markup—while TEI is now the standard means of storing and representing textual data, the most widely used schemas for markup eschew the page as an ontological entity within the xml. The marked-up textual data contained within books dominates current methodologies, reflected by metadata, text mining and aggregation, and textual analytics, disregarding the page as anything more than a mute bearer of meaning. However, many early-modern cultural artifacts, such as Shakespeare's Quartos and Folios, Spenser's 1590 *Faerie Queene* and *Shepheard's Calendar*, and the vast majority of medieval manuscripts, cannot be represented adequately by marking up their textual data within XML tags—features such as handwriting style, textual variants, textual decoration and embellishment, and accompanying figures remain only barely conformable to the TEI standard, and the visual representation of this information is not assimilable to a word-based markup. In many cases, nothing less than the ability to interact with images of the pages will suffice.

The Center for Digital Humanities at the University of South Carolina, partnering with the Computer Vision Group in Computer Science Engineering, is developing a suite of applications that responds to this need for software that reunites text and image of text in an intuitive manner. Our Sapheos project (sapheos.org), led by Randall Cream, associate director of the CDH @ SC, unites two of these applications into a powerful apparatus for interacting with early modern books. Our most ambitious project is our digital collation software, prototyped in MATLAB and delivered as an open-source project using C code. This software builds on existing projects and methodologies to deliver both a back-end collation tool and a powerful front-end interface for interacting with large datasets of books, a common occurrence for the best-known authors of the early modern period. An equally significant component of our Sapheos project is our software for automatically sectioning and generating (x,y) coordinate pairs for page images. Written in Java and being ported to MATLAB for C compilation, this software is designed to take images of pages with existing XML markup and insert (x,y) coordinate pairs into hierarchical elements—lines, paragraphs, stanzas (line groups), and figures—to allow XSL transformations to closely associate textual and image data for users.

These two applications are linked in their ability to allow the Sapheos project to deliver image and text as coequal components of early modern books. Sapheos project software links the words on a page to the images of the words, figure tags on a page to images of the figures, and transcriptions of marginalia within the XML to images of the handwriting. Together with the collation software (see screenshots

pp. 25-29), these two tools have the potential to transform scholarship on medieval and early modern print culture.

Enhancing the Humanities through Innovation

The Sapheos project contributes greatly to innovation in the humanities in three key ways: (a) in user interface, by producing a means by which users are able to interact with displays of large image-based datasets; (b) in optical collation, which remains a time-consuming and costly affair for scholars, as well as destructive for fragile primary materials; and (c) in bringing the flexibility of xml-based data into image-centered projects. The first two innovations concern the digital collation software; the third innovation is produced by the (x,y) coordinate software.

Our collation method draws from existing best practices in the field (see Environmental Scan, page 5) in order to produce meaningful results that can be used by scholars familiar with collation in early modern textual studies. Collation is the time-consuming but necessary comparison of two witnesses (copies) of an early modern edition in order to ascertain information about the printing process. Used extensively by historians of the book, bibliographic scholars, and those interested in the material culture of print, textual collation takes on a new importance when the underlying manuscript of the text is nonexistent. For many early modern authors whose manuscripts do not survive—Shakespeare and Spenser are perhaps the best known, but there are hundreds of lesser known authors for whom the printed text is the only known authority—textual collation is important not just for reasons of material culture and print history, but also to establish the authority of the underlying text, to separate error from accident, and isolate the way the text *is* from the way the text *ought to be*.

In performing collation, researchers isolate difference as a series of binary judgments, building alteration sequentially by comparing many individual witnesses to a given "control copy" that arbitrarily fixes the text to a given state. While many of the differences between texts consist chiefly of mechanical or human error—errors in typesetting and mechanical errors common to early presses and the methods for aligning and securing type within the forms—there are more than a few instances of variance *within an edition* that simply can't be explained in terms of error. For example, in the 1590 *Faerie Queene*, there are variants with whole words inserted or deleted, lines abridged or added, sonnets relocated or re-ordered. Coupled with the lack of a stable underlying manuscript to fix the ground of the text, collational variance becomes part of the assemblage of the text, irreducibly part of the play of its intricate meaning.

In "The Notion of Variant and the Zen of Collation," from *The Myth of Print Culture,* Joseph Dane interrogates the collational impulse as only partly arising out of any uncertainty within the text itself. Instead, Dane asserts that the process of collating witnesses (copies) exists as much as a feature of the bibliographic impulse, as a sort of "archive fever" for the textual scholar. Randall McLeod, long an innovator in collation technology, agrees that intra-edition variation exists as something of a mixed bag between meaningful difference, accidental error, and undecidability. Editorial practice, from McLeod's point of view, ought to fully represent, rather than disentangle, such undecidable moments.

With intra-edition collation as an inextricable element of humanities scholarship, the innovation of the Sapheos project's computer-assisted collation is greatly clarified. While the siren call of optical collation has resulted in several other projects building competing methodologies (see Environmental

Scan, below), our methodology innovates in several key ways. An outline of our methods will demonstrate this innovation.

Like all optical collation, Sapheos works by aligning images to suggest variance. Whereas some manual collators use the mind's ability to stereoscope two visible fields, most computing methods superimpose page images to produce the same visible effect. Using a linear opacity algorithm, images of the same page are combined to form one visible page, with variants easily detectable as "fuzzy" or out of focus areas within the page (see figure 2, Appendix I, page 26). Our innovation to existing approaches is two-fold. First of all, we extend collation beyond the binary, scaling to four and five simultaneous copies in our alpha version and extensible to as many as eight copies in manual proof-of-concept testing. (see figure 1, Appendix I, page 25) Secondly, our collation method works with off-the-shelf images of witnesses. This is a significant and important innovation, since most computer-based methods encounter insurmountable obstacles when comparing images taken at different resolutions, with different optical characteristics caused by lens distortion and focal field difference, and with the infinitely variable deformity of page curvature. Instead of manually producing uniform images by manipulating early modern books and thereby risking damage to priceless artifacts, Sapheos software deforms existing images, in all of their peculiarities. Our partner in this strategy is the Computer Vision team at the University of South Carolina. Accustomed to working with bio-medical imaging applications, our collaborative approach to the problem of images of multiple copies with idiosyncratic properties by applying methodologies derived from working with images of the human body in motion. Rather than approaching difference as binary—where image x is different from image y in discrete ways—we treat difference as a feature of *each* image, deforming and aligning each image to some intermediate position generated from the set (see Work Plan, page 7). The result is the ability to "stack" or layer multiple copies, each aligned through progressive deformation to a new state. Difference is instantly and visibly apparent as an easily noticeable blur on an otherwise crystal-clear page, represented with remarkable clarity, for any image within the stack.

By bringing multiple copies into the process of collation, Sapheos produces not just a back-end for collation but also suggests a powerful user interface for interacting with archival projects with multiple copies of extremely similar materials. Still in the planning stage, our user interface will allow individuals to select witnesses to layer together into sets, combining the images to produce an assemblage that approximates unity. As a user interface, Sapheos brings dissimilarity into focus by rendering agreement mute—similar texts, when layered, merely appear as *one* text, whereas differences between the layers manifests as visible blurs, disrupting the text's singularity (see mockup, Figure 2 Appendix I page 26). As a collation backend, Sapheos is a powerful tool for textual bibliographers; as a user interface, Sapheos has the potential to allow users to meaningfully interact with large datasets in ways that highlight and refine the characteristics of the collection. As part of the user interface for interacting with large datasets, Sapheos software links the image stacking algorithms to a sophisticated page-turning applet that loads underlying xml from the pages represented, thereby serving to facilitate a variety of means of interacting with books held within digital archives.

Environmental Scan

The ubiquity of the desktop computer brought a wide-eyed optimism to humanists in the early 1980s, when the promise of automating the difficult task of collating appeared to be just around the corner. On the Rare Books and Special Collections forum on Bitnet, we can easily find archived discussions

that attest to the dream of digital collation software. Of course, these expectations proved to be unfounded. Early experiments in using a computer to collate proved more or less as unwieldy as mechanical collation, with the added difficulty of accurately processing the images needed to digitize the copies to be collated.

Currently, collation software is very well developed for text-based projects, with an array of tools such as NINE's *Juxta* and Susan Schreibman's *Versioning Machine*. These flexible applications allow users to collate differences between well-marked up texts, essentially mining the two copies for textual difference. For early modern texts, though, such software is no solution to the problem of interacting with intra-edition variation. With OCR being notoriously unreliable with early modern type layout, optical collation remains the only available method.

The utility of optical collation is well-established by the few other projects that have taken up a similar problem recently. While there are an array of image comparators from commercial or open source projects—such as Bolide Software's Image Comparer, Tigris's TortoiseDiff, Adobe's Photoshop, ImageMagick, and Virtual Lightbox by Matthew Krischenbaum and Amit Kumar—none of these software solutions is adequate to correct for the multitude of differences that are a result of two copies from an edition having very different individual histories and provenance over a 500 year span. Simply put, available images of witnesses from a common edition reveal as much difference as similarities, highlighting the multiple attempts at rebinding and trimming, the vagaries of fading and coloration, and the almost insoluble problem of page curvature from tightened book spines.

Despite the proliferation of text-based humanities work, there are two noteworthy projects which take up the issue of digital collation. The most ambitious and well known of these two projects is the HUMI (Fumi) project launched almost a decade ago by Keio University, Tokyo, Japan. Working with the British Library, the Pierpont Morgan, and other prominent partners worldwide to digitize and collate the known copies of the Gutenberg Bible, faculty of the HUMI project, led by Toshiyuki Takamiya, developed innovative methods for working with both the single-volume and double-volume instances of this famous text. As detailed in an article in the Japanese-language *Journal of Library and Information Science* 53.3 (2005), "Toward Collation with Digital Images," the HUMI project developed software to collate these 15th century texts by superimposing the images and aligning the pages to produce difference as a visible blur, in much the same way that Sapheos software layers images. There are a few significant differences, however. As detailed by project member Mari Agata, HUMI software relies on manually flattening the images of the bibles using bamboo sticks to generate conformable images. Despite the care taken by project members, we believe this approach is unduly risky to valuable holdings and relies too greatly on care and expertise. Our approach to the same problem is to automate deformation of image sets algorithmically, thereby allowing the underlying artifact to remain undisturbed by human intervention.

Another promising candidate for digital collation is software being developed by a team of researchers at McGill, Harvard, and Geneva Universities, *Aruspix*. *Aruspix* is purpose-built to work with early modern music, collating manuscript and machine-printed music texts. One of the innovations of *Aruspix* is to represent divergent witnesses by using different colors to differentiate the content of each copy. Like Sapheos and HUMI, *Aruspix* works to perform digital collation by overlaying images of texts to make difference visible for users. However, due to project idiosyncrasies, *Aruspix* performs OCR as a crucial step prior to collation. This OCR step allows the project to collate between editions,

not just within editions, but it fundamentally constrains the applicability of the project's software—musical notes are relatively easy to differentiate, whereas early modern printed books routinely use ligatures, non-standard spacing, and embellished and decorative characters. An OCR-based collation can never isolate intra-editional variants as effectively as optical collation, and so remains an unsuitable application for textual scholarship.

Sapheos software addresses these deficiencies and adds a substantial capability that enhances its innovation: collation of multiple copies. By extending the process of collation to between four and eight copies simultaneously (see Figure 1Appendix I page 25), Sapheos brings remarkable efficiency to existing models of digital collation and representation.

History and Duration of Project

Closely aligned with the Spenser Project (spenserarchive.org), with partnering faculty at institutions such as Cambridge University, Washington University St. Louis, Pennsylvania State University, and the University of Virginia, the Sapheos project benefits from a strong association with established scholars working in digital projects in the humanities. As an early-modern editorial project, the Spenser Project has long recognized the need to conduct collation of early modern texts. Beginning in 2005, project members began developing digicoll, an application written in C++ to identify textual differences between image pairs. Digicoll, led by Craige Thomas and Aaron Zeide at Washington University St. Louis, used an open source OCR library, claraocr, to perform glyph recognition and page comparison. For the reasons outlined above, an OCR solution remains unsuitable, leading to software redevelopment. Led by Randall Cream, a new solution was devised in the Fall of 2007 and project development resumed with a robust collaboration with the Computer Vision Team at South Carolina.

With an initial proof-of-concept demonstrated in Photoshop, development on the algorithms necessary to collate n-n copies began in MATLAB. Our initial alpha-type software, demonstrated in Appendix I (Figure 3 page 27), achieves its results by manually selecting the registration points for image deformation. Work on the underlying mathematic models for automating the detection and processing of registration points for a number of images continues, subject to the vagaries of funding models.

Our (x,y) coordinate software remains a much more developed software package than the collation software, yet for continued development it must be ported over to MATLAB in order to integrate and deliver the application as an opensource project written in C.

Looking ahead, we plan to continue development of Sapheos project software, subject to funding models, over the next several years. With a publicly delivered beta solution, we'll move aggressively to build a user interface that takes advantage of the flexibility of the underlying software. We anticipate the total time to a public beta to be approximately 15 months, with a stable, feature-rich application to be completed in the range of 24 to 36 months.

Work Plan

Sapheos software is jointly developed by the Center for Digital Humanities at South Carolina, led by Associate Director Randall Cream, and the Computer Vision Team, led by Song Wang. Our

development of the software is anticipated to occur over the next 15 months. Working as a team, the Computer Vision Team serves as a resource for the Sapheos project, with a collaborative work structure and many opportunities for problem solving and joint critique. The project director, Dr. Randall Cream of the CDH @ SC, will use a course release each semester to directly supervise the project and ensure work is completed in a timely fashion. In order to participate fully in the Computer Vision Team, Sapheos will sponsor a graduate student developer. We will also sponsor two undergraduate students to work on the project, developing the user interface, software testing, and documentation. Jun Zhou, Lead Developer at the CDH @ SC, will coordinate between the undergraduate and graduate developers, working closely with the Project Director to ensure that the development schedule is maintained. Our work plan to develop this tool includes

1. Software design and implementation: 34 weeks
    - Image preprocessing: 2 weeks
    - Separating printed contents and handwritten contents: 10 weeks
    - Extending finding x-y coordinates of text lines and words: 6 weeks
    - Image overlaying: 10 weeks
    - Difference detection: 2 weeks
    - Data visualization: 4 weeks
2. Software testing and revising: 15 weeks
3. Document and packaging: 5 weeks

Project Staff

The Sapheos Project is fortunate to benefit from a variety of substantial relationships. Core project staff for software development include Dr. Randall Cream, English department and Center for Digital Humanities; Dr. Song Wang, Computer Science and Computer Vision Team leader; Jun Zhou, Lead Programmer at the CDH @ SC; Jarrell Waggoner, Ph.D. student in Computer Science and key personnel, Computer Vision Team; Jon Bolt, undergraduate Computer Science Engineering major; Ekshita Kumar, undergraduate Computer Science major; and Dr. David Miller, PI, Spenser Project. Cream, Wang, Zhou, and Waggoner will meet on a weekly basis for the duration of the project, ensuring that the team deploys its various strengths in solving unanticipated hurdles. Each of these team members will work for 10 hours a week over the course of the academic year. The undergraduate students, Bolt and Kumar, will devote on average 5 hours a week on the project over the course of the school year. As a textual scholar and a renaissance humanist, Miller will serve as a valuable resource or discussing and identifying difficulties and solutions in collating and representing early modern texts.

Final Product and Dissemination

Our project will release all of its work as opensource code, encouraging other researchers to use our work, benefit from our investment of resources, and alter our code to extend the benefits of our research. We'll present our work at DH2010, Computer Vision 2010, and TEI 2010. We'll release a whitepaper at the end of the grant detailing the lessons learned in adapting Sapheos software to current problems in humanities computing.

The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

Biographies

Dr. Randall Cream, Associate Director of the Center for Digital Humanities at the University of South Carolina and Project Director, Sapheos Project. For two years, Dr. Cream served as project manager for the Spenser Project at South Carolina, an NEH-funded Scholarly Editions project. His research projects include uniting text and image to create powerful user interfaces. Recent development projects include an XML-based page turning application, an extension to the opensource project Bischen to develop a metadata-aware flash-based pan and zoom applet, and a portable extension to Zotero to facilitate research behavior modeling. Trained as a researcher in the literature and philosophy of eighteenth-century Britain, Dr. Cream will serve as Project Director and coordinate the weekly development schedule of Sapheos software.

Dr. Song Wang, Associate Professor of Computer Science Engineering at the University of South Carolina, and leader, Computer Vision Group. Dr. Wang has extensive expertise in mathematical approaches to image-based applications in a variety of fields, including medical imaging and biology. Dr. Wang's research interest include computer-assisted vision as an interdisciplinary research area, algorithmic manipulation of images, and automated similarity and difference detection in images, static and moving. Dr. Wang will directly supervise the graduate student developing the software and coordinate the Computer Vision Group with the Sapheos team.

Dr. David Miller, Carolina Distinguished Professor of English and Comparative Literature at the University of South Carolina. Dr. Miller is trained as a scholar of the English renaissance, writing on Shakespeare and Spenser. Dr. Miller is PI on the Spenser Project at South Carolina, a NEH funded Scholarly Editions project. Dr. Miller will advise the Sapheos Project on issues of bibliographic and textual scholarship, ensuring that the suite of software tools proves useful to humanists working with early-modern texts.

Jun Zhou, Lead Programmer, Center for Digital Humanities at the University of South Carolina. Ms. Zhou is trained as an application developer and lead programmer, working with a variety of machine languages to build efficient applications that work in a variety of situations. Ms. Zhou's recent development platforms include UNIX, web-based Java applets, Windows and Mac OS, and mobile platform development. Ms. Zhou is responsible for supervising and directing the undergraduate developers, working closely with the graduate student developer, and coordinating with the Computer Vision Group.

Jarrell Waggoner, a graduate student in Computer Science Engineering at the University of South Carolina. Mr. Waggoner has worked on several projects for the Computer Vision Group at SC, collaboratively developing an array of software projects, with mostly biomedical applications. Mr. Waggoner will serve as primary developer and key team member on the Computer Vision team at South Carolina. Trained in mathematic analysis of image and a skilled programmer of MATLAB, Waggoner will primarily develop the digital collation application.

Jon Bolt, an undergraduate Computer Science Engineering major and Capstone Scholar at the University of South Carolina. Mr. Bolt has worked for the Center for Digital Humanities for the past year, working in Java, XML, and AJAX. Recently, Mr. Bolt developed an extension to an opensource page turning application to link the page animations to underlying XML data, allowing users to move between copies and stay at the same place in the text. Mr. Bolt will work on application testing, port the Java code to MATLAB, and assist in the documentation and debugging.

Ekshita Kumar, an undergraduate Computer Science major at the University of South Carolina. Ms. Kumar is a talented and committed programmer in a variety of languages, most flexibly in Java. Ms. Kumar brings a gifted understanding of mathematics to the Sapheos project, and is eminently capable of original research in algorithmic manipulation of image data. Ms. Kumar will develop the (X,Y) coordinate pairs application in Java, assist in porting the code to MATLAB, and debug and test the code for the Sapheos project.

The Sapheos Project: Transparency in Multi-image Collation, Analysis, and Representation

Appendix I

Screenshots

Figure 1
Collating Four Copies at Once

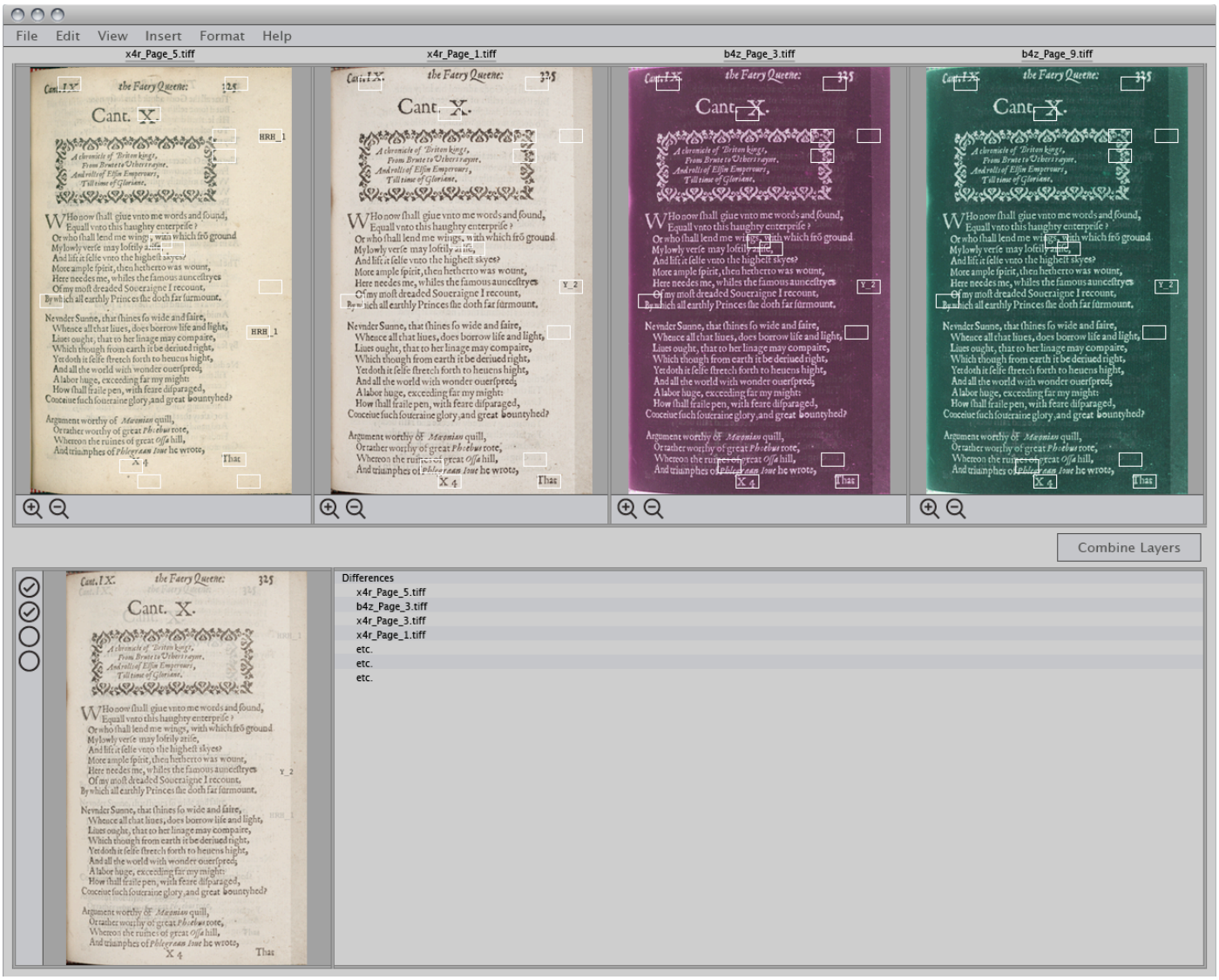

Figure 2

Layering Four Images to Detect Variants

(note visible blur at end of second line of first stanza)

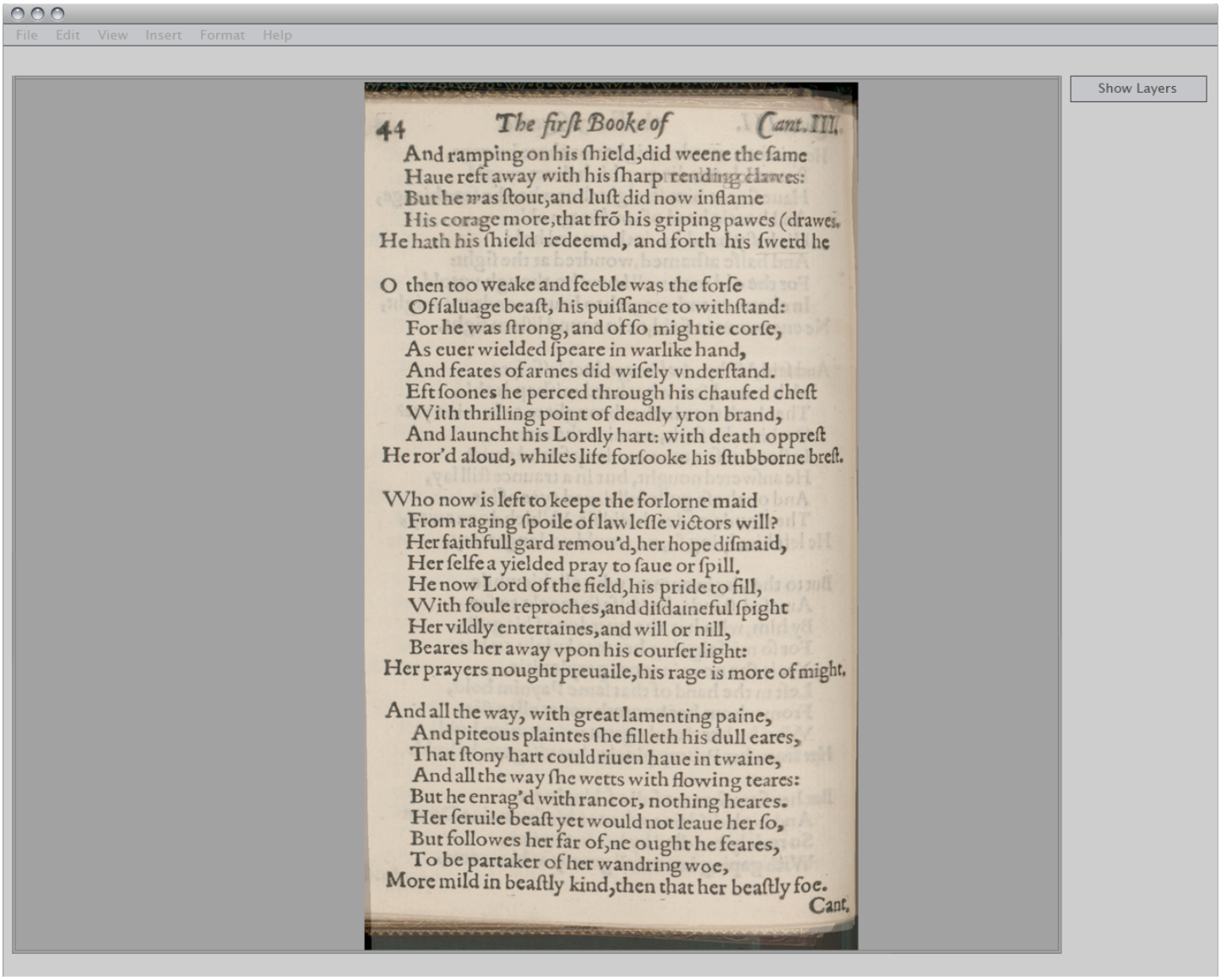

Figure 3

Generating Registration Points for Image Deformation

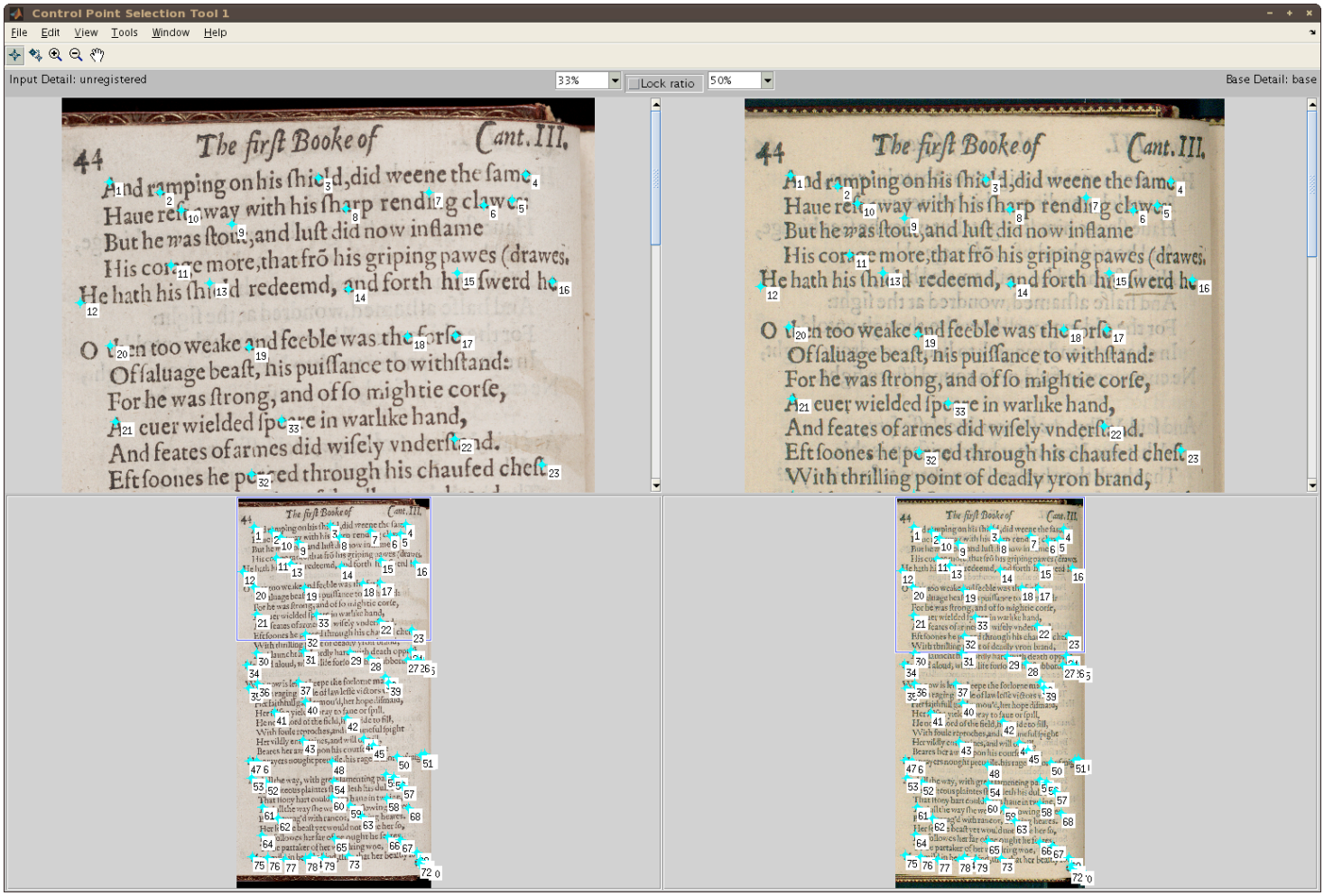

Figure 4
(X,Y) Coordinate Pairs for Page Images
Original Image

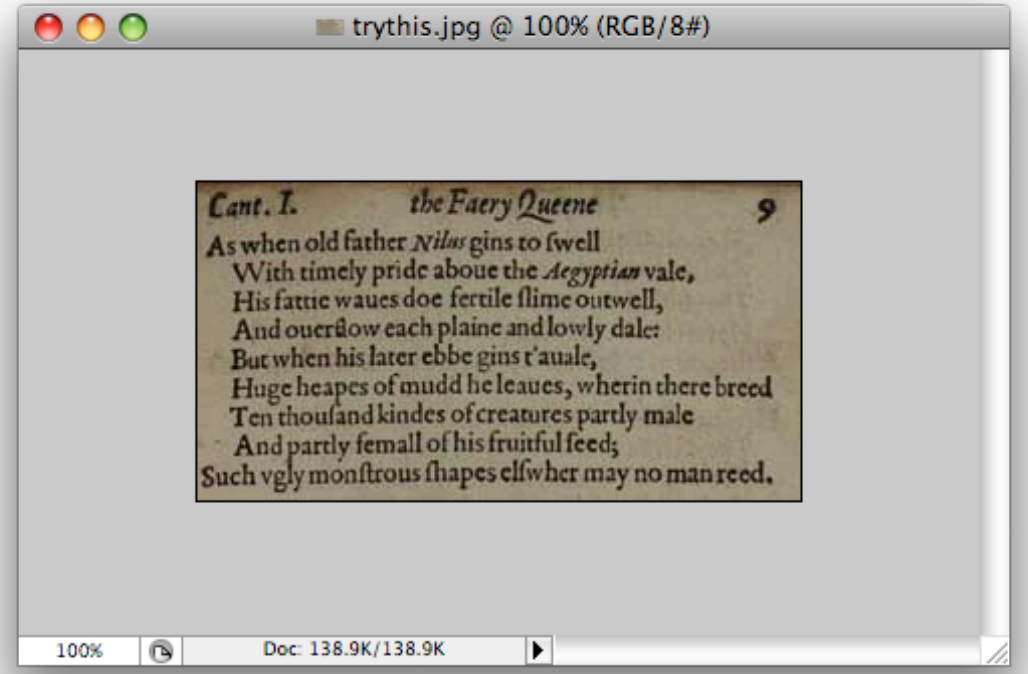

Figure 5

(X,Y) Coordinate Pairs for Page Images

B/W Image Generated to Reduce Noise

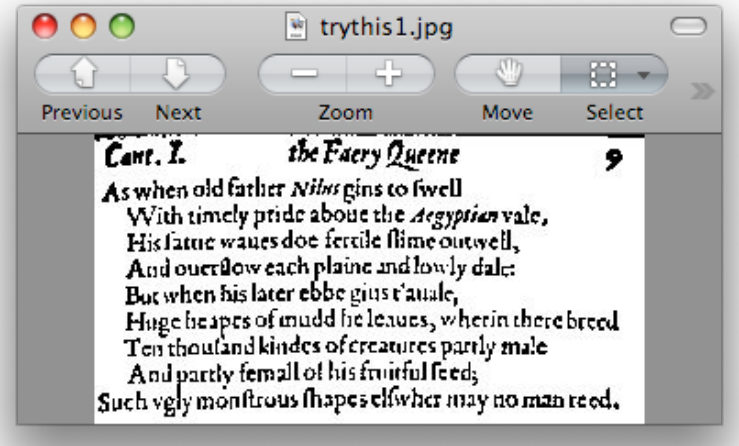

Figure 6

Source Code for (X,Y) Coordinate Pairs for Page Images