

Creating and Reading Realistic Electronic Books

➔ **Veronica Liesaputra, Ian H. Witten, and David Bainbridge**
University of Waikato

A digital library project aims to combine the look and feel of physical books with the advantages of online documents such as hyperlinks and multimedia. A lightweight open source implementation enables highly responsive page turning and works within standard Web browsers.

People like books. They're convenient and can be accessed easily and enjoyably. Their physical characteristics—their size, heft, and condition, along with the patina of page use—communicate ambient qualities of the document they represent.

In contrast, many view the experience of accessing and exploring online documents as dull and utilitarian; the appeal is to technophiles, not bibliophiles. But people also like electronic documents. They provide great added value over ink spots on paper: searching, hyperlinking, animation, multimedia. Can the advantages of both media be combined in a way that is suitable for widespread use?

The New Zealand Digital Library Project (www.nzdl.org) is experimenting with lightweight implementations of realistic electronic books with quick and highly responsive page-turning mechanisms that are usable within today's Web browsers. The approach sacrifices some 3D realism to achieve immediate interactivity with current technology. Nevertheless, the result is appealing and has many advantages over viewing documents as standard HTML or PDF files, as confirmed by a user evaluation.

BOOK ORIGINS

Figure 1 illustrates ancient devices for portable writing. The physical form of documents underwent many changes in the quest for a format that is economical, portable, and easy to use. The book form or "codex" was a technological innovation that significantly accelerated the transfer of knowledge.¹

History

Portable writing began with the clay tablet, invented by the Sumerians in 3500 BCE (Figure 1a). In 2400 BCE, Egyptians wrote on scrolls made of papyrus plants (Figure 1b). In other places, animal skins were used for parchment or vellum; elsewhere bamboo and silk served as writing surfaces. But lengthy scrolls become unwieldy—some, like the Great Harris Papyrus composed by King Ramses IV, were as long as 135 feet. A simple solution was to fold the scroll into a concertina² (Figure 1c). Folded scrolls are more compact, making them easier to handle while reading and to store afterward. Information can be readily accessed: When seeking particular passages, readers need not unroll and reroll the document. An unfolded concertina book is essentially a scroll, providing backward compatibility.

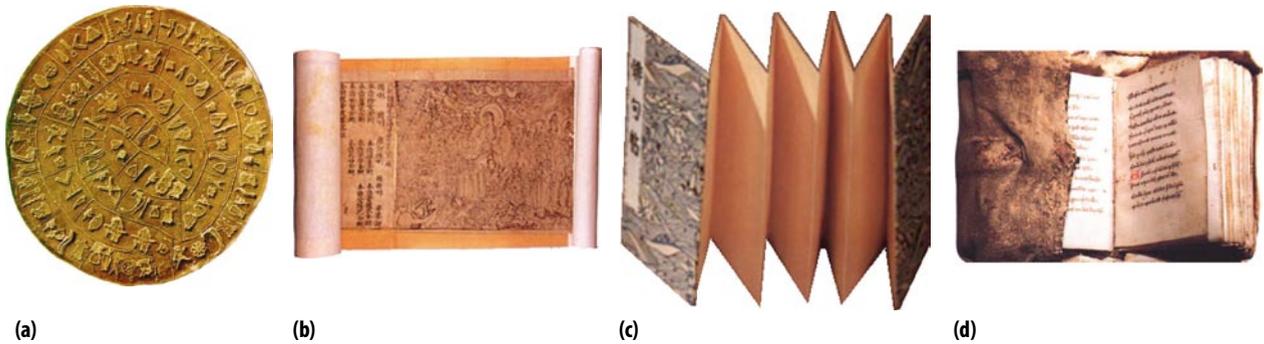


Figure 1. Ancient devices for portable writing: (a) clay tablet (www.philibiblon.com/isitabook/history), (b) scroll (<http://faculty.luther.edu/~martinka/art43/bhindex.html>), (c) concertina (<http://faculty.luther.edu/~martinka/art43/bhindex.html>), and (d) codex (www.danharlow.com/blog/index.php?tag=greek).

The codex superseded scrolls and concertinas (Figure 1d). Around 200 BCE, Greeks and Romans began to write on wax tablets backed with wood, sometimes connected with cords (resembling a ring binder). Later the Romans substituted the more durable vellum for the wood panels.

Because of its numerous virtues, the codex quickly became the preferred format for all literary works. Writers can use both sides; readers can access pages randomly. Codices are easy to read, store, carry, and search. The content is well protected, and thicker pigments can be used for decoration and illustration. The only real disadvantage is that readers can't view more than two pages at once. Books quickly became the standard document format, although concertinas are still in use today for brochures and maps.

Contemporary parallels

The evolution of computer output paralleled the book format's development. Early printers used paper rolls—scrolls; later, line printers used fanfold paper—concertinas. The latter innovation was spurred by accelerated print mechanisms, but old-timers still recall the convenience of fanfold listings over unruly rolls of paper. Indeed, paper rolls were eventually perforated so that they could be folded, boxed, and perused more easily. Today we print on pages and bind them into books.

The parallel isn't confined to print technology. Early display monitors scrolled; so do webpages, originating in the early 1990s. Adobe Reader and Microsoft Word Print Preview provide a paginated concertina-like view: Readers scroll through the document page by page using a scrollbar or Page Up and Down keys. Various auxiliary aids help readers understand the document's overall organization and length. However, not all users benefit: One study showed that even when page number and page count are displayed, few readers notice this information.³

Realistic electronic books

Research suggests that because the book metaphor is

well understood, it helps users understand a document's logical structure. Interactions such as turning pages, rotating the book, and highlighting often increase the reader's sense of engagement and fulfillment.

We have developed a lightweight technology for creating and reading realistic e-books inspired by the British Library's visionary Turning the Pages system (www.turningthepages.com). This system stores an animated sequence of images from an actual interaction with a real book; readers control the animation page by page, using gestures on a touch screen. The "Other Electronic Book Models" sidebar describes other research projects that have likewise modeled electronic documents using the codex format.

Our system is built upon the widely available Adobe Flash presentation display technology. Readers use a mouse to "grasp" the paper and sweep out the path of that point to turn the page. There is complete freedom to move the page within the physical constraints imposed by not tearing the paper, and the visual details follow instantly.

The paging method we adopted was originally proposed to handle overlapping windows⁴ and has been used for many page-turning demonstrations—for example, Sean O'Shell's dynamic page-flip effect.⁵ However, unlike earlier implementations, which show a two-page spread through which readers can move forward and backward, we simulate books by showing the page edges on both sides. This provides advantages for both visualization, such as indicating the reader's current position, and manipulation, such as moving around the book by clicking page edges.

The system incorporates several interface and interaction features: a hardcover option, chapter tabs, arbitrary zooming, ability to grasp pages anywhere near their border, defining the page-turn trajectory with the mouse, and turning pages with a simple click.

Turning pages

Readers can grasp pages anywhere along the top, right, or bottom edge—usually, but not necessarily, at a corner—by pointing there and pressing the mouse button. They

OTHER ELECTRONIC BOOK MODELS

Several research projects have modeled electronic documents using the codex format similar to the New Zealand Digital Library Project's system for creating realistic e-books.

Researchers at the University of Strathclyde in Glasgow, Scotland, created the Hyper-TextBook system,¹ which paginates documents but does not simulate a physical book. It provides linear and non-linear navigation: Readers can access different articles by clicking their link or specifying the page number in the browser's address bar, and they can search for text. A user study found that readers performed better with hyper-textbooks than with printed books, and the researchers conjectured that redesigning the appearance to resemble physical books would further improve performance.

Strathclyde researchers also developed the Visual Book,² which simulates the appearance and handling of a physical book in a 2D environment. Readers can annotate, bookmark, and randomly access articles; they can also search and edit content.

These researchers contend that the book metaphor is well accepted and understood by all, even those who feel uncomfortable with technology. Users acquire incidental memory of information location just by flicking through the pages. The visual representation helps them extract the logical structure; adding interactive features such as hyperlinking and page turning enhances the reading experience.

The 3Book³ developed at the Palo Alto Research Center in California retains the visual appearance and handling of a physical book that readers can rotate in 3D and turn pages interactively. It supports hyperlinking, annotation, and index term generation.

NZDL researchers experimented with a visually sophisticated e-book model in an earlier project.⁴ Readers enthusiastically appreciated the visualizations and interactive features provided, and some remarked that they rapidly became absorbed in the book. However, it was too slow to use for actual reading. Current work is motivated by the need for a more widely deployable page-turning technology.

References

1. F. Crestani and S.P. Ntioudis, "User Centered Evaluation of an Automatically Constructed Hyper-TextBook," *J. Educational Multimedia and Hypermedia*, vol. 11, no. 1, 2001, pp. 3-19.
2. M. Landoni, R. Wilson, and F. Gibb, "From the Visual Book to the Web Book: The Importance of Good Design," *Research and Advanced Technology for Digital Libraries*, LNCS 1923, Springer, 2000, pp. 305-314.
3. S.K. Card et al., "3Book: A Scalable 3D Virtual Book," *Proc. Symp. Computer-Human Interaction: Extended Abstracts on Human Factors in Computing Systems (CHI 04)*, ACM Press, 2004, pp. 1095-1098.
4. Y-C. Chu et al., "Realistic Books: A Bizarre Homage to an Obsolete Medium?" *Proc. 4th ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL 04)*, ACM Press, 2004, pp. 78-86.

move the mouse, and the page follows. If they release the button, the page either floats back to its original position or floats down to the turned position. An even better feeling of control is obtained using a touch panel instead of a mouse.

Page-turning styles

We employ two page-turning styles, one for rigid pages—hardback covers—and the other for flexible ones. Both employ 2D tricks rather than a proper 3D simulation. To turn a rigid page, the system applies a shear transform. The amount of shear increases as the page turns and decreases again once past the halfway point.

Turning a flexible page is harder. Imagine starting with the lower right-hand corner and creasing the page flat to reveal a triangular-shaped region of the page below, with a corresponding triangular region that shows the text on the page's reverse side, and then creating a sequence of successively larger dog-ears. This would be difficult to accomplish physically, and it would make a creased mess of the page, but it is trivial in a computer model. As the motion continues, the triangle grows and becomes a quadrilateral when it eventually subsumes the page's top right-hand corner.

Figure 2a illustrates the geometry of this *peeling* effect. The underlying model is 2D, but the system adds some visual details to simulate a smooth bend rather than a sharp crease: some shading on the bend and some shadowing just beneath it.

Peeling geometry

The system first partitions the page into three sections, shaded in Figure 3a and b: the visible portion of the page being turned, part of the reverse side of the page that the turn has made visible, and part of the following page that has been revealed.⁶

The dark-gray region formed by the crease can be triangular or quadrilateral. The light-gray area revealed has exactly the same shape, reflected in the axis formed by the crease. In these figures, the page's initial position is the rectangle ABCD, and the act of turning has moved the lower right corner C to position P. The creased region is either the triangle PRS (Figure 3a) or the quadrilateral PQRS (Figure 3b). The location of points Q, R, and S can be calculated from the position of P.

The crease is the perpendicular bisector of the line PC. The triangular dog-ear occurs when this bisector intersects the right-hand edge of the original page (point R of Figure 3a); the quadrilateral dog-ear occurs when it intersects the top edge (point R of Figure 3b). This allows R and S to be calculated. In Figure 3b, Q is obtained by drawing a line from B parallel to CP and extending it an equal distance past the point of intersection with the crease line.

The paper imposes a physical constraint on the page-turning point P: The distance PS cannot exceed CD. This constrains P to lie within the circle shown in Figure 3c; otherwise the page would tear from the book's spine. If the user moves the mouse outside the circle, say to position P_m , our implementation maps it to a point on the arc of the circle, namely P.

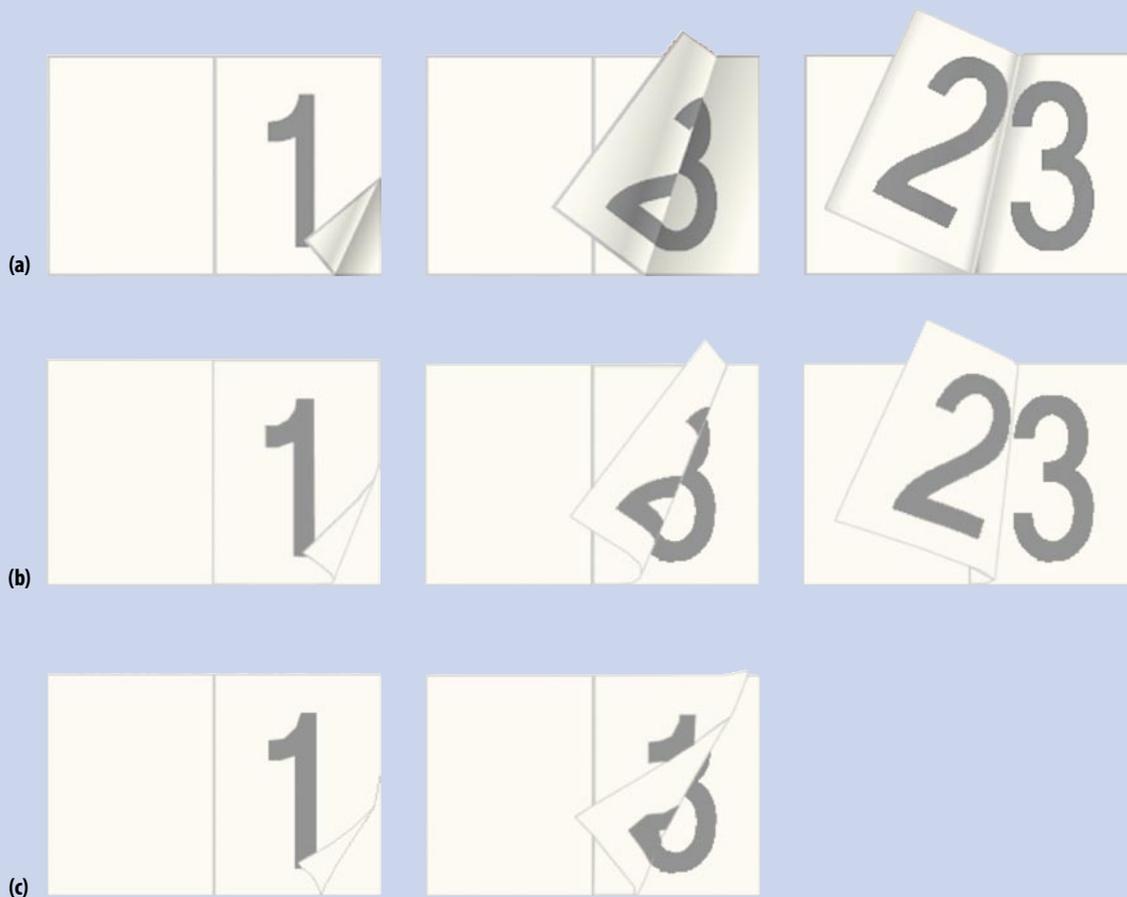


Figure 2. Page-turning models: (a) Two-dimensional geometric model, (b) 2D model enhanced by spline functions, and (c) particle-and-springs model.

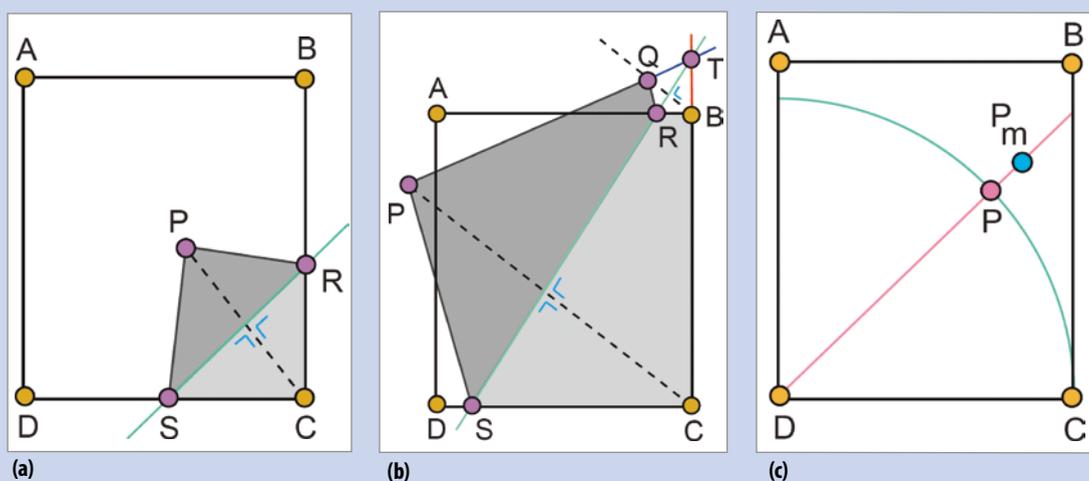


Figure 3. Peeling geometry. The region formed by the crease can be (a) triangular (PRS) or (b) quadrilateral (PQRS). (c) If the user moves the mouse outside the circle to position P_m , the system silently maps it to a point on the arc of the circle, namely P.

The system provides instant feedback by drawing the crease and filling in the regions' contents. Although this description assumes portrait rather than landscape layout and turns the lower corner of the page instead of the upper corner, our implementation extends the procedure to the general case, including turning from partway along an edge.

Peeling implementation

To implement peeling, the system maintains three images: the facing and reverse side of the page being turned, and the facing side of the page beneath. To show the creased region, it rotates the page's obverse image to position its lower corner at P and upper corner at Q (if applicable), and applies a mask with a shape corresponding to the creased region. To reveal the facing image of the page beneath, the system applies a mask shaped like the triangle RCS or quadrilateral RBCS.

The shadow effect uses a transparent bitmap image that includes shading: The system rotates the image to place its centerline along SR, masks it appropriately, and applies shading on both sides of the crease. It also applies a small shadow to the top of the visible edge of the page being turned. These subtle effects enhance the perception that the page is being turned in 3D. In addition, our implementation can make pages slightly transparent so that readers see a hint of the page underneath.

Image manipulation

Our implementation manipulates all images very speedily within Flash. The page follows mouse gestures instantaneously on ordinary computers and screen sizes. The system loads pages on demand rather than all at once before reading can commence. At any point, there are four images: the verso (left-hand) page, the recto (right-hand) one, the obverse image of the recto page, and the page beneath. When the reader turns a single page, the system loads two new images; when the reader accesses a random page, it loads four images. The system caches pages locally and only sends them once.

Adobe Acrobat Reader (from version 7) also loads pages incrementally, but more slowly: Tests revealed an initial load time of 3.3 ± 0.1 seconds (including browser startup) for realistic books versus 5.8 ± 0.1 seconds for Adobe Reader v8, while noncached pages load in 4.6 ± 0.8 seconds and 6.5 ± 0.4 seconds, respectively. Times vary slightly depending on book size.

Alternative page-turning models

A visual shortcoming of the peeling effect is that the creased portion PRS, being an exact reflection of the revealed portion CRS (in the quadrilateral case, PQRS is exactly the same as CBRS), has the same area. However, real paper extends upward, foreshortening the creased

area. This can be simulated by adding a z-value to P, Q, R, and S and utilizing a spline function to soften the bend, as Figure 2b shows, making the creased area smaller than the revealed area. A simple heuristic can be used to assign P's height, and from that, suitable values for Q, R, and S can readily be interpolated.

Careful comparison of Figures 2a and b shows that the latter definitely looks more realistic. However, it exacts a high price: 3D simulation necessitates dividing the paper into a grid, increasing the complexity in proportion to the number of grid elements (the square of the paper size). Moreover, the result is difficult to implement as a Flash animation; more suitable would be a stronger programming environment such as a Java applet, which would incur a far greater start-up delay.

There are other, more accurate, page-turning models.⁷

Geometric approaches wrap the page around a notional cylinder or cone and alter its axis and radius in a way that follows the user's mouse.

Particle-based simulations divide the material into a grid of small elements connected by notional springs that mimic the material's elasticity. This model balances the internal forces applied and exerted by each particle against external forces—gravity, the force exerted by the book's spine, and the force that the user exerts to turn the page—to determine the paper's deformation at any given time.

Finite-element methods divide the material into a similar grid, but of patches rather than particles (assuming that the paper has constant thickness). The forces on each patch—including the external forces listed above, plus those exerted by neighboring patches—produce stress resulting in strain, that is, deformation. The relation between stress and strain is given by a matrix for which the components depend on Young's modulus, Poisson's ratio, and the paper's shear correction factor.

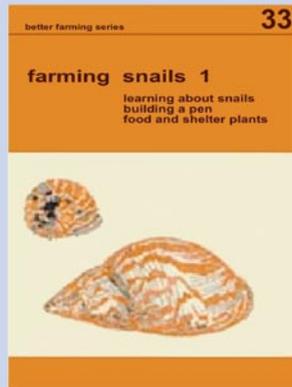
We have implemented and tested all these models, although their computational complexity precludes real-time use—at least on ordinary workstations. For example, Figure 2c shows the result of the particle-and-springs model. From this viewpoint, the model shows little improvement over the simple spline trick illustrated in Figure 2b; however, the partly turned page would appear authentic when viewed from other angles.

REALISTIC BOOK INTERACTION

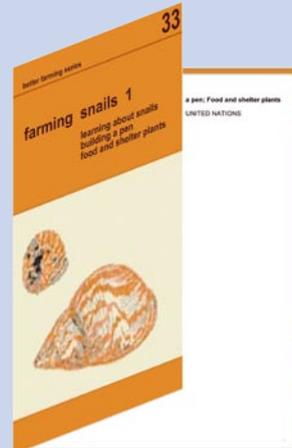
The books our system produces appear within a web-page. The basic action during reading is to turn the pages one by one, but other interactive features are present. Even the extended features can be understood with very little training.

Reading books

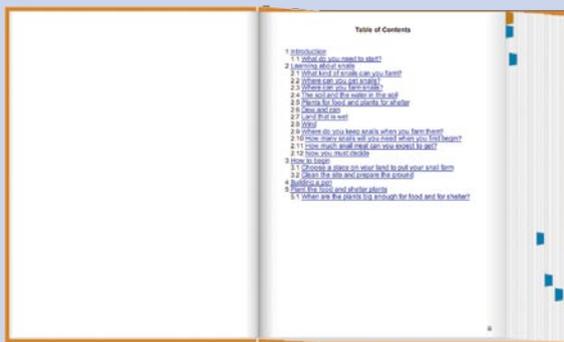
Figure 4 illustrates the user interface. Books consist of a cover (Figures 4a-b), a title page, a table of contents (Figures



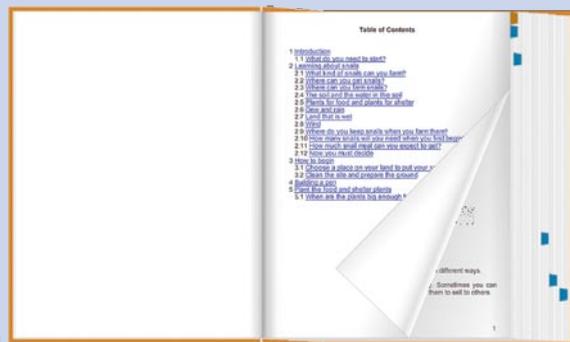
(a)



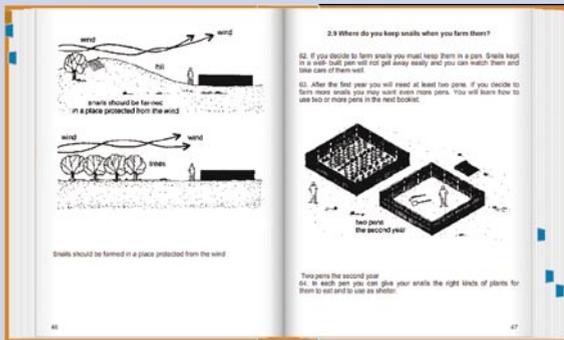
(b)



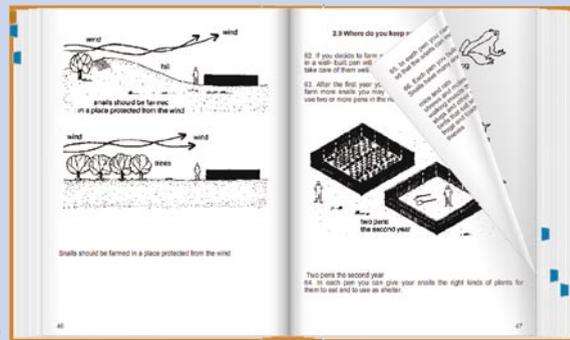
(c)



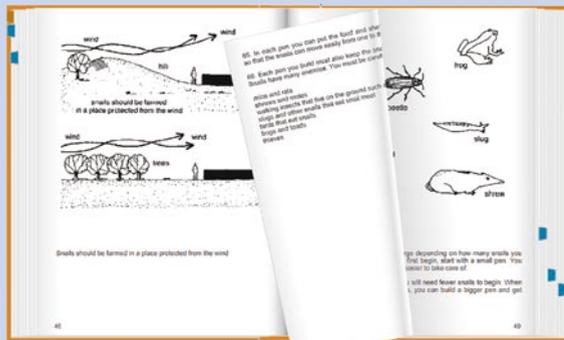
(d)



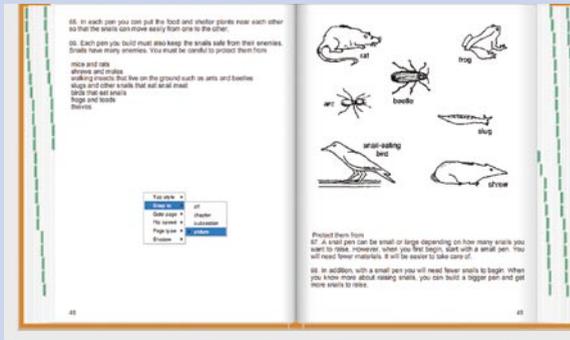
(e)



(f)



(g)



(h)

Figure 4. User interface: (a-b) opening the front cover, (c-d) turning the contents page, (e-g) turning another page, and (h) right-click menu and control bar.

4c-d), and the main text (Figures 4e-h). Covers can be hard (rigid) or soft (flexible); some physical books—particularly ones for young children—also have rigid pages. Sections begin on a new page and are paginated into physical pages. Contents entries are hyperlinked; clicking opens the book accordingly. Colored tabs mark chapters.

The reader turns pages by grasping with the mouse or simply clicking to turn them automatically. When the book is open, the cover's inside border is visible, and the reader can close it in either direction by clicking this.

Some books have tabs protruding from the page edges to mark the beginning of each section. Hovering over the tab reveals the section title as popup text. If the reader clicks a tab, the book opens to the beginning of that section. If the reader skips just a few pages, they turn one by one as if by an invisible hand; otherwise the book opens immediately at that page.



A page can have any number of different versions, perhaps with and without annotations; keys let the reader control what to view.

The right-hand mouse button opens a menu that controls specific aspects of the book simulation. Readers can determine the speed and style of automatic page-turning employed for internal hyperlinks. They can select hard or soft covers (and pages), alter the speed at which pages flip automatically, adjust the shadow effect, and view the source file from which the book was produced. Mousing page edges can open the book to any page or gravitate to the nearest tabbed page. Readers can choose whether to show tabs or not, and they can determine the style and color; they can also switch the tabs to mark pictures instead of chapter openings (as in Figure 4h) to facilitate browsing of illustrations. A control bar that optionally appears beneath the book lets readers access pages by number, magnify the display, and print the page.

The books can include dynamic media, transcending what is possible on paper. Both external and internal hyperlinks can be incorporated. Pages can contain sound and movies, along with player controls. A page can have any number of different versions, perhaps with and without annotations; keys let the reader control what to view. A presentation can successively reveal new objects on the page.

Displaying books

An HTML file (index.html) that can contain surrounding text or other material displays realistic books. The file includes a small JavaScript program that implements

a function into which a second HTML file (book.html) is passed and invokes the realistic book program, which is a substantial Flash application (book.swf).

The book file (book.html) defines the book itself. It includes basic metadata such as title and where the main text starts; Roman numerals refer to preceding pages. The following sequence of pages constitutes the book, along with any associated metadata—for example, section titles that pop up beside the section tabs.

Constructing books

We have constructed several demonstrations of realistic books: a children's book with brightly colored pages, a photograph album that closely mimics the real thing, simulated antique works with tooled leather covers and parchment pages, books containing video clips, books studded with external hyperlinks, and PowerPoint-style presentations.

The most powerful and flexible way to construct a realistic book is to create an assemblage of structured objects within the Flash authoring environment. This allows full control over the dynamic aspects of every page. Hyperlinks can be associated with text blocks. Sound and movies can be included, as can multiple versions of pages, where each page contains animations and objects that are revealed sequentially.

In addition, we have written an offline script that turns any PDF file into a book, stored as page image files in Flash format (.swf) that the book.swf program loads. The script generates all objects required to display the book, including book.html; users can edit this to indicate where the main text starts and add section tabs and title metadata.

Any HTML file (the file must be in XHTML format, but any HTML file can be accommodated using an offline conversion script) can serve in place of book.html. The book.swf program paginates this into a book dynamically, on the fly, as it processes the HTML file. Hyperlinks work as expected.

Finally, the system can process XML files in a format that includes a cover image and HTML text, along with internal metadata that defines section boundaries, paginating them on the fly into a book representation like that shown in Figure 4. The system automatically generates the cover page, hyperlinked contents, chapter and section tabs, and image-page tags. The New Zealand Digital Library Project uses this XML format internally, and the Greenstone digital library software (www.greenstone.org) makes it possible to display documents retrieved by a server-side script as a book via a configuration option.⁸

USER EVALUATION

We conducted a user study to quantitatively evaluate our realistic books against HTML and Adobe Reader PDF documents in terms of efficiency and effectiveness, as well as to subjectively assess the overall reading experi-

ence. We recruited 24 students aged 20 to 40 from various disciplines. Because we did not attempt to evaluate learnability and wished to bias our experiments against the new representation, we chose subjects who had used computers extensively for years and were already familiar with the HTML and PDF formats. The evaluation system is available at www.nzdl.org/books/evaluation. More details of the experimental procedure and results are available elsewhere.⁹

Methodology

In our study, we used three simple books: a children's book (*Alice's Adventures in Wonderland*), handbook (*Raising Ducks 1: How to Begin*), and history book (*The History of the United States: Told in One Syllable Words*), each about 100 pages (ranging from 80 to 140 pages in both PDF and realistic book formats) and divided into from 8 to 18 chapters (some of which have subsections). Each participant read the books once, in the same order. Participants were exposed to each of the three formats, a different one for each book—for example; some read the children's book in HTML format, the handbook in PDF format, and the history book in the realistic book format. The six possible orderings were allocated evenly to the participants: Eight participants read each book using the same document representation.

The task was to answer six multiple-choice questions for each book, such as “Which of the following would make the best title for the book?” “Who do you think this book was written for?” and “Put a check next to each statement that reflects the underlying beliefs or point of view of the book.” Half could be answered from the front matter and introductory section (like the first two examples); the remainder required searching the text. Participants were unlikely to know the answers already and often needed to browse more than one section to find them. Because the study's purpose was to evaluate the experience of browsing visually through the text, we asked participants not to use find commands where available.

HTML documents began with a table of contents. Users could click through to any section, and return by scrolling to the top or pressing the browser's back button. PDF documents were read using the standard Adobe Reader, initially in thumbnail view. Participants used the scrolling mechanism or the next and previous buttons. They could reach a page by clicking its thumbnail or entering its page number. They could switch from thumbnail to bookmark view and click through the table of contents to any section.

Readers navigated our realistic books by turning pages with the mouse, using arrow keys, or clicking next and previous buttons in the control bar beneath the book. They could go to a page by clicking its edge or entering its number in the control bar. They could access sections

Table 1. Average time, in seconds, to answer each question.

Book type	HTML	PDF	Realistic book
Children's book	137	78	74
Handbook	278	173	92
History book	305	219	107

through the table of contents link or the appropriate tab and return to the table of contents by clicking its tab.

To ensure that users could navigate freely, we had them browse a book in each format before beginning. We showed them how to switch modes in PDF and turn pages in the book viewer. They found the realistic book natural and needed no more time to familiarize themselves with it than the other formats. To avoid any differences in load or refresh time, all documents were prepared offline and fully loaded before work began. Participants evidently found the text shown in all three conditions to be easily readable, as none zoomed in or out.

After all tasks, the subjects completed a questionnaire to capture their opinions. They had to choose which formats were relevant and useful for them, easier to navigate and use, easier to locate information, more pleasant and engaging to use, and finally which they preferred overall. They were asked to explain their choices and state which features they liked and disliked, as well as suggest improvements.

Efficiency and effectiveness

Table 1 shows the average time in seconds that participants (who did not know they were being timed) took to answer each question. Averages range from just over 1 minute to 5 minutes, but there was a large variation: Times per question ranged from 3 seconds to 2 minutes for the first three questions and 1 to 14 minutes for the remaining three.

Most participants worked faster with the children's book—we observed that they referred back to the contents at most twice to answer each question, and when questioned stated that since they had already read or heard the story, they could recall approximately which section in the book was relevant to each question. Participants were new to both the other books and referred back to the contents at least three times before finding the relevant sections.

As the table shows, participants found realistic books much faster to use than the other formats. A one-way analysis of variance of the indicated times shows that the differences due to document format are statistically significant at the 5 percent level. The difference between the realistic book and its closest competitor (PDF) were significant at the 10 percent level for the children's book

and handbook, while for the history book it was significant at the 5 percent level.

An analysis of errors in answering questions revealed no significant difference between the three formats.

Reading experience

Participants did not read linearly through the documents to answer the questions. Instead, they perused the table of contents, clicked through to a section that seemed relevant, and scanned it for the answer. If they could not find it, they returned to the table of contents to seek other relevant sections. Even when they had already found a relevant passage, most subjects still checked other likely sections and compared the existing passage with the new text. They stated that it was important to know their current position in the document to return easily to previously found passages.

Most participants felt lost in the flow of information when reading HTML and, to a lesser extent, PDF documents. Some did not notice when they moved from one section to the next. They found it difficult to return to earlier passages because they were hazy about the approximate whereabouts (HTML) or forgot—or never noticed—the page number (PDF).

Images often serve as useful landmarks. In the handbook, pictures follow each paragraph. With the HTML and PDF formats, participants often used these illustrations and the contents to conjecture which sections were relevant, then found them by scrolling rapidly for an appropriate image. Images often helped them recall the location of previously noticed passages. However, the history book contained no illustrations.

Users complained about the time taken to scroll HTML text back to the beginning to revisit the table of contents. The PDF bookmark view was popular because it permitted easy return to the beginning.

Participants said they always knew where they were in the realistic books by looking at the left and right page edges. They could see where the current section ended by looking at the bookmark tabs. The act of turning a page helped them remember approximately how many pages they had read, which facilitated returning to particular passages.

Afterward, we asked participants which formats were most relevant and useful, easiest to navigate and to locate information, most pleasant and engaging, and preferable overall. Most readers commented that while HTML suffices for short documents, for long documents they prefer PDF images or realistic books because pagination helps them concentrate on searching rather than navigating.

All participants found the realistic books the most engaging, natural, and intuitive. However, around 15 per-

cent of users preferred the PDF format because of their previous experience with it. Some criticized the speed of page turning in realistic books: They wanted faster flipping through the book. Others suggested a PDF-style thumbnail preview.

Graphic page turning is a practical technique that websites can routinely offer to ordinary readers of Web documents. The method we use, a 2D visual simulation of a 3D effect, is less realistic than other page-turning implementations. However, more sophisticated schemes suffer from low speed and start-up delays. Even PDF document viewers—a far simpler piece of technology than a 3D graphical model—are a source of frustration for many readers, and there is anecdotal evidence that Google users, for example, frequently access the HTML version of a PDF file despite knowing that it will be grossly inferior in appearance.

Our method uses the Flash animation tool as the presentation device. This performs 2D rather than 3D image manipulation, but the system's highly reactive nature enhances realism, and it adds visual effects to enhance the 3D illusion. The implementation could easily be extended to provide further effects. For example, pages could show signs of wear—dirt, coffee stains, and bindings that open easily to well-thumbed pages—based on usage metadata.

We have focused on making the technology available for everyday use, allowing realistic books to be created from common formats—PDF, HTML, and a digital library—and read by standard browsers. Our evaluation revealed that users not only like the new interface but performed an information-finding browsing task more quickly using it. To facilitate more pragmatic assessment, we have made the complete system available for free download at www.nzdl.org/books and are interested to see Web users' responses. ■

Acknowledgments

We acknowledge the entire New Zealand Digital Library Project team for their unstinting work in providing an environment that makes this kind of research meaningful—and fun. This work is funded in part by Google. We thank the anonymous referees for their useful comments and suggestions.

References

1. G. Frost, "Adoption of the Codex Book: Parable of a New Reading Mode," *The Book and Paper Group Ann.*, vol. 17, 1998, pp. 1-15; <http://aic.stanford.edu/sg/bpg/annual/v17/bp17-10.html>.
2. K. Martinson, "History of the Book," 2000; <http://faculty.luther.edu/~martinka/art43/bhindex.html>.

3. K. O'Hara and A.J. Sellen, "A Comparison of Reading Paper and On-Line Documents," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 97), ACM Press, 1997, pp. 335-342; www.sigchi.org/chi97/proceedings/paper/koh.htm.
4. M. Beaudouin-Lafon, "Novel Interaction Techniques for Overlapping Windows," *Proc. 14th Ann. Symp. User Interface Software and Technology* (UIST 01), ACM Press, 2001, pp. 153-154.
5. S. O'Shell, "Page Flip," PiXELWiT, 11 Apr. 2007; www.pixelwit.com/blog/page-flip.
6. S. Bhangal, "The Page Turn Effect in Flash MX," 3 Sept. 2004, O'Reilly Network; www.oreillynet.com/pub/a/javascript/2004/09/03/flashhacks.html.
7. V. Liesaputra and I.H. Witten, "Computer Graphics Techniques for Modeling Page Turning," working paper 08/2007, Dept. Computer Science, Univ. of Waikato, New Zealand, 2007; www.cs.waikato.ac.nz/pubs/wp/2007/uow-cs-wp-2007-08.pdf.
8. V. Liesaputra, I.H. Witten, and D. Bainbridge, "Lightweight Realistic Books: The Greenstone Connection," *Proc. 7th ACM/IEEE-CS Joint Conf. Digital Libraries* (JCDL 07), ACM Press, 2007, p. 502.
9. V. Liesaputra and I.H. Witten, "Seeking Information in Realistic Books: A User Study," *Proc. 8th ACM/IEEE-CS Joint Conf. Digital Libraries* (JCDL 08), ACM Press, 2008, pp. 29-38.

Veronica Liesaputra is a PhD student in the Computer Science Department at the University of Waikato, Hamilton, New Zealand. Her research focuses on realistic electronic books. Liesaputra received an Honours degree in computer science from the University of Waikato. Contact her at vl6@cs.waikato.ac.nz.

Ian H. Witten is a professor of computer science at the University of Waikato, where he also directs the New Zealand Digital Library Project. His research interests include information retrieval, machine learning, text compression, and programming by demonstration. Witten received a PhD in electrical engineering from the University of Essex, Colchester, Essex, UK. He is a Fellow of the ACM and the Royal Society of New Zealand and a member of the IEEE. Contact him at ihw@cs.waikato.ac.nz.

David Bainbridge is a senior lecturer in computer science at the University of Waikato. His research interests include digital libraries, document image analysis, multimedia content analysis, and music information retrieval. Bainbridge received a PhD from the University of Canterbury, Christchurch, New Zealand. He is a member of the ACM. Contact him at davidb@cs.waikato.ac.nz.