SCAN YOUR LIFE:
Integrating OCR into your Personal Haystack!

by
Adam Holt
Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of
Masters of Engineering in Computer Science and Engineering
and
Bachelor of Science in Computer Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2000

Author . . . .
Department of
Electrical Engineering and Computer Science
September 11, 2000

Certified by . . . .
David R. Karger
Associate Professor
Thesis Supervisor

Certified by . . . .
Lynn Andrea Stein
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

SCAN YOUR LIFE:

Integrating OCR into your Personal Haystack!

by

Adam Holt

Submitted to the Department of
Electrical Engineering and Computer Science
on September 11, 2000, in partial fulfillment of the
requirements for the degrees of
Masters of Engineering in Computer Science and Engineering
and
Bachelor of Science in Computer Science and Engineering

Abstract

I built a self-serve OCR station where anybody can scan in documents at high-speed – a public yet private ATM that accepts document deposits of a wider assortment than just checks. Depending on whether you scan a business card, an article or your entire filing cabinet, CPU-intensive recognition continues after you leave the station, and you are emailed options for secure web pickup. Users of MIT's Haystack personal repositories can even do "1-click" merging of offline literary artifacts into their online lives.

The paperless pipe dream may never happen, but cheap digital optics and a mundane 40-year old technology (OCR) are converging to change the game. The mindless convenience of my $6000 kiosk suggests OCR will become a regulated munition* in the coming intellectual property and privacy wars. As OCR proliferates into cheap PDA's, neither publisher nor individual may ever again rely on humanity's oldest form of copy protection: paper.

(*) The Digital Millenium Copyright Act (1998) bans technology that circumvents copyright locks.

Thesis Supervisor: David R. Karger
Title: Associate Professor

Thesis Supervisor: Lynn Andrea Stein
Title: Associate Professor

Acknowledgments

THANKS!

Francois Labonte (design and PERL consulting)

Dan the Man @ ctscomputer.com (hardware support)

Ric Holt (all around fatherly advice)

Paul Quinn (literary historian and human search engine)

Adam Smith (UI/process design)

Abhi Shelat (chief photographer and scanner stress-tester)

Ambreen Amjad (legal dept, who pushed me into it)

Ian Lai (untouchable Java ace)

Raquel Faradji (inexhaustible Mexican jumping bean)

This project could not have happened without systems integration support from the soldiers at techsquare.com and www.ai.mit.edu/sysadmin:

Greg Shomo (design review and foreign languages stress-tester)

Toby Knudsen (security review and open sourcerer)

Jon Proulx (security design and Samba wizardy)

Aaron McKinnon (NT security superguru)

Leigh Heyman (wisely paranoid provocateur)

William Ang (all-around security whiz)

Contents

Chapter 1

Introduction

MIT's Haystack [73] is a personal information broker that emphasizes information retrieval (IR) within a user's own collection of electronic materials. Haystack assists users in aggregating from sources such as their email, their web-browsing and their electronic files. Haystack can adaptively manage a user's extensive personal repository, adding information and linkages that might be useful, even interactively customizing to the user over time [62, Adar99].

Utilities like Haystack are badly missing in this era where upper and middle class America have converted the majority of their correspondence and library activities into electronic form. Meanwhile, these same wired citizens also collect masses of paper, often for reasons of legal security and privacy [68, Greenwood99]. Individuals' digital corpora often bear little relation to their analog corpora. Yet in neither case do citizens have modern retrieval or data-mining tools to give them awareness and control over their personal effects.

If Haystack solves the former problem (a form of personal electronic empowerment), this thesis will contribute to the latter (personal papers empowerment). This thesis describes "demOCRacy," an easy-to-use prototype system I built that is making paper electronically accessible to many users today. Optionally the system incorporates materials directly into users' personal Haystacks. Hundreds of documents (about 1000 pages so far) were scanned and OCR'd by a dozen different people using this equipment.

This thesis sought to build a very user-friendly bridge from your paper life to your digital life using state-of-the-art Optical Character Recognition. Specifically, I developed a fully-integrated scheme to scan, recognize and remotely archive paper into individuals' personal Haystacks.

Much like you can point Haystack at a tree of directories you want archived, you might like to "point" Haystack at a not-entirely organized stack of papers on the other side of the room, and recover much of its long-forgotten order. You can also use demOCRacy to capture and digest the hundreds of interesting papers that regularly circulate through your life, for an increasingly low cost. Whether processing paper by batch or on-demand, conventional Haystack services will increasingly introduce order into your vast "digital filing cabinet."

Users view demOCRacy as a networked kiosk, perhaps in the office printer room or an airport lounge. They may approach the scanning booth with any number of documents to be scanned, perhaps 20 pages of class handouts for a sick friend or a phone bill with the contested smallprint on the back. After quickly logging in using their email address, they scan documents through a high-speed but small "copier."

Not long after, users are emailed options for secure web pickup of their newly electronified materials. To this day, scanning and OCR often remain as much art as science. To support these (optional) human aspects of quality control, demOCRacy supports web-based document reviewing as well as variously automatic batch pickups.

This thesis is organized into 6 substantive chapters:

Chapter 2 (Context) summarizes the evolution of OCR technology.

Chapter 3 (Scan/OCR Tools) explains the equipment we chose and why.

Chapter 4 (User Interface) shows how you take advantage of demOCRacy.

Chapter 5 (Architecture) details how demOCRacy was put together.

Chapter 6 (Manual) explains how to maintain demOCRacy.

Chapter 7 (Educational Value) analyzes the likely societal impact of OCR.

Skim chapters 2 and 3 if you want to understand the general constraints of the problem. Focus on chapters 4, 5 and 6 if you want to dive right into the engineering and operation details of my system. Chapter 7 probes how OCR technologies may affect intellectual property and privacy, to the extent that they are different anyway [113, Samuelson00]. Short conclusions follow in Chapter 8.

Even without demOCRacy, Haystack represents a highly personal system with incumbent security and privacy concerns. The banking (ATM) metaphor introduced in the abstract is intentional, if only to convey the increased implicit risks when people begin scanning in sensitive materials. This author concentrates on such security/policy and incumbent user-interface (UI) issues throughout.

One of the main reasons I chose this thesis topic was that well-defined interfaces on a well-defined problem prevent too-many-cooks. The independence that comes with owning idea(s) and running with them has disadvantages too, of course. Working more closely with colleagues on central data model issues would have been far more fruitful on an interpersonal level.

Chapter 2

Context

OCR is not just about software. In fact the first letter of the acronym (optical input) often represents the hardest part. To avoid others' mistakes, it is worth paying attention to longstanding data entry techniques. These market trends profoundly impact the usage and security design environment of a shared OCR system.

2.1 OCR History and Purpose

Optical flatbed scanners cost hundreds of dollars until 1997/1998, when competition caused prices to collapse, often to below $50 bundled with last year's OCR software. This democratization of flatbeds dried up the market for most handheld scanners (CompUSA used to carry many models), even forcing market grand-daddy HP to release its own $99 flatbed scanner. The only truly good reason left not to own a scanner in this era may be the valuable real estate on your (physical) desk.

America took notice: suddenly 25% of households with Internet access own a scanner [49, InfoTrends99], versus only about 8% that own a digital camera. These 10 million scanner households represent roughly the same 10 million that have added a separate computer line (of 100 million total US households). Microsoft took notice: their fall 1999 licensing of Xerox/Scansoft's OCR indicates that future incarnations of Windows and/or Office will include not only voice but also document recognition – pending US government approval, of course.

OCR is a remarkably mature technology that can recognize over 99% of words – in finely-tuned environments. For all its ongoing imperfections, the supposedly high error rates are much more a reflection of ergonomic and software integration shortcomings (very similar to the misdirected whining about digital camera resolution). Such complaints may also be a reflection of users' (conservative) herd mentality – which (eventually) smoothes technology adoption for all [78, Carr99]. For example, a decade after its introduction in 1981, more CD players were sold in a single year than all previous years combined. Technologies almost always have long gestation periods, and then suddenly experience explosive adoption curves. OCR could finally be nearing this point on the general adoption/diffusion curve.

OCR was first successfully used for forms entry in 1959. It is fast enough today that most off-the-shelf PC OCR users don't even notice that the OCR computation is in their computer rather than in their tediously slow flatbed scanner (where moving parts can take more than a minute per page). Certain high-end Xerox digital copiers now convert this perception to reality, taking over the OCR job from the computer.

Kurzweil's Reading Machine (an absolute revolution for blind people in 1976) was the first omni-font OCR, recognizing text regardless of type style. It took ten years before competitors duplicated this [20]. Accuracy today remains imperfect but should not be sneezed at: notoriously low-resolution faxes (200dpi) are very often OCRable today. Remember, the best-skilled and highest paid secretaries and stenographers make mistakes too – hence dual-pass and triple-pass keyed data entry. Such labor-intensive redundant keyboarding widely is used when the phone company refuses to release digital copies of public directories [44, Austin97].

Rescuing, or at least digitizing classical literature also keeps many Indians, Chinese and Philippinos busy as firms such as netlibrary.com race to market [56, Carvajal99]. OCR, like keyboarding, will be used as a competitive weapon no matter whether the source material happens to be (1) a paper original, (2) computer hardcopy or (3) a live computer display that restricts your ability to save.

Of course OCR speeds of even 300 cps on an old Pentium are 100 times faster than trained manual keying speeds of 3 cps (average based on regimented, heads-down data entry tasks, according to the Association for Work Process Improvement) [74, Caere]. In short, human-machine hybrids "each proofreading each other" are best today. Hypothetically of course, it may one day be possible for a computer to read marred text not only more cheaply but also more accurately than humans. Voice recognition advocates make similarly futurist speculations [82, Kurzweil99].

Forty years of Moore's Law did more than improve an already successful application. State-of-the-art OCR software goes miles beyond its namesake. Optical character recognition was once exactly that: decontextualized character by character recognition. Vendors today push acronyms such as ICR (intelligent content regognition) and IDR (intelligent document recognition) that indeed make more sense. But with OCR such an accepted noun (and verb) it may be far too late for an etymologically-correct re-christening. Different OCR-like hybrids could proliferate to become the converse of the current rage in next-day-delivery Internet printing – in a world of where "files" and "mail" exist equally online and off.

Today much OCR remains close to its roots: mundane forms and records capture, by the insurance and legal companies that drove the development of OCR almost half a century ago. But Boston-based Xerox/Scansoft's TextBridge Pro goes so far as to generate HTML that captures much paper document structure, starting with bolding, columns and tables. Adobe Capture 3.0 recognizes web addresses, email addresses and tables of content, making them all instantly clickable. Isn't it time to begin unifying two of the biggest repositories in your life today (all your paper with all your e-files/email)?

Such "reality merge" would allow you to rapidly search all avenues of your life's info-artifacts, physical and virtual, chipping away at Nicholas Negroponte's Berlin Wall between atoms and bits [96, Negroponte95]. This convergence between physical and virtual is precisely the topic of MIT's Technology Day 2000 [38]. In fact, using OCR to enhance information accessibility has been a dream since the earliest years of OCR development. How many times have you been stuck, wondering "Where on earth did I put that illuminating article I read only just last week?"

In contrast, on the electronic side of the railroad tracks, MIT's Haystack [73] provides an agent-like attempt to deliver more powerful, more individualized knowledge access in this era of extreme information overload. Haystack does this with a growing array of services that interlink your files and web pages based on increasingly sophisticated feedback-driven analyses of their contents and nature. Though lexical/semantic understanding is not part of Haystack today, its highly personalized searching represents a promising avenue towards breaking through the logjam of today's generic mass-market search engines.

In contrast, even the simplest keyword search is something profoundly missing from papers and books – unless the authors got carried away with their indexing. Given the Internet is often compared to a library with all the books dumped on the floor, we wholeheartedly concede that one big messy pile on the library floor may be no better than two (one physical and one electronic). But now your personalized Haystack can go further, imposing rich internal structure around all your files, online and off.

Our prototype system (demOCRacy) attempts to deliver convenient and seamless popular OCR, alongside simple integration with the Haystack information retrieval system. So we ask: is professional OCR another rich but elite technology waiting to emerge from gestation? One aspect of this thesis was to explore that possibility, which is touched upon throughout the rest of this chapter.

2.2 ePaper Contradictions

Old-fashioned (paper) fiber use continues quite resiliently against the onslaught of modern fiber (optics). Consider Business Week's 1975 prediction that the "paperless office" was just around the corner [46]. Twenty-five years later, office workers apparently consume 100 pounds more paper per annum than they did back then [47, BosGlobe00]. Document imaging systems for corporate back offices were all the rage in the 1980s yet fell on their face, again, largely because they were proprietary and as a consequence did not integrate well with other information systems, according to [17, HP99]. In the past decade, paper consumption has apparently grown from

87 million to 99 million tons a year [47, BosGlobe00] - more often for single-use and disposable rather than archival materials.

Is now really the time for the paperless pipe dream to rise like a phoenix once again? Even if all documents become electronic, how can they possibly be managed? This is more than an idle fear when the most well-funded document manager on the planet (the CIA) is under investigation for serious shortcomings in the recordkeeping practices of its own "knowledge management repository." [120, Verton00]

Yet every year, web pages are read on-screen more and more, and the "save a tree" lifestyle holds even more true with email (though it is common to print long emails). An experiment at the most populous campus in the USA (utexas.edu at Austin) very unexpectedly showed that people will not only read, but demand portable e-books, even with only 6000 titles available [56, Carvajal99]. Likewise business and consumer billing is slowly but surely moving online across entire sectors of the economy, as digital signature laws are rushed into place to sanction online transactions by re-apportioning liabilities between interested parties.

But the legal validity of digitally and/or optically stored documents took over a decade to emerge and still varies tremendously from state to state, not to mention nation to nation. The legal uncertainty of electronic documents has long restricted scanning and OCR to specialty niches. However as digital optics enter the consumer domain, the dam could burst and the law would suddenly be playing catchup with OCR technology. Technologies such as demOCRacy must take an affirmative role in implementing rigorous security policies to protect electronic documents where the law fails.

Of course, no paperless society prognostication will save us from the inundation of junkmail litter, be it one-cent-to-print fliers, handbills, circulars, coupons or their increasing multimedia equivalents. Even if paper were to disappear, its aura would persist forever in (1) artificial paper such as eink.com (think of a paper-thin television screen) and (2) bookshelves as decorative cultural/intellectual medallions (not to mention metaphors). Paper currency will be with us for decades to come – and may we all hope toilet paper is never digitized.

Internet ideology forever turns on enhancing rights and democracy by making buried information more easily available to citizens. But historically the purchase of a book, the tuning of a radio broadcast or television channel did not necessitate the acquisition (and particularly not the signing of) a license. On the other hand modern paperless purchases, rentals and viewing of digital media, generally deprive you of rights you once had in the analog world. OCR might become an important tool towards restoring some balance to a paperless world. OCR has the potential to empower individuals with (some) control over their "collection" of personal effects – in the face of rapidly increasing legal, contractual and technological attacks on the educational "fair use" of downloaded materials. In fact, some legal scholars suggest the right to create one's own digital imaging might be tantamount to a First Amendment right [36] [58, Cohen00]. Publishers on the other hand, wish to make most client-side caching illegal. There are many intellectual freedom concerns germane to a paperless society, some of which are discussed in Chapter 7.

2.3 Bulky Cameras and Lenses

Historically it has been very difficult to achieve quality bulk-scanning of legacy documents for significantly less than $1 per page [111, Saltzer]. Much of the cost arises from exception-handling: clearing paper jams, ever-changing fonts, page styles, color schemes, etc. So despite home scanners' surging popularity, most are used only a couple times a month, and mainly for art projects. Certainly no harm there, but image management is only a secondary objective of this thesis: why else would we spend $3300 on a high-speed document scanner without color? As an aside: the professional scanner business is suddenly intent on making a big push towards color, perhaps because of the influx of color printers – but looking at our source materials, color didn't seem a cost-wise priority.

A critical observation became apparent early on: long-term scanner ergonomics are unclear as scanners truly come in all shapes and sizes. There are a half dozen different OCR pens on the market, many of which even read non-Western languages.

There are powerful handheld portable scanners (capshare.hp.com) [14], business card readers (edti.com), actual signature scanners (penware.com), OCR license plate readers (perceptics.com) [23], and on and on. Surprisingly, these are much more than fools' gadgets. It's clear after interviewing a handful of academic and commercial users that many of these are highly appropriate money-savers for their particular niche.

Said another way, the physics of HCI (human-computer interaction) is a very real problem in the scanning and OCR domain, as in others – without even to mentioning the psychology and economics inherent in HCI. Today's huge diversity of paper (and non-paper) literary inputs require a huge diversity of ergonomically and optically adapted devices. Yet ergonomics is far less of a science than one of its Webster definitions ("human engineering") implies. The never-ending diversity of materials means there will never be a one-size-fits-all OCR camera. And yet two generalist solutions stand out.

One approach is to build an anti-printer, i.e. limiting yourself to standard-size office paper, which is such an important lowest common denominator (8.5x11 inches in North America). That is what demOCRacy (our prototype system) focuses on, despite its ability to handle much other input. Multifunction digital copiers represent another example of such anti-printers. In choosing a mechanical document feeder, we set ourselves up for occasional paperpath snafus, but this automated approach was appropriate today for evaluating the pervasive handheld OCR use of tomorrow. Thankfully, our feeder proved more reliable than I imagined possible.

The versatility of scanners will inevitably be compared to that of human vision, human dexterity and human mobility. By that measure anyway, we may never invent the perfect "universal" scanner. However, the next section discusses why a "near universal" scanner may soon be coming to a camera near you.

2.4 The Digital Camera Learns OCR

This section argues that scanning could soon become ubiquitous. I discuss an intriguing way that the form-factor bottleneck discussed above may eventually be breached, giving the topic of this thesis more long term relevance. However you should skip over this section if you (wisely) prefer to live in the present.

Another generalist input device may emerge in the form of digital cameras. Far more than just the "Polaroid of the 90s," basic vision software might very well allow digital cameras to double as fast consumer scanners within the visible future. In fact, high-end consumer digital cameras (4 million pixels) are today surpassing Group 3 fax (1700x2200 for 8.5x11 paper, for a theoretical maximum of 3.74 million pixels). This is very telling because clean faxes (i.e. 200 dpi) are the generally accepted lowest common denominator for OCR.

There is no doubt mobility would radically alter the picture for scanners – the cheap CCD's (charge-coupled devices) inside digital cameras could bring scanners outside of the copier room, and outside of your office. Scanners would thenceforth be known merely as cameras, as chemical-based cameras (in all likelihood) fade from popular usage.

How soon might this happen? A new study projects digital camera revenues will hit $1.9 billion in North America this year, exceeding that generated by film cameras by 10%. The same report predicts that on a per-unit basis, digital camera sales will exceed film camera sales by 2002 [98, InfoTrends/Forbes00].

This signals a brand new front for OCR – and arguably the next tentative step in the long AI ambition to make robots that can recognize the world around them in a way similar to humans. Software from Pixid is already available that attempts to capture the contents of your office whiteboard with a single photo [24].

Indeed scanning and OCR cannot help but spread like wildfire as soon as tedious mechanical passes become unnecessary. At present, these digital hi-resolution cameras cost over $500. Unfortunately the "disposable" cams atop many PC monitors generally have pathetic resolution at best (640x480), and fail to image even the largest fonts unless you hold your document absolutely still. Worse, as any vision researcher can tell you, lens calibration (especially barrel distortion) still causes endless headaches. But already you *can* get basic results by suspending a recent model $500 digital camera off of a tripod, simply pointing it downwards at your document. For faster image throughput mount your camera on your darkroom's raiseable enlarger (where the negative normally goes) and reverse the process – much like this thesis entailed building an anti-network-printer.

Xerox (UK) announced on February 23, 2000 that it would release PageCam "in April 2000" to provide exactly this. Users can position the camera over a document, image or even a 3-D object on their desk, and then drag-and-drop interesting parts of the page into their favorite applications. Philips is the first digital camera vendor to bundle this software (with its Vesta Pro VGA PC-camera).

For all the excitement over high-density CCD's (and now fully integrated CMOS eyes which are about to substantially drive down cost) it's worth asking: will all our handwritten scribbles, diaries and all, someday never be truly private again? This could happen very shortly if cellphone companies deliver on their latest hype of integrating digital cameras into all cellphones. The Xerox-machine-in-the-sky might indeed be ever present if, as futurists profess, people one day wear cams as part of their glasses.

These are but some long-term examples of how OCR could unexpectedly (and radically) alter humanity's privacy expectations [93]. It would be negligent of demOCRacy (our prototype system) not to begin addressing document privacy concerns in its security policies. I explain our users' privacy options at the end of Chapters 4 and some reactions in Chapter 7.

2.5 Personal Caching Empowers

Archivists and digital archeologists are the ultimate packrats, never letting you throw anything away. Haystack's "information maximization" design philosophy is an example of this, see [62, Adar99]. Contemporary OCR doctrine goes yet further, wisely advocating the preservation of intermediary digital image scans – in anticipation of OCR algorithm improvements. Perhaps to later archive your handwritten marginalia, even if it's only applied on demand to the most interesting parts? Yet images always entail storage/bandwidth costs: a single face of a single 8.5x11 sheet's *uncompressed* scan varies from sub-floppy 400kB (1bit 200dpi fax) to 400MB (24bit 1200dpi art), which is most of a CD.

How far can we push Haystack's "disk is free" e-packrat philosophy? Everyone demands that disk and tape systems be cheap, available 24x7, and secure, yet this remains crucially untrue in all but rare cases. While faxes (and to some degree digital scans) are becoming legally binding, "original" paper copies continue to be preserved at great expense. This is just as much for archival assurances as for legal assurances, in an era when commercial formats change so frequently that originals tend to become inaccessible even if the bits survive. Even as digitization (would) buy you much better search accessibility.

For text anyway (OCR'd or not) there is no longer any economic reason for deletion – only legal and/or privacy reasons remain – so why not keep all of it? As the ultimate in compression algorithms, OCR has in the past been perfect for such storage-constrained users.

Clipped articles from newspapers, popular magazines, and academic journals should – and will soon – be scanned into your personal Haystack. All the better to help explore how we can each manage our info-clutter portfolios. While confining ourselves mostly to the English language and latin-based characters (though many multilingual OCR components come for free), our demOCRacy prototype system is demonstrating that OCR is an empowering personal tool approaching primetime.

Whether one experiments with a PDA [14], a desk peripheral, or a batch processing sheet-fed scanner down the hall, an important research goal should be to observe individuals and organizations moving across the e-Rubicon. I set out to watch myself and other users' OCR and document-caching habits, if only with the most informal anthropological techniques. Sociology (not to mention applied psychology) is always a dreadfully imperfect science. Yet semi-detached observation of actual users can still be a fruitful way to refine policies and practices beyond the arbitrary caprices of initial designers [100, Norman90].

Chapter 3

OCR Tools

The hidden work (perhaps the majority) of this thesis was legwork in researching product integration. Search costs haven't gone to zero even in the Internet economy. Buying parts for a prototype is in the end harder than upgrading a known solution, and I knew mistakes would be costly. This meant weeks of value comparison and lifetime costing of complex uncertain integration issues. Hence I devote an entire chapter here to scanning and OCR tools and why they were chosen against certain competitors.

The professional scanning marketplace evolves so slowly (product life cycles are an order of magnitude longer than the consumer market) that the core toolset choices should continue to illuminate similar projects for several years beyond 2000. This could hold still longer given that demOCRacy (the prototype system) was not constrained by legacy integration issues. Coherent design is more important than rushed implementation, even in an academic environment; backtracking can always take weeks out of your schedule.

Indeed, a month of planning is not enough to guarantee successful integration when manufacturers don't always adhere to their promises. Perhaps more frustrating was the inevitable cutting through of bureaucracies at MIT and at ecommerce e-tailers and of course, shipping delays. This long chapter addresses the component tools to demOCRacy up-front because (1) they frame the problem, (2) I hope to save someone else the effort, and (3) software people tend to ignore what happens outside the box (literally in this case). Those not interested in "toolsmithing," should quickly skim this chapter, for perspective only.

I strongly recommend that anyone approaching such a problem first become familiar (as I did) with low-end scanners – despite their severe speed, paperpath, image-cleanup and reliability drawbacks. While generally inappropriate for departmental/production document scanning, as you will see below, their extremely low prices provide an on-ramp to the critical design issues one faces.

Two simple pre-integrated approaches are also considered, with their several limitations. Software integration will be addressed in Chapters 5 and 6.

3.1 Document Format Wars

Regrettably, formats matter: traditional OCR outputs only text but modern OCR retains layout, links and more. Should we output to HTML, Word or PDF? The old mantra "pick your software first, then buy your hardware" was especially apt given compatibility concerns with professional hi-speed duplex scanners that cost many thousands of dollars. Working to simultaneously avoid both hardware lock-in and software lock-in was harder, given that the components to our system were high-end enough that their corporate buyers don't share post-sale newsgroup advice. So be prepared for salespeople to misrepresent and deceive: these are simply the rules of the road as researching the facts just takes time.

After considering many recent projects such as "MIT Theses Online" [19] and the newspaper industry's increasing use of PDF for OCR [107, Outing99], we initially leaned strongly towards using PDF alone. Thankfully, Tony McKinley's paper-to-web.com contained a veritable wealth of overall PDF and OCR workflow advice. Despite their increasing age, and obvious Adobe Press bias, McKinley's book [91, McKinley97] and web site lead the trend towards PDF as the software-container of choice for OCR'd documents.

3.1.1 Adobe's "Portable Document Format"

Readers with a sense of humor will shortly understand why Adobe's "Portable Document Format" might better be known as the "Proprietary Document Format." Naturally, standards are not only useful, but essential to nearly any endeavor. Their development however is rarely guided by democratic processes, and even successful user-centered standards [10, Greenwood00] often stifle innovation as they grow old. For an extensive discussion of these topics see [84, Lessig99].

This thesis cannot do justice to such worldly issues, nor can it cover the full functional history of PDF. However I overview recent developments that guided our choice of this controversial [108, Ragica97] format. Standards wars can be incredibly tedious to newcomers: if you are not visually-impaired you may not care that PDF restricts certain populations' access. So readers are advised to skip past this section if they are not interested in Adobe's "Write Once, Publish Anywhere" format.

PDF began as Adobe's latest page description language; it seeks to express the precise image of a document regardless of whether rendered on screen or printed. Its layout and appearance attempt to match the author's original whether it comes from Word, LaTeX or Photoshop. Since its birth in the mid 1990's, PDF has increasingly evolved towards HTML: a PDF file can now be random access byte-served over the web, it can now offer browser-style hyperlink navigation, etc.

PDF is the format upon which Adobe's Acrobat products are built. While PDF evolved out of a Postscript background, it is not backwards compatible. Most programmability features were removed so that file size is often smaller than Postscript by a factor of 5 times or more. Specifically, PDF is based on Level 2 PostScript, and uses a limited number of operators – and no new operators can be defined. There are no iterative constructs such as for, loop and repeat.

A PDF file is structured as a number of separate objects which may refer to each other. So a page might refer to various resource objects, and links associated with the page, as well as the actual stream of operators which draw the page. These objects are numbered and may appear anywhere in the file. Random access works using a cross-referenced table at the end of the file which says where each object is, and which object forms the root of the file.

Sadly PDF has expropriated the best parts of HTML much faster than HTML has added traditional ideas such as faithful printability, i.e. consistent page numbering and layout. So PDF is showing signs of becoming an uber-standard “natural monopoly” for e-books and literary pay-per-view across all media. Surprisingly, the Microsoft Word document format has a real competitor for the first time in years – in addition to HTML, which of course isn’t dead yet.

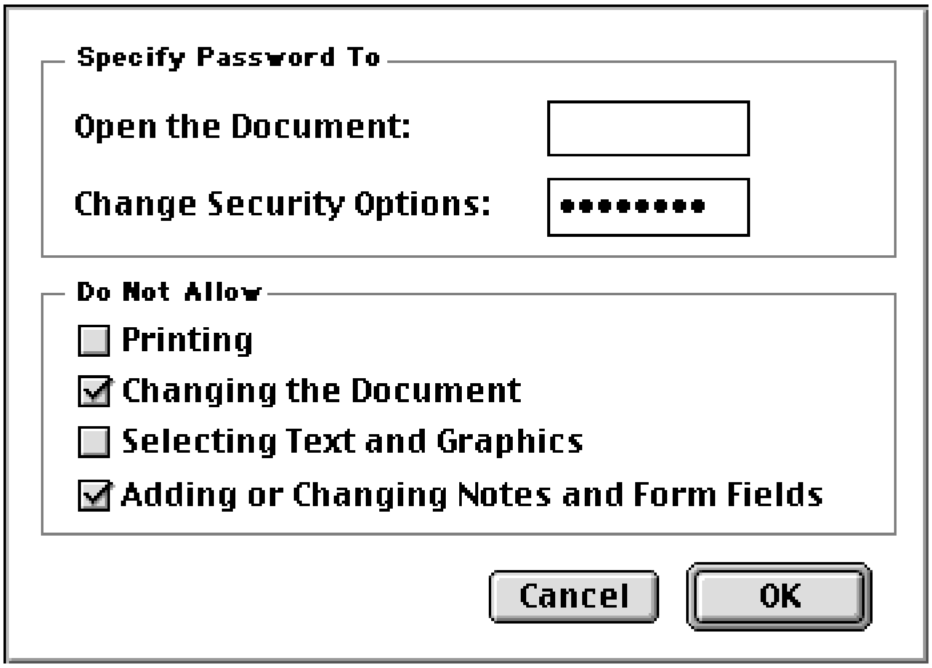
Electronic content-lock startups are blossoming everywhere, many of which are building their copyright mechanisms atop PDF, and the big guns of software are quickly following. Sadly, few in the academic computer science community understand the magnitude of the ongoing legal transitions in the status of Internet software (though individuals such as Michael Froomkin, Harold Abelson and Jonathan Zittrain continue to work extremely hard to change this [93, Froomkin] [72, 6.805]).

Of course proprietary formats like PDF and Word can still have their contents searched, eg. by keyword. Verity (unaffiliated with Adobe) is a very common corporate tool for indexing and searching intranet PDFs. Internet-wide search engines themselves are increasingly indexing the contents of PDF web pages. Unfortunately, searching PDF is not as transparent as with text and HTML (to the chagrin of many blind users who depend on browsers such as lynx [102]).

This very unfortunate situation is now gradually improving. Adobe has been accused of Microsoft-style monopolistic API-hoarding [39] but has published very open specifications to PDF 1.0, 1.1, 1.2 and most recently 1.3 in the fall of 1999. Acrobat Reader on Linux now supports searching, despite popular perceptions to the contrary. This spring (2000) Adobe at last announced its intentions to make PDF truly accessible to assistive technologies, particularly those for the blind [21] – appropriate considering many blind users depend critically on OCR.

Still, PDF can only be viewed in all its glory with Adobe Acrobat Reader, and to a lesser degree by open source clients like xpdf and ghostscript. While most web pages on the other hand can be saved using a “Save as” pull-down menu, the

Adobe Acrobat Reader (that the overwhelming majority of users depend on) blocks all saving. Not to mention modification, which is impossible unless you buy the full $250 Acrobat program. If you should be so lucky (i.e. you actually buy Adobe Acrobat 4.0), you can experiment with a rather new way of saving documents. When saving, there is an option to selectively turn off reader rights such as printing, cut and paste, modification, etc. You'll have to wait for a future Adobe release that supports page self-incineration the second you finish reading it – though companies such as disappearing.com support similar features today, for email privacy. Some Acrobat saving options appear below:



This spring (2000) Adobe took a small step [7] towards democratizing the format beyond its traditional publishing markets. Adobe now allows you to upload and convert three free documents (starting from most any other format). After your first three documents however, you are requested to pay per play. The $9.99/month service offers "unlimited" conversions to (but not away from) PDF.

PDF will soon be integrated into two significant platforms, Palm Computing's PalmOS and MacOS X, according to separate February 2000 announcements [94] [121]. PDF "will be the display technology of choice" not only on the web but also for handheld, pay-per-view e-books according to Patrick Ames, archivist and author of “Beyond Paper” [42]. In short, the controversial standard has tremendous momentum, just as portable e-book hype is exploding onto the front page, first with the release of Stephen King’s short story, and now with a torrent of daily celebrity titles.

As Microsoft and others dive into these waters [110], the format war to define the future of pay-per-view e-books is becoming especially competitive. So Adobe mustn’t have been thrilled that only hours after Stephen King’s no-can-print novelette was released in several competing formats, it was the secure PDF version that was cracked, to be subsequently published in newsgroups. Still, 400,000 copies were sold or legally downloaded to registered users for free. Nevertheless “the developments could temporarily slow the adoption of Adobe’s Portable Document Format (PDF) as a common standard for commercial eBooks.” [114, Sanders00] Through all the e-book media bonanza, Stephen King himself was surprised to learn that he could not read his own e-book because Macintosh was not supported. This is but a hint at the many other problems [67, Godwin00] concomitant with (even pre-video) e-books.

Most adopt PDF because of its tremendous compactness and faithful image-consistency, but many others use PDF for the very reason that the format is *less* portable[1] – despite its name. Adobe’s PDF readers not only hinder saving, but PDF files remain extremely difficult to modify even if you do spend $250 for the full Acrobat product (whose main feature is that it poses as a Windows printer, so that PDF output is available from any program that can print). Adobe’s support for selective blocking of (1) cut and paste, (2) printing, (3) modification (and cryptographic signing etc.) suggests a possible long-term business plan similar to that of a monopoly cable company (for some early years anti-competitive accusations against Adobe Acrobat/PDF see [39]).

After languishing for years on the sidelines of the web, PDF’s adoption is spreading beyond corporate back offices, and has recently been endorsed by heavyweights Microsoft and Xerox as part of their April 2000 ContentGuard.com joint venture. For ongoing PDF updates, see planetpdf.com [25] and pdfzone.com [22].

[1] privacy and confidentiality are discussed in Chapter 7

3.1.2 PDF Flavors

PDF files come in many different flavors, of which two are especially important for OCR: (1) PDF Searchable Image (Compact) and (2) PDF Searchable Image (Exact). In either case the file has two layers, with the original image of the scanned document on top, and the OCR'd text "hidden underneath." This text is typically only available by cutting and pasting. However sophisticated OCR tools (such as we use), offer efficient ways of walking through your scans, and visually prompting you to fix suspicious words before saving. *Two-layer* PDF flavors include:

> (1) PDF SEARCHABLE IMAGE (COMPACT) is the most immediately useful to OCR: blank areas and words that are OCR'd with over 95% confidence are removed from the bitmap image (typical with Adobe's OCR products anyway). Its small file size and the fact that successfully OCR'd text becomes searchable makes this an ideal format for general use documents: color, grayscale or black and white. I call this flavor *"Compact PDF."* Example page size: 341kB
>
> (2) PDF SEARCHABLE IMAGE (EXACT) is not only the most visually pleasing (preserving a full bitmap), but also the most valuable for archival and legal purposes. Naturally these are the most bloated files. A key benefit however, is that the image can be re-OCR'd years later should recognition algorithms improve. Losing the original image (or hardcopy) suddenly becomes less of a problem, and Adobe should be widely heralded for pushing this two-files-in-one standard. I call this flavor *"Archival/Exact PDF."* Example page size: 381kB

Until recently both the above flavors of PDF, especially (2), fell under the rubric of "PDF IMAGE+TEXT." Given this "image+text" terminology is so strongly embedded within the OCR community, I continue to use it. Such PDF strategies are what many newspapers are using today (such as the Chicago Tribune which recently began digitally archiving back to 1847) to avoid ever again having to photograph their materials. [107, Outing99]. The example page sizes reflect a single-spaced typewritten page with bullet-style point-form indentations, that was OCR'd by our system. In closing, there are other important PDF flavors, which are less germane to this thesis. *Single-layer* PDF flavors include:

> (3) PDF IMAGE ONLY "for a cross-platform image of the entire scanned page" [34] but anyone in their right mind would of course use non-proprietary TIFF to ensure interoperable longevity. All scanning and OCR generally begins as a TIFF image file, though it is sometimes further compressed during long-term storage. Example page size: approx. 350kB
>
> (4) PDF FORMATTED TEXT AND GRAPHICS "for compact, searchable files with only one layer." [34] This is not unlike what's output from your word processor – a single layer containing all graphics and text. Graphics are preserved but images of text are replaced with OCR-formatted text wherever possible. Naturally this file type (formerly known as PDF Normal) has the smallest footprint of all Adobe PDF flavors, and can be an ideal web format in other cases, but it's far too lossy for our purposes. Obsessively disk-contrained users might use this lossy format for OCR, but its appearance of haphazard formatting typically make ASCII text more appropriate. Example page size: approx. 100kB

Archival/Exact PDF and various lightweight to gaudy instant-viewing formats were desirable for Haystack's scanning system (demOCRacy). To support a diverse user base (with varying bandwidth and storage constraints) we determined that demOCRacy should by default offer four formats: (1) Archival/Exact PDF, (2) Compact PDF, (3) HTML for immediacy/openness and (4) ASCII text as a baseline. Word was not chosen for the simple reason that Adobe Capture 3.0 (our OCR software) does not support this format.

In fact the finicky user can override these four formats and their parameters by tinkering with our OCR software when scanning. Their particular file preferences will faithfully show up in their secure web spool, but they risk annoying subsequent users.

Unfortunately the current unavailability of a Capture 3.0 API has so far made it impossible to restore these settings after each login (past and future workarounds are later discussed). However by designing in a rich choice of four formats (and various bundles thereof) demOCRacy is able to deliver a snappy self-service web GUI that pleases a very broad spectrum of users – at the small cost of a little extra CPU latency.

3.2 Choosing a (Fast) Scanner

Early on we favored an auto-feed scanner rather than a home-style flatbed. Prevailing usage requirements indicated many medium-length academic papers. As an example, (thesis supervisor) David Karger suggested he might like to quickly scan seven 10-page documents. In short we wanted speed, not tedium – our building already had a high-quality public flatbed scanner (with color and basic OCR, on the 2nd floor). Hedging for a semi-automated scanner could have been a possibility, eg. the $499 HP 6350Cse combined the best of both scanning techniques, as its low-volume feeder works *with* a conventional flatbed scanner. Sadly it was still much slower and more troublesome in its paperpath compared to high-end departmental (eg. Fujitsu) scanners.

The $799 HP R80 was an all-in-one unit that borrows from the 6350Cse but adds faxing, color printing and color copying. Truly a versatile home unit, you can scan a stack of 20 sheets at once with its feeder, or use the flatbed for books with spines or artwork. Feeder problems can be contained by restricting yourself to limited amounts of brand new paper, however in my experience and that of my friends, the feeder jams incessantly otherwise. Our system needed to work on real documents for untrained users, not provide a showy demonstration with pristine paper fresh from the mill. In short, the HP-class of sub-$1000 consumer products was insufficient for our needs.

Support for low-quality 200dpi fax scanning is never a problem on any scanner – and nice to have in light of the rash of recent “free” fax-2-email services like efax.com, fax4free.com, callwave.com and jfax.com. But then why not just use these services directly, many of which now offer basic OCR in their own right? Of course, they all profess “no more paper-jams” – so long as you provide the camera, which ironically requires your own fax machine. For all its faults, this (rigid, low-resolution) approach is based on an established open standard (Group 3 international faxing). Its platform-independence across both phone networks and Internet would have offered attractive worldwide ubiquity.

For demOCRacy (our prototype system), we wanted and got much more. The nature of this project made it impossible to anticipate obstacles, such as the costs of paper jams to hurried users. Photocopier-style paperpath nightmares are unpredictable until you finally try it with your materials and your user base. While I was urged to take risks in order to go forward, I chose to go through an earnest competitive analysis for our most expensive acquisition (the scanner) to avoid later integration nightmares.

True, a modern digital copier, if well-administered, would also have avoided much of the incessant paper-fussing of “cheap desktop OCR.” Such copiers increasingly act as scanners (and fax machines) that jack right in to your office Ethernet. This would have entailed a tremendous investment with an equally expensive service contract, as well a new skills investment tax for users – whenever the “Swiss Army Knife” Xerox machine sputters. Out of this all it became clear that the decade-long service life of most copiers – and their brethren professional scanners – was a fundamental (if unexpected and undesirable) ongoing design constraint.

Users simply won’t OCR (or print) their documents at all if they face ongoing paper jams and shortages – if only for self-cleaning paperpaths. So the choice we eventually made (see “Fujitsu 3093DG” section below) included a straight paperpath so simple it is essentially self-cleaning, with just one huge button for rare double-feeds or jams of strangely sized paper.

While a far more unreliable solution, users might well have felt more empowered if we had instead budgeted the thousands of dollars to give them each their own scanner or multifunction printer. While decentralization is often appropriate, much as authoritarian mainframes migrated to the anarchy of personal computation – interfaces to the offline world are more error-prone. This is not as straightforward a problem as distributed computation (a challenge in itself), but bridges between physical and electronic (actuators and especially sensors) should similarly penetrate closer to users over time.

It may be partly psychological, but putting a multifunction printer/scanner in your own room takes up prescious little desk space and you'll never wait for printhogs – or leave your cubicle again. 1996's bastardized low-end $1000 digital copiers are now available for only $250 and frankly would be a more appropriate solution for certain users, if the accounting (and jealousy) issues could ever be solved. Of course, to enterprise sysadmins, supporting "cheap desktop OCR" may present even more headaches than PC anarchy. Bulbs burn out and mechanics fail regularly in today's fickle $49 scanners. Arguably, even crumpled paper and errant staples are not the worst of it:

> "the simple task of Verifying the Output of OCR (emphasized within original) to correct the recognition and formatting errors is always the most cost-intensive component of any OCR application. The raw OCR speed of many hundreds of pages per hour is limited in the process by the bottleneck of clean-up, which usually proceeds at more human rates than the computer process...the user is presented with a crisp view of each suspect image next to the text in question. With the original and the OCR result viewed side-by-side, the editing process is quick and efficient." [74, Caere]

Here OCR vendor Caere (now part of ScanSoft) admits to some of OCR's hidden verification and quality control costs such as layout alignment, spell-checking and other "proofing" – even after accounting for the costs of paper jams. Clearly Caere has a vested interest in selling OCR acceleration boards to stick in the back of your PC, for high volume capture/conversion, when you're fed up with the frustrations of their low-end products. Regardless, they touch upon a profound truth: OCR cannot be fully automated – unless users have a significant tolerance for error.

For Haystack's purposes, with keyword searchability of documents a primary goal, imperfect recognition should often suffice. Typically well over 90% of a document's words are faithfully recognized by a well-tuned OCR process, if the text is large enough and uses common fonts. Still, it is well worth noting the (quality-oriented) standard industry practises we chose to defy.

Expensive network OCR servers used *in conjunction with* human proofreaders are generally far more cost-effective, given volume. This perhaps unfortunate fact is widely documented amongst experienced data entry experts [77, Haley94]. It is the reason demOCRacy (Haystack's OCR subsystem) attempts to offer a rich user-interface for web previewing in addition to simply auto-archiving. If we were wealthier, we could have tightened the feedback loop of document retries towards increasingly automatic archival-quality OCR – using multiprocessors, clustering and specialized accelerator hardware *combined with* highly-trained human editors.

> The key maximum ROI (return on investment) of OCR is achieved by concentrating all of the people time on tasks that only people can do, and automating all of the rest. [74, Caere]

These truths were profoundly illustrated during my visit to LASON in Needham, Massachusetts where dozens of employees scan 60,000-90,000 pages per day at one location alone. LASON, a professional scanning multinational with three centers in Massachusetts, provides integrated outsourcing services for all sorts of information management needs such as image and data capture, data management and output processing – not unlike a Kinko's chain for governments and banks. As a very rough guideline (everything depends on the details of your job), they charge 3 to 5 cents per page (face), plus 1.5 cents per keystroke for manually keyed metadata. They also happen to be the largest single printer of bulk "personalized commercial mail" of all the solicitations that enter the US Postal Service (for example delivering GM and Ford's recall notices) [18].

LASON represents an even more centralized solution among the spectrum of solutions to the problem this thesis set out to conquer. Their outsourcing solution is most appropriate for those who want to batch-OCR large filing cabinets. They clearly know their business: globally they manage 2.5 billion documents on-line and print over 150 million documents a month throughout the world, and indicated an eager willingness to work with Haystack over the long-term.

To simulate the mass-market distributed future, I initially favored consumer peripherals and handheld scanners – to better get a handle on popular use patterns for the future of OCR. However, for our prototype system (demOCRacy) the scanner purchase decision was reoriented towards machines that deliver speed and general ergonomic usability *today*. Truly democratic portable OCR can wait.

We nearly bought a $1000 Fujitsu singled-sided scanner (there are several good ones that break the 10ppm minute barrier). But after visually sampling 100 documents in (thesis supervisor) Lynn Stein's filing cabinets I found that roughly 60% of her academic papers were printed on both sides. Did we really want to feed so many of our documents twice, and worry about collation errors? On the other hand, two-sided (duplex) scanning is a luxury that pushes up the price by thousands of dollars, with two simultaneous scanners, one on each side of the page.

Ironically these duplex machines sacrifice both resolution and color, however the budget was there and we decided to go for it. The document scanning business has its own particular quirks which can be traced back decades, but as LASON proved, the basic job of OCR gets done right even with decades-old technology. Innovation is slow in a business shaped by the banking, insurance and legal industries – but even the bulk-scanning business is now heading towards color, given the growing popularity of color printers. If only we had waited another six months, this may be what we would have bought, as high-speed color Fujitsu models are finally now being rolled out after many years of anticipation. This might have been especially appropriate considering MIT's AI department also just decided to invest heavily in color (printing).

Still, alternative usage scenarios could have favored completely different architectures; and others will make completely appropriate scanner purchases towards other ends. Initially we demanded a TWAIN-based scanner ("Technology Without An Interesting Name" is a scanner driver consortium, as is ISIS) to guarantee maximum choice over OCR software without being locked in by any particular scanner.

For the moment however, popular TWAIN drivers remain on the low end (this may change) and Fujitsu's high-end ISIS driver proved far more functional – though both work. If you are shopping for low-end to mid-end scanners you're best checking USENET newsgroups [1]. On the high end, there is a very useful chart of $1000+ scanners available online[2] [5].

3.2.1 HP9100C Digital Sender: why we're better

A tantalizing product that just about made this entire thesis redundant is the HP Digital Sender. The $1100 (street price) low-end model is nothing more than a slow 4ppm fax and scanner that plugs into your office LAN. But the $2600 (street price) high-end 9100C dispatches scanned images (at up to 15ppm) directly to email, which can then be OCR'd by the client:

Its use of SMTP (Internet email) for multi-megabyte documents means your output will sometimes be delayed – our system, demOCRacy solved that problem with a secure web server. So HP worked with Adobe to cook up a clever variable compression (within each page) PDF flavor that can mitigate the severity of these delays. HP's approach fundamentally limits resolution; indeed it is a very expensive 200/300dpi solution. Clearly, their design model is inappropriate for long multi-megabyte documents given that SMTP (email) often back-throttles causing uncertain delays even in high-speed LAN environments.

[2]http://www.cddimensions.com/document_scanner/

Admittedly, I actually considered using SMTP document transmission until realizing (1) we occasionally want high-quality (eg. grayscale) documents well over 100MB and (2) security actually matters! HP's solution recommends installing special software on each client including "send to application" for automatic MIME unpacking of documents into specified directories on the client's computer. We implemented much the same ourselves – but in either case many shops will frown on the security risks. More important: our limited user testing showed me that users passionately prefer self-service directory-style web interfaces when picking up their documents – *together with* email notification. In the end, do-it-yourself on-demand document "pre-trieval" became one of the most popular aspects of our system.

An irritating problem with Digital Sender's homebrew PDF format is that its images can inherently only be OCR'd by their (homebrew, again) Adobe Circulate, which is a strictly limited package and not available otherwise. Again you must install this special software on every client – and the user license limits you to 25 simultaneous users. Assuming your clients are all compatible with this Windows software, this has the positive potential of distributing the OCR load – so long as your users don't mind their personal workstations freezing.

The greater problem with HP's Digital Sender is a fundamental lack of security (SMTP email and all maintenance passwords are sent in the clear). This is unacceptable in this era when the very documents we tend to scan (and fax) are sensitive documents that have resisted electronification for financial, evidenciary, archival and privacy reasons. This could very well be the reason you've heard little about this very innovative business-oriented machine almost 2 years after it hit the market.

Admittedly, demOCRacy (our prototype system) also has severe *trust* issues: the bandwidth and security problems of email have forced us to set up an Automated Teller Machine-like document escrow service, which not all users will be willing to trust. All we can do for now is ease these fears in the same way that a bank promises to safeguard checks you deposit into their ATM. Even if a kiosk-friendly public key infrastructure existed (eg. an offline PKI with the user's PGP public key on a smart card), documents would still be vulnerable to cracker or management surveillance prior to encryption.

So I make clear that demOCRacy and its administrator(s) make every effort to protect the confidences of peoples' documents unless forced by court order, and that we actively support the "anon" account, whose unfortunate limitations are later addressed. The specific security measures put in place within our own solution will be discussed in Chapter 4 especially, but merely as a preview: we chose to automatically delete all demOCRacy documents one week after scanning.

Even if its security problems were suddenly solved, SMTP email is still far too erratic and capacity-constrained to yet be appropriate for our heavy byte streams. Our own security assurances will clearly not be enough for risk-sensitive users – though the surveillance capabilities of our service will be more than enough for certain corporate/government applications. The fact that demOCRacy.lcs.mit.edu is today not only our UNIX server, but also a workstation for general Haystack research, is even less reassuring for those seeking confidences in our short-term document escrow service.

Caching of personal data always represents some measure of security risk, however if nothing else our user files are very likely protected by the U.S. Electronic Communication Privacy Act (of 1986) which prohibits unauthorized eavesdropping by all persons and businesses (not only government). The ECPA protects all text and images of personal communications, even in storage, and even in many cases where the medium is not email [53, Bowman96]. Unfortunately the ECPA's criminal penalties (and fines) are only applicable to system administrators themselves if they publically reveal users' data.

In short, HP's Digital Sender [15] is a dandy toy with very easy installation, that will suit other environments – but whose architecture is far more appropriate to a ubiquitous broadband future. Today, it lacks important features that our solution achieved: (1) speed, (2) security and (3) download flexibility (web self-service *or* automatic-incorporation with approval).

3.2.2 Fujitsu 3093DG Scanner

The business scanner market is completely unlike the retail peripheral market. The consumer market works around product cycles so short that the shelves are full of "planned obsolescence" printers and scanners completely unrecognizable 6 months later (which are often cheaper to replace than repair). It was shocking how little resemblance this commercial reality bears to the business machine marketplace [76, Haley94]; some of today's best-rated (eg. Fujitsu) business scanners are currently unchanged over almost half a decade after their release [12] – with the exception of several of driver updates to support new OS's.

The quality of professional scanners is so much higher (and the production volume so much lower) that manufacturers simply cannot re-engineer every year. To be clear, their product life cycle is (for now anyway) generally an *order of magnitude* longer than that of consumer scanners. Of course manufacturers gradually drop prices over their five year (or more) product cycles. A working high-speed data entry system often lasts more than *ten years* [76, Haley94]. This does not bode well for the short term prospects of OCR proliferation – and those who live for nothing but frothing innovation. However this means (quite unexpectedly) that, (1) it is easier to compare competitive offerings and (2) at least the basis of this technological outlook will continue to be relevant for several years.

Our extraordinary Fujitsu 3093DG was introduced in 1996 yet is still near-universally acclaimed among departmental (up to 40ppm) scanners. It cost about $3300 after haggling the price down from the $4000 that full-service dealers charge. I had done enough comparison shopping over the web and phone that I felt (just about) comfortable enough avoiding the implicit $700 service contract. I knew I had a one-year on-site warranty that would not expire until February 2001. I have been fortunate to have made a highly appropriate choice (details will follow) but I suggest others pay full price for 800-number hand-holding, especially if they have not spent a month researching all the right tools.

At 200 dpi, our scanner does 27 pages per minute with letter sized paper, and in duplex mode 45 *faces* per minute. Separate charge-coupled devices read each side of a sheet as it is pulled through the feeder. Running at 300 dpi (as we encourage) roughly halves these speeds but this is still more than fast enough given the PC's CPU recognition bottleneck. The CCDs themselves read 400 dpi so real grayscale is possible up to 400dpi – so black and white can be very accurately interpolated up to 600dpi.

Real grayscale is not possible in duplex mode however: only in simplex mode or on the flatbed. Bitonal scanning, ie. black and white, is generally prefered for OCR so this is hardly an inconvenience. The sometimes Japanese documentation (Fujitsu is a Japanese company) for the scanner may be somewhat lacking but its trouble-free operation far and away makes up for it.

OCR gospel claims that 300 dpi is readable by humans and should always be used for scanning [89] (though bulk scanners such as LASON [18] often still use 200dpi). Since humanly unreadable documents tend not to be OCR'd, most industry software has been trained at these medium resolutions. I confirmed this industry rule-of-thumb when I tried various 600dpi scans that were indeed genuinely much sharper, but whose post-processed OCR results tended not to be any better, if sometimes worse.

Myself I still feel that the more (non-interpolated) pixels the merrier, if only for archival reasons. For example, MIT wisely scans its theses at 600dpi (bitonal) [19] in order to reliably capture the 8-point fonts which are the minimum permitted by MIT guidelines. In a similar vein, the use of color is widely acknowledged not to improve performance [89] – this too may change over time.

There is no doubt however that illumination and adaptive thresholding are critical to obtaining clean bitonal document images for OCR. Low-end flatbed users struggle endlessly with contrast calibration issues. Our 8MB Fujitsu 3093DG includes "Scan-Right" hardware that uses its own intermediary grayscale processing to deliver such dynamic contrasting [11]. Very expensive high-end Fujitsu scanners provide better separation, noise removal, image edge-enhancement and dynamic thresholding.

Our Fujitsu's Automatic Document Feeder (ADF) holds roughly 50 sheets, depending on the paper you put in it of course. Genuinely reliable, absolutely no maintenance was required after scanning over 1000 sheets. The one large button you need to disengage jammed materials was only used 3 times despite the incredible diversity of paper, if not cardboard, that I threw at it. In short, paperpath frustrations can be contained (in a controled environment) as evidenced by the northern Europeans who (repeatedly, and successfully) OCR'd PGP source code. After upgrading to a professional ADF, they experienced "not a single double-feed over thousands of pages." [99, Nijssen98]

Remember that Fujitsu 93DG scanners are capable of scanning up to 27-45 OCRable faces per minute while the ADF (auto doc feeder) has only a 50-sheet tray. This is confusing for newbies to the world of high-speed document scanning: who would ever sell a 50ppm printer that filled up its 50-page output tray in a single minute? As my visit to LASON proved, a complete scanning job flow is inherently a *quality-control* intensive process with many human exception-handling requirements such as image enhancement, paper misfeed, software snafus, human error (wrong pages or wrong doc), etc.

While very high-end production scanners sometimes have hopper capacities of 1000 sheets, nearly every workgroup or departmental scanner on the market today limits you to a 50-sheet feeder. This is unfortunate, but perhaps manufacturers want to keep customer expectations realistic in this typically labor-intensive operation. Traditionally, most problems are quickly solved by rescanning the document.

Fujitsu has been a leader in high-speed scanning hardware for years and it's increasingly clear that money on our prize possession was well spent – if only the PC, its OS and OCR software were so reliable! Please examine Chapter 6 (the Manual) for more scanner operation details.

3.3 Choosing (Modern) OCR Software

Knowing in advance that the greatest challenge would likely be interfacing with proprietary OCR software turned out not to help. How to know what difficulties might arise without buying the product first or spending a fortune for access to the proprietary API? Xerox/Scansoft will license access to their consumer OCR to HTML and PDF APIs for $15,000 [29]. Adobe publishes most of its ancient Capture 2.0 API from 1997 [2] but leaves you hanging if you need access to today's product – Capture 3.0, which we chose largely because it offers better "document understanding."

Much like the OCR FAQ itself [89], open source OCR development has unfortunately languished. While the Linux world is no longer so "bottom heavy" as it once was (open source developers concentrated on reinforcing the operating system rather than user programs) it continues to lack certain state-of-the-art applications. So while there is an ongoing open source Voice Recognition effort, the two "public domain" Linux OCR projects are very forlorn [33]. SOCR.org is untouched since November 1998 and a similar project [4] officially has "fallen into a coma" since mid 1999.

Modern OCR applications tend to (1) be Windows-exclusive and (2) have GUIs that automatically pop up. We chose to tolerate Windows as an input device for now. While vividata.com offers Caere Omnipage's engine on UNIX, so far it only outputs text for Linux. Solving or more accurately mollifying (2) was a harder problem: in the end we created a resilient web interface that sits next to the commercial OCR GUI. Our Netscape "login-box" gives the user control over their job for later secure pickup over the network. The kiosk's 19 inch screen comfortably accomodates these two GUIs, which are paired side-by-side – as well as permitting detailed image inspection if necessary.

While we focused on the high-speed processing of long documents, we also explicitly permit individuals to scan diverse materials, such as newspaper columns or glossy articles. Especially in these cases, extracting as much paper document structure as possible becomes highly desirable. Modern OCR "document understanding" or "document analysis" recognizes many text/layout substructures, maintaining the integrity of bulleted sections and chapters, while integrating images and increasingly recognizing/codifying web addresses and the like. We achieved this outputting to HTML (with the help of javascript) despite arduous limitations in both Capture 3.0 and HTML itself.

The most recent (year 2000) OCR packages deliver on most of their promises. Of course if contrast is lacking, or if the original sheet itself is tattered, or if humans themselves can barely read it then you can write it off. OCR is an interface to the real world of atoms and peanut butter smears – not a business where one achieves perfection. In the case of scanning fragile antiquarian documents, the Heisenberg Uncertainty Principle is often said to apply. In these cases the binding is usually cut off before the deteriorating pages are scanned once and never again – one more reason to use the full PDF IMAGE+TEXT.

As previously discussed, our OCR software selection was partly guided by the need to simultaneously deliver multiple useful output formats, and Capture 3.0 fit the bill very well. As a result of this choice, our speedy NT kiosk spends a solid minute processing complex pages, a large portion of which is spent converting into various file formats after recognition is complete.

That modern OCR packages extract metadata such as web addresses (and occasionally phone numbers, and more) is a powerful semantic benefit. Haystack will increasingly key into this (eg. HTML) data via conventional services (Capture 3.0 simply encodes email and web addresses into HTML and PDF as web hyperlinks). Expanding this metadata capture will be further discussed in closing chapters.

We could have chosen Scansoft's OCR [28] which integrates HP's JetSend protocol for rich inter-appliance communication, a scheme supposedly active in 5 million devices today. But this was far too bleeding-edge for our users. All we needed was a tight, well-defined interface to the document directory where OCR'd documents are dispensed.

Surprisingly, the free OCR software that comes with even $29 scanners is often not too different from the $100 COTS (commercial off the shelf) packages they tell you to upgrade to. As previously alluded to, consumer scanners are so slow compared to modern PCs that a faster OCR engine won't buy you much. Programmability is absolutely lacking even in these more expensive consumer OCR packages, and the market is increasingly monopolized (Xerox affiliated scansoft.com acquired industry stalwart caere.com in January 2000). These roughly $100 OCR programs lacked both the customizability and the improved document understanding available in the $700 PDF-oriented program we eventually chose.

Still, we looked at these OCR companion document management applications for inspiration – eg. industry leader PaperPort from ScanSoft, whose SDK exports images and more. Onetime OCR mainstay Caere's $29.99 PageKeeper is a more open though less used competitor, with sophisticated functionality. Again, these two leading companies in consumer-grade OCR have just merged: their products bear rough similarities to Haystack, with far better image management. Other products like Nolo RecordKeeper do similar OCR document management for the legal profession (perhaps the biggest non-forms user of OCR today, where many legal searches used to take days).

What's in it for Haystack? For mere companions apps, these visually-oriented information managers replicate an amazing number of Haystack features including typeguessing, querying, and more. They include many powerful format converters that could have been useful to us – and also offer innovative visual file-browser interfaces that Haystack could one day borrow. Unfortunately such low-end products still necessitate "managing your doc manager," despite their increasing power. Even high-end scanner companies are increasingly bundling these visual-collection managers, but they proved inappropriate for our use by lacking multi-user support and overall scalability.

On the bright side, even consumer OCR packages often now promise a dozen different languages, though not as many as Capture 3.0 (our eventual choice discussed below) which delivers 20 dictionaries. Future operators of demOCRacy (our prototype system) may wish to explore other sophisticated industry add-ons. Many OCR industry niches cater to domain-specific plug-ins, eg. math/science symbology. I did not research these avenues (software for particular sectors of the economy, much of it menial metadata-extraction, eg. forms). I also did not delve into 3rd party add-ons available to Adobe Capture and others such as (1) hardware accelerators and (2) CPU load-balancers.

3.3.1 Capture 2.0

I played with Scansoft's [28] $99 consumer OCR suite (Pagis Pro) but my first real successes came with Adobe Capture 2.0. Surprisingly, this 1997 offering is a solid product that is still very widely used. Intriguingly, this $700 software package comes with a parallel port hardware dongle that leaves many wondering who is being "captured." Regardless, it meters you at 3.5 cents per page-face, or 7 cents per page for double-sided scans. Presumably this dongle is not the utmost in security and simply contains a unique identifier. Anyway the pennies are only debited when that particular page face is OCR'd, so image-only scans are indeed "free."

To be clear, the fact that most Haystack users simultaneously generate four output formats per face (Archival/Exact PDF, Compact PDF, HTML and text) does not increase the price from 3.5 cent price to 14 cents. Yes, additional dongles must be purchased (typically for $700) but for now there are about 20,000 OCR faces left on the 21,000-face startup dongle.

Capture 2.0 operates on an extremely simple visual metaphor: a staging-area folder for TIFF bitmaps, and an output folder where your favorite output formats are dumped. This is very much the same as what the $99 programs present to you, only with the added bonus of full PDF. Despite the age of this Windows 95-compatible

OCR engine, results are remarkably good – though customizability and rich document analysis (including metadata extraction) are sorely lacking.

3.3.2 Capture 3.0

This February 2000 release could easily have been labeled the "Microsoft Word" version of Capture, in that it comes with a myriad of features attempting to please everyone. Unfortunately many simply don't work at all, and its bugs are all the more numerous running over Windows 2000, despite the assurances of Adobe representatives (eg. emailing output failed completely). The 2.0 product's core functionality is preserved but its simple/elegant UI has been completely gutted in favor of a half dozen side-by-side input and output panels.

We chose the single-CPU non-clustering edition: the low-end $700 product. Capture 3.0 uses the exact same dongle as Capture 2.0, but it now requires Windows NT, or perhaps (depending who you ask) Windows 2000. The OCR itself is 30-50% better, according to Capture mailing list consensus [3]. But that's not the full story: 3.0 is really an entirely different product, packed with power and fragility.

Released prematurely in February 2000, Capture 3.0 took several months to become generally available to outside software vendors. I ordered directly from Adobe the day it became available, and quickly discovered very grating Capture 3.0 UI featuritis bugs. Moral: stick with the core functionality if you don't want it to freeze on you. One frustrating bug is that it could not produce simple HTML pages as advertised. It does succeed in producing HTML javascripted among frames however, all bundled up in a pkzip file. In short the core functionality works – despite the fact that it continuously wastes 60% of our very fast CPU, even when idle. The broken features will just have to wait for a maintenance release – hopefully soon in 2000.

On the bright side, Capture 3.0 multitasks wonderfully. You can scan seven ten-page documents and demOCRacy will email you a confirmation receipt a few minutes later (depending on your input) after processing is complete. Or you can stick around and watch animations as it displays (in different panels) various churning gears and diagnostics for the ongoing parallel steps of its documents processing.

Capture 3.0 imposes on you to select a maximum of 16 different fonts that it will use for recognition. In keeping with our aim of giving instant access to a very diverse user population, who shouldn't have to learn their way around all of Capture's bells and whistles, I selected 16 of the most common fonts as part of the default workflow ("AUTO-HAYSTACK" to be discussed below). This is despite the fact that across the many documents I test-OCR'd, many more fonts seemed to be recognized than the original three default fonts that were selected.

Granted, there are times when nothing is recognized (eg. small fonts, colored paper, or too many scribbles). Capture 3.0 claims to have certain powers to adapt to your work over time, which may be true, but I personally have not observed evidence of this. Certainly it recognizes fonts from those among its palette (imperfectly) but myself I cannot claim to understand the subtle underlying geometry engines of OCR.

The Capture 3.0 claim to fame is that it adds easy programmability, so called "workflows" that allow sophisticated fine-tuning towards particular kinds of documents, output or use requirements. The amount of fine-tuning available to you is astounding, however after you finally learn your way around the interface, it suffers from two serious flaws. First, a simple textual scripting language would have permitted greater flexibility (for example, while you can set a fixed outgoing email address for all documents, there is no quick way to change this within a workflow). Second, adding more than a dozen workflows slows the machine to a crawl, in particular this caused Capture 3.0 to take 5 minutes to launch.

In the end I was forced to remove my workflows, reinstall the program, leaving little aside from our customized default workflow. This is the "AUTO-HAYSTACK" workflow I set up to generate our four desired formats (Archival/Exact PDF, Compact PDF, HTML and text). For all their power, per-user workflows were just not the right tool to support our busy Kinko's style self-service hit-and-run customers – especially given Capture 3.0's current bugginess.

There are many other aspect of Capture 3.0 that I could share with readers. However much of this program is self-explanatory, and in the end, after a select few tweaks (eg. technical dictionaries and PDF's optional per-page thumbnails were added), the OCR subcomponent works reliably to Haystack users' satisfaction. We owe our thanks to Adobe – despite the inevitable OCR and file reformatting bottlenecks that will continue to bottleneck scanner output over the course of a few more cycles of Moore's Law.

A minute to process one page seems outrageous considering we spent $3300 for such a fast scanner. Much of this is due to Capture 3.0's widely acknowledged bugginess and memory leaks [3] – its speed may well double or triple when Adobe releases patches within the year. Still, sufficient usage can justify these (time) costs given our unusual non-interactive design goals. Defying the perfectionist traditions of the document scanning business, we batch-automated the process, allowing you to run back to your office (or home), await notification, and select from many different output styles. Should serious errors arise, you can always come back to (a) scanner kiosk to later retry your problematic documents, perhaps with the more traditional interactive quality-control touchup tools.

Any user interface requires some degree of education: in our case users will quickly learn which classes of documents are problematic, to reliably batch-OCR documents with only rare retries. Users who dislike our pipelining process (i.e. are impatient for their immediate final output) can take consolation that our $3300 scanner will outlive several generations PCs – even a cheap OCR uniprocessor PC should be able to keep up with high-speed departmental scanners within the decade.

The bursty nature of customer arrivals to the kiosk makes this design possible. Job processing from all users are properly queued up, allowing the scanning party to immediately walk back to their office to do other work – useful if our receipt/notification and download scheme hasn't yet kicked in. It's all very automatic and it works. The Capture software keeps realtime log files that we were able to snoop into with demOCRacy's user accounting wrapper. On completion, a document's files are moved to our Linux server where they are directly placed into the user's secure web directory for pickup. Moments later, all originals on the somewhat less secure Windows kiosk are deleted.

Chapter 5 will graphically illustrate this operation. The casual user however, need not understand that their scanning and OCR on our NT+Capture kiosk is remotely monitored and managed. Eg. such users may not care that our secure web account server “demOCRacy” pulls OCR output off the kiosk as soon as it becomes available. Instead, dead-simple user interfaces were created to help all users coordinate their sessions’ documents. These UIs are the subject of the next chapter.

Chapter 4

The User Interface to demOCRacy

The objective of this thesis was to develop a seamless and fully integrated scheme to help you scan, OCR and archive – so that Haystack services can at last operate on your favorite paper as well as online artifacts.

When you need to scan, you just walk up to our NT-based kiosk, ideally located in a nearby public area. Relatively anonymous users intermittently approach this scanning booth with sheaves of documents to be scanned. Currently, we are making our system available on a word of mouth basis to members of the MIT AI and LCS communities and friends, which represents a potential of up to 1000 people. For our most up-to-date accounts policy see [8].

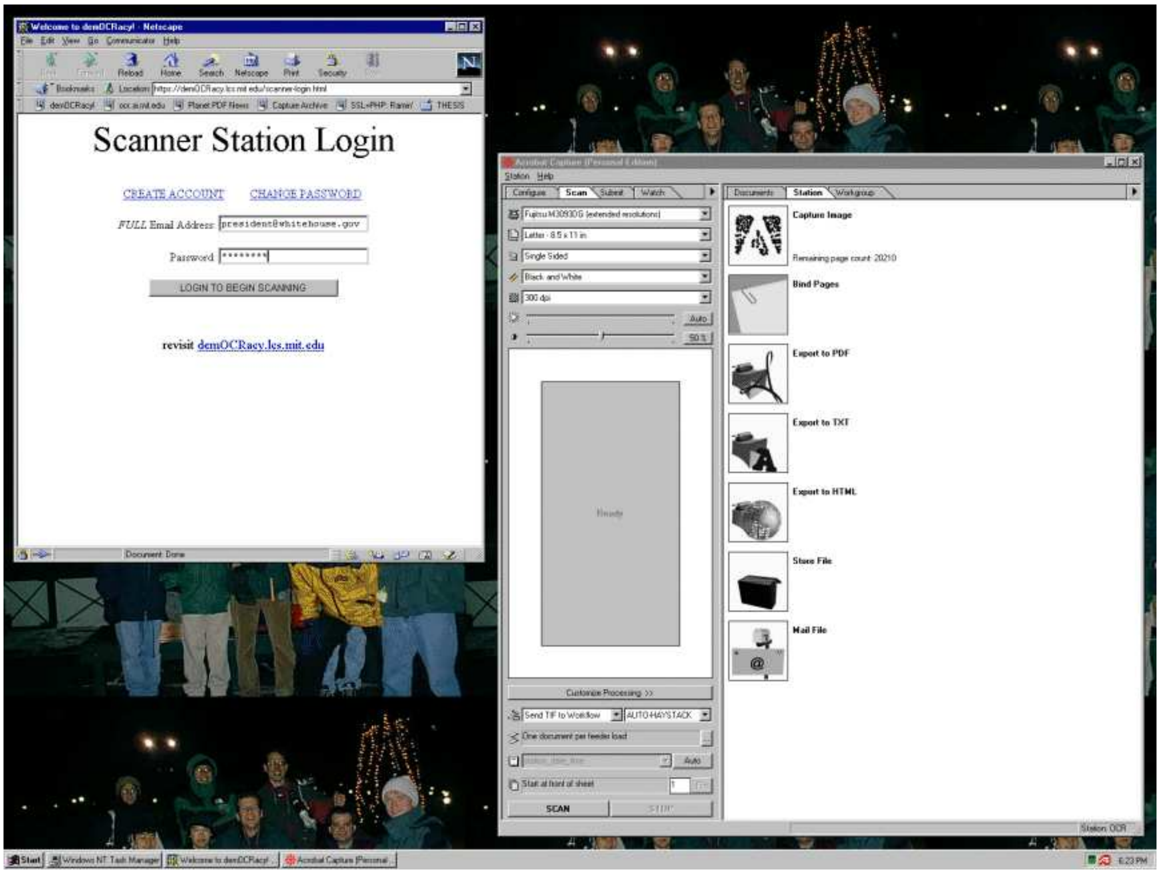
Any new user with physical access to the kiosk can quickly register today with nothing more than their email address. They should select a password used mainly for later web pickup of their OCR'd documents. After logging in and pumping their documents through the feeder (one at a time) they log out of their session and are on their way.

A convenient email receipt later summarizes their job and explains web (pre)viewing and download options. No credit card is required in our subsidized academic system – though it is conceivable our (primitive) accounting system could later be merged with LCS/AI departments' (similarly primitive) per-advisor photocopier codes.

4.1 Session Logins

Unfortunately, Capture 3.0 (our OCR software) was not designed for a multi-user environment – ironic considering it now requires Windows NT to run. Even if it were possible for MIT's AI users to log in with their regular UNIX accounts and passwords (as administrator Aaron McKinnon promised would be possible this summer) our problems here would remain. This is because any user who logs out of Windows NT itself (on leaving the scanner room) would necessarily cause Capture to exit, killing their own OCR job processing. This would not only prevent multi-document batch jobs: even single long documents would arbitrarily be held up until Capture 3.0 was later relaunched.

Despite the fact that users hate registering in general, we had to layer on (very low hassle) per-user accounts – if for no other reason than to permit people to keep their documents confidential, should they so choose. Graphic details of our user accounts model follow. First we introduce our opening screen, a resilient self-explanatory Netscape-based login-box. This secure web login-box launches automatically and sits permanently next to Capture 3.0's GUI on the Windows NT kiosk's large screen:



On the top-left you see the SSL-enabled Netscape login-box which we use to guide the user through their scanning session. On the bottom-right you see Adobe Capture 3.0, which controls the scanner and performs the OCR. The per-document “SCAN” button is in Capture’s lower left-hand corner: this is the *only* button users will need to touch while logged in.

Simple web forms support account creation and maintenance. Today setting up an account asks for *nothing more than* an Internet email address and a password you will use for remote web pickup. For now the email address is only used for notification, so you can get away with using a bitbucket (i.e. false email address) if you don’t want notification.

Today we insist that users be physical present at the kiosk in order to register, but you can remotely (and securely) change your password. This is particularly useful if you want your secretary or spouse to run some scans for you with a temporary low-security password. Like any good bank, we permit joint accounts. Simply choose your username to be "romeo@mit.edu,juliet@mit.edu" and you will each receive job receipts.

For additional security, we may in future require users to register with email addresses that end in "@lcs.mit.edu" or "@ai.mit.edu" – however this simple technical change has become a rather contentious policy quagmire. First, it runs in the face of MIT's LCS and AI communities' decades-long history of supporting "tourist accounts" for (more or less) trusted guests. Perhaps worse, forced registration would clearly harm the many LCS and AI users who distrust such procedures in general, i.e. many users strongly prefer the "anon" account for a few OCR test sheets before registering (which comes with its own privacy benefits). Forced registration may even provide false comfort given that the Windows kiosk itself cannot easily be physically secured. For now the kiosk is kept in an "invitation only" locked laboratory, but these issues will need to be re-addressed when or if the scanning station is moved into a high-traffic public area.

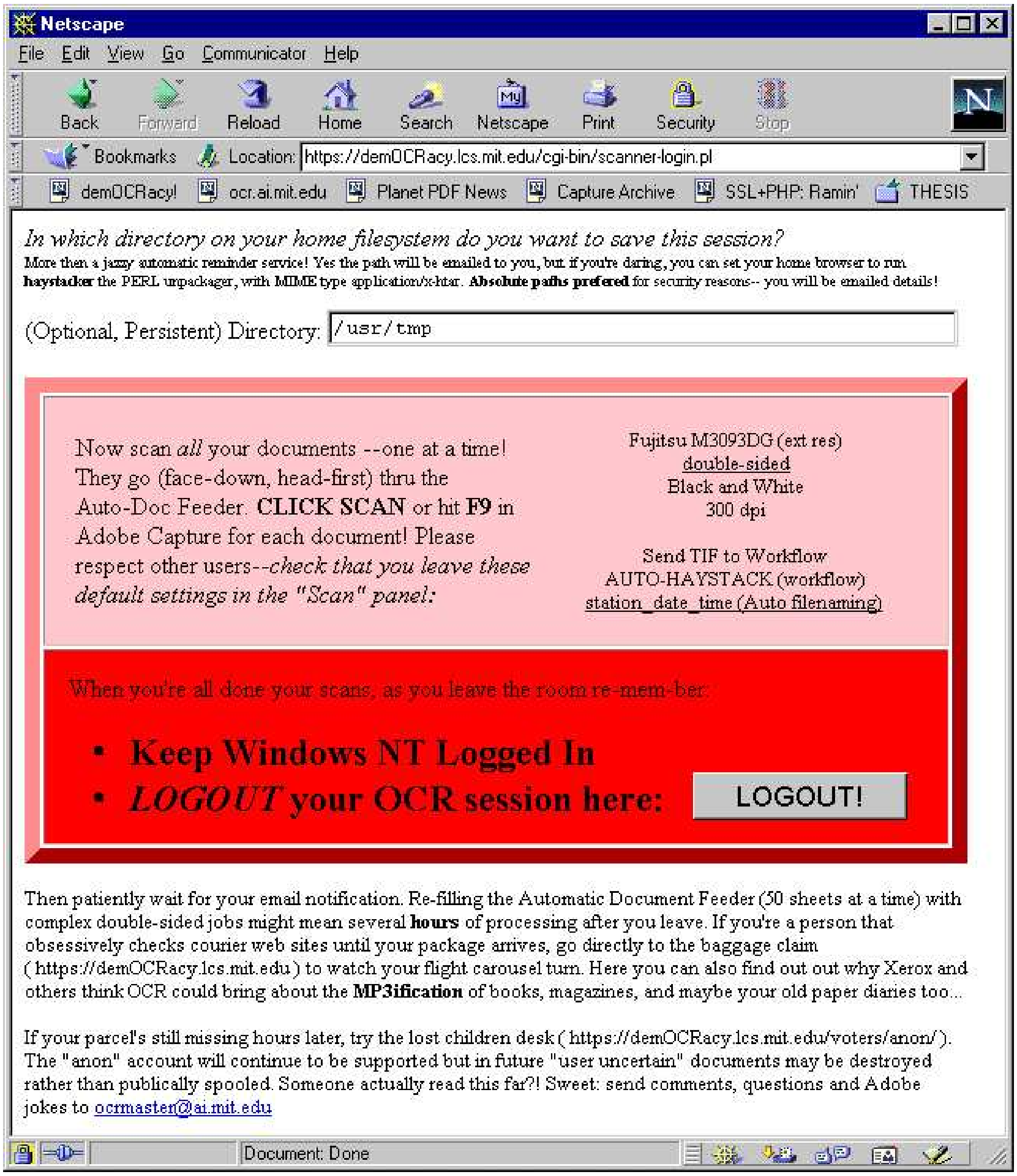
Authenticating the real live user to their email address was deemed unnecessary in the current environment, as this buys little but inconvenience. Most new users would be forced to go back to their offices for a confirmation code, scaring away many new users who just want to try their first quick scan. While the password is also used for repeat visits to the scanner kiosk itself, this is only to control spam (preventing other users from pushing errant documents into your secure web directory, and avoiding the associated "your documents are ready" email receipts).

Users that forget their password must consult the administrator at present (ocrmaster@ai.mit.edu). Of course, creating another account with another valid email address (or using the "anon" account) represent harmless workarounds. In future a full account service center (built on email auto-responses and https) could evolve out of the password-changing page, perhaps offering password-hinting based on "town of birth"-style shared secrets initially. This service center could in future support many account personalizations such as whether account-uncertain files generated right before (or right after) your login should be published to the "anon" account or "perished."

The “anon” account was set up specifically to support anonymous users but remember: this (shared) password-disabled account is of limited value if you scan sensitive user-identifiable documents. It is however appropriate if all you need to scan are public or non-traceable documents. The “anon” account can be explicitly logged into if you want batch-downloadable .tar or our .htar auto-downloadable packages. The “anon” account is also triggered implicitly when regular users log out (from our SSL-enabled Netscape login-box). In either case, Capture 3.0 processes your documents as usual, whereupon they move off the NT kiosk to the web-public directory:

http://demOCRacy.lcs.mit.edu/voters/anon/

Upon logging in (whether anon or not) we present the user with bright red instructions for making their scans. The user can also specify a directory to be used for later automatic downloading of the finished goods:



Hurried users can bypass all the above smallprint if all they want to do is try some scans. They may ignore the default directory for new users shown above (/usr/tmp), which they can try on a later visit. Thereupon, any (registered) user's "save to" directory preference will persist between scanning jobs. User-interface designers will recognize our practice of removing all distractory hyperlinks from the above page. A primary purpose of this page is to emphatically remind users (as much as possible) that they should actually *logout* of our Netscape login-box when they leave.

Our inability to restore Capture 3.0's settings upon kiosk login (due to the frustrating unavailability of Capture 3.0's API mentioned in the previous chapter) explains our prominent behavioral user warning/reminder. Even if the launch of Capture and Netscape were quick (sadly Capture can take several minutes to auto-launch after NT login) the fact that Capture's settings are not reset on shutdown made the Windows NT login process all the more inappropriate for demOCRacy. The available technologies can only go so far in enforcing good behavior – no less given users have physical access to the machine. This was one of many critical advantages to firewalling the web account server as a separate machine from our lower-security kiosk. While we discourage it in general, there is no doubt that experienced OCR users will take advantage of Capture's sophisticated interactive cleanup tools.

A timed auto-logout has not yet been implemented for several complex technical reasons, many of which have to do with our unavoidable current network topology. Until this unfortunate situation is overcome, you as a forgetful user should be reassured that malicious users cannot view your documents *nor* seize your password anymore than they can to users that have properly logged out. Unlike UNIX-style logins, there is nothing additional you expose, aside from your desired "save to" directory – so having the next user log you out only delays your document delivery. The right way to force auto-logouts will be to re-architect the job tracking account server to include time-stamps throughout, for properly executed timeouts regardless of fail mode.

Still, we strongly encourage everyone to log out: for now we offer the (intentional) incentive of letting you submit your "save to" package-download directory *only* upon logout (as well a quicker email job receipt). For all their convenience, it's worth noting that browser-based logins always bear some risk (http was designed to be stateless) as web-mail and other commercial providers continue to "discover" on a near daily basis [117, Smith].

4.2 Remote Pickup

Users are notified when their full OCR processing and conversions eventually complete. An email receipt is sent summarizing the job, including its size (alluded to in dollars instead of megabytes) and URLs for secure user pickup of their documents:

```
Date: Sun, 28 May 2000 20:36:27 -0400
From: root@pochard.lcs.mit.edu
To: buddy@whitehouse.gov
Subject: OCR #20000528_203451 receipt: $12.78

Your documents are ready for pickup; download within 7 days.


INDIVIDUALLY:

 https://demOCRacy.lcs.mit.edu/voters/buddy@whitehouse.gov


ALL TOGETHER:

 https://demOCRacy.lcs.mit.edu/voters/buddy@whitehouse.gov/_20000528_203451.tar
 https://demOCRacy.lcs.mit.edu/voters/buddy@whitehouse.gov/_20000528_203451.htar

 (for non-destructive unpack into "/usr/tmp")


--------------------------------------------------------------------
Thanks for doing business with d  e  m  O  C  R  a  c  y.lcs.mit.edu
--------------------------------------------------------------------

Haystack users: dare your browser to execute this "haystacker %s %u"
                whenever encountering MIME type "application/x-htar"
                TRY IT: http://demOCRacy.lcs.mit.edu/haystacker.html

If you received this message in error, contact: ocrmaster@ai.mit.edu
```

All users get a live web directory with a chronological listing of all output created for them over the prior week. As a rule, if your email address were "president@whitehouse.gov," then your password-protected output directory is available at:

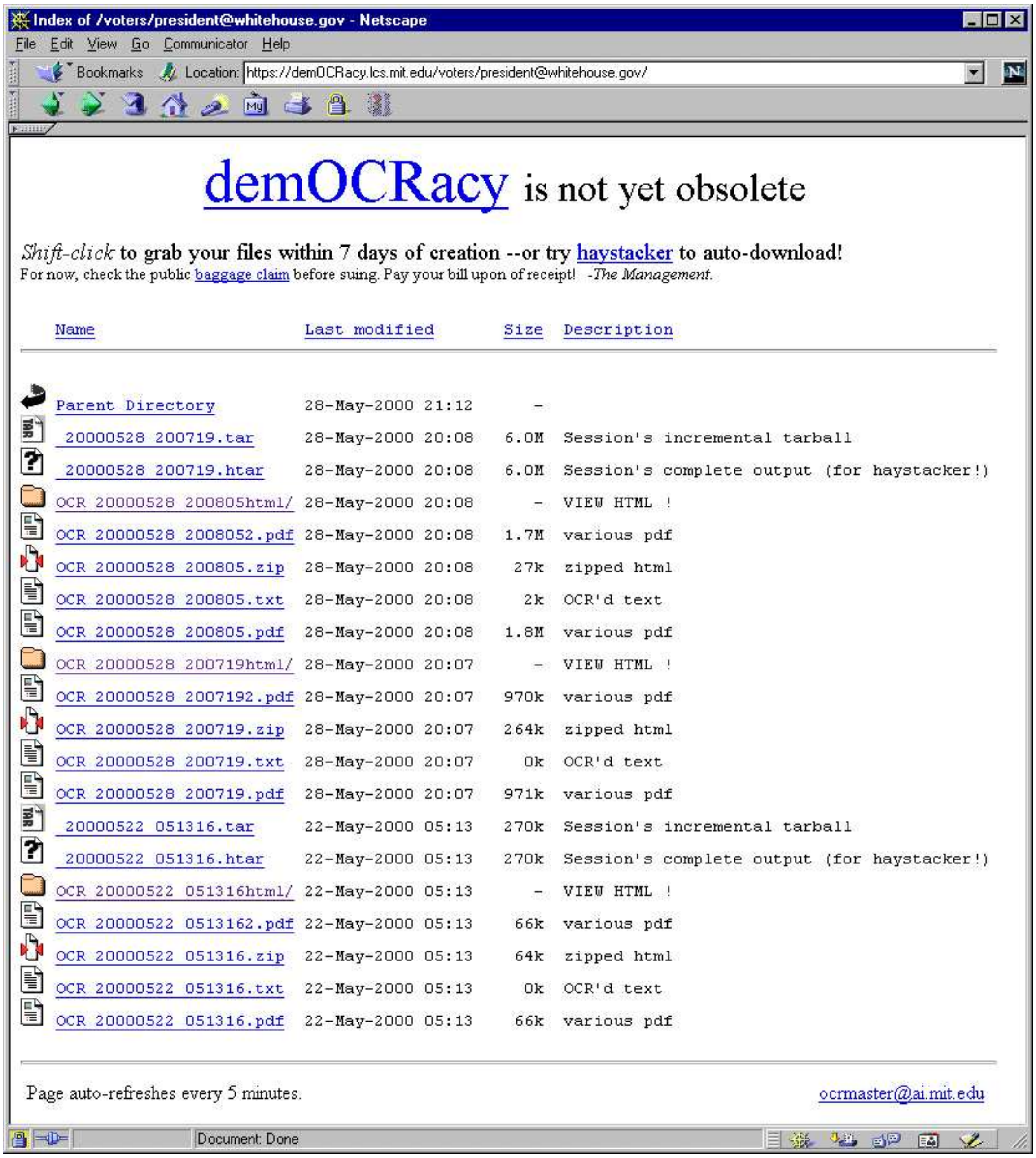
https://demOCRacy.lcs.mit.edu/voters/president@whitehouse.gov/

We encourage the above use of "https" (notice the 's') for users' security. However, if unencrypted access is necessary, eg. using scriptable downloaders such as wget or lynx, we also permit "http" downloads in the clear:

http://demOCRacy.lcs.mit.edu/voters/president@whitehouse.gov/

These self-serve directories have been layed out graphically so that the divisions between successive document batches each stand out – one job per login session –

with clear visual cues denoting divisions. Sub-directories (eg. per-session and/or per-document) were considered but over-ruled due to several users' strong preferences for a rolling historical view rather than more clicking. We may offer both options in the future (spool view and hierarchical). When a login session is fully processed (or even before, if you are impatient) you can inspect your document spool on your home browser with an interface like the following:



This example user has completed two sessions. Earlier, on May 22, a small document was scanned, without recognizable text. The most recent session (on May 28) is displayed on top with details of its two documents. In general, any of four formats are instantly web-viewable – depending on the client's bandwidth anyway. The per-document .zip bundles are offered in addition, for informal archiving of compact HTML. The .tar and .htar files on the other hand, contain all of a session's documents in all four formats. Our "haystacker" program that (optionally) auto-downloads such .htar files [35] is introduced in the next chapter. We do not offer an (live) aggregate bundle of all a user's documents from the previous week, as the per-session bundles were deemed coarse enough.

It's always unclear how diverse users wish to (or will) structure their digital filing cabinets, even if in our case these are but temporary web staging areas. Right now your documents are spooled chronologically into your web directory, which is what most users I asked voted for. We could have offered each user much more in the way of interactive rearranging, rebundling and deleting of their documents, not only after, not only before, but even during image-heavy downloads. With exception-handling "quality control" issues rife at every stage of the complete OCR dataflow, this would have been an especially useful addition. Building such an enhanced "windows explorer" interface requires a more complex MySQL and PHP3 back-end and will have to wait for a future release.

The Apache web server's many new directory listing options were deemed more than sufficient for demOCRacy's prototype. Apache's flexibility and easy configurability allowed us to display elegant flat personal directories that come alive for users in just the right order, with descriptions, visual separation of batches, and color-coded icon cues for each file format. Convenient HTML headers and footers above and below each user's listing of downloadable output were perfect for tip-of-the-day explanatory announcements.

Please note the author is well aware that using email addresses within URLs is frowned on by professional web designers, often for reasons of account maintenance and privacy. However, desiring to avoid an additional database with additional user identifiers, we chose to make our prototype system far more transparent. In so doing, demOCRacy seamlessly supports an environment where users often maintain several distinct email addresses.

4.3 User Security

A full understanding of the risks and ramifications to your documents requires examining the architecture of demOCRacy (next chapter). However we address the most immediate concerns to users now.

Passwords are never stored in the clear, only a hash is kept. HTTP basic authentication is used both for remote pickup and for kiosk login. This does nothing more than MIME-style base64 encoding of passwords as they cross the network, so we wrap this in SSL (https) encryption at all times using OpenSSL [59] and mod_SSL [61] on our Apache web server.

Documents themselves can be encrypted during download using either strong or weak SSL cryptography, depending on your browser. Our configuration is unusual, layering basic authentication within SSL – which itself could have certified client to server as well as just server to client. However we are not actually a bank, and in our case security requirements do not yet call for a licensing infrastructure (i.e. certificate authority). Consider that most stock-trading web sites today use no certificates at all: while user certificates (might) prevent more sophisticated man-in-the-middle attacks, it would have made location-independent access unnecessarily complicated for our users.

Our default SSL (https) users have absolutely all their browsing and downloading traffic encrypted, whether they are accessing demOCRacy from home or work. All users' kiosk logins and logouts are SSL-encrypted whenever they use the kiosk. Note that SSL even encrypts all URLs. Netscape meant it literally when they designed the "secure sockets layer" – i.e. SSL tunneling is such a low-level handshake that sniffers cannot even detect the very first URL you access.

While many of our users considered these precautions excessive, they provide added comfort given (1) your email address is currently part of our URLs and (2) we don't currently plan to use user certificates. Most importantly, a strong security environment has been built in from the start – so you avoid surprises months later during that rush job to scan personal/financial letters – that you never expected.

The phraseology "secure web directory" is used somewhat loosely throughout this thesis because we also support non-encrypted (but password-protected) access. In this fashion we support users of "wget," a powerful automated http downloading tool useful to those scanning non-sensitive documents. This works because wget supports HTTP's basic authentication passwords (a simple URL is insufficient to gain access unless you are using the "anon" account). While we encourage all users to take advantage of SSL (it is always the default, despite our significantly increased CPU load), non-SSL passworded web directories offer quite satisfactory security for many demOCRacy documents today.

Today users can take their own security decisions for general OCR pickup – until the day too many people scan (auto-OCR'd) credit card bills, some of which will inevitably be downloaded in the clear (and sniffed). In future we may demand all users use SSL as our security responsibilities change. Whereas online banks permitted lower security solutions as recently as 1997, today they require customers to upgrade to strong SSL. This general trend is illustrated in cryptanalyst Ross Anderson's paper "Why Cryptosystems Fail" [43, Anderson93] which profiles why *real* ATM's have been more secure in the US than in the UK. Simply by assigning liability to those best able to manage risk (the banks rather than the customers) the US achieved far lower ATM fraud than in Britain – at lower cost to banks. As a form of "document bank," we took this lesson to heart.

Note that with SSL or without, we cannot solve the real systemic/endemic risk of a user sharing a password between too many web (and non-web) services. Protect yourself: one of the Internet's best known security experts (Richard M. Smith) has repeatedly warned web designers of this problem [117].

Users face a rather more immediate security decision when they choose .htar auto-downloading, which is introduced in Chapter 5, and whose many risks are further described at the end of Chapter 6.

Chapter 5

The Architecture of demOCRacy

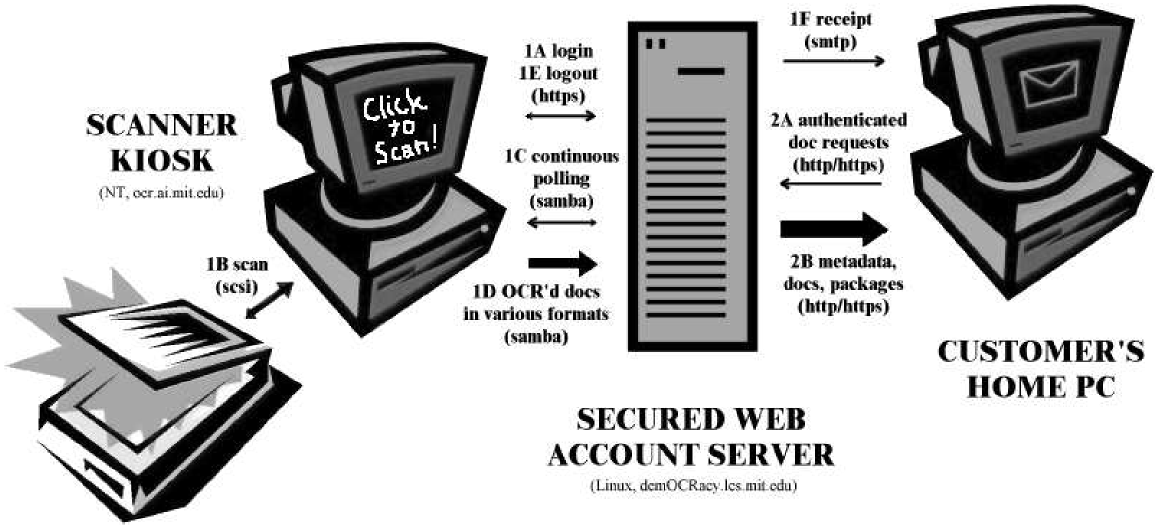
Our scanning system architecture is built around two standard Intel-compatible computers. The first machine is the centerpiece of our public scanner booth, driving its scanning and OCR processing. Documents quickly leave this kiosk computer as soon as they are OCR'd, whereupon they are transfered to a (more secure) 1-week holding area. This secured holding area is our web account server. While this second machine is not physically visible to our scanner kiosk users, it represents the vault for pick-up of their OCR'd documents using their home browsers.

This second machine (our Linux server) is not only where users retrieve their documents: it's where email notifications are dispatched and where all tracking of users and documents takes place. http://demOCRacy.lcs.mit.edu is the public interface to this relatively more secure (compared to the NT kiosk) machine. Customer billing or quota-ing might later be added to this centralized accounting machine.

This demOCRacy.lcs.mit.edu [37] server is the key to the entire operation. The NT+Capture kiosk (with Fujitsu scanner attached) is but a mere appendage, or client to this web account server. Harking back to the metaphor I chose for my abstract, the NT+Capture kiosk is the *ATM*, while the Linux web account server is the *bank*. Colloquially of course, "demOCRacy" can also refer to our entire prototype system. In summary, if the NT-based ATM is cracked, tears will be shed (eg. a sniffer could be installed) – but if the web account server is cracked, the bank's deposits are liquidated, i.e. past documents are exposed or destroyed (in our case a week's worth).

The Linux account server must be prepared to wait extended periods of time for the NT kiosk to process documents – which may occasionally come off "the OCR production line" out of order. Regardless of its other tasks, the Linux account server must always keep polling across the network, every 10 seconds (into Capture's proprietary log files on the NT kiosk) to watch for new scans that might have begun. At last, email notification is sent to the user when OCRing and packaging of their session's documents are finally done.

Largely peripheral, a third tier of our architecture can also be modeled. We include the user's home system, because its security is especially relevant if its owner enables our "haystacker" semi-automated download scheme. This mechanism and its security implications will be carefully profiled later in Chapters 5 and 6. Other users prefer to review their output documents over the web in conjunction with do-it-yourself downloading. Note that (pre)viewing certainly doesn't prevent you from later auto-downloading our self-unpacking .tar-like bundle [92].



Our use of the word "customer" here is intended to convey the implicit (banking-like) security contract, despite the fact that users today are not charged. Note that our block diagram's 2-way arrows only emphasize actions that are particularly bidirectional; this aspect is *not* critical to understanding the system (every TCP and SCSI connection by definition has some bidirectional flows). While steps such as 1B, 1C and 1D often happen continuously, the *typical* OCR sequence follows:

1A Server authenticates and records user's kiosk login.

1B User scans documents, one at a time, using single-click kiosk scanning.

1C Server continuously polls into kiosk, probing for finished documents.

1D Finished documents are pulled off the kiosk by the server.

1E Server records user logout, transcribing session's "save to" directory.

1F Email receipt notifies of completely processed sessions.

Typically much later, when the customer returns to their home computer:

2A User's browser requests session output, for possible examination.

2B Documents are batch-downloaded for possible Haystack auto-archiving.

Whether we are dealing with an interactive or automatic user, documents and tar files are placed in each user's password-protected web directory, generally only seconds after their creation on the NT kiosk. Output is automatically deleted after seven days. We elaborate on the details in the next section.

Security is more important than one might initially imagine. The reason the ATM metaphor (essentially a secure networked kiosk) was used in the abstract of this thesis is that so many of the documents that we have not yet electronified, are those of a sensitive nature. While HP failed to deliver network security in its HP 9100C Digital Sender (Chapter 3 describes how it sends all documents and passwords in the clear) this lax approach is increasingly untenable. Again, the very documents you tend to scan (and fax) are those sensitive ones that have resisted electronification over many years for legal, fiduciary, judicial and privacy reasons.

5.1 Essential Operation

I outlined the central client-server architecture above – or ATM-bank architecture if you prefer – where the two machines could easily be perceived by the user as one monolithic kiosk. The actual kiosk is the Windows NT-based scanner station whose status happens to be continuously monitored by (and whose output is quickly pulled by) the Linux-based web account server.

Again, this dumb Windows machine is exactly that (not much of an ATM) – these unfortunate security design constraints were described in the previous chapter. All the NT kiosk does is run Capture 3.0 with our Netscape login-box hovering nearby. After logging into the account server's CGI page, users simply click SCAN or hit F9 (once per document) and finally log out back in the Netscape login-box. Like a real ATM kiosk, users should be unable to lock the screen, it should time out users who forget to log out, it should evict excessive users, and of course it should never need to be rebooted. Genuine design constraints so far prevent our prototype from fully enforcing these last golden ideals.

Capture 3.0 may occasionally process the documents in non-chronological order, in extreme cases hours after the user has physically left the scanner. So a document entering the scanner is flagged within seconds and assigned to a login session that "owns it" all the way up to the point when its last sundry outputs are safely pulled into the user's secure web directory.

Samba [27] was used to allow Linux to mount two crucial Windows directories: Capture's "SystemLog" folder and Capture's "Out" folder. Two daemons on the web account server act in concert to continuously poll and pull documents from the NT kiosk. The keystone to this entire process was extracting the names of newly begun scans from Capture's binary log file ActiveDoc.DBF. PERL parsing of this cryptic file was made possible by sticking to Capture's convention of naming all files "OCR_datestamp_timestamp.*"

"POLL" runs every 10 seconds (explained below) watching ActiveDoc.DBF for any newly begun and newly completed scans, each of which are tabulated on a scorecard. "DELIVER" runs at lower priority, waiting for access to the scorecard. Which it uses to pull scan output off the kiosk, and then update the appropriate user's account. Specifically, here's what each daemon does:

> POLL adds any *newly begun* scan to the current login's session on the scorecard and flags it as "processing." When a *newly completed* scan is

noticed, its name is looked up in the scorecard, and its flag is changed to "completed" – regardless of whether the current login is the owner.

DELIVER polls the scorecard for *any completed* scan. It slowly copies that scan's four output formats from the remote kiosk into the appropriate user's secure web directory. It erases the files on the remote kiosk, incrementally tars the new files onto the session tarball and removes the scan from the scorecard. Finally, if the owner session was logged out *and* all its scans have been processed, the daemon packages up an auto-downloadable ".htar" and dispatches receipt/notification email to the owner.

Even while a user is still logged into the kiosk, fully OCR'd documents spool into their web pickup "carousel" where their .tar file for the ongoing session is incrementally built. Documents are moved across the network to the secured web account server as soon as OCR completes. This protects users' output from most computer crashes, network crashes and over-curious NT kiosk users waiting in line behind them. The a priori mentioned limitations of Capture 3.0 prevent us from offering absolute protection of documents undergoing OCR processing. Of course our no-delay updates also provide instant gratification to interactive-style users, who might open a second kiosk browser to examine and usually fine-tune their output.

POLL obsequiously checks for new scans every 10 seconds to ensure tracking of all new scans. This time was chosen to make sure even simple-to-OCR sparse pages are polled at least once as they fleetingly appear in ActiveDoc.DBF (we depend on the fact that even blank documents take about 15 seconds to process given the souped-up OCR and format-creation settings I've set in "AUTO-HAYSTACK"). The interval may be changed to 5 seconds or less when Adobe Capture one day runs faster.

These two daemons (infinite loops written in PERL) represent the core of what's necessary for smooth operation. DELIVER can take its merry time to copy files over: subsequent copying is just queued up, as if we had any other choice during a network traffic jam. This network-shipper even survives network outages intact with the help of its basic transaction semaphores – though a backlog of subsequently completed scans can take a while to transfer after the network reconnects.

Unfortunately, there is no way POLL can provide service during (or recover the damage after) a network outage. Luckily the a priori logged-in user has warning upon logout, as the NT kiosk loses access to the remote CGI logout sequence. This system will be upgraded as discussed below to provide more warning to the user *during* their job – warning that network trouble has arisen and documents may be misplaced. While demOCRacy's daemon system has proved very robust (operating independently for weeks), users will lose an occasional document if the network flaps. Hence our spooling of lost files into "anon"'s public directory (more on this later, and its risks).

While it's very unfortunate that the two critical PC's are separated by 2 router hops, in addition to 2 ethernet hubs (and 4 floors of a building) this unusual network arrangement generally causes far fewer *document retries* than the array of quality control exceptions inherent to scanning and OCR. Still, network topology may change later this year. An ideal network topology would put these two machines on the same subnet with the document/account server isolated behind physical security (eg. a locked closet). This way all previous users' documents will remain protected, even if the public Windows machine is cracked. The best possible solution (for reliability and security) would be to hang another network off the account server using a second NIC Ethernet card, completely isolating all Samba traffic. In future it should also be possible to encrypt all Samba packets, but today Windows NT does not support this.

Cognizant of POLL's 10-second cycle, the kiosk logout sequence is careful to spend 15 seconds logging out to make sure any final documents are tagged for ownership by the correct user session. It was decided that registered users' login sequences should not be delayed however – remember that such logins are equivalent to logging out "anon." So this (regrettably) second-class user does not receive a 15-second logout grace period. Consequently there is a very minimal risk registered users will inherit one of "anon"'s document; this would happen only if you instantly logged in the moment after "anon" began a scan. Today we cannot appease both users, so the anonymous user (whose documents are publically broadcast anyway) is hereby warned not to stick their document into the scanner when a pushy customer is approaching the kiosk (and guard the scanner for 10 seconds after completion to be safe).

It is indeed tempting to move POLL to the NT kiosk to keep the scorecard up to date during network flaps, but this change would necessitate moving the login and logout scripts to the NT kiosk as well, in short merging the bank with the ATM. This would not only violate the crux of our security model (Windows NT must stay logged in as previously discussed, and hence offers no protection), but would represent far more ambitious programming (a number of powerful UNIX calls are made from these PERL scripts and others which are not portable to Windows). Soon we will put a script on NT to warn the user when the network goes down, risking misplacement of their documents.

The glues that hold this all together are the simple PERL CGI forms that process user input: username, password, and the optional/persistent directory for subsequent saving into your home filesystem. All user-persistent account information is stored as traditional dot-files within their web output directory. Apache never lists any dot-files nor does the output-deletion cron job (explained in the manual) affect them.

5.2 Auto-download and Haystack Integration

Our "haystacker" PERL program [35] is the retrieval/unpacking client triggered by your browser for one-click downloading and untarring into the directory you specified at scan-time. It can optionally auto-archive into your Haystack as well, using the Haystack client's "-archive" flag I built expressly for this purpose.

Multitudinous auto-download options were extensively considered, including powerful recursive directory grabber "wget," which unfortunately lacks SSL, and similar but SSL-enhanced tools such as Curl and Pavuk. There is an excellent comparison table of such snarfing tools available online[1] [123].

On Windows two powerful such programs (that still lack SSL) are getright.com and WS_FTP from ipswitch.com. It soon became apparent however that the web

[1] http://www.xach.com/snarf/comparison-table.php3

browser is the universal crypto client, largely due to US government regulations. So in order to support a broad clientele we were going to have to depend on browser-based SSL downloads.

Our haystacker script uses a special MIME type we've created (application/x-htar) that your browser recognizes when beginning the (SSL'd by default) download. Very rigorous security precautions work in conjunction with your browser to visually warn you of anomalous behaviors. As an introduction to our specially formatted .htar files, haystacker's unpacking process goes to great lengths to be strictly non-destructive, both verifying existence of the user-specified directory and pre-testing for even a single file collision. Log files are kept and announced to the user in a small browser popup receipt. The format of htar-1.0 files is as follows:

line 1: htar-1.0

line 2: (the package name, eg. 20000512_215627)

line 3: directory/specified/by/user

line 4: (the .tar bundle follows from this line onwards)

As unforeseen security flaws appear, haystacker comes with a built-in mechanism to alert users of the need to upgrade (and prevent their further use of the flawed version). Currently the user is asked to upgrade if the first line of .htar files downloaded from the DNS-trusted host does not match "htar-1.0".

Providing selective write-access (of personal files) to select others across the open Internet is in general a security and logistical quagmire. In reality Haystack's small user community today protects you in subtle ways. It is a longstanding principle of systems security that as a system gradually proliferates, not only are more security holes discovered, but the assumed security requirements and threat model change. As Haystack and in particular demOCRacy scale into new classes of users, their current security models (not to mention implementations) will necessarily need to be re-addressed.

Until then, users should use protection: we'll all grow old and frail before certain unnamed browser corporations (which provide the only universal client-side crypto) begin including rich yet transparent UIs for downloading and unpacking. A truly standardized protocol similar to Red Hat's rpm [26] might be best – if "save to directory" suggestions were added via client UI overrides. Not even tar and pkzip include such guided client flexibility, though Windows tools such as GetRight [13] and InstallShield [16] have the right idea. The recent surge of online photo processors all face this problem, with no satisfying resolution in sight.

Finally, by (optionally) piping haystacker's outputs into the Haystack client's command-line, we achieve end-to-end paper-to-Haystack integration. The inaccessibility of your paper files can at last give way to convenient Haystack retrieval. Having no access to Capture's API, an annoying difficulty is that our compact PDF and archival/exact PDF outputs cannot be auto-differentiated for a particular document (both having the same file suffix, and similar unpredictable file prefixes).

The larger filesize may not be an absolute guarantee you have the archival/exact PDF rather than the compact PDF of poorly recognized documents. Thus, guaranteed deterministic archiving is impossible without subsequent "typeguessing." Intriguingly Apache has a file typeguesser that indeed looks into the first lines of files, however even this is insufficient as fuller processing of the PDF files is necessary to fully disambiguate here.

This last (optional) step of haystacker takes advantage of a modified haystack.bin.HaystackCL which provides a command-line "-archive" flag to directly batch archive your OCR output. Haystack then creates the essential Tie.Reference ties between these new Document Bales for each file format – and a master Document Bale to tie them all together. Haystacker provides a clear visual browser popup explanation useful for users who do not already have a Haystack root server running. Your paper documents can then finally be disposed, at your discretion.

Security considerations of our client-side downloader are obviously crucial, given rife possibilities for cracker exploitation (or merely accidental pollution) of the user's home filesystem. With haystacker's architectural risks of such critical interest to users themselves, the user's manual (next chapter) contains a "must read" design/security analysis with additional detail relevant to specific attacks.

5.3 Future Directions

After about 1000 page faces were scanned across many users, per-user interfaces are now quite polished with extensive self-explanatory HTML and CGI GUI hand-holding. The iterated feedback of almost a dozen casual users was added (admittedly this was hard when opinions conflicted). All systems can be enhanced (and in this case will be) but most agree demOCRacy (our prototype system) is already a robust, simple operation: pleasant and self-explanatory to use. Our demOCRacy system could most immediately benefit from the several classes of improvements that follow. Note that certain of these upgrades relate to specific technical aspects of our system only discussed in the following chapter (the Manual) [9].

In terms of security, demOCRacy would foremost benefit from:

1. Deleting lost documents instead of placing them in the public "anon" directory. We could also make this a per-user option, expanding the password-changing customer-care page. This could be helpful but not infallible: whose deletion preferences dominate among the two most likely owners of a document? Other per-user options that could be added are (a) disabling of non-SSL access (b) disabling the auto-refresh of web-pickup directory pages and (c) UI choices discussed below.

2. Stress-testing account integrity with new daemons: the system does not crash but again files are misplaced (see above) in select cases. An NT script should loudly warn of network flaps. Server-based timestamps should auto-logout users after a specified timeout, likely an hour of no new document activity.

3. In terms of general security, experienced CGI developers should help fortify our CGIs to vette out security nasties, eg. syscall overflows. A short summary of users' privacy choices should be placed on the kiosk login page so that users incrementally develop trust for demOCRacy. A more experienced NT user could help fortify what is likely our weakest link: NT permissions on the physically-accessible machine.

4. Security policy updates should eventually be considered, such as authenticating users to their email address before their 2nd kiosk login, or integrating registration with existing institutional accounting schemes. Requiring email "*@lcs.mit.edu" or "*@ai.mit.edu" might be a judicious choice if the scanner kiosk is moved into a high-traffic public area. The user account "anon" should (hopefully) be preserved, though it should be locked down so its documents are only retrievable from LCS and AI subnets. In so doing, remember to update http://demOCRacy.lcs.mit.edu/policy.html. Finally, such policy changes do *not* preclude fortifying Windows' security as described just above, which is paramount.

In term of usability (arguably a lower priority, depending who you ask) demOCRacy would most benefit from:

1. Optionally offering more views of users' directories: per-session and or per-document folders. PHP and MySQL could enable voluntary deletion and repackaging of files from users' web directories within seven days, perhaps even while scanning or downloading? Users would love these "quality assurance" features (building a Windows Explorer-like fine-grained packager into their web interface) but it may not be worth the rather extensive effort.

2. On a deeper functional level, we should support uploading of files to be OCR'd. This should not be too difficult an addition, and would support many more OCR uses (though abuse of our remote storage and Adobe software licensing issues would have to be delicately solved). Real metadata extraction is another such addition discussed below.

Capture should be upgraded to release "3.0.1" as soon as it's available (generally expected later in 2000). This should improve stability and speed tremendously [3]. Capture 3.0's API should be explored as soon as it's released, i.e. for much tighter integration, perhaps in 2001. Some of the many smaller tweaks that could improve demOCRacy include:

1. Deleting "anon"'s documents after just 24 or 48 hours, not seven days. This is less of a disk capacity issue than a privacy benefit for such anonymous users.
2. Currently users are notified with the job size in megabytes and time stamp. Snazzier email notifications could include CPU time and network copying time, perhaps one day billing your advisor?
3. Compact PDF could be auto-disambiguated from Archival/Exact PDF using a sophisticated typeguesser. We could port "haystacker" to Windows, etc.

Metadata extraction, one of our original ambitions, relates to those painful but crucial issues of format and filetype. The medical industry just spent 8 years arguing over DICOM and SL7 digital image+metadata formats, delaying digitization a decade before finally settling on the same ungainly formats they began with. The mantra: production OCR systems would benefit from open image standards, supplemented with domain-specific XML metadata languages. So what metadata technique should we use to encapsulate academic papers? A good place to start looking might be BIOSIS, the publisher of the Biological Abstracts and Zoological Records (whose production OCR "lifting" of academic papers has contributed to over 2 million records). The Dublin Core view of metadata politics could be useful too.

In our case, we wanted academic paper abstracts and authors, but how to incentivize this and other metadata extraction? Such extraction is always hard. While we failed to integrate even a crude version of this into Haystack, abstract and authors are at least now more available (than they were on paper) for unstructured Haystack indexing. Unfortunately Capture 3.0 doesn't support explicit recognition of title and author, but at least email, URLs and Tables of Content (when recognized) are embedded regular hyperlinks. At the bare minimum, many vivid forms of metadata (date, size, etc) are made human-viewable – and re-sortable with a single click – thanks to Apache's classy directory listings.

Again, specific services should now be written to provide value-added to the OCR output. Here, I believe there are still plenty of design opportunities for Haystack, if only as image hoarding initially, and later targetting more semantically extractable / scannable real world artifacts.

Our Fujitsu scanner doesn't provide color unfortunately, but simple image albuming, thumbnailing and databasing could be a very useful addition to Haystack. One study [75, BusWeek99] projected that Japan would take 6 times as many total pictures in 1999 compared to 1998, due to the explosion in digital photography. While personal photos and images are not the focus of this scanning project, it would be very wise to build in groundwork for personal [imagery] collections, as this is what increasingly draws people to scanners today.

Chapter 6

The Manual to demOCRacy

This is a preliminary version of demOCRacy's manual. An up-to-date version is kept at [9]:

http://demOCRacy.lcs.mit.edu/manual

6.1 Scanner Maintenance

Please see the book "OCR With a Smile: An Operator's Guide to Optical Character Recognition" [64, Ross98] for solid operation advice. Please see Chapter 3 (OCR Tools) early in this thesis for characteristics of our Fujitsu 3093DG scanner and its professional scanning market context. The Fujitsu's manuals themselves [11] are kept right next to the scanner.

Users should remember to taper their document (so that its profile appears like a parallelogram) if feeding a large stack or paper of unusual thickness through the feeder. These simple instructions are clearly printed on the ADF (automatic document feeder) so this and other paper alignment issues are straightforward. A tiny replacement bracket has been provided if the machine later begins to double-feed excessively. The bracket is stored in its tiny cardboard box next to the scanner; more can be ordered if necessary.

If you don't clean the scanner paperpath regularly, paper will begin to double-feed and jam. That's what Fujitsu says about its 3093DG anyway; so far only about 1000 sheets have pass through so I wouldn't know. Cleaning is apparently important, but the freebie cleaning kit Fujitsu promised to send by mail still hasn't shown up after repeated calls. Again, surprisingly: there have been absolutely no mechanical snafus so far. The uptime problems have all been caused by *software*, be they PC crashes (nothing you can do about NT4's blue screens of death) or the fragility of Adobe's new Capture 3.0 software.

If necessary, take advantage of the one-year on-site warranty that will expire in February of 2001.

6.2 NT Maintenance (ocr.ai.mit.edu)

Simply reboot the NT scanner station whenever it crashes. The "blue screen of death" will unfortunately become familiar to frequent users. In fact it is necessary to tape clear instructions onto the machine as this happens as often as weekly or more, unfortunately. This is odd considering we did a clean install of Windows NT4 (Service Pack 6), Adobe Capture 3.0, and Netscape 4.72.

It is plausible that the blame for the much increased crashing lies not with Microsoft, but with our addition of a 256MB memory card. The evidence is inconclusive however, as I (simultaneously) upgraded memory when I downgraded from Windows 2000 to NT4. However, two different NT gurus who examined the operating system's log files suspect memory is not to blame – they suspect Microsoft (NT4) or Adobe (Capture 3.0) are more likely to be at fault here. Memory crashes have been significantly reduced in months since.

The Windows NT box is an AMD Athlon (K7) PC running at 650MHz, initially with 192MB of memory and later upgraded to 384MB so we could throw more heavy documents at it simultaneously. Included was a large cheap disk (27GB) but it turned out not to be so necessary given the final architecture chosen: documents are served to customers off the higher-security Linux companion PC (demOCRacy.lcs) whose maintenance is summarized below.

If this kiosk machine is ever moved from the AI to the LCS subdomain, it would be best to name it ocr.lcs.mit.edu so that Capture continues to name output files correctly (OCR_datestamp_timestamp.*). This name has been reserved using WebDNS[1] to guarantee this will be possible.

6.2.1 Windows OS's and Drivers

Complete OS and driver disks are stored in a labeled (keyboard-shaped) brown cardboard box on top of the computer ocr.ai.mit.edu.

Unfortunately four different OS's were painfully installed (Windows 98, Windows 2000, Windows NT4 and Red Hat 6.2) and often reinstalled on different machines until the successful combination was finally arrived at (NT4 at the scanning station and Red Hat 6.2 on the account server). After upgrading Windows 98 to Windows 2000 on the advice of an Adobe representative, I later downgraded from Windows 2000 to NT4 (Service Pack 6) because of frustrating and untraceable Capture 3.0 problems – notoriously the package's total failure to email out documents via SMTP, which was being considered at the time. Unfortunately the subsequent reduction in Capture crashes was offset by an increase in NT4's blue screens of death.

General Advice: don't forget to install the Fujitsu ISIS scanner driver. The TWAIN driver is not necessary. Enough said: if all else fails, keep checking manufacturer's web page, installing drivers, and rebooting often.

6.2.2 Capture 3.0

See chapter 3 for an overview of this gargantuan OCR software package. Print out a copy of both the introductory guide and manual [34] (both in PDF) to learn about workflows and its excellent quality-control touchup tools. Support is available from the popular mailing list [3] which has kept complete archives over the years. Adobe will answer certain tech support questions during our first year (until February, 2001) if you call 800 272-3623.

[1]MIT LCS and AI departments' name registration service: http://webdns.lcs.mit.edu/cgi-bin/webdns

You can also use this number to order new dongles when our 20,000 sheet quota runs out. Remember that the registration code is original to our copy of Capture 2.0 – look in the Capture 2.0.1 cardboard box right by our scanner (this purchase code cannot be included in this MIT-distributed and web-published thesis). Upon installing or upgrading Capture, the dongle often fails to be recognized, but as usual reboots are the universal remedy.

6.3 Linux Maintenance (demOCRacy.lcs.mit.edu)

Our web account server's[2] [37] daemons (deliver.pl and poll.pl) are launched automatically from /etc/rc.d/rc.local on reboot, so nothing more is necessary. Daemons have never crashed or hung, but the commands "ps ax" and "kill" are useful to make sure they are still running. The more forceful "kill -9" is not necessary. Note that when rebooting this (and other Red Hat 6.2) machines it is often necessary to type "reboot" *twice* in order to force the disconnection of RPC/mounts.

Take note that Samba very confusingly announces "session request to OCR.AI.MIT.EDU failed" whenever it *succeeds* in mounting one of NT's shares (i.e. one of the two directories "SystemLog" or "Out"). Far more disheartening messages are displayed when the remote disk mounts *genuinely* fail. The details of these mounts are currently stored in /etc/fstab. There is a chance mount details may be moved to within the daemons themselves in future.

/etc/cron.daily/gerontOCRacy is a predictable script. It discovers user files that are too old and executes trash removal, all as a nightly cron job. The UNIX command "find /www/voters/*/* -mtime +8" is used to identify all files older than 8 days, carefully preserving the state of demOCRacy's dotfiles in each user's directory. All such stale documents are deleted, which will generally tax the load of the machine for a few seconds to a few minutes whenever it runs at 4:02am (Red Hat 6.2's default).

The 5 dot-files in each user's directory are:

[2] aliased to pochard.lcs.mit.edu

.htaccess - apache security configuration

.htpasswd - user's password (encrypted)

.directory - user's persistent directory

.HEADER.html - announcement prepended atop user's viewable web spool

.README.html - contact info appended below user's viewable web spool

Poorly recognized pages may occupy more than 300kB per PDF. So it is theoretically possible to overload demOCRacy's disk (8GB) in a single (long) day of non-stop OCR usage (before the 7-day auto-deletion bot kicks in). *All output formats together* typically occupy little more than 1MB per page, meaning that our current server could potentially cache as many as 8000 pages. Such usage exceeds the maximum daily duty cycle (3000 pages or 6000 images) of even our departmental scanner. For now, this is a very distant risk. It is a risk very puposefully chosen – in order to store *each* file format in triplicate, i.e. individually and within the .tar and .htar bundles (in the end, every document spawns a total of six PDF files, six HTML groupings and three text files).

Should users override the default Capture settings, for example creating high-resolution grayscale images, they quickly inflict heavy network and processing demands on our system. For example I scanned a 400 dpi image (the maximum for grayscale) whose two resultant PDF format flavors were both about 13MB. Each of these two was replicated three times for package/download purposes for a total of 78MB for a single face of a single page.

Maintainers should remind themselves of our file format choices (discussed in Chapter 3), before carefully tweaking our bundle of file formats. Generating six heavy PDF files per document may seem very wasteful. This is especially true from the point-of-view of professionals such as LASON who often compress all page data down to 30kB of less (basic images included). Certainly, TIFF+text old-timers (eg. my Kofax OCR dealer) consider all PDF files to be incredibly wasteful. Again, it was our conscious decision to cater to diverse users' formatting whims, each of which (generated formats) should be effortlessly viewable and/or quickly downloadable.

Our 8GB disk was not meant as a storage service; it is merely a cache for users' short-term convenience. But if the disk ever reaches capacity, it would very likely be most cost-effective to buy a much bigger disk. Or we could make our gerontOCRacy deletion-bot more strict than it already is. In truth, any number of policies are possible, such as changing to per-user quotas based on things like file count, system impact, or user seniority.

Security administration doesn't come for free, as Haystack users have come to realize with even our most recent Linux machines recently cracked and corrupted by distant hackers. So the days spent installing, tuning and securing Red Hat Linux 6.2 on our hub account server were necessary to make the final system more robust. Thankfully recent versions of the Apache web server are powerful and easy to maintain, so administration headaches were contained.

6.4 Auto-downloading with "haystacker"

All you have to do to configure our auto-downloader is to set your browser to run "haystacker %s %u" upon encountering files of MIME type application/x-htar (which should end in .htar suffixes). Download haystacker from:

http://demOCRacy.lcs.mit.edu/haystacker.html

The security impact of using haystacker is very significant. Still, users should glance at Chapter 5 (Architecture) to see how the top three lines of .htar files protect them. If nothing else, be careful with relative paths as the files are dumped in a directory relative to where you launched your browser. Stick to absolute paths if you have the habit of launching your browser from random directories. We could just as easily have forced relative paths with respect to users' home directories, but decided in favor of full flexibility for now. While haystacker's unpacking process is strictly non-destructive (it does an awful lot more checking than just tar's -k flag), it's useful for users to know that it:

1. insists that the destination directory already exist

2. insists that none of the files in the package collide with existing files
3. carefully writes a [job].files to /usr/tmp

Despite this paranoia, a naive user can still of course accidentally pollute directories they care about, with untold clutter or perhaps worse, flood their disk/quota. It all depends how you define destruction. And clutter.

As a consequence over 90% of the roughly 150-line PERL script is (1) a popup to explain to you exactly what was unpacked where and (2) comments, so you can verify the script does what (little) I say it does. While our email notification service reminds you which directory you chose to download in, a truly malicious party would try to take advantage of your habits by falsifying an email ticket from @demOCRacy.lcs.mit.edu in order to "social engineer" you into flooding your disk.

Another annoying D.O.S. (denial-of-service) attack on your disk is theoretically possible, especially given that Netscape supports falsifying the URL in the browser's Status Bar using Javascript. To make things worse: Netscape also ignores the file suffix. So an active attacker could theoretically fill up your home directory with child porn by tricking you into clicking on childporn.htar, disguised as an innocent web link. Our current SSL implementation includes a server-side certificate that eliminates one class of such UI habituation attacks. This is insufficient however, as your browser launches the MIME handler (haystacker) regardless of the certificate.

Haystacker safeguards against all the above attacks by checking the URL that your browser passes it using the "%u" flag. This guards against all but the most sophisticated DNS (name server) infrastructure attacks. Haystacker will only process URLs that case-independently match "https://demOCRacy.lcs.mit.edu/*" or "http://demOCRacy.lcs.mit.edu/*". In addition, users can hard code their email addresses into the first lines of the haystacker script (a subtle benefit of our abnormal URLs that contain email addresses). This guards a user's filesystem against certain attacks from other users of their same server. This supplements our Apache server logs, which generally serve as a strong deterrent against such internal-style attacks.

6.5 Backups

Backups of demOCRacy.lcs.mit.edu's code and configuration files will be stored in matching directory names (www* and cgi-bin*) within the directory "demOCRacy-Backups", under hayweb@theory's home directory.

Chapter 7

Educational Value

Today, the primary purpose of IP (the Internet Protocol) is to ship IP (Intellectual Property) – regardless of what its 1970s designers had in mind [70, Hafner96]. Who is writing the rules and policies for this new e-copyright regime? How will OCR, Haystack and agent-like sharing technologies affect the right to read [88, Litman94] [118, Stallman97] in this Brave New World? This chapter probes below the surface of this critical issue, as technology and law collide, and merge, across many fronts [84, Lessig99].

By connecting the dots, it is not difficult to anticipate many natural consequences of OCR technology, as I do here. Keep in mind however: all readers are advised to consult an experienced lawyer (or better yet, write a Ph.D.) if they want to finalize answers here, rather than just raise questions.

A recent study by IDC [40, IDC99] claims that 55% of all typing on PCs is actually retyping of data already on paper. Other studies suggest that 90% of all information is still on paper even many decades into the computer age.

Paper represents a medium implicitly associated with many costs, rights and expectations, legally and socially. Occasional photocopying and faxing are today both part of that picture, as well as timeless affordances such as mould/mildew, bloody papercuts, red tape, bureaucratic paper pushers, armored cabinets, shredders, book-burning, draft cards and of course, dumpster-diving.

Truly many documents remain offline for no other reason than the law typically requires preservation of the paper originals. Regardless, ubiquitous cheap portable OCR would radically alter this stable panorama – a medium still richly endowed with high social expectations of confidentiality, interwoven with a long history of permanence yet also transience.

7.1 Cut and Paste Reality

What could be more classically educational than a cutting and pasting project for kids? Yet RealVideo takes advantage of Windows APIs to make it impossible for students to cut and paste even static images from short video clips. Both Windows' "screendump" and "window dump" functions (each command uses the Print Screen button above the arrow keys on your keyboard) have been effectively disabled. Why hasn't Microsoft built in OCR-like rights to cut and paste (and similar client-empowering technologies) that would let students creatively interact as more than passive edutainment consumers? Ubiquitous consumer OCR could be a powerful tool for kids learning to read everywhere – but much like a calculator, it could of course be used to cheat as well. More immediately: how best to deliver this to learners everywhere?

Modern software makes it impossibly difficult for reasonable adults to keep their own copies of the "I agree" licenses they must click on to even begin to use commercial software. OCR would at the very least let you keep a searchable copy of these non-negotiable adhesion contracts [45] that anyone who uses modern software is forced to "sign" [103, Kaner98] (despite the fact both the IEEE and the ACM publically oppose such contracts [105] [60]).

Legal reasons for paper preservation notwithstanding, John Seely Brown (chief scientist of Xerox and Director of Xerox PARC) believes that the Internet will not lead traditional institutions to demise, as widely predicted. Brown's recent book[1] [55, Brown00] argues that structures often serve valuable but unnoticed functions, social and otherwise. Similarly, paper continues to provide many classic affordances still

[1] "The Social Life of Information"

missing online, not the least of which being seamless annotations (eg. notes in the margin or highlighting).

Is OCR a munition? If so, an incredibly benevolent one – even the $99 OCR pen is mightier than the sword. OCR will no doubt become part of the content arms race raging between Hollywood, Silicon Valley and D.C. as we speak. Current disk trends easily support the ability of individuals to OCR every bit of text writing they see during their entire lives, from tombstone inscriptions all the way to false advertisements you saw on the way to work that you might need to check up on later if...

Xerox/Scansoft goes so far at to advertise its OCR products with the tantalizing slogan "LOST YOUR ORIGINAL?" Of course arm dealers have always sold to both sides in any war, but will individuals have the right to "Xerox" things they see? OCR is increasingly both cheaper and faster than a secretary (even the army of Chinese phone directory typists hired by insurgent American telcos over the last decade are gradually being phased out as OCR improves). So why not empower everyone [115, Shapiro99] with near-photographic textual memory?

7.2 OCR Hurts Publishers

I personally would not hesitate to OCR newspaper articles instead of storing the yellowed clippings I now keep. The "fair use" ethic (and law) commonly associated with photocopiers is now deeply embedded in Western society: "Who are they to tell me what kind of filing system I should use for a newspaper that I paid for?" If the trickle becomes a flood I'm sure the newspaper company will opine otherwise – and if "OCR Xerox machines" really emerge in cellphones, will electronic books really be able to secure their content?

These fears continue do draw publishers to novel locks such as microbarcodes, e-books, set-top boxes, trusted systems as well as "digital rights" euphemisms and markup languages such as XrML (extensible rights markup language). Dissemination is dogma for the research community but unspeakable to the accountants tallying the bottom-line. Paper and other analog containers simply can't continue to guard information.

Article I of the U.S. Constitution authorizes Congress to:

> "promote the Progress of Science and useful Arts, by securing for *limited Times* (emphasis added) to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." [31]

But dozens of companies such as intertrust.com and contentguard.com (the late April alliance between Xerox and Microsoft [6]) are setting out to replace this delicate balance with their own technological containers – where e-copyrights never expire [71, Hamilton96].

One such technology is watermark-based policing ("tattletale technology") that, like OCR, is becoming part of the e-content arms race. Watermarking uses clever mathematical techniques that survive Digital to Analog to Digital conversions and more. At its simplest, watermarking is used by the White House when confiding secret documents: leaks can later be detected based on the slightly different wording given to each party.

As more sophisticated e-bottles come to market they will need government protection because (1) Intel recently announced that it will no longer include serial numbers in its x86-class CPUs (2) watermark-based policing is not yet fully mature. More proprietary platforms (such as cellphones and Palm Pilots) support far more traceability of the piracy: each user's device has a separate unique identifier. The source of a leak can then be fined and/or jailed. This is a crucial difference according to Xerox PARC researcher Mark Stefik who pioneered ideas for trusted content systems, who believes [90, Stefik98] these schemes cannot work unless they embed such hardware dongles.

The DMCA [48] (Digital Millenium Copyright Act of 1998) amended the Copyright Act to buttress such trusted content systems against "circumvention" technologies. Similar "anti-circumvention" laws are being enacted worldwide – despite the fact that trusted content networks still don't exist online to this day. This is a very rare case where technology trails the law, due to intense lobbying by the US record and movie industries.

The very long arm of this law was recently demonstrated when foreign pressure caused Norway to prosecute a 15-year old Norwegian boy for writing a simplistic program (now printed on T-shirts) that reads DVDs. This highly controversial law [112, Samuelson99] and its anti-circumvention clause ban many forms of exploratory hacking and reverse engineering that were traditionally legal. While some legal experts believe that OCR will be banned [80, Kim97], extensive searches turned up few conclusive answers (or even speculation) on the future of OCR law. It's far too early to know how courts will interpret the constitutionality of the DMCA, particularly as to how it regulates dual-use technologies such as digital photocopiers.

Web users consistently demand their news fast and free. Could OCR guarantee just enough leakage to guarantee certain public access? Former Harvard mathematician Tom Lehrer is famous for singing "Plagiarize! Let no one else's work evade your eyes... Only be sure to always - to call it, please - research." Yet all modern knowledge is incremental value-added on top of previous peoples' work[2] – it's but one unfortunate aspect of human nature to take credit without attributing.

Knowledge recyling has a bad rap: "advice is something you fish out of the trash, wipe off the ugly parts and recycle as new" according to Baz Luhrmann. Publishers large and small have little patience for empowering personal caching technologies because they represent a loss of control. So many digital rights architectures promise novel (re)bundling and (re)marketing at the expense of personal privacy, including Mark Stefik's trusted systems [119, Stefik97] and Tom Bell's "Fared Use" [50, Bell98].

But some loss of author and publisher control is inevitable, and in fact necessary, to protect parody and whistleblowering rights [65] for example. The competing interests between writer, publisher and reader were vividly illustrated by my scanning and OCRing of MIT's sailing manual, which sells for 50 cents. It will soon be freely available for the world to view [30] but tellingly this decision came only after

[2] "I have only seen so far because I have stood on the shoulders of giants" –Isaac Newton, in a somewhat out of character moment

protracted internal discussions as to how much access was appropriate, who should inherit control/responsibility/ownership, etc.

In short, the tension between information wanting to be (1) freed, and on the other hand (2) price discriminated has been ongoing for years [54, Branscomb94]. Of course information wants to be shared (or else it has no value) and certainly Thomas Jefferson's metaphor of ideas as a taper (the wick of a torch) reminds us that giving away ideas certainly doesn't deprive the giver of their original idea. But experience goods have production costs too, and if they are all sold at marginal cost ($0) it's undeniable that we've all got a problem [116, Shapiro98]. In the new fiberspace, many small publishers face the cruel Hobson's choice: "sell water for $5/gallon or go out of business?" Compromise architectures will be battled over for years [57, DiBona99].

Hopefully micropayment-based schemes [109, Rivest96] such as transcopyright [97, Nelson97] and superdistribution [95, Mori90] will not be crowded out of the standards market as publishers try to uninvent the "world's biggest copy machine" (an old nickname for the Internet).

Once, the expression "everyone is a publisher" described a hopeful view of the Internet; today that view is increasingly Panglossian as the network of networks begins converging with "the [television] networks". Today the U.S. Library of Congress increasingly uses OCR on abstracts and tables of content [104]. But as publishers rearchitect the Net using Silicon Valley and D.C. to shape its code towards their desires, it would be tragic if educational and academic uses continue to be overlooked in this rush for control [85, Lessig00].

Innovative publishers want to turn "consumer OCR" to their own advantage. For example The Post and Courier in Charleston, SC is giving away free pen-style scanners [69, Guernsey00] that work with tiny codes printed next to headlines in its print editions. This is a step beyond barcoding your customers themselves (essentially what all modern grocery stores do with their club cards). Readers are taken directly to highly personalized web sites based on the articles they've waved over with the magic wand. A powerful educational tool yet one has to wonder: what else will the newspaper do with the barcoded article-by-article thought-trails of its customers?

7.3 OCR Hurts Privacy

One day a paper shredder might be combined with a classic feeder-based scanner – if quality control ever approaches human levels – to at least provide privacy to the user. An MIT AI departmental shredder is located right down the hall for from demOCRacy (our protoype system) for similar reasons. But what of the privacy of others, eg. the many investors whose social security numbers were widely propagated when the U.S. Securities and Exchange Commission electronified its documents [66, Garfinkel00]?

Already certain classes of documents are, in fact, designed expressly to defeat OCR "digitization." MIT Media Lab members sometimes scan artsy business cards onto their web-pages to prevent machine-readability by various crawlers fishing for spam victims. Cheap accessible OCR, quite like MP3, would further threaten copyright interests (and privacy), leading to techno-legal restrictions internationally. Even today companies like HP and Xerox are said to to be working with the US Secret Service to block scanning and/or printing of paper currency (such printers are already sold in Japan).

While fostering registration databases of course threatens privacy interests [66, Garfinkel00], long term threats might more directly invade the privacy of the papers lying on your desk. Hypothetically future visitors to your office may be able to permanently record every title on your shelves and every memo lying on your desk with a single digital glance. In the short term, both watermarking and recognition software are in their infancy, so fostering cultural acceptance of digitization should only lead to a wider exchange of ideas.

Yet viable business models for usage or bundling of OCR products remain unclear. For example, one could hypothetically imagine notarizing a photograph of one's actual home bookshelf to be eligible for discount "upgrade prices" when viewing previously-bought books electronically. Publishers would favor this if it permits them to amortize the cost of a much higher quality scan across all readers – who unfortunately lose their valuable margin annotations. Record companies are now trying the same centralization strategy to fight back against MP3. Not incidentally, this permits publishers to look over your shoulder, acquiring detailed user profiles.

No the CIA can't yet OCR your car's license plate via satellite. But your license plate is OCR'd whenever you drive across US borders as well as on modern international highways such as Toronto's 407 [63, Feng00] where highway-speed video/OCR tolling mounted on gantries above the roads has been used since 1997. Regular electronic tolls such as the Massachusetts Turnpike's FAST LANE and the entire U.S. Northeast's EZPASS now videotape all license plates of cars that use their new systems. U.S. Customs Service contractor perceptics.com promises [23] that with their license plate readers, "every highway is an open book." This Orwellian picture is deeply disturbing to many – yet would be welcomed in countries like Colombia where kidnappings are a daily occurance, and police are in short supply.

OCR could even change office water-cooler dynamics. You might no longer be comfortable giving sensitive documents to colleagues in an age where powerful Xerox machines sit on every desk. A telling example of this occurred when I OCR'd my girlfriend's resume, which instantly appeared on a web site to her chagrin. Now she is more careful when visiting the OCR room: Orwell's Character Recognition?

In a milieu where almost 90% of surveyed Americans say they've lost control of their personal information, it is only appropriate for people to be skeptical towards one more example of an innocent input device that could oneday grow into round-the-clock surveillance. This is sad given that OCR, if widely deployed, could be so intellectually empowering.

7.4 Regulating OCR Rights

Given that so much of today's esoterica comes with preprinted URLs emblazoned on its packaging, some wonder whether OCR/scanning is still necessary. This popular perception that "OCR is no longer necessary" once we cross the digital divide is shortsighted for many reasons. (1) Most of the world's people are not as URL-crazy as Americans , and will not be for years. (2) Magazines like InformationWeek no longer print URLs under each article as they did 3 years ago. (3) Pre-web documents don't have URLs attached. (4) URLs degrade faster than paper today. (5) Paper confers many reading-without-surveillance rights that are not preserved online. (6) URLs map to different web pages depending on your country domain, your time zone, your cookies, your IP address, etc. Self-tagged URLs are not always trustworthy: even New York Times stories sometimes differ offline and online.

Who knows if OCR advantages will outweigh the drawbacks? On balance I believe that the profound educational potential [79, Katz99] of textual OCR outweighs its harms to publishers and privacy. Savvy lawyers may try to force test cases to trial affirming one's right to record digital images under First Amendment protection (Julie Cohen is one of many legal scholars that argue [58, Cohen00] that the right to free speech implies a certain right to read). First Amendment lawyer Martin Garbus is attempting precisely this defense of fair use – based on free speech principles – in the ongoing DeCSS trial [36] (so far unsuccessfully, but it's now appears destined for the U.S. Supreme Court).

Instead of OCR'ing vehicle plates (as perceptics.com does at all US border crossings [23], and is increasingly part of automated highway tolls) what would happen if the playing field was leveled so that cars could also OCR the world around them? Would I then ask my car to read road signs aloud – that I happened to miss – just like a conscientious companion sitting in the passenger seat? Would I ask my car to record the license plates of all hit-and-runs, whether I'm in the car or not?

One way to lay the groundwork for such legal rights is by seeding open source PDF and OCR projects. In the short term, the best strategy may be to interface with proprietary software. Indeed, working around a hostile API is one of the most common problems faced by system integrators. Later, providing future-proof open interfaces could incentivize other metadata extraction, perhaps matching or outdoing Scansoft's 12 languages of Capture's "20 dictionary" support.

With the OCR industry rapidly consolidating (onetime leader Caere acquired both Calera and Budapest's Recognita shortly before itself being bought ought by Xerox/Scansoft in Jan 2000), choice is diminishing. Innovation on the OCR fringe is at risk; it's time to bring competitive alternatives back to the market.

While inherently an antagonistic technology for info-producers (all technologies affect balances of power) OCR's outputs are semi-structured snapshots almost by definition, and recognition algorithms will always follow months if not years behind the layout artists designing original content. This may represent a crucial legal difference between OCR and other sharing tools like napster and gnutella, which make perfect digital copies.

Courts may take this into account when they eventually pronounce opinions on OCR. The duration, scope and limitations of copyright never cease to evolve, well beyond the U.S. Copyright Act of 1976 [32]. Does OCR output resemble altered "derivative works" such as a student's scribbled notes taken from a professor's blackboard? Exactly how much mutation and/or degradation qualify you for derivative work protection? Such analog bootlegging is frowned upon but often not as nefarious as digital bitlegging – it has not only been tolerated but sometimes explicitly sanctioned under years of "fair use" case law.

Regardless, fair use is always adjudicated on a sliding scale of subtleties like educational purpose, length, market effect [86] and of course nation and jurisdiction. Binding legal precedents for online works are still forthcoming [87], but as a dual-use technology (OCR being a lifeline for both blind people and digital libraries) the case for restraining OCR will necessarily be all that much harder.

A certain precedent has also been set with non-OCR scans of copyrighted documents being publically permitted online through the 1990s. While rarely qualifying as derivative works, this again suggests a possible interim compromise where bitmaps may retain more fair use access rights – if the client is forced to do the recognition themselves. One can stretch this to the realm of the absurd: remote library systems might allow library users to control "robots" or at least conveyor belts to do live remote scanning for them. Business Week's very recent editorializing against the DMCA's anti-circumvention clause [52] offers some hope for the modern salvation of fair use.

In the end of course, it's about nothing more than financial interests, technological costs of enforcement, social mores, and who controls society's policymaking processes. While the ACM condones authors' rights to OCR – so long as notice is given that recognition may be imperfect [41] – popular OCR could again be restrained by laws such as the draconian Digital Millenium Copyright Act of 1998 [48] if the technology becomes too accurate and accessible.

The DMCA is still very open to interpretation, but as a crude example of machine vision, precedents set with OCR will be far-ranging [81]. The word "scan" has profoundly different (and contradictory) meanings, much like the loaded word "paper." What will these two words mean to us in 2025? Over time both technology and law will have to carve out deeper protections for memex's (human memory extenders [101, Smith91] [122, Zachary97]) in general, be they Haystacks, AOL profiles, IMAP email archives or whatever. The privacy law proposed by the U.S. Federal Trade Commission in late May, 2000 offers new hope in this direction.

Chapter 8

Conclusion

OCR "document understanding" has recently improved significantly. Yet OCR is still the tool of insurance companies, legal departments, border crossings, and increasingly highway tolling. But the economics could soon change, and the explosion in consumer digital cameras could in future alleviate the input conundrum, in particular the slowness of flatbed scanners.

This transition will entail sidestepping many of the labor-intensive traditions of the perfection-oriented OCR business. By minimizing the quality-assurance costs of archiving today, our fully-integrated self-service kiosk confirmed that production OCR can be made nearly automatic for convenient general use. The incumbent document quality degradations were manageable for many Haystack applications.

Our limited user testing confirmed that a leap in OCR convenience can radically alter users' behavior and attitudes towards paper. Such tools may become powerful triggers for personal intellectual growth – even imperfect OCR unifies searchable literary artifacts of all stripe into your Haystack, instead of leaving them buried under a pile of books in your attic.

Yet property and privacy concerns abound, as the values coded into all technologies are guided by the biases of those who pay the piper. Xerox/Scansoft, Adobe and Microsoft each give lip service to education but all three sell products that block cutting and pasting. As they continue to orient their businesses around pay-per-view ebooks and privacy-busting, entertainment-oriented content locks, the "Portable Document Format” may one day be used against (instead of for) OCR.

Two open source OCR efforts recently dissolved, despite the greatest of intentions. Yet unlike cryptography, the core technology of OCR is so old that foundational patents have expired. Every effort should be made to resuscitate such projects – even a partial success will force vendors to quickly build more fair-use into their products – or risk irrelevance. Unless OCR is ruled illegal under the Digital Millenium Copyright Act (1998), now is the time to fight for your legal right to OCR.

Bibliography

[1] Newsgroups useful for comparison shopping and debugging: comp.periphs.scanners, alt.comp.periphs.scanner, comp.ai.doc-analysis.ocr and comp.text.pdf.

[2] Acrobat Capture API Reference. WWW documentation. (only the 2.0 API is available as of this writing) http://partners.adobe.com/asn/developer/technotes.html.

[3] Adobe Capture Mailing List and Archives. WWW resource. (popular and priceless) http://www.pdfzone.org/cgi-bin/wilma.cgi/capture.

[4] An As-Yet-Unnamed OCR Project. (open source OCR project that has officially 'fallen into a coma') http://starship.python.net/crew/amk/ocr/.

[5] CD Dimensions Inc. (excellent high-end Scanner Comparison Table) http://www.cddimensions.com/document_scanner/.

[6] ContentGuard: the catalyst for the revolution in eContent. (Microsoft / Xerox content-lock spinoff) http://www.contentguard.com.

[7] Create Adobe PDF Online. (first three conversions are free) http://createpdf.adobe.com.

[8] demOCRacy Accounts Policy. http://demOCRacy.lcs.mit.edu/policy.html.

[9] demOCRacy's Manual. http://demOCRacy.lcs.mit.edu/manual.

[10] Designing Better Systems Based on Business, Policy and Social Goals. (MIT class) http://ecitizen.mit.edu/seminar1.htm and http://ecitizen.mit.edu/ecap.

[11] Fujitsu 3093DG Scanner Manuals. (Included with scanner purchase. Ourselves, we keep them right next to our scanner).

[12] Fujitsu Scanners: Workgroup, Departmental and Production. http://www.fcpa.com/product/scn/scn_cat.html.

[13] GetRight. (Windows downloading tool) http://www.getright.com.

[14] HP CapShare - handheld electronic copier for portable computing. (A promising infocapture appliance) http://www.capshare.hp.com.

[15] HP Digital Sender. (SMTP/email-based Workgroup Scanner) http://www.digitalsender.hp.com.

[16] InstallShield. (Windows packaging and downloading tool) http://www.installshield.com.

[17] Integrating Paper into Lotus notes Applications Using WebArchive and the HP 9100C Digital Sender. WWW publication. (Visions of a Paperless Office, p2), http://www.pandi.hp.com/pandi/pdf/digitalsender_intpaper.pdf.

[18] LASON: The Information Management Company. (International professional scanning service, whose liaison Peter Berry at the Needham, Massachusetts branch was very helpful) http://www.lason.com.

[19] MIT Theses Online. (WWW publications which use PDF) http://thesis.mit.edu (joined the Networked Digital Library of Theses and Dissertations in March 2000, http://www.theses.org).

[20] National Federation of the Blind: (Ray) Kurzweil Honored. WWW publication. (Kurzweil's Reading Machine was called the most significant advance since Braille in the 19th century) http://www.nfb.org/bm000311.htm.

[21] PDF Accessibility Information and Resources. WWW publication. (Adobe announced its intentions to make PDF more accessible to the disabled on April 18, 2000) http://access.adobe.com/information.html.

[22] PDFzone.com: The online authority for Acrobat, PDF and Document Management Professionals. WWW resource. (Excellent PDF Resources) http://www.pdfzone.com.

[23] Perceptics' License Plate Reader. Product description. http://www.perceptics.com/lpr/lpr.htm.

[24] Pixid.com's Whiteboard Photo Software. WWW publication. (take notes with your digital camera, reviewed at) http://www.dcresource.com/specials/WhiteBoardPhoto/.

[25] Planet PDF. WWW resource. (Up-to-date PDF News) http://www.planetpdf.com.

[26] RPM. WWW documentation. (Linux packaging and downloading tool) http://www.rpm.org.

[27] Samba. (Windows-Unix disk-sharing utility) http://www.samba.org.

[28] Scansoft Inc. (The king of consumer OCR, a Xerox affiliate, has now acquired its onetime chief competitor Caere) http://www.scansoft.com.

[29] ScanSoft SDK version 5.0. WWW documentation. http://www.scansoft.com/products/sssdk/.

[30] The MIT Sailing Homepage. (their sailing manual, OCR'd using demOCRacy, should be posted soon at) http://www.mit.edu/˜mit-sailing.

[31] U.S. Constitution, Article I, 1787. (limits duration of copyrights and patents) http://www.law.cornell.edu/constitution/constitution.articlei.html.

[32] Copyright Act of 1976 [Title 17, United States Code], 1976. http://www.twsu.edu/library/specialcollections/c1.html.

[33] Public Domain OCR Resources, 1999. http://documents.cfar.umd.edu/ocr/.

[34] Adobe Acrobat Capture 3.0 Documentation, 2000. (introductory guide and manual available in PDF form on the software package's CD, available from) http://www.adobe.com/products/acrcapture/main.html.

[35] haystacker, 2000. (demOCRacy's auto-downloader) http://demOCRacy.lcs.mit.edu/haystacker.html.

[36] Martin Garbus interview, 2000. WWW publication. (argues for modernizing fair use rights based on First Amendment principles) http://www.feedmag.com/re/re340_master.html.

[37] Welcome to demOCRacy!, 2000. (Haystack's secure account server for OCR, and documentation) http://demOCRacy.lcs.mit.edu.

[38] MIT Technology Day (The Future of Atoms in an Age of Bits), June 3, 2000. (Overview:) http://web.mit.edu/newsoffice/tt/2000/may17/techday.html, (Agenda:) http://web.mit.edu/alum/reunions/techday.html.

[39] PDF and Publishing, May, 1997. WWW publication. (competitors dispute PDF interoperability) http://www.seyboldseminars.com/Events/ny97/ShowUpdates/PS160002.HTM.

[40] Iris Pen FAQ, Web page dated August 19, 1999. WWW publication. (IDC Study on redundant typing and paper dependency) http://www.ausmedia.com.au/irispenfaq.htm#10.

[41] ACM. Production of Digitized Copies (Policy), December 18, 1998. (ACM policy for authors' own web sites [section 5.3] and OCR'ing [section 5.4]) http://www.acm.org/pubs/copyright_policy/#Distributions.

[42] Patrick Ames. *Beyond Paper: The Official Guide to Adobe Acrobat.* ISBN/ASIN 1568300506 (publisher/year unknown).

[43] Ross Anderson. Why Cryptosystems Fail. *ACM 1st Conference - Computer and Communications Security '93*, pages 215–227, November, 1993. http://www.cl.cam.ac.uk/users/rja14/wcf.html.

[44] Wesley L. Austin. A Thoughtful and Practical Analysis of Database Protection under Copyright Law, and a Critique of Sui Generis Protection. *Journal Technology Law & Policy*, 3(1), 1997. WWW publication. (the common practice of typing in phone directories discussed at) http://journal.law.ufl.edu/~techlaw/3-1/austin.html#ENIIb.

[45] Author unknown. Definitions of Adhesion Contracts. WWW publication. http://www.waukesha.tec.wi.us/busocc/law/adhes.html.

[46] Author unknown. The Office of the Future. *Business Week*, June 30, 1975. no. 2387: 48-70. (predicted imminent paperless offices).

[47] Author unknown. (Business Section lead article). *Boston Globe*, page B1, March 9, 2000.

[48] Author withheld. Digital Millenium Copyright (DMCA) Information, 2000. WWW publication. (links discussing the 1998 U.S. anti-circumvention law) http://www.tuxers.net/dmca/.

[49] Author withheld. Online Photo Sharing is the Next Hot Internet Application, September 20, 1999. http://www.infotrends-rgi.com/press/1999092089476.html.

[50] Tom W. Bell. Fair Use Vs. Fared Use: The Impact of Automated Rights Management on Copyright's Fair Use Doctrine. *North Carolina Law Review*, 76:557, 1998. http://www.tomwbell.com/writings/FullFared.html.

[51] Sven Birkerts. *The Gutenberg Elegies: The Fate of Reading in the Electronic Age*. Ballantine Books, New York, 1994. (the bible of paper loyalists).

[52] Business Week Editorial Board. How to foil internet pirates. *Business Week*, August 14, 2000. WWW publication. (argues against the DMCA anti-circumvention clause) http://www.businessweek.com/premium/00_33/b3694187.htm.

[53] Kevin Lee Bowman. Privacy And The Internet: What Is The Electronic Communications Privacy Act (ECPA), 1996. WWW publication. http://www.people.virginia.edu/~klb6q/infopaper/ECPA.html.

[54] Anne Wells Branscomb. *Who Owns Information? From Privacy to Public Access*. HarperCollins, New York, 1994.

[55] John Seely Brown and Paul Duguid. *The Social Life of Information*. Harvard Business School Press, Boston, Massachusetts, 2000. (is reviewed at) http://www.salon.com/tech/books/2000/03/09/social_information/.

[56] Doreen Carvajal. Evolving Market for E-Titles: Racing to Convert Books to Bytes. *The New York Times*, December 9, 1999. WWW publication. http://www.nytimes.com/library/tech/99/12/biztech/articles/09book.html.

[57] Chris DiBona, Sam Ockman & Mark Stone (Edited by). *Open Sources: Voices from the Open Source Revolution*. O'Reilly & Associates, Sebastopol, California, 1999.

[58] Julie E. Cohen. Unfair Use: Call it the Digital Millennium Censorship Act. *The New Republic*, May 23, 2000. WWW publication. (argues that the DMCA violates the First Amendment) http://www.thenewrepublic.com/online/cohen052300.html.

[59] The OpenSSL Core and Development Team. OpenSSL, based on SSLeay. WWW documentation. (full-featured commercial-grade SSL toolkit) http://www.openssl.org.

[60] Dr. Barbara Simons, President of the Association for Computing Machinery. (UCITA Opposition Statements, including letters to legislators), 1999 and 2000. http://www.acm.org/usacm/copyright/.

[61] Ralf S. Engelschall. mod_SSL, 2000. WWW documentation. (better-documented derivative of the Apache SSL secure web server) http://www.modssl.org.

[62] Eytan Adar, David Karger and Lynn Andrea Stein. Haystack: Per-User Information Environments. *ACM 1999 Conference on Information and Knowledge Management*, pages 413–422, May 17, 1999. http://haystack.lcs.mit.edu/papers/.

[63] Patrick Feng. When Social Meets Technical: Ethics and the Design of Social Technologies. *Conference on Freedom and Privacy 2000*, pages 295–301, April, 2000. (discusses Toronto's Highway 407, which uses OCR) http://www.cfp2000.org/papers/feng.pdf.

[64] Fred F. Ross, Nick Zellinger (Illustrator), Judy French (Editor). *OCR With a Smile: An Operator's Guide to Optical Character Recognition.* House of Scanning, LLC, Englewood, Colorado, 1998. http://www.hosc.net.

[65] Government Accountability Project (GAP). Survival Tips for Whistleblowing. WWW publication. (based on the book: The Whistleblower's Survival Guide, Courage without Martyrdom) http://www.whistleblower.org/www/Tips.htm.

[66] Simson Garfinkel. *Database Nation: The Death of Privacy in the 21st Century.* O'Reilly & Associates, Sebastopol, California, 2000.

[67] Mike Godwin. Is Stephen King's New eBook Riding the DMCA Bullet? *LawNewsNetwork.com*, March 31, 2000. WWW publication. http://www.lawnewsnetwork.com/stories/A20129-2000Mar30.html.

[68] Dan Greenwood. esig-law: Electronic Signature Area, 1999. WWW publication. (analyzes legality issues of paper, faxes, email, web pages) http://www.civics.com/old-site/esig-law.htm.

[69] Lisa Guernsey. Scan the Headlines? No, Just the Bar Codes. *The New York Times*, May 4, 2000. WWW publication. http://www.nytimes.com/library/tech/00/05/circuits/articles/04bar.html.

[70] Katie Hafner and Matthew Lyon. *Where Wizards Stay Up Late: The Origins of the Internet*. Simon & Schuster, New York, 1996.

[71] Marci A. Hamilton. Copyright Duration Extension and the Dark Heart of Copyright. *Cardozo Arts & Entertainment Law Journal*, 14(3):655, 1996. http://www.public.asu.edu/~dkarjala/commentary/hamilton-art.html.

[72] Harold Abelson (and associates). MIT 6.805/STS085: Ethics and Law on the Electronic Frontier. MIT/Harvard Class and WWW publications. http://mit.edu/6.805.

[73] Welcome to Haystack! - Personal information retrieval. WWW publications. http://haystack.lcs.mit.edu.

[74] Caere/ScanSoft Inc. High Performance Centralized OCR (including at bottom) Simple ROI Analysis of Manual Data Entry vs. A Centralized OCR Server. WWW publication. http://www.caere.com/products/productionocr/white_paper.asp.

[75] Irene M. Kunii, Geoffrey Smith and Neil Gross. Fuji: Beyond film. *Business Week*, November 22, 1999. WWW publication. http://www.businessweek.com/1999/99_47/b3656012.htm.

[76] John Haley and Mike Glover of Viking Software Services, Inc. A Guide to Evaluating Data Entry Systems (a White Paper), 1994. WWW publication. http://www.vikingsoft.com/vdewp.htm.

[77] John Haley and Mike Glover of Viking Software Services, Inc. The Importance of (...) Precision Data Entry to Document Imaging, a White Paper, 1994. WWW publication. http://www.vikingsoft.com/wp1.htm#accuracyissues.

[78] V.H. Carr Jr. Technology Adoption and Diffusion, 1999. WWW publication. http://tlc.nlm.nih.gov/resources/publications/sourcebook/adoptiondiffusion.html.

[79] Richard N. Katz and Associates. *Dancing with the Devil: Information Technology and the New Competition in Higher Education*. Jossey-Bass, San Francisco, California, 1999.

[80] Young Wook Kim. Law and cyberspace – liability of online service providers in 1996, 1997. WWW publication. http://wings.buffalo.edu/law/Complaw/CompLawPapers/kim.html.

[81] Brad King. Tuning Up Digital Copyright Law. *WIRED*, May 16, 2000. WWW publication. http://www.wired.com/news/business/0,1367,36323,00.html.

[82] Ray Kurzweil. *The Age of Spiritual Machines : When Computers Exceed Human Intelligence*. Viking Press, New York, 1999. (Kurzweil pioneered voice recognition and OCR).

[83] Adam Laurie and Ben Laurie. Apache-SSL, 2000. (Crypto-secured web server) http://www.apache-ssl.org.

[84] Lawrence Lessig. *Code: And Other Laws of Cyberspace*. Perseus Books, New York, 1999.

[85] Lawrence Lessig. In Search of Skeptics: We need to be willing to think about the effects of regulation on the process of innovation. *The Standard*, April 17, 2000. WWW publication. (addresses failures of intellectual property law) http://www.thestandard.com/article/display/0,1151,14103,00.html.

[86] (Stanford Libraries). Copyright & Fair Use: Frequently Asked Questions. WWW publication. (When is copying allowed by fair use provisions of the law?) http://fairuse.stanford.edu/library/faq.html.

[87] (Stanford Libraries). Copyright & Fair Use: Multimedia. WWW publication. http://fairuse.stanford.edu/multimed/.

[88] Jessica Litman. The Exclusive Right to Read. *Cardozo Arts & Entertainment Law Journal*, 13(1):29, 1994. http://www.msen.com/˜litman/read.htm.

[89] Omid E. Kia (maintained by). O.C.R. Frequently Asked Questions, 1997. WWW publication. http://www.cfar.umd.edu/˜kia/ocr-faq.html.

[90] Mark Stefik, John Perry Barlow, Lawrence Lessig and Charles C. Mann. Life, Liberty, and the Pursuit of Copyright? *The Atlantic Monthly*, September, 1998. WWW publication. (serial interviews) http://www.theatlantic.com/unbound/forum/copyright/stefik1.htm.

[91] Tony McKinley. *Paper to Web: How to Make Information Instantly Accessible*. Adobe Press, Indianapolis, Indiana, 1997. (Influential PDF/OCR book that is now available online for free) http://www.paper-to-web.com/id206.htm.

[92] Melissa Weisshaus, Jay Fenlason, Thomas Bushnell, n/BSG, Amy Gorin. Introduction to Tar, and Manual, April 24, 1997. WWW documentation. (pkzip-like file packager) http://www.gnu.org/software/tar/tar.html.

[93] Michael Froomkin, Professor of Law at the University of Miami. WWW resource. (Publications of a leading thinker in Internet Law, includes his upcoming paper 'The Death of Privacy?') http://www.law.miami.edu/˜froomkin/.

[94] Stephanie Miles. Palm extends hand to Adobe document technology. *CNET News*, February 8, 2000. WWW publication. http://news.cnet.com/news/0-1006-200-1545280.html.

[95] Ryoichi Mori and Masaji Kawahara. Superdistribution: The Concept and the Architecture. *The Transactions of the IEICE; VOL.E 73, NO.7, Special Issue on Cryptography and Information Security*, July, 1990. http://www.virtualschool.edu/mon/ElectronicProperty/MoriSuperdist.html.

[96] Nicholas Negroponte. *Being Digital.* Vintage Books, New York, 1995.

[97] Theodor Holm Nelson. Transcopyright: Pre-Permission for Virtual Republishing / Dealing with the Dilemma of Digital Copyright. *Educom Review*, 32(1), January/February 1997. WWW publications. http://www.sfc.keio.ac.jp/~ted/TPUB/transcopy.html or http://www.educause.edu/pub/er/review/reviewArticles/32132.html.

[98] Nicole Koffey. Digital Cameras: High Demand, No Profits. *Forbes*, May 5, 2000. WWW publication. (cites InfoTrends Research Group study) http://www.forbes.com/tool/html/00/may/0505/mu4.htm.

[99] Teun Nijssen. Cryptoscan.org, 1998. WWW publication. (How to OCR source code) http://www.pgpi.org/pgpi/project/scanning/ and http://www.cryptoscan.org.

[100] Donald A. Norman. *The Design of Everyday Things.* Doubleday, New York, 1990. (bible of user-centered design, for reviews see) http://www.peterme.com/edgewise/top10.html.

[101] J.M. Nyce and P. Kahn (Edited by). *From Memex to Hypertext: Vannevar Bush and The Mind's Machine.* Academic Press, San Diego, California, 1991/92. ('Memex as an Image of Potentiality Revisited' by Linda C Smith discusses how Memex is misunderstood and misappropriated, see also 'As We Will Think' by Theodor Nelson).

[102] oedipal enterprises, Gregory J. Rosmaita. Blindness-Related Resources on the Web and Beyond. WWW publication. (blind users often depend on the lynx browser for text and html screenreading) http://www.hicom.net/~oedipus/blind.html.

[103] Cem Kaner (Law Office of). Bad Software: What To Do When Software Fails. WWW publication. http://www.badsoftware.com.

[104] Winston Tabb (Associate Librarian of Congress). ALA Briefing, January 12, 2000. WWW publication. (discusses their increasing use of OCR) http://lcweb.loc.gov/library/alamw00.html.

[105] IEEE-USA Board of Directors. Opposing Adoption of the Uniform Computer Information Transactions Act (UCITA) By the States, February, 2000. WWW publication. http://www.ieeeusa.org/forum/POSITIONS/ucita.html.

[106] Walter J. Ong. *Orality & Literacy: The Technologizing of the Word.* Methuen/Routledge, London, 1982. (argues that humanity's very way of thinking changes as media technologies change).

[107] Steve Outing. The Business Case for Digitizing Oldest Archives. *Editor & Publisher*, October 27, 1999. WWW publication. http://www.editorandpublisher.com/ephome/news/newshtm/stop/st102799.htm.

[108] Ragica. PDF: introductory/historical notes (all you need to crack it), 1997. WWW publication. http://www.instinct.org/fravia/ragica1.htm.

[109] Ronald Rivest and Adi Shamir. PayWord and MicroMint: Two Simple Micropayment Schemes. *CryptoBytes, (RSA Laboratories, Spring 1996), 7–11*, 2(1), May 7, 1996. (also in Proceedings of 1996 International Workshop on Security Protocols) http://theory.lcs.mit.edu/˜rivest/RivestShamir-mpay.ps.

[110] M.J. Rose. IPublish Praised, But Will URead? *WIRED*, May 24, 2000. WWW publication. http://www.wired.com/news/business/0,1367,36548,00.html.

[111] Jerome H. Saltzer. MIT/LCS Library 2000. (personal communication with director of project, for related info see http://ltt-www.lcs.mit.edu/ltt-www/).

[112] Pamela Samuelson. Intellectual Property And The Digital Economy: Why The Anti-Circumvention Regulations Need To Be Revised. *Berkeley Technology Law Journal*, 14(2):519, 1999. http://www.sims.berkeley.edu/˜pam/papers.html or http://www.law.berkeley.edu/journals/btlj/articles/14_2/Samuelson/html/reader.html.

[113] Pamela Samuelson. Privacy as Intellectual Property? *Stanford Law Review*, draft, forthcoming 2000. http://www.sims.berkeley.edu/~pam/papers.html.

[114] Glenn Sanders and Wade Roush. Cracking the Bullet: Hackers Decrypt PDF Version of Stephen King eBook, an eBookNet Special Report. March 23, 2000. WWW publication. http://www.ebooknet.com/printerVersion.jsp?id=1671.

[115] Andrew L. Shapiro. *How the Internet is Putting Individuals in Charge and Changing the World We Know*. Perseus Books, New York, 1999.

[116] Carl Shapiro and Hal R. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston, Massachusetts, November, 1998.

[117] Richard M. Smith. Advanced web programming. (Internet Security and Privacy Expert) http://www.tiac.net/users/smiths/.

[118] Richard Stallman. The Right to Read. *Communications of the ACM*, 40(2), 1997. (provocative and entertaining allegory) http://www.gnu.org/philosophy/right-to-read.html.

[119] Mark Stefik. Trusted Systems. *Scientific American*, 1997. WWW publication. http://www.sciam.com/0397issue/0397stefik.html.

[120] Dan Verton. CIA tackles records nightmare. April 20, 2000. WWW publication. http://www.cnn.com/2000/TECH/computing/04/20/cia.nightmare.idg/.

[121] Stephen Wildstrom. Mac Hits Another Home Run. *Business Week*, February 28, 2000. WWW publication. (MacOS X promises to deliver PDF later in 2000) http://www.businessweek.com/2000/00_09/c3670091.htm.

[122] G. Pascal Zachary. *Endless Frontier: Vannevar Bush, Engineer of the American Century*. Simon & Schuster, New York, 1997. (Memex cited throughout).

[123] (SNARF's developer) Zachary Beane. Downloader Comparison Table. WWW publication. http://www.xach.com/snarf/comparison-table.php3.