CHAPTER 3

Ghosts in the Machine

Secrets and Surprises of Electronic Documents

What You See Is Not What the Computer Knows

On March 4, 2005, Italian journalist Giuliana Sgrena was released from captivity in Baghdad, where she had been held hostage for a month. As the car conveying her to safety approached a checkpoint, it was struck with gunfire from American soldiers. The shots wounded Sgrena and her driver and killed an Italian intelligence agent, Nicola Calipari, who had helped engineer her release.

A fierce dispute ensued about why U.S soldiers had rained gunfire on a car carrying citizens of one of its Iraq war allies. The Americans claimed that the car was speeding and did not slow when warned. The Italians denied both claims. The issue caused diplomatic tension between the U.S. and Italy and was a significant political problem for the Italian prime minister.

The U.S. produced a 42-page report on the incident, exonerating the U.S. soldiers. The report enraged Italian officials. The Italians quickly released their own report, which differed from the U.S. report in crucial details.

Because the U.S. report included sensitive military information, it was heavily redacted before being shared outside military circles (see Figure 3.1). In another time, passages would have been blacked out with a felt marker, and the document would have been photocopied and given to reporters. But in the information age, the document was redacted and distributed electronically, not physically. The redacted report was posted on a web site the allies used to provide war information to the media. In an instant, it was visible to any of the world's hundreds of millions of Internet users.

(U) [redacted] has Direct Liaison Authorized (DIRLAUTH) to coordinate directly with [redacted] for security along Route Irish. This is the same level of coordination previously authorized by [redacted] Division to [redacted]. When executing DIRLAUTH, [redacted] directly coordinates an action with units internal or external to its command and keeps the [redacted] commander informed. The [redacted] TOC passes all coordination efforts through the [redacted] Brigade TOC to [redacted] JOC. (Annex 58C).

Source: http://www.corriere.it/Media/Documenti/Classified.pdf, extract from page 10.

FIGURE 3.1 Section from page 10 of redacted U.S. report on the death of Italian journalist Nicola Calipari. Information that might have been useful to the enemy was blacked out.

One of those Internet users was an Italian blogger, who scrutinized the U.S. report and quickly recovered the redacted text using ordinary office software. The blogger posted the full text of the report (see Figure 3.2) on his own web site. The unredacted text disclosed positions of troops and equipment, rules of engagement, procedures followed by allied troops, and other information of interest to the enemy. The revelations were both dangerous to U.S. soldiers and acutely embarrassing to the U.S. government, at a moment when tempers were high among Italian and U.S. officials. In the middle of the most high-tech war in history, how could this fiasco have happened?

(U) 1-76 FA has Direct Liaison Authorized (DIRLAUTH) to coordinate directly with 1-69 IN for security along Route Irish. This is the same level of coordination previously authorized by 1st Cavalry Division to 2-82 FA. When executing DIRLAUTH, 1-76 FA directly coordinates an action with units internal or external to its command and keeps the 31D commander informed. The 1-76 FA TOC passes all coordination efforts through the 4th Brigade TOC to 31D JOC. (Annex 58C).

Source: http://www.corriere.it/Media/Documenti/Unclassified.doc.

FIGURE 3.2 The text of Figure 3.1 with the redaction bars electronically removed.

Paper documents and electronic documents are useful in many of the same ways. Both can be inspected, copied, and stored. But they are not equally useful for all purposes. Electronic documents are easier to change, but paper documents are easier to read in the bathtub. In fact, the metaphor of a series of bits as a "document" can be taken only so far. When stretched beyond its breaking point, the "document" metaphor can produce surprising and damaging results—as happened with the Calipari report.

Office workers love "WYSIWYG" interfaces—"What You See Is What You Get." They edit the electronic document on the screen, and when they print it, it looks just the same. They are deceived into thinking that what is in the

computer is a sort of miniaturized duplicate of the image on the screen, instead of computer codes that produce the picture on the screen. In fact, the WYSIWYG metaphor is imperfect, and therefore risky. The report on the death of Nicola Calipari illustrates what can go wrong when users accept such a metaphor too literally. What the authors of the document saw was dramatically different from what they got.

The report had been prepared using software that creates PDF files. Such software often includes a "Highlighter Tool," meant to mimic the felt markers that leave a pale mark on ordinary paper, through which the underlying text is visible (see Figure 3.3). The software interface shows the tool's icon as a marker writing a yellow stripe, but the user can change the color of the stripe. Probably someone tried to turn the Highlighter Tool into a redaction tool by changing its color to black, unaware that what was visible on the screen was not the same as the contents of the electronic document.

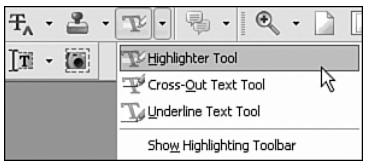
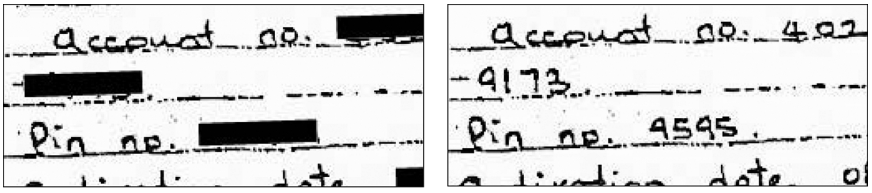


FIGURE 3.3 Adobe Acrobat Highlighter Tool, just above the middle. On the screen, the "highlighter" is writing yellow ink, but with a menu command, it can be changed to any other color.

The Italian blogger guessed that the black bars were nothing more than overlays created using the Highlighter Tool, and that the ghostly traces of the invisible words were still part of the electronic document that was posted on the web. With that realization, he easily undid the black "highlighting" to reveal the text beneath.

Just as disturbing as this mistake is the fact that two major newspapers had quite publicly made the same mistake only a few years before. On April 16, 2000, the *New York Times* had detailed a secret CIA history of attempts by the U.S. to overthrow Iran's government in 1953. The newspaper reproduced sections of the CIA report, with black redaction bars to obscure the names of CIA operatives within Iran. The article was posted on the Web in mid-June, 2000, accompanied by PDFs of several pages of the CIA report. John Young, who administers a web site devoted to publishing government-restricted documents, removed the redaction bars and revealed the names of CIA agents. A controversy ensued about the ethics and legality of the disclosure, but the names are still available on the Web as of this writing.

The *Washington Post* made exactly the same mistake in 2002, when it published an article about a demand letter left by the Washington snipers, John Allen Muhammad and John Lee Malvo. As posted on the *Post*'s web site, certain information was redacted in a way that was easily reversed by an inquisitive reader of the online edition of the paper (see Figure 3.4). The paper fixed the problem quickly after its discovery, but not quickly enough to prevent copies from being saved.



Source: Washington Post web site, transferred to web.bham.ac.uk/forensic/news/02/sniper2.html. Actual images taken from slide 29 of http://www.ccc.de/congress/2004/fahrplan/files/316-hidden-data-slides.pdf.

FIGURE 3.4 Letter from the Washington snipers. On the left, the redacted letter as posted on the *Washington Post* web site. On the right, the letter with the redaction bars electronically removed.

What might have been done in these cases, instead of posting the PDF with the redacted text hidden but discoverable? The Adobe Acrobat software has a security feature, which uses encryption (discussed in Chapter 5, "Secret Bits") to make it impossible for documents to be altered by unauthorized persons, while still enabling anyone to view them. Probably those who created these documents did not know about this feature, or about commercially available software called Redax, which government agencies use to redact text from documents created by Adobe Acrobat.

A clumsier, but effective, option would be to scan the printed page, complete with its redaction bars. The resulting file would record only a series of black and white dots, losing all the underlying typographical structure—font names and margins, for example. Whatever letters had once been "hidden" under the redaction bars could certainly not be recovered, yet this solution has an important disadvantage.

One of the merits of formatted text documents such as PDFs is that they can be "read" by a computer. They can be searched, and the text they contain can be copied. With the document reduced to a mass of black and white dots, it could no longer be manipulated as text.

A more important capability would be lost as well. The report would be unusable by programs that vocalize documents for visually impaired readers. A blind reader could "read" the U.S. report on the Calipari incident, because software is available that "speaks" the contents of PDF documents. A blind reader would find a scanned version of the same document useless.

Tracking Changes—and Forgetting That They Are Remembered

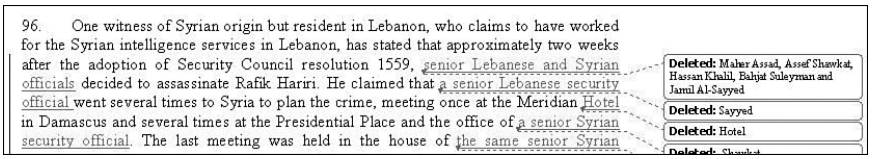
In October, 2005, UN prosecutor Detlev Mehlis released to the media a report on the assassination of former Lebanese Prime Minister Rafik Hariri. Syria had been suspected of engineering the killing, but Syrian President Bashar al-Assad denied any involvement. The report was not final, Mehlis said, but there was "evidence of both Lebanese and Syrian involvement." Deleted, and yet uncovered by the reporters who were given the document, was an incendiary claim: that Assad's brother Maher, commander of the Republican Guard, was personally involved in the assassination.

Microsoft Word offers a "Track Changes" option. If enabled, every change made to the document is logged as part of the document itself—but ordinarily not shown. The document bears its entire creation history: who made each change, when, and what it was. Those editing the document can also add comments—which would not appear in the final document, but may help editors explain their thinking to their colleagues as the document moves around electronically within an office.

Of course, information about strategic planning is not meant for outsiders to see, and in the case of legal documents, can have catastrophic consequences if revealed. It is a simple matter to remove these notes about the document's history—but someone has to remember to do it! The UN prosecutor neglected to remove the change history from his Microsoft Word document, and a reporter discovered the deleted text (see Figure 3.5). (Of course, in Middle Eastern affairs, one cannot be too suspicious. Some thought that Mehlis had intentionally left the text in the document, as a warning to the Syrians that he knew more than he was yet prepared to acknowledge.)

A particularly negligent example of document editing involved SCO Corporation, which claimed that several corporations violated its intellectual property rights. In early 2004, SCO filed suit in a Michigan court against Daimler Chrysler, claiming Daimler had violated terms of its Unix software agreement with SCO. But the electronic version of its complaint carried its modification history with it, revealing a great deal of information about SCO's litigation planning. In particular, when the change history was revealed, it

turned out that until exactly 11:10 a.m. on February 18, 2004, SCO had instead planned to sue a different company, Bank of America, in federal rather than state court, for copyright infringement rather than breach of contract!


96. One witness of Syrian origin but resident in Lebanon, who claims to have worked for the Syrian intelligence services in Lebanon, has stated that approximately two weeks after the adoption of Security Council resolution 1559, senior Lebanese and Syrian officials decided to assassinate Rafik Hariri. He claimed that a senior Lebanese security official went several times to Syria to plan the crime, meeting once at the Meridian Hotel in Damascus and several times at the Presidential Place and the office of a senior Syrian security official. The last meeting was held in the house of the same senior Syrian

Deleted: Maher Assad, Assef Shawkat, Hassan Khalil, Bahjat Suleyman and Jamil Al-Sayyed

Deleted: Sayyed

Deleted: Hotel

Source: Section of UN report, posted on Washington Post web site, `www.washingtonpost.com/wp-srv/world/syria/mehlis.report.doc`.

FIGURE 3.5 Section from the UN report on the assassination of Rafik Hariri. An earlier draft stated that Maher Assad and others were suspected of involvement in the killing, but in the document as it was released, their names were replaced with the phrase "senior Lebanese and Syrian officials."

Saved Information About a Document

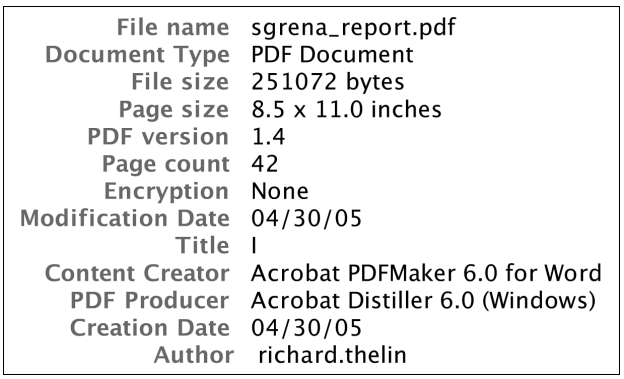
FORGING METADATA

Metadata can help prove or refute claims. Suppose Sam emails his teacher a homework paper after the due date, with a plea that the work had been completed by the deadline, but was undeliverable due to a network failure. If Sam is a cheater, he could be exposed if he doesn't realize that the "last modified" date is part of the document. However, if Sam *is* aware of this, he could "stamp" the document with the right time by re-setting the computer's clock before saving the file. The name in which the computer is registered and other metadata are also forgeable, and therefore are of limited use as evidence in court cases.

An electronic document (for example, one produced by text-processing software) often includes information that is *about* the document—so-called *metadata*. The most obvious example is the name of the file itself. File names carry few risks. For example, when we send someone a file as an email attachment, we realize that the recipient is going to see the name of the file as well as its contents.

But the file is often tagged with much more information than just its name. The metadata generally includes the name associated with the owner of the computer, and the dates the file was created and last modified—often useful information, since the recipient can tell whether she is receiving an older or newer version than the version she already

has. Some word processors include version information as well, a record of who changed what, when, and why. But the unaware can be trapped even by such innocent information, since it tends not to be visible unless the recipient asks to see it. In Figure 3.6, the metadata reveals the name of the military officer who created the redacted report on the death of Nicola Calipari.



File name	sgrena_report.pdf
Document Type	PDF Document
File size	251072 bytes
Page size	8.5 x 11.0 inches
PDF version	1.4
Page count	42
Encryption	None
Modification Date	04/30/05
Title	I
Content Creator	Acrobat PDFMaker 6.0 for Word
PDF Producer	Acrobat Distiller 6.0 (Windows)
Creation Date	04/30/05
Author	richard.thelin

FIGURE 3.6 Part of the metadata of the Calipari report, as revealed by the "Properties" command of Adobe Acrobat Reader. The data shows that Richard Thelin was the author, and that he altered the file less than two minutes after creating it. Thelin was a Lieutenant Colonel in the U.S. Marine Corps at the time of the incident.

Authorship information leaked in this way can have real consequences. In 2003, the British government of Tony Blair released documentation of its case for joining the U.S. war effort in Iraq. The document had many problems—large parts of it turned out to have been plagiarized from a 13-year-old PhD thesis. Equally embarrassing was that the electronic fingerprints of four civil servants who created it were left on the document when it was released electronically on the No. 10 Downing Street web site. According to the *Evening Standard of London*, "All worked in propaganda units controlled by Alastair Campbell, Tony Blair's director of strategy and communications," although the report had supposedly been the work of the Foreign Office. The case of the "dodgy dossier" caused an uproar in Parliament.

You don't have to be a businessperson or government official to be victimized by documents bearing fingerprints. When you send someone a document as an attachment to an email, very likely the document's metadata shows who actually created it, and when. If you received it from someone else

and then altered it, that may show as well. If you put the text of the document into the body of your email instead, the metadata won't be included; the message will be just the text you see on the screen. Be sure of what you are sending before you send it!

Can the Leaks Be Stopped?

Even in the most professional organizations, and certainly in ordinary households, knowledge about technological dangers and risks does not spread instantaneously to everyone who should know it. The Calipari report was published five years after the *New York Times* had been embarrassed. How can users of modern information technology—today, almost all literate people—stay abreast of knowledge about when and how to protect their information?

It is not easy to prevent the leakage of sensitive information that is hidden in documents but forgotten by their creators, or that is captured as metadata. In principle, offices should have a check-out protocol so that documents are cleansed before release. But in a networked world, where email is a critical utility, how can offices enforce document release protocols without rendering simple tasks cumbersome? A rather harsh measure is to prohibit use of software that retains such information; that was the solution adopted by the British government in the aftermath of the "dodgy dossier" scandal. But the useful features of the software are then lost at the same time. A protocol can be established for converting "rich" document formats such as that of Microsoft Word to formats that retain less information, such as Adobe PDF. But it turns out that measures used to eradicate personally identifiable information from documents don't achieve as thorough a cleansing as is commonly assumed.

At a minimum, office workers need education. Their software has great capabilities they may find useful, but many of those useful features have risks as well. And we all just need to think about what we are doing with our documents. We all too mindlessly re-type keystrokes we have typed a hundred times in the past, not pausing to think that the hundred and first situation may be different in some critical way!

Representation, Reality, and Illusion

René Magritte, in his famous painting of a pipe, said "This isn't a pipe" (see Figure 3.7). Of course it isn't; it's a painting of a pipe. The image is made out

of paint, and Magritte was making a metaphysical joke. The painting is entitled "The treachery of images," and the statement that the image isn't the reality is part of the image itself.

Los Angeles County Museum of Art. Purchased with funds provided by the Mr. and Mrs. William Preston Harrison Collection. Photograph © 2007 Museum Associates/LACMA.

FIGURE 3.7 Painting by Magritte. The legend says "This isn't a pipe." Indeed, it's only smudges of paint that make you think of a pipe, just as an electronic document is only bits representing a document.

When you take a photograph, you capture inside the camera something from which an image can be produced. In a digital camera, the bits in an electronic memory are altered according to some pattern. The image, we say, is "represented" in the camera's memory. But if you took out the memory and looked at it, you couldn't see the image. Even if you printed the pattern of 0s and 1s stored in the memory, the image wouldn't appear. You'd have to know *how* the bits represent the image in order to get at the image itself. In the world of digital photography, the format of the bits has been standardized, so that photographs taken on a variety of cameras can be displayed on a variety of computers and printed on a variety of printers.

The general process of digital photography is shown in Figure 3.8. Some external reality—a scene viewed through a camera lens, for example—is turned into a string of bits. The bits somehow capture useful information

about reality, but there is nothing "natural" about the way reality is captured. The representation is a sort of ghost of the original, not identical to the original and actually quite unlike it, but containing enough of the soul of the original to be useful later on. The representation follows rules. The rules are arbitrary conventions and the product of human invention, but they have been widely accepted so photographs can be exchanged.

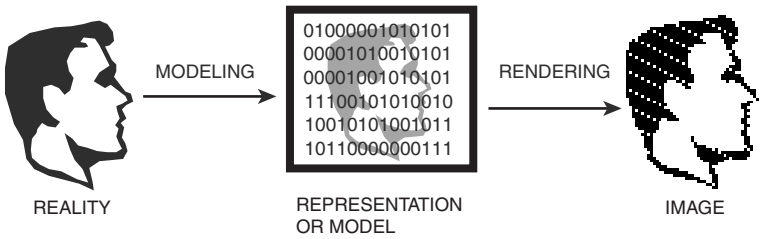


FIGURE 3.8 Reproducing an image electronically is a two-stage process. First, the scene is translated into bits, creating a digital model. Then the model is rendered as a visible image. The model can be stored indefinitely, communicated from one place to another, or computationally analyzed and enhanced to produce a different model before it is rendered. The same basic structure applies to the reproduction of video and audio.

The representation of the photograph in bits is called a *model* and the process of capturing it is called *modeling*. The model is turned into an image by *rendering* the model; this is what happens when you transfer the bits representing a digital photograph to a computer screen or printer. Rendering brings the ghost back to life. The image resembles, to the human eye, the original reality—provided that the model is good enough. Typically, a model that is not good enough—has too few bits, for example—cannot produce an image that convincingly resembles the reality it was meant to capture.

Modeling always omits information. Magritte's painting doesn't smell like a pipe; it has a different patina than a pipe; and you can't turn it around to see what the other side of the pipe looks like. Whether the omitted information is irrelevant or essential can't be judged without knowing how the model is going to be used. Whoever creates the model and renders it has the power to shape the experience of the viewer.

The process of modeling followed by rendering applies to many situations other than digital photography. For example, the same transformations happen when music is captured on a CD or as an MP3. The rendering process produces audible music from a digital representation, via stereo speakers or a

headset. CDs and MP3s use quite distinct modeling methods, with CDs generally capturing music more accurately, using a larger number of bits.

Knowing that digital representations don't resemble the things they represent explains the difference between the terms "analog" and "digital." An analog telephone uses a continuously varying electric signal to represent a continuously varying sound—the voltage of the telephone signal is an "analog" of the sound it resembles—in the same way that Magritte applied paint smoothly to canvas to mimic the shape of the pipe. The shift from analog to digital technologies, in telephones, televisions, cameras, X-ray machines, and many other devices, at first seems to lose the immediacy and simplicity of the old devices. But the enormous processing power of modern computers makes the digital representation far more flexible and useful.

Indeed, the same general processes are at work in situations where *there is no "reality" because the images are of things that have never existed.* Examples are video games, animated films, and virtual walk-throughs of unbuilt architecture. In these cases, the first step of Figure 3.8 is truncated. The "model" is created not by capturing reality in an approximate way, but by pure synthesis: as the strokes of an artist's electronic pen, or the output of computer-aided design software.

The severing of the immediate connection between representation and reality in the digital world has created opportunities, dangers, and puzzles. One of the earliest triumphs of "digital signal processing," the science of doing computations on the digital representations of reality, was to remove the scratches and noise from old recordings of the great singer Enrico Caruso. No amount of analog electronics could have cleaned up the old records and restored the clarity to Caruso's voice.

And yet the growth of digital "editing" has its dark side as well. Photo-editing software such as Photoshop can be used to alter photographic evidence presented to courts of law.

Can We Be Sure a Photo Is Unretouched?

Cryptographic methods (discussed in Chapter 5) can establish that a digital photograph has not been altered. A special camera gets a digital key from the "image verification system," attaches a "digital signature" (see Chapter 5) to the image and uploads the image and the signature to the verification system. The system processes the received image with the same key and verifies that the same signature results. The system is secure because it is impossible, with any reasonable amount of computation, to produce another image that would yield the same signature with this key.

The movie *Toy Story* and its descendants are unlikely to put human actors out of work in the near future, but how should society think about synthetic child pornography? "Kiddie porn" is absolutely illegal, unlike other forms of pornography, because of the harm done to the children who are abused to produce it. But what about pornographic images of children who do not exist and never have—who are simply the creation of a skilled graphic synthesizer? Congress outlawed such virtual kiddie porn in 1996, in a law that prohibited any image that "is, or appears to be, of a minor engaging in sexually explicit conduct." The Supreme Court overturned the law on First Amendment grounds. Prohibiting images that "appear to" depict children is going too far, the court ruled—such synthetic pictures, no matter how abhorrent, are constitutionally protected free speech.

In the world of exploded assumptions about reality and artifice, laws that combat society's problems may also compromise rights of free expression.

In this instance at least, reality matters, not what images appear to show. Chapter 7, "You Can't Say That on the Internet," discusses other cases in which society is struggling to control social evils that are facilitated by information technology. In the world of exploded assumptions about reality and artifice, laws that combat society's problems may also compromise rights of free expression.

What Is the Right Representation?

DIGITAL CAMERAS AND MEGAPIXELS

Megapixels—millions of pixels—are a standard figure of merit for digital cameras. If a camera captures too few pixels, it can't take good photographs. But no one should think that more pixels invariably yield a better image. If a digital camera has a low-quality lens, more pixels will simply produce a more precise representation of a blurry picture!

Figure 3.9 is a page from the Book of Kells, one of the masterpieces of medieval manuscript illumination, produced around A.D. 800 in an Irish monastery. The page contains a few words of Latin, portrayed in an astoundingly complex interwoven lacework of human and animal figures, whorls, and crosshatching. The book is hundreds of pages long, and in the entire work no two of the letters or decorative ornaments are drawn the same way. The elaborately ornate graphic shows just 21 letters (see Figure 3.10).

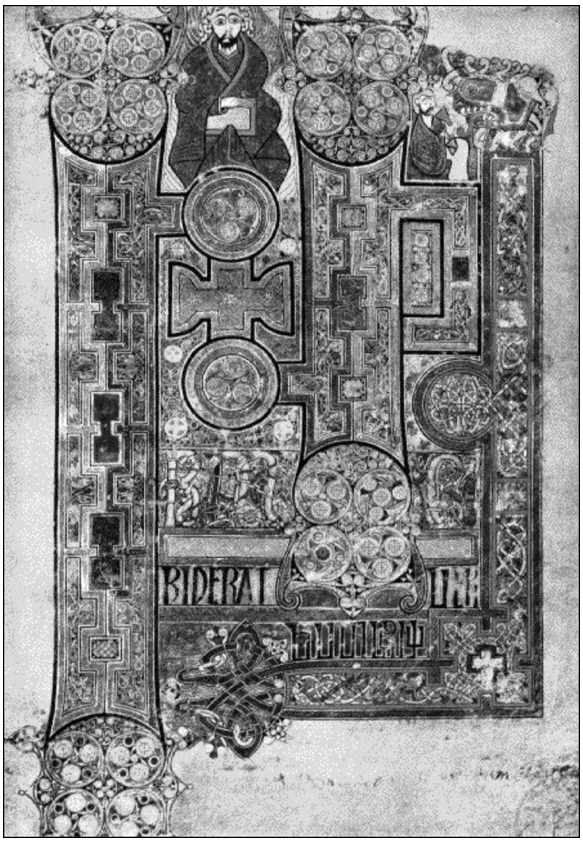


FIGURE 3.9 Opening page of the Gospel of St. John from the Book of Kells.

IN PRINCIPIO ERAT VERBUM

FIGURE 3.10 The words of the beginning of the gospel of St. John. In the book of Kells, the easiest word to spot is ERAT, just to the left of center about a quarter of the way up the page.

Do these two illustrations contain the same information? The answer depends on what information is meant to be recorded. If the only important thing were the Latin prose, then either representation might be equally good, though Figure 3.10 is easier to read. But the words themselves are far from

the only important thing in the Book of Kells. It is one of the great works of Western art and craftsmanship.

A graphic image such as Figure 3.9 is represented as a rectangular grid of many rows and columns, by recording the color at each position in the grid (see Figure 3.11). To produce such a representation, the page itself is scanned, one narrow row after the next, and each row is divided horizontally into tiny square "picture elements" or *pixels*. An image representation based on a division into pixels is called a *raster* or *bitmap representation*. The representation corresponds to the structure of a computer screen (or a digital TV screen), which is also divided into a grid of individual pixels—how many pixels, and how small they are, affect the quality and price of the display.

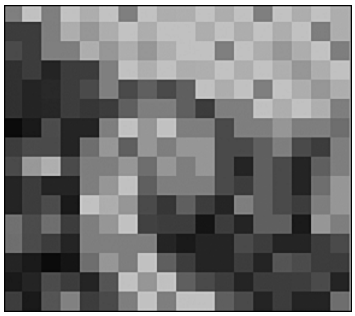


FIGURE 3.11 A detail enlarged from the upper-right corner of the opening page of John from the Book of Kells.

What would be the computer representation of the mere Latin text, Figure 3.10? The standard code for the Roman alphabet, called ASCII for the American Standard Code for Information Interchange, assigns a different 8-bit code to each letter or symbol. ASCII uses one byte (8 bits) per character. For example, A = `01000001`, a = `01100001`, $ = `00100100`, and 7 = `00110111`.

The equation 7 = `00110111` means that the bit pattern used to represent the symbol "7" in a string of text is `00110111`. The space character has its own code, `00100000`. Figure 3.12 shows the ASCII representation of the characters "IN PRINCIPIO ERAT VERBUM," a string of 24 bytes or 192 bits. We've separated the long string of bits into bytes to improve readability ever so slightly! But inside the computer, it would just be one bit after the next.

```
01001001 01001110 00100000 01010000
01010010 01001001 01001110 01000011
01001001 01010000 01001001 01001111
00100000 01000101 01010010 01000001
01010100 00100000 01010110 01000101
01010010 01000010 01010101 01001101
```

FIGURE 3.12 ASCII bit string for the characters of "IN PRINCIPIO ERAT VERBUM."

So `01001001` represents the letter I. But not always! Bit strings are used to represent many things other than characters. For example, the same bit string `01001001`, if interpreted as the representation of a whole number in binary notation, represents 73. A computer cannot simply look at a bit string `01001001` and know whether it is supposed to represent the letter I or the number 73 or data of some other type, a color perhaps. A computer can interpret a bit string only if it knows the conventions that were used to create the document—the intended interpretation of the bits that make up the file.

The meaning of a bit string is a matter of convention. Such conventions are arbitrary at first. The code for the letter I could have been `11000101` or pretty much anything else. Once conventions have become accepted through a social process of agreement and economic incentive, they became nearly as inflexible as if they were physical laws. Today, millions of computers assume

FILENAME EXTENSIONS

The three letters after the dot at the end of a filename indicate how the contents are to be interpreted. Some examples are as follows:

Extension	File Type
.doc	Microsoft Word document
.odt	OpenDocument text document
.ppt	Microsoft PowerPoint document
.ods	OpenDocument Spreadsheet
.pdf	Adobe Portable Document Format
.exe	Executable program
.gif	Graphics Interchange Format (uses 256-color palette)
.jpg	JPEG graphic file (Joint Photographic Experts Group)
.mpg	MPEG movie file (Moving Picture Experts Group)

that 01001001, if interpreted as a character, represents the letter I, and the universal acceptance of such conventions is what makes worldwide information flows possible.

The document format is the key to turning the representation into a viewable document. If a program misinterprets a document as being in a different format from the one in which it was created, only nonsense will be rendered. Computers not equipped with software matching the program that created a document generally refuse to open it.

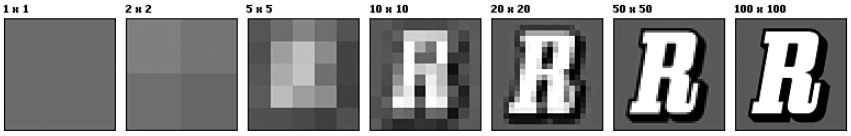
Which representation is "better," a raster image or ASCII? The answer depends on the use to which the document is to be put. For representation of freeform shapes in a great variety of shades and hues, a raster representation is unbeatable, provided the pixels are small enough and there are enough of them. But it is hard even for a trained human to find the individual letters within Figure 3.9, and it would be virtually impossible for a computer program. On the other hand, a document format based on ASCII codes for characters, such as the PDF format, can easily be searched for text strings.

The PDF format includes more than simply the ASCII codes for the text. PDF files include information about typefaces, the colors of the text and of the background, and the size and exact positions of the letters. Software that produces PDFs is used to typeset elegant documents such as this one. In other words, PDF is actually a *page description language* and describes visible features that are typographically meaningful. But for complicated pictures, a graphical format such as JPG must be used. A mixed document, such as these pages, includes graphics within PDF files.

Reducing Data, Sometimes Without Losing Information

Let's take another look at the page from the Book of Kells, Figure 3.9, and the enlargement of a small detail of that image, Figure 3.11. The computer file from which Figure 3.9 was printed is 463 pixels wide and 651 pixels tall, for a total of about 300,000 individual pixels. The pages of the Book of Kells measure about 10 by 13 inches, so the raster image has only about 50 pixels per inch of the original work. That is too few to capture the rich detail of the original—Figure 3.11 actually shows one of the animal heads in the top-right corner of the page. A great deal of detail was lost when the original page was scanned and turned into pixels. The technical term for the problem is *undersampling*. The scanning device "samples" the color value of the original document at discrete points to create the representation of the document, and in this case, the samples are too far apart to preserve detail that is visible to the naked eye in the original.

The answer to undersampling is to increase the resolution of the scan—the number of samples per inch. Figure 3.13 shows how the quality of an image improves with the resolution. In each image, each pixel is colored with the "average" color of part of the original.



Credit as in Wikipedia, en.wikipedia.org/wiki/Image:Resolution_illustration.png.

FIGURE 3.13 A shape shown at various resolutions, from 1×1 to 100×100 pixels. A square block consisting of many pixels of a single shade can be represented much more compactly than by repeating the code for that shade as many times as there are pixels.

But, of course, a price is paid for increased resolution. The more pixels in the representation of an image, the more memory is needed to hold the representation. Double the resolution, and the memory needed goes up by a factor of four, since the resolution doubles both vertically and horizontally.

Standard software uses a variety of representational techniques to represent raster graphics more concisely. Compression techniques are of two kinds: "lossless" and "lossy." A *lossless* representation is one that allows exactly the same image to be rendered. A *lossy* representation allows an approximation to the same image to be rendered—an image that is different from the original in ways the human eye may or may not be able to discern.

One method used for lossless image compression takes advantage of the fact that in most images, the color doesn't change from pixel to pixel—the image has *spatial coherence*, to use the official term. Looking at the middle and rightmost images in Figure 3.13, for example, makes clear that in the 100×100 resolution image, the 100 pixels in a

AUDIO COMPRESSION

MP3 is a lossy compression method for audio. It uses a variety of tricks to create small data files. For example, human ears are not far enough apart to hear low-frequency sounds stereophonically, so MP3s may record low frequencies in mono and play the same sound to both speakers, while recording and playing the higher frequencies in stereo! MP3s are "good enough" for many purposes, but a trained and sensitive ear can detect the loss of sound quality.

10 × 10 square in the top-left corner are all the same color; there is no need to repeat a 24-bit color value 100 times in the representation of the image.

Accordingly, graphic representations have ways of saying "all pixels in this block have the same color value." Doing so can reduce the number of bits significantly.

Depending on how an image will be used, a lossy compression method might be acceptable. What flashes on your TV is gone before you have time to scrutinize the individual pixels. But in some cases, only lossless compression is satisfactory. If you have the famous Zapruder film of the Kennedy assassination and want to preserve it in a digital archive, you want to use a lossless compression method once you have digitized it at a suitably fine resolution. But if you are just shipping off the image to a low-quality printer such as those used to print newspapers, lossy compression might be fine.

Technological Birth and Death

The digital revolution was possible because the capacity of memory chips increased, relentlessly following Moore's Law. Eventually, it became possible to store digitized images and sounds at such high resolution that their quality was higher than analog representations. Moreover, the price became low enough that the storage chips could be included in consumer goods. But more than electrical engineering is involved. At more than a megabyte per image, digital cameras and HD televisions would still be exotic rarities. A *megabyte* is about a million bytes, and that is just too much data per image. The revolution also required better algorithms—better computational methods, not just better hardware—and fast, cheap processing chips to carry out those algorithms.

For example, digital video compression utilizes *temporal coherence* as well as spatial coherence. Any portion of the image is unlikely to change much in color from frame to frame, so large parts of a picture typically do not have to be retransmitted to the home when the frame changes after a thirtieth of a second. At least, that is true in principle. If a woman in a TV image walks across a fixed landscape, only her image, and a bit of landscape that newly appears from behind her once she passes it, needs be transmitted—*if* it is computationally feasible to compare the second frame to the first before it is transmitted and determine exactly where it differs from its predecessor. To keep up with the video speed, there is only a thirtieth of a second to do that computation. And a complementary computation has to be carried out at the other end—the previously transmitted frame must be modified to reflect the newly transmitted information about what part of it should change one frame time later.

Digital movies could not have happened without an extraordinary increase in speed and drop in price in computing power. Decompression algorithms are built into desktop photo printers and cable TV boxes, cast in silicon in chips more powerful than the fastest computers of only a few years ago. Such compact representations can be sent quickly through cables and as satellite signals. The computing power in the cable boxes and television sets is today powerful enough to reconstruct the image from the representation of what has changed. Processing is power.

By contrast, part of the reason the compact disk is dying as a medium for distributing music is that it doesn't hold enough data. At the time the CD format was adopted as a standard, decompression circuitry for CD players would have been too costly for use in homes and automobiles, so music could not be recorded in compressed form. The magic of Apple's iPod is not just the huge capacity and tiny physical size of its disk—it is the power of the processing chip that renders the stored model as music.

The birth of new technologies presage the death of old technologies. Digital cameras killed the silver halide film industry; analog television sets will soon be gone; phonograph records gave way to cassette tapes, which in turn gave way to compact disks, which are themselves now dying in favor of digital music players with their highly compressed data formats.

The periods of transition between technologies, when one emerges and threatens another that is already in wide use, are often marked by the exercise of power, not always progressively. Businesses that dominate old technologies are sometimes innovators, but often their past successes make them slow to change. At their worst, they may throw up roadblocks to progress in an attempt to hold their ground in the marketplace. Those roadblocks may include efforts to scare the public about potential disruptions to familiar practices, or about the dollar costs of progress.

Data formats, the mere conventions used to intercommunicate information, can be remarkably contentious, when a change threatens the business of an incumbent party, as the Commonwealth of Massachusetts learned when it tried to change its document formats. The tale of Massachusetts and OpenDocument illustrates how hard change can be in the digital world, although it sometimes seems to change on an almost daily basis.

Data Formats as Public Property

No one owns the Internet, and everyone owns the Internet. No government controls the whole system, and in the U.S., the federal government controls only the computers of government agencies. If you download a web page to

your home computer, it will reach you through the cooperation of several, perhaps dozens, of private companies between the web server and you.

Uploading and Downloading

Historically, we thought of the Internet as consisting of powerful corporate "server" machines located "above" our little home computers. So when we retrieved material from a server, we were said to be "downloading," and when we transferred material from our machine to a server, we were "uploading." Many personal machines are now so powerful that the "up" and "down" metaphors are no longer descriptive, but the language is still with us. See the Appendix, and also the explanation of "peer-to-peer" in Chapter 6, "Balance Toppled."

This flexible and constantly changing configuration of computers and communication links developed because the Internet is in its essence not hardware, but protocols—the conventions that computers use for sending bits to each other (see the Appendix). The most basic Internet Protocol is known as IP. The Internet was a success because IP and the designs for the other protocols became public standards, available for anyone to use. Anyone could build on top of IP. Any proposed higher-level protocol could be adopted as a public standard if it met the approval of the networking community. The most important protocol exploiting IP is known as TCP. TCP is used by email and web software to ship messages reliably between computers, and the pair of protocols is known as TCP/IP. The Internet might not have developed that way had proprietary networking protocols taken hold in the early days of networking.

It was not always thus. Twenty to thirty years ago, all the major computer companies—IBM, DEC, Novell, and Apple—had their own networking protocols. The machines of different companies did not intercommunicate easily, and each company hoped that the rest of the world would adopt its protocols as standards. TCP/IP emerged as a standard because agencies of the U.S. government insisted on its use in research that it sponsored—the Defense Department for the ARPANET, and the National Science Foundation for NSFnet. TCP/IP was embedded in the Berkeley Unix operating system, which was developed under federal grants and came to be widely used in universities. Small companies quickly moved to use TCP/IP for their new products. The big companies moved to adopt it more slowly. The Internet, with all of its profusion of services and manufacturers, could not have come into existence had one of the incumbent manufacturers won the argument—and they failed even though their networking products were technologically superior to the early TCP/IP implementations.

File formats stand at a similar fork in the road today. There is increasing concern about the risks of commercial products evolving into standards. Society will be better served, goes the argument, if documents are stored in formats hammered out by standards organizations, rather than disseminated as part of commercial software packages. But consensus around one *de facto* commercial standard, the `.doc` format of Microsoft Word, is already well advanced.

Word's `.doc` format is proprietary, developed by Microsoft and owned by Microsoft. Its details are now public, but Microsoft can change them at any time, without consultation. Indeed, it does so regularly, in order to enhance the capabilities of its software—and new releases create incompatibilities with legacy documents. Some documents created with Word 2007 can't be opened in Word 2003 without a software add-on, so even all-Microsoft offices risk document incompatibilities if they don't adjust to Microsoft's format changes. Microsoft does not exclude competitors from adopting its format as their own document standard—but competitors would run great risks in building on a format they do not control.

In a large organization, the cost of licensing Microsoft Office products for thousands of machines can run into the millions of dollars. In an effort to create competition and to save money, in 2004 the European Union advanced the use of an "OpenDocument Format" for exchange of documents among EU businesses and governments. Using ODF, multiple companies could enter the market, all able to read documents produced using each other's software.

In September, 2005, the Commonwealth of Massachusetts decided to follow the EU initiative. Massachusetts announced that effective 15 months later, all the state's documents would have to be stored in OpenDocument Format. About 50,000 state-owned computers would be affected. State officials estimated the cost savings at about $45 million. But Eric Kriss, the state's secretary of administration and finance, said that more than software cost was at stake. Public documents were public property; access should never require the cooperation of a single private corporation.

Microsoft did not accept the state's decision without an argument. The company rallied advocates for the disabled to its side, claiming that no available OpenDocument software had the accessibility features Microsoft offered. Microsoft, which already had state contracts that extended beyond the switchover date, also argued that adopting the ODF standard would be unfair to Microsoft and costly to Massachusetts. "Were this proposal to be adopted, the significant costs incurred by the Commonwealth, its citizens, and the private sector would be matched only by the levels of confusion and incompatibility that would result...." Kriss replied, "The question is whether a sovereign state has the obligation to ensure that its public documents remain forever free

OpenDocument, Open Source, Free

These three distinct concepts all aim, at least in part, to slow the development of software monopolies. OpenDocument (`opendocument.xml.org`) is an open standard for file formats. Several major computer corporations have backed the effort, and have promised not to raise intellectual property issues that would inhibit the development of software meeting the standards. Open source (`opensource.org`) is a software development methodology emphasizing shared effort and peer review to improve quality. The site `openoffice.org` provides a full suite of open source office productivity tools, available without charge. Free software—"Free as in freedom, not free beer" (`www.fsf.org`, `www.gnu.org`)—"is a matter of the users' freedom to run, copy, distribute, study, change, and improve the software."

and unencumbered by patent, license, or other technical impediments. We say, yes, this is an imperative. Microsoft says they disagree and want the world to use their proprietary formats." The rhetoric quieted down, but the pressure increased. The stakes were high for Microsoft, since where Massachusetts went, other states might follow.

Three months later, neither Kriss nor Quinn was working for the state. Kriss returned to private industry as he had planned to do before joining the state government. The *Boston Globe* published an investigation of Quinn's travel expenses, but the state found him blameless. Tired of the mudslinging, under attack for his decision about open standards, and lacking Kriss's support, on December 24, Quinn announced his resignation. Quinn suspected "Microsoft money and its lobbyist machine" of being behind the *Globe* investigation and the legislature's resistance to his open standard initiative.

The deadline for Massachusetts to move to OpenDocuments has passed, and as of the fall of 2007, the state's web site still says the switchover will occur in the future. In the intervening months, the state explains, it became possible for Microsoft software to read and write OpenDocument formats, so the shift to OpenDocument would not eliminate Microsoft from the office software competition. Nonetheless, other software companies would not be allowed to compete for the state's office software business until "accessibility characteristics of the applications meet or exceed those of the currently deployed office suite"—i.e., Microsoft's. For the time being, Microsoft has the upper hand, despite the state's effort to wrest from private hands the formats of its public documents.

Which bits mean what in a document format is a multi-billion dollar business. As in any big business decisions, money and politics count, reason becomes entangled with rhetoric, and the public is only one of the stakeholders with an interest in the outcome.