Papier und digitale Kodierung: Selbsterklärende Codes

Paper and Digital Encoding: Toward Self-Explaining Codes

Florian Müller, Peter Fornaro, Rudolf Gschwind
Imaging and Media Lab, Universität Basel
Bernoullistrasse 32, CH-4056 Basel
Tel: +41 61 267 04 89, Fax: +41 61 267 04 85
E-Mail: florian.mueller@unibas.ch, Internet: www.iml.unibas.ch

Zusammenfassung:

Digital Archivierung auf Computersystemen ist problematisch: Trägermedien können zerfallen, Lesegeräte werden im Zuge des technologischen Wandels unverfügbar und Programme sowie Datenformate werden obsolet und nicht mehr ausführ- oder interpretierbar. Bekannte Techniken zur Handhabung dieser Problematik – Migration und Emulation – sind verfügbar, jedoch teuer und riskant. Mit PEVIAR (Permanent Visual Archive) entwickelt das Imaging und Media Lab der Universität Basel ein digitales Archivierungssystem, das eine Lösung für die beschriebenen Probleme bietet. Dieser Aufsatz konzentriert sich auf das Problem der Software- und Formatinkompatibilität und beschreibt Methoden, die einen langfristigen Erhalt von digitalen Objekten und ihrer Interpretierbarkeit erlaubt.

Abstract:

Digital archiving in the context of computers is confronted with several problems, among them material decay of carrier media, obsolescence of data access hardware and obsolescence of software for device and data access as well as digital object formats. Today, there are two well-known techniques for dealing with these problems: migration and emulation. Both techniques, however, are both costly and risky. With PEVIAR, the University of Basel's Imaging and Media Lab investigates a new digital archiving system that provides a solution to saidissues. This paper focuses on the problem of software and format obsolescence and describes methods that allow long-term preservation of digital objects and the ability to interpret them.

1 Digital Archiving

In a digital archive, information represented by digital data is preserved for long periods of time. Digital archiving is an important topic in computer storage technology and as of today still poses many unsolved problems. We would like to point out three of them. First, digital data is always stored on a carrier medium, and as any material object, these carriers are subject to decay. Typical lifetimes of digital data carriers range from around 5 to 30 years (CD/DVD, magnetic tape). After a certain level of decay has been reached, the data on the damaged carrier is lost completely. Second, the data carriers are accessed through special hardware devices. Data storage technologies are in constant development, and the lifetime of specific technologies is severely limited. Data access hardware can quickly become incompatible with current computer architectures or simply unavailable. Again, complete loss of information is possible. Third, digital objects are always encoded in a specific form and are of a certain format. Software formats may become obsolete, and accessing information stored in obsolete formats can be impossible or involve considerable efforts in migration or emulation techniques.

The PEVIAR (Permanent Visual Archive) project is developing a digital archiving system that effectively solves the mentioned problems (see [1] for details). It uses photographic microfilm as its carrier and stores data in the form of a visual two-dimensional barcode. Microfilm is a very stable carrier, offering lifetimes of up to 500 years [2]. The technology required for data access is nonspecific and basic: in order to decode information stored on microfilm, only optical magnification, image capture and software image processing are required, all technologies quite certainly available (and probably improved) in the future. The question of software formats can be addressed through the hybrid nature of microfilm. In the visual domain, placement not only of digital data, but of arbitrary analogue information is possible. This allows the inclusion of metadata that is directly readable and that helps in understanding the encoding and format of the digital data on the film. This property can help in overcoming problems of obsolete formats in digital archiving. The effectiveness and the practicability of the approach is discussed in more detail later. First, a broader perspective on digital archiving is helpful.

Digital archiving is not an invention of the computer age. In fact, there are cultural and natural techniques of digital archiving that predate computers by millennia. Many human languages that have developed a written form use digital alphabets. The Latin alphabet, for example, has 26 letters, and including upper and lowercase letters as well as special characters and numbers, cultures using the Latin alphabet are able to digitally archive their written texts using between 60 and 100 digits. This is equivalent to around 5 to 7 bits encoded per digit. In a broader perspective, nature uses a 4 digit code (the DNA) to store vital information for living cells, i.e. a 2 bit code. Procreation of life can be seen as a technique to preserve the information digitally encoded in the DNA from generation to generation. In this context, the computer uses merely a special case of a digital alphabet, namely the binary alphabet with two digits, where each digit encodes one bit. Being a cultural technique, digital archiving of written texts is quite similar to digital archiving of computer data. Digital archiving of text has allowed us to preserve information created thousands of years ago using only digital alphabets, paper and an organizational workflow. This organizational workflow is crucial. It guarantees that the destruction or decay of the carrier medium does not lead to information loss, creating copies of the text distributed across space and time. Also, it is able to adapt the text in order to allow its continued understanding: it can provide annotations and accompanying texts, or it can rewrite the text in a form compliant with current syntactical standards.

The case of digital archiving of written texts is most interesting in the case where the organizational workflow is disrupted. If a book is forgotten, say for 100 years, and then found, what are the chances that it can still be read? Quite good. If the paper has not been kept in a bad environment, it is still readable. The language the book was written in is probably still spoken and while it has evolved, its speakers are still able to understand it in the now archaic form. In comparison, if a magnetic tape is forgotten for 100 years, things do not look so good. Even if the tape is still intact and the magnetization has not been lost, the tape drive used to record it is certainly not available any more. If we are lucky, we are able to find an old tape drive (which may still work), and if we are really lucky, we find alongside an old computer system that allows the connection of this tape drive - for most certainly, it cannot be connected to state-of-the-art computers. And these dependencies are only for data access. Once we have recovered the data, we still have to interpret it.

With the use of a visual data carrier[1], the archiving of digital data originating in the electronic world of computers moves closer to the domain of the established and successful archiving of digital data on paper. Through careful study of the tradition of written texts, it is possible to create a digital archiving system that allows the preservation of our current digital heritage beyond the horizon of computer technology lifecycles.

2 Information and Prior Knowledge

2.1 Information

A digital storage system is a physical entity that in the best case allows us to preserve information over long time spans while keeping it reasonably accessible. For now, we understand ‘information’ to be the result of the interpretation of a structured set of noticeable physical properties. The physical properties are directly accessible to the human senses, and their interpretation can either

[1] By visual, we mean that the human eye can directly see the information. This is opposed to the case of optical storage: while the information is also read using light (laser), it cannot be read without the use of specialized hardware devices.

be based on said sense experiences (audio, video) or on their further processing through a learnt ability (such as reading). This understanding of information covers, among others, written text, images on paper or other carriers and audible sound.

If we want to store information in digital computer systems, we are not in an arbitrary domain, but in the binary digital domain. All information will ultimately be stored as zeroes and ones and as physical properties that represent binary digits (electrical voltage, magnetization etc.). Since these are not directly accessible to human senses and learning the ability to process zeroes and ones would be very hard for humans, computers are needed to present the information represented by binary sequences in a human-readable form. We call this presentation rendering. An example is the displaying of an image or of a page of text on the screen. We so far have a two-stage process of processing: first, the computer accesses data on a medium and renders it to the user. Second, the user perceives the rendering of the computer and gains information from it. In this work, we are concerned with the first stage. Note that in the case of digital text on paper, the first stage is not applicable: we can see text on paper without any further ado.

2.2 Encoding and Codes

Encoding is closely related to information. If we have some Information I_n and a representation of that information R_n, then R_n is said to encode I_n. An example would be the fact that a train departs from the main station at noon (I_n) and the written English sentence 'The train departs from the main station at noon' (R_n). When we read the sentence and obtain the information encoded in it, we decode the information (more precisely, we actually decode the encoded form of the information). The same information can be encoded in many ways, and any specific form of encoding of information is called a format. For example, given an image of an object, we can encode it using any desirable image file format (jpeg, tiff, bmp, etc.). Encoded information is only accessible through the appropriate decoding process. Without knowledge of the decoding process, we can have no knowledge of the information.

The encoding as well as the decoding is part of the same code. The code is a set of rules that describes the encoding and the decoding process and that is the actual relation between an information I_n and its representation R_n. Without the appropriate code, we are unable to relate information and representation. To go back to our image example: if we have an image file, that is, a sequence of zeroes and ones, but are unaware of the image file format (or the fact that it is an image at all), then this sequence is just that: zeroes and ones.

2.3 Prior Knowledge and Obsolescence

The decoding of information requires prior knowledge about the code used in the encoding. We call this the prior knowledge requirement. You are only able to read this text and obtain the information contained in it (that is, decode it) because you have knowledge of the Latin alphabet and the English language. For someone without that knowledge, the information is inaccessible. Knowledge is a human concept and does not apply to computers. We call the equivalent of knowledge in the case of a computer a capability, usually provided by software. The prior knowledge requirement is the requirement of the availability of specific software.

The problem of unavailable software required for decoding information is broadly known as the problem of obsolescence. A file format for which appropriate software no longer exists is called obsolete. If we want to preserve information, preservation of the knowledge required for decoding is essential. For written text in a certain language, this implies the preservation of the knowledge about that language. So far, we have been quite successful in doing so: we (or some of us) are still able to read and understand Latin and Greek texts, and the information contained in them has been well preserved. Unlike human languages, computer software and file formats are not only very young, but also short-lived. The constant advancement of data and executable file formats leads to incompatibility that, if nothing is done, results in a loss of information due to unavailable prior knowledge.

There are two ways of dealing with obsolescence: migration and emulation. Details on these are not provided here, but they shall be outlined briefly. In migration, the format in which given information is encoded is periodically altered, reflecting the advancement in format development. Through this process, the encoding always has a form supported by current software environments. In emulation, the format in which the information is encoded remains unaltered. Instead, the software environment in which the format is supported is preserved as well. Because this software environment itself is encoded in a specific way (it consists, ultimately, of executable instructions that are encoded in a form suitable for contemporary computer architectures), it is in danger of becoming unsupported by future computerarchitectures. Therefore, a special program, the emulator, is created. It emulates an architecture on which the software environment is supported and allows the operation of that environment within the emulator. The emulator itself, of course, must be executable on whatever architecture is current, and therefore must be migrated. Instead of migrating the digital objects themselves (which may be very diverse and numerous), one single emulator is migrated and allows accessing the original objects on any future platform. Migration and emulation actually work, but they require a lot of effort, and they require it periodically.

3 Self-explaining Code

In order for an encoding to be supported (i.e. we have the ability of decoding and gain access to the encoded information), prior knowledge is necessary. The prior knowledge can be considered the explanation of the code. When we speak of self-explaining codes, we mean codes that are such that they contain the prior knowledge necessary for their decoding. The problem structure seems recursive: for this prior knowledge must be encoded in some form, and this additional encoding would require additional prior knowledge. In order to avoid this structure, we introduce a constraint. We do not want to eliminate the prior knowledge requirement, but minimize it. The required prior knowledge serves as an entry point for acquiring the knowledge for the decoding of the code.

So far, the notion of self-explaining does not provide us with details about the design and the actual properties of the code. However, the environment of their implementation - the PEVIAR microfilm - allows us to capture some of their specifics. Our self-explaining codes are visual in nature (they are visible on the microfilm) and are spatially distributed in two dimensions (they are spread out on the microfilm). The digits, which are geometrical forms, may be colored (i.e. we can use a higher order alphabet on color film). Codes defined by visual marking in one or two dimensions are called one- or two-dimensional barcodes. They are widely available and employed, and they allow the encoding of constrained numbers or character strings (one-dimensional barcodes, see [3]) or of arbitrary digital data (two-dimensional barcodes, see [4], [5]). Decoding of barcodes consists in processing their image and obtaining a sequence of binary digits form their visual representation. The image of the barcode is data on the spatial distribution of luminance intensity and color, and the decoding actually describes how we get from that distribution to a sequence of binary digits. State-of-the-art barcodes are not trivial to decode. They employ forward error correction to allow compensation of damaged code surface. In order to explore the possibility of self-explaining barcodes, we will take as an example a hypothetical and very simple barcode, which we will call the simple barcode.

3.1 Simple Barcode

The simple barcode consists of squares that are either black or white, and that are ordered in sequence and placed on one line. Every instance of the code contains exactly eight squares. The simple barcode is a binary code and is able to encode one byte (256 states, such as the numbers from 0 to 255). In order to provide an explanation for the code, we could place an image of this code with eight alternating black and white squares next to the image of the number sequence '01010101' (see Figure 1). This states that a black square encodes a '0' and a white square encodes a '1', and that the order of the squares is the same as the order of the bits. If we want to encode text, we could go one step further and provide a relation between numbers and letters, namely the ASCII (American Standard Code for Information Interchange) text encoding. This would state that the number '97' encodes the letter 'a', '98' encodes the letter 'b', and so forth. The possibility to provide such a dictionary-like mapping between different forms of representation distinguishes a visual medium from other media such as magnetic tape. This mapping is not intended for automated processing, but would provide the basis for the implementation of software that can process the barcode in question and produce the respective output.

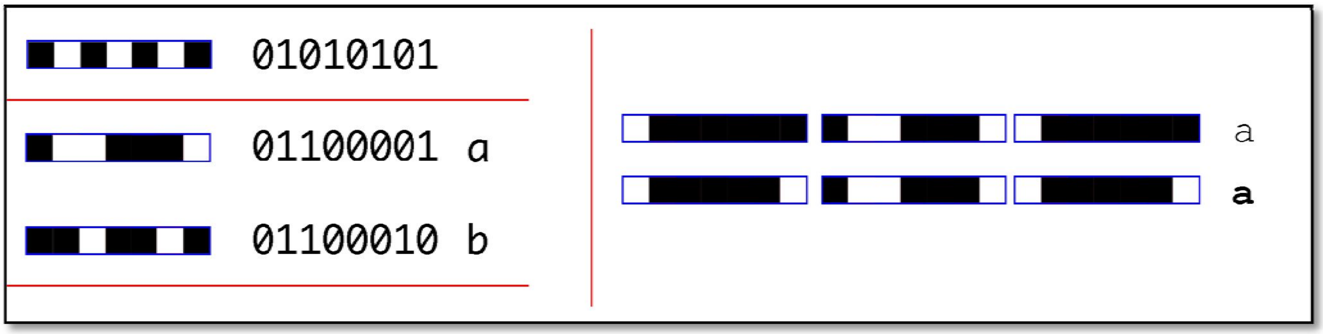

Figure 1 - Simple Barcode Examples

The mapping could be cascaded further and applied for other cases.The right example inFigure 1 shows the possible use of text markup. Since 7 bits are sufficient for encoding the original ASCII, the leftmost bit of the code could be used to designate formatting instructions. Here, '1000000' encodes regular font weight and '10000001' encodes bold font weight.The tag structure is implicit in the notation.It would also be possible to define a mapping according to this scheme for image file data and actual images (i.e. express the relation that three numbers at certain locations in the barcode make up the RGB values of a pixel at a certain location in the rendered image). While such mappings are certainly not a suitable format for every-day data storage, their strength lies in their independence from existing software: if we have a text or an image encoded in this form, we are able to decode it by interpreting the given code explanation. The encoding and decoding processis asymmetric: in order to encode information with a self-explaining code, this code must be constructed and applied, and original data must be transformed. Encoding is computationally more expensive. At decoding time, only the explanation of the code – which is designed to be minimal – must be implemented, and the information can then be rendered.

4 Outlook

With the application of self-explaining code, the PEVIAR archiving system is able to demonstrate a solution to three fundamental problems of archiving (hardware and software obsolescence, carrier decay). Our approach to the format problem will be further generalized in order to support as many formats as possible. We hope to come up with encoding schemes that allow the highest possible degree of automation. In a first reference implementation, we will cover structured text and uncompressed image data encoding.

References

[1] A. Amir, F. Müller, R. Gschwind, J. Rosenthal, L. Rosenthaler: Towards a Channel Model for Microfilm. IS&T's 2008 Archiving Conference, Bern, June 2008. Conference Proceedings. IS&T: The Society for Imaging Science and Technology, Springfield (VA), USA.

[2] Armin Meyer, Daniel Bermane: The Stability and Permanence of Cibachrome Images, in: Journal of Applied Photographic Engineering. August 1983, vol. 9, no. 4; pp. 117-120.

[3] ISO 15420: Information technology – Automatic identification and data capture techniques – Bar code symbology specification. ISO, 2006.

[4] ISO 16022: Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification. ISO, 2006.

[5] Microsoft Research: High Capacity Color Barcodes. See http://research.microsoft.com/research/hccb/