DIGITIZATION AND LONG-TERM ARCHIVING OF PHOTOGRAPHIC MATERIAL

Lukas Rosenthaler
Imaging & Media lab. Universitat de Basilea. Suïssa

Introduction

Long-term archiving of photographic materials has always been a difficult task. The majority of photographic processes have not been designed with longevity in mind. Other goals such as color reproduction, ease of development and last but not least the costs have been more important. Therefore, photography is inherently unstable and *all* photographic material will decay with time. While B/W-photographs have a life expectancy of about 100 years (until the first effects of decaying become visible), color photography is usually decaying much faster. Even modern photographic material, which is chemically more stable, will decay with time. The decay rate can vary and is dependent mainly on storage conditions but also on materials used, processing methods, handling etc. There is one type of photographic material, which has a much longer live expectancy: Microfilm, both B/W and color microfilm[1] have a longevity of more than 500 years if treated properly.

Since many objects in archives are unique artifacts irrecoverable in case of damage or total loss, every handling of these items always poses great risks for damage or total loss. Often, the handling is also impractical and cumbersome. Hence many photographs in collections and archives are being digitized. In general the digitized object has lost all "materiality" – it is an immaterial representation of a part of the original object. A digitized photograph can capture only the visual content of the original photograph whereas most aspects of the original material such as physical properties (thickness, surface properties, smell etc.) will generally be lost. Metadata that describes the original object is often only an insufficient substitute for the materiality.

Most digitization projects have been launched or are conceived without long-term archival in mind. These projects have been started for improving access to the assets through databases and the Internet. But as soon as the first digital data arrives, the question of archival arises: the digitization process is slow, expensive and cumbersome and therefore will usually not be repeated in foreseeable time. In addition, every additional handling of the original photographs should be avoided in order to minimize the risk of damage. As a consequence, anybody involved in digitization processes is suddenly facing the problem of long-term archival of digital data.

In order to complicate the situation, most photographs created today are "born digital", this means that they are of digital origin. But not only photographs, also motion pictures, computer animations, and videos using modern technology result in "originals" of digital nature. This fact will increase the pressure to find solutions for digital long-term preservation "real soon now".

[1] i.e. Ilfochome Microgaphic

From our daily experience, digital data seems to be very volatile and unstable. Everybody working with computers of any scale has had the bad experience of data loss, be it a word-processor document that becomes unreadable, an external storage medium that cannot be accessed anymore etc. It looks like "long-term archival" and "digital" are diametrically opposed concepts. However, the very properties of digital data comprise the possibility of an unlimited storage in time.

Digital Data and Information

Introduction

Digital images do not exist – there is only ***digital data*** *that* ***represents*** *an image!* This axiomatic fact has always to be kept in mind when dealing with digital images. Digital data representing an image always has to be converted by some technical means (LCD screen, a beamer, a printer etc.) into an analogue distribution of light, which human beings will perceive as image. The digital representation of an image – aka the "digital image" – can be *created*, *manipulated* and *analyzed* by computer algorithms, but the computer can not *see* an image.

In a broad sense, digital data can be defined as anything recorded using a symbol based code on a medium. Such a code uses a finite set *S* of Symbols

$$S = \{s_1, s_2, .., s_n\}, n \geq 2$$

as for example the Latin alphabet, Egyptian hieroglyphics etc. If $n = 2$ and thus the code uses only two symbols, it would be called a "*binary code*". Binary codes are the simplest codes, which can easily be implemented by computing machinery[2].

Digitization

Most digital data is the result of the conversion of an analogue physical signal that may vary in time, space and other properties. For example, a color movie camera records the amount of light falling onto a light sensor as a function of time and location within the boundaries of the sensor and frequency (color) of the light. The digitization process converts such an analogue signal into a series of symbols allowed by the code:

$$F(x, y, z, t, ...) \rightarrow s_i, s_j, s_k, ... \in S$$

Usually these symbols represent numbers related to the physical amplitude of the analogue signal. In order to understand such a code, the meaning of the code has to be known. E.g., the numbers (53,130, 211) at the position 23'877 in a file may represent the intensity of the light in red, green and blue, where the number 0 relates to an intensity of 0 and the number 255 to the maximal intensity the light sensor may capture. The position 23'877 indicates that the values have been measured at a location that is 3.75mm from the left and 2.15mm from the right of the upper left corner of the sensor. In order to properly interpret these numbers, additional information should be known, e.g. the characteristics of the color filters used to capture the light in the red, green and blue

[2] E.g. S={0,1}, S={true, false}, S={+5V, -5V} or S={↑, ↓}

areas of the light spectrum or if the values of the recorded numbers increase linearly with the intensity of light etc. Usually not all of these parameters are known, but reasonably assumptions may be made. However, especially when dealing with unique historic material, much more knowledge is required in order to get a meaningful and useful digitization result.

In principle, the digitization process involves two distinct parts:

1. *Sampling or Rasterization*
   The physical properties such as time, space, etc. have to be measured at well-defined, discrete points. In the time-domain this process is usually called *sampling* and the *sampling frequency* defines how much time elapses between two measurements.
   a. *Space sampling*
      In space usually the term *rasterization* is used and the distance in between two points where the physical property is measured gives the *resolution*. It is evident that the quality of the sampled signal is directly correlated to the sampling frequency: the higher the sampling frequency, the better the quality. The minimal sampling frequency or resolution is given by the Nyquist-frequency, which is defined as 2-times the highest frequency occuringin the signal. If the sampling frequency is below this limit, irreversible artifacts such as distortions or moiré-patterns may occur. A well known example can be seen in many western movies: the wheels of a stage coach suddenly seem to turn backwards (e.g., during a chase) if the frequency of the passing spokes of the wheel is higher than the sampling frequency of the film (which is 24 images/second).
   b. *Color Sampling*
      In case of color imaging, the frequency of the light is also sampled. Usually only three samples are taken (corresponding to red, green and blue), but there are multispectral cameras and scanners which use many more sampling points[3]. The use of only 3 samples is possible because the color vision of the human eye also depends on 3 types of light receptors that are sensitive to the red, green and blue part of the visible light. However, since the sensitivity curve of the cones in the human eye usually differs from the response of the color filters used in digital sensors, color artifacts may occur. Such metamerisms may surface in two ways. First, different colorants that have the same color to the human eye may differ in the digitized image. Second, colorants that are different to the human eye may result in the same numbers during digitization and thus will be indistinguishable in the digital image. Both effects are also dependent on the illumination. But these are effects usually only important in digital photography, e.g. for the digitization of paintings, objects etc.. The digitization of color photography is less critical, because color film itself is composed of only three layers, sensitive to the red, green and blue parts of visible light spectrum.

[3] See for example. A. Ribes, H. Brettel, F. Schmitt, H. Liang, J. Cupitt and D. Saunders, *Color and Spectral Imaging With the Crisatel Acquisition System*, PICS03 The Digital Photography Conference (2003)

2. *Quantization*
   The analogue measurements taken at the sampling points have to be converted into a numeric value in a symbolic representation using a finite number of symbols. Therefore the accuracy of such a conversion is *always limited*. Most often modern computing machinery uses a binary code with the number of symbols being a multiple of 8. The number of different values that can be represented is given by the number of binary digits (called *bits*) raised to the power of 2 (n = 8 → 256 levels, n = 12 → 4096 levels, n = 16 → 65536 levels). A quantization using *n* bits is said to have a *bit depth* of *n*.

Thus, the result of a digitization process is a series of symbols, which *represent* the original analogue signal. In theory, the accuracy of such a symbolic representation is not limited and can be increased by increasing the sampling frequency and the bit depths. However, also the size of the resulting digital code will grow accordingly. In practice, the mechanical, electronic and other properties of the digitization process will limit the achievable accuracy.

Media convergence

Digital data is able to represent most media types, be it text, sound, image, moving image or new media types such as hypertext or relational databases etc., in a unified way. In the end, everything is just a *bit stream*.

Recording of digital codes

In practice, there is another obstacle: a digital code is only an idealistic, logical construct: in any case, the symbols of a code have to be *physically* represented by some physical property that in its essence always has an *analogue* nature[4]. Examples for such properties are the color of the ink which makes the shape of the symbols written on a piece of paper visible, it may consist of a change of the direction of a magnetic field on the surface of a ferro-magnetic material or it may consist of a pit imprinted on the reflective layer of a Compact Disk. Thus, every recording process, which makes a digital code "permanent", leaves *analogue* physical marks on the recording medium. These marks *represent* the symbols of the digital code.

In order to read back the data, these analogue marks have to be identified and the appropriate digital symbols have to be assigned to the marks. This is done reading the analogue signal and applying a *decision process*. In case of a binary code, these decisions may be made by a simple thresholding of the physical signal, but often, signal processing and more complex procedures like shape recognition etc. are required. Thus there is no such thing like a "digital recording" of data. All "digital" recordings are analogue in nature and during the read-process the digital code is generated "on the

[4] This statement is not true in the world of quantum physics: in this world there are physical properties like the spin of an electron, which do have 2 distinct values: up or down. From this point of view, the spin of an electron might represent a bit in a true digital way.

fly"[5]. Whereas *binary* codes that require only two distinct symbols or states are the most common encodings, there are many examples of encodings using more symbols, the most prominent being written text.

However, in the world of computing machines, the binary code using only two symbols '0' and '1', 'true' and 'false' or, as the information is represented in the computer, '+3.3V' and '-3.3V' has been predominant since many decades. It is a code that is easy to implement because the binary decision process is the most simple process possible and thus very robust. Nevertheless, in practice errors do occur and special mechanisms have to be used to detect and correct such errors of the decision process. Especially for the permanent recording of digital data, mathematical methods called Error Correction Codes (ECC) play an important role (see below).

Reading and decoding of digital codes

Since digital data is recorded using analogue marks on a physical medium, the first step in reading and decoding digital data usually requires the identification of the recording method. While this seems a trivial task given that basic recording methods (optical, magnetic) can easily be distinguished, the devil is in the details. While enclosure and form factor of modern tapes of the same family (e.g DAT, LTO DLT,…) are the same, the recording standards have changed significantly going through the different generations. The following table shows some of the relevant parameters of the LTO family:

	Year	Capacity	Parallel tracks	Tracks written/pass
LTO-1	2000	100 GB	384	8
LTO-2	2003	200 GB	512	8
LTO-3	2005	400 GB	704	16
LTO-4	2007	800 GB	896	16
LTO-5	2010	1400 GB	1280	

Recent generations obviously do have a much higher data density, which translates to narrower magnetic tracks. Usually magnetic tape machines are therefore only able to read and write tapes one generation back and read tapes two generations back. Older tapes can neither be read nor written.

From this follows that the first step in reading and decoding digital data is the identification of the specific recording method of the data on a data carrier and to procure a machine that is able to read the specific physical marks. Since usually the lifespan of a specific recording technology is only a few years, this first step may prove to be difficult. Finding a tape machine to read an LTO-1 tape or a drive to read an old Iomega ZIP-dive may be extremely hard today.

Decoding a digital code is the process of extracting *meaningful information* from a sequence of symbols. In order to read and understand a written text, not only the symbols – the characters or signs – have to be identified, but also the language – syntax

[5] Also within a computer the same thresholding takes place. The 0's and 1's are represented by electrical current or voltage, and somewhere there must be defined a "tipping" point which distinguishes in-between the two states.

and semantics –, in which the text has been written, must be known. Only the famous Rosetta stone that contains the same text in two languages (Egyptian and Greek), using three scripts (hieroglyphic, demotic and Greek), finally allowed reading and understanding hieroglyphic scripts. Every digital code has explicit or implicit syntactic and semantic rules, which have to be known in order to interpret the code properly.

The meaning of a symbol often depends on the *position* within the sequence of symbols. Frequently, symbols are combined in groups to form new symbols (commonly known as "words") which themselves are combined into higher units ("sentences"). Computers which use binary codes usually use words with a size of 8, 16, 32 and 64 bits. The rules which define the syntax and the semantics of a digital code are usually known as the *file format*. The knowledge of the file format is therefore essential for reading digital data. Explicit knowledge is given by *open* file formats, where a detailed and complete description of the syntax and semantics is available. Such a description may be given by a textual description in plain English (or any other common language) or by the (hopefully commented) source of a computer program that reads the data. For *proprietary* file formats, no such description is available. At most, a binary program or library is available for interpreting the digital code of proprietary file formats.

Thus, reading and understanding a digital code requires two distinct steps:

1. The recording method has to be identified.
2. The file format has to be identified.
3. The appropriate syntactic and semantic rules of the file format have to be applied in order to interpret the digital code.

If a data file is identified as being a TIFF image file, but the specification of the TIFF format is not known, the image represented by the data cannot be extracted. Therefore, if the semantic system of a digital code can not be identified or is not known, the information contained in the digital data can not be extracted.

This leads to the following prerequisites for retrieving digital data:

1. The physical property used to create the marks has to be known. For current media types the physical property used to create the marks is usually known. We *know* that a floppy disk has magnetic marks and that a CD-R has optically detectable marks. However, future digital archeologists might have problems to determine which physical property has been used to create the marks, especially for new, emerging recording technologies.

2. The physical marks on the medium must be detectable and convertible into symbols. If, due to damage and ageing, this is no longer possible, the medium has to be considered as "destroyed" and unreadable.

3. The syntactic and semantic system (file format) has to be identified and known.

If any of these tasks cannot be accomplished, the digital data will no longer be readable and the recorded information is lost.

Redundancy, lossless compression and error correction codes

Claude Shannon's fundamental work about information theory "A mathematical theory of communication" [6] contains two important statements, given here in a simplified form:

1. Any code where the probability of occurrence is not the same for all symbols contains redundancy. In such a case it is possible to find a new code which minimizing redundancy.

2. If a communication path introduces errors into the transmitted symbols, a new code can be found allowing correcting for these errors.

The first statement addresses the possibility of *lossless compression* whereas the second statement deals with the possibilities of *error correction codes*. Shannon's theory shows that there is a tradeoff between efficiency (lossless compression) on the one hand and error correction (redundancy) on the other hand. Many codes such as the written language contain a lot of redundancy and are therefore quite fault tolerant. For digital computer systems however, a high efficiency is required and therefore often compression techniques are used.

It is interesting to know that most computer storage devices use internally some redundancy in order to gain error correction capability. In fact, disk drives, optical drives and magnetic tape drives would not work without internal error correction. The conversion of the analogue physical signal into distinct symbols has a non-negligible probability of being wrong. Even with error correction, a typical modern hard-disk has a non-zero probability of error: statistically 1 out of 10^{14} bits is wrong[7]. These errors are called *non-recoverable read errors*. This results statistically in one corrupted file each 25th time a 400GB hard-disk is copied. The CD-ROM and CD-R technology would not work without error correction. In fact, a CD-R would have a raw-capacity of more than 1 GB, but 33% percent of the raw capacity of a CD-ROM or CD-R are used for adding redundancy for error correction.

Checksums

Checksums are the digital equivalent of human fingerprints. For a given sequence of digital symbols (e.g. a bit stream or a data file), the ideal checksum algorithm calculates a *unique* new sequence of symbols which is usually much shorter than the original sequence. However, such ideal algorithms do not exist. In practice, the probability that two files that differ will have the same checksum is negligible. There are many checksum algorithms available which producing checksums of different length. The most common checksum types are given in the following table, including the resulting checksums for the ASCII-coded text "*To be, or not to be: that is the question*":

Checksum-Algorithm	*Checksum*
MD5	eaf606c87569b2f97e230e792049833e
SHA-1	71d7726d2db38295ddea57c5dccd3be388fc0ab5

[6] Claude E. Shannon, A mathematical theory of communication, Bell System Technical Journal (1948)

[7] This value is taken from the data sheet of a major hard-disk manufacturer for a 400GB IDE hard-disk.

RIPEMD-160	c5fd0db228230b0f0813ace8376150527bf24588
WHIRLPOOL	ca4685900bbc481f3d8a1c71b17512aa4c62b4fbda19df8969da9052a2c54ea2e7073ba5241838908cbc865a0d4ac6cc74e284f12f90f4dc90d9864d94fdd900

Checksums are therefore an ideal instrument to check if two digital files are identical: if both files have the same checksum, they must be identical.

Checksums are an ideal mean to check for the integrity of a digital file, since the checksum will be completely different if even one bit of the file has been changed. If the checksum of a file has been stored separately from the file, at any point in time the checksum may be calculated again and compared with the original checksum. If both checksums are identical, the file has not been changed, if not, the file has been corrupted. Therefore, checksums are also used to guarantee that a file copy process has been successful. If the "original" and the "copied" files do have the same checksum, the copy process has been without error.

Properties of digital data

From the precedent section "Digital Data and Information", the following properties of digital data can be deduced:

Loss of the notion of an "original"

For digital information, the notion of an *original* is meaningless. Digital data can be copied without any loss by reproducing the same sequence of symbols from the "original" sequence. The two copies will be indistinguishable from each other and therefore it is not possible to determine which one is the "original". However, since the physical representation of a digital code always has an analogue nature which may result in errors, the digital copy process is only completed if the two copies have been *verified* to be *identical* either by a symbol wise (or, in case of binary data, bitwise) comparison or by using checksums.

Therefore, digital data can be copied without limits and there will be no generational loss.

Independence of the recording medium

Digital data is independent from the medium it is recorded on as long as the symbols can be deciphered. For example a binary computer file representing an image using the JPEG format could be engraved into a stone – it would be not very handy to work with but nevertheless feasible. Thus, digital data can copied from one medium to any other without loss.

Nullification of space

Digital data can be transported through space with the speed of light without the need of moving atoms or matter. This property allows to *tele-copy* digital data without loss at the speed of light.

Sources of data loss

In order to understand the problems of long-term archiving of digital data, the possible sources of data loss have to be assessed:

1. *Failure in reading the bits*
   If the symbols of the code cannot be identified any more, the recorded data is lost. There are several causes for the inability to identify the recorded symbols (to "read the bits"):

   a. *Physical damage to the medium*
      The physical recording marks can no longer be read because of a damaged medium (ageing, rough handling of the medium, defects etc.)
   b. *No more reading device available*
      The medium cannot be read because the device needed is no longer available. For example, tape drives to read DLT I magnetic tapes are no longer commercially available.
   c. *Human error*
      The data recorded on the medium has been erased by human error, the medium has been mislabeled or the data that was supposed to was never written to it…

   Most often, the physical lifetime of a recording medium is longer than the lifespan of a specific recording system. Therefore the aging of the recording media, given that it is properly stored, is usually not the limiting factor. For example, according to John W.C. Van Bogart magnetic media (tapes) have a lifetime of about 30+ years[8]. However, the lifetime of the recording system as a whole depends on the availability of support for the necessary hard- and software by the manufacturer(s). This lifespan can be quite short (3-10 years) and is usually the limiting factor for the length of life of recorded digital data on one medium.

2. *Failure in reading the file format*
   There are several reasons why a file format cannot be read:
   a. *File format identification*
      If there is no metadata to indicate which file format has been used to write the data, it may be very difficult to identify the file format. There are many thousand of file formats in use[9]. Some of the most common formats can be easily identified by the so-called "magic number" consisting of the first 4 or 8 bytes of the file. However for all other formats the identification may be very difficult.
   b. *File format specification lost*
      If the file format can be identified, the next obstacle is to find software that can read and interpret this format. Either there is still usable software available that is able to read the data, or new software has to be written to read the format. The first is often hard to get, the latter however requires that the full specification of the format is available to the programmer.

[8] John W.C. Van Bogart, "Magnetic Tape Storage and Handling", National Media Laboratory (1995)

[9] See for example PNDesign, "Data formats, file extensions database", at http://extensions.pndesign.cz

Therefore proprietary formats, where the format specification is not available, are usually not suitable for long-term archival.

Table of common image file formats with magic number

File type	**Typical extension**	**Hex digits** **xx = variable**	**ASCII digits**
GIF format	.gif	47 49 46 38	GIF8
FITS format	.fits	53 49 4d 50 4c 45	SIMPLE
Bitmap format	.bmp	42 4d	BM
Graphics Kernel System	.gks	47 4b 53 4d	GKSM
IRIS rgb format	.rgb	01 da	..
ITC (CMU WM) format	.itc	f1 00 40 bb	....
JPEG File Interchange Format	.jpg	ff d8 ff e0	....
NIFF (Navy TIFF)	.nif	49 49 4e 31	IIN1
PM format	.pm	56 49 45 57	VIEW
PNG format	.png	89 50 4e 47	.PNG
Postscript format	.[e]ps	25 21	%!
Sun Rasterfile	.ras	59 a6 6a 95	Y.j.
Targa format	.tga	xx xx xx	...
TIFF format (Motorola - big endian)	.tif	4d 4d 00 2a	MM.*
TIFF format (Intel - little endian)	.tif	49 49 2a 00	II*.
X11 Bitmap format	.xbm	xx xx	
XCF Gimp file structure	.xcf	67 69 6d 70 20 78 63 66 20 76	gimp xcf
Xfig format	.fig	23 46 49 47	#FIG
XPM format	.xpm	2f 2a 20 58 50 4d 20 2a 2f	/* XPM */

File formats – as stated above – define the *meaning of the bits*. That is, the file format defines the semantics of the bits. Usually the file format can be described by a plain text in some human language. E.g. the basic specification of the TIFF image file format is 121 pages document in plain English, which describes in detail the structure and the semantics of the TIFF format. Since one file format can be used for different subtypes of a digital object (e.g. the TIFF format may be used for many different kinds of digital images), many file formats include so called *technical metadata* that describes such technical aspects as the resolution, bit-depth, colorimetric interpretation etc. of a specific image. These technical metadata are required to decode and render the content of a digital file properly and are thus integral parts of the file format. Technical metadata must not be confounded with *descriptive metadata*, which is related to the content the digital file represents. Many file formats allow adding at least some descriptive metadata to the content of a file.

3. *Loosing the descriptive metadata*
Most digital data files are meaningless if not accompanied by describing

metadata. A collection of 10'000 unknown CD-R's labeled "00001" to "10000", with each CD-R containing 50 files named "000.dat" to "049.dat" is almost worthless, if there is not more information available. The metadata can be as simple as a human readable, meaningful filename and a meaningful labeling of the data carriers, or it can be a complex XML-based metadata scheme. It matters, that there is some metadata available. Best praxis is to include some metadata into the data file itself. Many data formats (e.g. TIFF image files etc.) allow adding metadata into the *file header* that can be extracted automatically for efficient processing.

As a consequence, there are two basic problems regarding the longevity of digital data:

1. Aging or damage of the media the digital data is recorded on
2. Obsolescence of hardware, software and file formats. The rapid development of computer technology results in a system lifetime (hardware and software) of often less than 5 years. If the data formats are chosen carefully, their lifetime may attain 10 and even more years.

Methods of long-term archival of digital data

Kenneth Thibodeau, director of the "Electronic Records Archives Program" at the National Archives and Records Administration (NARA), identifies more than a dozen different methods for long-term archival of digital data[10]. These can be grouped into five basic methods:

1. *Nothing*
   Do nothing -- future digital archeologists will do the work for you...
2. *Computer museum*
   Archiving the whole system (media, peripherals and computer including all software) in working condition.
3. *Emulation*
   Emulation and virtualization of obsolete systems on up to date computers allows to run obsolete software on modern computers
4. *Migration*
   Copy (and convert to current, up to date format if necessary) the data onto a new medium, if the precedent medium starts aging or the format is becoming obsolete.
5. *Permanent media*
   Record the digital data on a medium with very high longevity either using a self-describing format or preserving the syntactical and semantic information (e.g. as *metadata*).

All of these methods do have advantages and drawbacks, but it seems that the first method -- having confidence in the ability of future digital archeologists -- is still quite prevalent, despite the obvious drawbacks it has.

[10] Kenneth Thibodeau, *Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years*, Conference Proceedings: The State of Digital Preservation: An International Perspective (2002)

Do Nothing

This is still a widespread method for the preservation of digital data. The reason is that all other methods require both at least some significant funding and some special knowledge. If one of each or both is not available, doing nothing – just keeping the data carriers and hoping, someone will be able to read and interpret the data in the future – may be the only alternative to actively destroying the data. While being dissatisfactory, keeping a data carrier in an adequate environment may at least slow down the decay of the material of the data carrier and of the physical marks written on it and therefore open the possibility that a future effort may bring back the data.

Computer museum: archiving media, hard- and software

The way to preserve not only the media, but also the hard- and software, seems to be a palpable solution to the problem of long-term archival of digital data. However, it is very difficult to maintain complex computing machinery in a functioning condition. Not only the media will age, but also all other components of the computers and their peripherals are not stable in time. Integrated circuits, circuit boards, solder joints etc. will age and at some point stop working. In addition, with the equipment getting older, it will become more and more difficult to find spare parts, to find the technical and repair manuals – and to find technicians who are still able to repair old technology. Therefore – as important preserved computing machinery is for digital archeology – it is not a generally advisable way to achieve longevity for digital data. However, in cases where old recording media are discovered in an archive or estate, it may be a “life saver” to have access to old computing machinery in order to transfer data once from old an recording medium to a new one.

Emulation

Software emulation creates a software environment allowing computer programs to run on a different platform (computer architecture and/or operating system) than the one they have originally been written for. Thus, writing an emulator for an old, obsolete computer system allows the old programs to run on a new computer. With the help of the emulator, file formats used on the old system can still be read on a new system. Emulation has the following problems with regards to long-term archival of digital data:

- Since the emulation is usually not able to emulate the peripheral devices such as floppy drives, magnetic tape drives etc., the data files and programs have to be migrated to modern media in order to be readable by the host system of the emulation.
- The emulation software itself will have to be preserved. This may be achieved by migrating it to new hardware and software (basically rewriting it for every new generation of computers) or by using a nested concept of hierarchical emulations (e.g. Wordstar[11] runs on a CP/M emulation which runs on an OS-9 emulation which runs on a Window XP PC). Raymond Lorie proposed a “Universal Virtual Computer” (UVC) in order to facilitate the migration of emulators[12]. The basic idea behind the UVC is to define a simple virtual machine that can be

[11] WordStar was a very successful word processor for the CP/M operating system and was developed in 1978.

[12] R. A. Lorie, *Long-Term Preservation of Complex Processes*, Proceedings of the IS\&T Archiving Conference (2005)

implemented on today’s and future hardware. In addition to the digital data files, also the rendering software (that is the software which make the information of a digital data file usable for humans) has to be written for the UVC and preserved together with the data. Since the specification of the UVC should not change, also in the future the rendering software will run on a UVC and can be used either to make the data usable for humans or to transform it into a modern data format.

Because of the problems that emulation poses in general, emulation may be a solution only for cases where the computer programs themselves are essential to be preserved. This may be the case for example for games or in cases, where the look and feel of a program is essential and has to be preserved. Emulation could also be used to read undocumented, proprietary data formats that require certain proprietary programs that do not exist anymore on modern computers. However, since emulation only allows to run an old program *within* the simulated environment, there is usually no way to get the information out of the emulation onto the modern machine. That is, it is generally not possible to transfer the data from the emulation to a useful format on the modern computer.

Migration

Migration uses the property of digital data that it can be copied without loss. There are two levels of migration:

1. *Migration of the bit stream*
   In this case, only the data carrier is exchanged by copying the digital data from one generation of storage medium to the next. If the new medium has a different storage capacity, some re-packaging of the data files is required. In general, the copy process is only completed, if a bitwise comparison of the data files showed no errors.
2. *Format migration*
   In this case, not only the data carriers are exchanged, but also the format of the digital data is changed (e.g., from uncompressed TIFF to uncompressed JPEG2000). Such a migration step is quite difficult:
   a. It has to be guaranteed that no data loss occurs with the format conversion (e.g., no lossy compression).
   b. Generating the proof that the copy process succeeded is more difficult. The file in the new format has to be converted back to the old format and then compared with the “original” file. This comparison must be made on a logical level (e.g., comparing pixel values in case of images) and not on a bitwise basis of the resulting files. The reason is that the structure of the files may differ, even if they represent exactly the same content (e.g., two TIFF files may differ on a bit level, but represent identical images with identical information content). This is due to the fact that there are often equally valid variants of common file formats.

With current media, a bit stream migration is necessary about every five years. Format conversions will become necessary only if a format becomes obsolete, that is, if new software does not support the format any longer. However, a format conversion may make sense if there is a new format, which has eminent advantages.

Permanent Medium

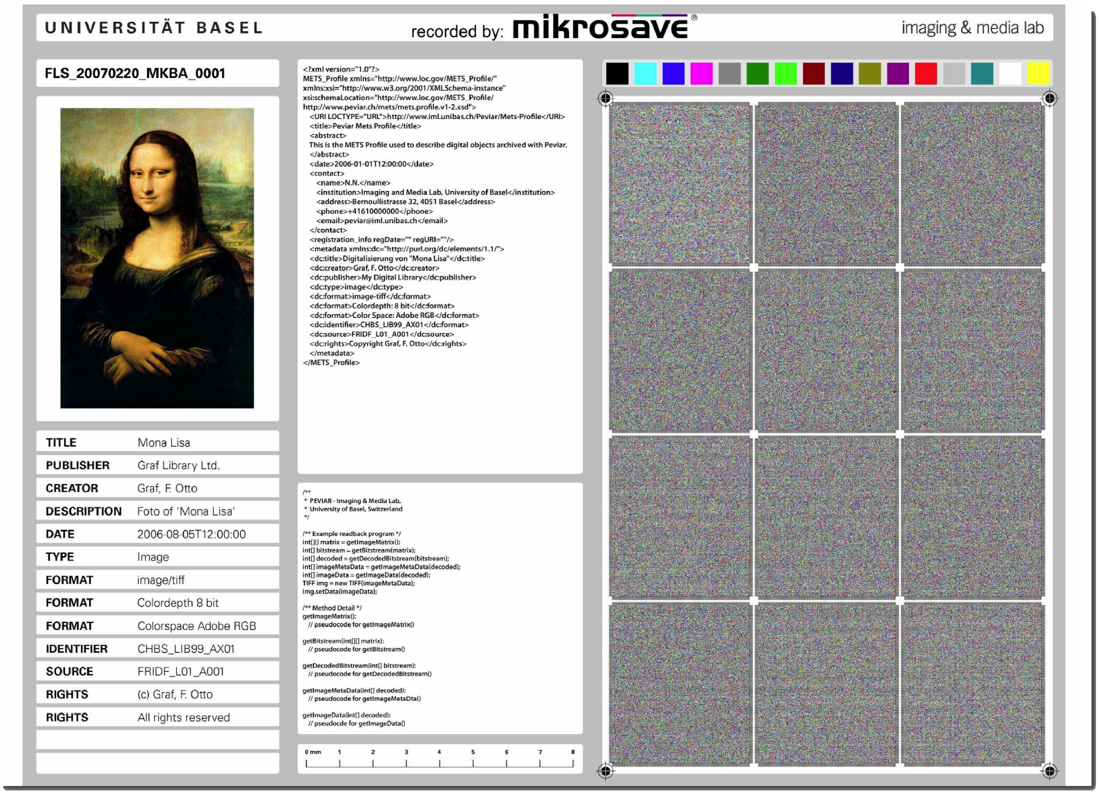
Digital data could be recorded on a permanent medium that has a high intrinsic longevity. However the data must be recorded in a way that is self-explanatory and format independent. In addition, reading back the data must not depend on specialized hardware, which will become obsolete in a short time.

One way, as developed by the Imaging & Media Lab[13] [14][15], to achieve this goal is using visual encoding of digital data on microfilm. The bits can be recorded as bit patterns forming sort of a two-dimensional barcode. In addition, text-based information and analogue images can be recorded on the same support.

[13] Normand, C.; Gschwind, R.; Fornaro, P., *Digital images for eternity: color microfilm as archival medium*, Color Imaging XII: Processing, Hardcopy, and Applications. Edited by Eschbach, Reiner; Marcu, Gabriel G.. Proceedings of the SPIE, Volume 6493, pp. 649307 (2007).

[14] Ariel Amir, Florian Müller, Peter Fornaro, Rudolf Gschwind, Joachim Rosenthal, Lukas Rosenthaler: *Toward a Channel Model for Microfilm*. IS&T's 2008 Archiving Conference Proceedings, Bern, June 2008. IS&T: The Society for Imaging Science and Technology, Springfield (VA), USA.

[15] Müller, F., Fornaro, P., Rosenthaler, L., and Gschwind, R. 2010. PEVIAR: Digital originals. ACM J. Comput. Cult. Herit. 3, 1, Article 2 (June 2010), 12 pages. DOI = 10.1145/1805961.1805963, http://doi.acm.org/10.1145/1805961.1805963



TITLE	Mona Lisa
PUBLISHER	Graf Library Ltd.
CREATOR	Graf, F. Otto
DESCRIPTION	Foto of 'Mona Lisa'
DATE	2006-08-05T12:00:00
TYPE	Image
FORMAT	image/tiff
FORMAT	Colordepth 8 bit
FORMAT	Colorspace Adobe RGB
IDENTIFIER	CHBS_LIB99_AX01
SOURCE	FRIDF_L01_A001
RIGHTS	(c) Graf, F. Otto
RIGHTS	All rights reserved

```
<?xml version="1.0"?>
METS_Profile xmlns="http://www.loc.gov/METS_Profile/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.loc.gov/METS_Profile/
http://www.peviar.ch/mets/mets.profile.v1-2.xsd">
  <URI LOCTYPE="URL">http://www.iml.unibas.ch/Peviar/Mets-Profile</URI>
  <title>Peviar Mets Profile</title>
  <abstract>
  This is the METS Profile used to describe digital objects archived with Peviar.
  </abstract>
  <date>2006-01-01T12:00:00</date>
  <contact>
    <name>N.N.</name>
    <institution>Imaging and Media Lab, University of Basel</institution>
    <address>Bernoullistrasse 32, 4051 Basel</address>
    <phone>+41610000000</phone>
    <email>peviar@iml.unibas.ch</email>
  </contact>
  <registration_info regDate="" regURI=""/>
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Digitalisierung von "Mona Lisa"</dc:title>
  <dc:creator>Graf, F. Otto</dc:creator>
  <dc:publisher>My Digital Library</dc:publisher>
  <dc:type>image</dc:type>
  <dc:format>image-tiff</dc:format>
  <dc:format>Colordepth: 8 bit</dc:format>
  <dc:format>Color Space: Adobe RGB</dc:format>
  <dc:identifier>CHBS_LIB99_AX01</dc:format>
  <dc:source>FRIDF_L01_A001</dc:source>
  <dc:rights>Copyright Graf, F. Otto</dc:rights>
  </metadata>
</METS_Profile>
```

```
/**
 * PEVIAR - Imaging & Media Lab,
 * University of Basel, Switzerland
 */

/** Example readback program */
int[][] matrix = getImageMatrix();
int[] bitstream = getBitstream(matrix);
int[] decoded = getDecodedBitstream(bitstream);
int[] imageMetaData = getImageMetaData(decoded);
int[] imageData = getImageData(decoded);
TIFF img = new TIFF(imageMetaData);
img.setData(imageData);

/** Method Detail */
getImageMatrix():
  // pseudocode for getImageMatrix()

getBitstream(int[][] matrix):
  // pseudocode for getBitstream()

getDecodedBitstream(int[] bitstream):
  // pseudocode for getDecodedBitstream()

getImageMetaData(int[] decoded):
  // pseudocode for getImageMetaDta()

getImageData(int[] decoded):
  // pseudocde for getImageData()
```

Figure 1: Microfilm with an analogue image, text-based information (left side) and binary digital data (right side) recorded on it.

Such a microfilm-based digital recording can be read back without using any special equipment, just any digital camera or scanner with enough resolution will do it. In addition to the analogue images that will help to identify the digital object, the text-based information may contain the instructions on how to decode the bit pattern as well as information about the file format. Such a storage medium is truly independent of any specific technology and will therefore not become unreadable because of technical obsolescence. Microfilm has an expected longevity of more than 500 years.

The OAIS Reference model

The Open Archival Information System (OAIS) reference model for digital long-term archival has been established in 2003 as an ISO-standard (ISO 14721:2003). In fact, the OAIS is a complex *reference model* that tries to identify and define all possible tasks and processes in a digital long-term archive. It is important to note that the OAIS does not represent a real architecture of an archive. It is a *model* that helps to identify and define the components, tasks and processes within a real-world implementation of an archive. Most digital archives do not implement all available processes from the OAIS-model.

In Addition, the OAIS model makes some implicit assumptions about the long-term archive:

1. The archiving method is based on migration. The model does not fit for other methods of long-term archiving.

2. It supposes large institutions with a secured long-term funding that manage the digital long-term archive.

While the OAIS-Model is very helpful to identify processes and gives hints at best practices, it may serve only in very few cases as a blueprint for building a digital long-term archive.

Application to Photography

While most of the statements and reasoning so far can be applied to any digital data, photography, be it digitized from analogue photographs or be it born-digital, represents a special kind of digital data.

Digitization of analogue photographs

During the actual scanning process (digitization), the following parameters have to be chosen carefully and permanently monitored:

1. *Spatial resolution*
   The spatial resolution has to be adapted to the photographic original, i.e. the information content. Important parameters are the spatial resolution of the photographic emulsion, but also the quality of the optical system used to make the original image.

2. *Photometric resolution (gray scale reproduction)*
   The brightness range (contrast) has to be reproduced completely. In the digital domain, the numbers of gray values (or, in case of color images, the number of values per primary color) determine the degree of accuracy. Within 8 Bit, $2^8 = 256$ distinct levels can be represented, with 12 Bit there are 4096 levels and 16 Bit allow for 65536 distinct gray levels. The number of bits required is determined by the contrast of the original. However these values are almost worthless without a physical interpretation: it could be transmission/reflection (a linear scale), optical density (a logarithmic scale), a visual brightness (CIE L*) or uncalibrated "values". Therefore, a proper photometric calibration of the scanner is necessary, and the meaning of the digital gray or color values has to be recorded with the image. The properties of the different photographic material leads to the following recommendation for minimal photometric resolution:

transparent positive (slide)	≥	12	Bit
transparent negative	≥	14	Bit
reflection print, linear scale	≥	10	Bit
reflection print, logarithmic scale	≥	8	Bit

   For practical reasons (computer architecture) the storage of the data has to be done either in 8 Bit or in 16 Bit format (for color images it is 3 x 8 Bit = 24 Bit or 3 x 16 Bit = 48 Bit):

negative	16	bit
slide	16	bit
reflection print	8	bit

For reflection prints, it is possible to reduce the 10-12 Bit internal representation with a logarithmic transformation or a “gamma”-correction to 8 Bit. In this case, the transformation curve has to be recorded. The calibration for gray value images can be done with gray wedges, whereas the calibration for color images requires the IT 8.7 scanner calibration standard that exists both for transmission and reflection.

3. *Hardware calibration (scanner)*
   Scanners have 3 properties that require special attention:
   1. The so-called “dark current” gives a signal even if the sensor is in complete darkness.
   2. Noise introduces small random modifications to the resulting brightness values. Especially in dark areas, noise can effectively destroy fine details.
   3. In addition, the illumination may be inhomogeneous and may vary with time.

These properties that are usually not constant during the lifetime of a scanner have to be taken into account through proper calibration and regular quality monitoring.

Digitization for long term archival requires a strict quality control. On one hand, the calibration of hardware as described above has to be repeated regularly. On the other hand, a *complete* visual control of *each* digitized image in full resolution is required: The following properties have to be monitored through this visual control:

1. *Sharpness*
   Mechanical wear, vibrations, etc. may change the geometry of the optical system of the scanner and introduce systematic blur.
2. *Dust and dirt*
   The scanner (the glass plate) may get dirty through dust and residues from originals.
3. *Geometry*
   Are all images scanned in the correct way or mirrored?
4. *Scan errors*
   Is the image as expected?

This monitoring should permanently accompany the digitization process in order to recognize systematic errors as early as possible.

Another aspect is the completeness and integrity of the digitized collection. Photographic collections valuable enough to be preserved form an ensemble that has to remain complete. For large collections, where the digitization process lasts for a long time, the completeness has to be carefully checked, as e.g. an image may be forgotten in the process or a file name may be used twice etc.

Image file formats

Besides the generic rules for the long-term preservation of digital data (open, well documented format), which also hold for digital image files, the careful selection of the correct file format plays an important role. Fortunately digital images are relatively simple digital objects (compared for example to a relational database). Still, digital image files can be rather complex. Since the digital representation of images usually generates large (a few MB[16]) to huge (200-300MB) digital objects, many image file formats incorporate methods to reduce the foot print of digital images by using compression techniques. Two basically different compression methods have to be distinguished:

- *Lossless compression*
  Lossless compression schemes try to reduce the redundancy that is usually found within digital data. Lossless compression schemes do not depend on specific image properties but can be applied to all kind of digital data. Since images usually contain some redundancy (a pixel value at a certain position is often similar to the neighboring pixel values), a slight reduction of the amount of data can usually be achieved (about 30% to 50%). The reduction will be greater if the image contains large homogeneous areas or areas with repetitive patterns. For images, which contain large areas of irregular patterns (random noise), the reduction will be very small. In some cases, lossless compression may even inflate the size of the image file. A typical lossless compression scheme is the LZW[17]-Algorithm which can be used within the TIFF-format.

- *Lossy compression*
  Lossy compression does eliminate information, which the compression algorithm decides to be of little interest. Therefore, lossy compression algorithms directly depend on the properties of images and particularities of the human visual system. Lossy compression algorithms do therefore *modify* the image in an irreversible way. They usually eliminate only information, which is considered of little importance to the viewer. However, if the compression factor is too high – or more important – the compressed image is manipulated with image processing methods in a later stage (e.g. contrast enhancement), artifacts may become visible. Therefore lossy compression should be only applied to images, which are used *only for viewing*. Images, which are to be archived, or which are to be manipulated in a later stage should *never* be stored in a lossy format! Common compression algorithms (and corresponding file formats) are:

  - ***JPEG***
    JPEG stands for **J**oint **P**hotographers **E**xperts **G**roup and is a well-established lossy compression scheme, which uses the *Discrete Cosine Transform*[18] (DCT) as base for the compression. The image is divided into 8x8 pixel blocks. For each block the DCT is calculated. According to the compression level, only the major coefficients of the DCT are used. The JPEG algorithm may lead to very particular artifacts, which make the block-structure of the algorithm visible. The compression ratio should usually be in the range of 5 to 25. Higher compression ratios will often lead to visible artifacts.
  - ***JPEG2000***
    JPEG2000 is a successor of the JPEG algorithm, but uses a totally different approach. It relies on the *Wavelet-transform, which* builds up a resolution

[16] 1 MB represents 1’000’000 Bytes of data

[17] Jacob Ziv and Abraham Lempel; *Compression of Individual Sequences Via Variable-Rate Coding*, IEEE Transactions on Information Theory, September 1978

[18] The Discrete Cosine Transform is like the Fourier Transform a mathematical method to describe a one- or two-dimensional function in the form of a frequency spectrum.

pyramid of the image. It creates less visible artifacts, but the compressed image may give the impression of less sharpness or crispness. There is also a lossless variant of the JPEG2000 algorithm.

Another very interesting feature of the JPEG2000 algorithm is that not always the whole image file has to be read. If only the first part is read, the whole image can be displayed as if it was compresses at a higher compression rate. For example, in order to display a thumbnail image only about the first 5%-10% of a JPEG2000 (e.g. lossless variant) image has to be read.

Usually, for archival purposes lossy compression schemes cannot be recommended. Lossy compression algorithms *do modify* the image content and always may introduce artifacts and/or reduce the sharpness of the image which will result in a certain loss of details. However, if for certain reasons (e.g. funding, storage space etc.) a lossless compression is not possible, it's still better to apply the rules of long-term archival to the digital images with lossy compression than to do nothing.

The lossless variant of JPEG2000, which will reduce the foot print of a digital image by usually more than 50% may be a good alternative to lossy compression, even if this format is not very as widespread. It is an open, documented, but very complex image file format, which is unfortunately is not so widespread as it should be.

Thus, if no compression is necessary, the TIFF format is a good choice for long-term archival of digital image files. If compression is necessary, JPEG2000 lossless, or a carefully chosen lossy compression scheme of the JPEG2000 variant may be optimal.

Color information and color management

A special issue is the color information of digital images. As stated above, digital cameras usually depend on sensors using three color filters in the red, green and blue band of visible light. In the ideal case, this information should be sufficient to reproduce the *color impression* a colorant produces for the human observer. The limitation to three filters is possible because the human eye has also three different cell types, which are sensitive in the red, green and blue band. In reality however, the filters of electronic sensors differ significantly from the sensitivity curves of the cells in the human eye. Putting away issues like metamerism which would require multispectral imaging with a large number for filters to cope with, the knowledge of the characteristics of the camera or scanner together with the color characteristics of an output device will allow for a mathematical mapping of the color values in order to create an output image that is as close to the original image as technically possible with the specific combination of camera and output device. The process of creating such a mathematical relationship is called *color calibration*. In order to perform a color calibration, both the input device (camera, scanner, etc.) and the output device (printer, screen, beamer) have to be calibrated in order to establish its color characteristics. Usually this is done using specially designed color patches ("e.g. IT8 color chart") and colorimeters. There are several commercial systems available to do this. The resulting color profiles (both for the input device and the output device) then produce together a mathematical transformation that maps the color values from the input device to the output device in such a way that the colors look the same to the human observer. Therefore the input characteristic, that is the *input color profile*, has to be stored with the image data. The International Color Consortium (ICC) established a standard for color profiles. These ICC-Profiles can be embedded in many image file formats as an opaque data element (e.g. TIFF, JPEG, PNG, EPS, PDF, SVG etc. allow the embedding of color profiles). It is to note that the image file itself does not “know” anything about the color profile. It just stores it as a chunk of data and transfers it to an image-rendering program (e.g. for display on a screen or for printing) which then interprets the color profile data to render the colors properly.

Thus, each digital color image should be accompanied by a color profile that is specific for each input device. However it is often more convenient to transform the color information to one of the widespread standard color profiles such as the sRGB, AdobeRGB or ProPhoto profiles. It is to note that during such a transformation some information may be permanently be lost, since the amount of colors (called the gamut) that can be represented by a specific color profile may be more limited than the gamut of the original color profile. Yet, using a standard color profile has the great advantage that the color profile data can be omitted from the image file as long as the information is included to which standard color profile the image data conforms to.

Conclusion

At the moment, for most cases the longevity of digital data can be best achieved by implementing a migration model based on the following rules:

1. *Redundancy*
   Data have to be kept with a high level of redundancy. At least 3 copies on a minimum of 2 different types of storage media (e.g., two copies on hard disk, one copy on magnetic tape) should be kept at geographically different locations.
2. *Checksums*
   For all data files, checksums should be calculated and archived with the data files. This allows at any time to check if a data files has changed or contains errors due to aging.
3. *Proofreading*
   Every 12-24 months, the data should be proofread and the checksums compared. If errors are detected, a migration should be launched immediately.
4. *Migration*
   Migrations have to be planned in advance including financing. A bitstream migration is necessary about every 5 years. A format migration is advised if a new file format becomes standard and the conversion can be done without loss of data.
5. *Documentation*
   Every step has to be documented in detail, all media have to be labeled properly.

Following these rules, digital data may be preserved indefinitely. However, a constant care is required. If this care is not possible for a certain length of time, the data will be lost — and only digital archaeology may possibly recover part of the data.

An alternative may be the use of a permanent visual medium such as microfilm to record analogue, text-based and digital binary data. Such a data carrier is independent of any specific technology and can be read back in future times using simple image capturing devices. The interpretation of the bits can be aided by text-based information containing the instructions.

While the initial cost of a storing digital data on microfilm may be higher, it may be less expensive than the migration method on the long-term. In addition, it does in contrast to the migration mode not require a continuous flow of funding.