Journal of
Electronic Imaging

JElectronicImaging.org

JPEG-1 standard 25 years: past, present, and future reasons for a success

Graham Hudson
Alain Léger
Birger Niss
István Sebestyén
Jørgen Vaaben

SPIE. IS&T imaging.org

Graham Hudson, Alain Léger, Birger Niss, István Sebestyén, Jørgen Vaaben, “JPEG-1 standard 25 years: past, present, and future reasons for a success,” *J. Electron. Imaging* **27**(4), 040901 (2018), doi: 10.1117/1.JEI.27.4.040901.

REVIEW

JPEG-1 standard 25 years: past, present, and future reasons for a success

Graham Hudson,[a] **Alain Léger,**[b,*] **Birger Niss,**[c,d] **István Sebestyén,**[e] **and Jørgen Vaaben**[c,f]
[a]British Telecom/BT Labs, Martlesham Heath, England, United Kingdom
[b]France Telecom R&D/CCETT, Cesson-Sévigné, France
[c]Copenhagen Telephone/Kjøbenhavns Telefon Aktie Selskab—Copenhagen Telephone Company, Copenhagen, Denmark
[d]NNIT A/S, Søborg, Copenhagen, Denmark
[e]Secretary General Ecma International, Geneva, Switzerland
[f]The Easy Copy Company, Farum, Denmark

Abstract. Digital image capture, processing, storage, transmission, and display are now taken for granted as part of the technology of modern everyday life. Digital image compression is one of the enabling technologies of the present multimedia world. The image compression technique used for application as diverse as photography, web pages, medical imaging, and public records is JPEG, named after the ISO/CCITT "joint photographic experts group," established in 1986, which developed the technique in the late 1980s and produced the international standard in the early '90s. ITU-T T.81¦ISO/IEC 10918-1, also called "JPEG-1" has become one of the most successful standards in information and communication technologies (ICT) history. The authors of this paper—all members of the original JPEG development team—were all intimately involved in image-coding research and JPEG in particular. The paper goes behind the scenes explaining why and how JPEG came about and looks under the bonnet of the technique explaining the different components that give the standard of its efficiency, versatility, and robustness that have made a technique that has stood the test of time and evolved to cover applications beyond its original scope. In addition, the authors give a short outlook of the main milestones in coding schemes of still images since "JPEG-1." *© 2018 SPIE and IS&T* [DOI: 10.1117/1.JEI.27.4.040901]

Keywords: image compression; standards; joint photographic experts group; history; photographic.

Paper 180443V received May 17, 2018; accepted for publication Jul. 30, 2018; published online Aug. 31, 2018.

1 Introduction

The ISO/IEC-ITU JPEG image compression standard is celebrating its 25th anniversary. It is remarkable, even for those originally involved, that in a world where information processing technology, telecommunication services, and web applications are moving so fast, this successful compression algorithm has shown such resilience. The JPEG team laid down a framework, selection processes, and key features for future extensions of JPEG in this fast-changing technological environment [JPEG2000, high efficiency video coding (HEVC)].

In this paper, key historical contributors give an insight to the drivers of the international team from many major research groups in image coding during the 1980s toward the worldwide ISO/IEC-ITU JPEG standard [ITU-T T.81 (1992)¦ISO/IEC IS 10918 (1993)] used in everyday life today.

A large part of the paper focuses on the key decisions made during the development of JPEG in the 1980s. At that time, more than 25 years ago, the context was drastically different from that known today. Capturing, storing, editing, sending, and exchanging images or simply playing with image content like any textual content was technically very challenging.

In those days, capturing a digital image to the CCIR 601 (ITU-R 601, February 1982)[1] digital studio resolution (720 pixels × 576 lines, square pixel, 4:3 aspect ratio) required specialized computer equipment (a digital TV frame-store and mini-computer with multiple hard disks). Developing efficient image processing techniques to compress–decompress with the highest possible compression factor while preserving the image quality (also known as "rate-distortion theory") was achieved by programming in the Fortran and C languages (C++ was used in late '80s) in a rudimentary software development environment that required overnight processing. The results were evaluated by people in a TV studio environment using analogue studio quality CRT monitors (no LCD displays!).

The very large body of the scientific literature available through conference proceedings described various attempts to compress images since the '50s, spatial differential pulse code modulation (DPCM) in the '60s, and transform coding techniques in the '70s, but no leading research directions were obvious in the early '80s for a compression scheme to be standardized. The international standards team of over 50 dedicated scientists/engineers took over 5 years to evaluate development and reach an agreement on a technique (1988) and eventually approved and published the ISO/IEC-ITU standard (1993/1992).

The paper will outline the requirements that were established for the compression technique and the procedure adopted for the evaluation of the contenders. The key decisions the group made during the building process of the JPEG compression scheme and format will be explained. Techniques evaluated included transform [discrete cosine transform (DCT), Karhunen–Loeve and later wavelets], psychovisual thresholding for quantization, mean value of

*Address all correspondence to: Alain Léger, E-mail: a-t.leger@wanadoo.fr

a block of pixels (DC) prediction, two-dimensional (2-D) source modeling, and Huffman entropy coder versus arithmetic entropy coder.

The topic of patents, reasonable and nondiscriminatory (RAND) and royalty-free (RF) licenses will be discussed as will the early implementation of JPEG by one of the first open source development groups (independent JPEG group). The lessons learned for maintaining an accurate standards archive of all the technical papers produced will be outlined. Lastly, the strategic importance will be stressed of the European R&D ESPRIT framework [Conférence Européenne des Postes et Télécommunications (CEPT), photographic image coding algorithm (PICA) project, JPEG early days] that had a leading role in the international standardization process in ISO and ITU.

Many functions and features—some of which are now being considered for improving the compression scheme—were in fact already considered/investigated during the development of JPEG: Integer DCT that would allow for subsequent build-up to lossless, 16 × 16 DCT, alternative to zigzag scanning through the AC coefficients, AC prediction, context-dependent quantization, and addition of intelligent noise to reduce blocking.

The initial applications that make large use of the JPEG compression scheme will be reviewed.

Lastly, a brief historical outline and a table of milestones of the evolution of image compression standard specifications will be given.

2 Original JPEG Requirements, Technological Context, and the Selection Process

2.1 *Original (1986) JPEG Requirements for Compressed Image*

As early as 1976 at the Kodak corporate research center in Rochester, New York, a single prototype digital camera was built. No one at Kodak at that time was attempting to build a camera and playback system like this. It was just for demonstration that proved that digital photography was not yet a competitor to analog photography (Fig. 1).[2]

The sensor (Fairchild CCD 201) had a resolution of 100 × 100 black/white "pixels." It was an interline architecture, which means that this resolution was obtained through two interlaced "fields" that were read out in succession. The 2-Mpixels resolution goal came from some estimates of the image quality equivalent for a frame of 110-format consumer film (the lowest quality consumer format at the time).

Kodak chose 30 images per magnetic cassette tape to make the system look a bit like the analogue film systems (24- or 36-exposure film rolls). The storage of a single 100 × 100 b/w image took 27 s. Scaling up for the imaginary 2-Mpixels resolution, the storage time on the magnetic cassette would have taken a very, very long time. Last but not least they did not use any digital image compression method, such as JPEG. This was certainly not needed for the storage of the 100 × 100 black/white images, but for the long-term 2-Mpixels color images, should have been required.

It was thought that digital photography was probably decades away. Unfortunately for Kodak, thanks to the development of several key components, such as JPEG, the industry hit the 2-Mpixels image point for cameras around 1999 (Kodak DC 280), which was around the time that the sale of film cameras peaked along with film sales. Today, analog photography is almost extinct. Unfortunately, as a result, Kodak went out of business. Nevertheless, digital photography has become a key market for JPEG. The emergence of compact digital cameras and especially mobile phones with photo-camera features gave great impetus to the development of the digital photography market [see Figs. 5(a) and 5(b)].

For other applications, such as digital still-picture TV images, engineers saw the practical possibilities of digital image coding and processing in the early '80s. In particular, telecom companies were looking to include graphics and photographic information as an enhancement into their embryonic videotex systems (which can be regarded as the forerunner of the world wide web). The challenge they faced was to compress the large amount of data in a photographic image to enable it to be transmitted over a telephone line in a few seconds, and then decompressed and displayed on a personal computer or modified television, also in a few seconds. In fact, at that time, the main scope was to offer "real-time" multimedia (text + photos) videotex services over a telephone pair.

2.1.1 *Main scope*

A full-frame digital picture has 720 × 576 pixels as to CCIR 601 (ITU-R 601, February 1982),[1] requires 828 kbytes of storage and takes over 104 s to transmit over an ISDN-64 kbit/s telephone network. The original goal was then to produce a technique capable of a compression of 16:1 or 1 bit/pixel (bbp), to get a full screen image at the user terminal in 6 s.

In 1979, BT Labs vision for enhancing the text/graphic Prestel (videotex) display was to introduce photographic images.[3] A demonstration of "picture Prestel" was given at the Viewdata'80 conference in London, using a Prestel television modified with a one-sixth display area picture frame store [Fig. 2(b)].[4] Two compression techniques were evaluated by BT–DPCM/entropy coding providing a sequential picture buildup and Walsh–Hadamard transform to give a progressive buildup. Using a standard Prestel database with a 4.8 kbit/s modem over the analogue PSTN,[5] picture download took about 15 s. In 1983 at Telecom'83 in Geneva, BT demonstrated full-frame photovideotex transmitted at 64 kbit/s over the ISDN.[6]

In the early '70s, Centre for the Study of Television broadcasting and Telecommunication (CCETT) started research on digital image representation and different compression techniques (MICD, discrete transforms and others) initially for digital television for contributions to Rec. 601 at CCIR (ITU-R).[1]

In the late '70s, CCETT launched the Minitel (videotex) and in 1981 demonstrations of DCT compression for photovideotex were shown at different conferences (NAB Las Vegas'81, NCTA'81 Los Angeles, CER'81 Montreux, Videotex'81 Toronto, and Viewdata'81 London) running at 38.4 kbit/s [Fig. 2(a)]. In 1991, a new version of the Minitel included the presentation of photographic images (320 × 240, 64 gray levels) compressed in JPEG, and running on PSTN 4.8 kbit/s was shown at Telecom'91 in Geneva.[7] The services envisioned at that time were e-commerce catalog, enterprise internal directory, telesurveillance, and tourism promotion services.

An early example (1982) of photographic videotex over the PSTN (1200/75 bps) was the Austrian Mupid

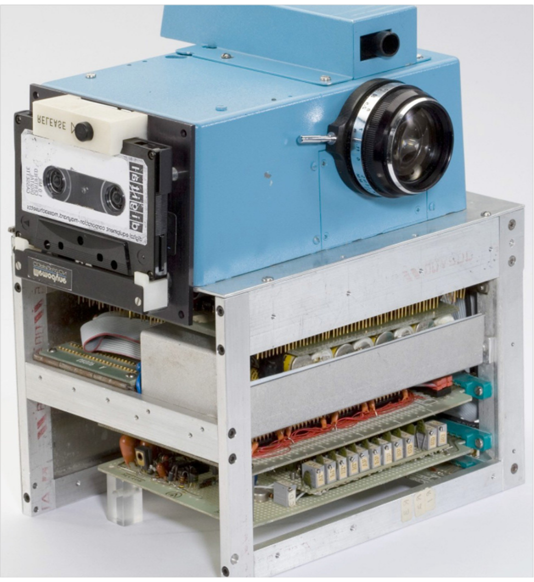


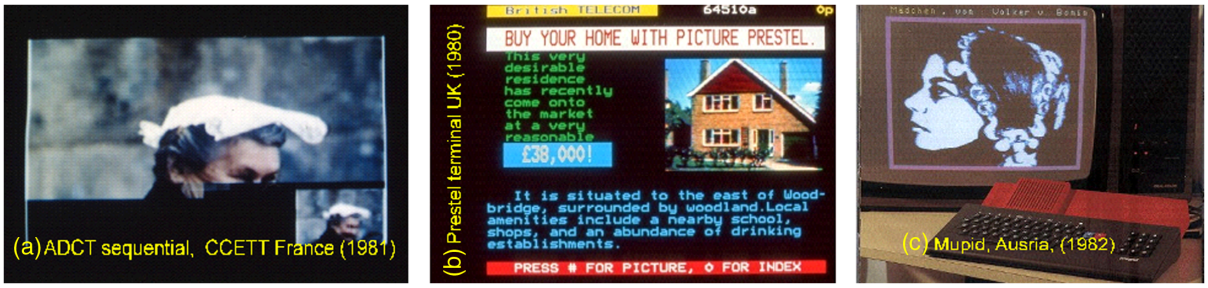
Fig. 1 The experimental Kodak digital camera of Steven Sasson (1976).[2]

microcomputer terminal, which incorporated its proprietary tele-software (TU Graz, Austria, Prof. H. Maurer) and took 4 min to download and display the low quality (low resolution, 16 shades of gray) [Fig. 2(c)].

2.1.2 *Test pictures*

A set of reference images was produced by the independent broadcasting authority at 720 × 576 pixels according to CCIR-601 (Fig. 3).

At that time very few scanners were available for digitizing images, and the SMPTE–CCIR Rec. 601(1982) was the only standardized color image format available. The JPEG group chose to use RGB conversion to Y–Cr–Cb to decorrelate color components and to use subsampling in the chroma signals to further compress the image. However, color space conversion and subsampling is not required in the JPEG standard, only a recommended possibility for some image types. The standard is a framework for any image representation system (up to 255 components). It allows the inclusion of color spaces, such as RGB, YCrCb, CMYK, CMYK+ plus spot colors, and seven-channel remote sensing. A specific, initially small set of images


(a) ADCT sequential, CCETT France (1981)
(b) Prestel terminal UK (1980)
British TELECOM
64510a
0p
(c) Mupid, Ausria, (1982)
Madchen , von Volker v. Bonin

BUY YOUR HOME WITH PICTURE PRESTEL.

This very desirable residence has recently come onto the market at a very reasonable £38,000!

It is situated to the east of Woodbridge, surrounded by woodland.Local amenities include a nearby school, shops, and an abundance of drinking establishments.

PRESS # FOR PICTURE, 0 FOR INDEX

Fig. 2 Photovideotex experiments in Europe: (a) photovideotex terminal in France (1981) (source: CCETT; used with permission), (b) photovideotex page on the Prestel terminal in the UK (1980) (sourced British Telecom; used with permission), and (c) Austrian Mupid terminal with an early videotex photographic image (1982).

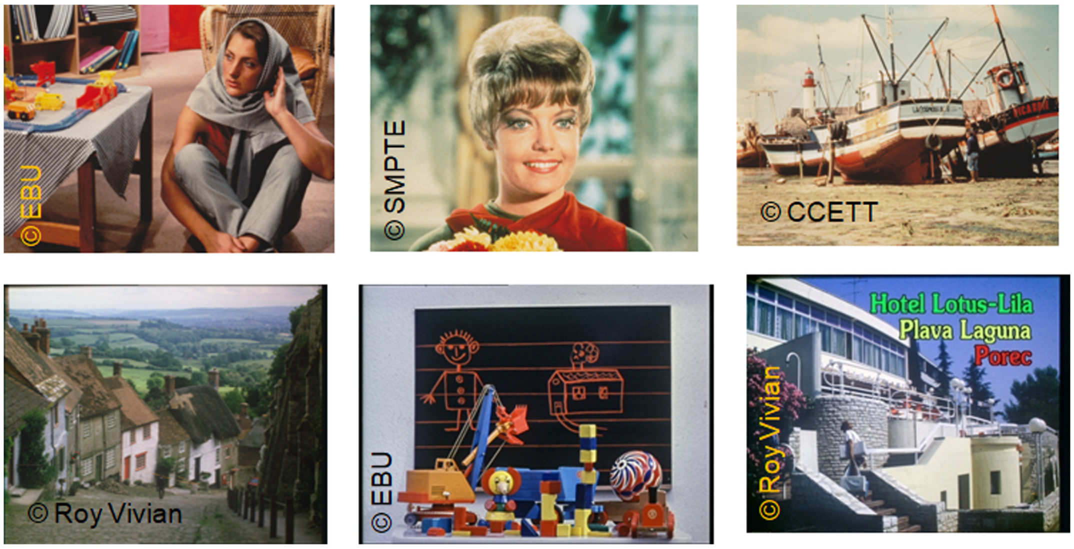


Fig. 3 Some JPEG test pictures.

was used for evaluating compression methods. The set was later amended, but still with the same TV resolution of 720×576 pixels. Even if the initial intention was to compress *a priori* any kind of images, the training set mainly comprised real-life content. Later in 1994, the ITU-T T.24 digitized image set that contains high-resolution images (for high quality printing) was compressed well by JPEG.

2.1.3 *Progressive versus sequential*

Coding the image "from one end to the other" was not good enough. The team had to find a method of sending the data in such a way that a recognizable image was produced pretty early in the data stream. This requirement came originally from CCITT in 1985 (Fig. 4). At that time, Facsimile Gr3 devices were already extremely successful worldwide. Nevertheless, they were paper oriented and not "real time." However, real-time exchange of soft-copy facsimile pictures (first at lower resolution appearing quickly on the display, with the final image at high quality that could be printed such as a normal fax), or real-time access to facsimile databases was in growing demand. Although today facsimile soft-copy communication and databases do not exist, nevertheless in today's web environment these types of applications are widespread. Nevertheless, historically the requirement for progressive/sequential image buildup came from b/w facsimile extended to multilevel and full-color images.

2.1.4 *Scalability*

Several applications have demanded from the very beginning some sort of scalability.

- Scalability in image quality: from low image quality, to better image quality, to invisibly loss-less image quality, to "real" loss-less image quality (required for some medical applications). This is also used as a means for controlling the storage requirement for images.
- Scalability in image resolution: from low-resolution images by transferring more information to higher image resolution to the final image resolution that equals with the source image. This feature is often required in image databases (first image search on image icons and by selection one gets with the remaining information the full image), and in the printing industry (e.g., what started in the 1980s as "desktop publishing").
- Scalability in image build-up: (as mentioned already in Fig. 4). Transmission of a low quality image first for

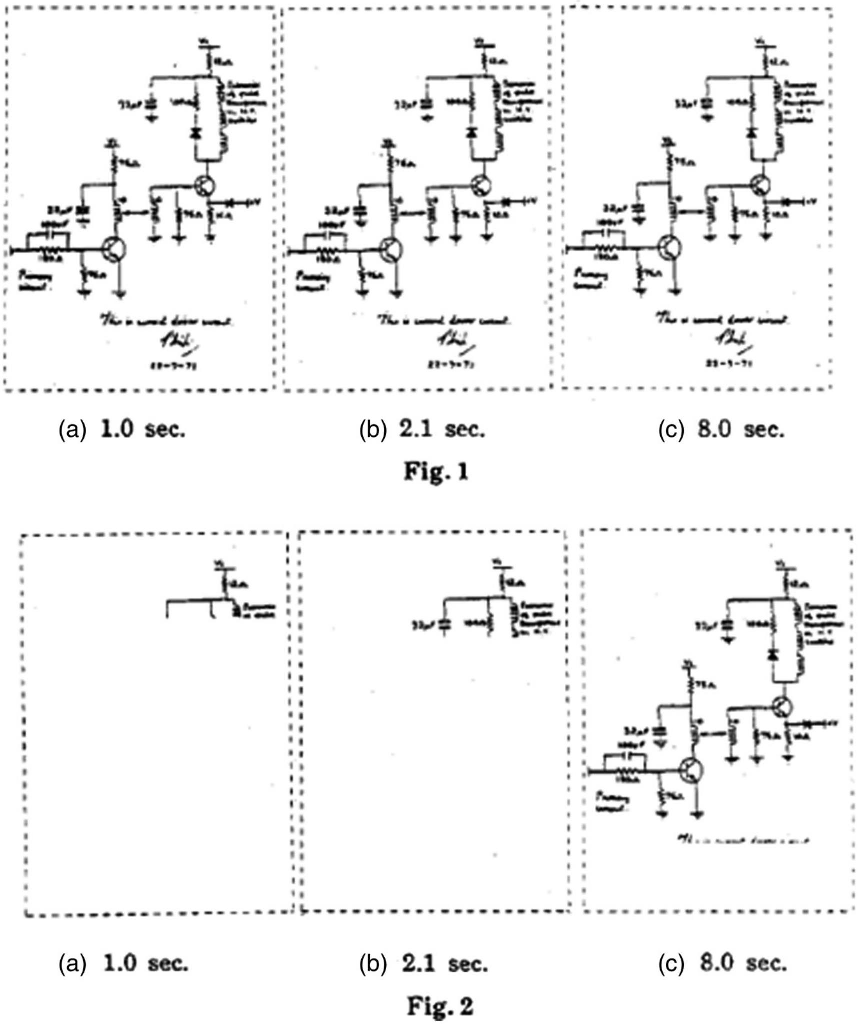

(a) 1.0 sec.
(b) 2.1 sec.
(c) 8.0 sec.
Fig. 1
(a) 1.0 sec.
(b) 2.1 sec.
(c) 8.0 sec.
Fig. 2

Fig. 4 "White document" 1 from KDD—1985 Kyoto CCITT SGVIII meeting.

instant display that gets better and better as the transmission process goes on until the final image quality is reached.

There were also some other requirements that are not listed here. However, the conclusions from these different requirements are very important.

- To satisfy the broad variety of applications with the possibility of achieving a good level of similarity and easy interconnection, it was decided to adopt for JPEG the "tool box" principle (similar to the well-known LEGO building bricks). In other words, the JPEG standard describes a framework of a compatible family of image compression techniques, rather than a single compression technique. Applications can select elements of the "tool box" that fits their own requirements (baseline, profiles, and many possible arrangements of technique elements).
- In practice, it turned out that the "tool box" defined by the early set of telecommunication applications proved to be good and flexible enough also for applications that they were originally not designed for or even foreseen (e.g., for the aforementioned digital photography with components such as the JPEG compression digital photography became the winner or the JPEG images on mobile phones that did not exist at that time).
- JPEG made the strategic decision to declare the base JPEG components (the so-called "JPEG baseline") as "royalty-free." This allowed easier implementation and rapid adoption of the standard. Also the "toolbox" principle is particularly suitable for "open source implementations."

Some of the original CCITT requirements (CCITT Recommendation T.80-1992[8]) could not be satisfied by the JPEG algorithm. For example, one requirement calling for a "universal coding scheme" suitable for all color spaces (from black and white up to continuous tone color space) could not be satisfied by the original JPEG, only later by the JBIG2 and JPEG2000 standards.

The requirements for a standardized JPEG file format varied from different organizations. CCITT (ITU) required JPEG to be included in CCITT/ITU applications and services (such as videotex or color facsimile), but ISO/IEC had not recognized at the beginning the need for a standardized file format. This was only done years later (too late) in "JPEG Part 3."[9,10] In the meantime this gap was filled by Eric Hamilton's *de-facto* standard JPEG file interchange format (JFIF) specification, which became ratified by an standard defining organization (SDO) some 20 years later.[11]

2.2 *Later Requirements for Compression (10 to 100 Mpixels)*

In the late '80s, it was estimated that imaging would match the visual system by the early '90s.

We are still not quite there yet, but we are close: the resolution of the eye at the foveal point for a person with an extremely good visual acuity (20/10) is ∼100 μrad.[12] If we assume that the field of vision is about one steradian, we would require about 100 Mpixels in an image. The foveal area only covers about one degree of arc, and the resolution outside that area is much less. However, we never know where a person will be focusing so we need the larger area with high resolution. The dynamic range of the eye (the ability to see detail in deep shadows and highlights) is higher, than we can represent in eight bits, but there are attempts to address that especially in the new image formats and coding schemes. For temporal resolution, some people may detect flicker (in their peripheral vision) up to near 100 frames per second (fps).

The standard full HDTV resolution is 1920×1080 p (∼2 Mpixels), 4K TV has double the resolution in each dimension (∼8 Mpixels), and 8K double again (over 32 Mpixels). For high-end still cameras even higher resolutions are not uncommon. As to the dynamic range, there are strong standardization efforts to define high dynamic range (HDR) formats. The flicker issue is gradually being addressed in different ways: traditional cinemas run at 24 fps, but they may make multiple illuminations of each frame—it does work! For newer formats, the frame rate will typically be 60 fps or higher. Advanced PC gamers may require frame rates way above 100 fps. Some of the above numbers may have to be doubled for 3-D viewing with the currently used technologies.

There is work in progress to specify 8K at up to 120 fps and a depth of 12 bits per sample (super MHL, ITU-R Rec BT.2100).

The largest screens in practical use are the domes in IMAX theaters—and they are up to ∼6 steradians. It is speculated that 8K and even higher will become the video format of choice for TV sets; 4 K TV-sets have been in domestic use since 2017, and there are publicly available 8K video cameras. It is said that to obtain the full benefit of an 80-in. 4 K TV-set, the viewer needs to sit a few inches away from the screen and the corners cannot be seen. User interfaces that utilize images containing larger resolutions, higher dynamic ranges (HDR) (12 bits per components or higher), wider color gamuts (larger space coverage on, e.g., CIE Lab or CIE Luv), further contribute to larger volumes of data in higher bandwidth environments (Table 1).

How will JPEG survive all this? Probably in some application areas it will not, but it has shown a remarkable resilience. It was not built for the high resolutions of today's technology, and the 8×8 DCT does not decorrelate the many blocks as well as it could have. However, the quantization matrix that sits in every image may be tweaked and that may alleviate some of the problem. A 16×16 matrix would probably have been preferable for higher resolutions, but in the late '80s the computational resources did not exist. The discrete wavelet transform (DWT) is in principle open-ended for image size, it compresses a little better than JPEG, but the computation required was prohibitive at that time.

But for "the low end—consumer—quality range" JPEG popularity will remain in many areas, such as everyday photography: "In the decade from year 2000 to 2010 we witnessed the golden age of photography. In it the global user base of cameras grew tenfold [Figs. 5(a) and 5(b)]. The number of pictures taken grew so dramatically, most pictures ever taken have been taken within the past 2 years. Yet in the golden age of photography, all of the past giants of the camera industry struggled or even died. The market opportunity was taken by mobile phone makers, Nokia, Apple, Samsung,

Table 1 Image type evolution (2018).

Image type	Size	Bits/pixel	Uncompressed size
Text	A4	16 b/character	4 to 8 KB
Facsimile two-tone G4 (200 dpi)	1653 × 2338 (A4)	2 bits/pixel	4 Mbit
Photovideotex (1988)	768 × 576	16 bits/pixel	884 KB
Gray scale	512 × 512	8 bits/pixel	262 KB
TV CCIR 601 (1982) (4.2.2)	768 × 576	16 bits/pixel	884 KB
HDTV	1280 × 720	24 bits/pixel	2.8 MB
Full HD	1920 × 1080	24 bits/pixel	6.2 MB
HD stereovision 3-D multiview[13]	1924 × 1080 × nViews	24 bits/pixel	N × 6.2 MB
Ultra HD TV 4K	3840 × 2160	24 bits to 30 bits/pixel	32 MB
Camera full frame (24.3 Mpixels)	4024 × 6036	24 bits/pixel	72 MB
Ultra HD TV 8K	7680 × 4320	24 bits to 36bits/pixel	150 MB
HDR[14]	Full HD—ultra HD	36 bits/pixel	150 MB
DCI HD cinema	2048 × 1080	30 bits to 36 bits/pixel	10 MB
360 camera	Up to 12,000 × 6000	24 bits/pixel	50 MB to 2 GB
Holography	Giga pixels	24 bits to 36 bits/pixel	Giga bytes
Plenoptic imaging[15]	Giga pixels	Pixel and other dimensions	Giga bytes

LG, etc. none of which even made one camera at the start of the decade."[16]

"Stand-alone digital cameras were introduced in 1990 and after a long struggle, took over from film-based cameras. According to the CIPA (Camera Imaging Products Association), by 2006 film-based cameras formed only 4% of the total world camera shipments (excluding disposable cameras). IDC said that in 2008 stand-alone digital camera sales reached 111 million units.

The first camera-phones came from Japan in 2001. Soon Nokia too started to install cameras to phones. The world has shipped 3.8 billion camera-phones in the past 9 years and 2.5 Billion of those camera-phones are in use today or 65% of all phones in use. So, Nokia's installed base of camera-phones in use is about 1 billion."

"Now compare, even counting all film based cameras, and all digital cameras, ever made: Nokia branded camera-phones in use exceed all non-phone cameras ever made, counted together."

Four years later, the 2014 data show a similar picture:

"World new sales total cameras 2014: 1.8 billion cameras for consumers (excluding webcams and security cams, etc). 95% of those are camera phones on mobile phones, 5% are traditional stand-alone cameras.

The global installed base of cameras still operational is 5.8 billion units. Out of those only 4 billion are in use (as cameras, most that 'are not used' are on mobile phones/smartphones which are used in other ways but not for their camera). Of the 4 billion cameras in use, 440 million (11%) are stand-alone "traditional" digital cameras and 3.56 billion (89%) of all cameras in use on the planet today are on mobile phones/smartphones as camera phones. Beyond those there are another 1.2 billion camera phones not used because their user has a better camera on his/her other smartphone/camera phone, and 180 million older digital cameras sit in our homes forgotten and forlorn.

The average stand-alone digital camera user took 375 pictures in 2014 while the average cameraphone user snaps 259 pictures this year. When multiplied across the total user bases, that produces 1 trillion (1,000 billion) photographs taken this year by digital camera owners. That brings humankind's cumulative picture production total to 5.7 trillion photographs taken since the first camera was invented." (All stats are from TomiAhonen Phone Book 2014.)[17]

What is not told in the statistics, but what we think is obvious, is that close to 100% of the pictures taken by digital cameras have been using JPEG-1. The rest is marginal. As a result, the many trillions of JPEG images created and yet to be created stay here (except those that will not survive long-term storage—another open issue), and from the practical point of view it is hard to imagine that those JPEG-1 images could be mass-converted to a new "post-JPEG" format even if that new method was superior. Consequently JPEG-1—which apparently today fully satisfies average

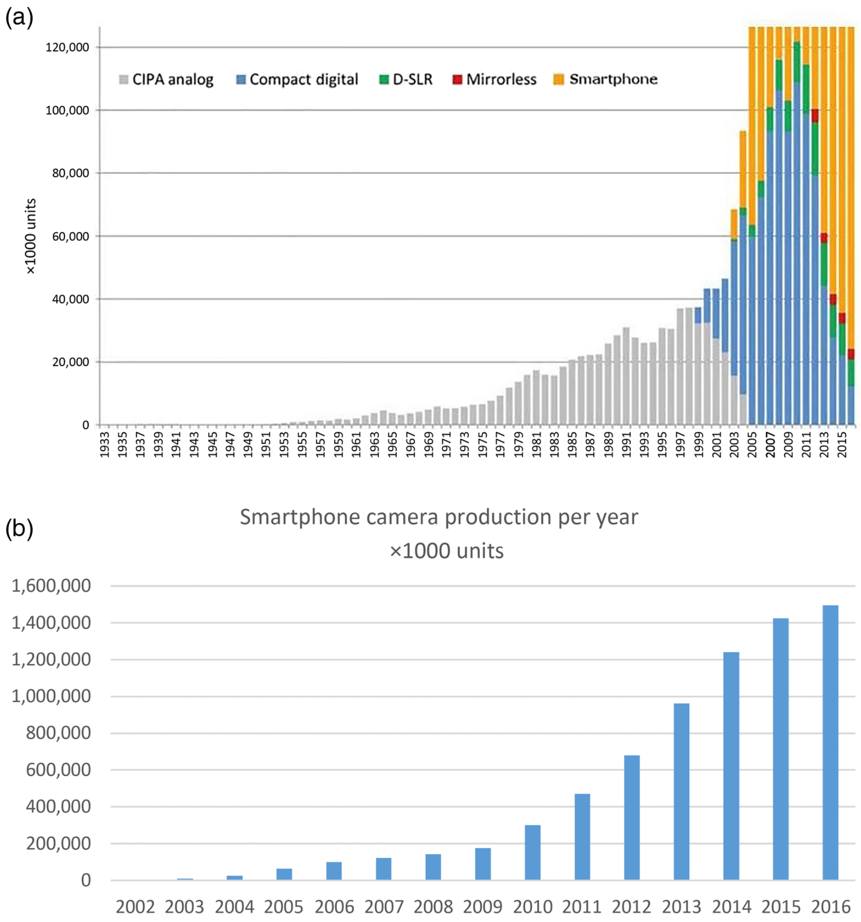


Fig. 5 (a) Share of photo-camera-types between 1933 and 2016 (TomiAhonen Phone Book 2016, Copyright TomiAhonen Consulting 2016) and (b) smartphone camera production per year. (CIPA analog, compact digital, D-SLR, Mirrorless source: CIPA and Mayflower Concepts. Smartphone sales source: Gartner Inc. Does not include PDAs or feature phones.)

user demand—is expected to stay here at least for a few more decades. This also means that a potential successor of JPEG-1 in the consumer area must be either backwards compatible to JPEG-1 or it has to implement two parallel compression and coding methods including JPEG-1.

2.3 *Original (1986) JPEG Requirements for Further CCITT Applications and Services*

The basic idea in ITU was that the CCITT T.80 Series of Recommendations[8] should provide the building blocks for various different CCITT applications and services. Due to the use of the common T.80-series components, the aim was to provide an easy interworking between some of the applications by the use of common picture-coding components. Following the approval of the JPEG-1 standard (CCITT T.81) in 1992 other ITU-T applications incorporated JPEG for image coding. Examples are given below.

- Videotex photographic mode (ITU-T T.101 Annex F).[18]

  ITU-T T.101Recommendation defines the rules applicable to the international interworking between videotex services (Data Syntaxes I-III). In addition, common extensions to the various data syntaxes are defined, including Photographic Data Syntax (Annex F)—using JPEG.

- Data protocols for multimedia conferencing (ITU-T T.120 Series).[19]

  ITU-T recommendation T.126 defines a protocol supporting the management of common multilayer visual spaces and the multipoint exchange of graphical information directed to these spaces including images (hard and soft copy), pointers, and filled and unfilled parametric drawing elements (points, lines, polygons, and ellipses). Support for rendering out-of-band video streams within T.126 workspaces is also included. This protocol uses services provided by ITU-T recommendations T.122 and T.124 and complies with the guidelines specified in ITU-T recommendation T.121.

- TU-T T.417 information technology—open document architecture and interchange format: raster graphics content architectures.[20]

  This recommendation is one of the recommendations of the T.410-Series. It defines the raster graphics content architecture used to include images (including JPEG) in a document. Image-coding methods used in the raster graphics content architecture include methods used in the facsimile environments as well as methods used in nonfacsimile environments.

- ITU-T T.30 procedures for document facsimile transmission in the general switched telephone network.[21]

This recommendation defines the procedures used by group 3 facsimile terminals as defined in ITU-T Rec. T.4. These procedures enable documents to be transmitted on the general switched telephone network, international leased circuits, and the integrated services digital network (ISDN). It also defines the color facsimile group 3 standard using JPEG.

- ITU-T T.563 terminal characteristics for group 4 facsimile apparatus.[22]

This recommendation defines the terminal characteristics for group 4 facsimile apparatus. The descriptions of the terminal characteristics for color extension are added as an option by this recommendation (JPEG in facsimile group 4). The coding schemes for color image type and optional functions for color facsimile are mainly defined.

In conclusion, which of the aforementioned have survived until today? Not many.

Maybe to some extent color facsimile group 3 and certainly the concept of ITU-T T.101 Annex F (Videotex photographic mode) can recognize how JPEG in used in web browsers.

The modular structure of JPEG enabled the aforementioned applications and numerous diverse and unforeseen further applications, making JPEG not only robust, but also multifunctional.

3 Key Technical Choices (1992)

At the first meeting (CCETT-ANT, 1985), held at ANT Backnang, Germany, the general coding pattern was set and agreed. It was not obvious at that time that it clearly separated the compression scheme into three boxes: (1) image signal transformation (no compression), (2) visual redundancies elimination (compression), and (3) statistical redundancies elimination (compression) (Fig. 6).

This choice proved to be very robust during the definition of the coding scheme to be approved in January 1988, and remains today totally valid for this very large class of image/sound/video intracompression schemes. If more AI-based compression schemes evolve, this general pattern may no longer be right.

3.1 *Transform Image Data (DCT, Karhunen–Loeve, and Wavelets)*

The "raison d'être" of the transformation in so-called "transform coding" is going from one vector space base, representing—say—an 8×8 image block, where all base vectors are equally important, to another base spanning the same vector space, where the base vectors have very different importance. The aim is to find a transform where all the important information—the energy—in the image is represented by very few base vectors. The optimum transform is the Karhunen–Loeve transform (KLT). The KLT analyzes the image and extracts the principal components, thus compacting the energy very efficiently. It is, however, highly computational intensive, far more than realistically available in the late '80s—and most of all the calculated transformation kernels depend on the image content, so it should be calculated for each image.

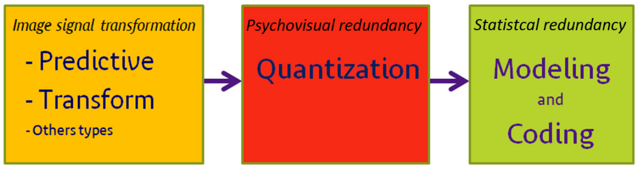


Fig. 6 General coding pattern (1985).

Various other much simpler transforms were examined during the development of JPEG: high- and low-correlation transforms, where all operations could be done using only shifts and adds, and the DCT, which could be calculated using lookups tables. It quickly turned out that the DCT was by far the best of the second best with an energy compaction approaching the KLT. It was, therefore, decided to continue with the DCT as the transform of choice.[23,24]

Now, the transform block size came into play: Why does JPEG choose 8×8 blocks?

From an energy compaction point of view, the optimum block size should be one where the pixels in an average block are correlated. Using too small, a block size misses important pixel-to-pixel correlation. Using too large, a block size tries to take advantage of a correlation that might not exist.

Working with the typical image sizes of the late 1980s (720×575 pixels), 4×4 blocks were too small to catch important correlations, and 16×16 blocks often contained uncorrelated pixels and increased calculation complexity for no gain. So out came the 8×8 block!

Today, with 4K and 8K and higher display resolutions, larger block sizes (16×16 or even higher) are an obvious consideration. However with increased block size, more complex calculations come and the optimum quantization matrices (see Sec. 3.2) remain to be found.

3.2 *Psychovisual Quantization*

Having performed the discrete cosine transform on an 8×8 block, 64 pixel values have been transformed into 64 amplitudes of 2-D cosine functions of various frequencies. The eye, however, is not equally sensitive to all frequencies. Low-frequency variation within the 8×8 block is much more visible than the high-frequency variation. This is where quantization comes into play: we have to represent the low frequencies with a high accuracy, whereas we can use coarse measuring sticks to represent the high frequencies without jeopardizing the visual content of the blocks.[25–27]

In JPEG, all blocks in a given channel are quantized with the same quantization values, independent of the content of the blocks. The question is whether that is a good strategy—apart from being simple? An image normally contains objects or areas of varying psychovisual importance. In a portrait, obviously the person portrayed draws the main attention, so perhaps it would be fair to spend more energy (read data) on the blocks containing the person than on the blocks containing the background. This could be achieved using finer quantization steps on the important blocks than on the background blocks. For this to work in real life, however, the compressor would have to rely on an automatic segmentation of the image in different objects and subsequently assign importance to each object—not an easy task at that time.

During the development of JPEG, we made some much simpler experiments: just as high-frequency variation is less visible than low-frequency variation, so are dark areas less visible than well-illuminated areas. That led us to experiment with DC-dependent quantization. Blocks with low DC-values (dark blocks) could be quantized harsher than blocks with medium- or high-DC-values. The way we determined the quantization steps for each 2-D frequency was to find the amplitude of the corresponding 2-D cosine function, where it was just visible in the image; that limiting amplitude would then determine the quantization step.[25,26] It was obvious that the lower the DC-value, the larger the limiting amplitude and thus the quantization step.

These experiments showed, however, a very prominent problem with such content-dependent strategies: adjacent blocks treated with different quantization matrices are visually very different and thus add heavily to the very annoying blocking artifacts that are seen at high compression without really improving the compression rate.

The quantizing values are fully dependent on the imagery service application. The JPEG standard (part 1, Annex K) gives two examples of quantization tables that have been drawn empirically for Rec. 601 compliant images. So, the JPEG standard does not define any quantization matrices. The user can define his own 64 values for a given matrix, and can use different matrices or the same for the various color components. For the majority of images, matrices are chosen that treat low frequencies very finely and high frequencies more coarsely. But just a word of caution, do not use it as a general rule! Some images need another treatment, e.g., medical images—the high-frequency details are very important (e.g., fine lines in a pneumothorax x-ray).

The strategy for Y–Cr–Cb matrices was to find the smallest amplitude for each of the 64 basis functions that would render it visible under standardized viewing conditions [ANT Labs,[25] CCETT dedicated Lab for psychovisual evaluation[26] and Kjøbenhavns Telefon Aktie Selskab—Copenhagen Telephone Company (KTAS) Labs] and then use that amplitude as the quantization step for that 2-D frequency. One example is that Adobe Photoshop uses a set of different matrices for their different quality levels.

3.3 *Modeling and Coding*

Creating data for data compression consists of generating symbols that can later be coded by an entropy coder.

Transformation and quantization together produce datasets with a statistical structure that lends itself to complementary compression. The process is to ensure that this is the modeling (optimal-source symbols selection) and encoding of the selected symbols. Given that the majority of the quantized amplitudes are either zero or very small, and that most of the nonzero or larger quantized amplitudes pertain to the low frequencies, KTAS devised an ingenious way to encode these using value pairs. The first value in the pair tells how many zero-amplitudes to skip before the next nonzero amplitude (run length), and the second value in the pair tells how many bits are necessary to represent that amplitude. The value pair is then followed by the amplitude. When there are no more nonzero amplitudes in the block, an end-of-block code is emitted.

The statistical distribution of these value pairs is heavily skewed toward small values of both runs and number of bits, so the 2-D Huffman coding was the obvious choice.[28] With this encoding scheme (lossless entropy coding), significantly higher compression rates were obtained in JPEG.

3.3.1 *Coding of DC*

Coding of the DC values is kept very simple: After a level shift, the values in the first block row are predicted from the left neighbor. In the next rows, they are predicted from their left and above neighbors. The resulting values (which are peaked strongly around zero) are entropy coded.

3.3.2 *Coding of AC*

A JPEG block has one DC-value and 63 AC values, and the nonzero AC values (and large AC values) are strongly clustered toward the upper left corner of the coefficient matrix. In transmitting such data, it is very advantageous to make some kind of scan from the upper left corner of the matrix to the lower right corner, and to transmit the distance from one nonzero coefficient to the next rather than transmitting the zeros individually. This was generally accepted quite some time before JPEG began.[29]

An obvious and simple way of doing that is to first code the length of a “run” (up to 63 different symbols), and then to code the value (2048 different symbols). However, this would require two symbols per data-point (nonzero coefficient), and it would also neglect the very important correlation between run-length and coefficient size. Shorter runs are typically followed by larger coefficients.

The symbols in JPEG consist of tuples (run, log-value) followed by one or more bits to give the actual coefficient value (more about that later).

There are 63 possible runs plus one end of block (EOB) code, which is used to indicate that there are no more nonzero values in the block. That gives a total of 64 different run-length codes.

The number of log-value codes depends on the data depth. For 8-bit data it is 11 (the transformed and quantized data for an 8-bit input are 11-bits depth), for 12-bit data the number is 15, and for 16-bit data it is 19. This gives the number of different symbols for the tuple

8-bit data: 704 symbols,
12-bit data: 960 symbols, and
16-bit data: 1216 symbols

For Huffman coding, this is also the number of possible codes for a given bit-depth.

It would have been possible to code (run-length, value) instead of (run-length, log-value), but it would have given a very large Huffman tree over 130,000 possible codes for an 8-bit image, and even though we found a very efficient way of transmitting the coding tables, the table overhead would have been prohibitive.

The extra bits per coefficient are calculated as follows for the nonzero coefficients. If the coefficient is 1 or −1, we need just one bit: the sign. If it is in the range $[-3, 3]$, we need 2 bits: the sign and information on whether the absolute value of the coefficient is 2 or 3. If it is in the range $[-7, 7]$: 3 bits, $[-15, 15]$: 4 bits, $[-31, 31]$: 5 bits, etc. It may seem inefficient to send the remaining bits for the codes rather than entropy coding it, but actually for a given group the probability distribution in the group is rather flat, so you would not gain much by entropy coding it.

3.3.3 *Progressive modes*

The data sent in the progressive modes (successive approximation, spectral selection, and combinations thereof) give the same reconstructed image as the sequential modes, but you get a recognizable image earlier in the data-stream.

The coding is, however, somewhat different and the compression ratio is not quite as good as the sequential modes.

In the progressive modes, the image is sent in two or more "scans."

For successive approximation, the high-order coefficient bits are sent before the low-order bits (for example, bits 3 to 7), and this means, that for the first scan(s) there are fewer nonzero coefficients, and thus the run-lengths are larger for those scans. That of course gives different statistics for each scan and therefore also different coding tables for the different scans.

For spectral selection, you transmit the earlier coefficients in the zigzag pattern in the first scan and then transmit the rest of the coefficients in one or more later scan(s). It does mean that you will code more EOB-s, and it also means that the first runs in the later scans will typically be longer than in sequential transmission.

It is possible to use hierarchical (pyramidal) coding in JPEG, but it is not much used. Also it does not combine with the two other progressive modes. However, spectral selection is a "poor man's" version of it: the first four AC coefficients in the zigzag scan will let you reconstruct a perfect 4:1 subsampled image (and the first 31 coefficients will let you reconstruct a 1:2 subsampled image.). And then of course you also get a fine 8:1 image by just looking at the DC.

After quantization, the majority of nonzero amplitudes pertain to low-frequency basis functions clustered in the upper left-hand corner. Zigzag scanning through these amplitudes is but one way to arrange them in the descending order of importance—the way chosen in JPEG-1. Other ways could, however, be envisaged, e.g., an orthogonal scan. The optimal scan order is image dependent. It would have been extremely cheap—costing only says 3 bits in the JPEG header—to tell which of say eight different scan orders where used in a particular image, and one of those should be user-definable.

Arithmetic coding did not become popular, because while Huffman coding was believed to be royalty free on the JPEG-defined arithmetic coding International Business Machines Corporation (IBM), AT&T, and Melco filed RAND patent statements to ISO and ITU. Strangely enough in an early phase of the independent JPEG group (IJG) code, it was implemented and even released but when it became clear that it was not royalty free the IJG removed it from the code and suggested not to use it, also because of the expected compression gain of about 6% did not make it too attractive and also the complexity of the entropy codec was too high. Likewise it took a very long time for progressive build-up to be widely used, but the W3C usage and demand for progressive build-up on slower Internet speeds made the application finally attractive.

3.3.4 *Baseline and profiles*

Writing an international standard is a major task. During the first stage of drafting, it was proposed to have a kernel (baseline) that fulfilled most expected requirements of videotex and envisaged telecommunications services. Furthermore, it was decided that it should be royalty free. Around the kernel, "onion like", profiles are added for specific applications and allowing options such as arithmetic coding (which may be RAND licensed).

The initial robust coding scheme has a very low algorithmic complexity with three coding/decoding stages: FDCT is a few adds/multiplications; quantization is a simple division or a shift; and coding is a look-up-table. It was easy to understand and implement. The baseline has proven to be adequate for the majority of applications and is massively used.

3.4 *Evaluation Process*

JPEG had big ambitions to produce a compression technique for continuous tone images for applications ranging from photovideotex to medical imaging. Control of compression versus quality was key and both progressive and sequential images build-up were required. As well as normal lossy compression, a lossless option was needed (for applications, such as medical imaging and surveillance).

A process was agreed to evaluate techniques submitted by countries/organizations in JPEG. This involved subjective testing of image quality at defined compression stages and a demonstration that candidates' techniques could be decoded in real-time by the hardware/software available at the time. A set of documentation also had to be provided with each submission.

Ten compression techniques were registered for the initial selection process held in Copenhagen in June 1987. These included the two PICA[30–32] techniques from Europe (adaptive discrete cosine transform and progressive recursive binary nesting) and other techniques from Japan and America.[33] The techniques included most of the coding methods that had been researched and published in the scientific/engineering community, i.e., predictive, transform, and vector quantization.

3.5 *Final Selection Process*

Three techniques stood out at the initial selection process—the European (PICA) adaptive discrete cosine transform (ADCT) technique, the American (IBM) DPCM-based technique, and the Japanese block truncation coding scheme. These three techniques were used as the basis for further development by international teams led by Europe, America, and Japan, respectively, for the final selection meeting held at Copenhagen Telecom's Laboratories (KTAS) in January 1988.

For the final selection, the test requirements were increased. Subjective testing took place at 2.25, 0.75, 0.25, and 0.08 bpp (compression 200:1) using five new test photos for which the candidate algorithms were not trained before the selection meeting (Fig. 7). A double stimulus technique was employed whereby images were compared with the original.[34]

It was clear from the subjective testing that the ADCT technique produced better quality results at all the compression stages (Fig. 8). Excellent results were achieved at 0.75 bpp (i.e., close to 20:1 compression) and results indistinguishable from the original were produced at 2.25 bpp.

For the JPEG ADCT candidate, it was KTAS (Birger Niss) that produced the reference software coded in FORTRAN on a VAX computer (Virtual Address

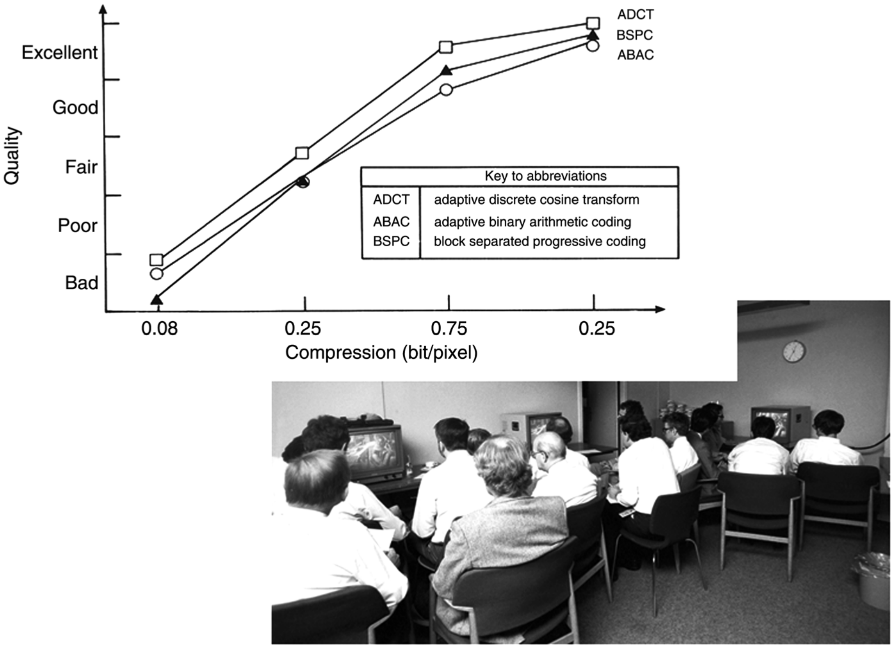


Fig. 7 Final subjective testing results (January 1988).

eXtension, a discontinued computer from Digital Equipment Corporation) and sent out on a VAX tape.

3.6 *Real-Time Implementations*

In the final selection round, the JPEG committee imposed a new requirement that demonstrators of software implementation on a 25-Mhz IBM PC were to decode a CCIR 601 image in real-time (suitable for ISDN 64 kbit/s speed). This was shown to be possible at the final selection meeting (Copenhagen, January 1988) by KTAS with a real-time implementation done in C/C++ and a little bit of x386-assembly language. This was certainly a good decision as it showed that JPEG was decodable in software.

At the same selection meeting, another implementation of the ADCT algorithm (close to the future JPEG baseline) was shown running at real-time (ISDN) implemented in software on DSP (Texas Instruments TMS320) on a PC/AT by

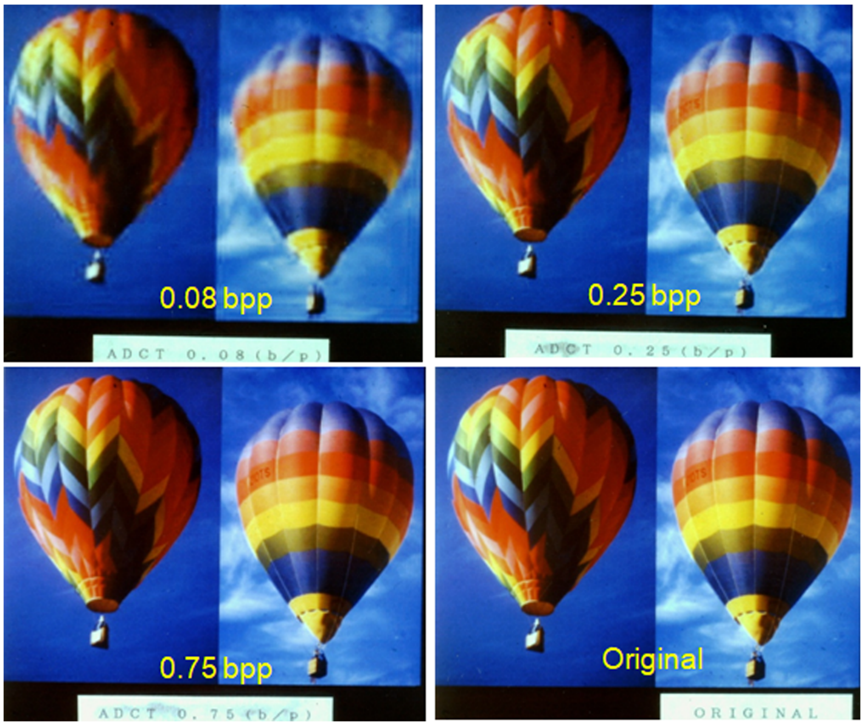


Fig. 8 Final ADCT compressed results (0.08, 0.25, and 0.75 bpp, original) (January 1988).

SAT-CCETT.[35] Lastly, the silicon version (CCETT-Matra Communication) for real-time coding and decoding was announced.[36]

These convincing results launched the open-source community to develop rapidly compliant software before the final writing and approval of the standard. Note that this is now a common practice in web and programming language-related standardization, such as in W3C (e.g., RDF, OWL, and SPARQL) or in Ecma International (e.g., ECMAScript/JavaScript).

It was unanimously agreed to go ahead with the development of a standard based on the ADCT technique.[37–39]

4 Known Limitations (1992)

The JPEG compression algorithm was clearly focusing on natural image or realistic scenes. Its performances were best on this class of images. For graphics and text, JPEG was not as well suited, especially at very low bit rates where artifacts appear at the boundaries of high contrast areas. So, for this class of images the JBIG (ISO|ITU Joint Bilevel Image experts Group) started the study of specific algorithms in 1988 and delivered an international standard in 1993 (ITU-T T.82), actually only a few months after the JPEG standard's approval.

Otherwise, the lossy mode of compression was the major application need at this time (e.g., in facsimile or in medical imaging), and the research activity was focused on the finding of the best rate versus distortion achievement. So, the lossless mode was considered later in the drafting of the recommendation and a simple DPCM lossless solution was proposed in JPEG. Later, a specific JPEG recommendation for a more efficient lossless and near lossless mode was standardized (ITU-T T.87 in 1994).

Other improvements were also envisioned during the drafting of the JPEG standard as described as follows.

4.1 *Blocking Artifacts*

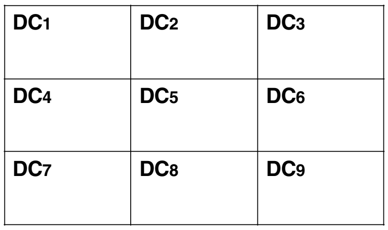
Blocking artifacts appear especially at low bitrate according to the image content. This artifact is clear on Fig. 8 at a bitrate of 0.08 bpp (compression 200:1). The wavelets used in a JPEG2000 standard[36] being a multiresolution signal transformation by nature reduces this major artifact and shows a graceful degradation (more linear) when compression increases. However, wavelets were just not feasible with the hardware of the day, and with the speed requirements (real-time decoding at ISDN 64 kbs). Furthermore, JPEG2000 standardized later with DWT but was not chosen to replace JPEG (1992) for the mainstream applications.

4.2 *AC Prediction*

A vital part of image compression is decorrelation. The DCT is close to optimal for decorrelating the values within (intra) the 8 × 8 pixel blocks. Using the DC value of the preceding block as a predictor for the current block is used in the standard.

During the development of JPEG (1992), a scheme for a more advanced interblock decorrelation, namely AC prediction was suggested by KTAS.[40]

Let us assume that the pixel field within nine blocks (Fig. 9) can be modeled by a biquadratic function Eq. (1)



DC1	DC2	DC3
DC4	DC5	DC6
DC7	DC8	DC9

Fig. 9 Nine blocks of DC values.

$$P(x,y) = A_1x^2y^2 + A_2x^2y + A_3xy^2 + A_4x^2 + A_5xy + A_6y^2 + A_7x + A_8y + A_9, \tag{1}$$

where $x, y = [-11.5; +11.5]$, $P(x, y)$ is the pixel value at (x, y) with origin at the center of the central block.

The nine coefficients A_1 through A_9 can be uniquely calculated from the constraint, that the sum of pixels in each of the nine blocks equals 64 times the DC-value. Equipped with the calculated coefficients, all pixel values in the central block and thus its DCT can be calculated. This calculation turns out to be very simple indeed requiring only a few operations. Although all 63 AC coefficients can in principle be predicted, it really makes little sense to use the predicted high-frequency AC coefficients, so only the 14 low-frequency coefficients are predicted.

Having performed the DCT on the central block, the predicted AC coefficients are subtracted from the real values and the residuals are then quantized. Ideally the quantized residuals will be zero, or at least smaller than the original thus resulting in an increased compression. This may be the result in slowly varying parts of an image. In very active parts of an image, the prediction may well fail resulting in residuals larger than the original, leading to reduced compression.

An example of AC prediction is shown as follows.

Figure 10(a) shows a 256 × 256 test image containing only low-frequency variations. Compressing this image using the Independent JPEG Group implementation and quality level 50, gives a compressed size of 9363 bytes. Figure 10(b) shows the same image, where the predicted AC coefficients have been subtracted. The compressed size is 4935 bytes (47% reduction). Finally, Fig. 10(c) shows the resulting image where the predicted AC coefficients are added again after dequantization but before the IDCT. Note that such a large increase in compression is rarely seen.

However, JPEG has not integrated this scheme for prediction of the AC values due to the increased complexity. Instead, it was suggested as a decoder-only option. That, however, makes little sense, as adding extra AC values to existing values would ruin the image.

4.3 *Postprocessing and Artifacts Reduction*

To obtain a substantial compression, harsh quantization is needed. With that comes, however, visible artifacts. When low-frequency coefficients are mistreated, very annoying blocking artifacts appear. When high-frequency coefficients are absent or illrepresented, the so-called ringing around sharp edges appears.

Two different approaches may be considered to reduce these artifacts: intelligent noise and AC prediction.

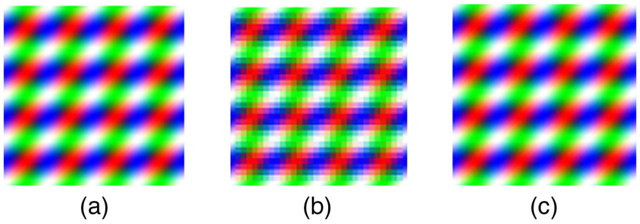


Fig. 10 Benefits of AC prediction (a, b, and c).

4.3.1 *Intelligent noise*

When a given coefficient is quantized it may take on only certain discrete values determined by the quantization step. When the quantized value is zero, compression really kicks in. Dequantization simply consists in multiplying the quantized coefficient with the quantization step. Example: When a quantized coefficient is zero, the dequantized coefficient is zero as well. It is, however, not very likely that the original value was zero.

It could have any value in the interval: $[-Q/2; Q/2]$, where Q is the quantization step. Why not assign the dequantized coefficient an arbitrary value in that interval? The same goes for nonzero quantized coefficients—let the dequantized coefficient have an arbitrary value in the interval around the central value. Obviously this scheme does not add true information to the image, but it does replace the well-known JPEG artifacts with a grainy look not unlike old-fashioned high ISO films.

4.3.2 *AC prediction*

AC prediction is described in detail already. It is shown that predicted low-frequency coefficients taken out of the image before quantization and inserted again after dequantization often give a substantial reduction in blocking as shown in Fig. 11.

Where intelligent noise is a decoder-only remedy, AC prediction must be used both at the encoder and the decoder.

4.4 *Lossless*

Although a great amount of research energy went into the development of the lossy modes of JPEG, the lossless mode required by the standardization committee was developed in haste at that time.

As an approximation for the cosine transform was not properly defined, as integer DCT would have provided, the obvious first choice was the straightforward DPCM applied in the pixel domain, in combination with entropy coding (Huffman), where the value of a given pixel in a given color component was represented by the difference between the true value and a predicted value. Seven predictors are defined in the standard.[41,30]

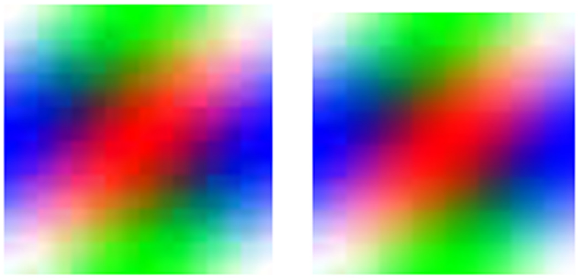


Fig. 11 (a) Without AC prediction and (b) with AC prediction.

On real-life images, the compression can vary substantially (25% to 30%) with the choice of predictor. That said, however, JPEG lossless mode is not very efficient. However due to its simplicity, it has found applications in the transmission of images from the Mars Rover and in satellite imaging applications. Typical compression factors between 2 and 3 can be obtained depending on the complexity of the image and—very importantly—the pixel noise in the image. Curiously, plain zip applications such as 7z perform almost as well as a JPEG lossless mode. For that reason, JPEG LS (ITU-T T.87 (1988)|ISO/IEC 14495-1:1999) based on the LOCO-I algorithm.[42,40] was standardized with Huffman coding in 1999 and with extensions such as arithmetic coding in 2003. JPEG LS can typically give compression factors better than 4.

However, the camera and scanner industry adopted the RAW or similar format (ISO 12234-2 in 2001) for storing original images in case of particular postprocessing needs.

5 Main Milestones

The seven European PICA partners developed and evaluated 10 different techniques and two were submitted to JPEG for consideration—a predictive technique and a transform technique. The results of the Esprit project were presented to the ESPRIT conferences[43] (1986 to 1988).

The PICA methodology used to evaluate techniques and the resulting compression techniques developed provided a great impetus for the JPEG standards activity. The entire JPEG group after 1988 works very hard to turn the technology into an international standard (Table 2).

5.1 *Drafting of the Standard and the Validation Process (1988 to 1991)*

During the 3 years drafting, the standard based on the ADCT technique that fulfilled all the criteria of the international selection committee, further development work was done to enhance the technique to cover more application needs and to allow for verification and deployment. It was also the time for worldwide diffusion especially through national standard organizations (AFNOR, ANSI, DIN, BSI, etc.) to check in particular, any possible patent infringements.

Although the developments and the verifications of the drafted standard were done cooperatively by all the JPEG members, the writing of the 10918-1 standard was mostly done by the IBM team (Joan Mitchell, William Pennebaker) and DEC (Greg Wallace) on a VAX text processor.

6 Some Successful Factors

6.1 *ESPRIT PICA and JPEG Early Days*

From early work in the European telecoms (CEPT) and the world standards arenas (ISO and CCITT), in 1985, the European Commission supported the establishment of a European collaborative project PICA under the ESPRIT program to produce a photographic compression technique for international standardization.

Realizing the importance of picture coding for future multimedia communication, a rationalization of the standards organizations took place in 1986 resulting in the

Table 2 Historical milestones.

Date/location	Historical foundations
1982	Introduce image coding for videotex at CEPT (Conférence Européenne des Administrations des Postes et Télécommunications)
June 1985, Ipswich	Launch the European Photovideotex Image Compression Algorithms (PICA) project
November 1986, Parsippany	ISO and CCITT form JPEG
March 1987, Darmstadt	Register coding schemes and define requirements and selection process
June 1987, Copenhagen	Hold initial selection meeting—10 techniques reduced to 3
October 1987, Washington	Complete first revision of final specification and selection process
December 1987, Winchester	Complete second revision of final specification and selection process
January 1988, Copenhagen	Hold final selection meeting—ADCT technique chosen
June 1989, Rennes	Refine and consolidate the ADCT technique by the JPEG international team
1989	Write the JPEG draft international standard with ITU/ISO/IEC common template
September 1992	Approve JPEG as Recommendation ITU-T T.81
November 1994	Approve JPEG as ISO/IEC 10918-1 standard

formation of the joint photographic experts group (JPEG). CCITT provided the service requirements for JPEG's technical experts to develop and evaluate a picture-coding technique for an ISO coding standard.

6.2 *Legacy*

Today in social media applications alone, more than 2 billion pictures are being distributed every day. There is a huge collection of JPEG pictures in archives around the world. The vast majority of these pictures are JPEG encoded and so JPEG will live on for decades to come.

6.3 *Patents and Open Source Implementation*

As it was pointed out earlier the base JPEG components, the so-called "JPEG baseline" is "royalty free" that has allowed easier implementation and rapid spreading of the standard. Also the "tool-box" principle is particularly suitable for "open source implementations."

- CCITT (ITU) and ISO/IEC have a so-called RAND (reasonable and nondiscriminatory patent/licensing policy), but not "royalty free" (like the patent policy of the WWW and some others). Therefore, the *de-facto* declaration of the JPEG committee of the "royalty-free" baseline JPEG was *de-jure* not supported by the ITU and ISO/IEC patent policies. Nevertheless, in practice it worked quite well, except for some cases as JPEG became the dominant image compression standard in the market but not all claimed patents had expired. Around the year 2000, a number of JPEG licensing litigation cases arose from companies who claimed to have patents on the "baseline" JPEG. Today the problem is solved by the simple fact that all possible JPEG patents have expired. Nevertheless, if today a similar IPR model was envisaged by the standardization community such as JPEG it would probably be better to select an SDO that can deal both with RAND- and RF-based patent policy regimes in an effective manner.
- When JPEG was developed, open-source projects hardly existed. The implementation of the JPEG "Toolbox" by the so-called IJG was launched by Mr. Tom Lane. IJG is an informal group that writes and distributes a widely used free library for JPEG image compression. The several demonstrator implementations of JPEG by JPEG committee member companies (even if their code were not made public) were very helpful to prove the feasibility of implementation. The first version of the IJG software code was released on October 07, 1991. This was *de-facto* in parallel with the approval and publication process of the JPEG standard in 1992 to 1993. This code has been a stable and solid foundation for many applications' JPEG support and helped tremendously to the widespread application and use of JPEG. It is interesting to note that the IJG has only used those JPEG components that were believed to be royalty free. For example, arithmetic coding (with RAND patent declaration) was not implemented and never became popular. Also to note the availability in the public domain of the JPEG coder and decoder in C++.[44] Recently, the JPEG Committee has issued reference software for the original JPEG-1 standard. This initiative closes a long-standing gap in the legacy JPEG standard by providing two reference implementations for this widely used and popular image-coding format.[45]

6.4 *JPEG Applications Markers*

Markers in the JPEG syntax such as the "application" markers let you put any defined information in a data stream (up to 64 kB in each marker). The extra information can be a thumbnail image or other application relevant data.

One particularly useful application marker is the EXIF header, which is used by almost every modern digital camera. It lets you describe the recording conditions, such as exposure time, aperture, geographical position, orientation plus many other parameters.

Only the format of application markers is defined by JPEG. The contents may be defined in other standards. For example, EXIF is defined by JEIDA (now JEITA). Applications markers may even be "private" to a vendor or an application.

One of the great benefits of the markers is that they remove the need for a "wrapper format." The data stream is self-contained: one image—one datastream.

The JPEG syntax lets you define a "restart marker" to insert at regular intervals "points of synchronization" that can have uses such as "limited" search ability in a data stream and restart point for a corrupted transmission channel.

6.5 *Document Archival of JPEG Standardization Documents*

A high-quality archival of standardization documents is an essential requirement for any standardization, and it is one of the criteria required by the World Trade Organization. For JPEG, such high-quality documentation became very important when the problems of the JPEG "royalty free" licensing emerged starting in 2000. Then it turned out that the most important documents for such cases lie within the JPEG committee, but not in the "mother" standardization organizations ITU, ISO, and IEC. (They only stored documents related to higher standardization groups, such as SCs, ITU-T SGs.) With a so-called "JPEG historical archive" program, JPEG was capable of restoring the original JPEG document archive, which then played an important role in the patent litigation cases. As one of such cases now the SDO, such as ITU and ISO have learned that all standardization documents are to be archived in the long run, and actually from the patent point of view, the documents of the appropriate lower-level working group such as JPEG are most important.

6.6 *Standardization Working Spirit*

JPEG had a committed, collaborative, and competitive atmosphere during the selection phases (1986 to 1988), but following the final selection meeting (January 1988) everyone worked together as one team to obtain the best scientific/engineering solution where everyone could share the results. Any intellectual property had to be declared and made available under at least fair and reasonable terms (RF for the "baseline"). Keys to the success of the development and standardization process were the following:

- A process developed and agreed by JPEG delegates.
- Any JPEG member was allowed to submit techniques for consideration.
- Throughout the evaluation process, some techniques were combined and the developers encouraged to work together and to produce new composite technique specific goals.

7 Initial Industrial Applications

The application of JPEG has in fact far exceeded our original expectations and it is clear today that the standard has been a building block of the digital information revolution.

After the selection of the foundations of the JPEG algorithm (January 1988), the 3 years of the writing of the technical specifications of the ISO/IEC JPEG standard (1988 to 1991) were devoted to technical refinements and extensions, numerous validations and corrections and not least, tests in a far larger range of applications than initially anticipated (videotex in early 1980s).

In the late '80s, JPEG became key to many of the emerging applications of the digital information revolution such as: "e-commerce" (online selling), "property selling" (online real estate), "e.medicine" (for medical image storage and remote diagnosis), prepress transmission of global event images to agencies such as AFP, "police" (scan of suspect's fingerprints), and in retrospect perhaps the most significant, the use of JPEG for digital cameras.

A clear sign of the potential success of JPEG in the late '80s was that before the completion of the final standard approval (ITU-T T.81 in 1992 and ISO/IEC IS 10918-1 in 1993), many entrepreneurs launched development projects and small companies based on the applications of the JPEG standard. To name some prominent early adopters: SAT (Sociétée Anonyme de Téléphonie), AT&T Microelectronics, Matra Microelectronics, Storm Technology Inc., Autograph International ApS, C-Cube Microsystems Inc., LSI Logic, Philips Kommunikations Industrie, and Zoran Corporation.

Eventually after completion and publication of the JPEG standard in ISO (1993) and ITU (1992), it was not the videotex initially envisioned application that was the killer application for JPEG, but the Web that adopted the JPEG standard in 1994 (W3C). Obviously, videotex experiments and deployments in many countries (Europe, USA, Japan, and Canada) were undoubtedly precursors of its generalization on the web, starting slowly from 1995 but with an exponential growth since then.

However, it is worth mentioning that, each application area has followed its own pace. For example, late acceptance in medical and e.commerce/banking environments was due to legal and other constraints. Mass production of digital cameras awaited CCD image sensors of sufficient resolutions, from about 1 Mpixel in 1993 to 24 Mpixels or more in 2016.

Clearly today, JPEG (lossy version) is massively used in all general public communication applications (Information and Telecommunication Technologies), and used less in some professional areas where decoded image quality is paramount.

Before this massive market penetration of JPEG, to overcome some known limitations of the JPEG standard the JPEG committee launched the JPEG2000 project (1997).

8 JPEG-1 Extensions

In the late '90s, a number of ideas were mooted for amending, developing, and extending the JPEG-1 algorithm.[45]

8.1 *JPEG Extension Needs*

Extensions were considered for the following:

- Lossless and lossy compression of continuous-tone with reduced distortion and superior subjective performance. In particular to improve image quality at low bit rate for graceful degradation of quality, for example, using larger or variable block sizes, or some interpolation.
- Random access to spatial regions (or regions of interest) as well as to components.

Each region can be accessed at a variety of resolutions and qualities.

- Operation modes (sequential, progressive, hierarchical, and lossless) to be integrated in it in a "compress once, decompress many" paradigm.
- Robustness to bit errors (e.g., for mobile image communication).
- New imaging functionalities handling, such as HDR image, 360-deg imaging, holography, and plenoptic imaging.

8.2 *JPEG 2000*

A great effort was then made to the development of JPEG2000 (1997 to 2004).[42,46] The new proposed standard followed basically the same evaluation and checking processes as the original JPEG algorithm and was approved in ITU-T and ISO (JPEG2000 Image Coding System Part 1: 2004).

It is now apparent that the industry did not choose to switch from JPEG to JPEG2000 for the mass market. Although JPEG-1 was over the WWW already very popular, it was considered that the little improvements gained in picture quality were not worth the increase in complexity and few other considerations in JPEG2000[42] such as the impossibility to select prior to coding, the picture quality level (e.g., extra fine, fine, and medium). At the beginning also the patent situation was not clear, e.g., the standard defined only the decoder, whereas the patents in the encoder were completely of the range of standardization. This gap was closed several years later when JPEG-1 gained further popularity with smartphone applications.

Another important reason was that the DCT was well understood, and easily implemented (and a core part of many firmware repertoires). The criteria were always on low power, memory and performance, and so, DCT was preferred overall.

JPEG2000 has been adopted in a number of professional image application areas (Digital Cinema DCI, medical DICOM, British Museum, Library of Congress, Open Geospatial Consortium, Google Imagery, etc.).[47]

It seems that the main uses for JPEG 2000 are as a central single source of data for transcoding on the fly to existing browsers. Its core usage is in "niche" markets, such as digitizing newspapers, geospatial imagery, census data, medical images, and so on where there are extensive metadata associated with an image, and even in those markets its prime use is in a single central repository (cloud based these days), which is likely to be subject to subsequent reprocessing for analytical or display purposes.

Launched in 2017, high-throughput JPEG 2000 (HTJ2K) aims to develop an alternate block-coding algorithm that can be used in place of the existing block coding algorithm specified in ISO/IEC 15444-1 (JPEG 2000 part 1). The objective is to significantly increase the throughput of JPEG 2000, at the expense of a small reduction in coding efficiency, while allowing mathematically lossless transcoding to and from code streams using the existing block coding algorithm.

A selected a block-coding algorithm has recently demonstrated an average 10-fold increase in encoding and decoding throughput, compared with the algorithms based on JPEG 2000 part 1. This increase in throughput results in <15% average loss in coding efficiency, and allows mathematically lossless transcoding to and from JPEG 2000 part 1 codestreams.

A working draft of part 15 to the JPEG 2000 suite of standards is currently under development.

8.3 *JPEG XT and Plenoptic Imaging*

8.3.1 *JPEG XT backward compatible with JPEG-1*

JPEG XT ("XT is short for eXTension")[48] is both backward-compatible to the legacy JPEG (JPEG-1), and offers the ability to encode images of higher precision (16 bits per component), HDR, in lossy or lossless modes, and also allows a transparency layer (alpha channels), 360-deg panoramic imaging, privacy protection, and security in image regions.

Any legacy JPEG decoder will be able to decode a JPEG XT file. In that sense, JPEG-1 decoder that only understands JPEG-1 and not JPEG XT will only get an 8-bit lossy image. Lossless decoding, or full sample precision, would still require a full JPEG XT decoder.

JPEG XT does this by first encoding an 8-bit version of the high-precision input, also called base layer, and hiding a second codestream known as enhancement layer, within this legacy codestream that enlarges its precision to a fuller range (up to 16 bits per component). Additional metadata, also embedded in the legacy codestream, tell a JPEG XT decoder how to combine the base layer and the enhancement layer to form one single image of a higher precision.

Embedding mechanism used in JPEG XT is possible thanks to a legacy JPEG structure called "application marker" (see Sec. 6.4).

8.3.2 *JPEG-PLENO*

The JPEG committee is now focusing on the representation and compression of new image modalities, such as light field, point cloud, and holographic content coding.

JPEG pleno light field finished a third round of core experiments for assessing the impact of individual coding modules and started work on creating software for a verification model. Moreover, additional test data have been studied and approved for use in future core experiments.

JPEG pleno point cloud use cases are under consideration. A final document on use cases and requirements for JPEG pleno point cloud is available. Working draft documents for JPEG pleno specifications parts 1 and 2 are also available.

JPEG pleno holography has edited the draft of a holography overview document. The current databases are

Table 3 Image compression standards.

	JPEG: digital compression and coding of continuous-tone still image
September 1992	ITU-T T.81|ISO/IEC 10918-1: requirements and guidelines
Nov 1994	ITU-T T.83|ISO/IEC 10918-2: compliance testing
July 1996	ITU-T T.84|ISO/IEC 10918-3: extensions
June 1998	ITU-T T.86|ISO/IEC 10918-4: registration of JPEG parameters
June 1998	ITU-T T.87|ISO/IEC 14495-1: lossless and near lossless
May 2011	ITU-T T.871|ISO/IEC 10918-5: JFIF
	ECMA TR/98 (JFIF) in June 2009
In process	ISO/IEC 10918-7: reference software for ISO/IEC 10918-2
	JBIG: bilevel image compression and coding
March 1993	JBIG ITU-T T.82|ISO/IEC 11544: “JBIG1” lossless compression for text, line art, and Halftone images
February 2000	ITU-T T.88|ISO/IEC 14492: “JBIG2” text, line art, halftone, visually-lossless text
	JPEG 2000: image coding system
December 2000	ITU-T T.800|ISO 15444-1: core coding system
November 2001	ITU-T T.801|ISO 15444-2: extensions
November 2001	ITU-T T.802|ISO 15444-3: motion JPEG 2000|
May 2002	ITU-T T.803|ISO 15444-4: conformance testing
November 2001	ITU-T T.804|ISO 15444-5: reference software
April 2003	ISO 15444-5|part 6: compound image file format
2006	ITU-T T.807|ISO/IEC 15444-8: JPSEC image security
October 2004	ITU-T T.808|ISO/IEC 15444-9: JPIP interactivity tools: APIs and protocols
June 2007	ITU-T T.810|ISO/IEC 15444-11: JPWL wireless applications
July 2003	ISO/IEC 15444-12: part 12: ISO media format
	JPEG: extensions—digital compression and coding of continuous-tone still image
March 2009	ITU-T T.832|ISO/IEC 29199-2, JPEG XT
In process	JPEG Pleno families: core experiments
	HEVC/HEIF high efficiency video coding and image file format
April 2015	ITU-T H.265|ISO/CEI 23008-2 HEVC—edition 3
December 2017	ISO/IEC 23008-12:2017 HEIF—edition 1
	JPEG XL next generation image coding
April 2018	Call for proposals

classified according to use cases, and plans to analyze numerical reconstruction tools are established.

8.3.3 *JPEG-XL*

The JPEG committee has launched recently (April 15, 2018) a next-generation image coding activity, referred to as JPEG XL.[45] This activity aims to develop a standard for image coding that offers substantially better compression efficiency than existing image formats (e.g., >60% improvement when compared with the widely used legacy JPEG format), along with features desirable for web distribution and efficient compression of high-quality images.

8.4 *Discussion*

In this article, we have focused our presentation on the very successful legacy format JPEG-1 and its past and current extensions; however, it should be noted that current new video compression standards such as HEVC (ISO|ITU standard since April 2012)[49,50] have demonstrated via a series of subjective and objective evaluations that HEVC intracoding High Efficiency Image File Format (HEIF) outperforms standard compression algorithms for still images with an average bit rate reduction ranging from 16% (versus JPEG 2000 4:4:4) up to 43% (versus JPEG-1).[51] Another possible recent contender could be the AV1 codec (in its still image format AVIF),[52] an open, royalty-free video coding format that is being developed by the Alliance for open media (industrial consortium comprising Amazon, Apple, ARM, Cisco, Facebook, Google, IBM, Intel Corporation, Microsoft, Mozilla, Netflix, and Nvidia) (September 2015), a competitor to HEVC.

However, for the video market, HEVC has been adopted by some large companies and is used daily by a lot of customers and this continues to grow, so the benefits of HEVC have already been proven. The adoption by industry of the AV1 format remains a hot-debated subject especially on the patents issue (royalty free versus RAND). Thus, it is difficult today to predict the industrial outcomes.

Finally, it is interesting to see that all those evolutions (see Table 3) of the original JPEG-1 compression algorithm are always following the same general scheme, using the same basic tool-box, borrowing many of our historical choices, such as on IPR for instance. It still appears to be extensions—compatible or not—of the original architecture, requested by the today's image environment that has drastically changed from our environment in the '80s. So, it is very rewarding for the initial JPEG committee to have laid down such a foundational basis.

9 Conclusions and Future Outlook

The JPEG-1 (ITU-T T.81¦ISO/IEC 10918-1) still picture coding is one of the biggest ICT standardization success-stories of the past decades. In the past 25 years, it enabled the creation, transmission, and storage of several trillions of still pictures worldwide and JPEG capturing and display devices in billions.

JPEG-1 is a "tool-box" standard. In the future, some tools may be obsolete and some new tools added to meet the requirement of new applications and services. However, for average mass-applications, such as picture taking with domestic digital cameras and mobile phone cameras and for web pages the starting "tool-box" is usually enough and will remain so for a significant time in the future. How long that will be is difficult to predict. However, the trillions of JPEG pictures taken and archived so far will always require that they should be easily and fully displayed any time in the future. In the history of mankind reminders from the past (whether in writing, in signs, in paintings, in sculptures, or by JPEG still pictures) will always be a significant part of the human heritage. Therefore, JPEG-1 has not only a great past but also a great future.

The authors of this paper are proud and grateful to be part of this exciting standardization process.

Acknowledgments

Referring to the original JPEG group (1985 to 1992) and its founders (1980 to 1985) who more or less directly contributed to the JPEG standard, the following longtime JPEG core members have spent days and days (no e-mail service at that time, simply hard-copy fax!) to make this collaborative international effort a 25-year-old success. Each has made specific substantive contributions to the JPEG proposal, Aharon Gill (Zoran, Israel), Eric Hamilton (C-Cube, USA), Barry Haskell (AT&T Bell Labs), Alain Léger (CCETT, France), Adriaan Ligtenberg (Storm, USA), Herbert Lohscheller (ANT, Germany), Joan Mitchell (IBM, USA), Michael Nier (Kodak, USA), Takao Omachi (NEC, Japan), Fumitaka Ono (Mitsubishi, Japan), William Pennebaker (IBM, USA), Henning Poulsen, Birger Niss, and Jorgen Vaaben (KTAS, Denmark), Yasuhiro Yamazaki (KDD, Japan), Dennis Tricker (BT Labs, UK), Mario Guglielmo, Luisa Conte and Garibaldi Conte (CSELT, Italy). The leadership efforts of Hiroshi Yasuda (NTT, Japan), the Convenor of JTC1/SC2/WG8 from which the JPEG standard was selected, refined, and written, Istvan Sebestyen (Siemens, Germany), the Special rapporteur from CCITT SGVIII, Gregory Wallace (DEC) later ISO JPEG chair and Graham Hudson (British Telecom, UK) first chair of ISO JPEG and chair of ESPRIT PICA the foundational European project that led to JPEG. A particular acknowledgment was addressed to the actual JPEG chair Touradj Ebrahimi, who has been the initiator in 2012 of the JPEG 25th anniversary celebrations and a supporter of this memorial work. The idea for this paper emerged thanks to the research teams and speakers who devoted considerable time to the preparation of the celebration of the 25th JPEG anniversary: Lausanne, CH (2012), Cesson-Sévigné, FR (2014), St. Malo, FR (2016), Leipzig, DE (2016), Torino IT (2017), Macau MO (2017), and San Diego, USA (2018).

References

1. D. Wood and S. Baron, "Rec. 601—the origins of the 4:2:2 DTV standard," EBU Technical Review and SMPTE Journal (2005).
2. S. Sasson, "The dawn of digital photography—megapixel," http://www.megapixel.co.il/english/archive/35884.
3. R. C. Nicol and B. A. Fenn, "Coding for a viewdata 'picture in picture facility'," in *Proc. of the Picture Coding Symp.*, Ipswich (1979).
4. K. E. Clarke, "What kind of pictures for videotex," in *Viewdata Conf.*, London (1980).
5. R. C. Nicol and B. A. Fenn, "Transmission techniques for picture viewdata," in *IEEE Int. Broadcasting Conf.*, Brighton (1980).
6. G. P. Hudson, "The development of photographic videotex in the UK," in *IEEE Globecom Conf.*, Atlanta (1984).
7. A. Léger, "Implementation of fast discrete cosine transform for full color videotex services and terminals," in *IEEE Globecom, Proc. of the Global Telecommunication Conf.*, Atlanta, Georgia, pp. 333–337 (1984).
8. ITU, "Common components for image compression and communication—basic principles," CCITT Recommendation T.80 (1992).

9. ITU, "Information technology—digital compression and coding of continuous-tone still images: extensions," CCITT Recommendation T.84 (1996).
10. ISO, "Information technology—digital compression and coding of continuous-tone still images: extensions," ISO/IEC 10918-3:1997 (1997).
11. ECMA International, "JPEG file interchange format (JFIF)," ECMA TR/98 (2009)
12. G. Wetzstein and Stanford University, "The human visual system," EE267 Virtual Reality Lecture 5, http://stanford.edu/class/ee267/.
13. A. Ortis and S. Battiato, "A new fast matching method for adaptive compression of stereoscopic images," *Proc. SPIE* **9393**, 93930K (2015).
14. SMPTE, "Study group report high-dynamic-range (HDR) imaging ecosystem," https://www.smpte.org/sites/default/files/Study%20Group%20On%20High-Dynamic-Range-HDR-Ecosystem.pdf (19 September 2015).
15. M. Řeřábek and T. Ebrahimi, "Light-field image Dataset," https://mmspg.epfl.ch/EPFL-light-field-image-dataset (2016).
16. T. T. Ahonen, *The Insider's Guide to Mobile*, TomiAhonen Consulting (2010). http://www.lulu.com/shop/tomi-t-ahonen/insiders-guide-to-mobile-free-edition/ebook/product-20157379.html
17. T. T. Ahonen, *TomiAhonen Phone Book 2014*, TomiAhonen Consulting 2014 (2016). http://communities-dominate.blogs.com/brands/2014/12/handset-market-and-installed-base-in-2014-the-total-handset-picture-by-communities-dominate-blog.html
18. ITU-T, "ITU-T T.101 International interworking for Videotex services Annex F videotex photographic mode," https://www.itu.int/rec/T-REC-T.101/en (1994).
19. ITU-T, "ITU-T T.120 Data protocols for multimedia conferencing," https://www.itu.int/rec/T-REC-T.120/en (2007).
20. ITU-T, "ITU-T T.417 Information technology—open document architecture (ODA) and interchange format: raster graphics content architectures," https://www.itu.int/rec/T-REC-T.417/en (1993).
21. ITU-T, "ITU-T T.30 Procedures for document facsimile transmission in the general switched telephone network," https://www.itu.int/rec/T-REC-T.30/en (2005).
22. ITU-T, "ITU-T T.563 Terminal characteristics for group 4 facsimile apparatus," https://www.itu.int/rec/T-REC-T.563/en (1996).
23. A. Léger, H. Poulsen, and M. Guglielmo, "Still picture compression algorithm for photovideotex on ISDN," Esprit Technical Week ETW 88 (1988).
24. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.* **C-32**, 90–93 (1974).
25. H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.* **32**(12), 1316–1322 (1984).
26. V. Thomas and J.-L. B. et Michel Hias, "Experiments on visibility thresholds of DCT coefficients," in *Proc. of the Picture Coding Symp.*, Torino, Italy (1988).
27. P. A. Wintz, "Transform picture coding," *Proc. IEEE* **60**, 809–820 (1972).
28. D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE* **40**(9), 1098–1101 (1952).
29. J. Parsons and A. Tescher, "An investigation of MSE contributions in transform image coding schemes," *Proc. SPIE* **66**, 196–206 (1975).
30. ITU, "Information technology—digital compression and coding of continuous-tone still images—requirements and guidelines," ITU-T T.81 (1992).
31. ISO, "Information technology—lossless and near-lossless compression of continuous-tone still images: baseline," ISO/IEC 14495-1:1999 (1999).
32. ITU, "Information technology—lossless and near-lossless compression of continuous-tone still images—baseline," ITU-T T.87 (1998).
33. G. Hudson, H. Yasuda, and I. Sebestyen, "The international standardization of a still picture compression technique," in *IEEE Globecom Conf., Proc. of the Global Telecommunication Conf.*, pp. 1015–1021 (1988).
34. H. Poulsen, R. Vivian, and G. Wallace, "Subjective testing results for still picture compression algorithms for international standardization," in *IEEE Globecom Conf., Proc. of the Global Telecommunication Conf.*, pp. 1022–1027 (1988).
35. A. Léger et al., "Telsat-Prisme: codeur-décodeur d'images fixes de haute qualité," *Commun. Transm.* **4**, 11–12 (1989).
36. J. C. Carlac'h and J. L. Sicre, "TCAD 2D 8x8 forward and inverse DCT chip," ISO/TC97/SC2/WG8 JPEG N146, CCETT (1988).
37. A. Léger, J. Mitchell, and Y. Yamazaki, "Still picture compression algorithms evaluated for international standardization," in *IEEE Globecom Conf., Proc. of the Global Telecommunication Conf.*, pp. 1028–1032 (1988).
38. W. B. Pennebaker, J. Vaaben, and J. Mitchell, "Progressive coding in the emerging JPEG compression standard," in *Proc. of the Picture Coding Symp.*, Boston, pp. 26–28 (1990).
39. A. Léger, T. Omachi, and G. Wallace, "The JPEG still picture compression algorithm and its applications," *Opt. Eng.* **30**, 947–954 (1991).
40. B. Niss, "Prediction of AC coefficients from DC values revisited," JPEG n147 (1988).
41. ISO, "Information technology—digital compression and coding of continuous-tone still images: requirements and guidelines," ISO/IEC 10918-1:1994 (1994).
42. T. Acharya and P. S. Tsai, *JPEG2000 Standard for Image Compression*, John Wiley and Sons, New Jersey (2005).
43. G. Hudson and D. Tricker, "PICA—photovideotex image compression algorithm," in *ESPRIT'86: Results and Achievements*, Directorate General XIII, Ed., Telecommunication, Information, Industries and Innovation of the European Commission, pp. 413–422, Elsevier, North Holland (1987).
44. R. Geldreich, "JPEG image compressor and decompressor classes in two C++ source files," https://code.google.com/archive/p/jpeg-compressor/
45. "Next-generation image compression (JPEG XL) revised draft call for proposals," https://jpeg.org (21 February 2018).
46. Y. T. Chan, *Wavelet Basics*, Kluwer Academic Publishers, Norwell, Massachusetts (1995).
47. "Applications for JPEG 2000," https://jpeg.org/jpeg2000/applications.html.
48. T. Richter, A. Artusi, and T. Ebrahimi, "'JPEG XT' a new family of JPEG backward-compatible standards," *IEEE Multimedia Mag.* **23**(3), 80–88 (2016).
49. G. J. Sullivan et al., "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012).
50. V. Sze, M. Budagavi, and G. J. Sullivan, *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Integrated Circuits and Systems, Springer, Cham (2014).
51. P. Hanhart et al., "Subjective quality evaluation of the upcoming HEVC video compression standard," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Joint Collaborative Team on Video Coding (JCT-VC), Geneva (2013).
52. P. Kerr, "AV1 image file format (AVIF)," Commit Snapshot, https://AOMediaCodec.github.io/av1-avif (26 June 2018).

Graham Hudson was manager of multimedia development at BT Laboratories, UK. He was responsible for developments in videotex, teleconferencing, and broadcast television. He graduated with his BSc (Hons) degree in electrical and electronic engineering from City University, London, and is a fellow of the Institute of Engineering and Technology. He was project leader of the European PICA project and founding chairman of JPEG. He has published papers on videotex and image coding standardization.

Alain Léger (ex CCETT FT R&D France) (DR habil., PhD, Ing.) was head of image coding (1980–1993), head at FT direction of research on "Knowledge Processing" (1995–2007). He led the ISO International ADCT-JPEG ad-hoc group during algorithm competition (1985–1988). He is the member or coordinator of many AI-related FP6-FP7 EU projects (1998–2006); he received a best paper award at IEEE Web Intelligence Conference (2008) and a second award AFIA (Lécué 2007). He is an associated Pr in KRR (Univ. Mines Saint-Etienne) (2002–2010).

Birger Niss (ex KTAS, NNIT A/S) received his MSc degree in physics and astronomy. He is a key contributor to the JPEG-1 algorithm together with Jørgen Vaaben (KTAS). After leaving KTAS, he continued to work with advanced software development, including JPEG toolkits, in various Danish companies. Currently, he holds a position as senior software engineer (NNIT A/S). He is an external examiner at the Technical University in Denmark, primarily the DTU Space Institute. He is an author of several astronomical papers and coauthor of Danish encyclopedias on data processing.

István Sebestyén (PhD, Dipl Ing) (Secretary general of Ecma International in Geneva, Switzerland, since 2007) was working as chief engineer and director of standards for various Siemens Communication Divisions in Munich, Germany. In the period 1987–2000, he was CCITT SGVIII special rapporteur on New Image Communication, later ITU-T SG8 rapporteur on Common Components for Image Communications. As such, these groups were the ITU-T "parents" of the JPEG Group.

Jørgen Vaaben received his MSC and PhD degrees from the University of Aarhus, postdoc at University of Copenhagen. He was then employed by the computer center at the Technical University of Denmark. He was employed by Copenhagen Telephone Company (KTAS), where he worked in close cooperation with Birger Niss on jpeg-1. After that he was employed by Autograph Intl. The company has been developing and selling JPEG tools with special emphasis on scientific and medical applications.