EUROPEAN TELECOMMUNICATION STANDARD

ETS 300 177

September 1992

Source: ETSI TC-TE

Reference: DE/TE-01006

ICS: 33.020, 33.040.40

Key words: Videotex, Photographic Syntax

Terminal Equipment (TE); Videotex Photographic Syntax

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

New presentation - see History box

Contents

Blank page

Foreword

This European Telecommunication Standard (ETS) has been prepared by the Terminal Equipment (TE) Technical Committee of the European Telecommunications Standards Institute (ETSI) in order to specify a new common data syntax for transmitting photographic images to be used by Videotex terminal equipment.

This ETS is part of a series of ETSs which describe the Videotex presentation layer data syntax.

This ETS defines a data syntax to be used for conveying photographic data in a Videotex environment. The necessary tools are provided for the transfer of photographic data, typically from a Videotex Host to a Videotex terminal. This data syntax is equally applicable to either storage or communication applications and is independent of physical device or transmission media.

This ETS does not deal with the visible appearance of the displayed pictures, however all the necessary source image information is provided to make the proper physical adaptation at the receiving side. The specification of post-processing techniques is left to the implementors and is, therefore, outside the scope of this ETS.

More precisely, this ETS defines the syntax and semantics of image data and image attributes for photographic Videotex interchange purposes. In particular, it addresses the various aspects of image dimensionality such as spatial, amplitudinal, temporal and spectral content, it provides some basic tools for positioning photographic images within a defined area, it also addresses the structure and organisation of the data and uses standardised compression schemes. In particular, the ISO-Joint Photographic Experts Group (JPEG) compression algorithm [13], based on the Discrete Cosine Transform (DCT) is used. In this ETS the algorithms or compression techniques themselves are not described, references are provided.

The intention of this ETS is primarily to provide Videotex application developers with a sufficient set of photographic transfer tools which are independent of the equipment used to implement/provide them. This ETS is intended to support operations on and display of, various classes of images from a wide variety of imaging applications. However, to ensure that compatibility can be achieved between various Videotex services supporting photographic mode, some realistic and specific characteristics are chosen and defined in the Clause on profiles (Clause 11). In the future, other selections might be made allowing the definition of new recommended profiles.

This ETS closely follows the concepts and coding techniques as described in ISO/IEC 9281, Part 1 [11] for the identification of pictorial information and for switching between picture environments and coding systems according to ISO 2022 [10].

Blank Page

1 Scope

This ETS specifies the data syntax to be used by Videotex services for conveying photographic data.

In general, it applies to the interchange of photographic data via storage or transmission media.

The ETS is applicable to Videotex terminals connected to various types of telecommunication networks including; a Public Switched Telephone Network (PSTN), a Public Switched Packet Data Network (PSPDN) or an Integrated Services Digital Network (ISDN). For the ISDN case, these terminals will typically support "ISDN Syntax-based Videotex" (see ETS 300 079 [3]).

The syntax allows for some private extensions beyond the transmission of still pictures. For example, a provision has been made for the transmission of a "difference" image to allow a slow scan television type of application.

2 Normative references

This ETS incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references subsequent amendments to or revisions of, any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] ETS 300 072: "Terminal Equipment (TE); Videotex presentation layer protocol, Videotex presentation layer data syntax".

[2] ETS 300 076 (1992): "Terminal Equipment (TE); Videotex, Terminal Facility Identifier (TFI)".

[3] ETS 300 079 (1991): "Integrated Services Digital Network (ISDN); Syntax-based Videotex, End to end protocols".

[4] CCITT Recommendation F.300 (1988): "Videotex service".

[5] CCITT Recommendation T.101 (1988): "International interworking for Videotex".

[6] CCITT Recommendation H.261 (1988): "Common intermediate format".

[7] CCITT Recommendation T.51 (1988): "Coded Character sets for Telematic services".

[8] CCITT Recommendation T.61 (1988): "Character repertoire and coded character sets for the international teletex service".

[9] CCIR Recommendation 601-1 (1986): "Encoding Parameters of Digital Television For Studios".

[10] ISO 2022 (1986): "Information Processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques".

[11] ISO/IEC 9281-1 (1990): "Information technology - Picture coding methods- Part 1: Identification".

[12] ISO/IEC 9281-2 (1990): "Information Technology - Picture coding methods - Part 2: Procedure for registration".

[13] ISO/IEC DIS 10918/CCITT Recommendation T.81: "Digital compression and coding of continuous-tone images".

[14] ISO 646 (1990): "Information processing - ISO 7-bit coded character set for information interchange".

[15] ISO 6937-1 (1991): "Information processing - coded character sets for text communication: Part 1 General introduction".

[16] ISO 2375 (1991): "Data Processing - Procedure for registration of escape sequences".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purpose of this ETS the following definitions apply.

Aspect ratio: the ratio of the width to the height of a rectangular area, such as the defined display area.

Attribute: a particular property or quantity defined in this syntax and described by a number of parameters (e.g. the source picture specifications).

Baseline: the basic sequential DCT-based encoding and decoding process specified in ISO/IEC DIS 10918 [13].

Continuous tone image: an image comprised of data which exhibits a first order continuity in the analogue domain and requires, when digitised, more than one bit to describe each sample contained in one or more of its components (monochrome (grey scale) or colour pictures) e.g., a monochrome image needs at least 6 bits/picture element (64 grey levels) to appear "continuous" to the eye.

Data syntax I: term used within CCITT for one of the recommended world-wide Videotex data syntaxes originating from the Japanese Character And Pattern Telephone Access Information Network (CAPTAIN) system.

Data syntax II: term used within CCITT for one of the recommended world-wide Videotex data syntaxes originating from the European CEPT Videotex syntax.

Data syntax III: term used within CCITT for one of the recommended world-wide Videotex data syntaxes originating from the North American Presentation Layer Protocol Syntax (NAPLPS).

Defined Display Area (DDA), Physical (Physical Defined Display Area (DDA)): a rectangular area of the full screen area where photographic data, text etc. shall be displayed.

Defined Display Area (DDA), Logical (Logical DDA): a unit square, the length of all sides being one unit, co-ordinates being defined as fractions of unity (unit screen concept). The origin is coincident with the bottom left corner of the physical DDA and one side is coincident with the longest side of the physical DDA.

Defined Display Area, Source (Source DDA): the virtual display space where the source image was encoded and which is to be mapped for display either to the full screen area or to the physical DDA.

Discrete Cosine Transformation (DCT): see ISO/IEC DIS 10918 [13].

Full screen area: the part of a display screen where photographic data can be displayed, it normally means a display with no borders.

Forward DCT: see ISO/IEC DIS 10918 [13].

Inverse DCT: see ISO/IEC DIS 10918 [13].

Image attribute: the various properties of a continuous tone image described by a number of parameters.

Image data: the data which represents a continuous tone image in digital form, it contains photographic header data and photographic data.

JPEG compression algorithm: a general term for referring to any one of the possible modes of encoding defined in ISO/IEC DIS 10918 [13].

Normalised co-ordinate: a co-ordinate specified in a device independent co-ordinate system, normalised to some range (usually to 1).

Parameter: a quantity which is described using one or more sub-parameters.

Photographic data: pixel based pictorial information usually in compressed digital form; the data includes any tables which are necessary to decode and decompress the data.

Photographic data syntax: the rules by which the photographic header data and the photographic data are formatted.

Photographic header data: coded data containing the values of the attributes and parameters used for describing the photographic image.

Photographic image: a continuous tone image, e.g. an image represented with 256 shades of grey.

Photographic mode: the mode of operation of a Videotex terminal while it is receiving photographic header data and photographic data.

Photographic profile: a collection of attributes with parameters set to a given value to represent a type of source image and define a mode of photographic image coding and photographic image transfer.

Photo Videotex: neologism used for Videotex photographic mode.

Physical device: any tangible piece of equipment (e.g., personal computer, display monitor, etc).

Pixel, picture element: it is the minimum displayable element of an image (see ISO/IEC DIS 10918 [13]).

Pixel density: expresses the number of pixels per physical unit (e.g. pixels/mm) in the horizontal and vertical directions.

Post-processing technique: image processing which is performed (e.g. for display) after the image has been decoded and decompressed.

Spatial resolution: definition of the size of the image, expressed in the number of pixels per horizontal line and the number of lines per image.

Storage media: a type of physical means to store data.

Sub-parameter: a quantity to which a value can be assigned.

NOTE: Example of the use of attribute, parameter and sub-parameter. Consider the ISDN, it has the following *attributes*, it is digital and supports data transmission with a speed of 64 kbit/s. For the ISDN the *parameter* network speed is assigned the value 64 kbit/s. Two *sub-parameters* can represent this quantity, "numerical speed" i.e., 64 and "unit of measure" i.e., kbit/s.

Spectral content: a physical quantity that measures the frequency content i.e., the amplitude and phase of each frequency contained in a given physical item. It generally refers to the fourier analysis. For the Discrete Cosine Transformation (DCT) it relates to the amplitude of each DCT basic function (Discrete Cosine) or sub-image. In simple terms, for the image, it gives an idea on the "level of detail" of the source image.

Transmission media: the type of physical means to transport data (e.g., coaxial cable, optical fibre, radio link).

Videotex system: text communication system, hardware and software, with the capability of running a Videotex service or application.

Videotex application: see CCITT Recommendation F.300 [4].

Videotex host computer: see CCITT Recommendation F.300 [4].

Videotex service: see CCITT Recommendation F.300 [4].

Videotex terminal: see CCITT Recommendation F.300 [4].

NOTE: Reference should also be made to ISO/IEC DIS 10918 [13] for definitions of terms related to image coding which are used in this ETS but not included in this Clause.

3.2 Symbols and abbreviations

For the purpose of this standard the following abbreviations and symbols apply.

AM	Alphamosaic
C_B	Chrominance colour difference, Blue
C_R	Chrominance colour difference, Red
CAPTAIN	Character And Pattern Telephone Access Information Network
CCIR	International Radio Consultative Committee
CCITT	International Telegraph and Telephone Consultative Committee
CD	Committee Draft
CEPT	Commission Européenne des Postes et Télécommunications
CIF	Common Intermediate Format
CMI	Coding Method Identifier
CMYK	Cyan, Magenta, Yellow, Black
DCT	Discrete Cosine Transformation
DDA	Defined Display Area
DPCM	Differential Pulse Code Modulation
E'_R, E'_G, E'_B	Primary (analogue) signals (red, green, blue)
ESC	ESCape
ETS	European Telecommunication Standard
ETSI	European Telecommunications Standards Institute
F	"Final character" registered by ISO 2022 registration authority
IEC	International Electrotechnical Committee

ISDN	Integrated Services Digital Network
ISO	Organisation for International Standardization
JPEG	Joint Photographic Experts Group
LI	Length Indicator
NAPLPS	North American Presentation Layer Protocol Syntax
P1 to P5	Compatible Photographic Profiles, numbered 1 to 5
P_{priv}	Private Photographic Profile
PCD	Picture Coding Delimiter
PCE	Picture Control Entity
PDDA	Physical Defined Display Area
PDE	Picture Data Entity
PE	Picture Entity
PI	Picture Identifier
PM	Picture Mode
PSPDN	Public Switched Packet Data Network
PSTN	Public Switched Telephone Network
RGB	Red, Green, Blue
SDDA	Source Defined Display Area
US	Unit Separator
VPCE	Videotex Presentation Control Element
Y	Luminance

4 Overview

The photographic data syntax allows for the transmission and display of an image consisting of individually defined Picture elements (Pixels) with many grey/colour levels. Digital signal processing techniques are used to compress the image for storage and transmission.

The photographic data syntax allows, in principle, the specification of a wide variety of different modes of photographic images for use in Videotex systems. However, to cater for the foreseen requirements of users and to aid compatibility, some recommended application profiles are specified, they are based on CCIR Recommendation 601, part 1 [9] and CCITT Recommendation H.261 [6], which describes the Common Intermediate Format (CIF).

This photographic data syntax is basically unidirectional; the photographic data shall be transported from a Videotex Host computer to a Videotex terminal using the syntax.

In Videotex photographic mode the size of a file containing an encoded photographic image can be rather large. One picture may be transmitted in several PEs. The use of several PEs could facilitate the termination of the transmission of a picture by the user.

This draft ETS clearly separates the functionalities, the syntax (the formatting rules i.e., structuring the data) and the coding rules (i.e. coding of the header, coding of the photographic data, etc.). The ISO 9281, Part 1 [11] syntax structure used to introduce the photographic header is encoded using 7 bits only. For coding in an 8 bit transfer environment the most significant bit (bit 8) is set to zero.

The photographic data is encoded and represented using 8 bits. For transmission, allowance is made for both a 7 bit and an 8 bit channel.

5 ISO/IEC 9281, Part 1 syntax and switching structure

5.1 Overall switching of coding environment

ISO/IEC 9281, Part 1 [11] describes a technique for identifying picture coding methods. The Videotex photographic mode is one of the picture coding methods identified by ISO/IEC 9281, Part 1 [11]. The diagram in figure 1 gives an overview of the relationship between the Videotex data syntaxes and ISO/IEC 9281, Part 1 [11] picture coding environments.

From an ISO 2022 [10] environment, a Videotex data syntax can be explicitly entered using an ESC 2/5 F code. This is also the mechanism used for entering from an ISO 2022 environment [10] into an ISO/IEC 9281 [11] environment. The F code ("final byte") is allocated and registered, according to ISO 2022 [10], by the registration authority ISO 2375 [16]. According to ISO 2375 [16], Annex B, the Videotex data syntaxes are regarded as "coding systems different to that of ISO 2022 [10]". The F codes are 4/3 for CCITT data syntax I, 4/4 for CCITT data syntax II and 4/1 for CCITT data syntax III.

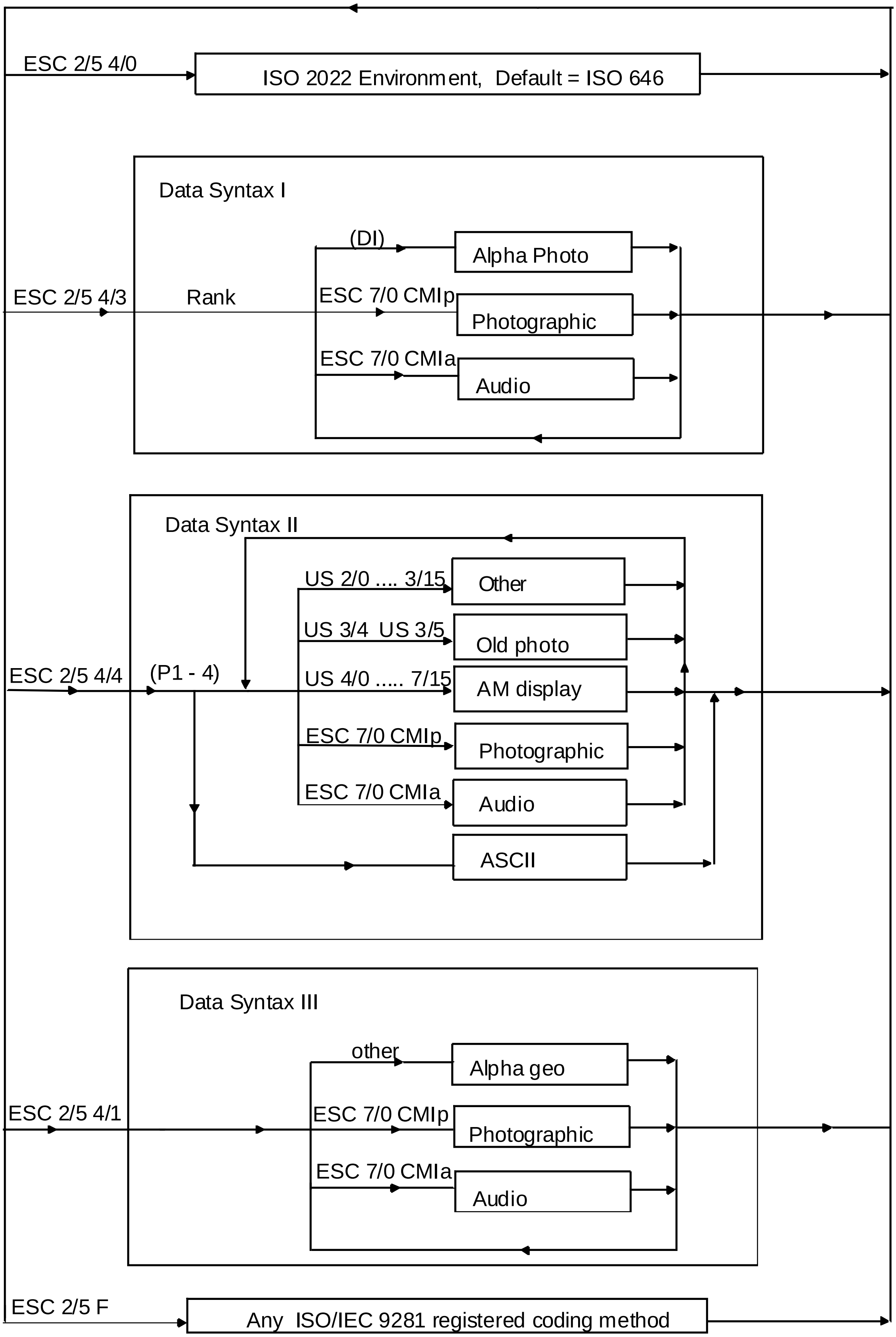


Figure 1: Global switching mechanism

Key to figure 1:

DI:	is data syntax I specific
P:	is a profile in data syntax II
F:	is a final code assigned by the ISO 2022 [10] registration authority
CMI_a:	is any CMI for Videotex audio data
CMI_p:	is any CMI for Videotex photographic data
Rank:	is data syntax I specific

Since a Videotex terminal usually begins operation, by default, in one of the data syntaxes, it shall not be mandatory to first send an ESC 2/5 F code (F is 4/1, 4/3 or 4/4). The diagram shows how these codes can be used to switch a Videotex terminal supporting more than one data syntax from one data syntax to another.

5.2 Switching into the photographic mode

A Videotex terminal operating within one of the data syntaxes (i.e., a coding system other than that described by ISO 2022 [10]) can enter the ISO/IEC 9281, Part 1 [11] environment of the photographic mode according to their own rules. In the case of Videotex for switching into the ISO/IEC 9281, Part 1 [11] environment of the photographic mode the Picture Code Delimiter (PCD) of the first Picture Element (PE) is used. The Coding Method Identifier (CMI) is used to distinguish between picture coding methods. In the case of Videotex this shall be, for example, a distinction between audio and photographic data.

5.3 ISO/IEC 9281, Part 1 syntax structure

The high level structure of the syntax is as defined in ISO/IEC 9281, Part 1 [11].

In the following description of the syntax 8 bit coding is assumed, thus the word "byte" is used with bit 8 set to zero. The coding described in ISO 9281, Part 1 [11] is also valid in a 7 bit environment. In this case the word "byte/octet" shall be interpreted as meaning "7 bit byte" and the most significant bit, bit 8, shall not be used.

The structure of the coding is as follows:

PE	::=	PCE PDE	
PCE	::=	PCD CMI LI	
PCD	::=	01/11 07/00	
CMI	::=	PM PI	
PM	::=	02/03	video photo coding (Videotex photographic mode)
PI	::=	<04/00 - 07/15>	
LI[1)]	::=	x111 1111 <byte$_1$> <byte$_2$>....<byte$_n$>	
<byte$_k$>	::=	x10D DDDD (k=n)	
	|	x11D DDDD (1 =< k < n)	

x indicates don't care.
D indicates binary number 0 or 1

[1] ISO/IEC 9281-1 [11], subclause 5.2.7 should be consulted for a description of the use of the Length Indicator.

Each piece of information, in this case encoded image data, is encoded as one or more Picture Entities (PEs). A PE consists of Picture Control Entity (PCE) which is followed by the actual data packed into a Picture Data Entity (PDE).

<table>
<tr><td colspan="5">PE Picture Entity</td></tr>
<tr><td colspan="4">PCE Picture Control Entity</td><td rowspan="3">PDE
Picture Data Entity</td></tr>
<tr><td rowspan="2">PCD
Picture Coding Delimiter</td><td colspan="2">CMI Coding Method Identifier</td><td rowspan="2">LI
Length Indicator</td></tr>
<tr><td>PM
Picture Mode</td><td>PI
Picture Identi-fier</td></tr>
</table>

Figure 2: Structure of a Picture Entity

NOTE 1: In Videotex photographic mode the size of a file containing an encoded photographic image can be rather large. One picture may be transmitted in several PEs. The use of several PEs could facilitate the termination of the transmission of a picture by the user.

The Picture Control Entity (PCE) consists of a Picture Coding Delimiter (PCD) and a Coding Method Identifier (CMI) followed by a Length Indicator (LI).

The Picture Coding Delimiter (PCD) is a fixed sequence of two octets: 01/11 07/00.

NOTE 2: 01/11 is ESC.

The Coding Method Identifier (CMI) consists of a Picture Mode (PM) octet, followed by a Picture Identifier octet (PI). For photo Videotex the PM is 02/03 as registered by ISO 9281, Part 2 [12]. The PI octet specifies the specific coding technique being used. The PI may take any value from the range 04/00 - 07/15. For the JPEG coding technique 04/00 shall be used.

5.3.1 General use of the Length Indicator (LI)

A length indicator technique is used to specify the number of bytes in the PDE which follows a PCE. The number of bytes is encoded using the LI code as defined in ISO/IEC 9281, Part 1 [11]. Image data can be divided into blocks and sent using a number of PEs. After a LI has expired the terminal shall check all incoming bytes. If transmission of the picture is still in progress the next bytes shall be ESC 7/0 CMI_p (start of a new photographic PE). If transmission of the picture has terminated the next byte shall be an ESC or US code (data syntax II).

5.3.2 Use of the Picture Identifier (PI) code

The PI shall take one of the values but, at present, it shall only be:

04/00 for the ISO/IEC DIS 10918 [13], Part 1 (JPEG) algorithm;

04/01 (using CCITT Recommendation T.4 compression method);

04/02 (using CCITT Recommendation T.6 compression method);

04/03 (using ISO/IEC CD 11544/CCITT Recommendation T.82 compression method).

6 Coding of the Picture Data Entity (PDE)

6.1 Introduction

As described in the previous clause a PE as defined by ISO 9281, Part 1 [11] consists of a, Picture Control Entity (PCE) followed by a PDE. The coding of the PCE, as it shall be used for the Videotex photographic data syntax, is defined in Clause 5. The PDE contains either photographic header data or photographic data. A mechanism is described below which shall be used for identifying the type of data in a PDE.

6.2 PDE data content identification mechanism

The first byte after the LI, i.e. the first byte of the PDE (denoted as PDE_1), identifies the type of information which follows. This byte can have the following values:

5/0 for the photographic header data with more photographic header data to follow;

5/1 for the last PE of photographic header data;

5/2 for the photographic data with no more data to follow;

5/3 for the last PE of photographic data.

The transmission of the photographic header shall not be interrupted.

The data following the PCE with the first byte of the PDE set to 5/0 is the photographic header data. The photographic header data shall be sent in one or more PEs, the transmission of the photographic header shall not be interrupted. The actual photographic data shall be sent in a number of PEs, with each PE having the PDE_1 value set to 5/2 except the last one which shall have the PDE_1 value set to 5/3. The LI is used in all the PEs (as illustrated in figure 3).

In a situation where an interruption of the photographic image (caused probably by user interaction) one of the two following scenarios shall occur:

Scenario 1:

In some Videotex systems, the PE being sent to the terminal shall be completely sent. The next PE sent to the terminal shall have the PDE_1 value set to 5/3 to indicate the end of the data. The PDE contains only one byte (length indicator = 1). This solution is preferred.

Scenario 2:

Where "scenario 1" is not supported by the Videotex system, the terminal shall not require an explicit notification that the transmission of the picture has terminated. The use of the PE with PDE_1 set to 5/3 for aborting the rest shall not be sent. The terminal shall, at the latest, resynchronise on receipt of the next header (5/0).

NOTE 1: If possible, the speed of the network should be taken into account when defining the number of bytes in a PDE.

NOTE 2: The receipt of the last PE indicated to the terminal by the PDE_1 value 5/3, could be used to indicate to the user that the transmission of the picture has been completed. For example, a message indicating this could be displayed for the user.

A series of pictures with an already stated header may be sent without redefining the attributes.

1	PDE 1	PDE 1 = 5/0	Header Data
2	PCE	PDE 1 = 5/1	Header Data
3	PCE	PDE 1 = 5/2	Photographic Data
4	PCE	PDE 1 = 5/2	Photographic Data
5	PCE	PDE 1 = 5/3	Photographic Data

Example 1: A complete sequence of PEs (one picture) with the Header Data in two PEs (1 and 2) and the Photographic Data in three PEs (3, 4 and 5).

1	PCE	PDE 1 = 5/1	Header Data
2	PCE	PDE 1 = 5/2	Photographic Data
3	PCE	PDE 1 = 5/2	Photographic Data
4	PCE	PDE 1 = 5/2	Photographic Data
5	PCE , LI=1	PDE 1 = 5/3	No data

Example 2: A complete sequence of PEs (one picture) with the Header Data in one PE (1), the Photographic Data in three PEs (2, 3 and 4). The last PE contains no data.

Figure 3: Examples for sequences of PEs

7 Photographic header

7.1 Introduction

The photographic header is the specific part of the photographic data syntax that specifies the parameters of the various photographic image attributes that support the needs of photographic Videotex transfer and display. In particular, it shall support various combinations of dimensionality, organisation and compression. For example, it could support static bilevel (document) images, static colour (e.g., luminance (Y), chrominance colour difference, red (C_R), chrominance colour difference, blue (C_B)) images, iconic images etc., with standardised compression schemes. Extensions for non-standardised use (e.g., animated images) shall be supported.

In the description of the attributes and parameters contained in this Clause no indication is given as to whether the parameters are optional or mandatory. In the Clause on profiles (Clause 11) the classification of optional or mandatory is assigned to each parameter for each profile.

Some of the parameters of the photographic image attributes may appear to duplicate those already provided in the signaling part of the photographic data (e.g., according to the header of the data stream structure of ISO/IEC DIS 10918 [13]). They are included in the photographic header to give the terminal access to this information without having to decode the JPEG data stream. If a parameter is duplicated the values shall be identical, otherwise the result is unpredictable.

The syntax allows for the omission of some parameters from a particular attribute of the photographic header. Depending upon the value of the parameter status attribute, a parameter shall take either the value of its last appearance in a photographic header or the value assigned as its default. Default values are defined in Clause 10.

In the following text, the header structure is first globally introduced, then each part of the structure is described in terms of attributes. The coding rules used and their application to the defined header shall be described in Clause 8.

7.2 Header structure

The general structure of the photographic header is shown in figure 4. The structure is composed of the following main attributes:

- parameter status attribute;
- picture display attributes;
- source picture attributes;
- source signal attributes;
- source coding algorithm attributes;
- transmission channel attributes.

Table 1 shows how the various attributes are described using a number of parameters.

The selection of the attribute classes follows a "layered structure", in a manner similar to the OSI-philosophy.

The highest "layer" gives an indication of whether default values should be used. On the "layer" below the picture display attributes are responsible for embedding the transmitted photographic image into the Videotex application environment (i.e., display). On the "layer" below, by using the source picture attributes, definitions are made for how and with which physical properties (such as image size) the physical source picture was captured. On the next "layer" the source signal attributes define in detail the specification of the source signal to be compressed. On the next "layer" the source coding algorithm attributes specify the picture compression technique which was used to create a compressed data stream. Finally, on the lowest "layer", transmission channel attributes, some specific aspects for the transmission of the compressed data stream are defined (e.g. the use of 7 or 8 bit transparent mode).

Table 1: Overview of the header functionalities

Photographic Header:	
Attributes	Parameters
Parameter Status Attribute <PSA>	Reset to Default <RTD>
Picture Display Attributes <PDA>	Full Screen Display <FSD>
	Source Aspect Ratio <ASR>
	Photo-area Location <LOC>
	Photo-area Size <PAS>
	Picture Placement <PPL>
	Clear Photo-area <CPA>
Source Picture Attributes <SPA>	Source Picture Comments <PCT>
	Source Picture Dimensions <PDS>
	Source Pixel Density <PID>
	Source Sweep Direction <SWD>
	DC Images <DCI>
Source Signal Attributes <SSA>	Source Component Description <SCD>
	Source Component Data Precision <CDP>
	Source Component Order <CMO>
	Source Level Assignment <LAS>
Source Coding Algorithm Attributes <SCA>	JPEG Coding Mode <JPG>
	Encoding Table Management <ETM>
	Application Marker Codes Assignment <AMA>
Transmission Channel Attributes <TCA>	Translation Mode Encoding <TME>

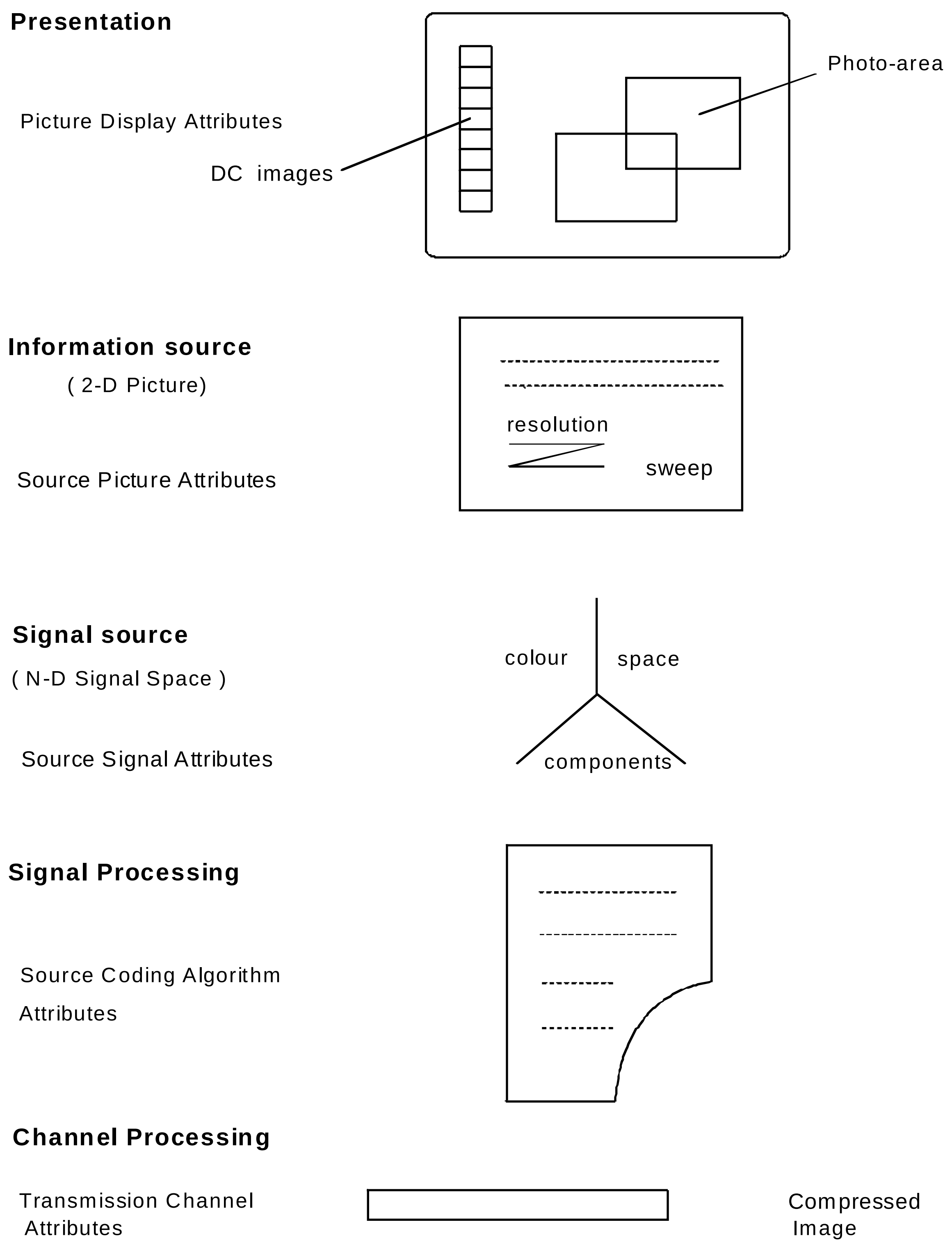


Figure 4: Photographic header structure (excluding the parameter status attribute)

7.3 Header functionalities

The name of each attribute and parameter is followed by an abbreviation enclosed in "<>" brackets. The abbreviation is used in the coding clause of this ETS (Clause 8) but is introduced here to make cross-referencing between the two Clauses easier.

7.3.1 Parameter Status Attribute <PSA>

A single parameter is used to define the status of all parameter values currently in use and the status of the new values being sent. Since this parameter shall always be sent first, to avoid any possible ambiguities, all parameters shall be sent in the order given in Clause 8 (Coding rules).

7.3.1.1 Reset To Default <RTD>

If this parameter is set to "true" all parameter values shall be reset to their default value. If a parameter value is subsequently sent, the value being sent supersedes the default value.

If this parameter is set to "false" all parameter values shall assume the value to which they were set in a previous header. If, as yet, they have not been set they shall assume the default value. If a parameter value is subsequently sent the value sent supersedes the "in use" value.

A single sub-parameter is used to encode the reset to default parameter:

<dev> default_value when "true" indicates that default values shall be used.

7.3.2 Picture Display Attributes <PDA>

The picture display attributes are defined by a set of parameters which describe some aspects of the display of a photographic source image at an application level (e.g., the placement of the photographic image).

Other aspects of the display which are strictly related to a given Videotex terminal are outside the scope of this standard and are left open to implementors. For example, rendering includes the alignment of the source image resolution to the physically displayed image or the alignment of the colour model of the source image to the colour capabilities of the physical display. Although such rendering is out of the scope of this ETS, some guidelines are provided in Annex B (informative).

In order to be independent of display hardware constraints, definitions in the following text are made with normalised co-ordinates. Normalised co-ordinates are derived from the unit screen concept, which defines co-ordinates as fractions of unity. The unit screen concept is illustrated in figure 5.

Figure 5 shows the full screen area which is the part of the screen where photographic data can be displayed (e.g. full active lines and all active lines).

Inside the full screen area the physical Defined Display Area (physical DDA) is a rectangular area used for displaying images, text etc. Pictorial information within the physical DDA shall always be displayed, while pictorial information outside of the physical DDA, but inside the full screen area, may be displayed.

The Logical Defined Display Area (logical DDA) is a unit screen i.e., it is a square where the origin (0,0) is coincident with the bottom left corner of the physical DDA and whose one side is coincident with the longest side of the physical DDA.

The logical DDA may be partly outside the full screen area. Moreover, if it is possible to draw anywhere on the logical DDA, only the portion of the logical DDA coincident with the physical DDA shall be visible.

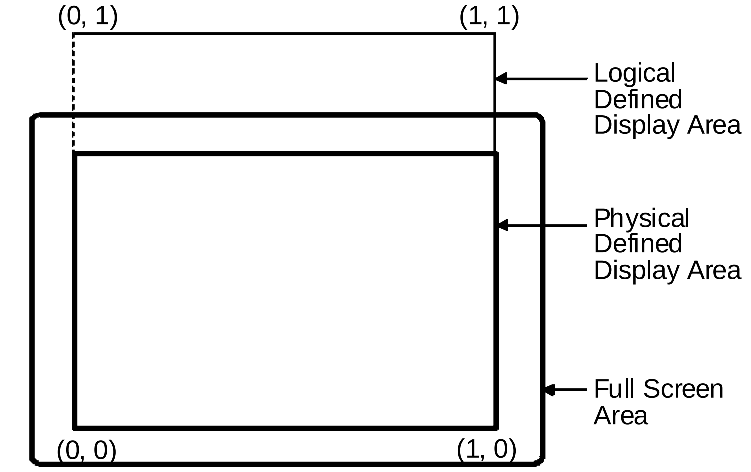


Figure 5: Unit screen concept

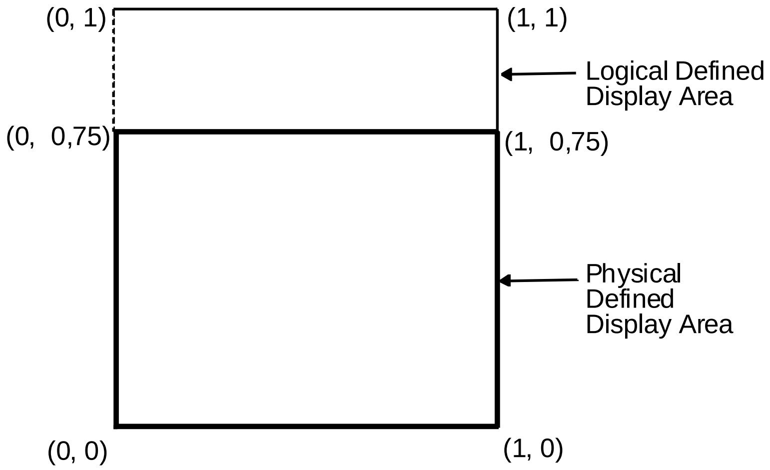


Figure 6: Source Defined Display Area (SDDA) with a 4/3 aspect ratio mapped to the physical DDA

The virtual display space where the source image was encoded is referred to as the Source Defined Display Area (SDDA). This virtual space is mapped according to the value of the full screen display parameter to the full screen area or to the physical DDA.

The ratio of the width to the height of the SDDA is the source aspect ratio. One typical example, as shown in figure 6, is the commonly used 4/3 aspect ratio of some Videotex terminals.

The picture display attributes are used to define a rectangular area in the logical DDA, this area is referred to as the photo-area. For photovideotex applications this area shall be filled with a photographic image. If the photographic image is larger than the photo-area then the part of the image outside the photo-area shall not be displayed. If the photographic image is smaller than the photo-area then part of the photo-area shall be made transparent and therefore filled by the full background colour.

The following space allocation parameters are used to encode the photo-area:

- full screen display;
- source aspect ratio;
- photo-area location;
- photo-area size;
- picture placement;
- clear photo-area.

7.3.2.1 Full Screen Display <FSD>

The full screen display parameter indicates that the SDDA shall be mapped to full screen area. All other parameters and sub-parameters are still functional. The full screen display parameter is defined using one sub-parameter:

<ful> full_screen_area   display full screen.

If the value of the sub-parameter is "true" the SDDA shall be mapped to the full screen area.

NOTE: If this parameter is used, there is no guarantee for the physical display of the image outside the Physical Defined Display Area (PDDA) and there is no support for the alignment of the photographic plane to the alphamosaic and if applicable the alpha-geometric planes.

7.3.2.2 Source Aspect Ratio <ASR>

The source aspect ratio parameter defines the ratio of the width over the height of the SDDA. This parameter is encoded using two integer values:

<araw> source_width,   the width of the source;
<arah> source_height,   the height of the source.

The source aspect ratio is obtained by dividing the value of <araw> by the value of <arah>. This is illustrated in figure 6.

7.3.2.3 Photo-area LOCation <LOC>

The photo-area location parameter defines the position of the bottom left hand corner of the photo-area in normalised co-ordinates relative to the origin of the logical DDA.

This parameter is encoded using two sub-parameters, as shown in figure 7, the horizontal and vertical normalised co-ordinates:

<loch> photo-area_location_horizontal,   the normalised horizontal co-ordinate;

<locv> photo-area_location_vertical,   the normalised vertical co-ordinate.

7.3.2.4 Photo-Area Size <PAS>

The photo-area size parameter defines the dimensions of the photo-area in normalised lengths. This parameter is encoded using two sub-parameters, the width and height of the photo-area:

<sizw> photo-area_width,   the normalised width of the photo-area;

<sizh> photo-area_height,   the normalised height of the photo-area.

This is illustrated in figure 7.

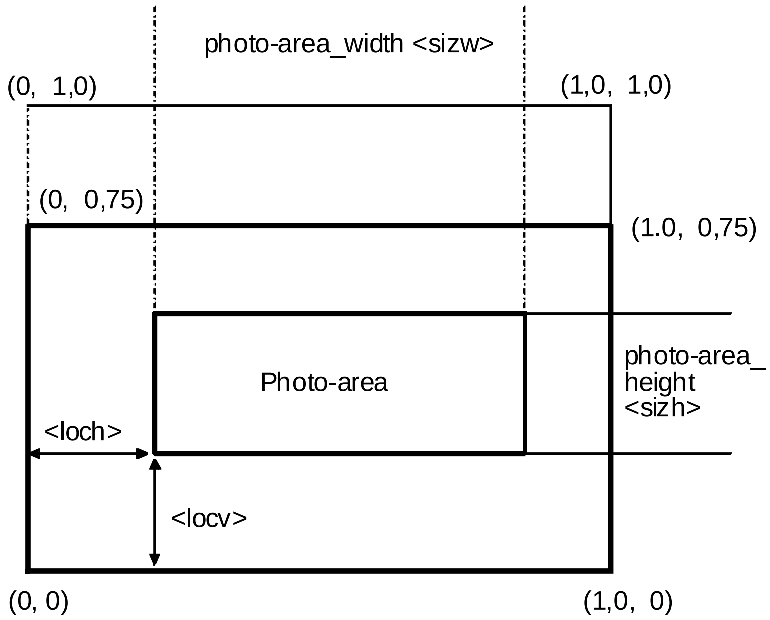


Figure 7: Unit screen co-ordinates and the photo-area

7.3.2.5 Picture PLacement <PPL>

The picture placement parameter defines how the source picture shall be positioned in the photo-area.

Using this parameter the position of the photographic image relative to the photo-area is fixed. It is envisaged that this parameter can be used to ensure that the most important part of the photographic image is correctly positioned in the photo-area.

Two points are identified, one on the photographic source image and the other in the photo-area. These two points are defined to be coincident and hence the position of the photographic image in the photo-area is fixed.

This parameter is defined by two sets of sub-parameters:

- reference_picture-point, on the source image;
- offset_picture-point, in the photo-area.

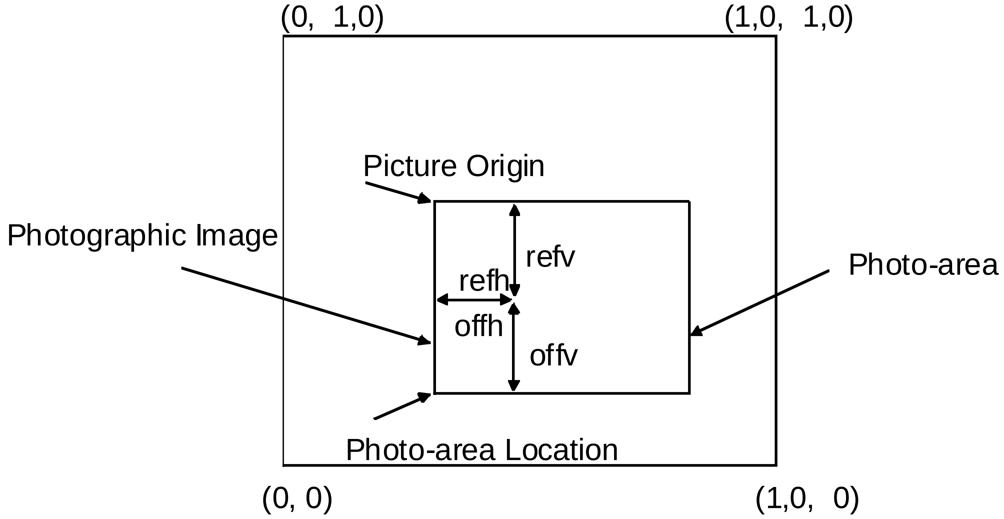


Figure 8a: The Photographic Image and the photo-area are the same size.

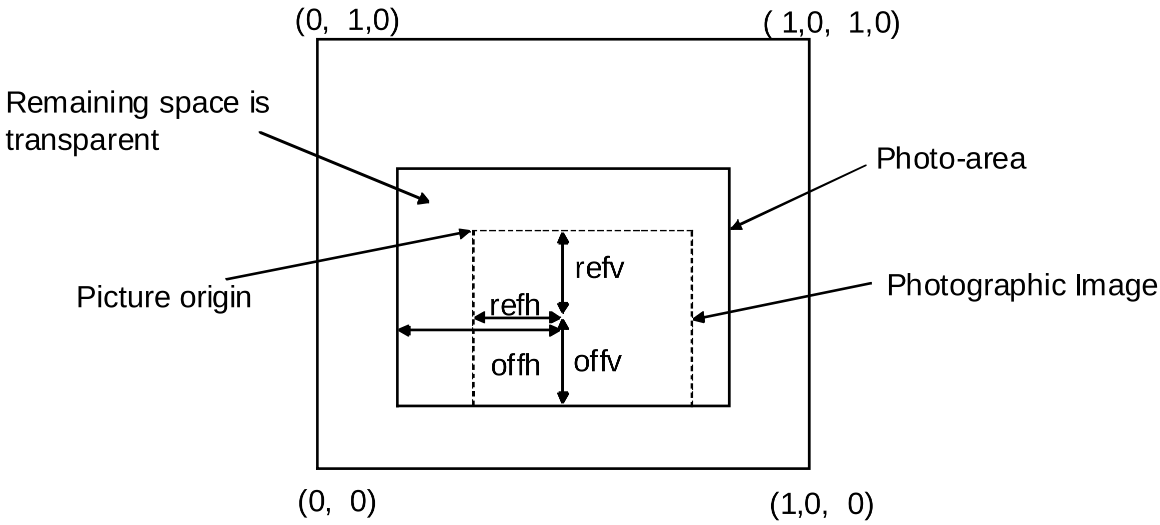


Figure 8b: The Photographic Image is smaller than the photo-area.

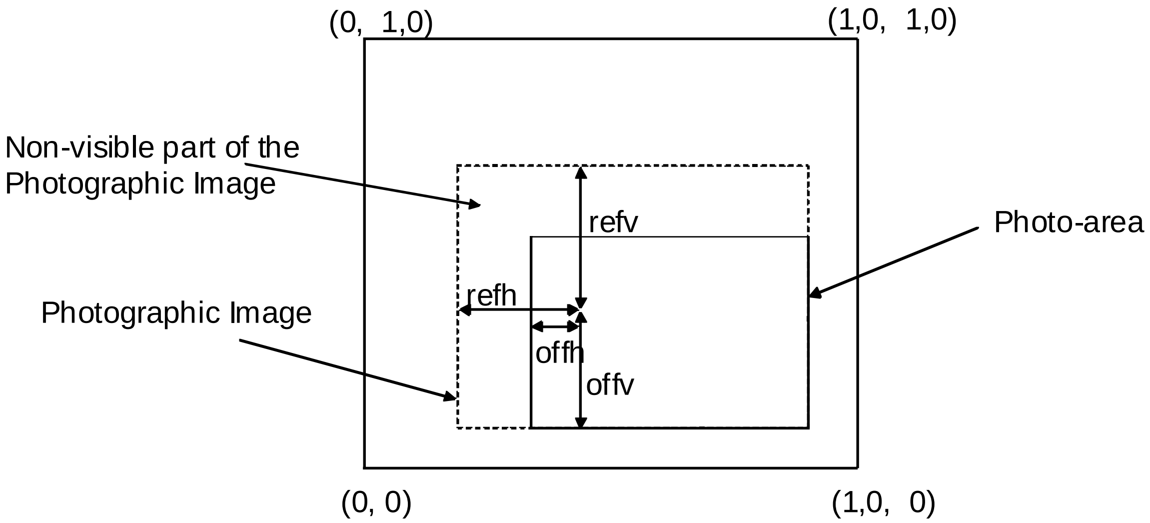


Figure 8c: The Photographic Image is larger than the photo-area

The reference_picture-point refers to the spatial reference on the source image. It is expressed as the number of pixels, in both the vertical and horizontal directions, from the top left-hand corner of the source image. For reasons of compatibility with the coding technique, the origin of the picture is defined as the top left-hand corner. The offset_picture-point refers to a point relative to the bottom left hand corner of the photo-area. The units of measurement relative to the photo-area are normalised co-ordinates (as defined for the whole logical DDA). See the illustrations in figure 8.

In some situations the photo-area is not completely filled by the photographic image. In such cases it is assumed that this unfilled area is transparent.

The following sub-parameters are used to encode the picture placement parameter:

<refh> ref_picture-point_horiz,	the number of pixels from the picture origin to the reference_picture-point in the horizontal direction;
<refv> ref_picture-point_vert,	the number of pixels from the picture origin to the reference_picture-point in the vertical direction;
<offh> offset_photo-area_horiz,	the horizontal co-ordinate of the photo-area offset_picture-point;
<offv> offset_photo-area_vert,	the vertical co-ordinate of the photo-area offset_picture-point.

The photo-area offset point and the picture reference point are coincident points.

In the default situation the picture origin shall be the default coincident point. This implies taking, for the sake of simplicity, the top left hand corner for both the photo-area and the photographic image.

7.3.2.6 Clear Photo-Area <CPA>

A specified photo-area is cleared to transparent by the Clear Photo-Area parameter. The photo-area shall be specified using the parameters photo-area location and photo-area size. If the photo-area parameters location and size, are not specified, the default shall be the full DDA. The Clear Photo-Area parameter shall be encoded using one sub-parameter:

<cle> clear_photo_area,	clear the photo-area.

If the value of the sub-parameter is "true" the photo-area shall be cleared.

NOTE: When the transparent colour does not exist in the terminal, the relevant background colour is used.

7.3.3 Source Picture Attributes <SPA>

The Source Picture Attributes are defined by a set of parameters which describe the spatial composition of the photographic image. All of the parameters are related to the environment where the source image was encoded.

7.3.3.1 Source Picture Comments <PCT>

The source picture comment parameter indicates a text string as part of the photographic data introduced by the JPEG marker code COM (X`FFFE'). This text string is, however, not part of the alpha-mosaic mode. If this parameter is used the receiving terminal shall not forced to (but may) display this text. Even if this text is displayed, any kind of "fall back" due to a limited character set is allowed (e.g., diacritical marks received are not displayed).

The source picture comments parameter includes character coded information about the source image, which normally does not form part of the Videotex frame. It may contain information about for example; the title of the source image, the author of the source, date of source production, copyright of source, size of the source, etc. Due to the simplicity and diversity of the possible uses of this option, no special logical record structuring of the text string is defined by this standard, it is left open for the application provider.

This parameter has two sub-parameters: code_size <csz> and code_set_identifier <cid>.

Code_size <csz> specifies in terms of the number of bits the length of each code. It can have the values 7, 8 or 16. If <csz> is 7 bits, bit 8 of a byte is not observed.

Code_set_identifier <cid> identifies the character set used (e.g. ISO 646 [14], CCITT Recommendation T.51 [7], CCITT Recommendation T.101 [5]) to code the text in the JPEG comment. (The default is CCITT Recommendation T.51 [7]).

The <PCT> parameter has two sub-parameters:

<csz><cid>

where

<csz> code_size — enumerated type (e.g. 7, 8, 16)

<cid> code_identifier — enumerated type:

- non-standard
- ISO 646 [14]
- CCITT Recommendation T.51 [7]
- CCITT Recommendation T.61 [8]
- data syntax I
- data syntax II
- data syntax III

EXAMPLE: In a 7 bit environment for the transmission of the text it can be recommended to assign to the code positions 2/00 to 7/15 the character codes of CCITT Recommendation T.51 [4] (or equivalent ISO 6937 parts 1 and 2 [15]), which are the CCITT and ISO recommended set of characters for the transmission of Latin characters (and which are a superset of the ETS 300 072 [1] primary and supplementary sets of graphic characters). All codes within the above code range should be recognised and accepted by the decoder, but their use (e.g. conversion into an internal code) is not within the scope of this standard.

For the transmission of control characters in CCITT Recommendation T.51 [5] the code range 0/00 to 1/15 is reserved. The code assignment in the above range should follow the primary control set of ETS 300 072 [1]. All decoders supporting this function should recognise and accept characters of the above code range. It is recommended that as a minimum, with the exception of the necessary code extension characters of ISO 2022 [10] all possible characters of CCITT Recommendation T.51 [7] should be recognised. The codes APR (0/13) (functionally equal to carriage return) and APD (0/10) (functionally equal to line feed) should be supported.

Thus:

<PCT><csz> "7-bit" <cid> "CCITT Recommendation T.51"

(the exact coding of the Sub-parameters is given in Clause 8)

An example of the use of this parameter is provided in Annex D (informative).

7.3.3.2 Source Picture Dimensions <PDS>

The dimensions of the source image are encoded with the source picture dimensions parameter. The dimensions of the image are given in terms of the number of pixels in the horizontal direction and the number of pixels in the vertical direction. The source picture dimensions parameter specifies the dimensions in terms of the highest resolution component. Two sub-parameters are therefore needed to encode this parameter:

<nph> num_pixels_horiz, the number of pixels in the horizontal direction;

<npv> num_pixels_vert, the number of pixels in the vertical direction.

7.3.3.3 Source PIxel Density <PID>

The source pixel density parameter expresses for the general case the number of pixels per unit (e.g., inch or cm) of resolution, in both the vertical and the horizontal direction. For standardised formats this parameter shall make a direct reference to a standardised set, typically a CCIR Recommendation.

The source pixel density parameter is a key piece of information in the general field of image applications. The source pixel density information is necessary to ensure that the photographic data encoded in the source environment is processed and displayed to give the intended aspect ratio. Also the source photographic data can be converted to the resolution of the local display (or printing device), with a minimal introduction of artifacts (e.g. aliasing). However, it is worth noting that to ease the development of compatible applications, a few compatible families of Pixel Densities are recommended in the Clause on profiles (Clause 11).

In the general case, the pixel density is expressed as the ratio of two integers, as for example 150/1. When the pixel density is expressed using decimal numbers, the proper rational value shall be found, for example 10,5 pixel/mm shall be represented by 21/2 pixel/mm. The parameter takes one of two forms:

<stf> standardised_form, enumerated type:
4:2:2 625 lines
4:2:2 525 lines
2:1:1 625 lines
2:1:1 525 lines
CIF
Not applicable

or

<p_h_num> pixels_per_unit_horizontal_numerator, integer;
<p_h_den> pixels_per_unit_horizontal_denominator, integer;
<p_v_num> pixels_per_unit_vertical_numerator, integer;
<p_v_den> pixels_per_unit_vertical_denominator, integer;
<unt> unit, enumerated (e.g. pixels/cm, pixels/inch, pixels/mm).

For a multi-component picture, the <PID> specification shall be made in sequence for each component.

EXAMPLE:

<PID>($p\_h\_num_1$, $p\_h\_den_1$, $p\_v\_num_1$, $p\_v\_den_1$, unt)
<PID>($p\_h\_num_2$, $p\_h\_den_2$, $p\_v\_num_2$, $p\_v\_den_2$, unt).

<PID>(150/1, 150/1, inch)
<PID>(75/1, 75/1, inch)
<PID>(75/1, 75/1, inch)

The Source Component Description parameter defines the order of the components.

NOTE: The parameters Source Picture Dimensions and Source Picture Density describe the source image. They are included to provide the terminal with the information which may be required for scaling or clipping of an image. If the source image is derived from a bit map, only the Source Picture Dimensions parameter contains information which is useful to the terminal. The value of the "<stf> standardised form" in this case is set to not applicable.

7.3.3.4 Source SWeep Direction <SWD>

The Source Sweep Direction parameter specifies the direction in which the image is built up. There are two possible values of the sweep direction of a row:

top to bottom, bottom to top.

There are two possible values of the sweep direction of a line:

left to right, right to left.

The sweep direction is completely specified using two sub-parameters.

<sdir> the sweep direction of a row;
<sdil> the sweep direction of a line.

7.3.3.5 DC Images <DCI>

Small images 1/64th of the size of the whole image can be obtained by decoding and extracting the DC values. A small image can also be coded as a normal JPEG image with only the DC values used. For a description of the term, DC value, reference should be made to Annex A (informative).

The DC Image parameter is encoded with one sub-parameter:

<dcv> dc_value, an indication of whether DC values only are in use.

This parameter can have two values: "true" or "false". When the value is set to "true", decoding shall take place to display DC sized images. When the value is set to "false", decoding shall take place to produce a full size image.

7.3.4 Source Signal Attributes <SSA>

The Source Signal Attributes describe the components which form the image and the relationship between the components. The parameters used to define the Source Signal Attributes allow for the use of a large variety of colour models, interleave structures etc.

7.3.4.1 Source Component Description <SCD>

An image can be formed from one or more source components. A monochrome image requires one component, a colour image needs at least three. The different sets of components which can be used are each assigned a separate code. The number of components used is defined by the set. The order of the components which is required by some other parameters is defined by this parameter.

Another important source picture characteristic is whether the image is block or component interleaved. The definition of that characteristic shall be given in ISO/IEC DIS 10918/CCITT Recommendation T.81 [13] and not in this ETS.

For example, if the set is RGB (Red, Green, Blue) then the number of components is three. The component sets which can be selected are:

- R,G,B
- Y,C_R,C_B
- C,M,Y,K
- Y

The sub-parameter

<com> component

is used to send the enumerated value identifying the component set.

7.3.4.2 Source Component Data Precision <CDP>

The Source Component Data Precision parameter specifies, for each image component, how many bits are required to represent each component sample associated with a pixel. For example, a luminance component with 256 grey levels needs to be represented using 8 bits.

The sub-parameter expressing the precision in terms of bits/component is

<cpt> component_data_precision.

For a multi-component picture, the <CDP> specification shall be made in sequence for each component.

EXAMPLE:

<CDP>(cpt)<CDP>(cpt)..

<CDP>(8)<CDP>(8)<CDP>(8)

7.3.4.3 Source Component Order <CMO>

The Source Component Order parameter specifies the order in which the components are sent to the terminal. The order is specified in terms of the order defined by the source component description parameter.

For example if the parameter source component description has the value "RGB" then the order is defined as red 1st, green 2nd, blue 3rd. If, however, the components are sent with the order green, blue, red the value of source component order shall be 2,3,1. This parameter only defines the order of component transmission, for all other parameters (i.e. <PID>, <CDP>, <LAS>) concerned with the order of the components refer to the source component description parameter.

The sub-parameter

<cor> component_order

shall be sent once for each component.

7.3.4.4 Source Level Assignment <LAS>

One explicit example of the use of this parameter is provided by the CCIR Recommendation 601, part 1 [9]. In this CCIR Recommendation, reference levels are used to fix the values (within the range 0 to 255) used to define certain colours of the source. For example, for a luminance signal the values used for black (16) and the value used for white (235) are fixed.

The Source Level Assignment parameter can be assigned a value which is either an enumerated type or an integer type.

The only enumerated type which is defined indicates that the level assignment follows the CCIR Recommendation 601, part 1 [9].

If the level assignment is not done according to CCIR Recommendation 601 [9], integer values shall specify the lowest and highest level per component. The full range of the colour component is defined by the source component data precision parameter.

The source level assignment parameter is encoded with one of two sets of sub-parameters:

<fix> fixed_assignment	an enumerated value refers to a standardised level assignment

or

<low> lowest_level — lowest level of the component

<hi> highest _level — highest level of the component

7.3.5 Source Coding Algorithm Attributes <SCA>

The source coding algorithm attributes identify two basic aspects:

1) the technique being used to compress and encode the pixel data; and

2) the application oriented coding functions.

There are a large number of techniques available, however only those techniques which have been standardised shall be included here.

The application oriented coding function is the term used to describe a few functions left open by a given coding technique for handshaking with the application.

Only techniques developed by ISO/IEC DIS 10918 [13] and some simple extensions shall be considered. All the parameters described here refer to these techniques.

Annex A (informative) contains a description of the compression techniques which have been developed by ISO/IEC DIS 10918 [13] and should be consulted in order to fully understand the meaning of these parameters.

Not all combinations of the parameters are possible. Table 2 summarises all the permitted combinations.

7.3.5.1 JPEG Coding Mode <JPG>

The JPEG coding mode parameter sets in an abbreviated form the basic modes of operation of the JPEG coding technique. This is typically specified in the ISO/IEC DIS 10918 [13], but via a marker code structure and in an expanded form. The intention is to give at the application level, before decoding the image, a compact description of the JPEG coding modes which have been used to compress the images conveyed by the photographic data syntax. This shall be in accordance with the profile definitions.

In the following, all the JPEG modes are described. However, in the photovideotex profiles (see Clause 11) only a subset is used.

Hierarchical mode

Hierarchical mode allows the resolution of the image to be increased in stages. From the user's point of view, first a low resolution image is completely displayed (either sequentially or progressively). After the completion of the first image a second iteration increases the resolution of the image, such iterations may be repeated several times.

The increase in resolution can be an increase of spatial resolution by a factor of two in either the horizontal or the vertical direction (or both). It can also be a refinement of image quality at a given spatial resolution.

The possible values are:

- hierarchical;
- non-hierarchical.

Table 2: JPEG coding mode

Hierarchical Mode	No						Yes					
Type Of Algorithm	DCT				DPCM spatial		DCT				DPCM spatial	
Build Up Mode	Sequ		Prog.		Sequ.		Sequ.		Prog.		Sequ.	
Entropy Coding Technique	H	A	H	A	H	A	H	A	H	A	H	A

Sequ. Sequential image build up
Prog. Progressive image build up
H Huffman coding
A Arithmetic coding

NOTE: The leftmost column corresponds to the modes of operation of the minimum capability requested for all DCT based systems referred to as the BASELINE system.

Type of algorithm

Two types of compression algorithm are used in the ISO/IEC DIS 10918 [13]. One is based on the Discrete Cosine Transform (DCT) which is lossy. The other is a specialised lossless algorithm based on a spatial prediction technique (DPCM).

The possible values are:

- DCT;
- DPCM.

Build-up mode

An image can be reproduced on the screen of the user's terminal using either a sequential or a progressive build-up mode.

In the sequential mode each image component is built up to full precision in one pass - one scan per component in non-interleaved mode (e.g. Y, C_R, C_B, coded in separated scan) or one scan per group of components in interleave mode (e.g. Y in one scan, and C_R, C_B, interleaved in one scan). If the algorithm type is DPCM the image is built up pixel by pixel, if the algorithm is of DCT type the image is built up from pixel blocks (a pixel block being 8 x 8 pixels).

In the progressive mode the entire image is built up from a number of scans, each successive scan improving the entire image quality. The build-up of a scan can occur in pixel blocks or on a pixel by pixel basis according to the type of algorithm, as for the sequential mode.

In both cases the order of colour build up is defined by the source component order parameter.

Two values are possible:

- sequential;
- progressive.

Entropy coding technique

One of two entropy coding techniques can be applied to the processed data in order to achieve further compression.

These two techniques are:

- Huffman Coding;
- Arithmetic Coding.

A single sub-parameter <cmp> is used to encode the values assigned to the parameter <JPG>. The use of this sub-parameter shall be defined in Clause 8.

7.3.5.2 Encoding Table Management <ETM>

Depending on the type of compression technique which has been selected different tables are needed to decode the image. The JPEG compression algorithm provides the tools for the transmission of the table data and for the attachment of the tables to the components of the image. However, the "JPEG algorithm" leaves to the application the task of properly managing all tables. Typically, at the application level a library of entropy tables (Huffman) and/or quantisation tables may be maintained and managed. To achieve this functionality the photographic data syntax contains three , sub-parameters: Table_type, Table_id and Table_status. It is important to note that the number of tables in use at one time (in a coder or decoder) is constrained as described in ISO/IEC DIS 10918 CCITT Recommendation T.81 [13]. At the application level the location of the tables, i.e. whether they are the storage area of the "in use environment" or in the storage area of the application, is managed. The tables are transferred as part of the JPEG data stream. Considering that the JPEG data stream and the <ETM> parameters are sent separately, a rule is needed to associate an occurrence of an <ETM> parameter with a particular table. The relation between the two is simply time-dependent, this means that the first <ETM> parameter sent shall correspond to the first table transferred in the JPEG data stream, the second to the second, etc.

For specific applications using the same set of custom tables for all subsequent pictures, it shall be ensured that the downloading of the tables has been successful.

Table_type <ttp>:

Three types of table are used: quantisation tables, Huffman tables and conditioning tables. Quantisation tables are needed for the lossy compression technique. Huffman tables are needed when Huffman coding has been selected for the entropy coding technique. When Table_type is set to Huffman it shall refer to the set of tables, the AC Huffman table and the DC Huffman table of a particular image component. Conditioning tables are needed when arithmetic entropy coding has been selected for the entropy coding technique.

Two values are possible for the sub-parameter <ttp>:

- Quantisation;
- Huffman.

NOTE: For arithmetic condition tables no valid subparameter exists since the Start Of Image (SOI) marker would reset the arithmetic tables to default.

Table_id <tid>:

The Table_id sub-parameter is needed to uniquely identify the tables to be used within a given application. In the case of Huffman tables, a single value of Table_id shall identify both DC and AC tables of a given image component. The number of tables of each type is theoretically not limited at the application level, but is limited for practical reasons in the terminal implementation. The number of tables of each type, used at any one time, is referred to as the "in-use environment" and is directly related to the profile definitions (Clause 11). The coding of this sub-parameter provides a mechanism for assigning the tables to a specific area of the terminal's "in-use environment". This sub-parameter shall take the form of an integer.

The sub-parameter

<tid> identifier of the table

shall be used to uniquely identify the table.

Table_status <tst>:

The table_status shall be assigned the following values: "load default table", "use the current table" or "table to be transferred".

The value "load default table" indicates that the decoder shall load the default tables which are permanently stored in the terminal.

The value "use the current table" indicates that the decoder shall use the tables which are currently in use, these could be default tables or the tables last sent.

The value "table will be transferred" indicates that new tables shall be transferred as part of the transparent data.

The sub-parameter

<tst> status of the table

shall be used to reflect the status given by the application to the table. The following three values are possible:

- load table default;
- use the current table;
- table will be transferred.

7.3.5.3 Application Marker codes Assignment <AMA>

The JPEG algorithm has reserved a marker code space for signalling application-dependent functionalities, not already provided by the application syntax. This gives the application some flexibility to adapt and to extend the functionalities defined in this photographic data syntax to not fully specified or as yet undefined applications.

As one example, the use of the still picture JPEG algorithm for animated images (e.g. one image per second) - not to be confused with motion images - is considered as a possible straight forward extension of this photographic syntax. In that case, one application marker code is reserved to signal this extension of the photographic data syntax and to convey the specific additional data required by such an application outside the Videotex service.

Another example, of an option within the Videotex service, is to allow for a special terminal display facility. A colour palette optimized to a given photographic image is embedded into the photographic data, with a specific marker code.

In these particular cases, the JPEG application marker codes shall be used to signal extension of the photographic syntax and to convey the specific additional data required. The formatting of this data is either fully application dependent or is specified in the profiles Clause.

The sub-parameter shall take the form of an enumeration

<mak> application marker code assignment

the value shall be in the range imposed by the JPEG codes "reserved for application segments (APP_n)" : X 'FFE0'- X 'FFEF'

In this standard, the following application marker codes shall be assigned:

The marker code X'FFE0' shall be assigned to the "animated images application".
The marker code X'FFE1' shall be assigned to the "colour palette definition".

7.3.6 Transmission Channel Attributes <TCA>

The Transmission Channel Attributes describe the transmission characteristics of the channel used to transport the photographic data.

7.3.6.1 Translation Mode Encoding <TME>

The Translation Mode Encoding parameter indicates which of the translation modes shall be used to transport the photographic data. The translation mode values shall be allocated as shown in table 4 (subclause 9.2). In order to offer compatibility with seven bit systems a mode supporting seven bit transparency shall be allowed. Due to the inefficiency of 7 bit modes their use is not recommended.

The sub-parameter

<mod> mode_of_translation

indicates which translation mode shall be used to send the photographic data.

8 Coding rules

8.1 Purpose

Coding rules shall be applied to the coding of the photographic header. A uniform structure is achieved by the use of attributes and parameters. A 7 bit coding technique shall be used to facilitate the implementation of the photographic data syntax in a 7 or an 8 bit environment. In general, a byte will be used to encode the 7 bit codes, bit 8 is, however, never used for coding purposes.

8.2 General rules for coding the header

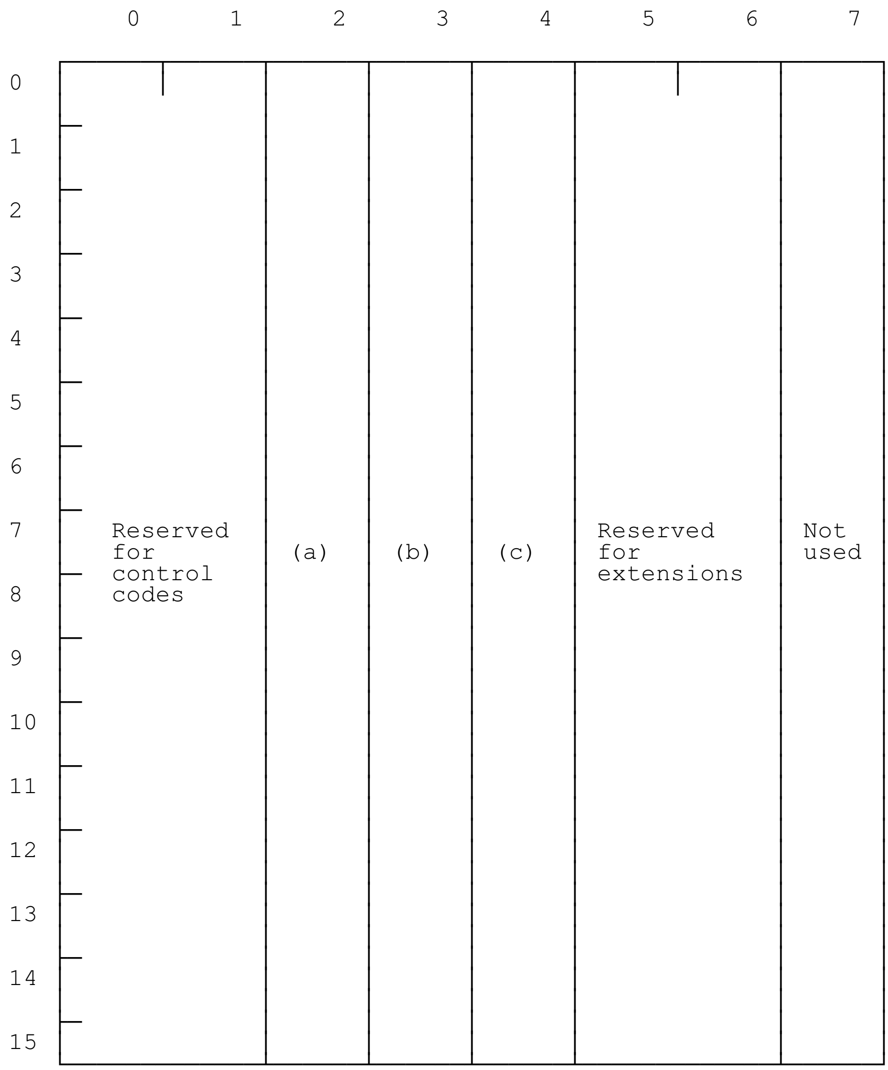
NOTE 1: In order to allow for future possible extensions of the attributes and parameters, a full set of coding is described in this subclause. The current coding of attributes and parameters does not utilise all the possibilities given in this subclause.

The method of coding the various parameters of the various attributes consists of one 7 bit code for the attribute followed by one 7 bit code for the parameter. The attributes and the parameters shall be followed by a number of sub-parameters. A sub-parameter code shall be represented using three fields: **type**, **length** and **value**. The **sub-parameter field type**, shall be encoded using one 7 bit code. The **sub-parameter field length**, consists of a number of 7 bit codes containing the length (number) of 7 bits codes used by the **sub-parameter field value**. The **sub-parameter field value**, consists of a number of 7 bit codes.

The abbreviations defined below are used in subsequent examples to illustrate how attributes, parameters and sub-parameters are coded. Figure 9 shows the structure of a Sub-parameter code.

Sub-parameter Code		
Type	Length	Value

Figure 9: Structure of a sub-parameter code



(a) photographic attributes
(b) photographic parameters
(c) sub-parameter types

Figure 10: Photovideotex code allocation

EXAMPLE 1: Attribute, parameter and sub-parameter

Attribute code (AC)
Parameter code (PC)
Sub-parameter code (SC)

SC = STx + SLx + SVx

Sub-parameter field Type (STx)
Sub-parameter field Length (SLx)
Sub-parameter field Value (SVx)

The following items a) to v) contain the coding rules which shall be observed when coding the attributes, parameters and sub-parameters of the photographic header.

a) in a 7 bit environment bit 8 of a byte shall be set to zero, only 7 of the 8 bits are used for coding purposes;

b) an attribute code shall be followed by a parameter code;

c) a parameter code shall be followed by at least one sub-parameter code;

EXAMPLE 2: [AC PC SC1]

EXAMPLE 3: [AC PC (ST1+SL1+SV1)]

d) if a sub-parameter code or group of sub-parameter codes is sent more than once, each occurrence of the code or group of codes shall be preceded by the relevant parameter code;

EXAMPLE 4: [AC PC SC1 SC2 SCx]

e) "|" means that the sender shall make a choice which sub-parameter or group of sub-parameters shall be sent. In the case of a group of sub-parameters all members of the group of sub-parameters shall always be sent;

f) a new attribute code shall follow directly after the last sub-parameter code of the previous attribute;

EXAMPLE 5: [AC PC SC1 SC2] + [AC PC (ST1+SL1+SV1)] + ...

g) all attributes shall be coded by one byte from column 2 of figure 10;

h) all parameters shall be coded by one byte from column 3 of figure 10;

j) A sub-parameter consists of a **type field**, a **length field** and a **value field**;

k) The **sub-parameter field type**, shall be coded by one byte from column 4 of figure 10;

l) the **sub-parameter field length** shall be coded in the following way:

The length indicating the number of subsequent bytes shall have the following structure, where:

bit 8 = not used;

bit 7 = Extension flag;

bits 6 to 1 = Specify the number of bytes used to encode the sub-parameter value.

Bit 7 of each byte shall be the extension flag. The value shall be specified in binary notation as an unsigned integer using bits 6 to 1 with the weights 2^5, 2^4, 2^3, 2^2, 2^1 and 2^0, respectively. If the value is less than, or equal to, 63 it shall be represented by one byte and the extension flag shall be set to ZERO. If the value is larger than 63 it shall be represented by more than one byte. The most significant part of this value shall be contained in the byte recorded first. The extension flag shall be set to ONE in all bytes except the last where it shall be set to ZERO.

m) The **sub-parameter field value**, shall be encoded using a number of bytes, the number shall be indicated in the **sub-parameter field length**;

n) The following sub-parameter types shall be defined: integer, real, normalised , decimal, enumeration, Boolean and string;

p) Encoding of an integer value:

- the byte or bytes of the **sub-parameter field value**, shall be equal to an integer value, consisting of bits 6 to 1 of the first byte, followed by bits 7 to 1 of the second byte, followed by bits 7 to 1 of each byte in turn up to and including the last byte of the **sub-parameter field value**;

- the value of the integer binary number shall be derived by numbering the bits of the bytes, N representing this number, in the **sub-parameter field value**, starting with bit 1 of the last byte as bit one (N=1) and ending the numbering with bit 6 of the first byte;

- each bit is assigned a numerical value 2^{N-1}, where N is its position in the above numbering sequence. The integer value shall be obtained by summing the numerical values assigned to each bit for those bits which are set to one, excluding bit 7 of the first byte;

- negative values are represented by the two's complement notation, which uses bit 7 of the first byte as a sign bit (0/1 positive/negative values).

  According to the above rules the integer values + 6837 and - 6837 shall be coded:

EXAMPLE 6: Coded representation of a positive integer number + 6837

byte 1 is X0110101

	X	Sign	N=13	N=12	N=11	N=10	N=9	N=8
	X	0	1	1	0	1	0	1
2^{N-1}	X	+	4096	2048	0	512	0	128

byte 2 is X0110101

	X	N=7	N=6	N=5	N=4	N=3	N=2	N=1
	X	0	1	1	0	1	0	1
2^{N-1}	X	0	32	16	0	4	0	1

+ 6837 = 4096 + 2048 + 512 + 128 + 32 + 16 + 4 + 1

EXAMPLE 7: Coded representation of a negative integer number - 6837

byte 1 byte 2

X 1 0 0 1 0 1 0 X 1 0 0 1 0 1 1 X = bit 8 and is not used

q) Encoding of a real value;

NOTE 2: CCITT Recommendation X.209 ("Specification of basic encoding rules for Abstract Syntax Notation One (ASN1)") can be consulted for further clarification on the coding of a real value. The coding of a real value described below, is based on CCITT Recommendation X.209.

A real value shall be specified by the following formula involving three integers (encoded according to item p), M, B and E:

value = M x B^E

M is the mantissa, B the base and E the exponent. M and E may take any values, positive or negative, while B shall take only the value 2, 8 or 16. All combinations of M, B and E are permitted.

The mantissa M shall be represented by a sign S, a non- negative integer value N and a binary scaling factor F, such that:

M = S x N x 2^F, 0 <= F < 4, S = +1 or -1

NOTE 3: The freedom to choose F is provided to enable easier generation of the transfer format by eliminating the need to align the implied decimal point of the mantissa with an octet boundary. The existence of F does not noticeably complicate the task of the decoder. F is only needed with base 8 and 16 representations.

The coding of S, B, F, E, and N shall be done in the following way:

1) bit 8 of the first byte is not used;

2) **S** = bit 7 of the first byte shall be 1 if S is - 1 and 0 otherwise;

3) **B** = bits 6 to 5 of the first byte shall encode the value of the base as shown in table 3;

Table 3: Base Allocation for real numbers

Bits 6 to 5	Base
00	base 2
01	base 8
10	base 16

4) **F** = bits 4 to 3 of the first byte shall encode the binary scaling factor F as an unsigned integer;

5) **E** = bits 2 to 1 of the first byte shall encode the format of the exponent as follows:

if bits 2 to 1 are 00, then the second byte encodes the value of the exponent as a one byte integer value;

if bits 2 to 1 are 01, the second and third bytes encode the value of the exponent as a two byte integer value;

if bits 2 to 1 are 10, then the second, third and fourth bytes encode the value of the exponent as a three byte integer value;

if bits 2 to 1 are 11, then the second byte encodes the number of bytes, say X, (as a positive integer value) used to encode the value of the exponent. The third byte up to including the $(X+3)^{th}$ byte contains the value of the exponent as an integer value.

6) **N** = the remaining bytes encode the value of the integer N as a positive integer number.

EXAMPLE 8: Coding of the positive real number + 0,58984375

byte 1 x0000000 real number header

X	sign	B	B	F	F	E	E
X	0	0	0	0	0	0	0
X	0	Base 2		F = 0		byte 2 = E	

byte 2 x1110011 exponent (two's compliment)

	X	sign	N=6	N=5	N=4	N=3	N=2	N=1
	X	1	1	1	0	0	1	1
2^{N-1}	X	-	32	16	0	0	2	1

E = (32 + 16 + 2 + 1) - 64 = -13

byte 3 x0100101 mantissa (MSB)

	X	Sign	N=13	N=12	N=11	N=10	N=9	N=8
	X	1	1	0	0	1	0	1
2^{N-1}	X	+	4096	0	0	512	0	128

byte 4 x1100000 mantissa (LSB)

	X	N=7	N=6	N=5	N=4	N=3	N=2	N=1
	X	1	1	0	0	0	0	0
2^{N-1}	X	64	32	0	0	0	0	0

N = 4096 + 512 + 128 + 64 + 32 = 4832

M = S x N x 2^F = + 4832

value = M x B^E = 4832 x 2^{-13} = - 0,58984375

NOTE 4: F is set to zero because the base is 2. E is selected to give an integer value for N. The same real number can be coded in different ways.

EXAMPLE 9: Coding of a negative real number - 0,58984375

byte 1	x 1 0 0 0 0 0 0	real number header
byte 2	x 1 1 1 0 0 1 1	exponent (two's compliment)
byte 3	x 0 1 0 0 1 0 1	mantissa (MSB)
byte 4	x 1 1 0 0 0 0 0	mantissa (LSB)

S = -1, N = 4832, F = 0, B = 2, E = -13

r) Encoding of a normalised value:

- the range of normalised values shall be from - 1 to + 1, inclusive;

NOTE 5: In this ETS, the term "normalised" is not used in its classical sense because it includes + 1 or - 1.

- if bit 6 of the first byte is set to one, all other bits shall be set to zero with the exception of the sign bit, bit 7;

- the bytes of the **sub-parameter field value**, shall be equal to a normalised value, consisting of bit 6 to 1 of the first byte, followed by bits 7 to 1 of the second byte, followed by bits 7 to 1 of each byte in turn up to and including the last byte of the sub-parameter field, value;

- the value of the normalised binary number shall be derived by numbering the bits of the bytes in the **Sub-parameter field value**, starting with bit 6 of the first byte as bit one (N=1) and ending the numbering with bit 1 of the last byte;

- each bit shall be assigned a numerical value $1/(2^{N-1})$, where N is its position in the above numbering sequence. The normalised value is obtained by summing the numerical fractions assigned to each bit for those bits which are set to one, excluding bit 7 of the first byte;

- negative values shall be represented by the two's complement, which uses bit 7 of the first byte as a sign bit (0/1 positive/negative values).

EXAMPLE 10: Coding of a positive normalised number + 0,58984375

byte 1	byte 2	
X 0 0 1 0 0 1 0	X 1 1 1 0 0 0 0	X = not used

EXAMPLE 11: Coding of a negative normalised number - 0,58984375

byte 1	byte 2
X 1 0 0 1 1 0 1	X 0 0 1 0 0 0 0

s) Encoding of a decimal value;

- the bytes of the **sub-parameter field value**, shall be the codes 3/0 to 3/9 representing the values 0 to 9, 3/10 representing a decimal point, 3/11 representing a positive number and 3/12 representing a negative number. The sign byte should be first.

EXAMPLE 12: Coding of decimal number - 75,2

byte 1	byte 2	byte 3
X 0 1 1 1 1 0 0 3/12 = negative	X 0 1 1 0 1 1 1 3/7 = 7	X 0 1 1 0 1 0 1 3/5 = 5

byte 4	byte 5
X 0 1 1 1 0 1 0 3/10 = ,	X 0 1 1 0 0 1 0 3/2 = 2

X = not used

t) Encoding of an enumerated value;

- an enumerated value shall be encoded with an integer. The value of the integer indicates which value is assigned to a sub-parameter.

u) Encoding of a Boolean value:

- if the Boolean value is "FALSE" the value of the byte shall be zero;
- if the Boolean value is "TRUE" the value of the byte shall be any non zero number (bit 8 is not used).

v) Encoding of a string value.

- the bytes of the **sub-parameter field value**, shall be the codes of the 7 bit environment of CCITT Recommendation T.51 [7].

EXAMPLE 13: Coding of a string "ETSI"

byte 1	byte 2	byte 3	byte 4
X 1 0 0 0 1 0 1 E	X 1 0 1 0 1 0 0 T	X 1 0 1 0 0 1 1 S	X 1 0 0 1 0 0 1 I

X = not used

8.3 Photographic header code assignment

8.3.1 Attribute codes

Parameter Status Attribute	<PSA>	2/0
Picture Display Attributes	<PDA>	2/1
Source Picture Attributes	<SPA>	2/2
Source Signal Attributes	<SSA>	2/3
Source Coding Algorithm Attributes	<SCA>	2/4
Transmission Channel Attributes	<TCA>	2/5

8.3.2 Parameter codes

8.3.2.1 Parameter Status Attribute

Reset to Default	<RTD>	2/0 3/0

8.3.2.2 Picture display attributes

Full Screen Display	<FSD>	2/1 3/0
Source Aspect Ratio	<ASR>	2/1 3/1
Photo-area Location	<LOC>	2/1 3/2
Photo-area Size	<PAS>	2/1 3/3
Picture Placement	<PPL>	2/1 3/4
Clear Photo-area	<CPA>	2/1 3/5

8.3.2.3 Source picture attributes

Source Picture Comments	<PCT>	2/2 3/0
Source Picture Dimensions	<PDS>	2/2 3/1
Source Pixel Density	<PID>	2/2 3/2
Source Sweep Direction	<SWD>	2/2 3/3
DC Images	<DCI>	2/2 3/4

8.3.2.4 Source signal attributes

Source Component Description	<SCD>	2/3 3/0
Source Component Data Precision	<CDP>	2/3 3/1
Source Component Order	<CMO>	2/3 3/2
Source Level Assignment	<LAS>	2/3 3/3

8.3.2.5 Source coding algorithm attributes

JPEG Coding Mode	<JPG>	2/4 3/0
Encoding Table Management	<ETM>	2/4 3/1
Application Marker Codes Assignment	<AMA>	2/4 3/2

8.3.2.6 Transmission channel attributes

Translation Mode Encoding	<TME>	2/5 3/0

8.3.3 Sub-parameter codes

Integer	4/0
Real	4/1
Normalised	4/2
Decimal	4/3

Enumeration	4/4
Boolean	4/5
String	4/6

8.4 Encoding of photographic header parameters

8.4.1 Parameter Status Attribute: <PSA>

8.4.1.1 Reset To Default <RTD>

<RTD> := 2/0 3/0 <dev>

<dev> := 4/5 <length> <value> (Boolean)

<value> := TRUE; default values
<value> := FALSE; not default values

8.4.2 Picture Display Attributes <PDA>

8.4.2.1 Full Screen Display <FSD>

<FSD> := 2/1 3/0 <ful>

<ful> := 4/5 <length> <value> (Boolean)

<value> := TRUE; full screen
<value> := FALSE; not full screen

8.4.2.2 Source ASpect Ratio <ASR>

<ASR> := 2/1 3/1 <araw><arah>

<araw> := 4/0 <length> <value> (Integer)

<arah> := 4/0 <length> <value> (Integer)

8.4.2.3 Photo-area LOCation <LOC>

<LOC> := 2/1 3/2 <loch> <locv>

<loch> := 4/2 <length> <value> (Normalised)

<locv> := 4/2 <length> <value> (Normalised)

8.4.2.4 Photo-Area Size <PAS>

<PAS> := 2/1 3/3 <sizw> <sizh>

<sizw> := 4/2 <length> <value> (Normalised)

<sizh> := 4/2 <length> <value> (Normalised)

8.4.2.5 Picture PLacement <PPL>

<PPL> := 2/1 3/4 <refh> <refv> <offh> <offv>

<refh> := 4/0 <length> <value> (Integer)

<refv> := 4/0 <length> <value> (Integer)

<offh> := 4/2 <length> <value> (Normalised)

<offv> := 4/2 <length> <value> (Normalised)

8.4.2.6 Clear Photo-Area <CPA>

<CPA> := 2/1 3/5 <cle>

<cle> := 4/5 <length> <value> (Boolean)

<value> := TRUE;	clear photo-area
<value> := FALSE;	do not clear photo-area

8.4.3 Source Picture Attributes <SPA>

8.4.3.1 Source Picture Comments <PCT>

<PCT> := 2/2 3/0 <csz><cid>

<csz> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	7 bit
	0/2	8 bit
	0/3	16 bit

<cid> := 4/4 <length><value> (Enumeration)

<value>:=	0/1	non-standard
	0/2	ISO 646 [14]
	0/3	T.51 [7]
	0/4	T.61 [8]
	0/5	Data Syntax I
	0/6	Data Syntax II
	0/7	Data syntax III

8.4.3.2 Source Picture Dimensions <PDS>

<PDS> := 2/2 3/1 <nph> <npv>

<nph> := 4/0 <length> <value> (Integer)

<npv> := 4/0 <length> <value> (Integer)

8.4.3.3 Source PIxel Density <PID>

<PID> := 2/2 3/2 <stf> |
2/2 3/2 <p_h_num><p_h_den><p_v_num><p_v_den><unt>

<stf> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	4:2:2 625 lines
	0/2	4:2:2 525 lines
	0/3	2:1:1 625 lines
	0/4	2:1:1 525 lines
	0/5	CIF
	0/6	Not applicable

<p_h_num> := 4/0 <length> <value> (Integer)
<p_h_den> := 4/0 <length> <value> (Integer)
<p_v_num> := 4/0 <length> <value> (Integer)

<p_v_den> := 4/0 <length> <value> (Integer)
<unt> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	pixels/inch
	0/2	pixels/cm
	0/3	pixels/mm

8.4.3.4 Source SWeep Direction <SWD>

<SWD> := 2/2 3/3 <sdir><sdil>

<sdir> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	top to bottom
	0/2	bottom to top

<sdil> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	left to right
	0/2	right to left

8.4.3.5 DC Images <DCI>

<DCI> := 2/2 3/4 <dcv>

<dcv> := 4/5 <length> <value> (Boolean)

<value> := TRUE;	DC sized images
<value> := FALSE ;	full sized images

8.4.4 Source Signal Attributes <SSA>

8.4.4.1 Source Component Description <SCD>

<SCD> := 2/3 3/0 <com>

<com> := 4/4 <length> <value> (Enumeration)

<value>:=	0/1	R,G,B
	0/2	Y,C_R,C_B
	0/3	C,M,Y,K
	0/4	Y

8.4.4.2 Source Component Data Precision <CDP>

<CDP> := 2/3 3/1 <cpt>

<cpt> := 4/0 <length> <value> (Integer)

8.4.4.3 Source Component Order <CMO>

<CMO> := 2/3 3/2 <cor>

<cor> := 4/0 <length> <value> (Integer)

8.4.4.4 Source Level ASsignment <LAS>

<LAS> := 2/2 3/3 <fix> |

2/2 3/3 <low><hi>

<fix>:= 4/4 <length> <value> (Enumeration)

<value>:= 0/1 CCIR Recommendation 601 part 1 [9] levels

<low> := 4/0 <length> <value> (Integer)

<hi>:= 4/0 <length> <value> (Integer)

8.4.5 Source Coding Algorithm Attributes <SAC>

8.4.5.1 JPEG Coding Mode <JPG>

<JPG> := 2/4 3/0 <cmp>

<cmp> := 4/0 <length> <value> (integer)

<value> = val1 + val2 = val3 + val4

val1 := 0 non-hierarchical
1 hierarchical

val2 := 0 DCT
2 DPCM

val3 := 0 sequential
4 progressive

val4 := 0 Huffman coding
8 Arithmetic coding

8.4.5.2 Encoding Table Management <ETM>

<ETM> := 2/4 3/1 <ttp><tid><tst>

<ttp> := 4/4 <length> <value> (Enumeration)

<value> := 0/1 Quantisation
0/2 Huffman

<tid> := 4/0 <length> <value> (Integer)

<tst> := 4/4 <length> <value> (Enumeration)

<value> := 0/1 load default table
0/2 use the current table
0/3 table will be transferred

8.4.5.3 Application Marker codes Assignment <AMA>

<AMA> := 2/4 3/2 <mak>

<mak> := 4/4 <length> <value> (Enumeration)

<value> := 0/0 X'FFE0' animated images
0/1 X'FFE1' colour palette definition
0/2 to 0/15 is X'FFE2' to X'FFEF' to be allocated

8.4.6 Transmission Channel Attributes <TCA>

8.4.6.1 Translation Mode Encoding <TME>

<TME> := 2/5 3/0 <mod>

<mod> := 4/4 <length> <value> (Enumeration)

<value>: = 0/0 translation mode 0 (no translation)
0/1 translation mode 1 (no translation except US)
0/2 translation mode 2 (3 in 4 encoding)
0/3 translation mode 3 (shift 8 bits)
0/4 translation mode 4 (shift 7 bits)
0/5 translation mode 5 (no translation except specific characters)

9 Photographic data

9.1 Introduction

The photographic data shall be sent to the terminal in one or more Picture Entities (PEs)[1)]. The structure and organisation of the photographic data is defined by ISO/IEC DIS 10918 [13]. The photographic data is encoded using all 8 bits of a byte. If only 7 bits of a byte are available for transmitting this data a translation mechanism shall be selected.

9.2 Translation modes

A number of translation mechanisms are available for sending the photographic data, they are summarised in table 4.

Table 4: Translation modes

Enumeration value	Translation Mode
0	Mode 0, no translation
1	Mode 1, no translation except US
2	Mode 2, 3 in 4 encoding
3	Mode 3, shift 8 bits
4	Mode 4, shift 7 bits
5	Mode 5, no translation except specific characters

Modes 2 and 4 can only be used when 7 bits of a byte are available for sending data.

Full transparency, mode 0, is without any translation, the coded data is sent using all 8 bits of a byte.

10 Defaults

Default values of parameters and tables shall be resident in a Videotex terminal supporting photographic mode. If a Videotex terminal receives a photographic header with some parameter values not present then

---

1) Clause 5 should be referred to for a description of the use of ISO/IEC 9281 for sending the Photographic Data in a number of PEs.

the current value shall be used. Whether to use default values is indicated by the value of the reset to default parameter. The current value can be either the default value or the value from the last photographic header. Similarly, for Table Data, defaults may be used, the parameter encoding table management shall indicate when default tables shall be used.

10.1 Default values for photographic header attributes

In the following tables default values are provided for all the sub-parameters used in the photographic header.

10.1.1 Default parameter status attribute

A single parameter, reset to default is used to indicate the status of all other parameters.

Table 5: Default parameter status attribute

Parameter	Sub-parameter	Default value	Comment
Reset to Default <RTD>	<dev>	FALSE	do not activate default values

10.1.2 Default picture display attributes

The picture display attributes position the photo-area and the photographic image in the physical DDA. The default size for the photo-area and for the photographic image is the full physical DDA. The default of the picture placement parameter defines the top left hand corner of the photo-area and that of the photographic image as the coincident point.

Table 6: Default picture display attributes

Parameter	Sub-parameter	Default value	Comment
Full Screen Display <FSD>	<ful>	FALSE	not full screen
Source Aspect Ratio <ASR>	<araw> <arah>	4 3	ratio 4:3
Photo-area Location <LOC>	<loch> <locv>	0,0 0,0	
Photo-area Size <PAS>	<sizw> <sizh>	1 0,75	The complete DDA
Picture Placement <PPL>	<refh> <refv> <offh> <offv>	0 0 0 0,75	Top left hand corner of image and photo-area
Clear Photo-area <CPA>	<cle>	FALSE	

10.1.3 Default source picture attributes

The default values of the source picture attributes a CIF image (as given in CCITT Recommendation H.261 [6]).

Table 7: Default source picture attributes

Parameter	Sub-parameter	Default value	Comment
Source Picture Comments <PCT>	<csz> <cid>	0/1 0/3	7 bit T.51
Source Picture Dimensions <PDS>	<nph> <npv>	320 240	minimum resolution image
Source Pixel Density <PID>	<stf>	0/5	CIF
Source Sweep Direction <SWD>	<sdir> <sdil>	0/1 0/1	top to bottom left to right
DC Images <DCI>	<dcv>	FALSE	full size image

10.1.4 Default source signal attributes

The default values of the source signal attributes are for a source image with three components (Y, C_R and C_B) encoded according to CCIR Recommendation 601, part 1 [9].

Table 8: Default source signal attributes

Parameter	Sub-parameter	Default value	Comment
Source Component Description <SCD>	<com>	0/2	Y, C_R, C_B
Source Component Data Precision <CDP>	<cpt> <cpt> <cpt>	8 8 8	precision Y precision C_R precision C_B
Source Component Order <CMO>	<cor>	1 2 3	Y, C_R, C_B
Source Level Assignment <LAS>	<fix>	0/1	CCIR rec. 601, part 1 [9]

10.1.5 Default source coding algorithm attributes

The default values of the source coding algorithm attributes are those values which select the baseline system as defined in ISO/IEC DIS 10981 [13].

Table 9: Default source coding algorithm attributes

Parameter	Sub-parameter	Default value	Comment
JPEG Coding Mode Parameter <JPG>	<cmp>	0	*
Encoding Table Management <ETM>	<ttp> <tid> <tst> <ttp> <tid> <tst>	0/1 0/1 0/2 0/1	quantisation no default load default Huffman no default load default
Application Marker Codes Assignment	<mak>		not used

* values which indicate the baseline system

10.1.6 Default transmission channel attributes

Table 10: Default transmission channel attributes

Parameter	Sub-parameter	Default value	Comment
Translation Mode Encoding <TME>	<mod>	0/0	Full 8 bit transparency

10.2 Default tables

10.2.1 Default quantisation tables

Examples of quantisation tables are given in tables 11 to 16. These quantisation tables have been used with good results on 8 bit/sample Y, C_R, C_B images.

10.2.1.1 Default quantisation tables for CIF images

Table 11: Default luminance quantisation table

16	11	11	11	12	13	16	20
11	12	12	13	14	16	20	25
12	12	13	14	16	18	27	32
12	14	14	16	21	25	34	42
14	16	18	20	35	40	50	58
14	18	25	35	45	55	65	77
16	20	30	40	50	60	75	85
20	25	35	45	60	72	85	90

Table 12: Default chrominance quantisation table

17	17	17	18	30	40	50	60
17	20	27	35	60	70	85	99
17	27	30	50	99	99	99	99
18	35	50	60	99	99	99	99
30	60	99	99	99	99	99	99
40	70	99	99	99	99	99	99
50	85	99	99	99	99	99	99
60	99	99	99	99	99	99	99

10.2.1.2 Default quantisation tables for 2:1:1 images

Table 13: Default luminance quantisation table

16	11	11	11	15	17	22	27
12	12	12	12	19	21	29	33
14	14	14	14	22	27	35	40
14	15	16	17	29	35	47	55
19	19	20	22	46	52	62	69
24	27	32	35	64	69	73	84
49	54	60	64	85	90	100	105
72	80	86	92	100	102	104	105

Table 14: Default chrominance quantisation table

17	17	17	19	30	37	50	62
18	19	20	21	40	58	90	99
24	24	25	26	75	92	99	99
47	50	57	66	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

10.2.1.3 Default quantisation tables for 4:2:2 images

Table 15: Default luminance quantisation table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 16: Default chrominance quantisation table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

10.2.2 Default Huffman tables

Tables 17 to 20 give Huffman tables which have been developed from the average statistics of a large set of video images with 8 bit precision. They can be applied to CCIR 4:2:2 images and CCIR 2:1:1 images.

10.2.2.1 Default Huffman table for luminance DC differences

For sequential mode only.

Table 17: Default luminance DC differences

Category	Codelength	Codeword
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

10.2.2.2 Default Huffman table for chrominance DC differences

For sequential mode only.

Table 18: Default chrominance DC differences

Category	Codelength	Codeword
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

10.2.2.3 Default Huffman table for luminance AC coefficients

For sequential mode only.

Table 19: Default table for AC luminance coefficients

Run/Size	Code length	Code word
0/0	(EOB) 4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101
4/1	6	111011
4/2	10	1111111000
4/3	16	1111111110010110
4/4	16	1111111110010111
4/5	16	1111111110011000
4/6	16	1111111110011001
4/7	16	1111111110011010
4/8	16	1111111110011011
4/9	16	1111111110011100
4/A	16	1111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
5/4	16	1111111110011111
5/5	16	1111111110100000
5/6	16	1111111110100001
5/7	16	1111111110100010
5/8	16	1111111110100011
5/9	16	1111111110100100
5/A	16	1111111110100101
6/1	7	1111011
6/2	12	111111110110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	111111110111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011
7/9	16	1111111110110100

Table 19 (concluded)

7/A	16	1111111110110101
8/1	9	111111000
8/2	15	111111111000000
8/3	16	1111111110110110
8/4	16	1111111110110111
8/5	16	1111111110111000
8/6	16	1111111110111001
8/7	16	1111111110111010
8/8	16	1111111110111011
8/9	16	1111111110111100
8/A	16	1111111110111101
9/1	9	111111001
9/2	16	1111111110111110
9/3	16	1111111110111111
9/4	16	1111111111000000
9/5	16	1111111111000001
9/6	16	1111111111000010
9/7	16	1111111111000011
9/8	16	1111111111000100
9/9	16	1111111111000101
9/A	16	1111111111000110
A/1	9	111111010
A/2	16	1111111111000111
A/3	16	1111111111001000
A/4	16	1111111111001001
A/5	16	1111111111001010
A/6	16	1111111111001011
A/7	16	1111111111001100
A/8	16	1111111111001101
A/9	16	1111111111001110
A/A	16	1111111111001111
B/1	10	1111111001
B/2	16	1111111111010000
B/3	16	1111111111010001
B/4	16	1111111111010010
B/5	16	1111111111010011
B/6	16	1111111111010100
B/7	16	1111111111010101
B/8	16	1111111111010110
B/9	16	1111111111010111
B/A	16	1111111111011000
C/1	10	1111111010
C/2	16	1111111111011001
C/3	16	1111111111011010
C/4	16	1111111111011011
C/5	16	1111111111011100
C/6	16	1111111111011101
C/7	16	1111111111011110
C/8	16	1111111111011111
C/9	16	1111111111100000
C/A	16	1111111111100001
D/1	11	11111111000
D/2	16	1111111111100010
D/3	16	1111111111100011
D/4	16	1111111111100100
D/5	16	1111111111100101
D/6	16	1111111111100110
D/7	16	1111111111100111
D/8	16	1111111111101000
D/9	16	1111111111101001
D/A	16	1111111111101010
E/1	16	1111111111101011
E/2	16	1111111111101100
E/3	16	1111111111101101
E/4	16	1111111111101110
E/5	16	1111111111101111
E/6	16	1111111111110000
E/7	16	1111111111110001
E/8	16	1111111111110010
E/9	16	1111111111110011
E/A	16	1111111111110100
F/0 (ZRL)	11	11111111001
F/1	16	1111111111110101
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

10.2.2.4 Default Huffman table for chrominance AC coefficients

For sequential mode only.

Table 20: Default table for AC chrominance coefficients

Run/Size	Code length	Code word
0/0	(EOB) 2	00
0/1	2	01
0/2	3	100
0/3	4	1010
0/4	5	11000
0/5	5	11001
0/6	6	111000
0/7	7	1111000
0/8	9	111110100
0/9	10	1111110110
0/A	12	111111110100
1/1	4	1011
1/2	6	111001
1/3	8	11110110
1/4	9	111110101
1/5	11	11111110110
1/6	12	111111110101
1/7	16	1111111110001000
1/8	16	1111111110001001
1/9	16	1111111110001010
1/A	16	1111111110001011
2/1	5	11010
2/2	8	11110111
2/3	10	1111110111
2/4	12	111111110110
2/5	15	111111111000010
2/6	16	1111111110001100
2/7	16	1111111110001101
2/8	16	1111111110001110
2/9	16	1111111110001111
2/A	16	1111111110010000
3/1	5	11011
3/2	8	11111000
3/3	10	1111111000
3/4	12	111111110111
3/5	16	1111111110010001
3/6	16	1111111110010010
3/7	16	1111111110010011
3/8	16	1111111110010100
3/9	16	1111111110010101
3/A	16	1111111110010110
4/1	6	111010
4/2	9	111110110
4/3	16	1111111110010111
4/3	16	1111111110011000
4/5	16	1111111110011001
4/6	16	1111111110011010
4/7	16	1111111110011011
4/8	16	1111111110011100
4/9	16	1111111110011101
4/A	16	1111111110011110
5/1	6	111011
5/2	10	1111111001
5/3	16	1111111110011111
5/4	16	1111111110100000
5/5	16	1111111110100001
5/6	16	1111111110100010
5/7	16	1111111110100011
5/8	16	1111111110100100
5/9	16	1111111110100101
5/A	16	1111111110100110
6/1	7	1111001
6/2	11	11111110111
6/3	16	1111111110100111
6/4	16	1111111110101000
6/5	16	1111111110101001
6/6	16	1111111110101010
6/7	16	1111111110101011
6/8	16	1111111110101100
6/9	16	1111111110101101
6/A	16	1111111110101110
7/1	7	1111010
7/2	11	11111111000
7/3	16	1111111110101111
7/4	16	1111111110110000
7/5	16	1111111110110001
7/6	16	1111111110110010
7/7	16	1111111110110011
7/8	16	1111111110110100

Table 20 (concluded)

7/9		16	1111111110110101
7/A		16	1111111110110110
8/1		8	11111001
8/2		16	1111111110110111
8/3		16	1111111110111000
8/4		16	1111111110111001
8/5		16	1111111110111010
8/6		16	1111111110111011
8/7		16	1111111110111100
8/8		16	1111111110111101
8/9		16	1111111110111110
8/A		16	1111111110111111
9/1		9	111110111
9/2		16	1111111111000000
9/3		16	1111111111000001
9/4		16	1111111111000010
9/5		16	1111111111000011
9/6		16	1111111111000100
9/7		16	1111111111000101
9/8		16	1111111111000110
9/9		16	1111111111000111
9/A		16	1111111111001000
A/1		9	111111000
A/2		16	1111111111001001
A/3		16	1111111111001010
A/4		16	1111111111001011
A/5		16	1111111111001100
A/6		16	1111111111001101
A/7		16	1111111111001110
A/8		16	1111111111001111
A/9		16	1111111111010000
A/A		16	1111111111010001
B/1		9	111111001
B/2		16	1111111111010010
B/3		16	1111111111010011
B/4		16	1111111111010100
B/5		16	1111111111010101
B/6		16	1111111111010110
B/7		16	1111111111010111
B/8		16	1111111111011000
B/9		16	1111111111011001
B/A		16	1111111111011010
C/1		9	111111010
C/2		16	1111111111011011
C/3		16	1111111111011100
C/4		16	1111111111011101
C/5		16	1111111111011110
C/6		16	1111111111011111
C/7		16	1111111111100000
C/8		16	1111111111100001
C/9		16	1111111111100010
C/A		16	1111111111100011
D/1		11	11111111001
D/2		16	1111111111100100
D/3		16	1111111111100101
D/4		16	1111111111100110
D/5		16	1111111111100111
D/6		16	1111111111101000
D/7		16	1111111111101001
D/8		16	1111111111101010
D/9		16	1111111111101011
D/A		16	1111111111101100
E/1		14	11111111100000
E/2		16	1111111111101101
E/3		16	1111111111101110
E/4		16	1111111111101111
E/5		16	1111111111110000
E/6		16	1111111111110001
E/7		16	1111111111110010
E/8		16	1111111111110011
E/9		16	1111111111110100
E/A		16	1111111111110101
F/0	(ZRL)	10	1111111010
F/1		15	111111111000011
F/2		16	1111111111110110
F/3		16	1111111111110111
F/4		16	1111111111111000
F/5		16	1111111111111001
F/6		16	1111111111111010
F/7		16	1111111111111011
F/8		16	1111111111111100
F/9		16	1111111111111101
F/A		16	1111111111111110

11 Photographic profiles

This ETS contains a data syntax for encoding many different facilities which may be provided by Videotex services supporting photographic mode. In order to ease interworking between photovideotex services and to harmonise these services a number of service profiles are specified. A service profile determines which specific facilities should be implemented.

Each profile defined below is identified in ETS 300 076 [2] by a specific "TFI" code used for photographic profiles.

Two kinds of profile are defined:

a) two groups of compatible profiles are defined, one for sequential image build-up and one for progressive image build-up, this is illustrated in figure 11;

   Backward compatibility from P5 to P4 to P3 to P2 to P1 is given, i.e. an image elaborated from an information provider according to P1 can be displayed by a terminal which supports any profile P1 to P5. Similarly an image elaborated according to P4 can only be displayed by a terminal which supports either P5 or P4. P5 and P4 also support progressive build-up of CIF and CCIR 2:1:1 image types;

   In ETS 300 076 [2] for the compatible photographic profiles, "TFI" codes are defined for the profiles P1 to P5;

b) in addition to the profiles described above, in ETS 300 076 [2] a "TFI" code is defined for a "private choice of a photographic profile P_{priv}". The use of that profile allows the selection of any attributes and parameters of this ETS. This profile is intended for "specialised private applications", (e.g. use of other colour models, resolutions, precisions) which are out of the scope of the profiles P1 to P5. However, users of this profiles have to assure within their own "private applications" that they can process and display their "private application" images.

Table 21: Summary of photographic profiles

Profile Characteristic	P1	P2	P3	P4	P5
Size of physical DDA	320 x 240	320 x 480	640 x 480	640 x 480	640 x 480
Source Pixel Density	CIF	CIF + CCIR 2:1:1	CIF + CCIR 2:1:1 + CCIR 4:2:2	CIF + CCIR 2:1:1 + CCIR 4:2:2	CIF + CCIR 2:1:1 + CCIR 4:2:2
JPEG Coding	non-hierar-chical DCT sequential Huffman	non-hierar-chical DCT sequential Huffman	non-hierar-chical DCT sequential Huffman	non-hierar-chical DCT sequential & progressive Huffman	non-hierar-chical DCT sequential & progressive Huffman
Conformance to ISO/IEC DIS 10918/CCITT Recommendation T.81	Baseline	Baseline	Baseline	Baseline + extended system, spec-tral selection	Baseline + extended sy-stem, spectral selection & suc-cessive approxi-mation

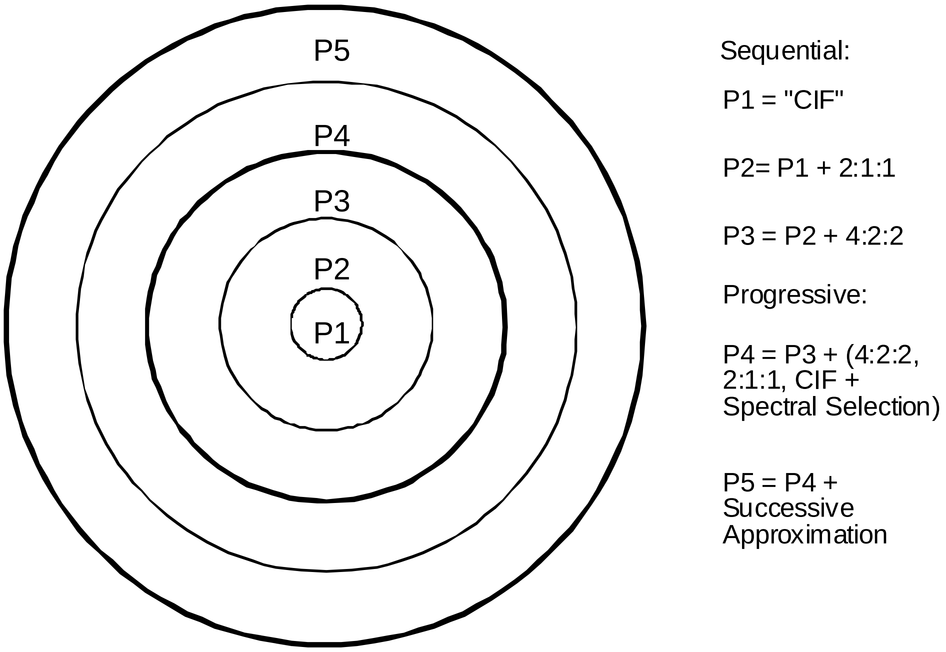


Figure 11: Compatible photovideotex profile structure

For the definitions of the compatible photographic profiles P1 to P5 and the private choice of photographic profile (P_{priv}) the following definitions apply:

Recognise: means to determine the syntactic form, but not necessarily the semantics of a code sequence.

Execute: means to process a code sequence to allow the display of information conveyed by the code sequence and by subsequent code sequences.

NOTE: The central part of a photovideotex system is the database to which editing equipment and user terminals are connected. Since, for compatibility reasons, some constraints (i.e. the JPEG demand for handling the baseline system) are imposed on the user's terminal which are not relevant for the database, the requirements apply only to user terminals. Requirements on editing equipment are outside the scope of this profile specification.

11.1 Compatible photographic profiles (P1 to P5)

In the following subclauses the five compatible photographic profiles are described. The profile is defined by stating which parts of this ETS shall be supported and which parts of ISO/IEC DIS 10918 [13] shall be supported.

11.1.1 Profile P1

This is the profile for a service which has a physical DDA of 320 pixels x 240 lines, with the pixel density defined in CCITT Recommendation H.261 [6] for CIF. It only uses the baseline system - for sequential mode of display - as defined in ISO/IEC DIS 10918 ("JPEG") [13].

The profile shall satisfy the following Clauses and subclauses of this ETS:

5 ISO/IEC 9281 syntax and switching structure: the described syntax and switching structure shall be recognised and executed.

6 Coding of the picture data entity: the described coding shall be recognised and executed.

7.3.1.1 Reset to default: shall be recognised and executed.

7.3.2.1 Full Screen display: shall be recognised and executed.

NOTE 1: This is a local matter in the terminal. Correct rendering of the full screen cannot be guaranteed.

7.3.2.2 Source aspect ratio: shall be recognised, but only the value 4/3 shall be valid.

7.3.2.3 Photo-area location: shall be recognised and executed (inside the physical DDA).

7.3.2.4 Photo-area size: shall be recognised and executed (less than or equal to the physical DDA).

7.3.2.5 Picture placement: shall be recognised and executed with the Reference Point in the top left-hand corner of the picture (inside the physical DDA).

NOTE 2: For some implementations, when used in conjunction with a one-layer alphamosaic display, location and size of the photographic image may be restricted by the mosaic character positions, e.g. multiple of 8 pixels horizontally and multiple of 10 pixels vertically.

7.3.2.6 Clear photo-area: shall be recognised and executed.

7.3.3.1 Source picture comments: shall be recognised, but use is optional.

7.3.3.2 Source picture dimensions: shall be recognised and executed.

7.3.3.3 Source pixel density: shall be recognised, but only CIF shall be valid.

7.3.3.4 Source sweep direction: shall be recognised, but only line-build-up: left to right and row-build-up: top to bottom shall be executed.

7.3.3.5 DC images: shall be recognised and executed.

7.3.4.1 Source component description: shall be recognised, but only Y or Y, C_R, C_B with block interleave shall be executed.

7.3.4.2 Source component data precision: shall be recognised, but only up to 8 bits per component shall be executed.

7.3.4.3 Source component order: shall be recognised and executed.

7.3.4.4 Source level assignment: shall be recognised, but only CCIR Recommendation 601, part 1 [9] shall be valid.

7.3.5.1 JPEG coding mode: shall be recognised:

for hierarchical mode only non-hierarchical mode shall be executed;
for type of algorithm only DCT shall be executed;
for build-up mode only sequential mode shall be executed;
for entropy coding technique only Huffman coding shall be executed.

7.3.5.3 Application marker codes assignment: shall be recognized. Use of the colour palette marker code is optional, when it is used, the formatting shall conform to the clause on colour definition of the ETS 300 072 [1].

7.3.5.2 Encoding table management: shall be recognised, and shall be executed except <ttp>= "conditioning tables".

7.3.6.1 Translation mode encoding: shall be recognised, mode 0 (no translation) shall be executed while modes 2 and 4 are optional.

NOTE 3: Manufacturers are strongly encouraged to implement both modes 0 and 2.

The full conformance to the baseline system described in ISO/IEC DIS 10918 [13] shall be valid.

NOTE 4: The ETSI compliance to ISO/IEC DIS 10918 [13] is still under discussion within ISO.

11.1.2 Profile P2

This is the profile of a service which has a physical DDA of 320 pixels x 480 lines with the pixel density defined by the CCIR Recommendation 601, part 1 [9], in the level "2:1:1" of the compatible family (see the note below). It only uses the baseline system - for sequential mode of display - as defined in ISO/IEC DIS 10918 ("JPEG") [13].

NOTE: The CCIR "2:1:1" is a straight forward derivation of the "CCIR 601 4:2:2". It is not fully described in CCIR Recommendation 601, part 1 [9], so refer to Annex F for a full description according to "CCIR Recommendation 601 [9]" rules.

The profile shall satisfy the following Clauses and subclauses of this ETS:

Profile P1 plus the following:

6.3.3.3 Source pixel density: shall be recognised, but only CIF and "CCIR 601 [9] 2:1:1" shall be valid.

The full conformance to the baseline system described in ISO/IEC DIS 10918 [13] shall be valid.

11.1.3 Profile P3

This is the profile of a service which implements normal resolution TV pictures (CCIR Recommendation 601, part 1, 4:2:2 [9]) with sequential picture build-up.

The profile shall satisfy the following Clauses and subclauses of this ETS:

Profile P2 plus the following:

6.3.2.5 Picture placement: shall be recognised, the default shall be executed, execution of other values shall be optional.

6.3.3.3 Source pixel density: shall be recognised, but only CIF, "CCIR 601 2:1:1" and "CCIR 601 4:2:2" shall be valid.

The full conformance to the baseline system described in ISO DIS 10918 [13] shall be valid.

11.1.4 Profile P4

This is the profile of a service which implements normal resolution TV pictures (CCIR Recommendation 601, part 1, 4:2:2) [9] with both sequential and progressive (spectral selection only) picture build-up.

The profile shall satisfy the following Clauses and subclauses of this ETS:

Profile P3 plus the following:

6.3.3.3 Source pixel density: shall be recognised, but only CIF, "CCIR 601 2:1:1" and "CCIR 601 4:2:2" shall be valid.

6.3.5.1 JPEG coding mode: shall be recognised:

for hierarchical mode only non-hierarchical mode shall be executed;
for type of algorithm only DCT shall be executed;
for build-up mode both sequential and progressive mode shall be executed;
for entropy coding technique only Huffman coding shall be executed.

The full conformance to the baseline system described in ISO/IEC DIS 10918 [13] shall be valid.

Conformance to the extended system for progressive picture build-up using spectral selection only as described in ISO/IEC DIS 10918 [13] shall be valid.

NOTE: For this profile the technique described in Annex C ("special spectral selection") can be used to allow compatibility between different resolution terminals.

11.1.5 Profile P5

This is the profile of a service which implements normal resolution TV pictures (CCIR Recommendation 601, part 1, 4:2:2 [9]) with both sequential and progressive (both spectral selection and successive approximation) picture build-up.

The profile shall satisfy the following clauses and subclauses of this ETS:

Profile P4 plus the following:

The full conformance to the baseline system described in ISO/IEC DIS 10918 [13] shall be valid.

Conformance to the extended system for progressive picture build-up using spectral selection and successive approximation as described in ISO/IEC DIS 10918 [13] shall be valid.

11.2 Private choice of photographic profile (Ppriv)

This profile gives users of "private applications" the flexibility to choose any attributes and parameters from this ETS (i.e. it allows the definition of applications that are out of the scope of the compatible photographic profiles described in subclause 11.1).

NOTE: Some examples of "private use applications" could be:

- an HDTV image coded with DCT and Arithmetic Coding for professional use;
- a 2000 x 2000 C,M,Y,K image with DCT and Huffman coding for a specific printing application;
- a 16 bit/pixel 512 x 512 luminance only image with lossless DPCM for certain medical applications.

Since the same "TFI" code shall be used for all types of "private application", there shall be no guarantee provided for compatibility among all users of "private use applications". For this reason, to achieve compatibility in "private use applications", users of a given application have to reach mutual agreement within their "private application" on the specific attributes and parameters used.

Terminals built according to the profile P_{priv} shall recognise all the attributes and parameters defined by this ETS, but whether they are actually executed is not defined by this profile.

The full conformance to the Baseline system, as described by ISO/IEC DIS 10918 [13] shall be valid. Conformance to the extended system, as described by ISO/IEC DIS 10918 [13] might be applicable to some "private applications". The relevance of this is outside the scope of this ETS.

Annex A (informative): Photovideotex tutorial

A.1 Introduction

For several years a standardisation effort known by the name JPEG (Joint Photographic Experts Group) has been working toward establishing the first international digital image compression standard for continuous-tone (multi-level) still images, both greyscale and colour, for very broad applications comprising computer applications and telecommunication services.

The "Joint Photographic Expert Group", JPEG, was established in 1986 as a joint committee, under the auspices of both ISO IEC/JTC1/SC2/WG8 (coded representation of picture and audio information) and CCITT/SGVIII (Q.16), for the purpose of standardising colour image compression techniques. In 1990, the ad-hoc group "JPEG" became a the new working group, (ISO IEC/JTC1/SC2/WG10) with WG8 taking on a planning and co-ordination role.

ISO/IEC DIS 10918 [13] describes this algorithm.

Videotex service providers will certainly be able to take advantage of this technique by implementing an applicative syntax for the photographic mode of Videotex.

This Annex describes the basic principles used by the JPEG algorithm in the compression of a photographic image. The aim is to give the reader a basic understanding of greyscale and continuous colour picture compression, and thus does not include all the technical details necessary to implement the algorithm.

A.2 The present state of photovideotex

Part 3 of ETS 300 072 [1] contains the photographic syntax for services existing at the beginning of 1990. There are currently two coding techniques in use:

- 1. Differential Pulse Code Modulation (DPCM) image coding;

- 2. Discrete Cosine Transformation (DCT) image coding (CEPT 1984).

Although these two techniques are adequate to encode photographic data they do not provide enough compression (ratio of input data to output data for the compression process) for large scale Photovideotex development. The new algorithm developed by JPEG, based on the DCT, provides a considerable improvement compared with the currently used techniques (note that JPEG also describes an algorithm for lossless compression, based on a DPCM, see later). As an example, a natural CCIR 601 4:2:2[1)] colour image is compressed by a factor in the order of 20 with the JPEG algorithm compared to the currently achieved ratio of 10. Such an image needs approximately 830 kbytes for uncompressed storage. By applying the DCT compression techniques developed by JPEG this can be reduced typically to 40 kbytes.

The JPEG algorithm offers sufficient flexibility to accommodate a wide variety of pixel resolutions, colour spaces and transmission bandwidths.

The JPE[1)] G algorithm also offers to display reconstructed images in two ways known as sequential mode and progressive mode. In sequential mode the image is coded or decoded to full quality in one pass, from the top to the bottom of the screen. In progressive mode the entire image is treated in separate passes the displayed image becoming progressively sharper.

The JPEG compression algorithm described in ISO/IEC DIS 10918 [13] consists of three basic functional parts: a "baseline sequential system"; an "extended system" for greater precision and progressive coding modes; and an "independent function" for sequential lossless (reversible) coding.

---

[1] CCIR 601 resolution for the luminance component is 720 x 576

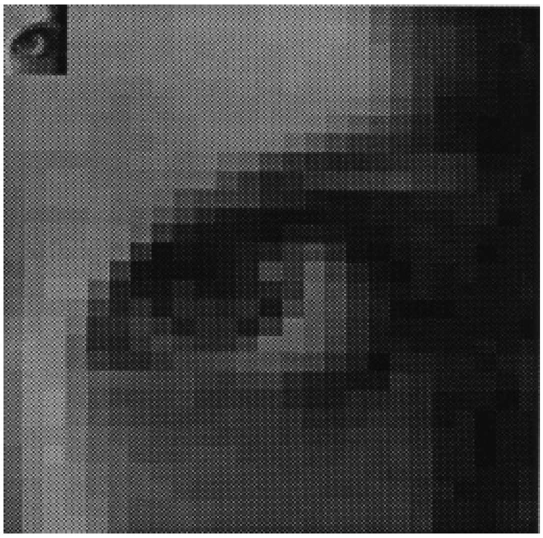
The baseline system will form the basis for photovideotex applications, it is the minimum implementation required by all DCT-based systems defined in ISO/IEC DIS 10918 [13]. The baseline system is described in more detail in Clause A.5.

A.3 Image representation

The ways one can obtain an image in a discrete form, are varied and constitute usually a specific domain of digital signal processing. However, this is not the subject of this tutorial; the reader is referred to the few selected books given in the bibliography Clause of this Annex.

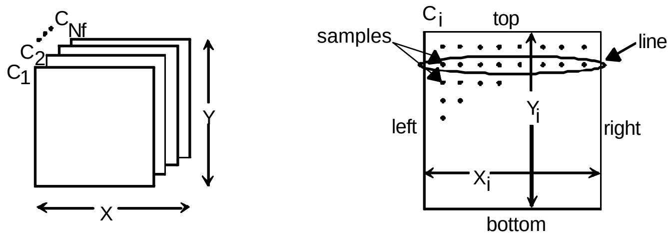
In general terms, the photographic image is usually defined as a set of 2-dimensional tables where each table represents an array of numbers (often natural integers), each number representing in turn, a value of a sample of a given component. As illustrated in figure A.2, a source image consists of a set of Nc components (e.g. Nc=3, for the "R,G,B" colour space), with overall dimensions X pixels horizontally (pixel=picture element, or the smallest element of the picture) by Y pixels vertically. Each component is represented by a rectangular array of samples of dimension x_i by y_i. The number of samples of each component is not necessarily identical. We are referring here to a sub-sampling processing (e.g. in the Y, C_R, C_B colour space, the C_R and C_B components are sub-sampled by a factor of 2 compared to the Y component).

From this representation, when reconstructing the image for display, each pixel shall be shown as a small rectangle filled with a colour calculated from the samples values of each component which represent the pixel. figure A.1 gives a practical illustration of the above definitions.



Pixel array of a portion of the image

Figure A.1: Illustration of a digitised image.



(a) Source image with multiple components (b) Characteristics of an image component

Figure A.2: Image formal representation

As examples, typical array dimensions are (for luminance):

320 x 576 (CCIR 2:1:1)
640 x 350 (EGA with border)
640 x 480 (VGA with border)
720 x 576 (CCIR 4:2:2)

The image in digital form is then suitable for further digital processing. The algorithms developed by JPEG are applied to the digital image to compress the amount of stored data and to make the transport and storage of the image data more efficient.

A.4 The JPEG compression technique

A.4.1 Lossy and lossless compression

The JPEG standard specifies two classes of compression processes, lossy and lossless processes. Those based on the Discrete Cosine Transform (DCT) are lossy, thereby allowing substantial compression to be achieved while producing a reconstructed image with high visual fidelity to the encoder's source image.

The simplest DCT-based compression process is referred to as the baseline sequential process. It provides a capability which is sufficient for many applications. There are additional DCT-based processes which extend the baseline sequential process to a broader range of applications. In many application environments which use extended DCT-based decoding processes, the baseline decoding process is required to be present in order to provide a default decoding capability.

The second class of compression process is not based upon the DCT and is provided to meet the needs of applications requiring lossless compression. This lossless compression scheme is used independently of any of the DCT-based processes.

The amount of compression provided by the various processes is dependent on the characteristics of the particular image being compressed, as well as on the picture quality desired by the application.

For colour images with moderately complex scenes, all DCT-based modes of operation typically produce the following levels of picture quality for the indicated ranges of compression. These levels are only a guide-line - quality and compression can vary largely according to source image characteristics and scene content:

0,25 - 0,5 bits/pixel: moderate to good quality, sufficient for some applications;

0,5 - 0,75 bits/pixel: good to very good quality, sufficient for many applications;

0,75 - 1,5 bits/pixel: excellent quality, sufficient for most applications;

1,5 - 2,0 bits/pixel: usually indistinguishable from the originals, sufficient for the most demanding applications.

A.4.2 Modes of encoding

There are 4 main modes of encoding available:

a) **Sequential**: each component of the image is encoded in a single left-to-right, top-to-bottom scan;

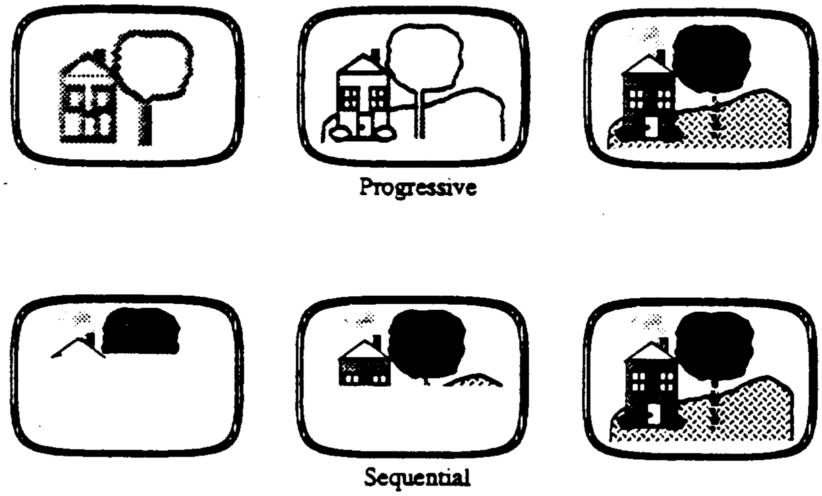


Figure A.3: Progressive versus sequential presentation

b) **Progressive**: the image is encoded in multiple scans, giving first a crude image refined by additional scans;

c) **Hierarchical**: the image is encoded with multiple spatial resolutions, for adaptation to network speed and to terminal resolution capability. Typically the lower-resolution versions of the image may be accessed without having to decompress the source image's full resolution;

In hierarchical mode an image is encoded as a sequence of frames. These frames provide reference reconstructed components which are usually needed for prediction in subsequent frames. Except for the first frame of a given component, differential frames encode the difference between source components and reference reconstructed components. The coding of the differences may be done using only DCT-based processes, only lossless processes, or DCT-based processes with a final lossless process for each component. Downsampling and upsampling filters may be used to provide a pyramid of spatial resolutions as shown in figure A.4. Alternatively, the hierarchical mode can be used to improve the quality of the reconstructed components at a given spatial resolution.

There exist different ways for implementing the hierarchical mode, one of them is described in Annex C of this ETS.

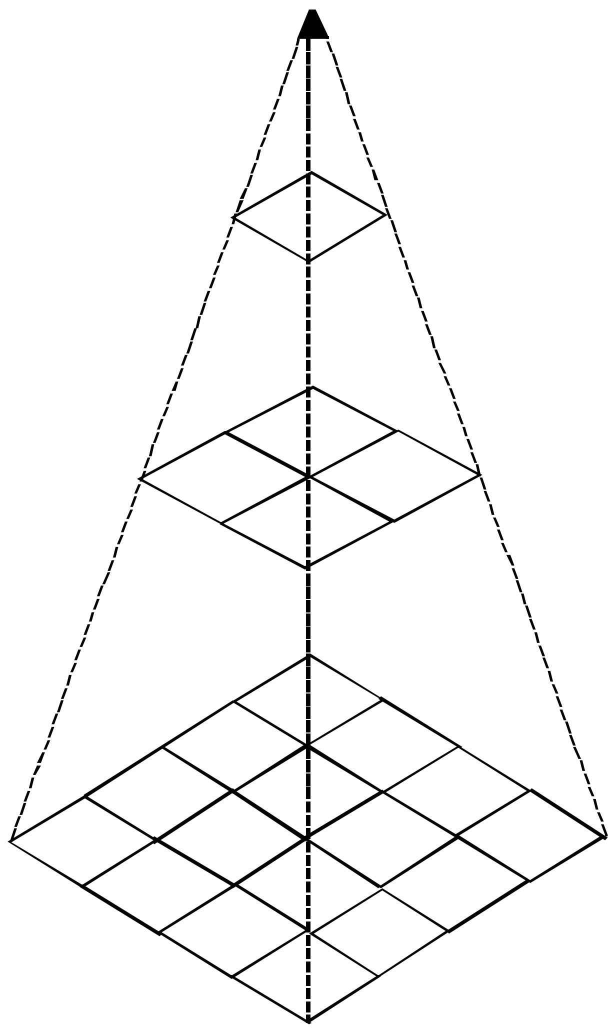


Figure A.4: Hierarchical multi-resolution encoding

d) **Lossless**: an encoding shall will guarantee exact recovery of the source image, for an acceptable compression ratio.

All those modes can be combined with however some limitations (see ISO/IEC DIS 10918 [13]).

A.4.3 The DCT-based coding

The JPEG compression algorithm described in ISO/IEC DIS 10918 [13] makes use of a variety of techniques to achieve compression. figure A.5 illustrates the key processing steps performed by all modes of encoding of the DCT-based algorithm (the only mode of encoding considered here) in the form of a block diagram.

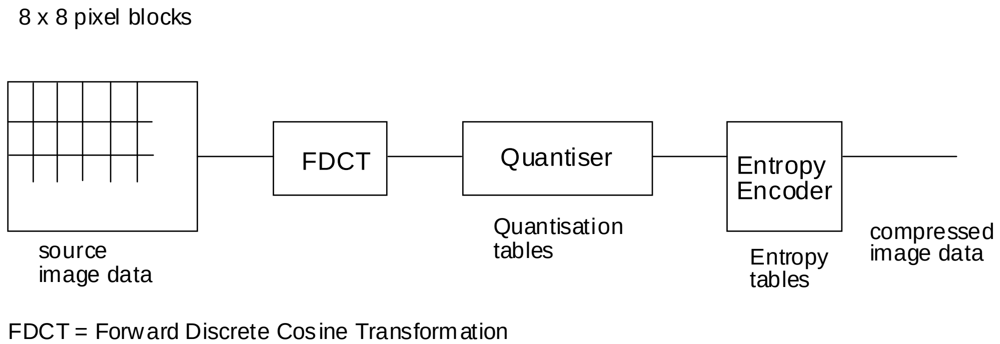


Figure A.5: DCT-based encoder functional blocks.

Common to all encoding modes of the DCT-based algorithm an 8 x 8 DCT is applied to the image data to concentrate the information present in the blocks of 8 x 8 component samples into a relatively small number of transform coefficients. The data is then quantised to reduce the number of bits used to encode the coefficients and Huffman coding or arithmetic coding is used to achieve further compression before the data is transmitted or stored.

A description of these techniques and the way in which they are applied by the JPEG algorithm is given in the following subclauses. In this annex only, Huffman coding will be described (for the description of the arithmetic coder see the ISO/IEC DIS 10918 [13]).

A.4.3.1 The discrete cosine transform

The discrete cosine transform is used for its good property of *energy compaction* when applied to an image signal. When the forward DCT (FDCT) is applied to a block of picture samples it is transformed to a block of coefficients. Each coefficient representing the "weight" of a given 8 x 8 pattern, so-called *"cosine image-basis"* (see figure A.6), present in the given 8 x 8 image samples. It can also be viewed as "the importance" of a *spectral band* in the original block of 8 x 8 image samples. The coefficient with zero frequency in both dimensions is known as the DC coefficient, and the other 63 are known as the AC coefficients. It is important to note that this transformation provides *no compression* at all, simply it changes the mode of representation of the source image, as does any linear transformation. However, a property of the FDCT is that, typically and statistically, it concentrates the energy contained in the block of picture samples into just a few of the transform coefficients. It is this compaction of energy property that finally allows later at the compression stages (quantisation and entropy coding) good compression.

For the DCT-based algorithm the image to be transformed is divided into 8 x 8 pixel blocks. The FDCT is applied to an 8 x 8 block of picture samples, this results in an 8 x 8 block of coefficients. The first coefficient termed the DC value is eight times the average of the 64 pixel values. Each of the remaining coefficients, referred to as AC_1 to AC_{63}, correspond to a two dimensional "DCT-filtered" signal within the pixel block. Due to the previously mentioned statistical properties, typically only a few of the 63 AC coefficients have a non-zero value.

A formalisation of the process is now outlined.

A general separable linear transformation on an image matrix [f] may be written in the form:

$$[F]=[U]^t[f][V];$$

where [F] is the transform of the image f(i,j)=[f], [U] and [V] are the operators, and the subscript t denotes the matrix transpose. [U] and [V] are orthogonal operators if:

$$[U]^t[U]=I;$$

$$[V]^t[V]=I;$$

Hence the inverse transform of (1) may be written in the form:

$$[f]=[U][F][V]^t;$$

If the operators [U] and [V] are written in the form of column-vectors:

$$[U]=[\mathbf{u}_1\ \mathbf{u}_2 \ldots\mathbf{u}_n];$$

$$[V]=[\mathbf{v}_1\ \mathbf{v}_2 \ldots\mathbf{v}_n];$$

then:

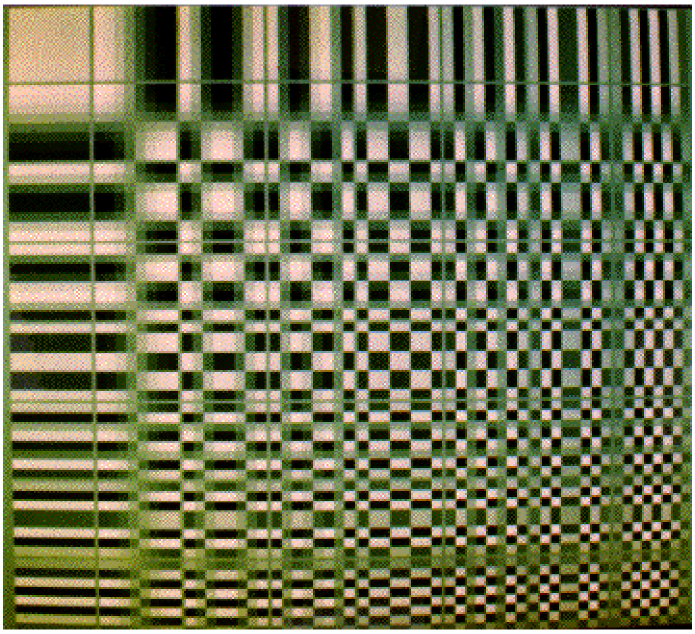
$$[F]=[\mathbf{u}_1\ \mathbf{u}_2 \ldots\mathbf{u}_n][f][\mathbf{v}_1\ \mathbf{v}_2 \ldots\mathbf{v}_n]^t;$$

it follows that:

$$[f]= \quad F_{ij}\,\mathbf{u}_i\,\mathbf{v}^t_j.$$

This means that the original image [f] is a weighted ("coefficient" Fij) sum of "DCT-subimages" ($\mathbf{u}_i\ \mathbf{v}^t_j$). The weight F_{ij} giving the importance of a given sub-image ($\mathbf{u}_i\ \mathbf{v}^t_j$) in the original image [f]. The terms (F_{ij}) are the so-called DCT coefficients and the sub-images are the 2-D representation of the DCT transformation kernel. Those 64 DCT-sub-images ($\mathbf{u}_i\ \mathbf{v}^t_j$) are shown below.

($U_0V^t_0$) = "DC term" ($U_0V^t_7$)



($U_7V^t_7$) = the "highest frequency term"

Figure A.6 : the 8 x 8 DCT basis sub-images

In order to simplify the implementation, the DC value is centred around zero by subtracting $2^{(P-1)}$ before the DCT is applied. P is the precision of the unsigned array elements in bits (8 for the baseline system). After the inverse DCT, the level shift is removed. Thus, both DC and AC can use the same 2's complement representation.

The cosine transform is intrinsically reversible, i.e. theoretically, after a cosine transform, an inverse transform reproduces the pixel values from the transform coefficients. However an actual implementation may cause non-reversibility due to calculation noise and differences in the type of algorithm used.

For the mathematical description of the DCT see ISO/IEC DIS 10918 [13] and informative references [A4] and [A5] given in Clause A.8 (Bibliography).

A.4.3.2 Quantisation

The next step in the encoding process is the quantisation of the 8 x 8 block containing the DCT coefficients. This step constitutes the main compression stage, the lossy stage and so is the most "crucial" point of the algorithm.

It is worth stressing that loss is only introduced during the quantisation process i.e. any artifacts which are visible in the decompressed image are due to the quantisation stage, which makes the whole compression scheme irreversible.

Depending on the viewing conditions, tests with human observers have shown that AC coefficients are not of equal importance. This "psychovisual" property allows for coarser quantisation of the "less sensitive" coefficients and finer quantisation of the "more sensitive" ones. Moreover, according to the application (i.e. type of images, viewing conditions, type of application, ...) the sensitivity law is drastically different! Furthermore, the derivation of the "optimum" quantisation steps is not a straightforward task! So, each application has to define its own quantisation values, nearly as "proprietary" values, since no default values are applicable. Typically in this ETS for the photovideotex service, we recommend using a proposed set of quantisation tables which have been tested. However, as the values are systematically sent with the compressed image, any other values could be used.

The quantisation values are specified in an 8 x 8 table (quantisation table), each of the DCT coefficients has a separately specified quantisation value. The linear quantisation procedure is as follows:

Q(u,v) represents the table of quantisation values indexed by u and v.
F(u,v) represents the unquantised DCT coefficients.
F'(u,v) represents the dequantised DCT coefficients.
C(u,v) represents the quantised DCT coefficients.

For F(u,v) >= 0

C(u,v) = { F(u,v) + Q(u,v)/2 } / Q(u,v)

and for F(u,v) < 0

C(u,v) = { F(u,v) - Q(u,v)/2 } / Q(u,v)

The dequantisation process is also linear:

F'(u,v) = C(u,v) x Q(u,v)

In principle a different quantisation table should be defined for each colour coordinate system, spatial resolution, data precision and application. Examples of tables which produce good results for CCIR 601 "4:2:2" images are given in ISO/IEC DIS 10918 [13], and examples of tables which produce good results for CCIR 601 "2:1:1" and CCITT "CIF" are given in this ETS. Two quantisation tables are used one for the luminance components and one for the chrominance components.

For example, the precision of the input data for the baseline system is restricted to 8 bits [- 128, 127] and consequently the quantised transformed coefficients are limited to 11 bits [- 1024, 1023] (i.e. if the quantisation step equals 1).

After quantisation the DCT coefficients of the baseline system have a precision as follows:

$$11 - \log_2(M)$$

where M is the quantisation value for the coefficient.

A.4.3.3 Huffman coding [A8]

Huffman coding is a form of entropy coding. Entropy coding techniques achieve compression by exploiting the redundancy that occurs in the data they encode. Redundancy is a characteristic which is related to "predictability" in the data. For example, an image of constant grey level is fully predictable once the grey level of the first pixel is known. On the other hand, a white noise random field is totally unpredictable and every pixel has to be stored to reproduce the image.

The aim of entropy coding is to encode a block of data with a bit rate which is close to the entropy of the block. Consider a block containing M transformed coefficients with each coefficient represented by B bits. The block contains a total of MB bits with probabilities p_i. The entropy of the block is given by:

$$\sum_i p_i(-\log_2 p_i) = H$$

where $i = 0,1,....,2^{MB}$.

If the data contains redundancy the entropy shall be less than B bits per coefficient.

Entropy coding results in variable length coding, i.e. highly probable coefficient values are represented by small-length codes, and vice versa.

Huffman coding is the technique applied to the AC and DC transform coefficients to assign a short code word to the most frequently occurring configuration of coefficients.

In the JPEG DCT-based algorithm the quantised DC coefficient is treated separately from the 63 AC coefficients. Differential Pulse Code Modulation (DPCM) is used to encode the DC value prior to Huffman coding. Zigzag ordering is applied to the AC coefficients prior to Huffman coding.

A.4.3.3.1 PCM encoding of the DC coefficients

Since there usually exists a strong correlation between DC coefficients of adjacent 8 x 8 blocks, the DC coefficient of a given block is differentially encoded with respect to that of the previous block.

For a given block i this is expressed by the relationship:

Diff_DC(i) = DC(i) - DC(i-1)

In the decoder the difference is decoded and added to the prediction. At the start of processing the value 0 is used for the prediction. As explained in subclause A.4.3.1. the DC values are level shifted and centred around zero.

A.4.3.3.2 Zigzag Ordering of AC Coefficients

For the purpose of Entropy coding, the two-dimensional array of DCT coefficients is rearranged using a zigzag scan into a one-dimensional array. The zigzag ordering of the coefficients is illustrated in figure A.7.

The coefficients are ordered so that the "low frequency" coefficients occur first. This scanning follows the statistical property of compaction of the energy "around the DC coefficient" where the coefficients are usually non-zero around the DC coefficient and zero for the high frequency AC. So, to increase the compression, AC values and runs of zeros are grouped and coded as (0 runs, AC value). In addition if the remaining ACs of the 8 x 8 bock are zeros, there are encoded explicitly as an EOB code (End of Block).

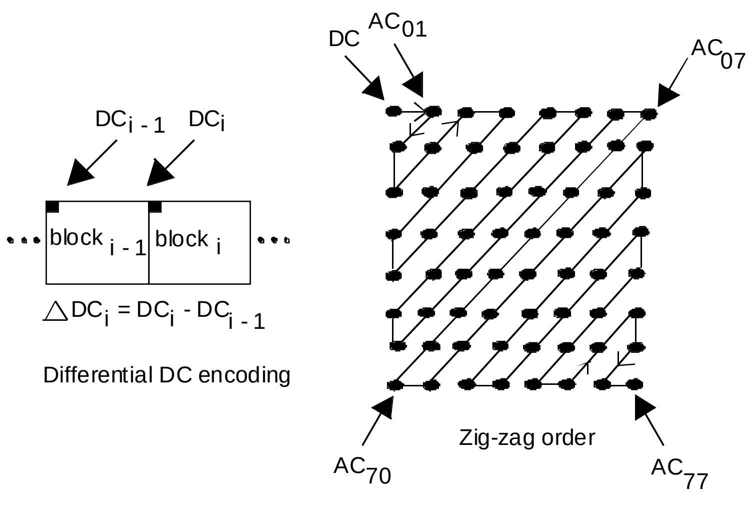


Figure A.7: Zigzag ordering of the AC coefficients.

A.4.4 Lossless coding

The main procedure for the lossless encoding process is depicted in figure A.8. A predictor combines the values of up to three neighbouring samples (A, B, and C) to form a prediction of the sample indicated by X in the figure A.8. This prediction is then subtracted from the actual value of sample X, and the difference is losslessly entropy coded by either Huffman or arithmetic coding.

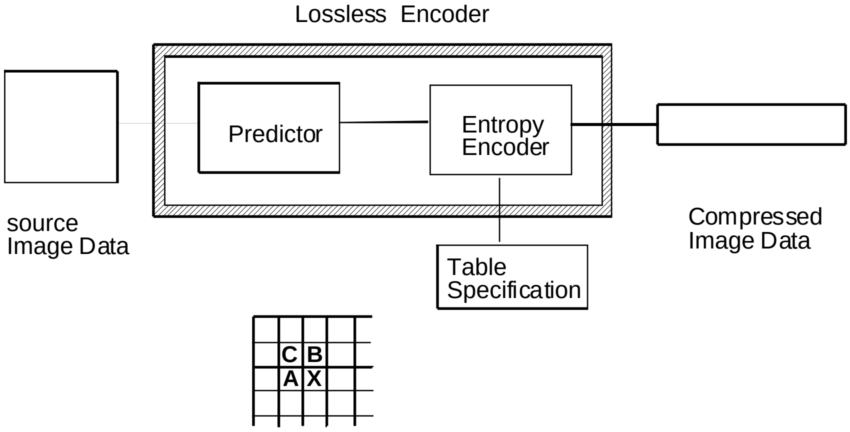


Figure A.8: Lossless encoder schematic

A.4.5 Source images and data interleaving

The previous subclauses described how a single component of an image is decomposed in 8 x 8 blocks and then transformed, quantised, and coded in a DCT-based mode of operation. Although the JPEG algorithm is able to treat only a single component image (monochrome or greyscale) it is also designed to handle a wide range of multiple-component images.

A multiple component image may be compressed in either interleaved or non-interleaved order. In the latter case each image component is entirely compressed and output as part of the compressed data stream in a separate scan before starting the next component. In the interleaved case, 8 x 8 blocks from each component are processed in a round-robin-fashion, and compressed blocks are output in interleaved order into the data stream. A maximum of four components may be interleaved within a scan.

In the sequential mode both orders of interleave are allowed. As an example a three components image, such as a CCIR 601 "4:2:2" image, should be compressed in three separate scans in non-interleaved mode but only in a single scan if the image components are interleaved. A mixture of interleaving order is also permitted, e.g. for the CCIR 601 "4:2:2", the luminance Y can be coded in a single non-interleave scan while the chrominance components can be coded together in a single interleave scan.

In the progressive mode only the DC coefficients of different components can be interleaved in the same scan, AC coefficients have to be compressed in a separate scan for each component.

A.4.6 Data organisation and signalling parameters

The encoded data stream contains not only the compressed image data but also parameters which contain information about the data which is being sent. For example, codes to indicate the start of the image or the start of a table are contained in the data stream.

A detailed discussion of these parameters is not included in this ETS. The normative part of this ETS contains a description of the data organisation and signalling parameters as applicable to photovideotex applications. Other issues which are relevant for an implementation of the JPEG algorithm in the photovideotex environment, such as the use of tables (quantisation and entropy tables), are also described in the normative part of this ETS.

A.5 The baseline system

The simplest DCT-based coding process is referred to as the baseline sequential process. It is required to be present in all DCT-based modes of operation for compliance to JPEG. It provides a capability which is cost-effective and sufficient for many applications and it acts as a "fall-back" or default mode, for international interworking between services.

The baseline sequential system codes an image to full quality in one pass, the processing usually starting at the top of the image and finishing at the bottom. At each point only a small part of the image is being buffered, the sequential mode is therefore suitable for applications without a full image buffer.

The DCT is applied in the sequential mode as described in subclause A.4.2. The transform coefficients are quantised as explained in subclause A.4.3.2. Following quantisation and ordering the coefficients under go Huffman encoding.

The baseline sequential system has the following main characteristics:

a) it operates on images with 8-bits/pixel/component only;

b) it uses Huffman coding only;

c) it uses, at most, two sets of Huffman tables per scan.

A.6 The extended system

The baseline system can be extended in many ways, one of them is to provide progressive coding of the DCT coefficients. In the progressive mode of operation a rough, but recognizable, version of the image appears quickly on the viewer's screen and is refined by successive passes to produce the same final image as the sequential mode. It is also extended to up to 12-bits source image, four sets of Huffman tables, arithmetic entropy coding option etc.

The progressive coding modes are defined in ISO DIS 10918 [13] by two complementary modes: successive approximation and spectral selection. For both progressive modes the stages prior to the Huffman encoding: DCT, quantisation and coefficient ordering, are the same as for the sequential mode.

A.6.1 Coding model for successive approximation

In the successive approximation mode, the DCT coefficients are divided by an integer power of two before they are coded. The precision of the coefficients is therefore reduced. The coefficients are multiplied by the same power of two before the inverse DCT is applied. The precision of the coefficients is increased in the subsequent stages by reducing the scaling coefficient each time by a factor of two. In the final pass the DCT coefficients are coded with full precision. All coefficients, AC and DC, are scaled in the same way.

A.6.2 Coding model for spectral selection

In the spectral selection mode, the zigzag array of DCT coefficients described in figure A.7 is segmented into "frequency" bands (called spectral bands) and each band is coded as a separate pass. In this progressive mode the coding model of DCT coefficients is the same as for the sequential mode. However, the code tables of the baseline sequential system are extended to encode runs to end-of-band.

A particular use of the spectral selection mode is described in Annex C.

A.6.3 Hierarchical encoding

ISO/IEC DIS 10918 [13] defines another form of progressive coding: the hierarchical mode. The hierarchical mode provides a "pyramidal" encoding of an image at different resolutions, each resolution differing from the adjacent by a factor of two in either the horizontal, or vertical direction or both. The encoding procedure can be summarized as follows:

1) the original image is filtered and down-sampled by the desired factor of 2^N in each dimension;

2) the image of reduced size is encoded either sequentially or progressively according to the methods previously described;

3) the image of reduced size is upsampled by 2 in horizontal and/or vertical direction and used as the prediction for the next (2^{N-1}) stage;

4) the 2^{N-1} difference image is encoded.

This procedure is repeated until the final stage is reached. This capability is useful in applications where it is desirable to access a smaller version of an image or to support terminals with different resolutions from one database.

A more detailed description of one example of implementation of the hierarchical mode can be found in Annex C.

A.7 Summary

The algorithms developed by JPEG provide an excellent basis for the development of photovideotex services.

The baseline system ensures compatibility between simple systems using 8 bits per pixel per component. The sequential mode allows for cost effective implementations of the JPEG algorithm. The progressive system has advantages when a low speed network is being used. The image is built-up in stages and although the total build-up time is the same as with the sequential mode, the first scan of the image is displayed in less than the total time. The first scan of the image is often enough to allow the user to identify the image.

Within the different options provide by the JPEG algorithm, some useful selections have been made for "basic photovideotex" (compatible profiles in Clause 11), some are left for possible other selections for a private choice of photographic profile.

A.8 Bibliography

[A1] G. Wallace, "Overview of the JPEG still image compression standard", electronic imaging 1990, Boston.

[A2] A. Jain, "Image data compression: A review", proceedings of the IEEE vol.69 No. 3 March 1981.

[A3] A. Jain, "Fundamentals of digital image processing", Prentice Hall information and system sciences series.

[A4] N. Ahmed, K.R. Rao, "Orthogonal transform for digital signal processing", Springer-Verlag, New York.

[A5] N. Ahmed, T. Natarajan, K.R. Rao, "Discrete cosine transform", IEEE Trans. on Comput., January 1974.

[A6] D.A. Huffman, "A method for the construction of minimum redundancy codes", Proc.IRE, vol.40, Sept. 1952.

Annex B (informative): Implementation guidelines on display rendering

B.1 Introduction

Due to the fact that the three base Videotex syntaxes (CCITT data syntax I-III) differ not only in their fundamental underlying parameters (such as: screen resolution, aspect ratio, terminal model, planes, order of planes, number of planes etc.), but also in their terminal profiles (data syntax II), the photographic capability of this ETS has been defined in a way which is independent of any base Videotex system.

This is necessary to ensure that any image can be displayed on any system, although certain artifacts might result if the display characteristics of the terminal differ from the transmitted source image. The process in which the terminal tries to cope with the mapping of the characteristics of the source image to the characteristics of it's own display capability, is called rendering.

There are two types of rendering which are described in this annex: rendering of resolution and rendering of colour.

Since all rendering functions are local functions to the terminal (depending on the hardware and software capabilities of a given terminal configuration) and not part of the photographic data transfer, they are clearly outside of the scope of the normative part of this ETS. However, because of the importance of these issues for terminal implementors some guidelines are provided to cope with rendering problems in the terminal.

B.2 Rendering of resolution

B.2.1 Resolution independence

Independence of display resolution is an important factor even within a single Videotex system, since there may be a large number of different display platforms; such as various personal computer display boards or dedicated terminals. It is not desirable to restrict an information service to only certain groups, since this would restrict the market for the information provider.

In general, it is possible to display any array of pixels on any resolution display screen although certain presentation artifacts will result if the source and the target pixel array size do not match exactly. There are two ways of handling a mismatch between the source pixel array size and the target display pixel resolution. These are shrinking an image to fit or scaling the pixel array.

The following series of images show a photographic image which is rendered on a display device of different resolution, first by the shrinking technique and then by the scaling technique.

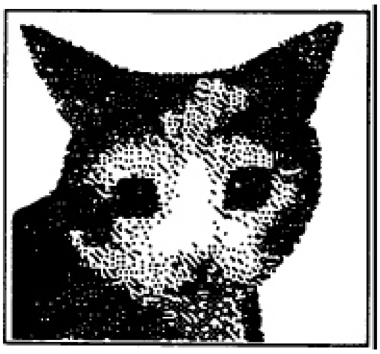


Figure B.1: Original image as intended by the information provider

The shrinking technique results in an image which is somewhat smaller than intended by the information source; however, after the shrinking there results a simple ratio between the source (logical pixels) and the target (physical pixels). For example, if a source image was defined to occupy the full display area on a 255 by 300 pixel display screen and was rendered on a 240 by 320 display screen, then the image could be shrunk by 6,25% before display.

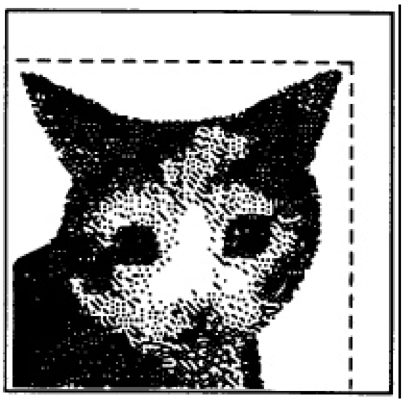


Figure B.2: Image rendered by shrinking technique

The other technique which can be used to render the display of a photographic image on a display screen of a different inherent resolution is to scale the pixel array. Logical pixels will be rendered onto the corresponding physical pixels of the target display system one by one across the display until the corresponding positional error is greater than a physical pixel position. The alignment of the source and the target pixel arrays would then be shifted over by one physical pixel and the rendering process would continue. This ensures that the relative size of the photographic image remains the same on target display devices. Effectively, this technique spreads the mismatch across the image. This is illustrated below.

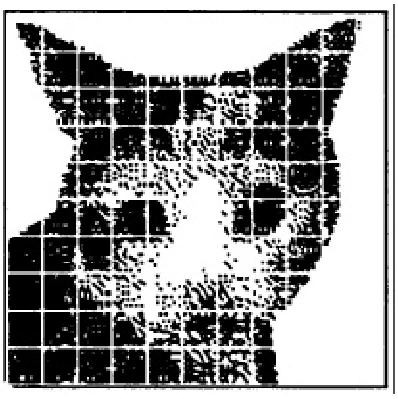


Figure B.3: Image rendered by scaling technique (worst case approach - no averaging)

The image above is a worst case situation. The gaps resulting from the scaling can be filled in by duplicating the previous pixel or by some more sophisticated averaging algorithm. The more resolution available in the luminosity or colour dimensions, then the better such an averaging technique works.

Figure B.4: Image rendered by scaling technique (averaging used to fill in gaps)

In this ETS any resolution is possible for both the target and the source display devices. Rendering of the image is the responsibility of the display device, scaling or shrinking of the image should be expected in some situations.

NOTE: Due to the possible effects of scaling or shrinking of a photographic image as a result of rendering, there is no guarantee that text, mosaic or geometric information will align with a photographic image exactly. It is not advisable to, for example, attempt to draw a moustache on a photographic image of a face or align text exactly with a photographic image. Such a restriction also helps to resolve the difference between the terminal models of the various Videotex systems, some of which ascribe separate underlying or overlaying display planes or a shared plane to the display of a photographic image. It is advisable to treat a photographic image separately and care should be taken when images are mixed with other drawing techniques.

B.2.2 Display rendering guidelines for Data Syntax II profiles

B.2.2.1 Pixel alignment

In principle the pixel resolution for the different layers (i.e. alpha-mosaic and photographic) are independent of each other, but in most implementations the pixel resolution should, most probably, be the same for all layers.

Since the shape of the characters are defined locally in the terminal the dot matrix size is only of importance for Dynamically Redefinable Character Sets (DRCS). At present, out of the four Videotex profiles of data syntax II defined in ETS 300 072 [1], only profile 1 includes DRCS.

A typical profile 1 dot matrix size is 12 X 10 giving a defined display area resolution of 480 X 240 or if the dots are grouped together 2 X 2, 960 X 480 in the alpha-mosaic layer. The dots of the DRCS could also be grouped in the vertical direction giving a defined display area resolution of only 480 X 480.

The following examples show how different resolutions of the photographic picture can be combined with profile 1 displays.

B.2.2.1.1 CCIR Recommendation 601, Part 1 4:2:2 resolution

The *full visible display* resolution is 720 X 576 pixels gives a *defined display area* resolution of 640 X 480 pixels.

If a received 640 X 480 image is to be displayed on a 960 X 480 display and the aspect ratio should be kept (which it should), it is necessary to "add" 50% extra pixels horizontally. How that could be done is described in subclause B.2.2.2.1.

The profile 1 terminal could also choose the 480 X 480 resolution in which case 25% of the pixels should be "removed" horizontally from the received 640 X 480 image. How that could be done is described in subclause B.2.2.2.2.

B.2.2.1.2 CCIR Recommendation 601, Part 1 2:1:1 resolution

The *full visible display* resolution is 360 X 576 pixels giving a *defined display area* resolution of 320 X 480 pixels.

In this case a profile 1 terminal would most probably choose a resolution of 480 X 480 giving the need for "adding" 50% pixels horizontally (subclause B.2.2.2.1).

B.2.2.1.3 CIF format

The *full visible display* resolution is 360 X 288 pixels giving a *defined display area* resolution of 320 X 240 pixels.

In this case a profile 1 terminal would most probably choose a resolution of 480 X 240 giving the need for "adding" 50% pixels horizontally (subclause B.2.2.2.1).

B.2.2.1.4 QCIF format

The *full visible display* resolution is 180 X 144 pixels giving a *defined display area* resolution of 160 X 120 pixels.

It is unlikely that this poor resolution will be chosen for Videotex terminals, but images in QCIF format can be sent to CIF terminals occupying one quarter of the screen.

B.2.2.2 Adjustment of horizontal resolution

In some situations it is necessary to adjust the horizontal resolution to the resolution of the actual terminal. Some ways to do this are described below.

B.2.2.2.1 "Adding" pixels

"Adding" 50% extra pixels (e.g. expansion of 640 pixels to 960 pixels) can be depicted as follows:

Pixel	A	B	C		in original picture
	o	o	o		
	o	o	o	o	
Pixel	a	b	c	d	in expanded picture

1) The simplest solution is:

a = A
b = B
c = B
d = C

2) Simple interpolation gives:

a = A
b = (A + 2*B)/3
c = (2*B + C)/3
d = C

3) Using a 2nd degree polynomial gives:

a = A
b = (2*A + 8*B - C)/9
c = (-A + 8*B + 2*C)/9
d = C

Also polynomials of higher degree can be used, but the small quality improvement observed between 1, 2 and 3 indicates that there is no extra benefit in doing this.

Subjective testing of the results of the proposed expansion methods shows that 1) gives visible artifacts where as artifacts from 2) and 3) can only be seen if the picture is enlarged by zooming. The conclusion is that 2) is sufficient for most applications while 1) can be used in terminals where computing power is limited.

B.2.2.2.2 "Removing" pixels

"Removing" 25% pixels (e.g. reducing 640 pixels to 480 pixels) can be depicted as follows:

Pixel	A	B	C	D	E	in original picture
	o	o	o	o	o	
	o	o	o	o		
Pixel	a	b	c	d		in reduced picture

1) The simplest solution is:

a = A
b = B
c = D
d = E

2) Simple interpolation gives:

a = A
b = (2*B + C)/3
c = (C + 2*D)/3
d = E

3) Using a 2nd degree polynomial gives:

a = A
b = (-A + 8*B +2* C)/9
c = (2*C + 8*D - E)/9
d = E

Also polynomials of higher degree can be used, but the small quality improvement observed between 1, 2 and 3 indicates that there is no extra benefit in doing this.

It is obvious that removing pixels results in loss of information and in the light of that 2) is sufficient for most applications while 1) can be used in terminals where computing power is limited.

B.3 The concept of normalised colour space

The basic concept of a normalised colour space is that the photographic data syntax carries numbers from 0 (inclusive) to 1 (exclusive) which represents the range of the whole colour space.

If, for example, a byte of data is used to represent the colour, then 256 different colours can be represented. If, however, a terminal is unable to display 256 colours, the normalising of the colour space allows the terminal to perform the "best fit mapping" of the available colour to the normalised value.

Annex C (informative): Solutions for common compatible photovideotex databases serving different resolution terminals

C.1 Introduction

One of the major problems which has to be faced in the development of image services in general, and Videotex in particular, is the problem of a common database being accessed by terminals of different display resolutions.

In a Videotex database an image is always encoded and stored in the database prior to the user's access. Since many types of terminal are expected to be able to handle data from different databases, it is essential to provide interworking between a common database and the different resolution terminals, in order to provide:

- the opportunity for database providers to code images within the database with the aim that these images will be displayable on different resolution terminals;

- the opportunity for the users to choose their own terminal resolution with the aim that they have access to a broad range of images, but have some freedom in the selection of their display device.

To achieve this two solutions are offered:

a) the "Hierarchical Mode" as defined in ISO/IEC DIS 10918 [13];

b) "special Spectral selection", a filtering technique in the transform domain, taking full advantages of the ISO/IEC DIS 10918 [13] progressive coding algorithm (see Annex H).

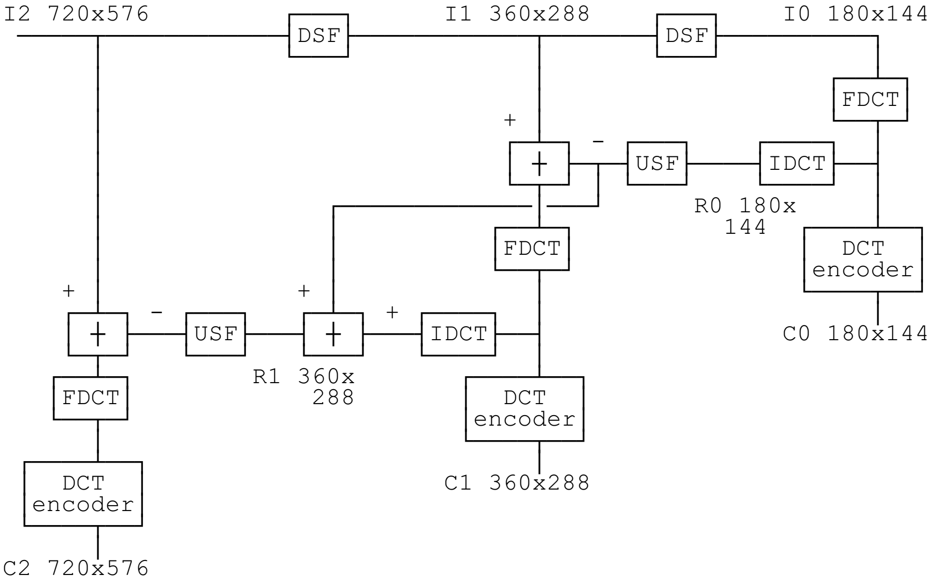
C.2 Hierarchical mode

In this technique, the basic idea is to use a sequence of downsampling filters to create a progression of reduced resolution images. Each downsampling reduces the resolution by a factor of two in either the horizontal or the vertical direction (or both).

The reduced resolution images are the input for each stage of the hierarchical progression.

C.2.1 Coding

As an example, figure C.1 describes a three stages hierarchical DCT coding process giving output images with resolution: CCITT Recommendation H.261 [6] QCIF and CIF and CCIR Recommendation 601 [9] 4:2:2.



Ii starting image stage i.

Ri reconstructed image stage i.

Ci compressed image stage i.

DSF downsampling filter, resolution divided by two in both directions.

USF upsampling filter, resolution multiplied by two in both directions.

FDCT forward DCT, IDCT inverse DCT.

Figure C.1: Hierarchical mode

Example of a three stages DCT coding process. Resolution is expressed as : number of pixels per line x number of lines, for the luminance.

At the first stage of encoding (stage 0), the lowest resolution image I_0 is coded using any of the compression algorithms defined in the ISO/IEC DIS 10918 [13]: sequential, progressive, spatial, etc......

For further stages, say stage i, the reconstructed image from the previous stage, R_{i-1}, is upsampled and used as a prediction. The reconstructed image is obtained by summing the prediction and difference images. For these stages only difference images are coded giving compressed difference images, C_i, which can be stored or transmitted.

Each stage of progression provides an output compressed image whose resolution increases by a factor of two in both directions.

C.2.2 Decoding

Figure C.2 gives the decoding scheme corresponding to the coding process described in figure C.1.

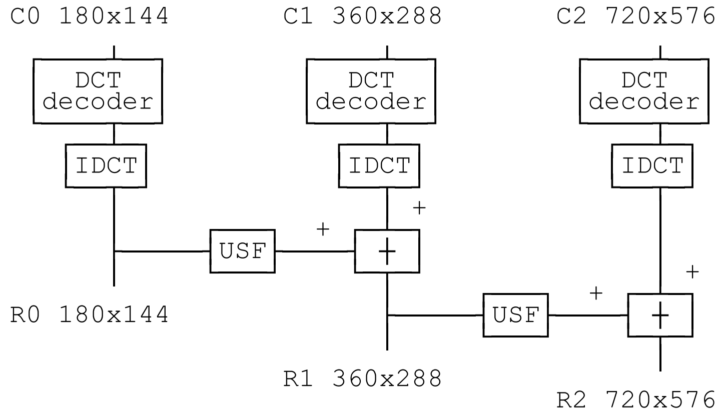


Figure C.2: Hierarchical mode

Example of a three stages DCT decoding process. All notations are the same as in figure C.1.

At the receiving end, the lowest resolution compressed image, C_0, is first decoded in the same manner as for a non-hierarchical algorithm, giving the reconstructed image R_0.

At further stages of progression, say stage i, the previous reconstructed image, R_{i-1}, is upsampled and summed to the difference decoded image in order to obtain a reconstructed image, R_i, whose resolution is increased by a factor of two in both directions. So the receiver can stop to decode the incoming data stream when the desired resolution is reached.

The characteristics of the downsampling filter are left to the system designer but should be consistent with the upsampling bilinear interpolation filter, which is defined in the ISO/IEC DIS 10918 [13].

C.2.3 Example for a "resolution pyramid" for hierarchical build-up

In the example below the suggestion is to create a logical hierarchy of resolutions (both for luminance and chrominance) with appropriate spatial increases, covering the range from QCIF to HDTV.

The spatial increases start from a resolution which is obtained from the smallest size DC image (DC components of a CCIR Recommendation 601 [9] image). The spatial increase, horizontally and vertically, are decoupled and the single steps of resolution-scaling to the next higher level take the value of either 1 or 2, depending on a logical, strictly defined scaling strategy. The strategy is such, that during the scaling process to a higher resolution the "main stages" shown in table C.1 are covered.

Table C.1: Main stages in hierarchical progression

Main stages:	
A	DC only
B	CCITT Recommendation H.261 [6] QCIF
C	CCITT Recommendation H.261 [6] CIF
D	CCIR Recommendation 601 [9] 2:1:1
E	CCIR Recommendation 601 [9] 4:2:2
F	HDTV 4:2:2

This actually is achieved as shown in table C.2.

Table C.2: Details of hierarchical progression

Main Stage: / Progr. Step:	Luminance: Resolution:				Chrominance: Resolution:			
	Horiz.	Scale:	Vert.	Scale:	Horiz.	Scale	Vert.	Scale:
A / 1	90	2	72	2	90	1	72	1
B / 2	180	1	144	2	90	1	72	2
/ 3	180	2	288	1	90	2	144	1
C / 4	360	1	288	1	180	1	144	2
/ 5	360	1	288	2	180	1	288	2
D / 6	360	2	576	1	180	2	576	1
E / 7	720	1	576	2	360	1	576	2
/ 8	720	2	1152	1	360	2	1152	1
F / 9	1420	-	1152	-	720	-	1152	-

For example, to get to Stage F, 9 progressive stages would be needed. To get to stage C, 4 progressive stages would be needed.

C.2.3.1 Advantages of the suggested technique

C.2.3.1.1 Independence of transmission, decompression and display

It is up to the terminal to decide which progressive stage to display on the screen and how. For example, on a high speed network, the terminal may decide to show only the DC-scan, then leave out all progressive stages until "CCIR 601 [9] 2:1:1", and to display that as the final resolution. On a lower speed network, the terminal could show all intermediate stages, in order to achieve a more pleasing image build-up.

Since progressive image build-up requires a frame buffer, the speed of transmission, image decompression, and image display might be entirely decoupled from one another (for the terminal to decide). Thus, even if the network speed is high, the decompression and display could be done at lower speeds (the progressive build-up is more tolerant to variations of speed). This could be the case if a single database is to serve terminals linked through both high- and low-speed channels (thus only progressive image build-up is practical).

C.2.3.1.2 Independence of image resolution from the terminal resolution.

In the rendering process the adjustment of the output display seems to be easier if there are "fixed" progression stages, where the image adjustment can be fixed (e.g. through interpolation between two stages, or by simple selection of a given stage).

C.2.3.1.3 Storage gain in the database host through the pyramidal database

In hierarchical mode the information provider can select the final resolution (e.g. "CCIR 601 [9] 4:2:2") for each stored image separately. In this respect the "special spectral selection" technique is more rigid, because, as an analogy, it would require that all images be stored according to the highest resolution to be displayed (e.g. "CCIR 601 [9] 4:2:2").

C.2.3.2 Disadvantages

The hierarchical mode, elegant in principle, suffers from one drawback: its implementation complexity.

Compared to the non-hierarchical coding techniques, the hierarchical mode requires more calculation power and memory :

- in addition to computation of forward DCT, the encoder has to perform: downsampling and upsampling filtering, inverse DCT computation and difference computation;

  additional memory is also required to store the upsampled image prior to difference computation;

- in addition to computation of inverse DCT, the decoder has to perform bilinear interpolation filtering;

additional memory is required to store the upsampled reconstructed image prior to accumulation with the decoded difference image (plus extra precision bits for accumulation);

- reference images may diverge due to IDCT differences;

- substantially more bits are needed to achieve the final stage.

C.3 Special spectral selection

In the hierarchical mode, downsampling and upsampling filtering is used at both coding and decoding levels to obtain different resolution images.

The technique described below is based on the following idea:

- the image is coded with only one given resolution (basic resolution, typically "CCIR 601 [9] 4:2:2");

- at the receiving end, instead of performing a "traditional" 8 x 8 inverse DCT followed by a downsampling filtering of the image to obtain the desired resolution, the image is decoded using a variable size inverse DCT.

This sort of "DCT filtering" allows images with different resolutions to be displayed without having to perform any postfiltering. Variable size inverse DCTs are performed by dropping out "the unused" DCT coefficients (see figure C.3).

This technique can be applied to any of the DCT coding algorithms defined in ISO/IEC DIS 10918 [13], but due to the variable size inverse DCT, it is more efficient when used in combination with the progressive coding algorithm, as shown below.

NOTE: See also Annex H (informative) to this ETS.

C.3.1 Coding

When considering the encoding progressive DCT algorithm (described in ISO/IEC DIS 10918 [13]), a specific definition of the spectral bands can be used in order to improve the efficiency of the technique. See also Annex A, subclause A.6.2.

The benefits of "well specified" spectral bands are that the receiver will only have to decode the spectral bands containing the DCT coefficients required to perform a given size inverse DCT.

Figure C.3 shows the DCT coefficients required to perform the different size inverse DCT used to display a 4:2:2 coded image with QCIF, CIF, 2:1:1 and 4:2:2 resolutions.

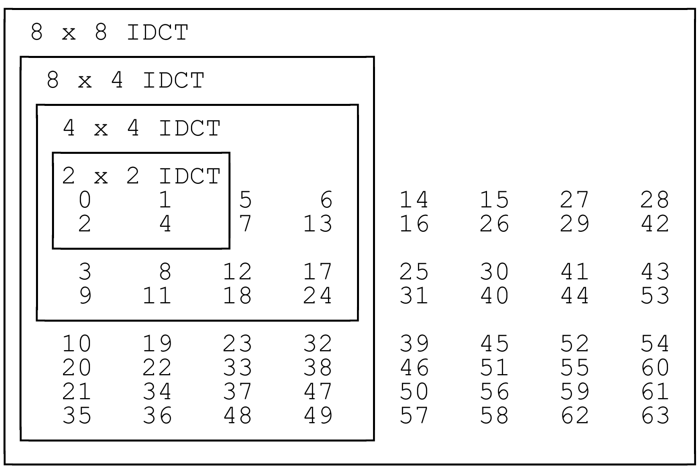


Figure C.3: DCT coefficients

Required to perform the different size inverse DCT used to display a "CCIR 601 [9] 4:2:2" coded image with QCIF, CIF, "CCIR 601 [9] 2:1:1" and "CCIR 601 [9] 4:2:2" resolutions.

According to the display resolutions mentioned above, a "CCIR 601 [9] 4:2:2" image can then be coded with the following spectral bands:

- first spectral band: coefficient 0 (DC coefficient only, since mixing of DC and AC coefficients in the same spectral band is not allowed according to ISO/IEC DIS 10918 [13]);

- second spectral band: from coefficients 1 to 4;

- third spectral band: from coefficients 5 to 24;

- fourth spectral band : from coefficients 25 to 49;

- fifth spectral band : from coefficients 50 to 63.

Considering these proposed spectral bands, the image is coded using the "classical" progressive coding algorithm defined by ISO/IEC DIS 10918 [13].

This definition of the spectral bands is not mandatory, other spectral bands can be defined. Spectral bands can be added (within each spectral band defined above) in order to improve the progressive aspect of the display. As another example, the third spectral band can be reduced from coefficients 1 to 13 in order to reduce from 9 to 1 the number of extra coefficients included in the spectral band due to the zigzag scanning of DCT coefficients. In this case, the transmission time of the three first spectral bands is noticeably reduced with only minor degradations of the image quality (due to the absence of AC coefficients 17,18 and 24).

C.3.2 Decoding

At the decoding level, the size of the inverse DCT is adjusted to fit with the desired image resolution. The inverse DCT calculation involves only the required DCT coefficients, as described in figure C.3. As a result, only relevant spectral bands have to be decoded.

Extra coefficients included in the spectral bands, due to the zigzag scanning of the DCT coefficients, are dropped out by the decoder before performing the inverse DCT.

The display of a "CCIR 601 [9] 4:2:2" coded image with the resolution mentioned above required the following processing:

- **icons :**

  first spectral band decoded (DC coefficient)

  1 x 1 inverse DCT

- **180 pixels x 144 lines resolution (QCIF format):**

  two first spectral bands decoded

  2 x 2 inverse DCT

- **360 pixels x 288 lines resolution (CIF format):**

  three first spectral bands decoded

  4 x 4 inverse DCT

- **360 pixels x 576 lines resolution (2:1:1 format):**

  four first spectral bands decoded

  8 x 4 inverse DCT

- **720 pixels x 576 lines resolution (4:2:2 format):**

  all the spectral bands decoded

  8 x 8 inverse DCT

Other processing can be envisaged. For example, a CIF resolution image can be obtained by decoding only the two first spectral bands and by performing a 4 x 4 inverse DCT on these coefficients. This produces a reasonable image in terms of quality and resolution for a very low transmission and computation cost.

In order to take full benefit of the use of spectral bands, the inverse DCT can be calculated using an accumulation technique on the elementary sub-images corresponding to each DCT coefficient (depending on the number of coefficients to be processed in each spectral band).

Compared to the hierarchical mode, the main advantage of this technique is its simplicity.

- the coding technique is a "classical", non-hierarchical technique. So no increase in the calculation power is required and also no additional memory is required;

- the complexity of the decoder is adapted to the resolution of the terminal. Depending on its resolution, the terminal only decodes the necessary spectral bands and performs an adjusted size inverse DCT. So low resolution terminals only have to perform a simple inverse DCT (typically 2 x 2 or 4 x 4 inverse DCT for a 4:2:2 basic resolution);

- in addition, no post-filtering is required in the decoder.

Since the number of inverse DCTs with different sizes is limited, the range of resolution covered by this technique is also limited. The lowest available resolution (corresponding to 1 x 1 inverse DCT) is 1/64 of the upper resolution (basic resolution). However, 4:2:2, 2:1:1, CIF and QCIF resolutions can be handled from within the same image.

Annex D (informative): Coding examples

D.1 Introduction

This informative annex provides examples for coding according to the syntax defined by this ETS.

D.1.1 Example 1

In this example the simplest way of transmitting a photographic image is assumed, using as many default values of attributes, parameters and tables as possible.

The transmission sequence of two consecutive, but separate photographic frames is coded. Both Videotex frames include photographic information only and for simplicity no alpha-mosaic or geometric data. The photographic image of both frames should fill the entire DDA. The compressed pictures each have a size of 10 000 bytes. The picture data is sent in blocks with a maximum size of 4 000 bytes, which allows an interruption in the transmission if required.

Picture 1:

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 4/11>

Header information

<PDE_1>
<5/1>

Reset to Default

<RTD> <dev>
<2/0 3/0> <4/5 0/1 0/1>

Clear Photo-area

<CPA> <cle>
<2/1 3/5> <4/5 0/1 0/1>

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 7/14 5/1>

<PDE_1>
<5/3>

(2 000 bytes of Image Data)

Picture 2:

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 4/6>

Header information

<PDE_1>
<5/1>

Clear Photo-area

<CPA> <cle>
<2/1 3/5> <4/5 0/1 0/1>

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 7/14 5/1>

<PDE_1>
<5/3>

(2 000 bytes of Image Data)

D.1.2 Example 2

In this example a more complex way of transmitting a photographic image is assumed, using some customised tables.

The transmission sequence of two consecutive, but separate photographic frames is coded. Both Videotex frames should include photographic information only and for simplicity no alpha-mosaic or geometric data. The photographic image of both frames should fill the entire DDA. Frame 1 should include downloading of one custom quantisation table and two sets of custom Huffman tables. A set of Huffman tables includes both AC and DC Huffman tables for a given image component. A separate PE will be used for downloading the new custom tables. Frame 2 should not include the downloading of any tables, but uses the tables downloaded with frame 1.

The compressed pictures each have a size of 10 000 bytes. The picture data is sent in blocks with a maximum size of 4 000 bytes, which gives a shorter delay if an interruption in the transmission is required.

In this example a mixture of custom and default tables are used. The parameter reset to default instructs the application to load all the default tables, subsequently the encoding table management parameter informs the application that some custom tables will be sent.The length of the tables is assumed to be 2 000 bytes.

For the second picture the encoding table management parameter is used to inform the application that the "in use" tables are still valid.

Picture 1:

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 4/12>

Header information

(44 bytes of Photographic Header Data)

<PDE_1>
<5/1>

Reset to Default

<RTD> <dev>
<2/0 3/0> <4/5 0/1 0/1>

Clear Photo-area

<CPA> <cle>
<2/1 3/5> <4/5 0/1 0/1>

Encoding table management

<ETM> <ttp> <tid> <tst> <ETM> <ttp> <tid> <tst> <ETM> <ttp> <tid> <tst>
<2/4 3/1> <4/4 0/1 0/1> <4/0 0/1 0/7> <4/4 0/1 0/3>
<2/4 3/1> <4/4 0/1 0/2> <4/0 0/1 5/5> <4/4 0/1 0/3>
<2/4 3/1> <4/4 0/1 0/2> <4/0 0/1 5/6> <4/4 0/1 0/3>

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Photographic Data (with tables))

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Photographic Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Photographic Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 7/14 5/1>

<PDE_1>
<5/3>

(2 000 bytes Photographic Data)

Picture 2:

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 4/7>

Header information

(39 bytes of Photographic Header Data)

<PDE_1>
<5/1>

Clear Photo-area

<CPA> <cle>
<2/1 3/5> <4/5 0/1 0/1>

Source table specification

<ETM> <ttp> <tid> <tst> <ETM> <ttp> <tid> <tst> <ETM> <ttp> <tid> <tst>
<2/4 3/1> <4/4 0/1 0/1> <4/0 0/1 0/7> <4/4 0/1 0/2>
<2/4 3/1> <4/4 0/1 0/2> <4/0 0/1 5/5> <4/4 0/1 0/2>
<2/4 3/1> <4/4 0/1 0/2> <4/0 0/1 5/6> <4/4 0/1 0/2>

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 7/14 5/1>

<PDE_1>
<5/3>

(2 000 bytes of Image Data)

D.1.3 Example 3

In this example a single more complex Videotex frame including alpha-mosaic and photographic data will be coded. First, the alpha-mosaic part of the frame is transmitted, then by switching to the photographic mode a photo-area is defined where a baseline JPEG image according to CCIR 601 [9] 4:2:2 will be placed. The placement into the photo-area should be according to the default values. The compressed pictures each have a size of 10 000 bytes. The picture data is sent in blocks with a maximum size of 4 000 bytes, which gives a shorter delay if an interruption in the transmission is required.

Alpha-mosaic:

<US> <TEXT and Control codes>
<1/14> <4/1 4/13 270 7/4 6/5 7/3 7/4 0/10 0/13> "AM test"

Picture:

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 5/10>

Header information

(26 bytes of header information)

<PDE_1>
<5/1>

Reset to Default

<RTD> <dev>
<2/0 3/0> <4/5 0/1 0/1>

Clear Photo-area

<CPA> <cle>
<2/1 3/5> <4/5 0/1 0/1> "TRUE"

Source Picture Dimension.

<PDS> <nph> <npv>
<2/2 3/1> <4/0 0/2 0/5 0/0> <4/0 0/2 0/3 6/0> "640*480 pixel"

Source Pixel Density.

<PID> <stf>
<2/2 3/2> <4/4 0/1 0/1> "4:2:2 625 lines"

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/3 7/13 4/1>

<PDE_1>
<5/2>

(4 000 bytes of Image Data)

ISO 9281

<PCD> <PM> <PI> <LI>
<1/11 7/0> <2/3> <4/0> <7/15 6/1 7/14 5/1>

<PDE_1>
<5/3>

(2 000 bytes of Image Data)

D.2 Image positioning examples

In the examples only those sub-parameters needed to position the picture on the screen are included.

D.2.1 Example 1: 640 X 480 picture inside DDA

<FSD> <ful> = "false"

<ASR> <arah> = 4
<araw> = 3

<LOC> <loch> = 0
<locv> = 0

<PAS> <sizw> = 1
<sizh> = 0.75

<PPL> either (lower left corner ref.):

<refh> = 0
<refv> = 479
<offh> = 0
<offv> = 0

or (upper left corner ref.):

<refh> = 0
<refv> = 0
<offh> = 0
<offv> = 0.75

or:

default

<PDS> <nph> = 640
<npv> = 480

<PID> <stf> = 4:2:2, 625 lines

D.2.2 Example 2: 720 X 576 picture full screen

<FSD> <ful> = "true"

<ASR> <araw> = 5
<arah> = 4

<LOC> <loch> = 0
<locv> = 0

<PAS> <sizw> = 1
<sizh> = 0.8

<PPL> either (lower left corner ref.):

<refh> = 0
<refv> = 575
<offh> = 0
<offv> = 0

or (upper left corner ref.):

<refh> = 0
<refv> = 0
<offh> = 0
<offv> = 0.8

or:

default

<PDS> <nph> = 720
<npv> = 576

<PID> <stf> = 4:2:2, 625 lines

D.2.3 Example 3: 720 X 346 picture covering upper 60% of full screen

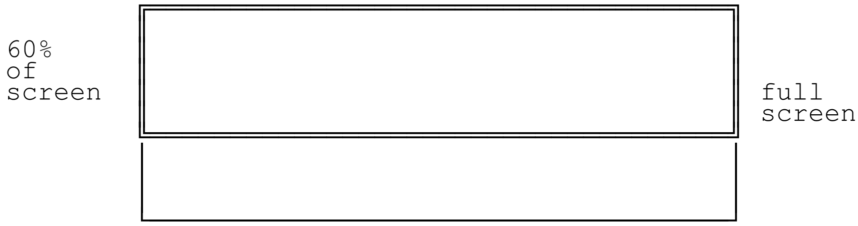


Figure D.1: 60% of the screen is filled with a Photographic Image

<FSD> <ful> = "true"

<ASR> <araw> = 5
<arah> = 4

<LOC> <loch> = 0
<locv> = 0.32

<PAS> <sizw> = 1
<sizh> = 0.48

<PPL> either (lower left corner ref.):

<refh> = 0
<refv> = 345
<offh> = 0
<offv> = 0

or (upper left corner ref.):

<refh> = 0
<refv> = 0
<offh> = 0
<offv> = 0.48

<PDS> <nph> = 720
<npv> = 346

<PID> <stf> = 4:2:2, 625 lines

D.3 Example for Source Picture Comments (PCT)

In the following a fictitious example of a possible use of the Source Picture Comment (PCT) is provided.

D.3.1 An application scenario

Assume that the photographic picture coded according to this standard of the "Mona Lisa" should be decorating the upper right corner of a Videotex frame describing the services of the "Beauty Salon Butterfly". Assume that no text covering the "Mona Lisa" as a painting (or as a source image) is displayed in Videotex Alpha-Mosaic mode, since this would be disturbing for the main text message of the frame. Also assume that "Studio Alpha", however, which sells this picture to several information providers (including the "Paris Tourist Board", the "Leonardo da Vinci Foundation of Homeless Painters") insist, as the picture source provider, to put, as a general policy of the Studio, in all applications a source picture comments linked to the "Mona Lisa" picture, for copyright and reference purposes.

The description of the logical record structure is provided by the source picture provider. If the use of the PCT parameter is permitted by the Photographic profile, then the actual use of PCT is left to the retrieving terminal. Thus, one class of terminals may simply recognize the code of the PCT parameter, and those terminals may simply wish to skip the number of bytes that is given for the length of the picture comment. Another class of terminals, which are for example linked to a small local picture data base, where images get downloaded to and from the central Videotex system, may wish to use the content of the picture comments as additional keywords.

For example, in the above case, the Studio might wish to know whether their source picture was incorporated into the Videotex frame of the beauty salon or not, and, if so, which version of their "Mona Lisa" was used. Further, the "Paris Tourist Board" may download and use the "Mona Lisa" in their own local data base.

The use of this parameter is intended to be extremely flexible.

D.3.2 Sample logical record of a Source picture:

Assume that the logical sequence of the fields in the record are fixed. Since the field-length is variable, each field is preceded by a length indicator, which is 1 byte long (thus each field can be maximally 256 bytes long, assuming binary coding for the length). Assume the following logical record:

- Picture ref. number (0): 123456
- Title of source (1): "Mona Lisa"
- Painter (2): Leonardo da Vinci
- Painted (3): 1502-1506
- Classification (4): Italian master
- Location (5): Louvre, Paris, France
- Reproduction date (6): 15/12/1995
- Photographer (8): X.Y.
- Copyright photo (9): Studio Alpha, Sollnerstr 10, 8000 München 70, Germany

Assume the following physical coding according to 7-bit CCITT Recommendation T.51 [7], where it is assumed that no control characters (except for code extension according to ISO 2022 [10]) are needed to "process" this comment by the receiving terminal.

Table D.1: Example of source picture comments coding

Field no.(i)	Text (i)	length code (i)	CCITT T.51 Text Code (i)
0	123456	X '06'	3/1 3/2 3/3 3/4 3/5 3/6
1	Mona Lisa	X '09"	4/13 6/15 5/14 6/1 2/0 4/12 6/9 7/3 6/2
2	Leonardo da Vinci	X '11'	4/12 6/5 6/15 6/14 6/1 7/2 6/4 6/15 2/0 6/4 6/2 2/0 5/6 6/9 6/14 6/3 6/9
etc.			
9	Studio Alpha, Sollnerstr. 10, 8000 München 70, Germany	X '35'	5/3 7/4 6/4 6/9 6/15 2/0 4/1 6/12 7/0 6/8 6/1 2/0 2/12 2/0 5/3 6/15 6/12 6/12 6/14 6/5 7/2 2/14 2/0 3/1 3/0 2/12 2/0 3/8 3/0 3/0 3/0 2/0 4/13 1/11 6/14 4/8 0/15 7/5 6/14 6/3 6/8 6/5 6/14 2/0 3/7 3/0 2/12 2/0 4/7 6/5 7/2 6/13 6/1 6/14 7/9

Then the picture comment character string is the following:

<PCT><cmt>

whereby: <PCT> = 2/1 3/0

<cle> = ...7-bit code...

<cid> = ...T.51...

In the JPEG Data stream: search for the comment (COM) marker-code (x'FFFE'):

<length><value>(string)

whereby <length> = Length of the "comment character string"(150 Bytes)
= 04/02 01/06

<value>(string) = <Length code (0)><Text code (0)>
<Length code (1)><Text code (1)>
... <Length code (9)><Text code (9)>

NOTE: The COM string in the JPEG assumes 8-bit "byte", through the text string was coded for transmission in the 7-bit "in-use" form of CCITT Recommendation T.51 [7]. Thus, the most significant bit of each JPEG COM Byte is remains insignificant.

Annex E (informative): ETSI/CCITT Cross-reference list

Table E.1: ETSI/CCITT cross reference list

European Telecommunication Standard	CCITT Recommendation
ETS 300 072 Videotex presentation layer protocol. Videotex Presentation layer data syntax	CCITT Recommendation T.101 (1988) International interworking for Videotex: Annex C
ETS 300 076 Videotex: Terminal Facility Identifier	CCITT Recommendation T.101 (1988) International interworking for Videotex Annex C
ETS 300 079 ISDN syntax-based Videotex Videotex services using ISDN	CCITT Draft Recommendation T.102 (1992) Protocols for syntax-based
ETS 300 177 Videotex: Photographic Mode	CCITT Recommendation T.101 Draft Annex E (1992) Videotex: Photographic Mode

Annex F (normative): Encoding parameters values for the 2:1:1 derived from CCIR Recommendation 601, Part 1

F.1 Introduction

Due to the fact that one of the compatible profiles defined in the Clause 11 of this ETS has a resolution based on the "2:1:1" format and that furthermore CCIR Recommendation 601 [9] does not fully define the "2:1:1" format, it is necessary to give in this annex, the technical specifications.

Also, this is absolutely necessary to ensure that any "2:1:1" image can be displayed on any system, to avoid annoying artifacts if the display characteristics of the terminal differ from the transmitted source image. The process in which the terminal tries to cope the mapping of the characteristics of the source image to the characteristics of it's own display capability, is treated in Annex B (display rendering guidelines).

The technical specifications of the 2:1:1 is based on the general principles outlined in CCIR Recommendation 601 [9] and is directly derived from the parameters defined for 4:4:4 and 4:2:2. So, the needed complements to the CCIR Recommendation 601 [9] for a definition of the 2:1:1 format are given in this Annex.

F.2 Encoding parameters for 2:1:1

F.2.1 Main body of CCIR Recommendation 601, Part 1

Paragraphs 1, 2 and 3 are fully applicable to the new set of parameters.

Paragraph 4 defines the encoding parameter values for the 4:2:2 member. This is still valid as a reference for the derivation of the 2:1:1 member. This leads to the definition of a new set of values as outlined in table F.1 of this Annex.

NOTE: The derivation is straightforward for the 625/50 system. However, for the 525/60 system some precautions should be taken (see table F.1).

F.2.2 Annex I of the CCIR Recommendation 601, Part 1

The tentative specification of the 4:4:4 member of the family is still valid as a reference.

F.2.3 Annex II of the CCIR Recommendation 601, Part 1

F.2.3.1 Relationship of active line to analogue synchronisation reference

The relationship between 360 digital active line luminance samples and the analogue synchronizing references for the 625-line systems and the 525-line systems is shown in table F.2 of this Annex.

The respective numbers of colour-difference samples can be obtained by dividing the number of luminance samples by two. The (6, 66) and (8, 61) were chosen symmetrically to position the digital active line within the permitted variations. They do not form part of the digital line specification and relate only to the analogue interface.

F.2.3.2 Definition of the digital signals Y, C_R, C_B, from the primary (analogue) signals E'_R, E'_G and E'_B

The definition applied to the 4:2:2 shall be fully applicable to the 2:1:1.

F.2.4 Annex III of the CCIR Recommendation 601

The filtering characteristics are modified according to tables F.3, F.4 and F.5 of this annex.

They are directly derived from the 4:2:2 filtering specifications.

NOTE: The passband ripple tolerance and the passband group-delay tolerance are specified in full respect to CCIR Recommendation 601 [9], but they might be found too stringent to certain applications. It is felt that the tolerances should be made more flexible for the 2:1:1. This flexibility should be specified. It is proposed in a first stage to permit the doubling of the tolerances on the passband ripple and group-delay.

Table F.1: Encoding parameter values for the 2:1:1 member

Notes to table F.1:

(1) See CCIR report 624 (Characteristics of TV systems), table 1.

(2) In 525/60 system the odd number of luminance samples per total line means a non-integer number of colour difference samples.

(3) In 525/60 system, the respect of item 3 implies half period offset of colour-difference sampling clock every line, in order to get co-sited samples (C_B C_R and Y) with the beginning of the digital active line.

(4) The sampling frequency of 6.75 MHz (luminance) is an integer multiple of 2.25 MHz, the lowest common multiple of the line frequencies in 525/60 and 625/50 systems.

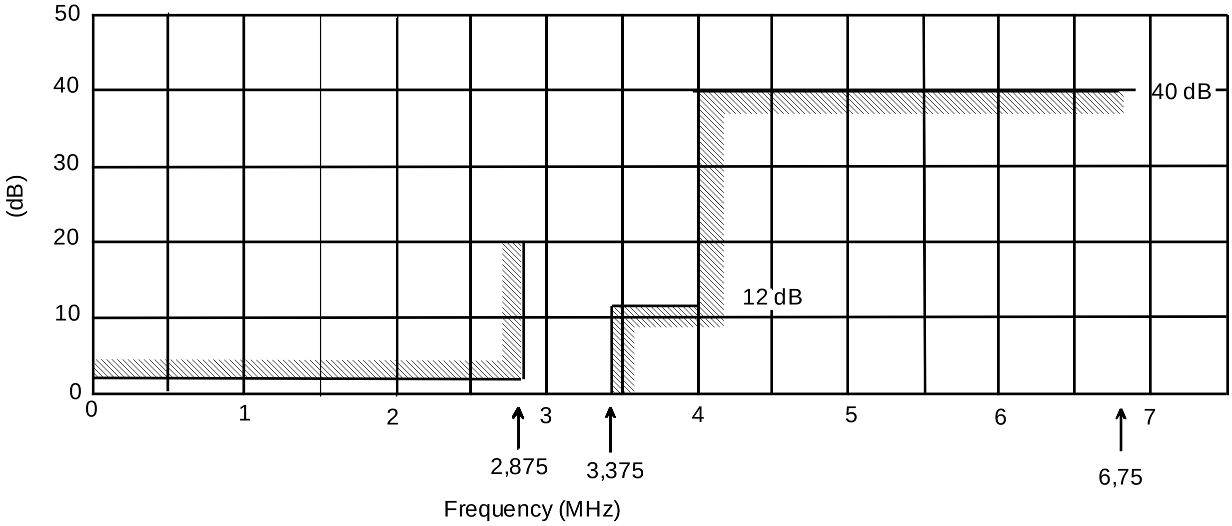
Parameters	525-line, 60 field/s(1) systems	625-line, 50 field/s(1) systems
1.Coded signals: Y, CR, CB	These signals are obtained from gamma pre-corrected signals, namely: E'Y, E'R - E'Y, E'B - E'Y (Annex II, ° 2 refers in Rec.601)	
2.Number of samples per total line: - luminance signal (Y) - each colour-difference signal (CR, CB)	 429 215 (2)	 432 216
3.Sampling structure	Orthogonal, line, field and frame repetitive, CR and CB samples co-cited with (3) even (2nd,4th,...) | odd (1st,3rd,...) Y samples in each line	
4.Sampling frequency: - luminace signal - each colour-difference signal	6,75 MHz (4) 3,375 MHz The tolerance for sampling frequencies should coincide with the tolerance for the line frequency of the relevant colour television standard	
5.Form of coding	Uniformly quantized PCM, 8bits per sample, for the luminance signal and each colour-difference signal	
6.Number of samples per digital active line: - luminance signal - each colour-difference signal	360 180	
7.Analogue-to-digital horizontal timing relationship: - from end of digital active line to OH	8 luminance clock periods	6 luminance clock periods
8.Correspondance between video signal levels and quantization levels: - scale - luminance signal - each colour-difference signal	0 to 255 220 quantization levels with the black level corresponding to level 16 and the peak white level 235. The signal level may occasionally excurse beyond level 235 225 quantization levels in the centre part of the quantization scale with zero signal corresponding to level 128	
9.Code-word usage	Code-words corresponding to quantization levels 0 and 255 are used exclusively for synchronization. Levels 1 to 254 are available for video	

Table F.2: Relationship of digital active line to analogue synchronisation reference

525-line, 60 fields/s systems	61*T*	360*T*	8*T*	
O*H* (Leading edge of line syncs., half-amplitude reference)		Digital active-line period	O*H*	Next line
625-line, 50 field/s systems	66*T*	360*T*	6*T*	

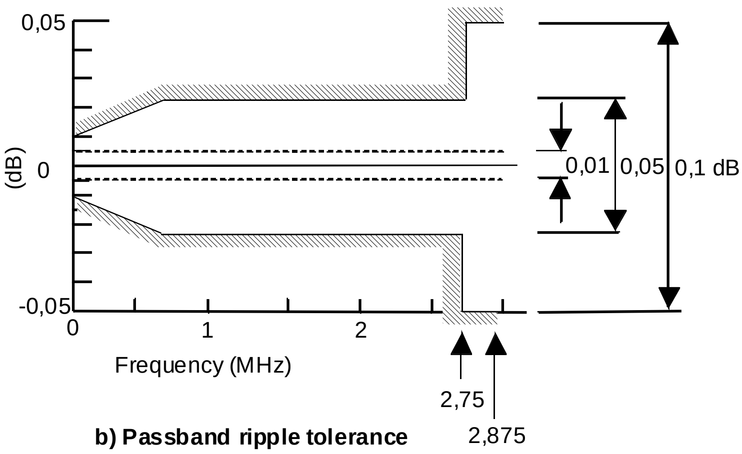
T: one luminance sampling clock period (128 ns nominal).

Table F.3: Specification for a luminance or RGB signal filter used when sampling at 6,75 MHz

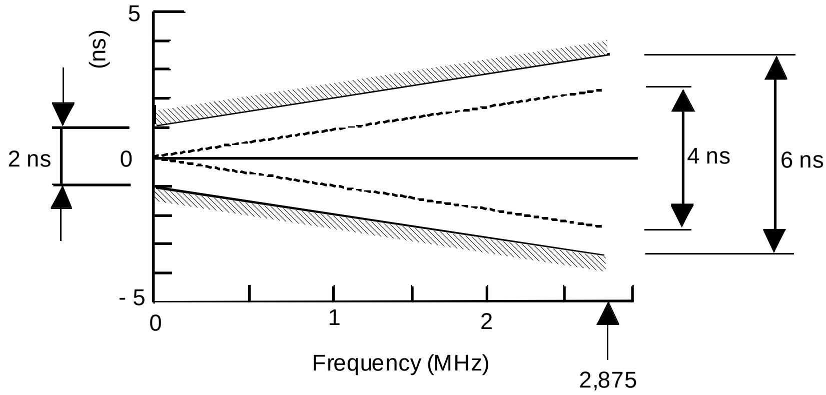


a) Template for insertion loss/frequency characteristic

NOTE: The lowest indicated values in b) and c) are for 1 kHz (instead of 0 MHz).

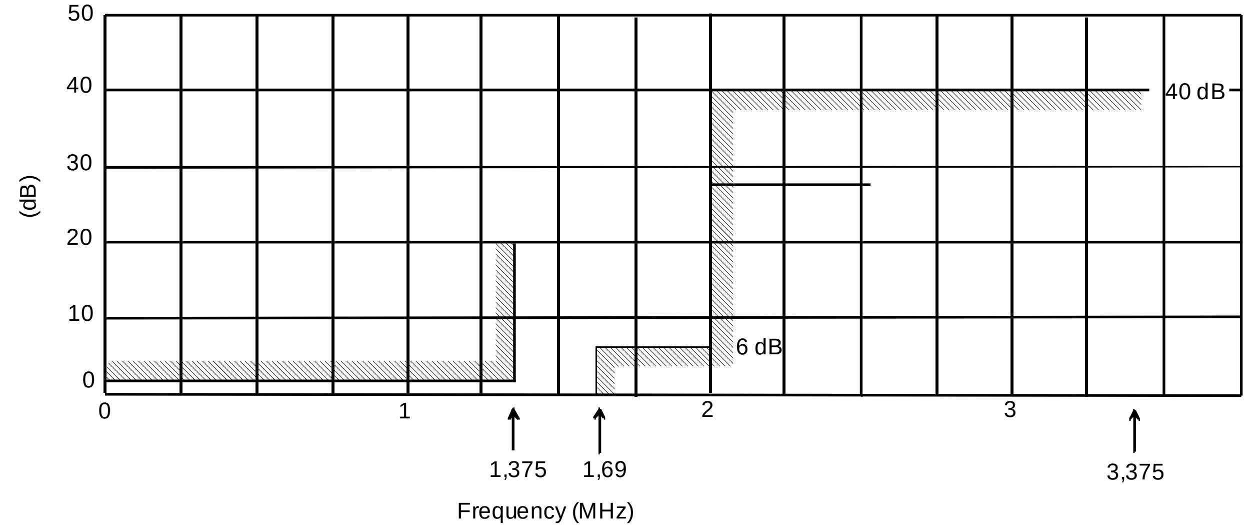


b) Passband ripple tolerance



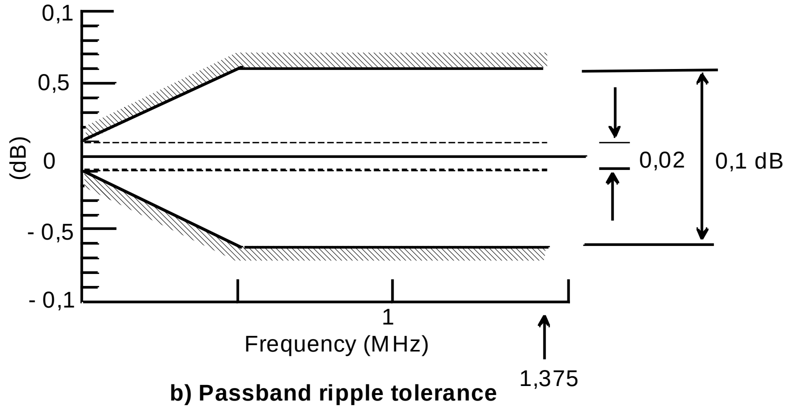
c) Passband group-delay tolerance

Table F.4: Specification for a colour-difference signal filter used when sampling at 3,375 MHz

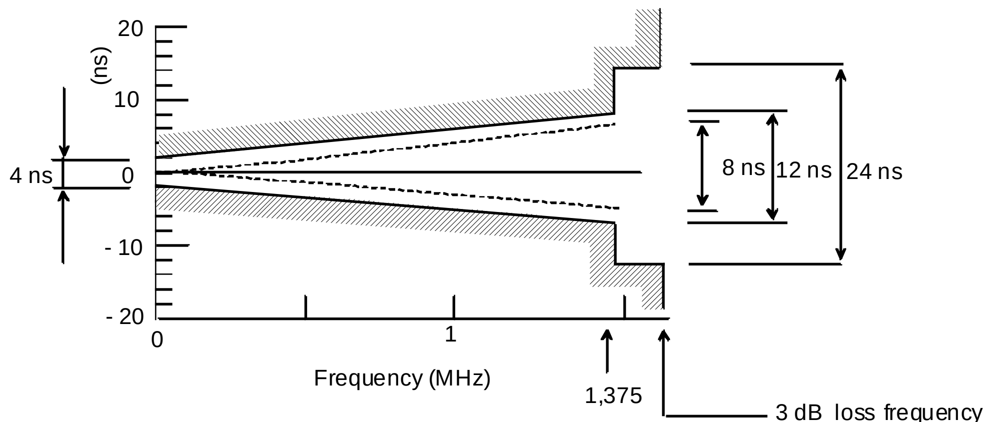


a) Template for insertion loss/frequency characteristic

The lowest indicated values in b) and c) are for 1 kHz (instead of 0 MHz).

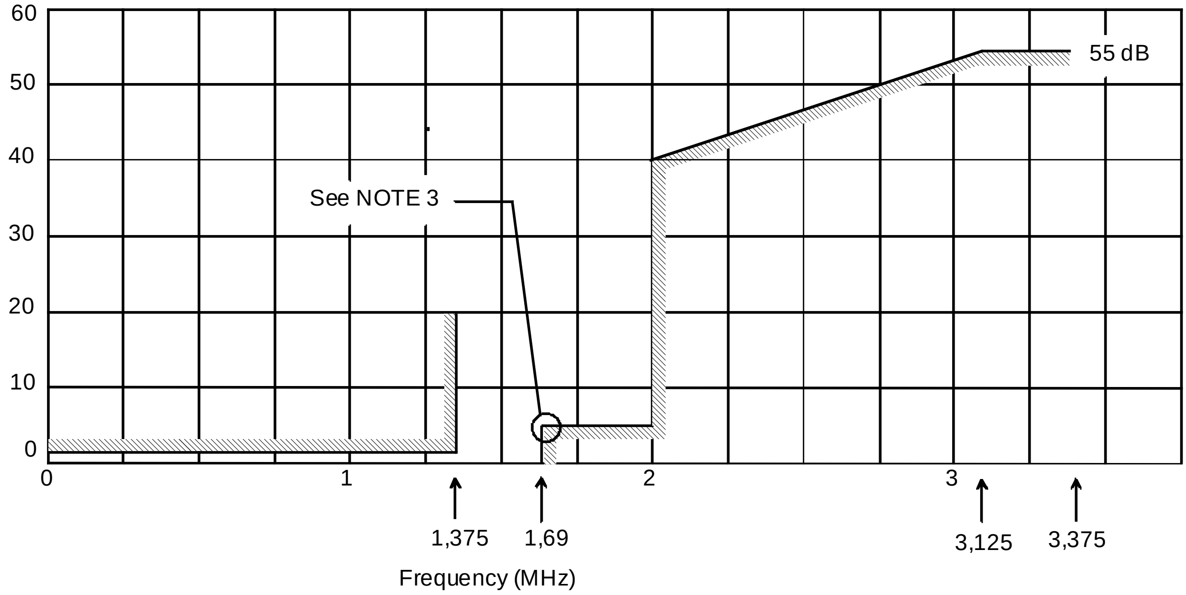


b) Passband ripple tolerance

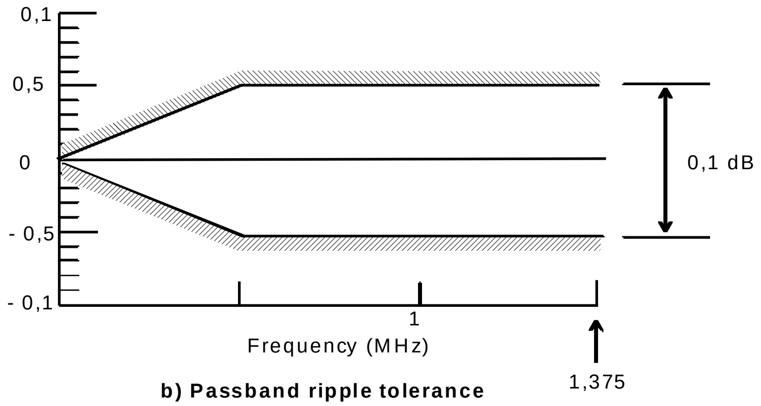


c) Passband group-delay tolerance

Table F.5: Specification for a digital filter for sampling-rate conversion from 2:2:2 to 2:1:1 colour-difference-signals



a) Template for insertion loss frequency characteristic



b) Passband ripple tolerance

Notes to tables F3, F4 and F5:

NOTE 1: Ripple and group delay are specified relative to their value at 1 kHz. The full lines are practical limits and the dashed lines give suggested limits for the theoretical design.

NOTE 2: In the digital filter, the practical and design limits are the same. The delay distortion is zero, by design.

NOTE 3: In the digital filter (table F5), the amplitude/frequency characteristic (on linear scales) should be skew-symmetrical about the half-amplitude point, which is indicated on the figure.

NOTE 4: In the proposals for the filters used in the encoding and decoding processes, it has been assumed that, in the post-filters which follow digital-to-analogue conversion, correction for the (sin x/x) characteristic of the sample-and-hold circuits is provided.

Annex G (normative): Translation modes

G.1 Mode 0 (No translation, full transparency)

Under this scheme, no translation of data shall be performed.

G.2 Mode 2 (3-in-4 coding)

Each group of three bytes in the photographic data is mapped onto four bytes for transmission as shown in table G.1. Any remaining group of one or two bytes at the end of a block of data is mapped onto two or three bytes respectively, with undefined bits set to zero.

Table G.1: 3-in-4 coding scheme

Photographic Data	3 bytes	3 bytes	1, 2 or 3 bytes
Transmitted	4 bytes	4 bytes	2, 3 or 4 bytes

Within each group, the bits are mapped as follows, where bxy denotes bit y of byte x in the Photographic Data. Bit 8 is not taken into account by this scheme, but may be determined by characteristics of the transmission path in use (for example if parity is required).

a) Three bytes of Photographic Data

Transmitted	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	x	1	b17	b16	b27	b26	b37	b36
2nd byte	x	1	b15	b14	b13	b12	b11	b10
3rd byte	x	1	b25	b24	b23	b22	b21	b20
4th byte	x	1	b35	b34	b33	b32	b31	b30

b) Two bytes of photographic data at the end of a sequence

Transmitted	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	x	1	b17	b16	b27	b26	0	0
2nd byte	x	1	b15	b14	b13	b12	b11	b10
3rd byte	x	1	b25	b24	b23	b22	b21	b20

c) One byte of photographic data at the end of a sequence

Transmitted	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	x	1	b17	b16	0	0	0	0
2nd byte	x	1	b15	b14	b13	b12	b11	b10

G.3 Mode 4 (Shift scheme - 7 bits)

In this scheme, bytes of photographic data are each mapped onto one or two bytes of transmitted data, as shown table G.2; in mode 4, the most significant bit of each byte is not taken into account.

NOTE: Most of the conversions are optional. A decoder should be prepared to accept any mixture of converted or unconverted data in these cases.

Table G.2: Code conversion for 7 bits shift scheme (mode 4)

Photographic Data	Sender's Conversion	O/M	Transmitted Data
0/0-1/14	7/14, x+80 (X+50 hex)	O	7/14, 5/0-6/14
01/15	7/14, 6/15	M	7/14, 6/15
2/0	7/13	O	7/13
2/1-7/10	None	-	2/1-7/10
7/11-7/15	7/11, x-88 (X-58 hex)	M	7/11, 2/3-2/7
8/0-13/0	7/11, x-88 (X-58 hex)	M	7/11, 2/8-7/8
13/1-15/15	7/14, x-80 (X+50 hex)	M	7/14. 2/1-4/15
-: Irrelevant O: Optional M: Mandatory			
NOTE: In mode 4 the transmitted bytes may have the most significant bit set.			

Annex H (informative): Huffman tables for the "special spectral selection"

H.1 Introduction

Huffman tables are defined as it is done in the interchange format defined in JPEG, in terms of:

- a 16 bytes list giving the number of codes of each codelength from 1 to 16 (list of codelength);

- a list of 8 bits values which are assigned to each code. The values are placed in the list in order of increasing codelength (list of values).

These two lists are followed by a table specifying the Huffman codes. Annex C of the JPEG specification explains how to build a Huffman code table from the two lists described above.

The Huffman codes are given as 16 bits hexadecimal values, with the root of the code placed toward the MSB. Only the number of bits defined in "codelength" have to be considered.

For example, considering the luminance DC differences Huffman codes:

Table H.1

Category	Codelength	Codeword
0	3	4000 0100 0000 0000 0000 codeword=010
11	9	FF00 1111 1111 0000 0000 codeword=1111 1111 0

H.2 Spectral bands

Luminance: (0,0) (1,5) (6,14) (15,63)

Chrominance: (0,0) (1,5) (6,63)

H.3 Luminance DC differences

H.3.1 List of codelengths

0 1 5 1 1 1 1 1 1 0 0 0 0 0 0 0

H.3.2 List of values

3 0 1 2 4 5 6 7 8 9 10 11

Table H.2

Category	Codelength	Codeword
0	3	4000
1	3	6000
2	3	8000
3	2	0
4	3	A000
5	3	C000
6	4	E000
7	5	F000
8	6	F800
9	7	FC00
10	8	FE00
11	9	FF00

H.4 Chrominance DC differences

H.4.1 List of codelengths

0 2 3 1 1 1 1 1 1 1 0 0 0 0 0 0

H.4.2 List of values

0 1 2 3 4 5 6 7 8 9 10 11

Table H.3

Category	Codelength	Codeword
0	2	0
1	2	4000
2	3	8000
3	3	A000
4	3	C000
5	4	E000
6	5	F000
7	6	F800
8	7	FC00
9	8	FE00
10	9	FF00
11	10	FF80

H.5 Luminance AC coefficients

H.5.1 List of codelengths

0 2 1 2 3 4 7 4 6 6 6 3 1 1 1 129

H.5.2 List of values

1 2 3 4 17 5 18 33 6 49 64 80 0 16 19 32 48 65 81 7 34 97
113 20 50 96 129 145 161 8 35 66 112 177 193 21 82 98 114 128
209 36 51 130 144 225 240 9 10 22 23 24 25 26 37 38 39 40 41
42 52 53 54 55 56 57 58 67 68 69 70 71 72 73 74 83 84 85 86
87 88 89 90 99 100 101 102 103 104 105 106 115 116 117 118 119
120 121 122 131 132 133 134 135 136 137 138 146 147 148 149 150
151 152 153 154 160 162 163 164 165 166 167 168 169 170 176 178
179 180 181 182 183 184 185 186 192 194 195 196 197 198 199 200
201 202 208 210 211 212 213 214 215 216 217 218 224 226 227 228
229 230 231 232 233 234 241 242 243 244 245 246 247 248 249 250

Table H.4

Values	Run/Size	Codelength	Codeword
0	0/0	7	E800
1	0/1	2	0
2	0/2	2	4000
3	0/3	3	8000
4	0/4	4	A000
5	0/5	5	C000
6	0/6	6	D800
7	0/7	8	F600
8	0/8	10	FD00
9	0/9	16	FF7E
10	0/A	16	FF7F
16	1/0	7	EA00
17	1/1	4	B000
18	1/2	5	C800
19	1/3	7	EC00
20	1/4	9	FA00
21	1/5	11	FE80
22	1/6	16	FF80
23	1/7	16	FF81
24	1/8	16	FF82
25	1/9	16	FF83
26	1/A	16	FF84
32	2/0	7	EE00
33	2/1	5	D000
34	2/2	8	F700
35	2/3	10	FD40
36	2/4	12	FF40
37	2/5	16	FF85
38	2/6	16	FF86
39	2/7	16	FF87
40	2/8	16	FF88
41	2/9	16	FF89
42	2/A	16	FF8A
48	3/0	7	F000
49	3/1	6	DC00
50	3/2	9	FA80
51	3/3	12	FF50
52	3/4	16	FF8B
53	3/5	16	FF8C
54	3/6	16	FF8D
55	3/7	16	FF8E
56	3/8	16	FF8F
57	3/9	16	FF90
58	3/A	16	FF91
64	4/0	6	E000
65	4/1	7	F200
66	4/2	10	FD80
67	4/3	16	FF92
68	4/4	16	FF93
69	4/5	16	FF94
70	4/6	16	FF95
71	4/7	16	FF96
72	4/8	16	FF97
73	4/9	16	FF98
74	4/A	16	FF99
80	5/0	6	E400
81	5/1	7	F400
82	5/2	11	FEA0
83	5/3	16	FF9A
84	5/4	16	FF9B
85	5/5	16	FF9C
86	5/6	16	FF9D
87	5/7	16	FF9E
88	5/8	16	FF9F
89	5/9	16	FFA0
90	5/A	16	FFA1
96	6/0	9	FB00
97	6/1	8	F800
98	6/2	11	FEC0
99	6/3	16	FFA2
100	6/4	16	FFA3
101	6/5	16	FFA4
102	6/6	16	FFA5
103	6/7	16	FFA6
104	6/8	16	FFA7
105	6/9	16	FFA8
106	6/A	16	FFA9
112	7/0	10	FDC0
113	7/1	8	F900
114	7/2	11	FEE0
115	7/3	16	FFAA
116	7/4	16	FFAB
117	7/5	16	FFAC
118	7/6	16	FFAD
119	7/7	16	FFAE

Table H.4 (continued)

120	7/8	16	FFAF
121	7/9	16	FFB0
122	7/A	16	FFB1
128	8/0	11	FF00
129	8/1	9	FB80
130	8/2	12	FF60
131	8/3	16	FFB2
132	8/4	16	FFB3
133	8/5	16	FFB4
134	8/6	16	FFB5
135	8/7	16	FFB6
136	8/8	16	FFB7
137	8/9	16	FFB8
138	8/A	16	FFB9
144	9/0	13	FF70
145	9/1	9	FC00
146	9/2	16	FFBA
147	9/3	16	FFBB
148	9/4	16	FFBC
149	9/5	16	FFBD
150	9/6	16	FFBE
151	9/7	16	FFBF
152	9/8	16	FFC0
153	9/9	16	FFC1
154	9/A	16	FFC2
160	A/0	16	FFC3
161	A/1	9	FC80
162	A/2	16	FFC4
163	A/3	16	FFC5
164	A/4	16	FFC6
165	A/5	16	FFC7
166	A/6	16	FFC8
167	A/7	16	FFC9
168	A/8	16	FFCA
169	A/9	16	FFCB
170	A/A	16	FFCC
176	B/0	16	FFCD
177	B/1	10	FE00
178	B/2	16	FFCE
179	B/3	16	FFCF
180	B/4	16	FFD0
181	B/5	16	FFD1
182	B/6	16	FFD2
183	B/7	16	FFD3
184	B/8	16	FFD4
185	B/9	16	FFD5
186	B/A	16	FFD6
192	C/0	16	FFD7
193	C/1	10	FE40
194	C/2	16	FFD8
195	C/3	16	FFD9
196	C/4	16	FFDA
197	C/5	16	FFDB
198	C/6	16	FFDC
199	C/7	16	FFDD
200	C/8	16	FFDE
201	C/9	16	FFDF
202	C/A	16	FFE0
208	D/0	16	FFE1
209	D/1	11	FF20
210	D/2	16	FFE2
211	D/3	16	FFE3
212	D/4	16	FFE4
213	D/5	16	FFE5
214	D/6	16	FFE6
215	D/7	16	FFE7
216	D/8	16	FFE8
217	D/9	16	FFE9
218	D/A	16	FFEA
224	E/0	16	FFEB
225	E/1	14	FF78
226	E/2	16	FFEC
227	E/3	16	FFED
228	E/4	16	FFEE
229	E/5	16	FFEF
230	E/6	16	FFF0
231	E/7	16	FFF1
232	E/8	16	FFF2
233	E/9	16	FFF3
234	E/A	16	FFF4
240	F/0	15	FF7C
241	F/1	16	FFF5
242	F/2	16	FFF6
243	F/3	16	FFF7
244	F/4	16	FFF8
245	F/5	16	FFF9
246	F/6	16	FFFA
247	F/7	16	FFFB

Table H.4 (concluded)

248	F/8	16	FFFC
249	F/9	16	FFFD
250	F/A	16	FFFE

H.6 Chrominance AC coefficients

H.6.1 List of codelengths

0 1 2 4 1 5 11 7 8 6 4 8 1 0 1 117

H.6.2 List of values

1 2 3 4 5 17 80 6 7 18 32 33 49 0 16 19 48 50 64 65 81 96 97
113 8 20 34 112 129 145 177 9 35 51 66 128 161 193 240 21 82 98
114 144 209 10 36 160 225 22 52 67 146 176 178 210 226 37 83 23
24 25 26 38 39 40 41 42 53 54 55 56 57 58 68 69 70 71 72 73
74 84 85 86 87 88 89 90 99 100 101 102 103 104 105 106 115 116
117 118 119 120 121 122 130 131 132 133 134 135 136 137 138 147
148 149 150 151 152 153 154 162 163 164 165 166 167 168 169 170
179 180 181 182 183 184 185 186 192 194 195 196 197 198 199 200
201 202 208 211 212 213 214 215 216 217 218 224 227 228 229 230
231 232 233 234 241 242 243 244 245 246 247 248 249 250

Table H.5

Values	Run/Size	Codelength	Codeword
0	0/0	7	DC00
1	0/1	2	0
2	0/2	3	4000
3	0/3	3	6000
4	0/4	4	8000
5	0/5	4	9000
6	0/6	5	C000
7	0/7	6	C800
8	0/8	8	F200
9	0/9	9	F900
10	0/A	11	FE80
16	1/0	7	DE00
17	1/1	4	A000
18	1/2	6	CC00
19	1/3	7	E000
20	1/4	8	F300
21	1/5	10	FD00
22	1/6	12	FF00
23	1/7	16	FF8A
24	1/8	16	FF8B
25	1/9	16	FF8C
26	1/A	16	FF8D
32	2/0	6	D000
33	2/1	6	D400
34	2/2	8	F400
35	2/3	9	F980
36	2/4	11	FEA0
37	2/5	13	FF80
38	2/6	16	FF8E
39	2/7	16	FF8F
40	2/8	16	FF90
41	2/9	16	FF91
42	2/A	16	FF92
48	3/0	7	E200
49	3/1	6	D800
50	3/2	7	E400
51	3/3	9	FA00
52	3/4	12	FF10
53	3/5	16	FF93
54	3/6	16	FF94
55	3/7	16	FF95
56	3/8	16	FF96
57	3/9	16	FF97
58	3/A	16	FF98
64	4/0	7	E600
65	4/1	7	E800

Table H.5 (continued)

66	4/2	9	FA80
67	4/3	12	FF20
68	4/4	16	FF99
37	2/5	13	FF80
38	2/6	16	FF8E
39	2/7	16	FF8F
40	2/8	16	FF90
41	2/9	16	FF91
42	2/A	16	FF92
48	3/0	7	E200
49	3/1	6	D800
50	3/2	7	E400
51	3/3	9	FA00
52	3/4	12	FF10
53	3/5	16	FF93
54	3/6	16	FF94
55	3/7	16	FF95
56	3/8	16	FF96
57	3/9	16	FF97
58	3/A	16	FF98
64	4/0	7	E600
65	4/1	7	E800
66	4/2	9	FA80
67	4/3	12	FF20
68	4/4	16	FF99
69	4/5	16	FF9A
70	4/6	16	FF9B
71	4/7	16	FF9C
72	4/8	16	FF9D
73	4/9	16	FF9E
74	4/A	16	FF9F
80	5/0	4	B000
81	5/1	7	EA00
82	5/2	10	FD40
83	5/3	15	FF88
84	5/4	16	FFA0
85	5/5	16	FFA1
86	5/6	16	FFA2
87	5/7	16	FFA3
88	5/8	16	FFA4
89	5/9	16	FFA5
90	5/A	16	FFA6
96	6/0	7	EC00
97	6/1	7	EE00
98	6/2	10	FD80
99	6/3	16	FFA7
100	6/4	16	FFA8
101	6/5	16	FFA9
102	6/6	16	FFAA
103	6/7	16	FFAB
104	6/8	16	FFAC
105	6/9	16	FFAD
106	6/A	16	FFAE
112	7/0	8	F500
113	7/1	7	F000
114	7/2	10	FDC0
115	7/3	16	FFAF
116	7/4	16	FFB0
117	7/5	16	FFB1
118	7/6	16	FFB2
119	7/7	16	FFB3
120	7/8	16	FFB4
121	7/9	16	FFB5
122	7/A	16	FFB6
128	8/0	9	FB00
129	8/1	8	F600
130	8/2	16	FFB7
131	8/3	16	FFB8
132	8/4	16	FFB9
133	8/5	16	FFBA
134	8/6	16	FFBB
135	8/7	16	FFBC
136	8/8	16	FFBD
137	8/9	16	FFBE
138	8/A	16	FFBF
144	9/0	10	FE00
145	9/1	8	F700
146	9/2	12	FF30
147	9/3	16	FFC0
148	9/4	16	FFC1
149	9/5	16	FFC2
150	9/6	16	FFC3
151	9/7	16	FFC4
152	9/8	16	FFC5
153	9/9	16	FFC6
154	9/A	16	FFC7
160	A/0	11	FEC0
161	A/1	9	FB80

Table H.5 (concluded)

162	A/2	16	FFC8
163	A/3	16	FFC9
164	A/4	16	FFCA
165	A/5	16	FFCB
166	A/6	16	FFCC
167	A/7	16	FFCD
168	A/8	16	FFCE
169	A/9	16	FFCF
170	A/A	16	FFD0
176	B/0	12	FF40
177	B/1	8	F800
178	B/2	12	FF50
179	B/3	16	FFD1
180	B/4	16	FFD2
181	B/5	16	FFD3
182	B/6	16	FFD4
183	B/7	16	FFD5
184	B/8	16	FFD6
185	B/9	16	FFD7
186	B/A	16	FFD8
192	C/0	16	FFD9
193	C/1	9	FC00
194	C/2	16	FFDA
195	C/3	16	FFDB
196	C/4	16	FFDC
197	C/5	16	FFDD
198	C/6	16	FFDE
199	C/7	16	FFDF
200	C/8	16	FFE0
201	C/9	16	FFE1
202	C/A	16	FFE2
208	D/0	16	FFE3
209	D/1	10	FE40
210	D/2	12	FF60
211	D/3	16	FFE4
212	D/4	16	FFE5
213	D/5	16	FFE6
214	D/6	16	FFE7
215	D/7	16	FFE8
216	D/8	16	FFE9
217	D/9	16	FFEA
218	D/A	16	FFEB
224	E/0	16	FFEC
225	E/1	11	FEE0
226	E/2	12	FF70
227	E/3	16	FFED
228	E/4	16	FFEE
229	E/5	16	FFEF
230	E/6	16	FFF0
231	E/7	16	FFF1
232	E/8	16	FFF2
233	E/9	16	FFF3
234	E/A	16	FFF4
240	F/0	9	FC80
241	F/1	16	FFF5
242	F/2	16	FFF6
243	F/3	16	FFF7
244	F/4	16	FFF8
245	F/5	16	FFF9
246	F/6	16	FFFA
247	F/7	16	FFFB
248	F/8	16	FFFC
249	F/9	16	FFFD
250	F/A	16	FFFE

History

Document history	
September 1992	First Edition
April 1996	Converted into Adobe Acrobat Portable Document Format (PDF)