ISO/IEC TR 15938-8:2002
TECHNICAL CORRIGENDUM 1

Published 2005-10-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Multimedia content description interface —

Part 8: Extraction and use of MPEG-7 descriptions

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Interface de description du contenu multimédia —

Partie 8: Extraction et utilisation des descriptions MPEG-7

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC TR 15938-8:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information.*

ICS 35.040 **Ref. No. ISO/IEC TR 15938-8:2002/Cor.1:2005(E)**

© ISO/IEC 2005 – All rights reserved

Published in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Page 266

Replace subclause 4.2.5.2.3 with the following:

4.2.5.2.3 Use of local representative colors (Visualization for Image Browsing)

The Color Layout descriptor is also useful to visualize image contents using a series of tiny picture icons for browsing since the descriptor values could be converted into small icons through the following two steps. The first step is coefficient decoding and the second one is inverse DCT transformation. Both processes are the inverse operation as shown in the semantics of **YDCCoeff**, **YACCoeff**, **CbDCCoeff**, **CbACCoeff**, **CrDCCoeff,** and **CrACCoeff** in subclause 6.6.4 in ISO/IEC 15938-3.

(1) Descriptor decoding

This process consists of three sub processes. The first one is the interpolation of truncated coefficients.

```
for(k=NumberOfYCoeff; k<64; k++) YCoeff[k]=16;
for(k=NumberOfCCoeff; k<64; k++) CbCoeff[k]=CrCoeff[k]=16;
```

The second one is the projection from zigzag-scanned 1D coefficient array into 2D-coefficient array.

```
YC[i][j]= YCoeff[zigzag(i, j)]
CbC[i][j]=CbCoeff[zigzag(i, j)]
CrC[i][j]=CrCoeff[zigzag(i, j)]
```

The third one is the inverse quantization as follows

```
yc[0][0]=iquant_Y_DC(YC[0][0]),      yc[i][j]=iquant_Y_AC(YC[i][j])
cbc[0][0]=iquant_CbCr_DC(YC[0][0]),  cbc[i][j]=iquant_CbCr_AC(CbC[i][j])
crc[0][0]=iquant_CbCr_DC(YC[0][0]),  crc[i][j]=iquant_CbCr_AC(CrC[i][j])
```

Here, the iquant functions should be implemented as in Table 23.

Table 23 - The inverse quantization table of DCT coefficients.

	Y	Cb, Cr

DC	`int iquant_Y_DC(int i) {`
	`  int j;`
	`  i=i<<1;`
	`  if(i>112) j=194+(i-112)*4;`
	`  else if(i>96) j=162+(i-96)*2;`
	`  else if(i>32) j=96+(i-32);`
	`  else if(i>16) j=66+(i-16)*2;`
	`  else j=i*4;`
	`  return j*8;`
	`}`
		`int iquant_CbCr_DC(int i) {`
		`  int j;`
		`  if(i>63) j=192;`
		`  else if(i>56) j=162+(i-56)*4;`
		`  else if(i>48) j=145+(i-48)*2;`
		`  else if(i>16) j=112+(i-16);`
		`  else if(i>8) j=97+ (i-8)*2;`
		`  else if(i>0) j=66+ i*4;`
		`  else j=64;`
		`  return j*8;`
		`}`

AC	`int iquant_Y_AC(int i) {`
	`  int j;`
	`  i=i<<3;`
	`  i-=128;`
	`  if(i>128) i= 128;`
	`  if(i<-128) i= -128;`
	`  if ((abs(i)) > 96 ) j= (abs(i))*4 - 256 ;`
	`  else if ((abs(i)) > 64) j=(abs(i))*2 - 64 ;`
	`  else j=abs(i);`
	`  j = (i<0)?-j:j;`
	`  return j*2;`
	`}`
		`int iquant_CbCr_AC(int i) {`
		`  int j;`
		`  i=i<<3;`
		`  i-=128;`
		`  if(i>128) i= 128;`
		`  if(i<-128) i= -128;`
		`  if ((abs(i)) > 96 ) j= (abs(i))*4 - 256 ;`
		`  else if ((abs(i)) > 64) j=(abs(i))*2 - 64 ;`
		`  else j=abs(i);`
		`  j = (i<0)?-j:j;`
		`  return j;`
		`}`

© ISO/IEC 2005 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

(2) Inverse DCT

The 8x8 icon (*yd*, *cbd*, *crd*) on Y/Cb/Cr color space can be obtained from decoded descriptor values:

*yd= ic*yc*ict, cbd=ic*cbc*ict, crd=ic*crc*ict*

where *ic* denotes the IDCT matrix and *ict* its transpose. For these purposes, the IDCT matrix can be specified as the inverse (equivalently, the transpose) of the forward DCT matrix, *m*, specified in ISO/IEC 15938-3 subclause 6.6.4. That is,

ict=m, *ic=transpose(m*)

The *transpose* of a matrix A is obtained by replacing each element a[i][j] with element a[j][i], where a[i][j] denotes the i-th row and j-th column element of the matrix A.

© ISO/IEC 2005 - All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale