Title:	**Version 2.4 of ICSP Codec**
Status:	Editor's Output
Purpose:	Text of ICSP Codec

Author:	Soonjong Jin
Tel:	+82-10-9417-0460
E-mail:	sjin@hanyang.ac.kr
Source:	Image Communication and Signal Processing Lab.

This document contains

- General description including theoretical terms.
- Decoder description including the player form of decoder.
- Encoder description including the window form of encoder.
- Final report form including configurations.

Revision History

REV.	REV. Date	REV. by	e-mail	Description
2.2	2010-12-29	김 태 호	crewx@hanyang.ac.kr	Adding intra 8x8 prediction, Modify syntax element
2.3	2011-12-28	김 용 훈	charismakyh@gmail.com	Adding MPM
2.4	2014-12-26	이 경 준	kjlee888@naver.com	Correcting some typographical errors. (Such as figure numbers etc.)

Table of Contents

Introduction

I am sure that this document can help your ICSP Codec design. Main purpose of this document is descriptions for encoder and decoder.

First of all, read suggestions in the next page, if you think that you need more study or understanding and then try after studying that.

Suggestions

2.1 Before reading this document

① Do not use this document as another purpose except education for ICSP Lab freshman because this document is only for project description which is basic procedure for ICSP Lab freshman.

② Before reading this document, you have to study about general image processing because this document does not deal with basic theory.

③ Some figures and tables may have a difference compared to truth or explanation.

④ Producer does not have any responsibilities for problem which is caused by this document.

⑤ Producer cannot endure any interference about this document without comment or advice.

⑥ It is positively necessary that you can implement after reading this whole document because all contents of this document are related to each other.

⑦ Do not modify this document without producer's permission.

⑧ This document may have a mistake. If you find any mistakes, please contact me (sjin@hanyang.ac.kr).

2.2 Before implementing ICSP codec

① First of all, you have to implement yourself completely. If you have a willingness to implement yourself, be sure to be master in your c language.

② For reference, if you need theoretical advice for c language, do not use any source based image processing books like "digital image processing using visual c++" or "c language for digital image processing". Only use dedicated language book such as c, c++, and visual c++.

③ Do not copy and paste any sources. For reference, do not use any web-sites or search engines.

④ Do not use any Open sources or functions such as OpenGL or OpenCV except MFC's pre-maded source for window programming.

⑤ You have to do window programming such as Visual C++ or MFC.

⑥ You might need considerable c language skill for implementation. So start after studying c, c++, and visual c++ languages.

⑦ Do your best and study hard because you have just two weeks until final report submission.

2.3 Background for ICSP codec

Before reading this document, you must know all terms below. If you have any terms which you don't know or understand, before next page, study that terms using search engine or books. (Recommend google [1], 그림으로 보는 최신 MPEG [2], 알기 쉬운 MPEG-2 [3], 영상 통신 시스템 [4], and H.264/AVC 비디오 압축 표준 [5].) We recommend you may study theoretical terms such as transform, entropy, basic enhancement, quantization, perception, filtering, and restoration. (Recommend Digital image processing by Gonzalez [6] and Fundamentals of digital image processing by Jain [7].)

① Basic terms of image processing
   A. Concepts of pixel, frame, field, slice, macro-block, frame rate, aspect ratio, bit-rate, and PSNR etc.
   B. Lossy and lossless compression
   C. Spatial, temporal, and statistical redundancies
   D. I, P, B, and D frames
   E. Meaning of Codec
   F. Interlace and progressive
   G. PAL, SECAM, and NTSC
   H. Color format such as RGB, YUV, and YCbCr

② Kinds and purpose of video standards
   A. JPEG
   B. H.261
   C. MPEG-1
   D. MPEG-2
   E. MPEG-4

③ Basic theories
   A. Image Perception
   B. Transform and quantization
   C. Enhancement and restoration
   D. Spatial and frequency domain filtering
   E. Entropy analysis

☞ It is obvious that you have to implement two separable programs, encoder and decoder. Do not contain decoder program in the encoder program. All results of final report are based on decoder side except encoding time.

General description for ICSP codec design

In this document, we just deal with 4:2:0 formatted sequence for encoder and decoder descriptions. 4:2:2 and 4:4:4 formats can be generated by 4:2:0 format sequence using interpolation.

3.1 Block coding order

Fig. 1 shows coding order of 4:2:0 format. As you can see, large character numbers mean macro-block order (raster scan order) and small character numbers mean real coding order after transform. In case of 4:2:0 format, sizes of Cb and Cr are half of luma.

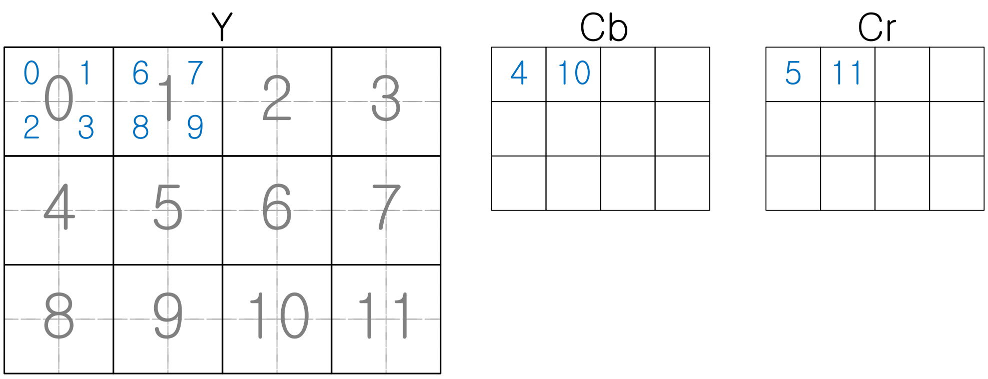


Fig. 1. Coding order for 4:2:0 format.

3.2 Parameter

In this section, we define some parameters such as size, quantization parameter, and some modes.

Size

- 176x144 (QCIF), 350x240(QVGA), 352x288 (CIF), 640x480 (VGA), 720x480 (SD), 1920x1088 (HD), 4kx2k (Digital Movie)
- 128 ~ 65535

Quantization Parameter

- For DC, 2 ~ 32
- For AC, 2 ~ 64

DPCM modes for DC prediction and pixel prediction

- Mode 0 : Median prediction mode
- Mode 1 : Mean prediction mode
- Mode 2 : Left prediction mode
- Mode 3 : Upper prediction mode
- Mode 4 : Upper right prediction mode
- Mode 5 : Left&Up prediction mode
- Mode 6 : no prediction

Motion Vector prediction modes

- Mode 0 : Median prediction mode
- Mode 1 : Mean prediction mode
- Mode 2 : Left prediction mode
- Mode 3 : Upper prediction mode
- Mode 4 : Upper right prediction mode
- Mode 5 : no prediction mode

Modes for motion estimation algorithm

- Mode 0 : Full Search Algorithm (FSA)
- Mode 1 : FSA + Partial Distortion Elimination (PDE)
- Mode 2 : Three Step Search (TSS)
- Mode 3 : TSS + PDE
- Mode 4 : Kim's Paper

Motion estimation block size mode for macro-block

- Mode 0 : 16x16 size motion estimation
- Mode 1 : 8x8 size motion estimation

Intra period for frame type

- Range : 0 ~ 31, 0 means all predicted frame except first frame.

Intra prediction flag

- Mode 0/1 : Disable / Enable

Encoder description

Fig. 2 shows the ICSP encoder structure. T and Q mean transform and quantization each. We use simplified intra 8x8 prediction, Discrete Cosine Transform (DCT) as T and uniform quantization with simple dead zone as Q. Zig-zag scan is used for reordering and entropy coder which is like JPEG's is used. For inter frame, Motion Estimation (ME) and Motion Compensation (MC) are used. Reconstructed frame has to be made using inverse transform and inverse quantization.

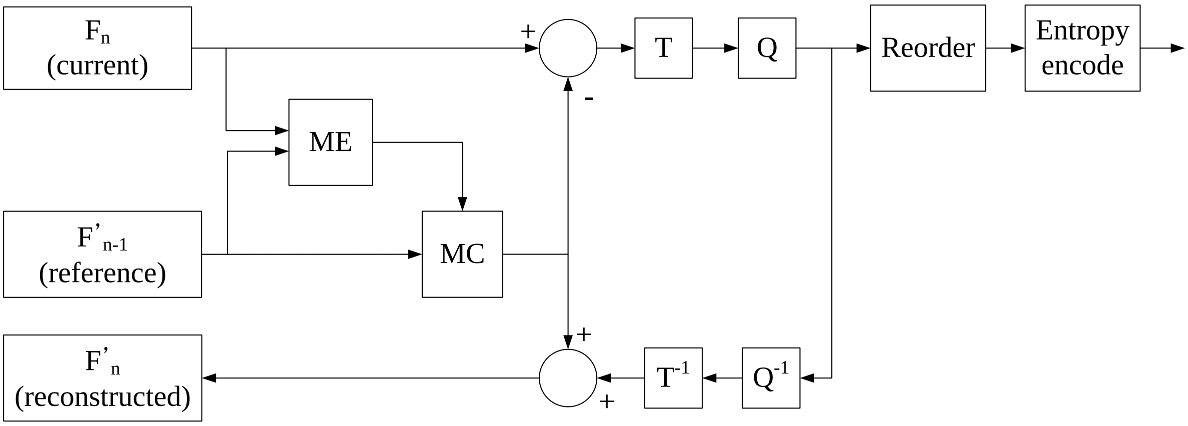


Fig. 2. Block diagram of ICSP encoder.

T : transform ➔ DCT

Q : quantization ➔ general uniform quantization with simple dead zone

Reorder ➔ zig-zag scan

Entropy encoder ➔ like JPEG

4.1 Encoding a intra frame

For encoding the intra frame, green units in Fig. 3 are used.

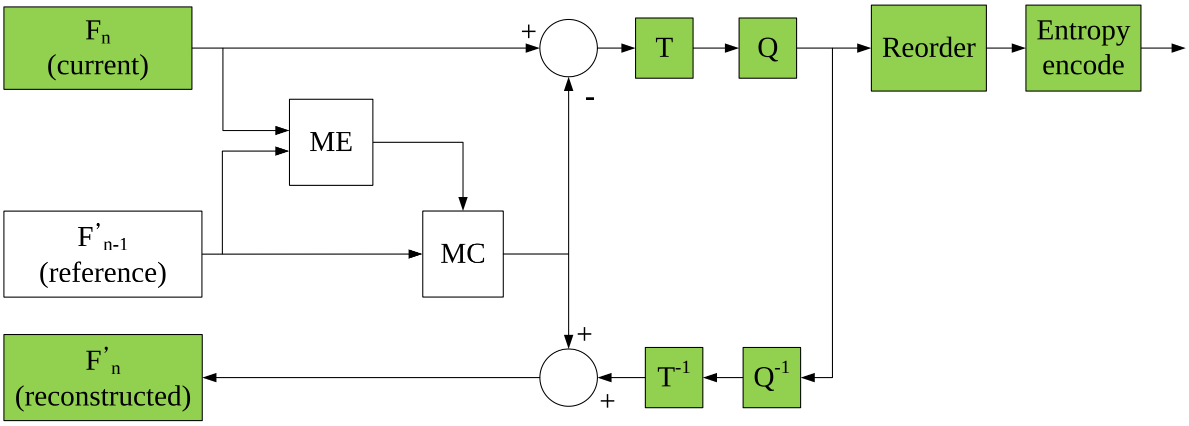


Fig. 3. Block diagram of intra frame.

In case of I frame, first of all, current frame is divided by macro-block. In raster macro-block scan order, after simplified intra 8x8 prediction, pixel based DPCM, discrete cosine transform, and quantization, we also use DPCM for reconstructed (i.e, quantized and inverse quantized) DC component. And then use the reordering (just zig-zag scan) and entropy encoder (like Huffman coding). While doing this forward process, we save the reconstructed data using inverse quantization, inverse transform, inverse DPCM and simplified intra 8x8 prediction for reference of next frame.

4.2 Encoding a inter frame

For encoding the inter frame, green units in Fig. 4 are used.

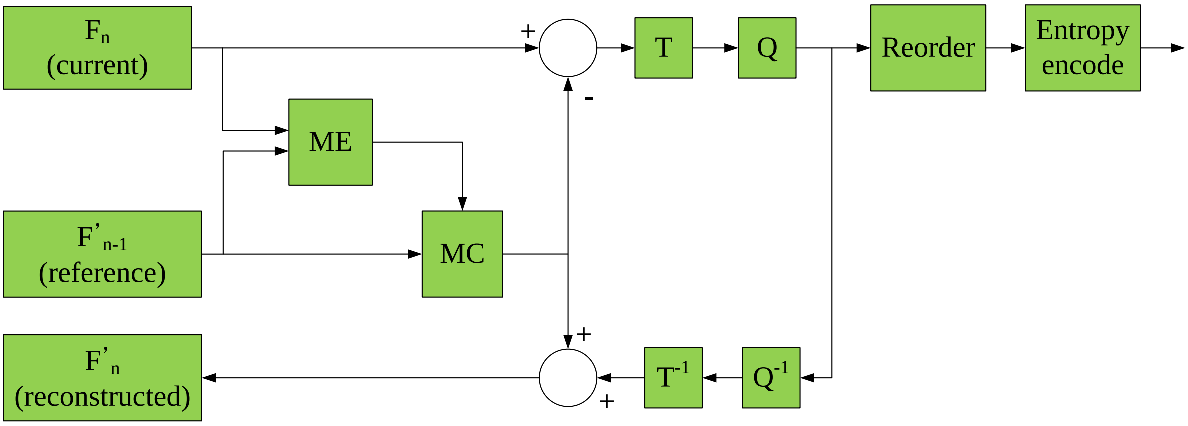


Fig. 4. Block diagram of inter frame.

In case of P frame, using previous reconstructed frame for reference, we use motion estimation to search the best motion vector of current macro-block. After motion estimation, we use DCT, quantization, DPCM for DC, reordering, and entropy coding for error image. We use motion compensation for reconstructed frame. In the inter prediction, we do not use pixel based DPCM.

4.3 Intra 8x8 prediction

We use simplified intra 8x8 prediction. Details of intra prediction is described as H.264/AVC 비디오 압축 표준 [5] and H.264/MPEG-4 Part 10 White Paper: Intra Prediction [10]. There are 8 intra prediction mode for ICSP codec as show in Fig.5. Each 8x8 prediction mode generates 64 predicted pixel values using some or all of the upper and left-hand neighboring pixels as show in Fig.5.

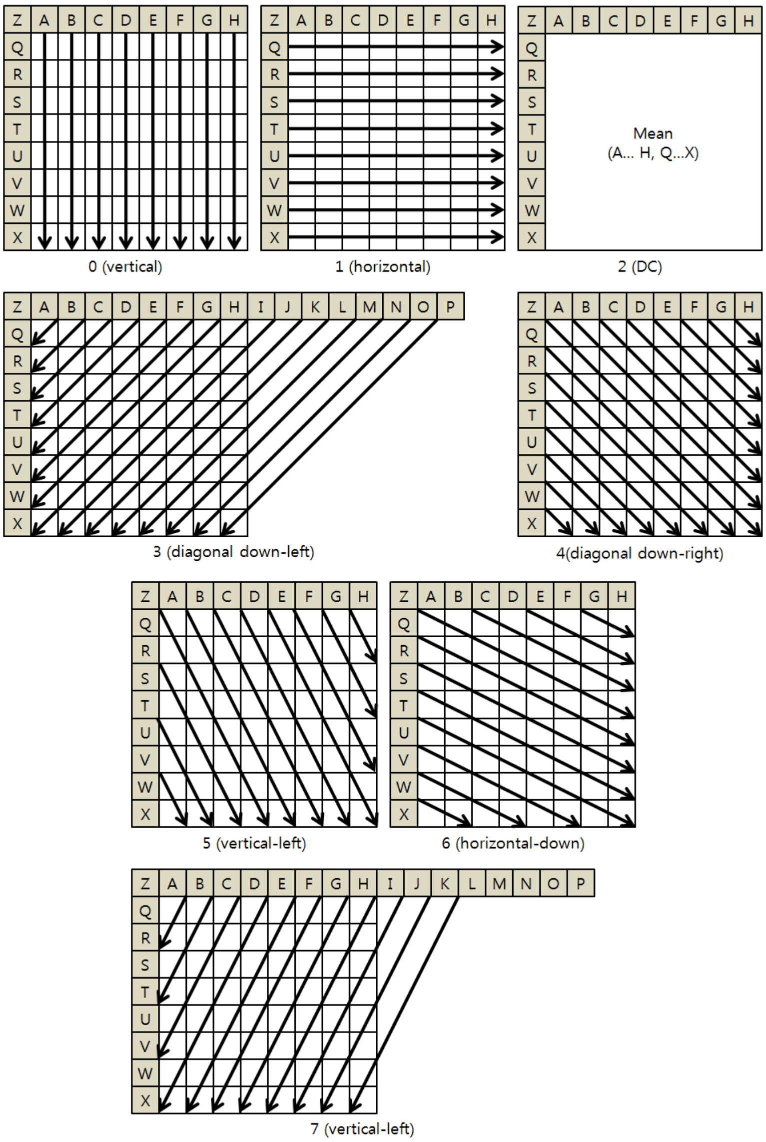


Fig. 5. Eight different modes for intra 8x8 prediction

The prediction block created by each of the predictions. The Sum of Absolute Errors (SAE) for each prediction indicates the magnitude of the prediction error. The best matching mode gives the smallest SAE between original block and prediction block. In ICSP codec, only luminance block is predicted. The best matching mode is coded as 3 bit.

4.4 Most probable mode

Upper left	Upper	Upper right
Left	Current	

Fig. 6. Positions of the 8x8 block or pixel.

To reduce the mode bit for intra mode, we use prediction using neighboring blocks. If median of Upper, Left, and Upper Left is equal to current mode, MPM_flag will be set to one. If not, MPM_flag will be set to zero and additional fixed mode bit is coded.

4.5 Transform

We use 8x8 block size DCT as transform. The following equations (1) and (2) specify the ideal forward DCT and backward DCT each. But these equations contain cosine terms. In our ICSP codec, we use "double" type instead of "float" type for cosine accuracy.

$$S(v,u)=\frac{1}{4}C_uC_v\sum_{x=0}^{7}\sum_{y=0}^{7}s(y,x)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{1}$$

$$s(y,x)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C_uC_vS(v,u)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{2}$$

where

$C_u, C_v = 1/\sqrt{2}$ for $u, v = 0$

$C_u, C_v = 1$ otherwise.

For one 8x8 block, we need 4 overlapping loops. It can be very heavy. We can implement equations (1) and (2) as separable two 3 overlapping loops each. You can reference Jain book [7], "Fundamental of digital image processing", page 132 to 180. In this book, you have to understand property of transform, energy compaction, and so on. And for each block, you think you may calculate the cosine terms but it is useless. Do not implement equation (1) and (2) themselves. Just use table method after making a cosine table once before transform process.

4.6 Quantization

We just use general quantization. There is only one thing. Be careful with round off. Forward quantization is defined as (3).

$$Sq(v,u) = round\left(\frac{S(v,u)}{Qstep}\right) \tag{3}$$

In (3), $Sq(v,u)$ is the quantized DCT coefficient, and $Qstep$ is the quantization step or quantization parameter. $S(v,u)$ is the DCT coefficient. Inverse quantization is defined as (4).

$$R(v,u) = Sq(v,u) \times Qstep \tag{4}$$

In (4), $R(v,u)$ is the inverse quantized DCT coefficients. We use DC $Qstep$ for DC coefficient and AC $Qstep$ for AC coefficients as $Qstep$.

Be sure that if 0.5 is rounded off, it becomes 1. But if -0.5 is rounded off, it becomes 0 not -1.

4.7 Reordering

We use zig-zag scan to increase a 0 probability sequencely. Zig-zag order of quantized DCT coefficients is defined as Fig. 6.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

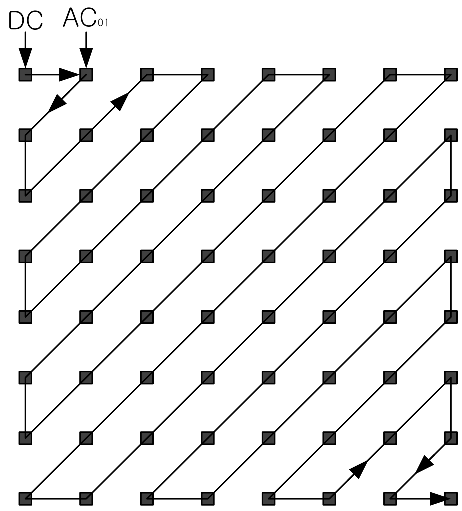


Fig. 7. Zig-zag order of quantized DCT coefficients.

4.8 Differential pulse code modulation (DPCM)

Since the current value is often large compared to 0, we predict the current value using neighboring value. We use DPCM for DC prediction and intra pixel prediction. In the two dimensional signal, we can use neighboring pixel values or DC values of neighboring 8x8 blocks. That's why we define the DPCM mode. Fig. 8 shows the neighboring 8x8 block positions or pixel positions which we could use.

Upper left	Upper	Upper right
Left	Current	

Fig. 8. Positions of the 8x8 block or pixel.

If upper right position is not available, it can be replaced by upper left position. If current DPCM mode is 0, i.e. median prediction mode, predicted value is median value of left, upper, and upper right positions' values (it can be DC values or pixel values). In prediction, if left and upper left positions are not available, then their values are replaced by upper position's value. If upper line is not available, then upper and upper right positions' values are replaced by left position's value. Finally, if all neighboring positions are not available, then we assume that predicted value is 1024 in case of DC value because 1024 is the DC value when all pixels in 8x8 block are 128, and in case of pixel, predicted value is 128. In the same way, we can get the predicted value in cases of other modes.

Mode 1 has same way compared with mode 0. Mode 2, 3, and 4 just use the value of each position. Mode 5 means that predicted value is average value of left and upper position's values.

After DPCM, error value, e, is defined as (5).

$$e = c - p \tag{5}$$

In (5), c and p mean values of current position and predicted value by DPCM mode each.

In case of intra prediction, we use pixel and DC based DPCMs. In case of inter prediction, we use DC based DPCM only because we use ME/MC.

4.9 Motion estimation and compensation

To reduce the temporal redundancy, the block matching algorithm (BMA) for ME has been widely adopted in many video compression standards, because it is simple to implement and effective at decreasing temporal redundancy. In BMA, the current frame is first divided into macro-blocks, which are fixed-sized square blocks, and the motion vector for each macro-block is estimated by finding the most similar block of pixels within its search range in a reference frame according to some matching criteria. The sum of absolute difference (SAD) is often used as a matching criterion between the current macro-block and the candidate blocks because of its lower complexity compared with other matching criteria such as the sum of squared difference (SSD) or the mean square error (MSE). The SAD between a current macro-block and a block of a candidate position within the search range of the reference frame at a relative coordinate (x, y) is defined as (6).

$$SAD(x, y) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left| f_{cur}(i, j) - f_{ref}(i + x, j + y)\right| \qquad (6)$$

where N is 16, $f_{cur}(i, j)$ and $f_{ref}(i, j)$ denote the pixel intensity of the current macro-block and the block of the candidate position, respectively.

In ICSP codec, we will implement the FSA [8] with PDE [9] for 16x16 and 8x8 modes each. The best-matched block can be obtained by the FSA, which entails a check of all candidate positions in the search range. PDE uses the partial sum of the matching error to eliminate the impossible candidates before finishing the calculation of the matching error. Decision rule, if current motion estimation mode is 16x16 or 8x8, is total size of bits for each mode. Namely, mode which has minimum bit size will be selected as best motion estimation mode.

4.10 Motion vector prediction

Motion vector prediction method of ICSP codec is almost same as DPCM. Horizontal and vertical directions of MV are separately predicted for each 8x8 size. And only macro-block has 1 MV for luma. MV for chroma is just half of luma's MV.

4.11 Entropy coding

With table I, we implement the entropy coding for motion vector, quantized DC after DPCM, and quantized ACs. It is very simple and looks like JPEG's table. Sign bit is 0 in case of negative value and 1 in case of positive value.

Table. I. Entropy code for ICSP codec.

Category	Code word	Range	Bit form Code word + Sign + Range	Total data length
0	00	0	00	2
1	010	1	010 x	4
2	011	2,3	011 x 0	5
3	100	4,…,7	100 x 00	6
4	101	8,…,15	101 x 000	7
5	110	16,…,31	110 x 0000	8
6	1110	32,…,63	1110 x 00000	10
7	11110	64,…,127	11110 x 000000	12
8	111110	128,…,255	111110 x 0000000	14
9	1111110	256,…,511	1111110 x 00000000	16

Assume that the value to be encoded is -3. The value -3 belongs to the category 2. So code word of -3 is '011'. Sign bit is '0' because the sign of -3 is negative. Finally, range of 3 becomes '1' because it is after 2. Consequently, -3 can be represented by '01101'. Another example is 18. The value 18 belongs to the category 5. And code word of 18 is '110'. Sign bit is '1'. Range of 18 is '0010'. So 18 can be represented by '11010010'.

For motion vector, if current motion estimation mode is 16x16, then we just encode only one MV for a macro-block in inter prediction. In other case, if current motion estimation mode is 8x8, then we encode four MVs for a macro-block. Fig. 13 shows the order of macro-block syntax. In Fi.g 13, if current motion estimation mode is 16x16, then MVmodeflag is '0'. If current motion estimation mode is 8x8, then MVmodefalg is '1'.

If ACs in current 8x8 block are all zero for intra and inter predictions, then AC flag is '1', and we skip encoding ACs in current 8x8 block. If ACs in current 8x8 block are not all zero, then AC flag is '0', and encoding all ACs.

4.12 Example of encoder

Fig. 9 shows an example for ICSP encoder.

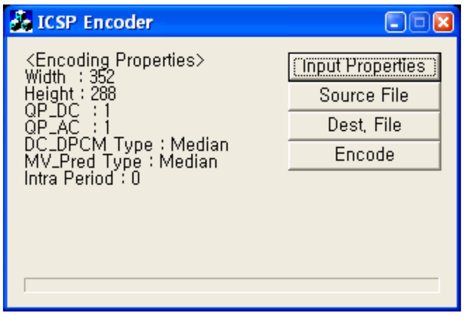


Fig. 9. Example of ICSP encoder.

In Fig. 9, there are 4 buttons. "Input properties" button is for encoding configuration. An example of this button is like Fig. 10.

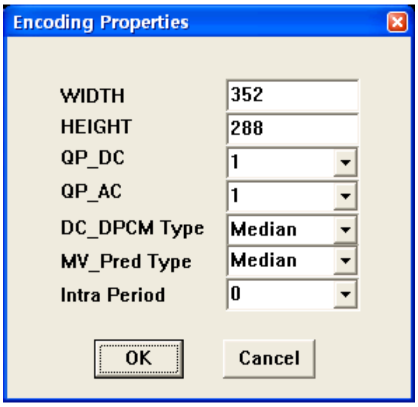


Fig. 10. Example of encoding configuration.

In Fig. 9, if you push "Source File" or "Dest. File" button, you can choose source or encoded stream file name. If you push "Encode" button, Fig. 9 change to Fig. 11. In Fig. 11, you can see that you cannot select any other buttons during the encoding process and progress bar tell you current state.

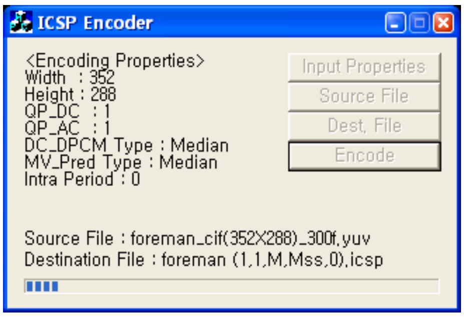


Fig. 11. Encoding state bar.

Decoder description

Generally, encoder contains decoder. Therefore decoder is very simple.

5.1 Decoder form

Fig. 12 shows the structure of ICSP decoder.

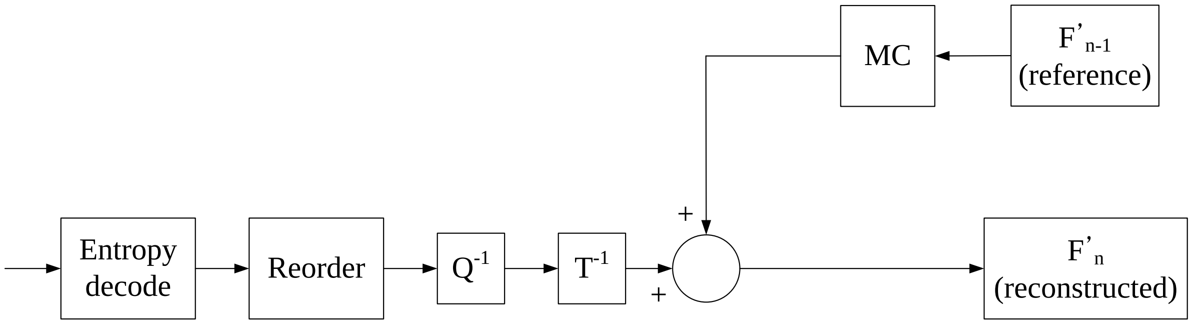


Fig. 12. Block diagram of ICSP decoder.

As you know, Fig. 2 contains Fig. 12.

5.2 Syntax

Stream header is defined as Fig. 13.

8 (256)	8 (256)	8 (256)	8 (256)	8 (256)
00000000	I	C	S	P

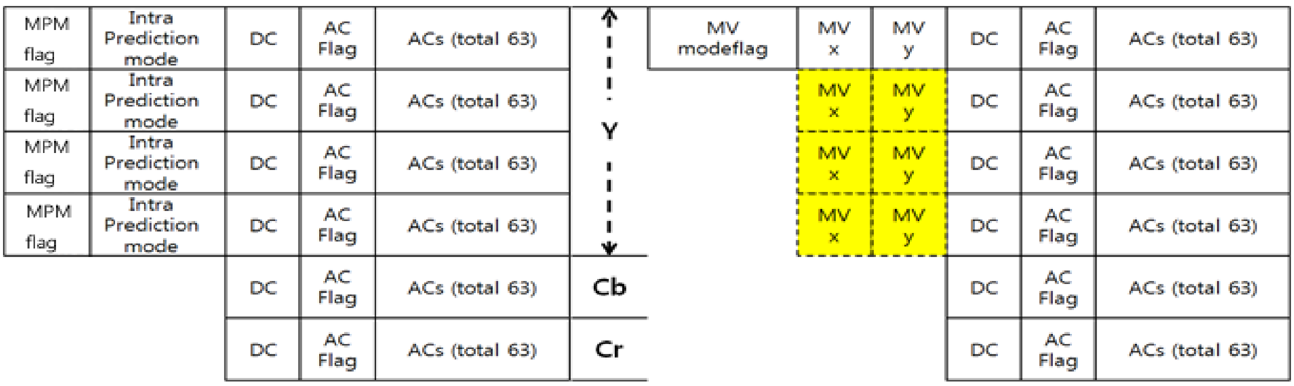
16 (65536)	16 (65536)	8 (256)	8 (256)	4 (16)
Height	Width	QP_DC, 2~32	QP_AC, 2~64	Pixel DPCM mode, 0~6

4 (16)	3 (8)	6 (64)	1(1)	6 (64)
DC DPCM mode, 0~6	MV_Pred Type, 0~5	Intra Period, 0~31	Intra prediction flag, 0 or 1	000000

Fig. 13. Stream header for ICSP codec.

It is all 14 bytes. First 8 bits and last 6 bits are all 0. After first 8 bits 0, ICSP characters mean ASCII code values.

Macro-block header is defined as Fig. 13. In Fig. 14., (a) and (b) mean intra and inter macro-block header each.



(a) Intra prediction (b) Inter prediction

Fig. 14. Macro-block header for intra and inter prediction.

5.3 Example of decoder

Fig. 15 shows an example of ICSP decoder.

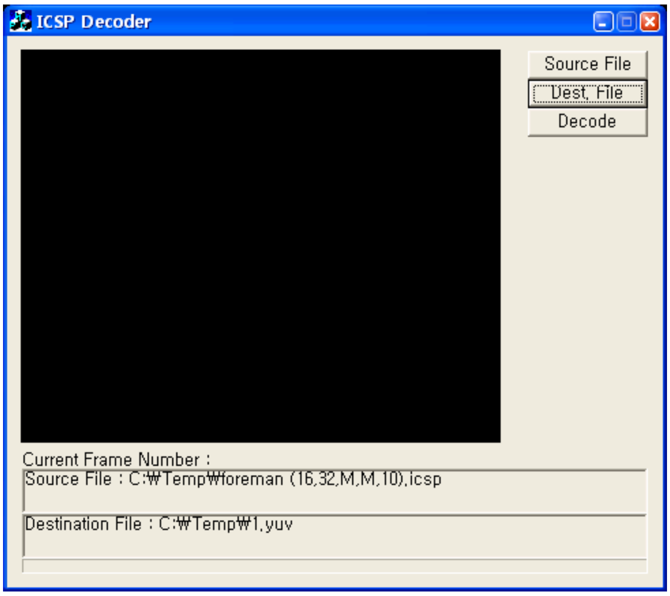


Fig. 15. Example of ICSP decoder.

In Fig. 15, there are 3 buttons and display window. “Source File” and “Dest. File” buttons are for names of bit-stream file and reconstructed file. If push the “Decode” button, decoding result is like Fig. 16.

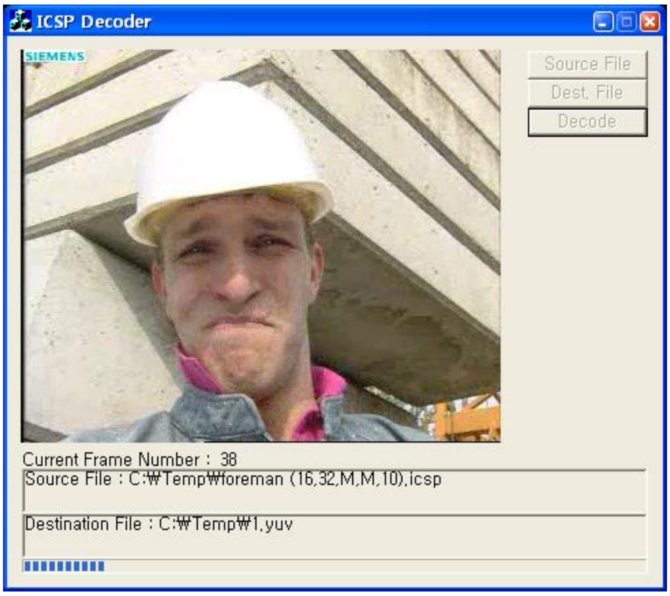


Fig. 16. Decoding state.

Display window has to show current decoded frame as moving picture and decoding progress bar tell you current state.

Final report form

You have to submit final report just like this document. It means that final report has to contain general document forms such as title, table of contents, lists of tables and figures, introduction, main encoder and decoder sources, some theories, experimental results, analysis, discussion, references, and two programming sources. Experimental results and analysis of results are very important. For following all configurations (it is recommend that you may do other configurations.), experimental results have to contain PSNR results, PSNR analysis, compressed file sizes, and size analysis.

Mandatory configurations for PSNR and size

Test sequence	: akiyo (300f), football (90f), and foreman (300f) (only CIF format).
DC QP	: 1, 8, and 16.
AC QP	: 1 and 16.
DC DPCM mode	: 0 and 6.
~~Pixel DPCM mode~~	~~: 0, 2, and 6.~~
MV prediction mode	: 0 only.
Intra period	: 0 and 10.
Intra prediction enable	: 0 and 1

☞ All combinations of above configurations are 432 times. (3x3x2x2x3x1x2x2=432).

Acknowledgements

A very special thanks to Hyunwoo Lee for giving his encoder and decoder examples.

References

[1] Online available - http://www.google.co.kr/ig?sp=korea&hl=ko

[2] 정제창 역, "*그림으로 보는 최신 MPEG*," 교보문고, 1999.

[3] 이호석, 김준기, "*알기 쉬운 MPEG-2*," 홍릉과학출판사, 2001.

[4] 김재균, "*영상통신시스템*," 영지문화사, 2000.

[5] 정제창 역, "*H.264/AVC 비디오 압축 표준,*" 홍릉과학출판사, 2005.

[6] Rafael C. Gonzalez and Richard E. Woods, "*Digital Image Processing*," Prentice Hall, 2001.

[7] Anil K. Jain, "*Fundamentals of Digital Image Processing*," Prentice Hall, 1989.

[8] Borko Furht, Joshua Greenberg, and Raymond Westwater, "*Motion estimation algorithms for video compression*," Kluwer academic publishers, pp. 64 – 67,

[9] Jong-Nam Kim and Tae-Sun Choi, "A Fast Full-Search Motion-Estimation Algorithm Using Representative Pixels and Adaptive Matching Scan," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 7, pp. 1040-1048, Oct. 2000.

[10] I. E. G. Richardson: "H.264/MPEG-4 Part 10 White Paper: Intra Prediction", May 12, 2004