MC_EZBC Codec

Original version of the manual Jan. 2002 by Peisong Chen
Updated May 2009 by IVB

Please note that this code is an earlier version of MC-EZBC, from J. W. Woods and P. Chen, *Improved MC-EZBC with Quarter-pixel Motion Vectors* , ISO/IEC JTC1/SC29/WG11 MPEG, m8366, Fairfax, VA, May 2002.

For current versions of MC-EZBC, go to http://www.cipr.rpi.edu/research/mcezbc/

Contents:
=========

1. How to compile?

Please note: these instructions are for **Microsoft Visual C++ version 6.0**

1. Open the file ‘Makefile1.dsw’ in the TempSub directory
2. On the left-hand side menu, click on FileView
3. Double-click on EZBC files-Makefile
4. Change the HOME directory to whatever is your main MC_EZBC directory
5. Make sure that LINK, Inc32, Lib and DLL directories are correct for your system
6. Save the Makefile
7. Double-click on Makefile1 files-Makefile
8. Again, change the HOME directory to whatever is your main MC_EZBC directory
9. Make sure that LINK, Inc32, Lib and DLL directories are correct for your system
10. Save the Makefile
11. Click on Build-Rebuild All
12. This should generate
    a. Libraries 3d_ezbc0a.lib and 3d_ezbc_dec0a.lib in the lib directory of MC_EZBC
    b. Executables 3DSBCde.exe, 3DSBCen.exe, and pull.exe in the bin directory of MC_EZBC
13. Depending on your system, you may experience warnings, but compilation should proceed without errors. If you get errors, don’t worry – executables are already provided in the bin directory, so you can run the example mentioned in Section 4. Of course, you won’t be able to modify and recompile the source until you figure out how to resolve compilation errors – you are on your own there.

2. How to run these programs?

a) how to run pre-encoder “3DSBCen”?

The execution template for the pre-encoder “3DSBCen”is:

> *3DSBCen parameter_file*

Coding parameters can be modified by editing the *parameter_file*.
We have provided a couple of sample parameter files in the directory of MC_EZBC\Exp.
Each parameter of the parameter file is described below:

“-inname”

Its arguments should be a printf format string defining the name of the input sequence file. It has to contain exactly one numerical descriptor (%d):

Example: -inname frame%04d.dpx
Then the encoder looks for files: frame0000.dpx, frame0001.dpx, frame0002.dpx, …
Input sequence files have to be in frame format.

“**-bitname**”

Name of output bitstream.

“**-statname**”

Name of output stattistic file containing some statistic information generated during encoding.

“**-format**”

Image format of each frame. Now we accept two formats:
-format DPX
-format YUV

“**-start**”

Number of the first frame

“**-last**”

Number of the last frame

“**-size**”

Size of each frame. There are four terms:
pixel width of the luminance component,
pixel height of the luminance component,
pixel width of the chrominance component,
pixel height of the chrominance component.

Example:
DPX sequence: -size 1920 1080 1920 1080
SIF sequence: -size 352 240 176 120

“**-framerate**”

Number of frames per second.

“-intra”

There are two definitions.

“-intra NO” which means interframe coding

“-intra YES” which means intraframe coding.

“**-denoise**”

There are two definitions.

“-denoise NO” which is recommended for SIF sequences

“-denoise YES” which is recommended for digital cinema sequences

Note: When the codec is in the intraframe coding mode, we should use “–denoise NO”

“**-motion**”

Currently, this parameter is fixed to be “hvsbm”, which means hierarchical variable block size block matching.

“**-search**”

This parameter defines the maximum search width/height at the lowest resolution levels in our hierarchical motion estimation.

If we have set “-denoise NO”, the equivalent maximum search width/height in the full resolution images will be 4 times the given argument.

If we have set “-denoise YES”, the equivalent maximum search width/height in the full resolution images will be 8 times the given argument.

“**-accuracy**”

This parameter defines the accuracy of motion vectors. Possible choices are: full, half, quater, and eighth corresponding to 1, 1/2, 1/4, and 1/8 pixel accuracy respectively. The default is quarter.

“**-lambda**”

In hierarchical variable size block matching, every block is split until reaching the size of 4x4. Pruning is then done using D + lambda R. Experimentation with a larger or smaller lambda can be used depending on the bit rate. The default is 24.

“**-tPyrLev**”

Levels of temporal subband decomposition.

We can infer the GOP size from this argument: GOP_size= 0x1<<tPyrLev, even in intraframe coding mode.

We recommend using 3 for digital cinema sequences, and using 4 for SIF sequences.

b) how to pull out bits from pre-encoded bit stream according to bit budget?

The software used here is called “pull”. The command line has the following general form

```
> Pull Precoded_Stream    -r   kilo-bits_per_second    [-C] [-h]
```

“precoded_stream” is generated by pre-encoder.

“-r” parameter with its argument specifies average bit rate.

The command-line options, with their arguments, are:
"**-h**": Display help message
"**-C**": When this option is used, it means CBR will be realized in bit allocation over GOPs. Default is VBR.

This software will output rate-constrained bitstream. The output bitstream is named according to the following convention:

sprintf(output_stream_name, "%.2s_%d.bit", **Precoded_Stream**, **kilo-bits_per_second**);

c) How to decode?

The software used here is called "3DSBCde". The command line has the following general form

>3DSBCde bitstream name_of_decoded_frame name_of_original_frame name_of_statistic_file"

You can use the output bitstream of "pull" as the input bitstream here.
The reason why **original_frame_name** is use here is to calculate the PSNRs, which will be put in **statistic_file**.
Note: Don't exchange the order of ***name_of_decoded_frame*** **and** ***name_of_original_frame,*** which will overwrite the original data.
Two printf format strings define **name_of_decoded_frame** and **name_of_original_frame** respectively.
For example, if the names of the original video frames are like car0000.dpx, car0001.dpx, …, we should substitute car%04.dpx for ***name_of_original_frame*** in the command line.

3. Memory requirements

To run digital cinema case, 1Gbyte memory is required.

4. Error concealment

Motion-compensated error concealment from I. V. Bajić and J. W. Woods, "Error concealment for scalable motion-compensated subband/wavelet video coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 508-514, Apr. 2007.

Various options for error concealment (e.g., uni- or bi-directional, with replacement or prediction) can be set in mctfN.c and decoderN.c.

- In order to enable error concealment, you must set the EC flag to 1 in decoderN.c.
- Then set the EC flag in mctfN.c to 1 for unidirectional concealment, or 2 for bidirectional concealment.
- Then choose the prediction mode by setting the PREDICTION flag in mctfN.c to 1 for replacement, or 2 for prediction.

Pre-compiled versions of the decoder with combinations of these options are provided in the bin directory:
3DSBCde_ec1p.exe – unidirectional concealment with prediction
3DSBCde_ec1r.exe – unidirectional concealment with replacement
3DSBCde_ec2p.exe – unidirectional concealment with prediction
3DSBCde_ec2r.exe – unidirectional concealment with replacement
3DSBCde_nc.exe – no concealment
3DSBCde_r.exe – no concealment of the current GOP, but generate the temporal LL band of the previous GOP by replacement
3DSBCde_p.exe – no concealment of the current GOP, but generate the temporal LL band of the previous GOP by prediction

(You can also regenerate these files by setting the appropriate flags as explained above, recompiling, and then renaming the decoder, since the default decoder executable filename will be 3DSBCde.exe.)

In order to test error concealment, it is best to encode and decode each GOP separately. Sample batch file for encoding the Y-component of the SIF-resolution Mobile Calendar sequence is provided in the bin directory (enc_moY.bat), with parameter files for each GOP in the par sub-directory of the bin directory. **The parameter files assume that the sequence is in the directory i:\sequence\sif\mobile\ with each frame stored in a separate file in the raw YUV format.** If your sequence is in a different directory, you need to change the first line in each of the par files accordingly. If you don't have sequences in this format, you can find them at
http://www.cipr.rpi.edu/resource/sequences/

Also provided is a sample batch file (dec_moY.bat) that decodes all GOPs at 512 kbps, except GOP 2, which is decoded at 130 kbps with various forms of concealment. Note that decoding of a GOP by 3DSBCde_r.exe or 3DSBCde_r.exe will, in addition to decoding the frames of that GOP, also generate the estimates of the last frame of the previous GOP (stored in the file named PredFrame) and the first frame of the next GOP (stored in the file named PredFrame2). One or both of these can be used in the concealment process, depending on the type of concealment. Please look at the batch file to figure out how to use the various decoder files. **Make sure that you change the directory i:\sequence\sif\mobile to wherever you keep the sequence.**

yuv2ras.exe file is also provided to convert frames from YUV to Sun rasterfile format. To learn how to use it, type "yuv2ras –h" or look at the batch file dec_moY.bat.