MCTF-JP2 Manual

Frøy Tureson Erlid

June 11, 2012

This document describes how to use the MCTF-JP2 Matlab codec prototype.

1 Prerequisites

Before using the codec, the following steps must be taken:

1. The JasPer software must be installed. JasPer is the reference software implementation of JPEG 2000, and the source can be downloaded from `http://www.ece.uvic.ca/~frodo/jasper/`. Build the JasPer software as described in its bundled manual.
2. The folder containing the JasPer executable must be added to the system path. A description of how to add a folder to the system path on a Windows system can be found at `http://www.mathworks.se/support/solutions/en/data/1-15ZLK/index.html`.
3. Download the `comp.zip` archive from `http://www.ux.uis.no/~karlsk/proj99/index.html`, and extract the file `Arith07.m` to the `matlab_prototype/utils` folder. This is an implementation of an arithmetic coder. This Matlab code is freely available, but is not included here in order to avoid violating any copyrights, as redistribution is not explicitly permitted.
4. Download the file `rle.m` from `http://www.mathworks.com/matlabcentral/fileexchange/4955-rle-deencoding` and place it in the `matlab_prototype/utils` folder. This is an implementation of a run-length coder. This Matlab code is freely available, but is not included here in order to avoid violating any copyrights, as redistribution is not explicitly permitted.
5. Finally, run the `setup.m` Matlab script. This script resides in the `matlab_prototype` folder.

Step 3 and 4 can be omitted. Canonical Huffman entropy coding must then be used, by setting the `mvEntropyCoder` parameter given to the encoder to 0 (see below).

2 Encoder Usage

The encoder works by first declaring an `Encoder` object. The encoding is started when the `encode(obj)` method is called on the `Encoder` object. The `Encoder` constructor takes a number of parameters that describes the input, the motion estimation, MCTF and rate control processes. The output stream files are stored in the specified output folder. Optionally, a log file can be written. An example of how to use the encoder is given in Listing 1.

```
% Encoder script

% ******** Parameters ***********
fileName = 'C:\ducks.yuv'; % Name of YUV4:2:0 file to encode
firstFrame = 0;            % First frame to encode (0-indexed)
lastFrame = 249;           % Last frame to encode (0-indexed)
frameWidth = 1280;         % Frame width
frameHeight = 720;         % Frame height
frameRate = 25;            % Frame rate
lumaBlockSize = 40;        % Block size for Motion Estimation (Luma, halved for chroma
    components)
lumaSearchRange = 64;      % Search range for Motion Estimation (Luma, halved for chroma
    components)
searchType = 'diamond';    % Motion Estimation search type
```

```
filterType = '1/3';          % Filter type for MCTF (Haar or 1/3 filter)
numOfTemporalLevels = 3;     % Number of temporal levels for MCTF decomposition
targetRate = 10000;          % Target output bitrate [kbps] (0 = lossless JP2k coding)
mvEntropyCoder = 1;          % 0: Canonical Huffman 1: Arithmetic coding
outputFolder = 'output';     % Name of output folder
lumaBudgetPercent = 0.85;    % Amount of byte budget given to Luma components
CbBudgetPercent = 0.12;      % Amount of byte budget given to Cb components
lpPercent = 0.25;            % Amount of byte budget given to LP subbands
logfile = 'log.txt';         % Name of encoder log file (optional).
% ****************************

enc = Encoder(fileName, firstFrame, lastFrame, frameWidth, frameHeight,...
                frameRate, lumaBlockSize, lumaSearchRange, searchType, filterType,...
                numOfTemporalLevels, targetRate, mvEntropyCoder, outputFolder,
                    lumaBudgetPercent, CbBudgetPercent, lpPercent, logfile );

encode(enc);

delete(enc);
```

Listing 1: Content of `../matlab_prototype/encoderScript.m`. This file provides an example of encoder usage.

3 Decoder Usage

The decoder works by first declaring a `Decoder` object. The decoding is started when the `decode(obj)` method is called on the `Decoder` object. The `Decoder` constructor takes the encoder stream folder as input, along with parameters describing the output folder and file name for the reconstructed yuv-file, and the name of a log file. Optionally, the path to the original yuv-file can be given as a parameter, in order to calculate PSNR and SSIM values. These values will then be written to the log file. These values are calculated with the `MeTriX MuX` software. This software is included in the codec, as permission for this is granted in the copyright notice. Note that the decoder log file that is specified can be the same file as the encoder log file, as the decoder logging will not overwrite the existing content of the file.

In addition to creating a reconstructed yuv-file, the decoder also outputs reconstructed subband frames as JPEG 2000 code streams. These files can be viewed with for example the program XnView (`http://www.xnview.com/`).

An example of how to use the decoder is given in listing 2.

```
% Decoder script

% Decoder parameters
inputFolder = 'str';             % Folder containing encoded stream files
outputFolder = 'rec';            % Folder to put reconstructed files
outputFileName = 'rec.yuv';      % Output yuv file
logFile = 'log.txt';             % Decoder log file
compareToFile = 'C:\ducks.yuv'; % Original yuv for PSNR/SSIM calculation (optional)

d = Decoder(inputFolder, outputFolder, outputFileName, logFile, compareToFile);

decode(d);

delete(d);
```

Listing 2: Content of `../matlab_prototype/decoderScript.m`. This file provides an example of decoder usage.

4 MeTriX MuX Licence Notes

In accordance with the permission notice of the `MeTriX MuX` software, the copyright and permission note is given below.

```
%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Copyright (c) 2007 Visual Communications Lab, Cornell University
%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Permission to use, copy, modify, distribute and sell this software
%%%  and its documentation for any purpose is hereby granted without fee,
%%%  provided that the above copyright notice appear in all copies and
```

```
%%%  that both that copyright notice and this permission notice appear
%%%  in supporting documentation.  VCL makes no representations about
%%%  the suitability of this software for any purpose.
%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Listing 3: MeTriX MuX licence notes