CODING & COMPRESSION OF MEDIA DATA

Project: JPEG Encoder for .avi Video

PROJECT DESCRIPTION

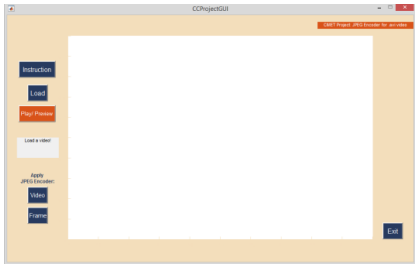
The project requires the implementation of JPEG encoder for .avi video, and application of JPEG compression on extracted frames of the video. The encoder should work on the principles of JPEG and each step of the encoder process should be coded as separate function. The tasks involve creation of GUI, loading of .avi video on the GUI, playing (and/ or preview) the video, extraction of a frame from the video while showing the step by step process of JPEG encoder, and application of JPEG encoder on the full video. The step by step process of JPEG encoder should allow the user to input the frame number to extract, display the extracted frame, apply JPEG encoder on the extracted frame, show all the intermediate steps of the encoder process on the GUI such as colour conversion, DCT, quantization, and save the compressed frame on the disk. At the end of the compression process, the evaluation rate (bit/pixel) and distortion (RMS error) before and after the compression will be evaluated.

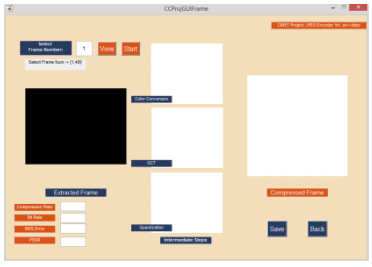
PROCESS

Creation of GUI

The Graphical User Interface (GUI) was created to allow the user to experience and view the result of the JPEG encoding process. It allows the user to operate each step of the process with point-and-click commands and ease in compressing .avi video without the need to study the Matlab language or enter commands to run the encoder.

For this project, three GUIs were created and contain interface command such as push buttons, axes, and textboxes.

a. 

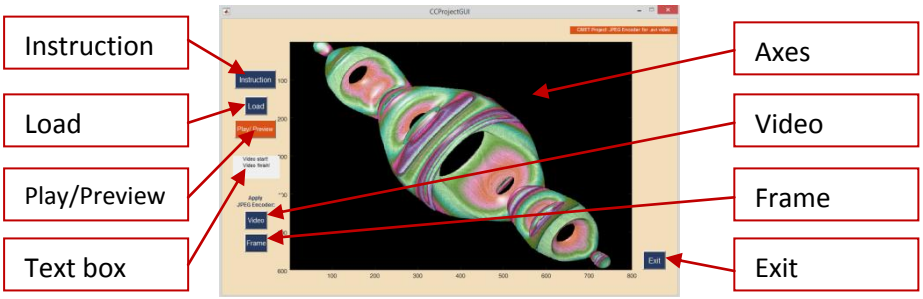
b. 

c. 

The three GUIs are: a) main GUI, b) GUI for compression of extracted frame, and c) GUI for instruction.

Main GUI

The main GUI contains six main push buttons and one axes. Compression of the whole .avi video will be implemented in this GUI. Through the use of push buttons, it will also direct the user to the GUI for instruction, and GUI for compression of extracted frame.



Push button

The push button callback function in GUIDE allows the user to execute specific command upon clicking the specific push button. Each push button in the GUI executes different command such as:

- Instruction – opens another GUI that contains instruction on how to operate the GUI and description of push buttons.
- Load – loads the chosen .avi video by the user for JPEG compression.
- Play/ Preview – plays the chosen video and/or provides a preview of the video before JPEG compression.
- Video – applies JPEG encoder on the full video and plays the compressed video on the axes.
- Frame – opens another GUI for compression of extracted frame chosen by the user.
- Exit – closes the main GUI.

Text box

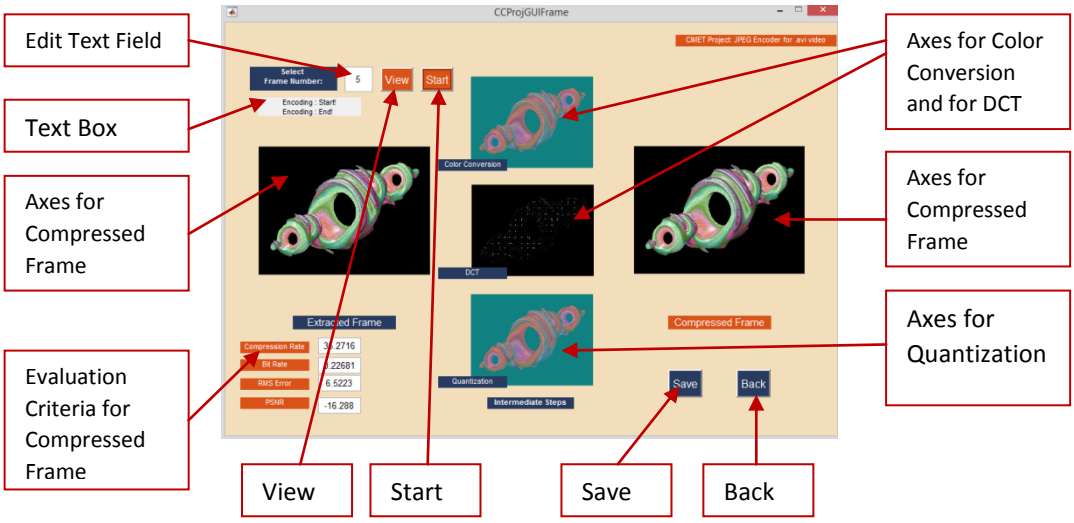
The text box displays the steps that the user must follow to execute the JPEG encoder. It serves as a guide for the user to successfully utilize the encoder.

Axes

The axes enable the GUI to display the chosen video by the user. Axes are not uicontrol but it can be programmed to implement callbacks through a click of each push button for a specific command.

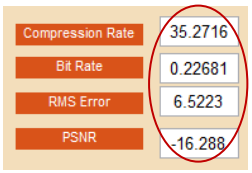
GUI for Extracted Frame

The GUI for compression of extracted frame contains two types of edit text field, five axes, four push buttons, and text boxes. This GUI is solely for compression of extracted frame of a chosen video. Through this GUI, the user can choose what frame number to be compressed using the JPEG encoder. It will allow the user to extract a frame from the video chosen using the main GUI. After extraction of frame, the user can apply the JPEG encoder on the extracted frame and observe the result of each intermediate step of the encoder process such as colour conversion, DCT, and quantization. The user can also save the compressed frame on the disk or return to the main GUI to compress the full video. Aside from the visual display of the compression, the user will have access to the information of the evaluation of compression rate (bit/pixel), PSNR, and distortion (RMS error) before and after compression.



Edit Text Field

a. 

b. 

Edit text field is usually associated with another callback function. For this project, there will be two types of edit text field. One edit text field (a) is where the user can input the number of the preferred frame to be extracted for compression. After the user typed the number, it will not execute any command upon clicking the Enter key. Instead, the input will be retrieved by another callback function, either View or Start. The other type of edit text field (b) displays the result of a specific computation.

Push button

The GUI for frame extraction and compression has four push buttons:

- View – displays the extracted frame chosen by the user and provided through the edit text field.
- Start – starts the JPEG encoder process on the extracted frame
- Save – saves the compressed extracted frame on the disk
- Back – closes the GUI for frame extraction and compression.

Text box

Aside from displaying the steps for the user to follow (gray text box), text box (blue and orange) also describes the axes and edit text fields for the user to further understand the process.

Axes

The five axes of this GUI provide visual display of a specific function or JPEG encoder process such as color conversion, DCT, quantization, and the final result of the process that is the compressed frame.

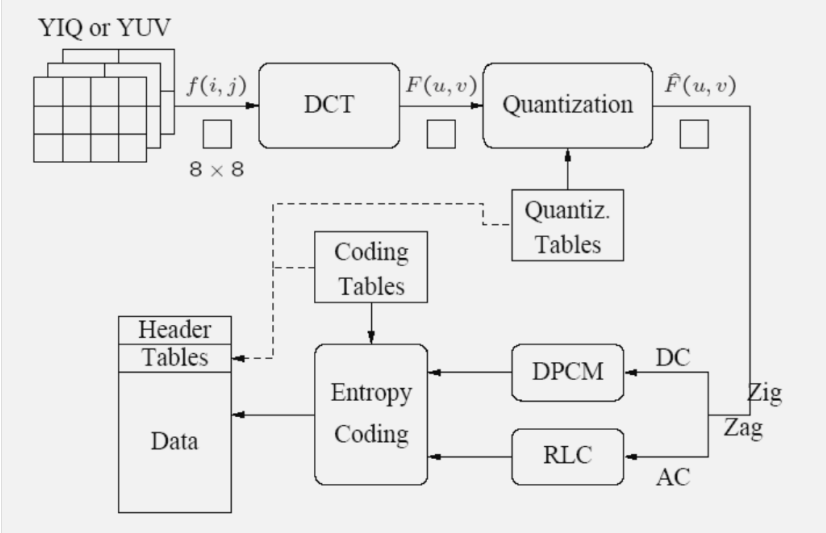
JPEG Encoder

Reference: Coding and Compression of Media Data Lecture Notes

Data compression or source coding is the process of encoding information utilizing smaller bits than the original un-encoded representation using a specific encoding algorithm. Compression is necessary to reduce the size of data files and provide safer and efficient storage and transmission of media data. Required storage for images is determined by the image dimension and byte per pixel while that of the video is determined by the image dimension, number of full frames per second and byte per pixel.

JPEG is an image compression standard developed by the Joint Photographic Experts Group. It compresses and/or decompresses data and provides a result that is not the same with the original data. It can result to small distortion but the small changes in the values of the pixel can be invisible. This lossy image compression method is suited for audio and video.

The project follows the step-by-step process of JPEG encoder.



The following steps are:

1. Color conversion and chroma subsampling – convert color images to a luminance-chrominance color space, YCbCr.

An image uses RGB color space and to allow different processing on the luminance and chroma channels, the RGB image will be converted to YCbCr color space. YCbCr has three components namely, Y for luminance, Cb for blue chrominance, and Cr for red chrominance. Due to different sensitivity of the rods and cones of the eye, some information is not as important as the others. Some information represented by Cb and Cr can be discarded or sub-sampled through compression.

2. DCT on image blocks – this process divides the image into 8x8 blocks and applies 2D DCT to each image block with DCT coefficients as output for each block. However, using blocks and isolating each block from its neighboring context can result to choppy or blocky JPEG images when a high compression ratio is utilized.

3. Quantization – introduces loss in JPEG compression and it also maximizes the compression ratio while minimizing the perceptual losses in JPEG compressed images using the Luminance and Chrominance Quantization tables.

4. Zig-zag ordering and Run-length Encoding – turns the 8x8 matrix into a 64-vector. The Run-length coding will be used for AC coefficients and Differential Pulse Code Modulation for DC Coefficients.

5. Entropy coding – in this step, the DC and AC coefficients will finally undergo an entropy coding step for further compression. Huffman coding will be used for the project.

Evaluation of Compression Rate and Distortion

To evaluate the efficiency of the JPEG encoder, the compression ratio, peak signal-to-noise ratio (PSNR), and root-mean-square error (RMS error) will be computed. Compression ratio will measure the reduction in size after running the JPEG encoder algorithm. It is equal to the number of bits of final file divided by number of bits of the original file. PSNR will measure the quality of the compression algorithm or the compressed image. A higher PSNR will usually indicate a higher quality of compression. RMS error will measure the efficiency of the compression algorithm based on the compressed and original images.

REFERENCES

_____. https://fr.mathworks.com/help/matlab/creating_guis/add-code-for-components-in-callbacks.html

_____. https://fr.mathworks.com/discovery/matlab-gui.html

_____. https://makarandtapaswi.wordpress.com/2009/07/20/why-the-rgb-to-ycbcr/

_____. http://www.mysticfractal.com/video/avi1.html

_____. http://www.robertstocker.co.uk/jpeg/jpeg_new_7.htm

_____. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

_____. https://en.wikipedia.org/wiki/Root-mean-square_deviation

For other References, see Matlab scripts.