360° Video Compression

User's Manual

Created by:

Saad Al-Khalifah

Alfonso Diaz

November 30, 2018

Revision Sheet

Release No.	Date	Revision Description
1.0	11/20/2018	Initial Revision
2.0	11/22/2018	Added general information section
3.0	11/27/2018	Added Getting started section (system overview and Running the code)
4.0	11/28/2018	Added result in 2.3
5.0	11/30/2019	Added External input video section

USER'S MANUAL

TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION

1.1 System Overview

The code takes video clip as an input and then preform sequence of operations to produce the compressed file of the original video. The package contains all required files to run the code and it contains two video clips used to test the code with. Each video will be sliced into frames in which they will passed to motion estimation and then to motion compensation. The result of both operations will produce predicated frame that will undergo transformations and finally encoded and converted to bit streams.

In addition, bit streams will be processed further to decode each frame and be written to video, so they can be compared with original video.

Several files will be saved in the same folder for further analysis and plot of operational rate-distortion curves and peak signal-to-noise ratio. Moreover, the compressed video will be available to view once the code completes its run. The code is designed to run perfectly with a black and white video, and with quality of 360p.

1.2 Purpose of This Manual

The purpose of this manual is to guide technical users and non-technical users in how to run the code correctly, and what kind of changes can be made to run the code with different desired parameters.

1.3 Points of Contact

1.3.1 Information

For additional information, we can be contacted at (salkhali@sfu.ca / adiazalo@sfu.ca).

2.0 GETTING STARTED

Extracting, Installing, and Running the video compression code

2.1 Extracting from the .ZIP archive

2.1.1 The code will be packed in .zip format which requires a proper software to unzip the package. In case user does not have the proper software, install it using this link www.7-zip.org.

2.2 MATLAB Environment

2.2.1 In addition to .zip archive, user needs to install the right environment to run the code which is MATLAB. In case user does not have MATLAB, install it using this link www.mathworks.com/downloads/.

2.3 Running the Code

2.3.1 **main.m File**

2.3.1.1 This MATLAB file is the main file and the only file can run the entire code with the desired parameters.

2.3.2 **Search Methods**

2.3.2.1 There are two main methods Exhaustive search and Dimond search in main.m file in which they can run using sum of absolute difference (SAD) or mean absolute difference (MAD). Choose the desired method with the proper const by uncomment the corresponding lines and comment the other methods. Keep in mind only one method can run at a time.

2.3.3 **Result**

2.3.3.1 Several files will be outputted in the current folder upon completion (Peak signal-to-noise ratio (PSNR), number of bits (Nbits), sum of the number bits (bitFileByte), sum of histograms bits (histFileBytes), header file, bit file and the reconstructed video.

2.3.3.2 Those result can be used to plot operational rate-distortion curves and peak signal-to-noise ratio in folder located in the same package “graphs”.

2.3.3.2.1 graph.m contains data to plot the two mentioned graphs and by running “res_final.m” all plot will be created.

3.0 EXTERNAL INPUT VIDEO

If user decides to run the code using another video, this section will guide user through the process. Several parameters can be changed such as quality, frame rate and number of frames to output different required results.

3.1 Video Requirement

The video must be black and white with video quality of 360p. Moreover, it must be .mp4 format to be processed.

3.2 Line to Be Modified

Video's name in line 39 and 43 must be changed to your supplied video name in order to run your clip.

3.3 Plot Result

After obtaining all result as the one in 2.3.3, then the following formulae must be used to obtain the expected graphs. For each quality, do the following

$$\text{Number of bytes} = \text{sum}(\text{bitFileByte}) + (8 * 120) + \text{sum}(\text{histFileBytes}) \quad (1)$$

$$\text{PSNR} = \frac{\text{PSNR}_{\text{for given quality}}}{\text{number of frames}} \quad (2)$$

As a result, operational rate-distortion curves and peak signal-to-noise ratio graphs can be plotted using the obtained results.