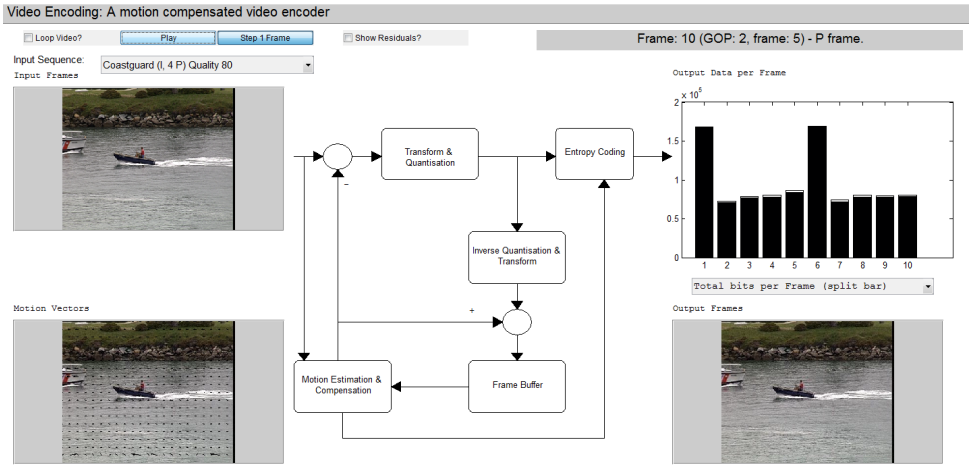
BRISTOL VISION INSTITUTE

VISTRA



Video Compression Demonstrator
User Manual

Contributors: Stephen Ierodiaconou, Aaron Zhang, Alex Mackin and Dave Bull

Introduction

The aim of this software is to act as a demonstration and teaching aid for courses in Image and Video Compression. The tool demonstrates principles such as spatial and temporal redundancy, colour channel subsampling, JPEG image compression, DCT transform coding, simple motion compensated video compression and motion estimation block matching.

Further details on the theory and techniques used throughout the software can be viewed in:

Communicating Pictures, Dave Bull, Academic Press, 2014

Getting Started

The software can be downloaded from www.bristol.ac.uk/vi-lab/demos

Open the software by using the command `start` in the root directory of the project. You can navigate between the demos by either clicking the home button, or by using the back and forward arrows in the screen toolbar. To show or hide more detailed elements of the demo screens use the Advanced Mode button.

Important Note

For some demo screens a setup is required the first time that they run. For example for the Video Encoding screen a cache of processed frames is created on first startup to speed up subsequent loads. Therefore for the purpose of demonstrations you should run the demos prior to using them live. The files created are stored in the root program folder and have a .mat extension. Once these have been created they can be taken with the program to other machines to be used there too.

Minimum Requirements

- 2GB RAM
- Core Duo 2.6GHz Processor
- MATLAB R2008a or newer
- The Image and Video Processing toolbox from Mathworks

The Demo Screens

The Toolbar

The main toolbar, located at the top of each screen, provides some common functionality. This includes:

- Navigation: The first 3 buttons allow navigation of the demo screens. The button will move the user to the intial demo screen. The and arrow buttons will move the user between the demo screens sequentially.
- Advanced Mode: If available the button enables an advanced mode for the screen. This results in the display of more on screen controls to allow further options to be configured for the given demo.
- Panning and Zooming: The zoom and pan buttons are toggle buttons, and clicking them will enable/disable their respective mode. These modes are to allow the user to zoom and pan any axes (images/figures) on the demo screen. For some screens the figures in a particular demo are linked, meaning pan or zooming one will perform the same operation on any linked figures.
- Refresh: You can refresh the current screen by using the button.

Screen 1: Redundancy - Spatial and Temporal Correlation

The top half of the screen shows the autocorrelation for the currently selected row of the input image. A row can be selected by clicking on the input image axes. Either an image can be chosen using the drop down menu or a random image can be generated using the **Random Image** button. The number of lags for the autocorrelation plot is set by default to 10 lags, however this can be changed via a slider in the top right of the screen which is enabled with the Advanced Mode of the screen.

The bottom half demonstrates temporal correlation. You can either cycle through the videos by clicking on the **Load Video** button, or a random sequence of frames can be generated using the **Random Video** button. The correlation plot shows the correlation between a selected pixel block in the first frame and subsequent frames in the sequence. A target pixel block can be selected by clicking on the image axes. Use the **Play** button to automatically play back the video frames, or use the **Step** button to single step each frame and watch the plot evolve.

Screen 2: Colour Channel Perceptual Redundancy (Subsampling)

This demo shows the perceptual redundancy associated with colour in images. The input image is selected from the input image combo box, and shows the given image in each axes with the subsampling mode show below each respectively. Changing the subsampling mode for a given axes updates the displayed

image automatically. To display each channel individually use the top middle combobox and choose between the intensity channel or the Cb/Cr channels. When selecting a colour channel the check box above this combo box becomes enabled and on selection displays the channel in the opposing colours it represents.

If a colour channel is on display the check box at the bottom of each image axis called **Show upsampled** button will become enabled. When unchecked the colour channel will be show in its subsampled size, if checked the channel will be shown upsampled to the original image size.

If Advanced mode is enabled a combo box in the top right is shown which gives the user the option of the type of interpolation filter to use when upsampling the colour channels. Advanced mode also enables a display at the bottom of each axes which show the selected block of 4×2 pixels (select a block on any image with a mouse click). The top left block shows the actual block pixels, the top right shows the subsampling structure for the given mode, the bottom left image shows the Cb channel for the current mode and the bottom right the Cr channel.

Screen 3: JPEG Image Compression

The screen demonstrates a top level view of JPEG image compression. An input image is selected and passes through a full MATLAB implementation of a JPEG encoder and decoder, where the JPEG Quality factor is set by the slider in top left. Upon completion the resulting images are displayed with statistics on compression and PSNR (image quality).

Screen 4: Transform Coding with the DCT

After selecting an input image, the input and output from the transform coding process are shown in the left and right image axes. In the centre of the frame a grid shows the DCT bases arranged so that top left is the DC coefficient, and the bottom right the highest frequency component in both the horizontal and vertical directions. Click on a basis to enable/disable that coefficient. NB: Disabling a DCT basis effectively zeros that coefficient in the inverse transform.

The JPEG quality slider sets the quantisation parameter for the coding process.

The tables of numbers in the bottom half of the screen show the pixels, coefficients, quantised coefficients, dequantised and inverse transformed coefficients and final output pixels of the currently selected 8x8 pixel block. To select a block simply click on the input image axes.

Screen 5: Motion Compensated Video Encoder

This demo displays the outputs from a number of stages of a generic video encoder. The input video sequence is selected from a set of fixed options in the top left. Playback is started or paused using the

Play toggle button. If playback is paused then the input can be stepped forward at a single frame at a time using the **Step 1** Frame button. The top left checkbox **Loop Video?** if enabled allows video playback to loop.

The checkbox in the top middle of the screen enables the display of the residual frames, both before and after coding.

To view a particular output in larger size simply click on the desired output. The first click will slightly expand the size of the output and the second click will expand the output to fullscreen. A third click restores the view to its original size.

The output graph has a drop down selection box underneath which allows the user to choose the output statistic to display. To see the output graph update after changing the video must be playing, or stepped forward one frame.

Screen 6: Block Matching for Motion Compensation

This demo displays the progress of given block match search techniques used in motion estimation and compensation. The input is selected using the top left selection box, and then the demo configured using the subsequent ones along the top, including selecting the frame to process in the sequence, the reference frame, the maximum search distance the search process (e.g. Diamond Search) and the macroblock size.

The actual block to process can be selected by clicking on the input image. Once a block has been selected the demo can be made to **Freerun** or **Step** through the search process which will animate in the figures on the left.

Appendix

Adding Example Images

Screens 1, 2, 3 and 4 all use input images from the directory **examples**/ located in the root program directory. Any JPEG, PNG or BMP images located in this folder will appear in the drop down selection boxes in these demos. To add simply add new images to this folder and restart the demo screens. Note: the images should not be too large as this will cause a very large memory requirement and slow operation. A good size is 256×256 or 512×512 pixels.

Available Packages

The following packages are included:

- GUIs: Contains the classes that implement the user interface screens. The base class contains any shared functionality and all screens should derive from this (and chain the parent constructor).
- Subsampling: Contains methods that implement functionality related to chroma subsampling and reconstruction.
- TransformCoding: Contains methods performing stages of the DCT process.
- EntropyCoding: Contains methods for the entropy encoding/decoding of JPEG (Huffman coding).
- JPEG: Contains the JPEG encoder and decoder bodies.
- Video: A motion compensated video coder, using JPEG for intraframe coding.
- MotionEstimation: The motion estimation functionality, such as difference calculation, block matching (full and diamond search).
- Utilities: Contains helper methods, such as methods to convert logical arrays of bits into numerical values and vice-versa.
- ThirdParty: 3rd party code, such as a faster implementation of the DCT.
- UnitTests: Unit tests for all packages.

Learning More

The source code is organised into packages of code according to functionality. The methods in each package are documented with description, usage and references to documents with more information. To view help on a specific method either open the file in a text editor or use the `help` MATLAB command, e.g. `help JPEG.encoder`.