MULTIMEDIA COMMUNICATION

Laboratory Session: Recommendation ITU-T H.261

Fernando Pereira

The objective of this lab session about Recommendation ITU-T H.261 is to get the students familiar with many aspects of video compression using the first standard video codec developed for videotelephony and videoconference over ISDN channels at $p \times 64$ *kbit/s*, with p = 1, ... 30. For that purpose, the application "Audio and Video Communication Simulation System" will be used as it includes among others a H.261 video codec module. Figure 1 shows some examples of typical videotelephony sequences; from these frame examples, it is clear that this type of content has rather specific features, notably face(s) as the main 'object', rather static background and simple motion.

Figure 1 - Typical videotelephony images.

This lab guide is organized in three parts corresponding to the three main modules in the video communication system available:

- H.261 video encoder (see Figure 2) Transmission channel (for bitstream error corruption)
- H.261 video decoder

1. H.261 ENCODER

The H.261 video codec is a so-called hybrid video codec since it acts in *time* through motion compensated predictions and *frequency* through the DCT, as shown in the simplified encoder architecture included in Figure 2.

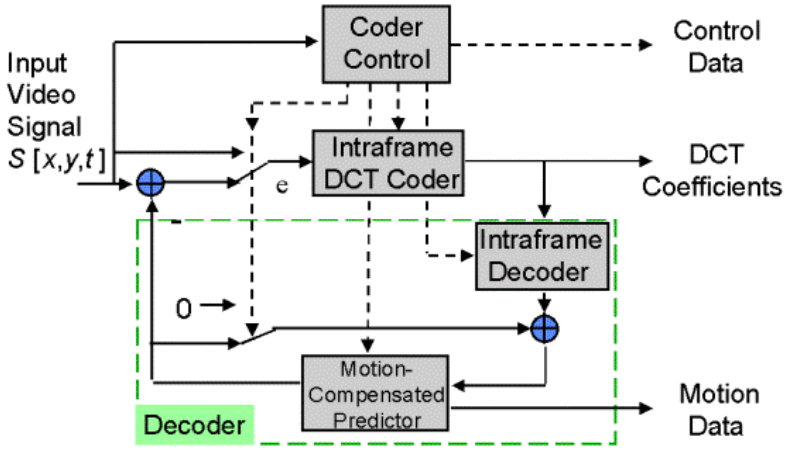

Figure 2 – Simplified H.261 encoder architecture.

1.1 *Select Sequence*

This button allows selecting the video sequence to code from those available. While most of the sequences are videotelephony like, some TV like sequences are also available.

1.2 Properties

This button allows controlling the encoder, notably determining the coding mode and parameters to use for the coding of the selected sequence. The sequence selected is always indicated in the encoder dialogue window.

- **Video Input –** This window allows specifying:
    - **Frame rate** (image/s) – The frame rate or temporal resolution by means of a reduction factor regarding the original frame rate indicated in the Recording frame rate box (which should not be changed as it refers to the frame rate of the original sequence stored in the disk); usually, 25/N or 30/N images/s are used which means that for N=1 the original frame rate is used and for N=2 half that frame rate is used. The frame rate is not completely free in this application since the user is obliged to a sub-multiple of the original frame rate, this means ½, 1/3, or 1/4 of the original frame rate (available in the disc). The frame rate reduction factor is controlled used the *Periodically dropped frames* box and the resulting frame rate is shown as Input frame rate.
    - **Frame resolution** – Two spatial resolutions may be used, both 4:2:0 (this means with chrominances having half the resolution in both directions): CIF (352×288 samples for the luminance and 176×144 for the chrominances) or QCIF (half the CIF resolution in each direction for the luminance and chrominances).
    - **Frames to encode** – Number of frames to code from the original sequence and the starting frame with counting starting in 0.
- **Outputs –** This window allows specifying the storage of decoded images and other information, such as motion vectors, during the encoding process.

- **Prediction –** This window allows specifying the type of predictions tools to be used, notably only Intra coding, only Inter coding (except the first image) or both Intra and Inter coding; it is also possible to indicate if motion compensation and the in-loop filter are used or not.
- **Motion detection –** Two motion estimation methods are available: full search and logarithmic or four steps search (for a $\pm x$ window with maximum x of 15). While the full search motion estimation allows finding the optimal motion vectors, this means those minimizing the selected error metric, the fast search estimation is computationally much lighter but does not guarantee that the optimal vectors are found for the selected error metric. For the full search estimation, it is also possible to define the size of the search window this means the maximum value of the motion vectors components (always ≤ 15 since this is the maximum value allowed by H.261).
- **Bitrate –** This window allows specifying constant or variable bitrate coding with variable bitrate implemented by using a constant quantization step. For constant bitrate coding, it is possible to select coding with any rate or coding with a rate multiple of 64 kbit/s, as specified by H.261. Also, it is possible to allow or forbid the usage of bit stuffing at macroblock level; bit stuffing allows assuring that the encoder has always some bits to transmit if the bitrate is too high for the video being coded (which is very unlikely!). The BCH (Bose-Chaudhuri-Hocquenghem) channel coding bit stuffing is not used in this application.
- **Buffer –** This window allows specifying the size of the encoder output buffer using a factor M applied regarding the default buffer size which corresponds to the average number of bits per image. M may take values from 0.1 to 10 and the default value is 1. The bigger the buffer size, the larger the delay between the two sides of the communication.
- **Quantization –** This window allows specifying:
    - **Constant bitrate coding** - The quantization step boundaries (following H.261, only even values, between 2 and 62), notably the minimum value (suggested value, 2), the maximum value (suggested value, 62) and the initial value (even value between the selected minimum and maximum);
    - **Variable bitrate coding** - The fixed quantization set to be used.
- **First frame –** This window allows specifying the coding of the first image using two modes:
    - **Normal mode** – The first frame is coded without any additional privileges in terms of bitrate which typically leads to bitrate overflow meaning that some macroblocks will not be coded since the output buffer is full; this is very likely as the available bitrate is typically not enough since the first image as to be coded using only Intra coding.
    - **Special unrestricted mode** - The first image is coed without bitrate in order the initial transitory situation is overcome. In this case, the output buffer fullness is set to half its size after coding the first frame, 'forgiving' the remaining bits to the first frame. When the special mode is selected, it is necessary to define the quantization step to be used of the first frame (suggested value, 16).
- **DCT –** This window allows specifying the DCT coefficients to transmit; if not all non-null DCT coefficients are transmitted, there will be bitrate savings but also a poorer video quality. The available possibilities in terms of DCT coefficients to use are:
    - No restrictions (send all coefficients);

- Transmit first X non null coefficients in zigzag scanning order;
- Discard coefficients with zig-zag order greater than X (corresponds to a low pass filter).

- **Channel encoding –** This window allows specifying the usage or not of the BCH channel codec, as defined by the H.261 recommendation. When present, the usage of the BCH bits is not mandatory at the decoder.

1.3 Code Sequence (button ►)

This button performs the encoding of the selected video sequence with the tools and parameters selected. The bits are stored in a file with a name which has to be indicated. The application always suggests a name which corresponds to the original name of the selected video sequence to which it is added the spatial resolution and the frame rate used; the suggested file extension is h261.

When encoding a sequence, the screen will show (see Figure 3):

- Original video sequence
- Decoded video sequence
- Matrix with the motion vectors for each macroblock
- Matrix with the coding mode for each macroblock using the following labels:

  I – Intra coding;

  D – Inter coding without motion compensation (only error differences);

  M – Inter coding with motion compensation;

  F – Inter coding with motion compensation and the in-loop filter;

  O - Overflow (indicating that the output buffer is too full and no bits may be sent).

- Several charts on the right side with the temporal evolution of various metrics, notably the PSNR (luminance and chrominances), the bitrate, the compression factor, the quantization step (minimum, maximum and average at frame level) and the output buffer fullness, see Figure 3. To move up and down in the charts zone use ▲ and ▼ on the top right corner.

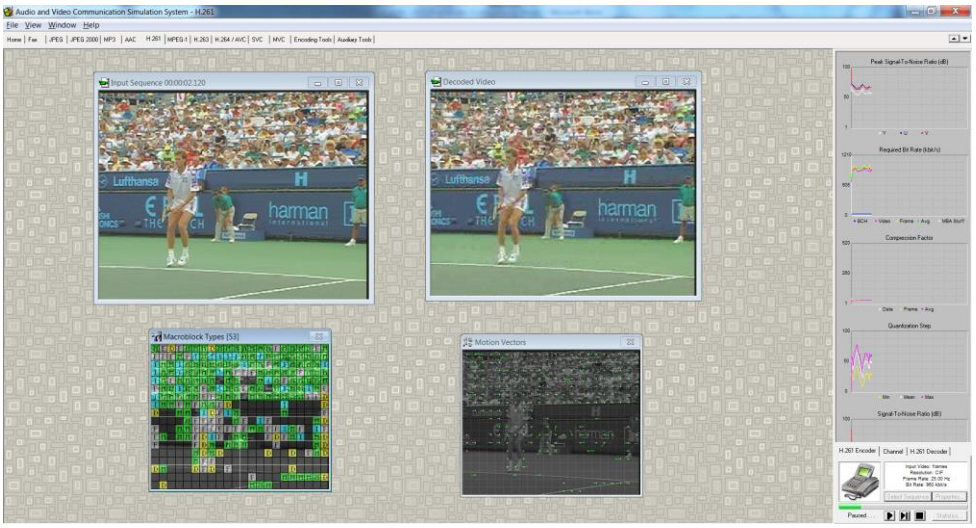

Figure 3 – General view of the application.

For better visualization, it is possible to use the button ►|| to encode frame by frame and the button ■ to stop decoding.

1.4 Statistics

The application makes available some encoding statistics in the Statistics window, notably the compression factor, the SNR and PSNR and the number of macroblocks of each coding type.

2. TRANSMISSION CHANNEL

This module simulates the transmission of a coded bitstream through a non-ideal channel by inserting bit error corruption with two different error patterns: uniform and burst.

2.1 Select Stream

This button allows selecting the file with the bitstream to corrupt. As mentioned above, the application suggested file name indicates the name of the coded sequence while its extension indicates the coding method and conditions used.

2.2 Properties

This button allows defining the type of bit errors to be introduced in the selected bitstream. The selected file is always mentioned in the dialogue window of the transmission channel module.

- **Type of Errors in the Corrupted Bitstream**:
    - **Uniform Errors** – The user must indicate in the corresponding check box if uniform errors must be inserted in the next bitstream 'transmission'.
    - **Error Bursts** - The user must indicate in the corresponding check box if burst errors must be inserted in the next bitstream 'transmission'
    - **Skip –** This parameter allows skipping some initial coded lines in the corruption process; this possibility allows visualizing in the same decoded and concealed image, parts which have been corrupted and not corrupted.
- **Uniform Errors** – For uniform bit errors, the user must define the desired bit error probability corresponding to the probability than a bit is corrupted independently of all the others.
- **Burst Errors** – For burst bit errors, the user must define the following parameters to characterize the desired errors:
    - **Burst Beginning Probability** – Probability that an error burst starts at a certain bit.
    - **Minimum Length of an Error Burst** – Minimum length of the bit sequence affected by a burst of errors.
    - **Maximum Length of an Error Burst** - Maximum length of the bit sequence affected by a burst of errors.

  The application computes and shows the bit error probability for the bursts understood as a fraction of bits corrupted.

2.3 Transmit Bits

This button implements the error corruption with the selected characteristics on the selected bitstream file. Before, the user has to define the name of the file where the corrupted bitstream is to be stored. The application always suggests giving the corrupted bitstream file the same name as the non-corrupted input bitstream file with the addition of a letter in the file extension signaling the type of errors introduced:

- 'b' for burst errors
- 'u' for uniform errors
- 'a' for both burst and uniform errors

2.4 Statistics

After the transmission with corruption, the Statistics button allows to obtain much statistical data about that process, notably:

- Number of uniform errors introduced in the bitstream.
- Bit error probability corresponding to the uniform errors (burst errors are excluded).
- Number of error bursts introduced in the bitstream.
- Average length of the error bursts introduced in the bitstream.
- Number of error bits introduced with the error bursts.
- Bit error probability corresponding to the error bursts (in relation to the total number of bits).

3. H.261 DECODER

This module simulates a H.261 video decoder. The application gets as input a file with a H.261 bitstream and provides as output the corresponding decoded video sequence. The input bitstream must include the BCH(511,493) channel coding bits and may also include bit stuffing. As option, it is possible to use some error concealment and post-processing tools to maximize the video subjective quality while minimizing the negative subjective impact of the channel errors.

The negative impact of the channel errors in the final decoded video sequence may be minimized in several ways, notably:

- **Error Correction** - The correction of errors is made by using a channel codec. For the H.261 recommendation, this channel codec is the BCH(511,493) codec; its usage at the decoder is optional. It is important to remind that the usage of a channel codec does not guarantee that all introduced channel errors are corrected, notably if the amount of errors inserted in the bitstream is above the error correcting capabilities of the selected channel codec (which may very likely happen when there are many errors, as for burst errors). This means that even when error correction is used by means of channel decoding still errors may result.
- **Error Concealment and Post-Processing -** When the source decoder detects errors, independently of the fact that channel decoding has already been used or not, it is possible (and wise) to use tools to minimize the subjective negative impact of these errors, e.g. by exploiting the information available for the previously decoded frame. The errors may be

syntactically – a certain sequence of bits does not exist as a codeword in the current decoding table - or semantically - a certain situation is impossible, e.g. the DCT coefficient with zig-zag order 75 (the maximum is 64 !) – detected.

For the macroblocks located between the error detection position and the next resynchronization position, e.g. a Picture Start Code (PSC) or a GOB Start Code (GSC), it is possible to use, in the worst case, the collocated information from the previous frame (better solutions exist !). However, it is common that the errors are typically detected after their real position in the bitstream, and thus after some coding artifacts have already been 'decoded', since some bits after the error were syntactically and semantically valid but not the same as transmitted by the encoder. To reduce the negative impact on the decoded video produced by the decoding of these syntactically and semantically valid but not correct bits, two types of tools may be used:

- **Error concealment for the areas without decoded data -** The error artifacts may be significantly reduced by discarding part of the decoded information, notably available for the macroblocks positioned immediately before the position where the error was detected and using for these macroblocks and the macroblocks until resynchronization, one of the following error concealment strategies:
  - Discard the full GOB (Group of Blocks) where the error occurred, and eventually the following GOBs until resynchronization, and make these GOBs the same as the corresponding GOBs in the previous frame.
  - Some macroblocks before the position where the error was detected are discarded and made the same as the corresponding GOBs in the previous frame; the MBs not available until resynchronization are made the same as the corresponding GOBs in the previous frame.
  - All decoded data is used and the MBs not available until resynchronization are made the same as the corresponding GOBs in the previous frame.
  - All decoded data is used and the MBs not decoded until resynchronization are taken from the previous frame after motion compensation with the collocated previous frame motion vectors (assuming that there is motion continuity in that image zone).
- **Post-processing 'wrongly' decoded image zones -** Independently of the previous usage or not of BCH error decoding and error concealment, it is possible that some errors are not detected or are not adequately concealed, originating unpleasant artifacts in the decoded video. The macroblocks in these zones may show a very different aspect from their neighbors, especially for the chrominances, strongly impacting the subjective quality. To overcome these situations, usually syntactically and semantically undetectable, it is possible to use some post-processing tools. In the available application, there is a post-processing tool which consists on a 'discontinuity filter' which measures the difference between the pixels at the borders of each macroblock and the neighboring pixels in the neighbor macroblocks since this is the best position to detect strange situations such as a uniform pink macroblock, completely different from its neighbors. If this discontinuity function is above a certain threshold, than the whole macroblock or at least parts of its blocks are

discarded (if the error is only concentrated in some blocks); after, some error concealment tools are used to find a good substitute for this image zone. The post-processing tool available may be controlled in two ways:

- **Dynamic Control -** Dynamic control implies that the threshold mentioned above is proportional to the maximum of the discontinuity function in the two previous macroblocks.
- **Static Control -** Static control implies that the threshold is fixed**.**

The post-processing tool is independently applied to the luminance and chrominances which means it is possible to post-process one component to discard its decoded information and not the others.

3.1 Select Stream

This button allows selecting the file with the H.261 bitstream to decode, corrupted or not. The name of the files with the bitstreams typically indicates the name of the original sequence which has been coded, the spatial resolution and the frame rate, while the file extension indicates the standard codec used and the type of error corruption inserted.

3.2 Properties

This button allows specifying the decoding conditions and tools to use. The selected bitstream to decode is always indicated in the decoder window.

- **Outputs** – Allows specifying how many frames must be decoded as well as to signal if the decoded frames must be stored in a disc file or not.
- **Channel Decoding** – Allows specifying the usage to give to the BCH data, notably if this data should be used or not for error correcting purposes, and the actions to take in the case there are errors that could not be corrected, e.g. completely discard the corresponding BCH blocks.
- **Error Concealment** – Allows specifying the error concealment tools to use, notably:
    - Replace the GOB where the error was detected with the corresponding GOB in the previous frame.
    - Replace the GOB where the error was detected with information from the previous frame obtained after motion compensation with the corresponding motion vectors from the previous frame (under the assumption that there is motion continuity).
    - Discard *X* decoded macroblocks before the error occurrence; the discarded and not decoded macroblocks are replaced with the corresponding macroblocks from the previous frame.
    - Discard *X* decoded macroblocks before the error occurrence; the discarded and not decoded macroblocks are replaced with information from the previous frame obtained after motion compensation with the corresponding motion vectors from the previous frame (under the assumption that there is motion continuity).

- **Smoothing Filter** – Allows using a low-pass filter to 'smooth' the decoded frames, notably when there are strong block artifacts. This filter is applied to all the pixels (with the exception of the image boundaries) and considers the eight neighbors of each pixel. The filter weights are 9 for the central pixel (the pixel under filtering), 3 for the 4-neighbor pixels (this means up, down, left and right) and 1 for the remaining four diagonal pixels.
- **Discontinuity Filter** – Allows using the post-processing described above, using either dynamic or static control (as explained above).
    - **Dynamic Control** – Allows specifying the proportionality factors to be used; suggested values are K_Y=2, K_U=1.5 and K_V=1.5.
    - **Static Control** – Allows specifying the control parameters; suggested values are 30, and 25 for the luminance and chrominances, respectively.

3.3 Decode Stream (button ►)

This button implements the decoding of the bitstream in the file selected using the error processing tools previously indicated for the creation of the decoded sequence. During the decoding process, the screen will show:

- Decoded video sequence without any error concealment and post-processing tools
- Decoded video sequence with the selected error concealment and post-processing tools
- Matrix with decoding synchronization indication, notably green and red when the decoder is in and out of sync, respectively

For better visualization, it is possible to use the button ►|| to decode frame by frame and the button ■ to stop decoding.

3.4 Statistics

The application makes available some statistics about the detected decoding errors.

Multimedia Communication Laboratories

Laboratory Session: Recommendation ITU-T H.261

Date _____ **Day of the week** _____

Nº _____ **Name** _____

Nº _____ **Name** _____

Try addressing the following questions in a speedy way but while always fully understanding what you are doing; if you don't understand, ask for help. The most important is to learn …

Whenever nothing specific is said, use the default modes and parameters provided by the application, notably those maximizing the encoding compression efficiency (e.g. do not switch off tools if not requested). Whenever you change question, reset the coding modes and parameters to the default situation. For each case, analyze the statistics available both in the charts and in the Statistics window, notably the average PSNR and bitrate.

1. **Quality versus Content –** Code the sequences VTPHCIF, Carphone and Stefan_cif with half the original frame rate and CIF spatial resolution at 64 kbit/s. Comment the video quality for the various sequences, notably explaining its evolution in time.

_____
_____
_____
_____
_____
_____
_____

2. **Quality versus Bitrate –** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 16, 32, 64 and 128 kbit/s. Comment the video quality for the various parts of the sequence and for the several bitrates, notably explaining the observed temporal evolution.

_____
_____
_____
_____
_____
_____
_____
_____

3. **Quality versus Coding Tools –** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 64 kbit/s using: *i) only Intra coding; ii) only Inter coding with motion compensation; iii) only Inter coding without motion compensation; iv) both Intra and Inter coding (with motion compensation).* Comment and compare the video quality for the four coding situations. For case iv), why does the number of Intra macroblocks increase if motion compensation is disabled ?

_____
_____
_____
_____
_____
_____
_____
_____
_____

4. **Macroblock Types Distribution –** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 64 kbit/s. Comment the distribution in space (*within each frame*) and time (*along the frames*) of the various macroblock coding types. *Try to understand why the encoder is doing a 'clever' job.*

_____
_____
_____
_____
_____
_____
_____
_____

5. **Motion Vectors Distribution -** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 128 kbit/s. Comment the distribution in space (*within each frame*) and time (*along the frames*) of the motion vectors as well their amplitude and direction. *Try to understand why the encoder is doing a 'clever' job.*

_____
_____
_____
_____
_____
_____
_____
_____
_____

6. **No-Rate Macroblocks Comparison -** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 128 kbit/s. What is the difference in terms of bitrate between the *grey* blocks and the *red hand* blocks ? And what is the difference in terms of their associated decoded quality?

7. **Quality versus Output Buffer Size -** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 64 kbit/s using an output buffer with an M scale factor equal to 1, 5 and 10. Comment the video quality obtained with the various values of M. When do you see the biggest differences ? Why ? Compare the PSNR and compression factor temporal evolution for the three cases.

8. **Motion Estimation Alternatives -** Code the sequence VTPHCIF with half the original frame rate and CIF spatial resolution at 64 kbit/s using both methods available for motion estimation, notably the *full and four steps searches*. Which difference did you observe when coding using the two alternative solutions (*pay attention to the running time*) ? Which advantages and disadvantages have these two alternative motion estimation solutions ?