ACM MHV 2024

Standards-Based Neural-Network Post-Filters for Improved Video Quality

Sachin Deshpande (Sharp)

SHARP

Agenda

- Introduction
- Neural-Network Post-Filter Characteristics (NNPFC) SEI
- Neural-Network Post-Filter Activation (NNPFA) SEI
- Neural-Network Post-Filter Extensions and Technologies Under Consideration
- Neural-Network Post-Filter Implementation
- Conclusion

Introduction

- Neural-Network based technologies are being increasingly investigated in video domain.
- Neural-network based video coding (NNVC) technologies are under exploration in Joint Video Experts Team (JVET).
- Versatile Supplemental Enhancement Information (VSEI) version 3 ((Rec. ITU-T H.274 | ISO/IEC 23002-7) has standardized Neural-Network Post-filter (NNPF).
- NNPF in VSEI v3 includes:
  - Neural-Network Post-Filter Characteristics (NNPFC) SEI
  - Neural-Network Post-Filter Activation (NNPFA) SEI
  - General process for applying NNPFs

Neural-Network Post-Filter Characteristics (NNPFC)

NNPFC SEI Provides information about:

- NNPF Purposes
- NNPF Modes and Identification
- NNPF Base and Update
- NNPF Pictures:
  - Input
  - Output
- NNPF Information:
  - Input pictures
  - Output pictures
- NNPF Patch, Overlap, Padding
- NNPF Interfaces: *DeriveInputTensors*, *StoreOutputTensors*
- NNPF Complexity Information

NNPFC: Purposes

- Different types of NNPFs are supported via an extensible syntax element : `nnpfc_purpose`
- NNPF may support one or more than one purpose
- Some combinations are forbidden
- Application defined NNPF is also supported

NNPF Purpose
General visual quality improvement
Chroma upsampling (from the 4:2:0 chroma format to the 4:2:2 or 4:4:4 chroma format, or from the 4:2:2 chroma format to the 4:4:4 chroma format)
Resolution resampling (increasing or decreasing the width or height)
Picture rate upsampling
Bit depth upsampling (increasing the luma bit depth or the chroma bit depth)
Colourization
Application defined

NNPFC: Mode and Identification

- `nnpfc_mode_idc`: Indicates if NN information is in-band and out-of-band.
- `nnpfc_mode_idc` equal to 1 allows usage of any neural-network format other than ISO/IEC 15938-17 in a decentralized manner.
- When `nnpfc_mode_idc` is equal to 1:
  - `nnpfc_tag_uri`: RFC 4151 Tag URI identifying format and associated information for the NNPF.
  - `nnpfc_uri`: IETF Internet Standard 66 identifying neural-network used as a post-filter.
- `nnpfc_id`: Identifies particular NNPFC

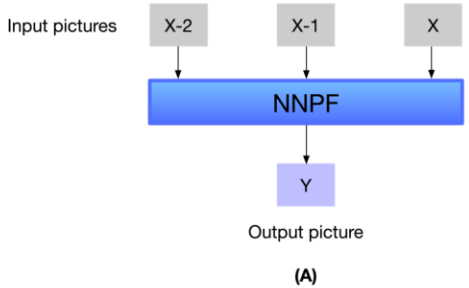
nnpfc_mode_idc	Interpretation
0	In-band signaling of neural-network information in an ISO/IEC 15938-17 bitstream.
1	Out-of-band signaling of neural network information with information and format identified by additional syntax elements
2-255	Reserved for future use

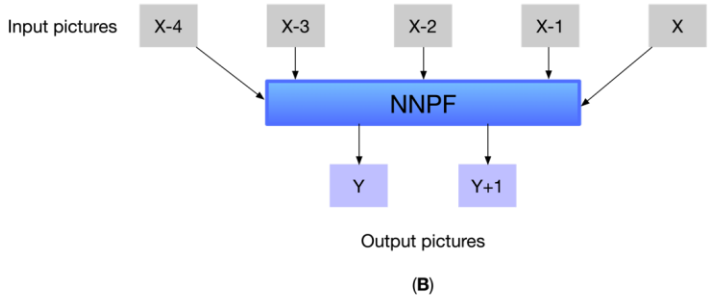
NNPFC: Base and Update

- A base NNPF may be further updated.
- `nnpfc_base_flag`: specifies if SEI defines base NNPF or update relative to a base
- The first NNPFC SEI message in decoding order with a particular `nnpfc_id` within current Coded Layer Video sequence (CLVS) shall be a base NNPF (`nnpfc_base_flag` equal to 1).
- All NNPFC SEI messages in a CLVS with a particular `nnpfc_id` value which specify a base NNPF (`nnpfc_base_flag` equal to 1) are required have the same SEI payload content.
- NNPF update is relative to preceding NNPF base.
- Updates are not cumulative.
- If base and update NNPFs are signalled NNPF activation SEI can activate either base NNPF or update NNPF (controlled by `nnpfa_target_base_flag`).

NNPFC: Input Pictures

- NNPF can use one or more input pictures.
- `nnpfc_num_input_pics_minus1`: specifies number of input pictures (up to 63).
- Picture rate upsampling requires use of at least two input pictures.
- Care needed to define input pictures at the start and end of a CLVS.
  - `nnpfc_absent_input_pic_zero_flag` : specifies if NNPF should use 0 value for sample arrays of input pictures which are not present (flag equal to 1) or should use an input picture that is closest to the input picture which is not present (flag equal to 0).





NNPFC: Output Pictures

- Syntax to control NNPF output pictures
  - `nnpfc_interpolated_pics[i]` specifies the number of interpolated pictures generated by the NNPF between the i-th and the (i + 1)-th input picture.
  - `nnpfc_ input_pic_filtering_flag[i]` equal to 1 (or equal to 0) specifies that i-th input picture to the NNPF generates (or does not generate) an output picture.
- Variables calculated:
  - `numPicsInOutputTensor` specifies the total number of pictures present in the output tensor of the NNPF
  - `NumInpPicsInOutputTensor` specifies the number of pictures that have a corresponding input picture and are present in the output tensor of the NNPF.
- Actual number of NNPF generated pictures output by NNPF process is further controlled by NNPFA SEI message

NNPFC: Input Pictures Information

- Properties and information about input pictures:
  - Input format indicator (`nnpfc_inp_format_idc`): specifies the method of converting sample value of input picture to NNPF input value – as either real numbers or unsigned integers.
  - Input order indicator (`nnpfc_inp_order_idc`): specifies method used for ordering sample arrays of input picture to input tensor to the NNPF (including luma, chroma and auxiliary input matrices).
  - Auxiliary input indicator (`nnpfc_auxiliary_inp_idc`): indicates if auxiliary input matrix data is included in input tensor to NNPF. The auxiliary input matrices include information about filtering strength control value array for each input picture.
  - Input tensor bit depth: specifies luma (`nnpfc_inp_tensor_luma_bitdepth_minus8`) and/or chroma (`nnpfc_inp_tensor_chroma bitdepth_minus8`) input integer tensor sample value bit depth.

NNPFC: Output Pictures Information

- Properties and information about output pictures:
  - Output format indicator (`nnpfc_out_format_idc`): specifies format of sample values output by NNPF value – as either real numbers of unsigned integers.
  - Output order indicator (`nnpfc_out_order_idc`): specifies method used for NNPF output sample order (including luma, chroma matrices).
  - Output tensor bit depth: specifies luma (`nnpfc_out_tensor_luma_bitdepth_minus8`) and/or chroma (`nnpfc_out_tensor_chroma bitdepth_minus8`) output integer tensor sample value bit depth.
- Properties and information about output tensor may be present:
  - Colour description: information is specified for the picture resulting from NNPF including colour primaries (`nnpfc_colour_primaries`), transfer characteristics (`nnpfc_transfer_characteristics`), matrix coefficient (`nnpfc_matrix_coeffs`), full range flag (`nnpfc_full_range_flag`).
  - Chroma information: chroma sample location type information is provided by the syntax element `nnpfc_chroma_sample_loc_type_frame`.

NNPFC: Patch, Overlap, Padding

- NNPF operates on patches and information about it is signaled via following syntax:
  - Overlap sample count: specifies the overlapping horizontal and vertical sample counts of adjacent input tensors for the NNPF (`nnpfc_overlap`).
  - Constant patch size flag: `nnpfc_constant_patch_size_flag` specifies if NNPF accepts exactly specified patch size or any patch size which is a multiple of a patch width and patch height.
  - Patch width and height: When NNPF accepts constant patch size, patch width and height is signaled with minus 1 offset via syntax elements `nnpfc_patch_width_minus1` and `nnpfc_patch_height_minus1`.
  - Extended patch width and height: When NNPF accepts any patch size which is multiple of a patch width and height, information about the common divisor of all allowed width and heights of the extended patch required as input to the NNPF is signaled via the syntax elements `nnpfc_extended_patch_width_cd_delta_minus1` and `nnpfc_extended_patch_height_cd_delta_minus1`.
  - Padding type: `nnpfc_padding_type` describes padding for sample locations outside boundaries of input picture (zero padding, replication padding, reflection padding, wrap-around padding, fixed padding, and other padding types reserved for future).

NNPFC: *DeriveInputTensors, StoreOutputTensors*

- Input and output tensors are created utilizing input, output picture information signalled.
- Two processes are defined:
  - The process *DeriveInputTensors( ):*

    for deriving the input tensor for a given top-left sample location for the patch of samples included in the input tensor is specified.
  - The process *StoreOutputTensors( )*:

    for deriving sample values in the filtered Y, Cb, Cr sample arrays for the NNPF-generated picture(s), from the output tensor for a given top-left sample location for the patch of samples is specified.

NNPFC: Complexity Information

- The neural-network post filtering may become complex. Information is optionally signaled regarding complexity of the NNPF to help selection.
- Complexity information includes:
  - Whether neural network uses integer, binary, floating point and integer parameters (`nnpfc_parameter_type_idc`) and bit length of the parameters (`nnpfc_log2_parameter_bit_length_minus3`).
  - Maximum number of neural network parameters for the NNPF (`nnpfc_num_parameters_idc`).
  - Maximum number of multiply-accumulate operation per sample of the NNPF (`nnpfc_num_kmac_operations_idc`)
  - Total size in kilobytes to store uncompressed neural network parameters (`nnpfc_total_kilobyte_size`).

Neural-Network Post-Filter Activation (NNPFA)

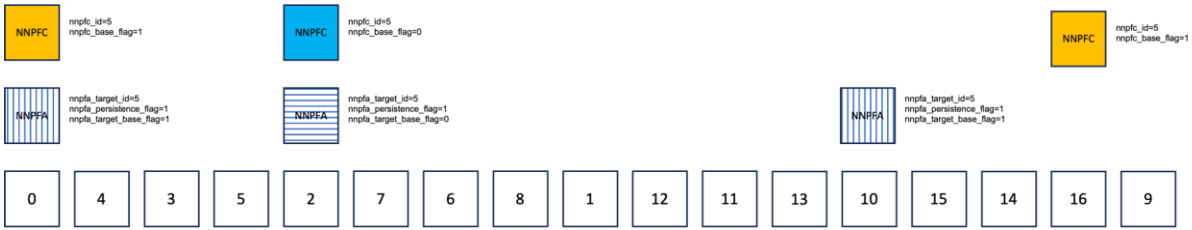
- The use of specified neural-network post-processing filters (NNPFs) for specific pictures is indicated with neural-network post-filter activation (NNPFA) SEI messages.
- NNPFA SEI message activates or deactivates use of a target NNPF.
  - NNPFA SEI includes a `nnpfa_target_id` syntax element which specifies the value of `nnpfc_id` of the target NNPF that is activated or deactivated.
- NNPFA SEI also includes `nnpfa_cancel_flag` and `nnpfa_persistence_flag` which are largely defined in the manner similar to cancel and persistence flags in other SEIs.
- NNPFA includes following information:
  - NNPF output pictures control information
  - Base or updated NNPF activation
  - NNPF coded layer video sequence (CLVS) flags

NNPFA: Output Pictures Control Information

- As mentioned, the actual number of NNPF-generated pictures output by the NNPF process is further controlled by activation of a NNPFA SEI message.
- This is based on the NNPFA syntax elements: `nnpfa_num_output_entries` and `nnpfa_output_flag[i]`, which have the following meaning:
  - `nnpfa_num_output_entries` specifies how many `nnpfa_output_flag[i]` values are signalled, which are upper bounded by how many input pictures are present in the output tensor (`NumInputPicsInOutputTensor`).
  - `nnpfa_output_flag[i]` equal to 1 (or equal to 0) specifies that NNPF-generated picture corresponding to input picture (having index `InpIdx[i]`) is output (or is not output) by the NNPF process.

NNPFA: Base and Updated NNPF Activation

- `nnpfa_target_base_flag` equal to 1 specifies that the target NNPF is the base NNPF with `nnpfc_id` equal to `nnpfa_target_id`. `nnpfa_target_base_flag` equal to 0 specifies that the target NNPF is the NNPF specified by the last NNPFC SEI message with `nnpfc_id` equal to `nnpfa_target_id` that precedes the first VCL NAL unit of the current picture in decoding order and is not a repetition of the NNPFC SEI message that contains the base NNPF.
- Example:
  - Picture 0: NNPFC and NNPFA is signaled which activates this base NNPF.
  - Picture 2 an NNPFC update with same `nnpfc_id` of 5 is signaled and that NNPF update is immediately activated by NNPFA with `nnpfa_target_base_flag` equal to 0.
  - Picture 10: base NNPF is re-activated with a new NNPFA and `nnpfa_target_base_flag` equal to 0.
  - Picture 16 the base NNPFC is repeated (for loss resilience/ random access).



NNPFA: CLVS Flags

- Special care is needed to specify how NNPF may be applied at the start and end of a CLVS, especially when the NNPF is used for picture rate upsampling with multiple input pictures.
- The goal is to specify if input pictures to NNPF are all from the same CLVS, to seamlessly handle a scenario where required number of input pictures for NNPF may cross a CLVS boundary and if splicing may have occurred.
- For this, two flags `nnpf_no_prev_clvs_flag`, and `nnpfa_no_foll_clvs_flag` are defined in NNPFA SEI.

NNPF Extensions & Technologies Under Consideration

- NNPF Extensions (VSEI v4 – preliminary WD):
  - Application Information Signaling:
    - When `nnpfc_purpose` is equal to 0, the NNPF may be used as determined by the application.
    - Defines further information about the application-specific purpose when `nnpfc_purpose` is equal to 0.
    - This is achieved by defining a new syntax element (in the NNPFC metadata extension section) `nnpfc_application_purpose_tag_uri` when `nnpfc_purpose` is equal to 0.
  - New NNPF purpose: temporal extrapolation (Jan 2024 meeting)
    - Aimed at low latency cloud gaming/ robotics/ autonomous driving applications.
    - NN for generating one or more pictures following all the input pictures in output order can be specified.
    - Backwards compatible with picture rate interpolation (upsampling).
- NNPF Technologies Under Consideration (TuC):
  - Neural-Network Post-Filter Group Characteristics (NNPFGC):
    - Specifies a neural network post-filter group.
    - The NNPF group can define NNPF cascade or alternatives to each other. An NNPF group may have members that are NNPF or NNPF group.
  - Neural-Network Post-Filter Group Activation (NNPFGA):
    - Activates/ deactivates possible use of the target NNPF group.

NNPF Implementation: Picture Rate Upsampling

- NNPF purpose picture rate upsampling was implemented
  - VSEI signaling makes it possible to integrate of-the-shelf neural networks easily.
  - For our implementation we used the neural network designed in the paper “Channel Attention is All You Need for Video Frame Interpolation,” by Choi et. al.
- Results for interpolating 3 pictures with 2 input pictures for:
  - Tango (4k)
  - Big Buck Bunny (1080P)

NNPF Implementation: Picture Rate Upsampling: Tango

NNPF Implementation: Picture Rate Upsampling: Tango

NNPF Implementation: Picture Rate Upsampling: Tango

NNPF Implementation: Picture Rate Upsampling: Tango

NNPF Implementation: Picture Rate Upsampling: Tango

NNPF Implementation: Picture Rate Upsampling: BBB

NNPF Implementation: Picture Rate Upsampling: BBB

NNPF Implementation: Picture Rate Upsampling: BBB

NNPF Implementation: Picture Rate Upsampling: BBB

NNPF Implementation: Picture Rate Upsampling: BBB

Conclusion

- This paper provided an overview of the VSEI design of NNPF.
- We described neural-network post-filter characteristics (NNPFC) and neural-network post-filter activation (NNPFA) SEIs.
- NNPF VSEI V4 work has started and Preliminary WD includes NNPF extensions
- Additional NNPF technologies are also under consideration for incorporation in future VSEI versions.
- We described our experiments to implement a picture-rate upsampling neural-network post-filter utilizing NNPF VSEI signaling.


SHARP