User Manual

Retos Colaboración RTC-2019-007434-7 - Data compression system.

Document version: 1.1
Date: 1 May 2022

UAB
Universitat Autònoma
de Barcelona

Departament d'Enginyeria
de la Informació
i de les Comunicacions

Group on Interactive
Coding of Images

Document Information

Document identifier: docs/user_manual.md
Title: User Manual
Number of pages: 9
Date: 1 May 2022
Author(s): Miguel Hernández-Cabronero et al.

Change Control

Revision	Date	Changes
1.0	1 Aug 2021	(initial version)
1.1	1 May 2022	added documentation for the new prediction modes

Contents

1 INTRODUCTION

This document is the user manual of the command-line tools implemented for the compression module (PT-03) of the Retos Colaboración RTC-2019-007434-7 project. This module has been developed by the Universitat Autònoma de Barcelona (UAB).

2 TOOLS

Two main tools are provided: a compressor and a decompressor: `v2f_compress` and `v2f_decompress`.

The `v2f_compress` tool accepts 1-byte and 2-byte raw samples from a file (e.g., an image) and produces a compact representation thereof into an output file. In turn, `v2f_decompress` applies the inverse process and produces reconstructs a file of the same length, potentially without loss.

Both tools require an additional input header file describing the V2F compression forests and other parameters to be used for compression. Typically, these headers are stored with `.v2fc` extension. A set of prebuilt headers is already provided. To generated new `.v2fc` files adapted to a given dataset, the scripts under the `../forest_generation` folder can be used to produce `.v2fh` files. These should be moved into `metasrc/prebuilt_forests` and `make` can then be run to produce the corresponding `.v2fc` files. See `metasrc/README.md` for further information on how to create new header files.

An auxiliary tool has been developed, `v2f_verify_codec`, which can be used to validate any existing header file. See its description below for more information on the format of this header files.

3 COMMAND-LINE TOOL USAGE

3.1 v2f_compress

`v2f_compress` - Compress raw data given a V2F codec definition.

3.1.1 Diagram

```
                 +--------------+
raw_file ---->|              |
v2f_codec --->| v2f_compress |---> compressed_file
[options] --->|              |
                 +--------------+
```

3.1.2 Synopsis

```
 v2f_compress [-h] [-v]
              [-q quantizer_mode]
              [-s quantizer_step_size]
              [-d decorrelator_mode]
              [-w samples_per_row [-y shadow_y_start_stop_list]]
              raw_file
              v2f_codec
              compressed_file
```

3.1.3 Description

The command-line program `v2f_compress` reads input data in raw format from `raw_file` and compresses it the file pointed by `compressed_file`.

The `v2f_codec` parameter must point to a file which describes the V2F compressor and other parameters to be used. These parameters can be overwritten using optional arguments in the invocation.

3.1.4 Options

When provided, -q, -s and -d overwrite the quantization and decorrelation parameters set in `v2f_codec` file. When not provided, these values are read from the definition in `v2f_codec`. The -w parameter indicates the number of samples per image row, i.e., the image width. If -w is provided, the -y parameter may be used to indicate a list of start and stop y indices that are to be considered shadow, i.e., they are not compressed and are reconstructed to all zeros. It is the coder-decoder pair's responsibility to employ the same `v2f_codec` /parameter combination to allow reconstruction of the expected output.

-q quantizer mode

Specify the quantization type applied in that stage:

- `0`: no quantization is applied.
- `1`: uniform quantization is applied

-s quantization step size

Specify the quantization step size. It must be an integer between 1 and 255. Use 1 for lossless compression. The quantization step size must be 1 if the quantizer mode is 0 (no quantization).

-d decorrelator mode

Specify the type of decorrelation to apply to the input data. It must be one of:

- `0`: don't apply any decorrelation. The (possibly quantized) samples are coded directly by the entropy coder.
- `1`: apply DPCM prediction using the immediately previous sample, and code the prediction errors.
- `2`: apply prediction using the average of the two previous samples.
- `3`: apply the prediction function employed by the JPEG-LS algorithm
- `4`: apply prediction using the average of two left, left, left-north and north samples

-w samples per row (width)

For 2D or 3D data, and for decorrelation modes 3 and 4, this parameter must be provided, indicating the number of samples per row, i.e., the image width.

-y list of shadow start and stop y indices

When -w is provided, the -y can be specified so that one or more horizontal shadow regions can be identified (and not coded). If present, the format of this argument is

```
-y start1,end1[,start2,end2[,...[,startN,endN]]]
```

where start1,end1 are the first and last y indices (starting with 0) of the first shadow region, and so on. For instance, an image with two shadows, one at rows 10 to 70 (both included) and another at rows 100 to 150 (both included) could be coded with the following argument:

```
-y 10,70,100,150
```

Note that:

- This string may contain only comma and number characters.
- The end y position of all regions must be at least as large as the corresponding start y position.
- No overlap is accepted between different shadow regions.

-v version information

Tool shows version information.

-h show help

Tool shows help information.

3.1.5 Data format specification

raw_file

The data in raw_file must be stored using 1-byte or 2-byte big-endian words in raster BSQ order.

The number of input samples in raw_file is automatically deducted from the codec configuration described in v2f_codec. In case 2D is being compressed, the -w parameter must be used to specify the row width.

The size of raw_file is arbitrary, except that it must be a multiple of the number of bytes per sample configured in the codec.

v2f_codec

The data in v2f_codec must describe a compressor/decompressor pair, including the quantization, decorrelation and entropy coding parameters. The format of this file is fully described in the description of the v2f_verify_codec tool below.

compressed_file

The sequence of input samples can be split by the encoder into one or more blocks of contiguous samples maintaining a raster, BSQ ordering. This sequence of blocks is compressed maintaining that same order.

Each block is compressed independently, producing a block envelope as a result. The contents of compressed_file are the concatenation of these envelopes.

The contents of each block envelope are as follows:

- compressed_bitstream_size: 4 bytes, unsigned big-endian integer. Number of bytes in the compressed_bitstream field, defined below. If this value is zero, it indicates a shadow region that has not been compressed and is intended to be reconstructed as all zeros.
- sample_count: 4 bytes, unsigned big-endian integer. The total number of input samples in the original block. Not to be confused with the number of emitted codewords.
- compressed_bitstream: compressed_bitstream_size bytes. It contains the compact representation of the data in the block. Its format is bit-precise defined by the input, the employed v2f_codec and any modifier options. This format is a succession of fixed-length V2F tree node indices, each of which represents one or more coded symbols. This field is only present if compressed_bitstream_size is not zero.

3.1.6 Usage example

The following example demonstrates how to compress the test image available at example_sample_u8be-1x256x256.raw. The compressed data are written into example.v2f. The V2F codec stored in v2f_codec.v2f is employed, applying uniform quantization (-q 1), a step size 5 (-s 5, resulting in a maximum absolute error of at most 2), and left DPCM prediction (-d 1).

```
./v2f_compress -q 1 -s 5 -d 1 \\
               example_sample_u8be-1x256x256.raw \\
               v2f_codec.v2fc \\
               example.v2f
```

3.1.7 Author information

Software development:

- Miguel Hernández-Cabronero miguel.hernandez@uab.cat
- Artur Llabrés artur.llabres@uab.cat
- Natalia Blasco natalia.blasco@uab.cat
- Ester Jara ester.jara@uab.cat

Project management:

- Miguel Hernández-Cabronero miguel.hernandez@uab.cat
- Joan Serra-Sagristà joan.serra@uab.cat

Technical supervision:

- Javier Marin javier.marin@satellogic.com
- David Vilaseca vila@satellogic.com

Produced by Universitat Autònoma de Barcelona (UAB) for Satellogic.

3.2 v2f_decompress

v2f_decompress - Decompress a file produced by v2f_compress.

3.2.1 Diagram

```
                     +----------------+
compressed_file ---->|                |
      v2f_codec ---->| v2f_decompress |---> reconstructed_file
      [options] ---->|                |
                     +----------------+
```

3.2.2 Synopsis

```
v2f_decompress [-h] [-v]
               [-q quantizer_mode]
               [-s quantizer_step_size]
               [-d decorrelator_mode]
               [-w samples_per_row]
               compressed_file
               v2f_codec
               reconstructed_file
```

3.2.3 Description

This tool accepts a file compressed_file produced by the v2f_compress tool, and stores the reconstructed image into reconstructed_file.

The v2f_codec parameter must point to a copy of the V2F header (typically with .v2fc extension) used for compression.

3.2.4 Options

When provided, -q, -s and -d overwrite the quantization and decorrelation parameters set in v2f_codec file. When not provided, these values are read from the definition in v2f_codec.

For 2D data, the -w argument must be passed to inform the decoder of the number of samples in each row. Note that the -y argument of the encoder is not present in the decoder.

It is the coder-decoder pair's responsibility to employ the same v2f_codec /parameter combination to allow reconstruction of the expected output.

-q quantizer mode

Specify the quantization type applied in that stage:

```
- 0: no quantization is applied.
- 1: uniform quantization is applied
```

-s quantization step size

Specify the quantization step size. It must be an integer between 1 and 255. Use 1 for lossless compression. The quantization step size must be 1 if the quantizer mode is 0 (no quantization).

-d decorrelator mode

Specify the type of decorrelation applied by the encoder. It must be one of:

- `0`: didn't apply any decorrelation. The (possibly quantized) samples were coded directly by the entropy coder.
- `1`: applied DPCM prediction using the immediately previous sample, and codeed the prediction errors.
- `2`: applied prediction using the average of the two previous samples.
- `3`: applied the prediction function employed by the JPEG-LS algorithm
- `4`: applied prediction using the average of two left, left, left-north and north samples

-w samples per row

For 2D data, if decorrelators `3` or `4` were selected, this parameter must be passed to indicate the number of samples per row, i.e., the image width.

-v show version

Tool shows version information.

-h show help

Tool shows help information.

3.2.5 Data format specification

`compressed_file`

Compressed file as produced by `v2f_compress`. Please refer to the help of that tool for more information on the compressed data format.

`v2f_codec`

The data in `v2f_codec` must describe a compressor/decompressor pair, including the quantization, decorrelation and entropy coding parameters. The format of this file is fully described in the description of the `v2f_verify_codec` tool below.

`reconstructed_file`

A reconstructed version of the original data. If no quantization is applied, or a quantization step size of 1 was selected, the reconstructed data are expected to be identical to the original data.

3.2.6 Usage example

The following example demonstrates how to decompress `example.v2f` and store the resulting samples into `reconstructed.raw`.

The `v2f_codec.v2fc` file should be identical to the one employed during compression, and the same option flags should be applied.

```
./v2f_compress example.v2f \\
               v2f_codec.v2fc \\
               reconstructed.raw \\
```

3.2.7 Author information

Software development:

- Miguel Hernández-Cabronero `miguel.hernandez@uab.cat`
- Artur Llabrés `artur.llabres@uab.cat`
- Natalia Blasco `natalia.blasco@uab.cat`
- Ester Jara `ester.jara@uab.cat`

Project management:

- Miguel Hernández-Cabronero `miguel.hernandez@uab.cat`
- Joan Serra-Sagristà `joan.serra@uab.cat`

Technical supervision:

- Javier Marin `javier.marin@satellogic.com`
- David Vilaseca `vila@satellogic.com`

Produced by Universitat Autònoma de Barcelona (UAB) for Satellogic.

3.3 v2f_verify_codec

`v2f_verify_codec` - Verify a V2F header file and show basic information about it.

3.3.1 Diagram

```
                +-------------------+
v2f_codec ---->| v2f_verify_codec |
                +-------------------+
```

3.3.2 Synopsis

```
v2f_verify_codec [-h] [-v] v2f_codec
```

3.3.3 Description

This tool verifies the V2F header in `v2f_codec` and shows basic information about it.

The verification includes both syntactic parsing, and a fast compression/decompression test using a small synthetic sample.

Any errors during loading, compression or decompression are reported by this tool. Otherwise, basic information is about the verified codec is shown.

3.3.4 Options

The following options are recognized by this tool.

-v

Tool shows version information.

-h

Tool shows help information.

3.3.5 Data format specifications

`v2f_codec`

A valid V2F codec definition should contain the following elements, in this order:

- Quantizer parameters:
  - `quantizer_mode`: 1 byte, must correspond to one of @ref v2f_quantizer_mode_t. See the -q parameter help for available quantizer modes.
  - `quantizer_step_size`: 4 bytes, unsigned big endian. See the -s parameter help for valid sizes.
- Decorrelator parameters:
  - `decorrelator_mode`: 2 bytes, unsigned big endian, must correspond to one of @ref v2f_decorrelator_mode_t. See the -d parameter help for available decorrelator modes.
  - `max_sample_value`: 4 bytes, unsigned big endian. Must be at least as large as the maximum sample value for the V2F forest defined below, if present. Note that unsigned big endian raw samples are expected.

- Entropy coding parameters:
  - `forest_id`: 4 bytes, unsigned bit endian. If this field is set to 0, it indicates that an explicit definition of the V2F forest is included afterwards. If set to a larger value v, the (v-1)-th prebuilt forest is used. In this case, the decoder must know what this index refers to.
  - `v2f_forest`: If `forest_id` is not 0, this field contains a full V2F entropy coder/decoder pair definition. If `forest_id` is 0, this field is not present.

When present, the format of `v2f_forest` must be as follows:

- `number of entries`: total number of entries, counting those that do not have a word assigned and are not needed by the decoder, 4 bytes, big endian, unsigned, See the `V2F_C_MIN_ENTRY_COUNT` and `V2F_C_MAX_ENTRY_COUNT` constant definitions in the code.
- `bytes per word`: number of bytes used to represent each of the nodes that are included in the V2F dictionary, 1 byte, unsigned, See the `V2F_C_MIN_BYTES_PER_WORD` and `V2F_C_MAX_BYTES_PER_WORD` constant definitions in the code.
- `bytes per sample`: samples are represented with these many bytes, 1 byte, unsigned, See the `V2F_C_MIN_BYTES_PER_SAMPLE` and `V2F_C_MAX_BYTES_PER_SAMPLE` constant definitions in the code.
- `max expected value`: max expected input sample value, 2 bytes, big endian, unsigned, See the `V2F_C_MAX_SAMPLE_VALUE` constant definition in the code. Must be consistent with `bytes per sample`.
- `root count`: n-1, where n is the number of root nodes included, must be >= 1, 2 bytes BE unsigned. See the `V2F_C_MAX_ROOT_COUNT` and `V2F_C_MAX_CHILD_COUNT` constant definitions in the code.
- `For each root`:
  - `total entry count` : total number of entries in this root, 4 bytes, big endian. See the `V2F_C_MIN_ENTRY_COUNT` and `V2F_C_MAX_ENTRY_COUNT` constant definitions in the code.
  - `number of included entries`: total number of entries, excluding those that do not have a word assigned, 4 bytes, big endian, unsigned. Must be consistent with the value of `bytes per word`.
  - `entries` : variable length, `total entry count` elements (included those not included in the decoder dict) ordered by index For each entry in this root:
    - `index`: 4 bytes, big endian, unsigned. Even though this value can be deduced by the `ordered by index` constraint, this value is maintained to facilitate verification.
    - `number of children` : 4 bytes, big endian, unsigned See the `V2F_C_MAX_CHILD_COUNT` constant definition in the code.

    If `number of children` > 0, then the following items are also included for this entry of this root:

    - `children indices`: `number of children` 4 bytes, ordered by corresponding sample index,

    If `number of children` is not identical to `max expected value` + 1, then node is included in the V2F decoder and has an assigned codeword. In this case, the following fields are present:

    - `sample count` : number of samples represented by this index, 2 bytes, big endian, See the `V2F_C_MIN_SAMPLE_COUNT` and `V2F_C_MAX_SAMPLE_COUNT` constant definitions in the code.
    - samples : `bytes per sample` `sample count` sample values in `bytes per sample` bytes per sample, big endian, unsigned. Not present if `sample count` is zero.
    - word : bytes corresponding to this included node, `bytes per word` bytes, big endian, unsigned
  - `number of children` of the root node: 4 bytes, big endian, unsigned This number may not exceed the maximum number of possible sample values, but it may be lower than that. If it is, the number of children must be (possible values - i), where i is the index of this root. In this case, children must be assigned for input sample values from i onwards.
  - `indices of the children` of the root node: indices of the root's entries order by input symbol. For each child:
    - `index`: 4 bytes, unsigned big endian
    - `input symbol value`: `bytes per sample` bytes, unsigned big endian

Note that root count can be smaller than `max expected value` In that case, it is to be interpreted that all remaining roots are identical to the last one included.

3.3.6 Usage example

The following invocation example verifies the contents of `codec_header.v2fc` and provides basic information about it in case it is valid:

```
./v2f_verify_codec codec_header.v2fc
```

3.3.7 Author information

Software development:

- Miguel Hernández-Cabronero `miguel.hernandez@uab.cat`

Project management:

- Miguel Hernández-Cabronero `miguel.hernandez@uab.cat`
- Joan Serra-Sagristà `joan.serra@uab.cat`

Technical supervision:

- Javier Marin `javier.marin@satellogic.com`
- David Vilaseca `vila@satellogic.com`

Produced by Universitat Autònoma de Barcelona (UAB) for Satellogic.