

## SMPTE STANDARD

# VC-5 Video Essence

## Part 1: Elementary Bitstream



Page 1 of 50 pages

| Table of Contents                        | Page |
|------------------------------------------|------|
| Foreword .....                           | 5    |
| Intellectual Property .....              | 5    |
| Introduction.....                        | 5    |
| 1 Scope .....                            | 7    |
| 2 Conformance Notation .....             | 7    |
| 3 Normative References .....             | 7    |
| 4 Glossary .....                         | 7    |
| 5 Notation.....                          | 12   |
| 5.1 Codec State Parameters .....         | 12   |
| 5.2 Number Representations.....          | 12   |
| 5.3 Arithmetic Right Shift .....         | 12   |
| 6 Overview (Informative) .....           | 12   |
| 7 Component Arrays .....                 | 14   |
| 7.1 Component Array Requirements .....   | 14   |
| 7.2 Component Array Representation ..... | 14   |
| 7.3 Component Array Dimensions.....      | 14   |
| 7.4 Component Value Precision .....      | 14   |
| 8 Bitstream Syntax .....                 | 16   |
| 8.1 Syntax Elements.....                 | 16   |
| 8.2 Bitstream Segments .....             | 16   |
| 8.3 Tag-Value Pairs .....                | 16   |
| 8.3.1 Tag-Value Syntax.....              | 16   |
| 8.3.2 Required Tag-Value Pairs .....     | 16   |
| 8.3.3 Optional Tag-Value Pairs .....     | 16   |
| 8.4 Chunk Elements.....                  | 16   |
| 8.4.1 Chunk Element Syntax.....          | 16   |
| 8.4.2 Chunk Payload Size .....           | 17   |
| 8.4.3 Small Chunk Payloads .....         | 17   |
| 8.4.4 Large Chunk Payloads .....         | 17   |
| 8.5 Bitstream Constraints .....          | 17   |
| 8.6 Bitstream Start Marker.....          | 18   |
| 8.7 Bitstream Header .....               | 18   |
| 8.8 Codeblock Elements.....              | 18   |

|         |                                                     |    |
|---------|-----------------------------------------------------|----|
| 9       | Codec State .....                                   | 18 |
| 9.1     | Codec State Parameters .....                        | 18 |
| 9.2     | Codec State Initialization .....                    | 19 |
| 9.3     | External Parameter Values .....                     | 19 |
| 9.4     | Explicit Parameter Changes .....                    | 19 |
| 9.5     | Codec State Changes .....                           | 19 |
| 9.5.1   | Subband State Change .....                          | 19 |
| 9.5.2   | Channel State Change .....                          | 19 |
| 10      | Subband Coding .....                                | 20 |
| 10.1    | Coefficient Order .....                             | 20 |
| 10.2    | Subband Codeblocks .....                            | 20 |
| 10.3    | Lowpass Band Codeblocks .....                       | 20 |
| 10.4    | Highpass Band Codeblocks .....                      | 20 |
| 10.4.1  | Run-Length Coding .....                             | 20 |
| 10.4.2  | Non-Zero Coded Values .....                         | 20 |
| 10.4.3  | Band End Marker .....                               | 21 |
| 11      | Wavelet Transforms .....                            | 21 |
| 11.1    | Spatial Wavelet Transform (Informative) .....       | 21 |
| 11.2    | Spatial Wavelet Sequence .....                      | 23 |
| 11.3    | Inverse Wavelet Transforms .....                    | 26 |
| 11.4    | Wavelet Transform Requirements .....                | 27 |
| 11.4.1  | Wavelet Height Requirement .....                    | 27 |
| 11.4.2  | Wavelet Width Requirement .....                     | 27 |
| 12      | Decoding Process .....                              | 28 |
| 12.1    | Subband Decoding .....                              | 28 |
| 12.2    | Entropy Decoding .....                              | 28 |
| 12.3    | Inverse Companding .....                            | 28 |
| 12.4    | Inverse Quantization .....                          | 28 |
| 12.5    | Inverse Wavelet Processing .....                    | 28 |
| 12.6    | Component Array Order .....                         | 28 |
| Annex A | Inverse Spatial Wavelet Transform (Normative) ..... | 29 |
| A.1     | Inverse Horizontal and Vertical Transforms .....    | 29 |
| Annex B | Codec State Parameters (Normative) .....            | 30 |
| B.1     | Parameter Descriptions .....                        | 30 |
| B.1.1   | ImageWidth .....                                    | 30 |
| B.1.2   | ImageHeight .....                                   | 30 |
| B.1.3   | BitsPerComponent .....                              | 30 |
| B.1.4   | ChannelCount .....                                  | 30 |
| B.1.5   | SubbandCount .....                                  | 30 |
| B.1.6   | ChannelNumber .....                                 | 30 |
| B.1.7   | SubbandNumber .....                                 | 30 |
| B.1.8   | LowpassPrecision .....                              | 30 |
| B.1.9   | Quantization .....                                  | 30 |
| B.1.10  | PrescaleShift .....                                 | 30 |
| B.1.11  | ChannelWidth .....                                  | 31 |
| B.1.12  | ChannelHeight .....                                 | 31 |
| B.2     | Tag-Value Pairs .....                               | 31 |
| B.3     | Chunk Headers .....                                 | 32 |
| B.3.1   | Small Chunk Headers .....                           | 32 |
| B.3.2   | Large Chunk Headers .....                           | 32 |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| B.4 Image Dimensions (Informative) .....                        | 33 |
| Annex C Codebook (Normative) .....                              | 34 |
| C.1 Coefficient Magnitudes and Zero Runs .....                  | 34 |
| C.2 Special Codewords .....                                     | 39 |
| Annex D Decoding Process (Informative) .....                    | 40 |
| D.1 Decoding Overview .....                                     | 40 |
| D.2 Initial Codec State .....                                   | 40 |
| D.3 Memory Allocation .....                                     | 40 |
| D.4 Entropy Decoding .....                                      | 40 |
| D.5 Lower Resolution Decoding .....                             | 41 |
| D.6 Image Repacking Process .....                               | 41 |
| D.7 Decoding Resource Requirements .....                        | 41 |
| Annex E Encoding Process (Informative) .....                    | 42 |
| E.1 Codec State Synchronization .....                           | 42 |
| E.2 Image Processing .....                                      | 43 |
| E.3 Image Unpacking .....                                       | 43 |
| E.4 Prescale Shifts .....                                       | 43 |
| E.5 Lowpass Transform .....                                     | 43 |
| E.6 Highpass Transform .....                                    | 43 |
| E.7 Quantization .....                                          | 44 |
| E.8 Companding .....                                            | 44 |
| E.9 Entropy Coding .....                                        | 45 |
| E.9.1 Non-Zero Coefficients .....                               | 45 |
| E.9.2 Runs of Zeros .....                                       | 45 |
| E.10 Subband Encoding Order .....                               | 45 |
| Annex F Inverse Companding and Dequantization (Normative) ..... | 46 |
| Annex G Additional Notation (Informative) .....                 | 47 |
| G.1 Arithmetic Operators .....                                  | 47 |
| G.2 Logical Operators .....                                     | 47 |
| G.3 Relational Operators .....                                  | 47 |
| G.4 Bitwise Operators .....                                     | 48 |
| G.5 Assignment Operator .....                                   | 48 |
| G.6 Operator Precedence .....                                   | 48 |
| G.7 Compound Statements .....                                   | 49 |
| G.8 Elementary Functions .....                                  | 49 |
| G.9 Bitstream Functions .....                                   | 49 |
| Annex H Bibliography (Informative) .....                        | 50 |

## Figures

|                                                                                                                                                                                                                                              |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1 – Overview of the encoding and decoding processes (informative). ....                                                                                                                                                               | 13 |
| Figure 2 – Component arrays are represented as separate channels in the bitstream. ....                                                                                                                                                      | 15 |
| Figure 3 – Two-dimensional spatial wavelet transform as a horizontal wavelet transform followed by<br>a vertical wavelet transform. ....                                                                                                     | 22 |
| Figure 4 – Three levels of spatial wavelet transform showing the lowpass bands used as inputs to the<br>next wavelet level and the numbering of subbands. Each channel is encoded using this<br>sequence of spatial wavelet transforms. .... | 25 |
| Figure 5 – Sequence of operations used to encode a component array into the bitstream. ....                                                                                                                                                  | 26 |
| Figure E.1 – Synchronizing the codec state between the encoder and decoder. ....                                                                                                                                                             | 42 |

## Tables

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| Table 1 – Subband numbering for the sequence of wavelets in each channel. ....             | 23 |
| Table 2 – Notation used in Figure 5. ....                                                  | 24 |
| Table B.1 – Packed representation of the prescale shifts. ....                             | 31 |
| Table B.2 – Codec state parameters. ....                                                   | 32 |
| Table B.3 – Symbolic names and numbers for tags that identify large chunk elements. ....   | 33 |
| Table C.1 – Codebook for non-negative coefficients (codewords listed in hexadecimal). .... | 35 |
| Table C.2 – Special codewords. ....                                                        | 39 |
| Table D.1 – Highpass band codeblock decoding algorithm. ....                               | 41 |
| Table E.1 – Prescale shifts versus the number of bits per component value. ....            | 43 |
| Table G.1 – Elementary arithmetic operators used in this document. ....                    | 47 |
| Table G.2 – Logical operators used in this document. ....                                  | 47 |
| Table G.3 – Relational operators used in the document. ....                                | 47 |
| Table G.4 – Bitwise operators used in this document. ....                                  | 48 |
| Table G.5 – Assignment operators used in this document. ....                               | 48 |
| Table G.6 – Operator precedence. ....                                                      | 48 |
| Table G.7 – Elementary functions operate on the argument x with unlimited precision. ....  | 49 |
| Table G.8 – Functions for sequentially reading the bitstream. ....                         | 49 |

## Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 2073-1 was prepared by Technology Committee 10E.

## Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

## Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

The VC-5 codec is a variable-bit-rate codec intended for high-quality video acquisition and post-production, capable of encoding diverse image and video formats.

The key design goals of the VC-5 codec are:

- (1) Very high visual quality (visually lossless in most applications);
- (2) Efficient implementation of both decoders and encoders;
- (3) Support for any color space or color difference component sampling;
- (4) Direct encoding of camera sensor output without conversion to a different format;
- (5) Adaptability and ease of use in video editing and post-production.

The VC-5 codec uses reversible integer wavelet transforms applied to the entire image to prevent blocking artifacts. Since the transforms are reversible, the only losses are due to quantization. Lowpass wavelet coefficients are encoded without quantization to preserve visual quality.

The wavelet transforms can be implemented efficiently using only integer shift and add operations. Highpass wavelet coefficient values equal to zero are run length coded. Runs of zeros and non-zero coefficients are represented using variable-length codes that are efficient for both decoding and encoding.

The VC-5 bitstream can represent any image that can be unpacked into component arrays by an image unpacking process prior to encoding (Figure 1). A component array is like an image plane and typically would contain a single type of color or data, although component arrays containing multiple types of image data are not precluded by this standard. Each of the component arrays represented in the bitstream can have different dimensions.

The image unpacking process is not specified by this standard and can implement image pre-processing algorithms, which also are not defined in this standard. The encoding process takes an ordered set of component arrays and represents the component arrays in the bitstream such that a decoding process can reconstruct the set of component arrays that were input to the encoding process in the same order as input to the encoding process. The reconstructed set of component arrays duplicates the component arrays input to the encoder except for losses due to compression.

The decoding process can be followed by an image repacking process that converts the ordered set of component arrays into a packed representation of the image. The image repacking process is not specified by this standard. The image repacking process can implement image post-processing algorithms as required by the application. An image display process that transforms the packed image into a displayable picture can follow the image repacking process. For example, the packed image can include data values that must be mapped to colors to create a displayable image.

The VC-5 codec is intended to accept common image formats, including images with RGB and YCbCr color components with an optional alpha channel. Image formats can use color difference component sampling, for example YCbCr with 4:2:2 or 4:2:0 sampling.

The VC-5 codec is not restricted to a particular set of image dimensions, pixel formats, or color standards. Color Filter Array (CFA) images, including Bayer pattern images, can be unpacked into component arrays that can be represented in a VC-5 bitstream without conversion to a different image format. Component arrays containing non-color data, such as disparity values, can be represented in a VC-5 bitstream. Component arrays can use up to 24 bits of precision to represent each value in the component array.

The syntax of a VC-5 bitstream is based on tag-value pairs that allow for future extensions. Each tag is a 16-bit number and only a small number of tags have been assigned. The value is a 16-bit number, but a tag-value pair can be used to represent the type and size of a large block of data, providing support for values larger than 16 bits.

The number of component arrays and the width and height of each component array is limited only by the scheme for representing information as tag-value pairs.

A VC-5 bitstream can include all of the information required by a decoding process to reconstruct the component arrays represented in the bitstream, but the bitstream does not have to include information that is provided by an external source. For example, a bitstream that is embedded in a media container can omit information that is provided by the container.

The VC-5 codec can adapt to specific applications by representing an image using minimal syntax or including extensive information required to describe the output of sophisticated image sensors or complex workflows.

Wavelet transforms are applied recursively to each component array with the lowpass band from each wavelet transform input to the next transform. Fewer inverse transforms can be applied during decoding to produce a lower resolution proxy image, allowing a decoder to balance decoding speed and image quality as required by specific applications.

## 1 Scope

This standard specifies the compressed representation and decoding process for the core VC-5 bitstream.

## 2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; tables shall be next; followed by formal languages; then figures; and then any other language forms.

## 3 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

None.

## 4 Glossary

For the purposes of this standard, the following definitions shall apply:

### 4.1

#### **big-endian**

order of bytes that comprise a number with more significant bytes before less significant bytes

## 4.2

### **bitstream**

ordered sequence of bits that forms the representation of an image

## 4.3

### **bitstream position**

number of bits counted from the beginning of the bitstream

## 4.4

### **byte-aligned**

bitstream position that is a multiple of 8 bits (one byte)

## 4.5

### **byte**

set of eight consecutive bits starting at a bitstream position that is a multiple of 8 bits

## 4.6

### **camel caps**

refers to the practice of writing compound words using one or more internal upper case letters; such words are usually formed from phrases where the individual words or elements are joined without spaces, and where each element begins with a capital letter within the compound word. For example, CamelCaps

## 4.7

### **channel**

representation of a component array in the bitstream

## 4.8

### **chunk**

sequence of bytes in the bitstream used to represent information that cannot be represented by the value in a tag-value pair

## 4.9

### **codeblock**

representation of a two-dimensional array of coefficients corresponding to a wavelet band

## 4.10

### **codec**

truncation of the terms “coder” or “encoder” and “decoder”

## 4.11

### **codec state**

set of parameters in effect at a specific position in the bitstream that govern how the decoder processes codeblocks and reconstructs component arrays

## 4.12

### **codeword**

contiguous sequence of bits that represents an entry in the codebook

## 4.13

### **coefficient**

scalar frequency value that is the result of a wavelet transform

## 4.14

### **companding**

transform using a monotonic non-decreasing function to reduce the range of quantized coefficients

## 4.15

### **component array**

two-dimensional array of scalar values

**4.16****component value**

scalar value in a component array

**4.17****convolution filter**

filter bank

**4.18****decoder**

embodiment of a decoding process and optionally an image repacking process

**4.19****decoding process**

process that consumes a bitstream to produce an ordered set of component arrays

**4.20****discrete wavelet transform (DWT)**

transform that computes a set of lowpass coefficients and a set of highpass coefficients through the iterated application of a filter bank.

**4.21****display process**

post-processing of the decoded image into a picture for display or further processing (informative)

**4.22****encoder**

embodiment of an encoding process and optionally an image unpacking process

**4.23****encoding process**

process that consumes an ordered set of component arrays to produce a bitstream

**4.24****entropy coding**

mathematical process intended to reduce the number of bits required for a lossless representation of data

**4.25****horizontal wavelet transform**

wavelet transform applied to each entire row in a two-dimensional array

**4.26****image**

ordered set of one or more component arrays

**4.27****image data**

component values that comprise an image

**4.28****image dimensions**

image width and height

**4.29****image height**

upper bound of the height of the component arrays represented in a bitstream (see Annex B.4)

#### 4.30

##### **image repacking process**

post-processing and repacking the decoded component arrays into a picture (informative)

#### 4.31

##### **image unpacking process**

unpacking and pre-processing a picture into an ordered set of component arrays for input to the encoding process (informative)

#### 4.32

##### **image width**

upper bound of the width of the component arrays represented in a bitstream (see Annex B.4)

#### 4.33

##### **initial codec state**

codec state at the beginning of the bitstream

#### 4.34

##### **inverse companding**

monotonic transform that restores companded values to an approximation of the quantized coefficient value before companding

#### 4.35

##### **inverse discrete wavelet transform (IDWT)**

inverse of the discrete wavelet transform

#### 4.36

##### **inverse horizontal wavelet transform**

inverse wavelet transform applied to each entire row of coefficients in an array

#### 4.37

##### **inverse quantization (dequantization)**

mapping a quantized value to a representative value for the sub-range identified by the quantized value

#### 4.38

##### **inverse spatial wavelet transform**

inverse vertical wavelet transform followed by application of the inverse horizontal wavelet transform to the result of the inverse vertical wavelet transform.

#### 4.39

##### **inverse vertical wavelet transform**

inverse wavelet transform applied to each entire column of coefficients in an array

#### 4.40

##### **inverse wavelet transform**

inverse of the wavelet transform (IDWT)

#### 4.41

##### **native order**

order of bytes on the encoding or decoding device

#### 4.42

##### **network order**

byte order that is big-endian

#### 4.43

##### **parameter**

variable in the codec state that may take one of a range of values

**4.44****picture**

displayable array of source, coded, or reconstructed image data with color values that may be packed or planar depending on the image format

**4.45****quantization**

process that partitions a range of discrete values into sub-ranges that are not necessarily equal and represents each partition by a unique value

**4.46****raster-scan order**

processing a two-dimensional array from the top line (row) to the bottom line (row) and from the left column to the right column within each line (row)

**4.47****reversible wavelet**

a discrete wavelet transform (DWT) and inverse discrete wavelet transform (IDWT) pair using elementary integer arithmetic operations such that the composition of the DWT and IDWT is the identity function

**4.48****run**

signed value and the number of times that the value is repeated

**4.49****spatial wavelet transform**

two-dimensional wavelet transform that is the separable composition of the horizontal wavelet transform followed by the vertical wavelet transform.

**4.50****spatial wavelet**

decomposition of an array into four wavelet bands by application of a spatial wavelet transform

**4.51****subband**

wavelet band that must be present in the bitstream

**4.52****subband coefficient**

value of a coefficient in a subband

**4.53****upper bound**

element in a range of values that is greater than or equal to the maximum element in a subset of the range

**4.54****variable length coding**

reversible procedure for entropy coding

**4.55****VLC**

variable length coding or the variable length code used for entropy coding

**4.56****vertical wavelet transform**

Wavelet transform applied to each entire column in a two-dimensional array

**4.57****wavelet band**

array of coefficients resulting from the application of the same combination of lowpass and highpass wavelet filters by the spatial wavelet transform

**4.58****wavelet level**

number of spatial wavelet transforms applied recursively to a component array to compute a spatial wavelet

**4.59****wavelet transform**

discrete wavelet transform (DWT)

**5 Notation****5.1 Codec State Parameters**

In this standard, parameters in the codec state are represented in camel caps and boldface with the first letter capitalized as in **ImageWidth**.

**5.2 Number Representations**

Numbers are decimal integers except that hexadecimal numbers are prefixed with “0x”.

The operators and functions used in this standard operate on and return numerical values with unlimited precision and perform internal calculations with unlimited precision.

Negative integers shall be represented using two’s complement arithmetic.

Integer division shall round towards zero.

**5.3 Arithmetic Right Shift**

The function  $\text{ash}(x, b)$  shall be an arithmetic right shift equivalent to the expression:

$$\left\lfloor \frac{x}{2^b} \right\rfloor$$

where  $x$  is a signed integer and  $b$  is a non-negative integer. Arithmetic right shift truncates towards negative infinity.

**6 Overview (Informative)**

The encoding process can be preceded by an image unpacking process that unpacks the input image into component arrays that are encoded in the bitstream as separate channels. A reversible wavelet transform is applied recursively to each component array to compute a sequence of three wavelets. The lowpass band for the smallest wavelet in the sequence and all of the highpass bands are encoded into the bitstream.

A decoding process decodes each channel separately into a component array and can deliver the component arrays to an image repacking process that packs the component arrays into an output image (Figure 1).

The image unpacking and repacking processes are not part of this standard.

Decoding the bitstream corresponding to each component array consists of the following steps:

- (1) The lowpass band in the bitstream is unpacked into the array for the lowpass band in the smallest wavelet in the sequence;
- (2) Each entropy coded highpass subband in the bitstream is decoded into the corresponding array for the wavelet band in the sequence;
- (3) The inverse companding curve is applied to the highpass band coefficients;
- (4) The decompanded highpass band coefficients are dequantized;
- (5) The inverse wavelet transform is applied recursively from the smallest to largest wavelet to reconstruct the lowpass bands that were not present in the bitstream;
- (6) The inverse wavelet transform is applied to the largest wavelet in the sequence to reconstruct the component array.

The highpass bands are entropy coded as runs of zeros and non-zero coefficients using a Huffman codebook.

Parameters that control decoding, such as the dimensions of each component array, are encoded in the bitstream as tag-value pairs to provide extensibility. The codec state is the current value of all parameters in effect at any point in the bitstream.

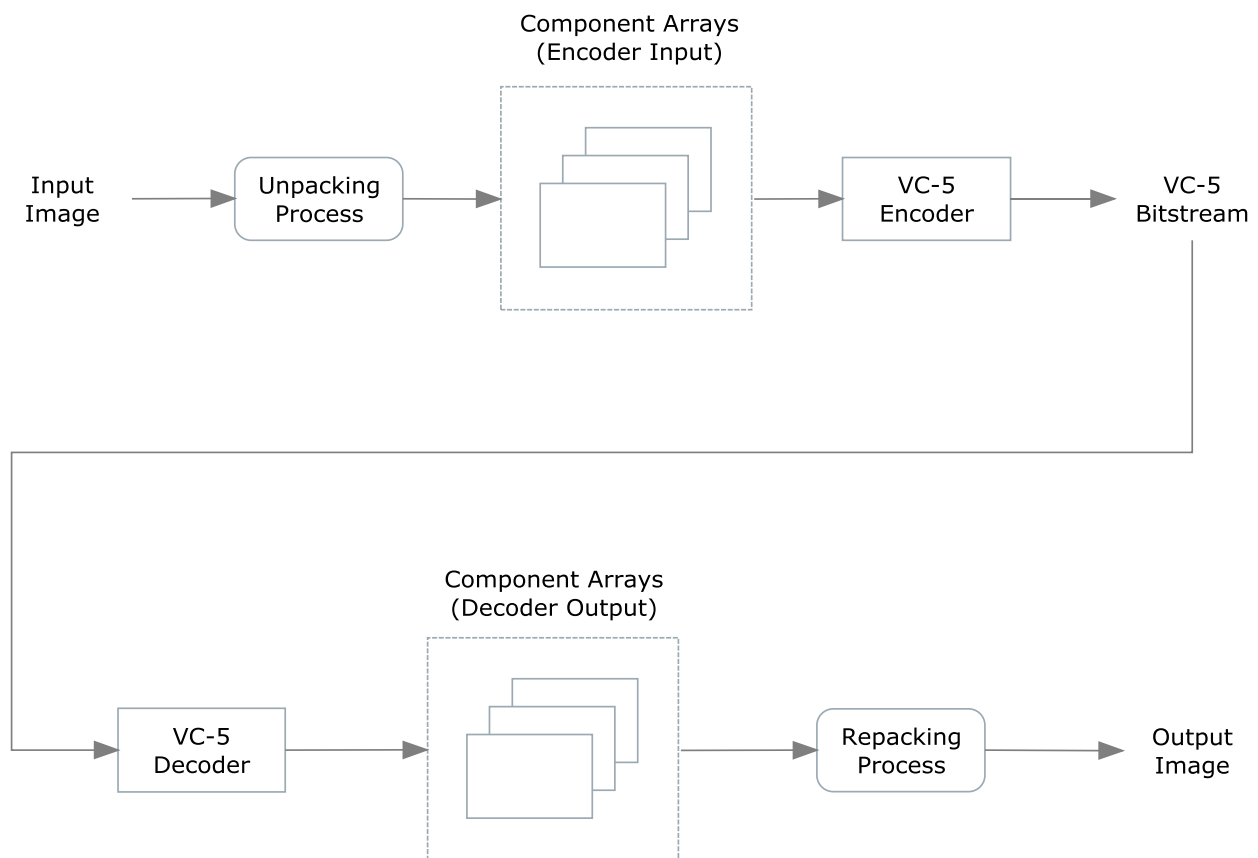


**Figure 1 – Overview of the encoding and decoding processes** (Informative)

## 7 Component Arrays

### 7.1 Component Array Requirements

The set of component arrays represented in the bitstream shall satisfy the following conditions:

- (1) All component values shall be unsigned integers;
- (2) The number of component arrays shall be the value of **ChannelCount**.

Note: In many applications, the component arrays are the output of an image unpacking process.

### 7.2 Component Array Representation

Each component array in the bitstream shall be represented as a separate channel in the bitstream (Figure 2). Channels shall be identified by channel numbers. Channels need not be present in the bitstream in numerical order.

### 7.3 Component Array Dimensions

**ChannelWidth** shall be the number of component values per row in a component array.

**ChannelHeight** shall be the number of rows in a component array.

### 7.4 Component Value Precision

**BitsPerComponent** shall be the number of significant bits per component value in a component array.

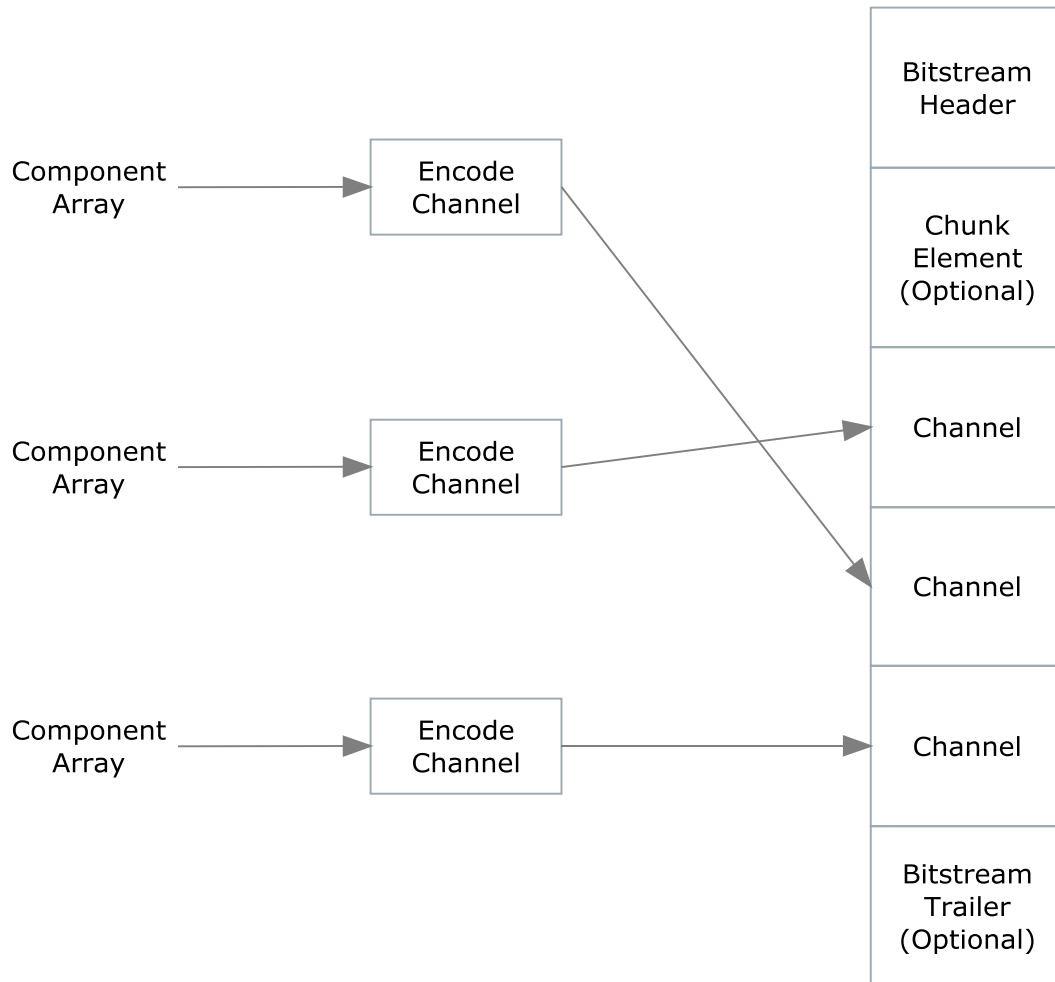


**Figure 2 – Component arrays are represented as separate channels in the bitstream**