ASCIIVideo (.ascv) File Format Specification

Version: v1.0

Author:
Vinetwigs

1 Introduction

The `ASCIIVideo (.ascv)` file format is designed to provide an efficient and standardized means of representing ASCII art[1] sequences in a video format. This specification outlines the structure of the `.ascv` file, the compression methods used, and the encoding techniques used to ensure that video data is stored in a compact, efficient manner.

This format is particularly suitable for applications where space efficiency is paramount, as it supports optional compression using **Run-Length Encoding (RLE)** and incorporates **Variable-Length Quantity (VLQ)** encoding to minimize storage requirements for video frame sizes.

The `.ascv` format is structured to allow both flexibility and extensibility, ensuring its utility across a wide range of systems, including embedded devices with limited storage capacity.

2 File Structure

A `.ascv` file consists of two major sections:

- **Header**: Contains essential metadata that defines the video properties.
- **Frame Data**: Contains the actual video frames, which are stored in a compact and efficient format.

2.1 Header Structure

The header section is located at the beginning of the file and is fixed in size, occupying exactly **32 bytes**. This section is composed of several fields, each of which serves to define key properties of the video, such as its dimensions, frame rate, number of frames, and compression settings. The fields of the header are as follows:

Field	**Type**	**Size**	**Description**
MAGIC	String	4 bytes	Magic number identifying the file (e.g., "AAVF")
VERSION	Integer	1 byte	Version number of the `.ascv` format
WIDTH	Integer	2 bytes	Width of the video in characters
HEIGHT	Integer	2 bytes	Height of the video in characters
FPS	Integer	1 byte	Frames per second (FPS)
FRAMES	Integer	4 bytes	Number of frames in the video
COMP	Integer	1 byte	Compression flag (0 = disabled, 1 = enabled)
CHARSET	Integer	1 byte	Character set identifier (0 = Standard ASCII)
RESERVED	Reserved	16 bytes	Reserved space for future extensions

2.2 Frame Data Structure

After the header, the frame data section contains the video frames, which are stored sequentially. Each frame consists of two main components:

- **Frame Size**: The size of the frame, encoded using the **Variable-Length Quantity (VLQ)** format. The frame size is encoded using a variable number of bytes, with smaller frames using fewer bytes.
- **Frame Content**: The content of the frame, consisting of ASCII characters representing the video frame. This content may be optionally compressed using **Run-Length Encoding (RLE)**.

2.3 Compression Methods

The `ASCIIVideo (.ascv)` format supports **Run-Length Encoding (RLE)**[2] compression, which is enabled if the `COMP` flag in the header is set to 1. RLE is a simple yet effective method of compressing sequences of repeated characters by storing each sequence as a pair of values:

- **Count**: The number of consecutive characters (1 to 255).
- **Character**: The repeated character.

For example:

- **Uncompressed Input**: AAAABBBCCDDDDDD
- **Compressed Output (RLE)**: 04A03B02C06D

2.4 Variable-Length Quantity (VLQ) Encoding

The `.ascv` format uses **Variable-Length Quantity (VLQ)** [3] to efficiently represent the frame sizes. VLQ encoding allows for the flexible representation of integers using a variable number of bytes. Smaller frame sizes use fewer bytes, reducing storage requirements for short frames.

- A size of 127 is encoded as a single byte: 01111111.
- A size of 300 is encoded using two bytes: 10010101 00101100.

This allows the `.ascv` format to handle video frames of varying sizes without excessive overhead.

3 Versioning

The versioning strategy for the `.ascv` format follows the semantic versioning principles. The version is expressed as `vX.Y`, where:

- `X` (the major version) is incremented when backward-incompatible changes are introduced to the file format, such as the addition of new header fields or changes to the way data is encoded.

- `Y` (the minor version) is incremented when non-breaking changes are introduced, such as the introduction of new optional features, optimizations, or clarifications in the specification.

Thus, for the initial version of the `.ascv` format, the version is `v1.0`. Future updates that introduce nonbreaking changes will increment the minor version (e.g. `v1.1`), while changes that affect backward compatibility will increment the major version (e.g., `v2.0`).

4 Example File

The following example illustrates a simple `.ascv` file that contains two frames, each with a width of 8 characters and a height of 4 characters. Compression is enabled (COMP = 1).

4.1 Header (32 bytes)

```
MAGIC:     AAVF
VERSION:   1
WIDTH:     8
HEIGHT:    4
FPS:       30
FRAMES:    2
COMP:      1
CHARSET:   0 (Standard ASCII)
RESERVED:  [reserved space]
```

4.2 Frame Data

Frame 0 (original content: “AAAABBBBCCCCDDDD“):

- **Frame Size**: 8 bytes (encoded as VLQ: “08”)
- **Frame Content (RLE)**: “04A04B04C04D“

Frame 1 (original content: “XXXXXXXXYYYYZZZZ”):

- **Frame Size**: 8 bytes (encoded as VLQ: “08“)
- **Frame Content (RLE)**: “08X04Y04Z“

4.3 Complete File Example

```
[AAVF][1][08][04][30][2][1][0][RESERVED] [08][04A04B04C04D] [08][08X04Y04Z]
```

5 Compatibility and Requirements

The `ASCIIVideo (.ascv)` format is designed to be lightweight and efficient, making it suitable for a wide range of applications, including those running on embedded systems with limited storage and processing power. It requires the following:

- **Hardware**: Any device capable of handling basic ASCII character sets and binary file formats.
- **Software**: A parser capable of handling variable-length encoding (VLQ) and Run-Length Encoding (RLE).
- **Operating System**: Cross-platform support with no specific OS requirements.

6 Conclusion

The `ASCIIVideo (.ascv)` file format provides a robust and space-efficient method for storing ASCII art video sequences. Using techniques such as Variable-Length Quantity (VLQ) encoding and Run-Length Encoding (RLE), it offers significant reductions in file size while ensuring that the format remains simple and extensible.

This specification is designed with both developers and end users in mind, providing a clear and comprehensive guide to working with the `.ascv` format while ensuring backward compatibility and flexibility for future versions.

References

[1] Wikipedia contributors. *ASCII art — Wikipedia, The Free Encyclopedia.* [Online; accessed 27-January-2025]. 2025. URL: `https://en.wikipedia.org/w/index.php?title=ASCII_art&oldid=1271222132`.

[2] Wikipedia contributors. *Run-length encoding — Wikipedia, The Free Encyclopedia.* [Online; accessed 27-January-2025]. 2025. URL: `https://en.wikipedia.org/w/index.php?title=Run-length_encoding&oldid=1270292084`.

[3] Wikipedia contributors. *Variable-length quantity — Wikipedia, The Free Encyclopedia.* [Online; accessed 27-January-2025]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=Variable-length_quantity&oldid=1255819394`.