

Specification of ASCIIVideo (.ascv) File Format
A file format to encode ASCII art sequences

v1.1.0

Abstract

The **ASCIIVideo** (**.ascv**) file format is proposed as an efficient method for encoding ASCII art sequences into a video format. This document outlines the specification of the **.ascv** format, including its structure, compression techniques, and encoding methods. The format is designed to offer high space efficiency while maintaining flexibility and extensibility across various systems, especially those with constrained resources. This document is intended for developers and implementers of software tools that work with the **.ascv** format.

1 Introduction

The ASCIIVideo (.ascv) file format provides an efficient mechanism for representing sequences of ASCII art [1] in video form. It is intended for applications that require high compression of graphical data, particularly for devices with limited storage capacity. The .ascv format uses several techniques to minimize storage requirements while preserving the integrity of the original video content.

This document provides a detailed specification of the .ascv format, including its structure, supported compression methods, and encoding techniques. Specifically, the format incorporates **differential frame compression** [2] and utilizes a **frame indexing system** [3] to enhance efficiency. These design decisions aim to reduce the file size of the video content, facilitating its use in environments with stringent storage constraints.

The specification is intended to assist developers in implementing parsers and encoders/decoders for the .ascv format, and it ensures backward compatibility while allowing for future extensibility.

2 File Structure

The .ascv file consists of two primary sections:

- **Header:** Contains essential metadata defining video properties such as dimensions, frame rate, and compression settings.
- **Frame Data:** Contains the actual video frames encoded in a compact, space-efficient manner.

2.1 Header Structure

The header is fixed in size, occupying **32 bytes** in total. It defines the core properties of the video, such as its dimensions, frame rate, and the total number of frames. Additionally, it contains flags for compression and encoding options. The header consists of the following fields:

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
COMPRESSION	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
INDEX OFFSET	Integer	8 bytes	Byte offset to the frame index within the file

The header ensures that the essential properties of the video are clearly defined, facilitating efficient parsing and processing by software implementations.

2.2 Frame Data Structure

Following the header, the frame data section contains the actual video frames. Each frame is encoded with two primary components:

- **Frame Size:** The total number of bytes used to store the frame data, encoded as a 4-byte integer.
- **Frame Content:** The actual frame data, which consists of ASCII characters. The content may be optionally compressed using differential compression between consecutive frames.

2.3 Compression Techniques

The `.ascv` format supports **Differential Frame Compression** [2] when the compression flag (COMP) is set to 1. This compression method minimizes redundancy by storing only the differences between consecutive frames. This is particularly useful for video sequences where many frames are similar to each other, allowing for significant storage savings.

- **Differential Compression:** The differences between a frame and the previous frame are stored, significantly reducing the amount of data required for similar frames.

For example, for two frames, Frame 0 and Frame 1:

- **Frame 0** (original content): “AAAABBBCCDDDD”
- **Frame 1 (diff from Frame 0):** “XXXXYYYYZZZZ”

Only the changed sections (“XXXXYYYYZZZZ”) are stored, reducing the overall file size compared to storing both frames in full.

2.4 Frame Indexing System

The `.ascv` format includes an **Index Offset** [3] field within the header. This field points to an index that lists the byte offsets of each frame within the file. This indexing mechanism allows for efficient random access to specific frames, which is beneficial for video playback and frame extraction.

The frame index is located after the video data and contains entries that specify the starting byte offset for each frame. The index facilitates efficient parsing and navigation within the file, particularly for applications requiring non-linear access to frame data.

2.5 Compression and Decompression with zlib

The `ASCIIVideo` (`.ascv`) format employs DEFLATE [4] compression to further optimize the storage of video frame data. When the `COMP` flag in the header is set to 1, the frame data is compressed using the DEFLATE compression algorithm. This method provides an effective means of reducing the size of the video frames, particularly for longer sequences of repetitive data.

DEFLATE algorithm offers both lossless compression and fast decompression, making it ideal for video formats where integrity and speed are crucial. The compression process involves the following steps:

- **Compression:** The frame data is first serialized into a byte stream and then passed through the DEFLATE compression function to generate a compressed byte sequence.
- **Decompression:** To retrieve the original frame data, the compressed byte sequence is passed through the DEFLATE decompression function, which restores the data to its original form for rendering or processing.

This compression process significantly reduces the storage requirements of the video frames, without compromising on data integrity. The `.ascv` format allows for easy integration of DEFLATE compression and decompression into existing software systems, providing flexibility for both encoding and playback.

3 Versioning

The versioning scheme used for the `.ascv` file format follows the Semantic Versioning convention with the format `1.X.Y`, where:

- **X** (major version) is incremented when backward-incompatible changes are introduced, such as changes to the file structure or compression algorithms.
- **Y** (minor version) is incremented for non-breaking changes, such as new features, optimizations, or clarifications to the specification.

The initial release of the `.ascv` format is `1.0.0`, with subsequent updates expected to follow the pattern `1.X.Y`.

4 Example File

This section provides a practical example of a simple `.ascv` file. The file 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]
INDEX OFFSET: [index offset]

4.2 Frame Data

Frame 0 (original content: “AAAABBBCCDDDD“):

- **Frame Size:** 16 bytes
- **Frame Content (Differential):** [No differential, as this is the first frame]

Frame 1 (original content: “XXXXXXXXXXYYYYZZZZ“):

- **Frame Size:** 16 bytes
- **Frame Content (Differential):** [Difference with Frame 0]

4.3 Complete File Example

[AAVF] [1] [08] [04] [30] [2] [1] [0] [RESERVED] [INDEX OFFSET]
[16] [AAAABBBCCDDDD]
[16] [XXXXXXXXXXYYYYZZZZ]

5 Compatibility and Requirements

The ASCIIVideo (.ascv) format is lightweight and optimized for environments with limited storage and processing capabilities. It requires the following:

- **Hardware:** Any device capable of handling basic ASCII character sets and binary file formats.
- **Software:** A parser capable of handling differential frame compression and binary file formats.
- **Operating System:** The format is cross-platform, with no specific OS requirements.

5.1 Extensibility and Future Features

The `.ascv` format is designed to be extensible, allowing for future features and improvements without breaking backward compatibility. The reserved bytes in the header provide a buffer for future field additions. These reserved spaces can be utilized in future versions of the format to introduce new features such as:

- Support for additional character sets or encoding schemes.
- Advanced compression techniques or hybrid algorithms combining differential and global compression.
- Storage of additional metadata.

6 Conclusion

The `ASCIIVideo (.ascv)` file format provides a highly efficient method for encoding ASCII art sequences in video form. By utilizing differential frame compression and a frame indexing system, it achieves significant reductions in file size while maintaining a flexible and extensible structure. This format is ideal for use in applications where storage and bandwidth are constrained, such as on embedded systems or low-bandwidth networks.

This document serves as a comprehensive guide to the `.ascv` format, offering developers a clear understanding of its structure, capabilities, and future extensibility. The `.ascv` format is designed to evolve with the needs of its users.

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] Talal Ali Hammouri Muzhir Al-Ani. *Video Compression Algorithm Based on Frame Difference Approaches*. https://www.researchgate.net/publication/269803555_Video_Compression_Algorithm_Based_on_Frame_Difference_Approaches. [Accessed 02-02-2025].
- [3] Wikipedia contributors. *Indexed file* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 2-February-2025]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Indexed_file&oldid=1242251239.
- [4] L. Peter Deutsch. *DEFLATE Compressed Data Format Specification version 1.3*. RFC 1951. May 1996. DOI: 10.17487/RFC1951. URL: <https://www.rfc-editor.org/info/rfc1951>.