



A Philips/PolyGram Corporation

AIM Technical Note #56

CARLIE Compressed AIM Run-Length Encoding

Richard Pferdner, Adaptive Design Co.

July 5, 1990

CARLIE is a technique for compressing run-length encoded images. The number of bytes required for a run-length encoded image is reduced by compressing the image in the vertical direction. CARLIE works best with cartoon-like images whose original size is under 20K.

Copyright © 1990 American Interactive Media.

All rights reserved.

This document is not to be duplicated or distributed without written permission from American Interactive Media.

CREDITS

Alty van Luijt of AIM designed the CARLIE algorithm. Rod Wood of Philips Interactive Media Systems implemented the CARLIE decoder, and Richard Pferdner of Adaptive Design Company implemented the CARLIE encoder.

CARLIE: Compressed AIM Run-Length Encoding

INTRODUCTION

The CD-I Full Functional Specification* (known as the Green Book) describes several image coding methods implemented in hardware on a CD-I player. The in-memory coding formats are specified for each of these coding methods. This opens the door to increasing image compression by augmenting these methods through software.

Run-Length Encoded images compress in the horizontal dimension. CARLIE uses vertical compression to further decrease the size of image data.

OVERVIEW

To use CARLIE, an image must first be encoded from run-length 7 (.RL7) format into CARLIE (.CAR) format. The encoded data can then be decoded at run time on the CD-I player.

The CARLIE decoder works by copying as many runs from the one scan line to the next. When copying is not possible, runs are inserted. Sometimes, runs are skipped over. The decoder is written in assembler language and has been optimized for speed and code size. CARLIE has the ability to modify the length of the individual runs while copying from the reference line to the line under construction, thus maximizing the opportunity for compression by copying.

CARLIE PRODUCTION PATHWAY

The production pathway for CARLIE-based images consists of:

- Encoding .RL7 images to .CAR using the program `tocarlie` on a SUN UNIX workstation.
- Decoding .CAR files back to .RL7 in the player at run-time.

Encoding is currently supported only on the SUN. The encoder `tocarlie` has two arguments, the source file and the destination file. The source file must be an IFF .RL7 format image. The destination file is written out as an IFF .CAR format image.

Decoding is performed by the routine called `carlieasm()`. This routine is written in 68000 assembler language. The decoder consists of three parameters: a buffer containing the CARLIE instructions, a buffer containing the source data, and a buffer containing the destination data. The size of the decoder program is 2 kilobytes.

* *CD-I Full Functional Specification, published by N.V. Philips and Sony Corporation, November 14, 1988.*

The same buffer should be used for the source data buffer and the destination data buffer. The decoder supports frame-to-frame compression, so there are separate parameters for source and destination buffers. The encoder, however, does not yet support frame-to-frame compression.

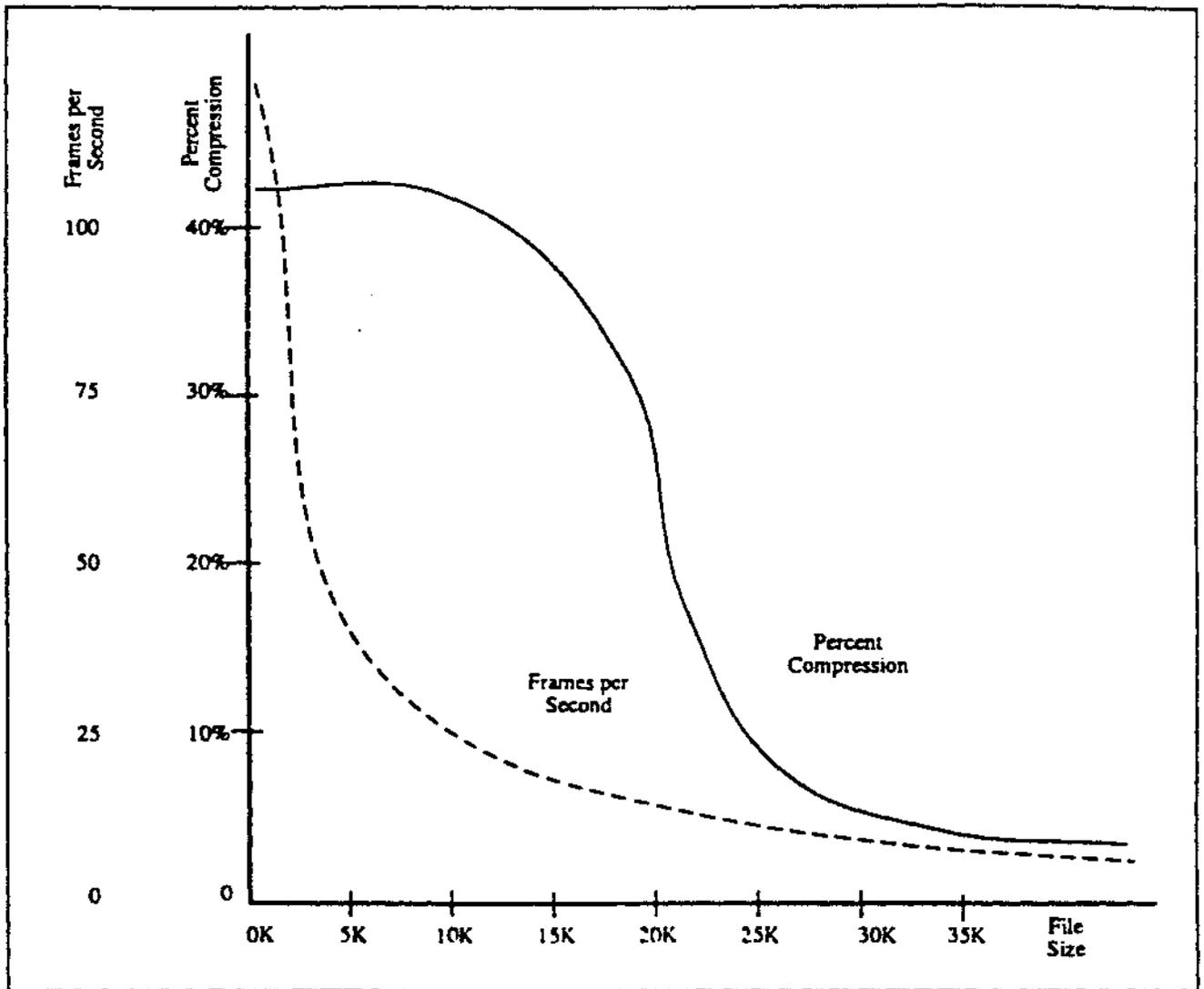
There is a sample program called `loadcarlie` that reads in .CAR files and displays them. In `loadcarlie`, there is code that reads past the IFF header and finds the CARLIE data. It also prepares the player to display the RL7 data, and makes a call to `carlieasm()` to decode the CARLIE data.

COMPRESSION AND RUN-TIME PERFORMANCE

The following images were encoded to show data compression performance, then decoded to show frame rate. During decoding, the CPU was used exclusively by the decoder.

Original File	Original Size	Compressed Size	Compression (% file reduced in size)	Decoding Speed in Frames per Second
cloud.rl7	1712	1008	41%	200
saints1.rl7	4500	2890	36%	66
saints2.rl7	5922	3866	35%	50
saints3.rl7	7530	4764	37%	37
saints4.rl7	9050	4978	45%	30
saints5.rl7	10510	5764	45%	25
saints6.rl7	13024	7902	39%	21
saints7.rl7	14218	8868	37%	18
saints8.rl7	21006	13712	35%	12
shell1.rl7	16148	11298	30%	21
shell2.rl7	22122	17794	20%	16
jetsons1.rl7	23892	22766	5%	14
jetsons2.rl7	26764	29178	8%	11
jetsons3.rl7	23804	22044	7%	13

The following graph shows the relationship of file size and percent of compression and file size and compression time in frame rate per second.



Percent of Compression and Rate for Encoding by File Size

Both compression and CPU loading is dependent on the image content. Note that the files `saints8.r17` and `jetsons1.r17` are almost the same size; however, the former compresses to a significantly smaller size. It should be noted that the `jetsons` images were digitized and contain a lot of noise. This reduces CARLIE compression.

To improve CARLIE compression performance, the following guidelines are offered:

- 1 A large sparse image compresses better than a small dense image.
- 2 Clean-up as much noise as possible from the image. Digitized images produce a lot of noise and compress poorly.
3. Pre-process images through a median filter.