Subject: : Amiga Classic
Topic: : Nightlong (FMV format/encoder?)
Re: Nightlong (FMV format/encoder?)
Author: : Raziel
Date: : 2021/9/22 23:12:08
URL:

@Raziel

Just so that it won't get lost in the depth of the interwebs

https://codecs.multimedia.cx/2021/07/f ... ut-h4m-and-nightlong-fmv/
Quote:

> Final news about H4M and Nightlong FMV
> In my previous post I said I was looking at them and while some details were unknown, I got some understanding.
>
> But it turns out The Multimedia Mike despite what he claims still hasn’t lost the knack for REing codecs—first you need to look if somebody else has done it already. And indeed, somebody else has spent two years on reverse engineering the codec. So all that was left to me was to improve its description it (which I did).
>
> The video compression there is quite curious. Huffman-coded data chunks are not that rare (Truemotion 2 had them, for example). Huffman-coded data chunks where one chunk has tree description coded in the beginning that is used for several other chunks is more interesting. But the transforms was the juiciest part there. It turns out you have several modes of operation: fill block, raw block, smoothed DC block (that one uses DCs from neighbour blocks on all sides to create a smooth transition between them all) and the actual Adaptive Orthogonal Transform that represents block as 1-5 patterns (scaled), selected from single matrix. It reminds me a bit about SVQ1 where you had a block constructed from 2D patterns of varying size. But even more it reminds me of Matching Pursuit experimental video codec from Berkley University that also performed partial transform and selected only some bases from a large set (does anybody remember this codec at all?).
>
> Nightlong FMV is a bit different. The format is simple but FMV2 has packed frames and there was no code for handling it in 68k version of the binary. Nevertheless I could unpack it simply by dumping frame data and extracting it with unar (PowerPacker is a strange compression format BTW, it starts reading data from the end of buffer and outputs data starting from the end as well). The unpacked data turned out to use the same format except that the frame size is now full 640×360 instead of 320×180 trying to look like twice as large. I documented it as well. (see below)

and
https://wiki.multimedia.cx/index.php?title=Nightlong_FMV

Quote:

Nightlong FMV

This is the video format used by Amiga version of Nightlong: Union City Conspiracy game.

Video comes in two flavours: 320x180 (with .fmv extension) and 640x360 (with .fmv2 extension). Frame is split into tiles (4x4 for 320x180 resolution, 8x8 for 640x360) and each tile can be either skipped or coded/updated with raw pixels.

Container format
FMV/FMV2 starts with 32-bit little-endian number telling the number of frames and a table of 32-bit numbers telling the frame sizes.

Frame data
Frame data for FMV is stored unpacked and FMV2 frames are packed with PowerPacker and can be distinguished by PP20 magic word at the start of the frame (it cannot happen in the ordinary frame data and thus can be used to detect FMV2 by content).

Frame data consists of several opcodes:

0x80 (should be the first one if present and there's at most one such code) - palette update. Next byte is palette entries count minus one followed by the actual palette triplets. Only low 5 bits are used;
0x20 signals this is a skip frame and no image data is coded;
0x21 signals a row skip. The following byte tells how many lines starting from the current one should be skipped (including the current row). For either variant of FMV this value assumes 8-pixel rows (so for 320x180 video it should be halved);
0x00 signals that current tile should be skipped;
0x42 signals a raw tile with the following 16 or 64 bytes being raw pixels
0x43 signals a sub-divided and partially updated tile. Each quarter starts either with 0x42 opcode signalling it is a raw sub-block and the following 4 or 16 pixels should be read from the stream, or the opcode is a combination of flags telling which quarter of sub-block (1 or 4 pixels) should be updated.