Supporting niche formats and hardware in open source software and operating systems

Kieran Kunhya <kieran@kunhya.com> - @kierank_

Who am I?

- Work on Open Source Multimedia in both a personal and professional capacity: FFmpeg and x264 in particular.
- Also interested in reverse engineering of codecs and hardware
- Presentation in reverse chronological order for ease of understanding
- This presentation given in a personal capacity

Why FFmpeg?

- De-facto open source multimedia processing tool and library
 - Many web browsers implement complex specifications like HTML and CSS
 - Nothing comparable in multimedia
 - Basis of video players such as VLC, browsers (Chrome + Firefox), Smart TVs etc
- Written in C, not in newfangled language of the week
 - Widely supported across computer architectures
 - Likely tens of decades or more of support
 - Works on RPI, watch, other weird device



Standing on the Shoulders of Giants

- Kostya Shishkov and Paul Mahol
- Master codec reverse engineers
 - Willingness to teach and help (in their own special way...)
- Both reverse engineered dozens of codecs
 - Monumental impact on media playback

Why support niche formats

- Someday, someone might discover and want to play these formats.
 - An impact now or long into the future
 - Most of you in the room are that impact
- Quite a lot of multimedia low-hanging fruit is "done", good way to stay familiar with concepts (entropy coding, DCT).
 - Something you can do over Easter/Christmas.
 - Also work for new students

This ticket

- There are so many variants of MPEG-4, why are these ones not supported?
- Why are they 4:2:2 and 4:4:4?
- Only 3 samples available, to this day no real-world samples found.

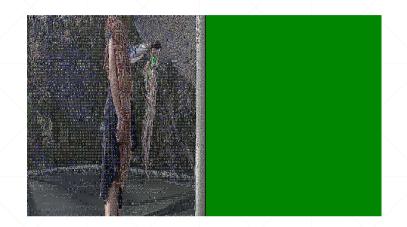
#4447 closed e	Infancement (Iixeu	9			Closed 7 months ago
Support MPEG-	4 Studio Profile				
Reported by:	cehoyos	Owr	ned by:		
Priority:	wish	Con	nponent:	avcodec	
Version:	git-master	Key	words:	sstp	
Cc:	kierank	Bloc	ked By:		
Blocking:			roduced by eloper:	no	
Analyzed by developer:	no				
Description ⇒http://ffmpeg.org/					
¹	MPEG-4 Simple Stu			ported by FFm	ipeg.

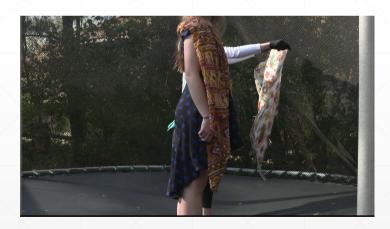
MPEG-4 Sstp (1)

- Spent a lot of time trawling the internet looking for recent copy of paid spec
 - Common in industry but a big problem for independent developers.
 - Eventually found it on some Chinese website
- Woah this is different
 - Up to 12-bit data, 4:2:2, 4:4:4, RGB none supported in existing MPEG-4 code
 - > 16-bit coefficients (not supported in FFmpeg IDCT)
 - Crazy DPCM mode with vertical block scanning
- Only one program on Windows, no ability to extract raw data

MPEG-4 Sstp (2)

- Entropy coding nice, either works or it doesn't
- Used float IDCT to get to working picture
- No reference decoder, tweaked picture until it looked ok.
- Took another year to get round to boring work of implementing templated integer IDCT.
- Also tedious work to hack-in 10/12-bit into FFmpeg, not ideal method.





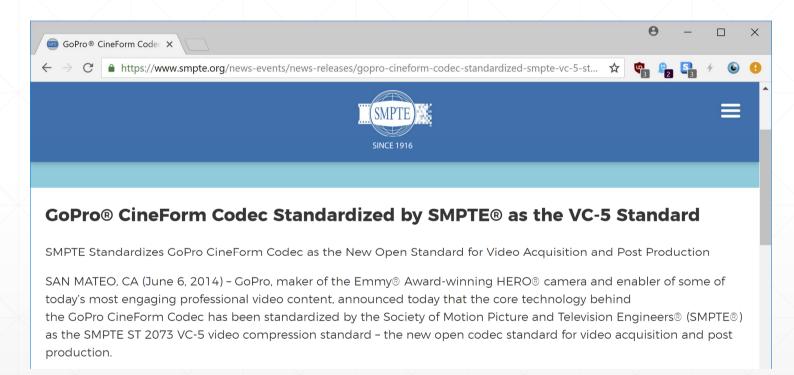
MPEG-4 Sstp (3)

- Was quite hard to verify DPCM blocks
- After a lot of digging online found conformance MPEG bitstreams, weird raw file format.
- Got to position where nearly all of image worked but some part looked weird.
- After even more digging found reference software. Typo found: -(x >> n) != ((-x) >> n)
- Might be some minor errors, probably IDCT precision problem, maybe > 32-bit intermediates?





This press release



Cineform

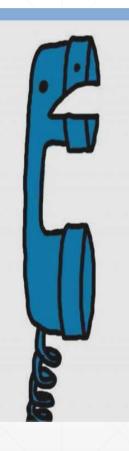
- Downloaded SMPTE VC-5 specification
 - Some things similar but some things in real world completely different
- Hints that lowpass coefficients were raw, by luck one sample had quite a lot of flat colours.
 - Helped identify region
- Lucky also this sample used the published codebook, though could have reverse engineered from binary decoder

3	/home/kieran/ffmpeg/trailer.	ave Oliess	- 8
le Edit View Search Tools Help			
🌡 🚞 💽 I (Marice) 🕌 📓	1 🛍 🤉 🦗		
ailer.avi ×			
022f36 2F 78 00 04 0F 0F	35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 2 25 35 2C 35 25 35 2C 35 2C 35 25 35 2C 35 25 35 2C		
	35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C		
022f93 25 35 2C 35 25 35			, 5%5, 5%5, 5%5, 5%5, 5%5, 5%5,
022fb2 35 25 35 2C 35 25 022fd1 2C 35 25 35 2C 35			15, 585, 585, 585, 585, 585, 585, 585 85, 585, 5
022f0f 35 2C 35 25 35 2C	35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 :	2C 35 25 35 2C 35 25 35 2C 35 5,5%5,	5%5, 5%5, 5%5, 5%5, 5%5, 5%5, 5%
	2c 35 25 35 2c 35 25 35 2c 35 25 35 2c 35 25 35 2c 35 25		, 5%5, 5%5, 5%5, 5%5, 5%5, 5%5,
02302e 35 25 35 2C 35 25 02304d 2C 35 25 35 2C 35	35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C 35 2 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 2C		15, 585, 585, 585, 585, 585, 585 185, 585, 585, 585, 585, 585, 58
02306c 35 2C 35 25 35 2C	35 25 35 2C 35 25 35 2C 35 25 35 2C 35 25 35 :		5%5, 5%5, 5%5, 5%5, 5%5, 5%5, 5
02308b 25 35 2C 35 25 35 0230aa 35 26 35 27 35 26			5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
0230c9 27 35 26 35 27 35			65'565'565'565'565'565'56
0230e8 35 27 35 26 35 27		27 35 26 35 27 35 26 35 27 35 5'565' 35 27 35 26 35 27 35 26 35 27 45'56	545'545'545'545'545'545'545'
023107 26 35 27 35 26 35 023126 35 26 35 27 35 26			
023145 27 35 26 35 27 35	26 35 27 35 26 35 27 35 26 35 27 35 26 35 27	35 26 35 27 35 26 35 27 35 26 '565'5	45'545'545'545'545'545'54
023164 35 27 35 26 35 27	35 26 35 27 35 26 35 27 35 26 35 27 35 26 35	27 35 26 35 27 35 26 35 27 35 5 5 5 5	545'545'545'545'545'545'5
irch for: 0004010		as Hexadecimal	♥ Find Next ♠ Find Previou
Signed 8 bit: 53	Signed 32 bit: 624241717	Hexadecimal: 35 2C 35 25	1
Unsigned 8 bit: 53	Unsigned 32 bit: 624241717	Decimal: 053 044 053 037	1
Signed 16 bit: 11317	Float 32 bit: 1.571423E-16	Octal: 065 054 065 045	1
nsigned 16 bit: 11317	Float 64 bit: 1.90905230962663E-129	Binary: 00110101 0010110	0 00110101 00100101
Show little endian dece	oding 🗌 Show unsigned as hexadecim	nal ASCII Text: 5,5%	
		Offset: 0x22f3c / 0x31fffff Selection	on: None INS
	PYUV - out.yuv	- • ×	
	<u>File Control View Set</u>	ttings <u>H</u> elp	
	S. 20 C.		
	100 C		
	2 Contraction of the second seco		
	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		
	Alexandren and a second se		
	Ara .		

Welcome to PYUV! 55 frames, 0'2", 6.79

Cineform (2)

- Continue reverse engineering tags, lots of people contributed samples.
 - Aligned coefficient layout
 - Simple and fast codec
 - Eventually get to working decoder
- Some samples had more complicated structures, "3Dtransform" frames, interlaced, bayer.



Cineform (3)

- <u>https://medium.com/@kierank_/reverse-engineering-the-gopro-cineform-codec-7411312bfe1c</u> 36k readers!
- Eventually led to Cineform being open-sourced by GoPro!
- Few missing pieces implemented by Google Summer of Code 2018 student
 - Not integrated into FFmpeg yet



CineForm Introduction

GoPro CineForm® is a 12-bit, full-frame wavelet compression video codec. It is designed for speed and quality, at the expense of a very high compression size. Image compression is a balance of size, speed and quality, and you can only choose two. CineForm was the first of its type to focus on speed, while supporting higher bit depths for image quality. More recent examples would be Avid DNxHD® and Apple ProRES®, although both divide the image into blocks using DCT. The full frame wavelet as a subject quality advantage over DCTs, so you can compression more without classic ringing or block artifact issues.

Pixel formats supported:

- 8/10/16-bit YUV 4:2:2 compressed as 10-bit, progressive or interlace
- 8/10/16-bit RGB 4:4:4 compressed at 12-bit progressive

But what about Physical Media files come on

- One day someone will come across these and wonder how to read files
- Panasonic P2 and Sony SxS cards
 - PCI and PCI Express-based respectively
 - Both common and widely used, precursors to modern NVMe storage
- Closed source Windows and Mac drivers, or proprietary readers.
 - Likely won't work in 100 years
 - Linux driver will last, nothing comparable hardware support

Panasonic P2 (1)

- PCMCIA based solid stage storage introduced in 2004
- Modern PCI Express backwards compatible with legacy PCI
- Bought cheap card off eBay.



Panasonic P2 (2)

- Request source code for Linux on camera
 - Dig through thousands of lines and find ancient Linux 2.4 driver.
- Go to parents house, find my high school PC from 2008 with legacy PCI slot.
 - Will it work without official hardware?
- A lot of messing with CentOS and wow, it worked!
- Might have newer cameras with more reasonable driver:
 - https://github.com/kierank/p2card

total 34882		1- mar	States of	Contraction of the second
dr-xr-xr-x	2 root roo	t 3276	8 Sen 21	2006
dr-xr-xr-x	8 root roo	t 3276	8 Sen 17	2006
-r-xr-xr-x	1 root roo	t 33180899	2 Sen 17	2006 00109
-r-xr-xr-x	1 root roo	t 31510499	2 Sen 17	2006 0014wl.mxf
-r-xr-xr-x	1 root roo	t 30531299	2 Sen 17	2006 00157h.mxf
-r-xr-xr-x	1 root roo	t 37788899	2 San 17	2006 0017gb.mxf
-r-xr-xr-x	1 root roo	t 4899299	2 Sep 17	2006 0017gb.mxf
r-xr-xr-x	1 root roo	+ 43894499	2 Sep 17	
r-xr-xr-x	1 root roo	+ 44520000	2 Sep 17	2006 002121.mxf
r-xr-xr-x	1 root roo	+ 59792000	2 Sep 17	2006 0023h3.mxf 2006 0024vz.mxf
P-YP-YP-Y	1 root roo	+ 16922400	2 Sep 11	2006 0025v7.mxf
n-vn-vn-v	1 mont mon	+ 6604000	2 Sep 21	2006 002507.mxf
	1 root roo	L 0001077	2 Sep 21	2006 0026wc.mxf
T-XF-XF-X	1 root roo	t (145099	2 Sep 21	2006 00276q.mxf
r-xr-xr-x	1 root roo	£ 18083633	2 Sep 21	2006 0028nw.mxf
				2006 0029b7.mxf
			2 Sep 21	2006 00301c.mxf
root@local	host video]# df -h		
ilesystem		Size Use	d Avail I	Jse% Mounted on
dev/mapper	/VolGroup0	0-LogVo100		
and the second second		2 56 1 7	6 1 76	50% /
dev/sda1		99M 19	M 76M	20% /boot
mpfs		1.56	0 1.5G	0% /dev/shm
dev/snda1		3.66 3.6		99% /mnt
	host video			

Sony SxS

- Also requested source code, no driver to be found ☺
- Instead looked into reverse engineering.
- Built rig to easily use card.
- Setup Windows XP QEMU to sniff memory mapped IO, how the driver is reading memory from the card.
 - https://hakzsam.wordpress.com/ 2015/02/21/471/



Sony SxS (2)

- At the time needed newest CPU with IOMMU
- Use "dd" to read block by block and see how the driver reads and writes. Change the block index, and continue
 - Very basic by modern NVMe standards
 - Request block, get an interrupt with buffer
- https://github.com/kierank/sxs-linux
 - No working DMA yet so very slow
 - Need time to get it into mainline Linux (Christmas?)

Conclusions

- Audio and Video codec reverse engineering is tending to completion
 - Codecs are being reverse engineered at a faster rate than they are created
 - This has remarkable historical consequences for media
 - Both the fact that users can "store" their content on sites like YouTube
 - Or that discovered media will be playable for decades
- That said there are still many proprietary storage formats out there
 - May not be easily possible to use commodity reader and reverse engineer