

# Migrating ProRes/MOV to FFV1/MKV

Kieran O'Leary - IFI Irish Film Archive

Budapest 2019-12-05

@kieranjol #nttw4



**Irish Film** Institute

# Why normalise lossy video to lossless?

- FFV1/MKV - Free/Open
- May already be designated preservation master format
- Move away from proprietary formats
- Safeguard your collections?????

# Dangers

- What if 'significant properties' are lost?
  - Also what even are those significant properties?
- How to validate correct normalisation
- Is Framemd5 enough?
- File size increase (depends on input, but we've seen 35-40% increase)

# Testing

- Raelene Casey, Gavin Martin and Kieran O'Leary
- A few hundred SD ProRes/PCM/MOV/tmcd files
- Normalise to FFV1/MKV
- Do Framemd5s match? - YES!
- Lossless? - Yes!

D6D471268	Clean Aperture (40 bytes)	
D6D471268	Header (8 bytes)	
D6D471268	Size:	40 (0x00000028)
D6D47126C	Name:	clap
D6D471270	apertureWidth_N:	41472 (0x0000A200)
D6D471274	apertureWidth_D:	59 (0x0000003B)
D6D471278	apertureHeight_N:	576 (0x00000240)
D6D47127C	apertureHeight_D:	1 (0x00000001)
D6D471280	horizOff_N:	0 (0x00000000)
D6D471284	horizOff_D:	1 (0x00000001)
D6D471288	vertOff_N:	0 (0x00000000)
D6D47128C	vertOff_D:	1 (0x00000001)
D6D471290	Pixel Aspect Ratio (16 bytes)	
D6D471290	Header (8 bytes)	
D6D471290	Size:	16 (0x00000010)
D6D471294	Name:	pasp
D6D471298	hSpacing:	59 (0x0000003B)
D6D47129C	vSpacing:	54 (0x00000036)

$$41472/59 * 59/54 = 768$$

$$768/576 = 1.3333333333 :)$$





# Aspect ratio/Crop

- Some MOV files contained clean aperture metadata (clap)
- Crop defined which displays correctly in QuickTime Player
- Clap metadata not mapped to Matroska PixelWidth.. But it can!
- Financial support needed to add this support
- By default, ffmpeg applies 59/54 PAR, ignoring the crop
  - $720 * (59/54) = 786.7777777$
  - $786.777 / 576 = 1.364:1$
  - Different aspect ratio 1.33333 vs 1.364:1

← Thread



**Kieran O'Leary**  
@kieranjol

btw @BMahol - would this be something that you'd be interested in working on if I could find funding?  
Specifically the @ffmpeg clap atom reading in MOV and the migration of that data possibly to Matroska (if this was actually an appropriate thing to do)

12:13 PM · Jun 10, 2019 · [Twitter Web Client](#)

||| View Tweet activity

1 Like



**Paul B Mahol** @BMahol · Jun 10

Replying to @kieranjol and @FFmpeg

I doubt matroska have proper raw video support. But I may be wrong if something recently changed.

2



⋯⋯⋯ @RetoKromer · Jun 10

You can use Matroska, but it's not the easiest choice.



← Tweet

 **Paul B Mahol** @BMahol · Jun 10

Agree, but I still miss important information, like what is currently missing.

1 1 1 1

 **Kieran O'Leary** @kieranjol · Jun 10

Ah,I can email tomorrow with details and links to the relevant trac tickets!

1 1 1 1 1

 **Paul B Mahol** @BMahol · Jun 15

I never received any mail.

1 1 1 1

 **Kieran O'Leary** @kieranjol · Jun 17

Will send something today all going well,sorry Paul.

1 1 1 1 1

 **Paul B Mahol** @BMahol

Replying to @kieranjol @dericed and @FFmpeg

I'm forgotten! :(

7:45 PM · Jul 2, 2019 · [Twitter Web App](#)

1 1 1 1

 **Kieran O'Leary** @kieranjol · Jul 2

Replying to @BMahol @dericed and @FFmpeg

Nope,just on a never ending todo list, sorry! Will get to this as soon as I can.

1 1 1 1 1

**#4489** [new enhancement](#)

Opened [5 years ago](#)  
Last modified [12 months ago](#)

## mkv cropping

Reported by:	<a href="#">cehoyos</a>	Owned by:	
Priority:	<a href="#">wish</a>	Component:	<a href="#">undetermined</a>
Version:	<a href="#">git-master</a>	Keywords:	<a href="#">videolan</a> <a href="#">mkv cropping</a>
Cc:	<a href="mailto:nfxjfg@googlemail.com">nfxjfg@googlemail.com</a> , <a href="mailto:dave@dericed.com">dave@dericed.com</a>	Blocked By:	
Blocking:		Reproduced by developer:	<a href="#">no</a>
Analyzed by developer:	<a href="#">no</a>		

### Description

([videolan ticket 13982](#))

[← Reply](#)

I will attach a sample that has cropping defined in its mkv headers.

```
$ ffmpeg -i Matroska\Crop_cut.mkv
ffmpeg version N-71483-g8768f8f Copyright (c) 2000-2015 the FFmpeg developers
built with gcc 4.7 (SUSE Linux)
configuration: --enable-gpl
libavutil      54. 22.101 / 54. 22.101
libavcodec     56. 34.100 / 56. 34.100
libavformat    56. 30.100 / 56. 30.100
libavdevice    56.  4.100 / 56.  4.100
libavfilter     5. 14.100 /  5. 14.100
libswscale     3.  1.101 /  3.  1.101
libswresample  1.  1.100 /  1.  1.100
libpostproc   53.  3.100 / 53.  3.100
Input #0, matroska,webm, from 'Matroska Crop_cut.mkv':
Metadata:
  encoder       : libebml v1.2.2 + libmatroska v1.3.0
  creation_time : 2012-08-03 22:48:21
Duration: 00:03:42.12, start: 0.000000, bitrate: 92 kb/s
Stream #0:0(eng): Audio: aac (LC), 44100 Hz, stereo, fltp (default)
```

#7437 new enhancement

Opened 15 months ago

Last modified 2 months ago

## clap atom values ignored by ffmpeg

Reported by:	kieranjol	Owned by:	
Priority:	wish	Component:	avformat
Version:	git-master	Keywords:	mov cropping
Cc:		Blocked By:	
Blocking:		Reproduced by developer:	no
Analyzed by developer:	no		

### Description

#### Summary of the bug:

[← Reply](#)

Sample file - <https://we.tl/t-KoBK9MnSLu>

My MOV file contains a clap atom that defines the clean aperture width as 703/576, but an actual width of 720/576. QuickTime? player will perform a crop and only display the 703/576 area. When remuxing to Matroska or other containers, I would have assumed that the expected behaviour would involve this cropping information being written to the remuxed file, probably using the various PixelCrop? values within Matroska ( <https://matroska.org/technical/specs/index.html> ) Specifically, this value is stored in the QuickTime? file using numerators and demoninators as such:

```
$ mediainfo --Details=1 clap.mov | grep clap -n10
284-ECE5C05C      Clean Aperture (40 bytes)
285-ECE5C05C      Header (8 bytes)
286-ECE5C05C      Size:                40 (0x00000028)
287-ECE5C060      Name:                clap
288-ECE5C064      apertureWidth_N:    41472 (0x0000A200)
289-ECE5C068      apertureWidth_D:    59 (0x0000003B)
290-ECE5C06C      apertureHeight_N:   576 (0x00000240)
291-ECE5C070      apertureHeight_D:   1 (0x00000001)
292-ECE5C074      horizOff_N:         0 (0x00000000)
293-ECE5C078      horizOff_D:         1 (0x00000001)
294-ECE5C07C      vertOff_N:          0 (0x00000000)
295-ECE5C080      vertOff_D:          1 (0x00000001)
```

It is somewhat related to this issue <https://trac.ffmpeg.org/ticket/1485>

#### How to reproduce:

Using my sample file - I would expect the following command to include the cropping information but it does not. ffprobe does not seem to read this cropping info, so mediainfo is useful to check the clap values. My sample was generated directly from a VHS capture using the Blackmagic Intensity Shuttle, with lots of dropped frames which doesn't relate to this issue

**#21192** [new defect](#)

Opened [15 months ago](#)

Last modified [12 months ago](#)

## Support pixelcrop values in Matroska

Reported by:	<a href="#">Kieran O Leary</a>	Owned by:	<a href="#">Steve Lhomme</a>
Priority:	<a href="#">normal</a>	Milestone:	<a href="#">Bugs paradize</a>
Component:	<a href="#">Demuxers: MKV</a>	Version:	<a href="#">3.0.x</a>
Severity:	<a href="#">normal</a>	Keywords:	<a href="#">pixelcrop</a> , <a href="#">matroska</a>
Cc:		Difficulty:	<a href="#">unknown</a>
Platform(s):	<a href="#">all</a>	Work status:	<a href="#">Not started</a>

### Description

It appears that VLC (version 3.0.4 on OSX was tested) ignores pixelcrop values. An examples can be downloaded here and I will try to attach. [⇒ https://archive.org/details/10sec](https://archive.org/details/10sec)

This video file was created with ffmpeg v4.0.2 using:

```
$ ffmpeg -f lavfi -i testsrc=size=720x576 -t 10 -pix_fmt yuv420p 10sec.mkv
```

PixelCrop? values were then added using mkvpropedit:

```
$ mkvpropedit --edit track:v1 --set pixel-crop-left=100 --set pixel-crop-right=234 10sec.mkv
```

which produces these values via mediainfo:

```
Width                : 386 pixels
Original width       : 720 pixels
Height               : 576 pixels
```

but no crop is present in vlc 3.0.4 on oSX.

**#21486** new defect

Opened [12 months ago](#)

Last modified [12 months ago](#)

## Turn PixelCrop/Clean Aperture cropping on or off for Matroska/MOV

Reported by:	<a href="#">Kieran O Leary</a>	Owned by:	<a href="#">Steve Lhomme</a>
Priority:	<a href="#">normal</a>	Milestone:	<a href="#">Bugs paradize</a>
Component:	<a href="#">Demuxers: MKV</a>	Version:	<a href="#">master git</a>
Severity:	<a href="#">normal</a>	Keywords:	
Cc:		Difficulty:	<a href="#">unknown</a>
Platform(s):	<a href="#">all</a>	Work status:	<a href="#">Not started</a>

Description (last modified by [Jean-Baptiste Kempf](#)) [Δ](#)

I'm making this ticket based on this comment by Steve Lhomme: [↗](#)  
<https://trac.videolan.org/vlc/ticket/21192#comment:4>

As per [#21192](#) and [#21179](#) VLC does not currently support the cropping values specified in Clean Aperture for MOV/MPEG-4 and PixelCrop? in Matroska.

When these are implemented, it would be ideal if there was an option to turn this cropping on or off, as is possible in QuickTime? Player.

**#21179** new enhancement

Opened 15 months ago

Last modified 14 months ago

## Support clean aperture/clap cropping in QuickTime files

Reported by:	<u>Kieran O Leary</u>	Owned by:	<u>Francois Cartegnie</u>
Priority:	<u>normal</u>	Milestone:	<u>Features paradize</u>
Component:	<u>Demuxers: MP4</u>	Version:	<u>master git</u>
Severity:	<u>normal</u>	Keywords:	<u>clap, clean aperture, quicktime, aspect ratio</u>
Cc:	<u>InTheWings</u>	Difficulty:	<u>unknown</u>
Platform(s):	<u>all</u>	Work status:	<u>Not started</u>

### Description

VLC ignores clean aperture widths and heights in QuickTime? files. QuickTime? Player reads these values and crops the video accordingly. For example this file: <https://we.tl/t-KoBK9MnSLu> Has 720/576 stored pixels, but there's a clean aperture width of 703 located in the clap atom. So the image gets cropped - which is usually preferable in SD PAL video which can have black padding in that area. Can VLC support these values, and possibly some sort of option to toggle the functionality on and off, in case a user wishes to display all the pixels?

# IFI outcome

- We preserve ProRes within MOV for now
- ProRes is capable of self-description in ways FFV1 is not
- VLC needs improvements for handling MOV/MKV
- Most likely not an appropriate migration YET - Could change in the future
- Report display issues in MOV/MKV in VLC and hopefully financially support a fix
- Report clean aperture metadata mapping between MOV and Matroska
- Focus more on improving the decoding and display tools rather than changing formats
- This is not a failed exercise, it's a process of continual improvement
- Lesson: Report issues and fund improvements!