GStreamer

Image conversion

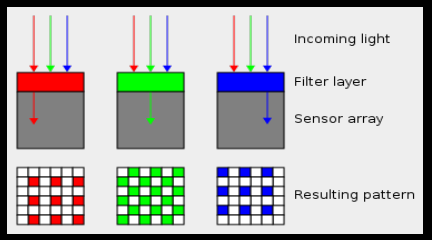
Wim Taymans

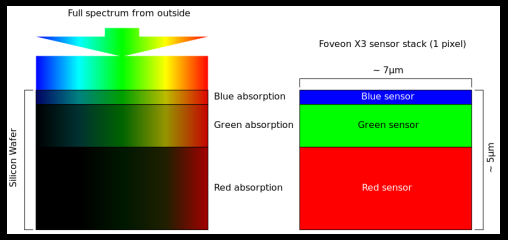
16 oct 2014 – GStreamer Conference
Düsseldorf, Germany

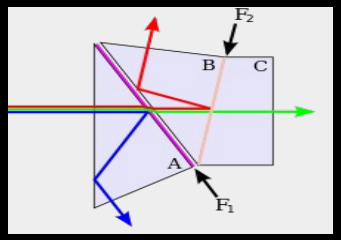
redhat.

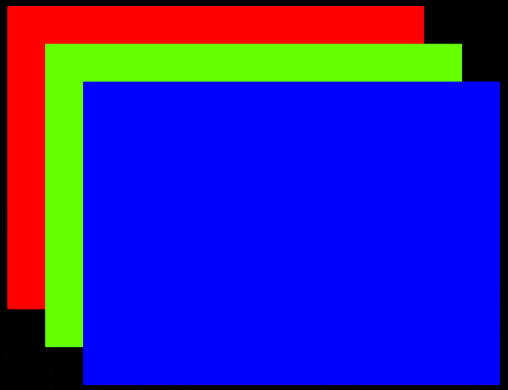
The life of a video frame

gstreamer

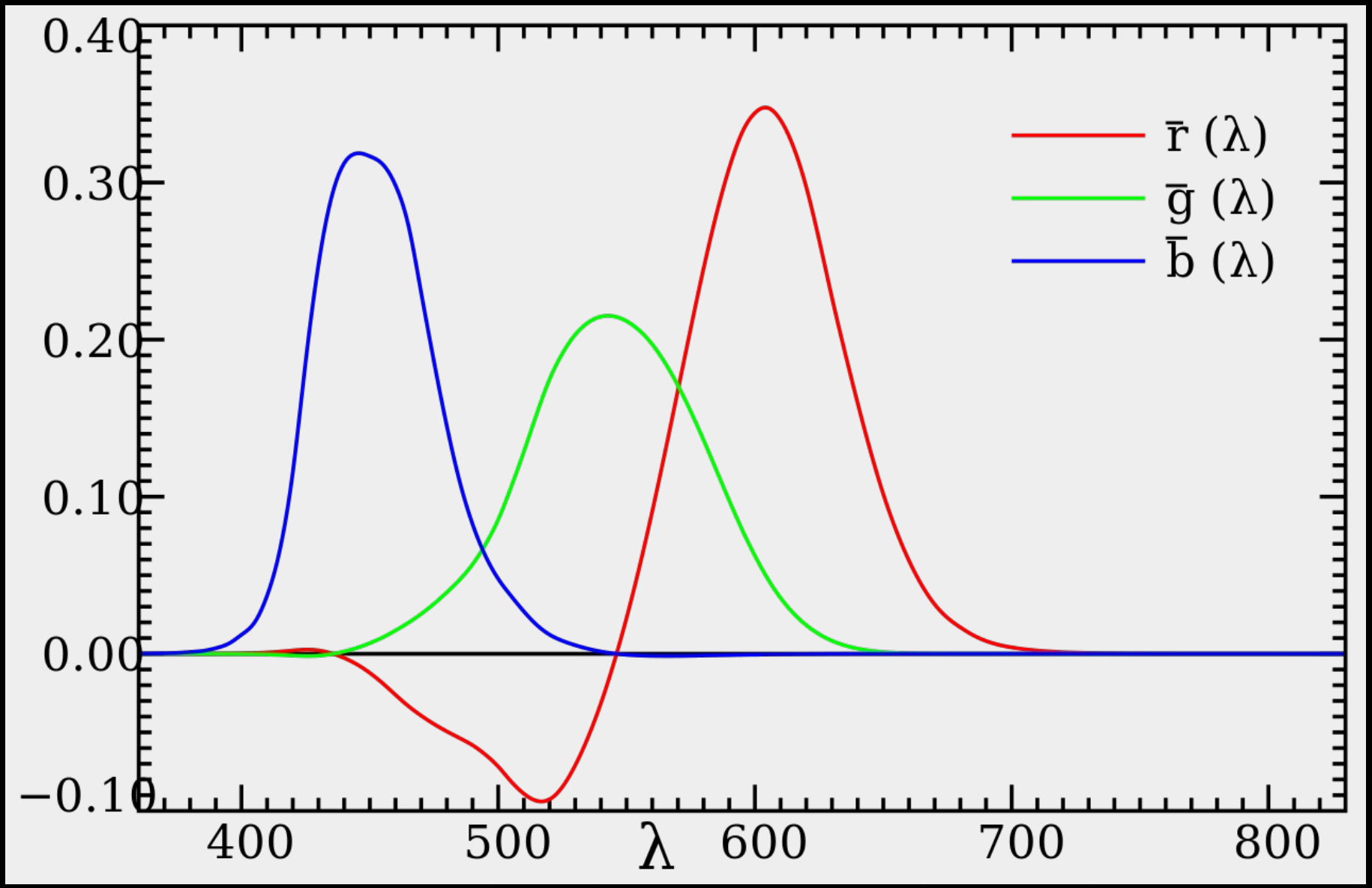

Incoming light
Filter layer
Sensor array
Resulting pattern


Full spectrum from outside
Foveon X3 sensor stack (1 pixel)
~ 7µm
Blue absorption
Green absorption
Red absorption
Blue sensor
Green sensor
Red sensor
Silicon Wafer
~ 5µm

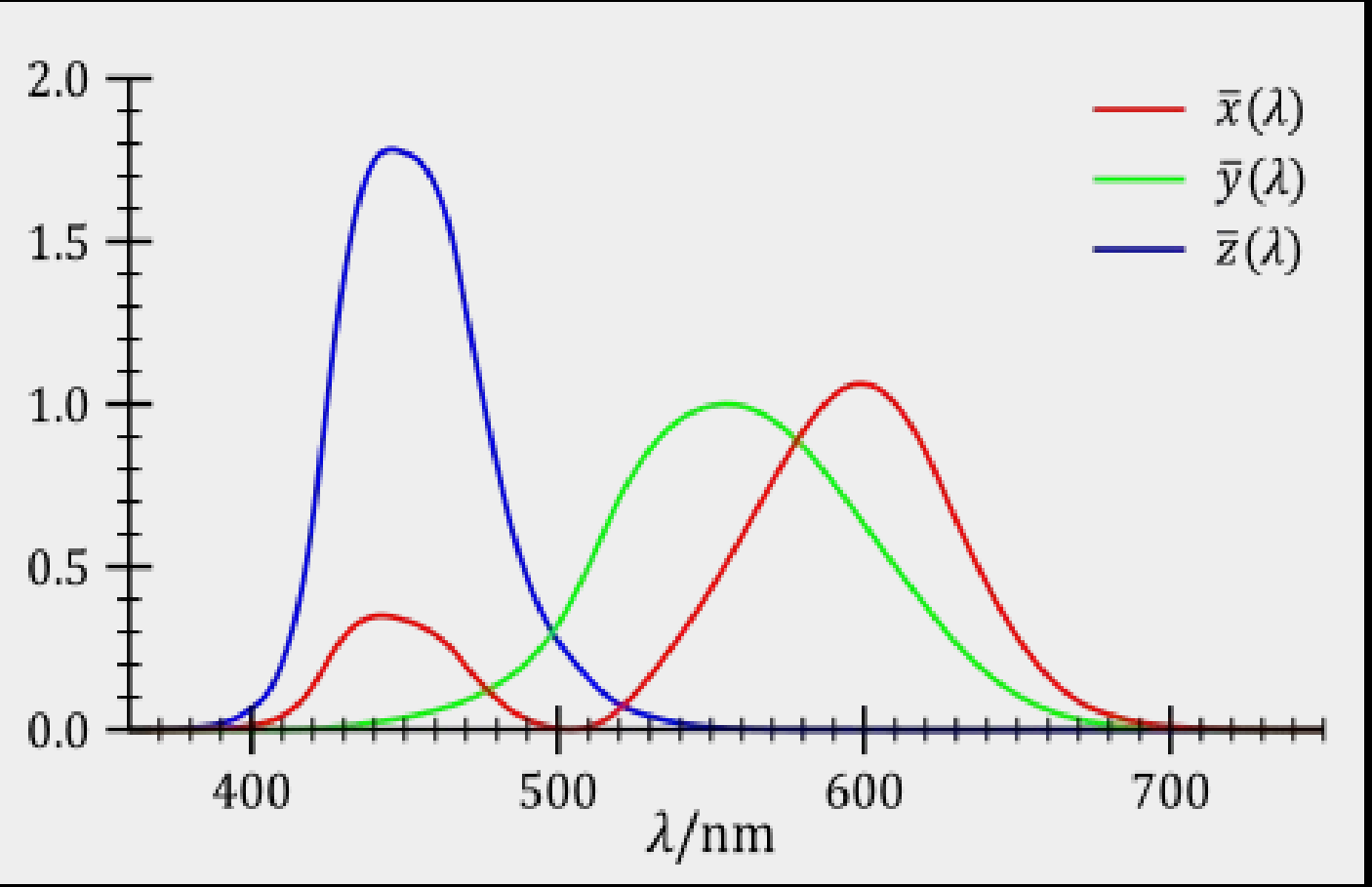

F2
B
C
A
F1



gstreamer



gstreamer



$$C = \bar{x}(\lambda)\,X + \bar{y}(\lambda)\,Y + \bar{z}(\lambda)\,Z$$

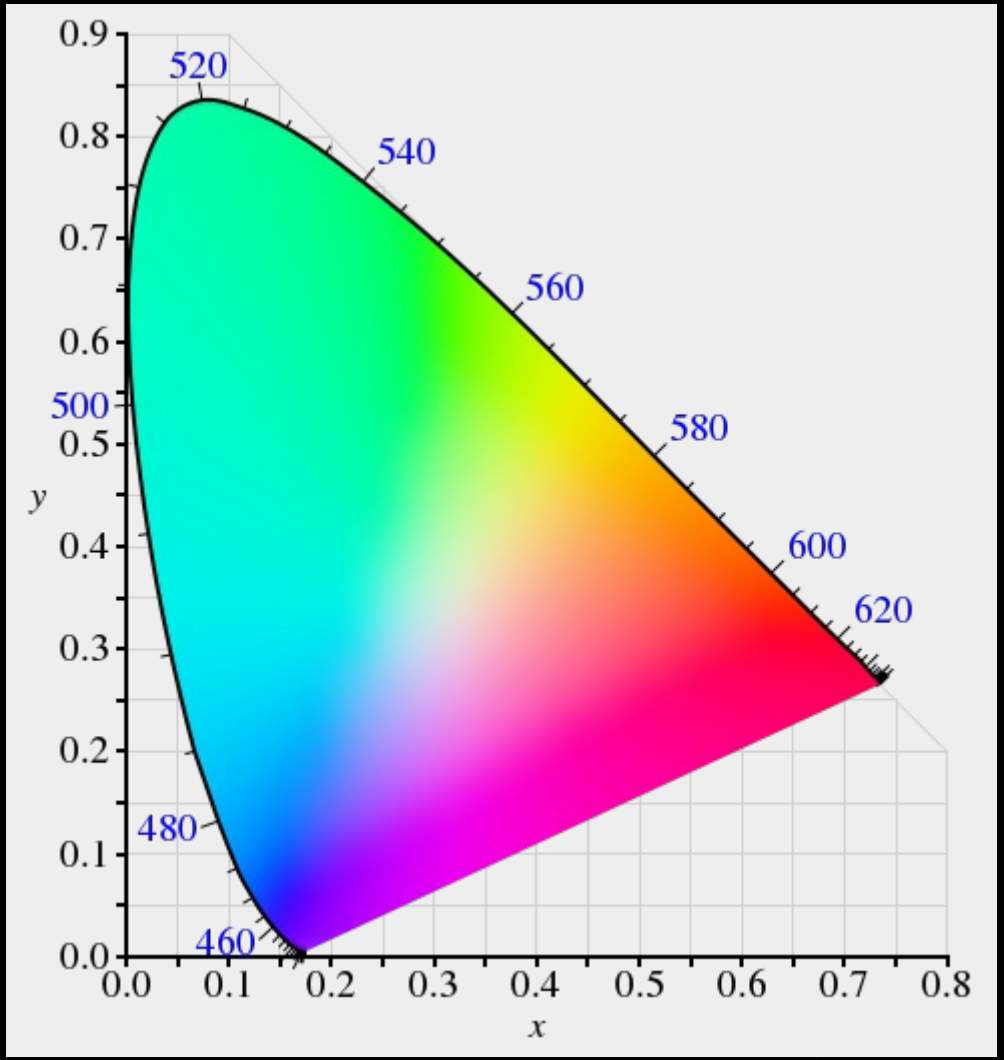
$$x = X / (X + Y + Z)$$

$$y = Y / (X + Y + Z)$$

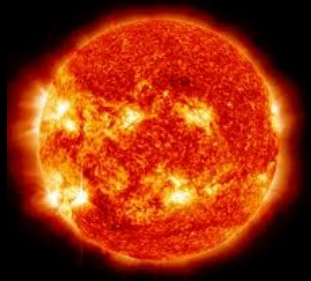
$$z = Z / (X + Y + Z)$$

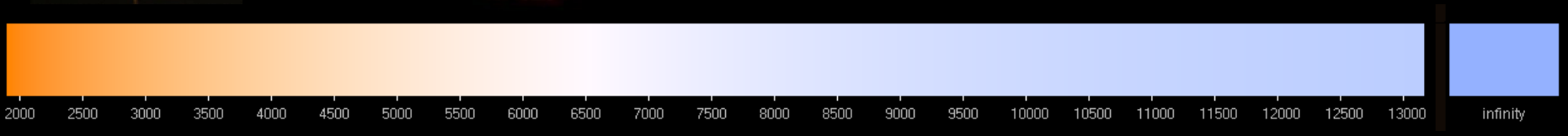
$$x + y + z = 1$$

gstreamer

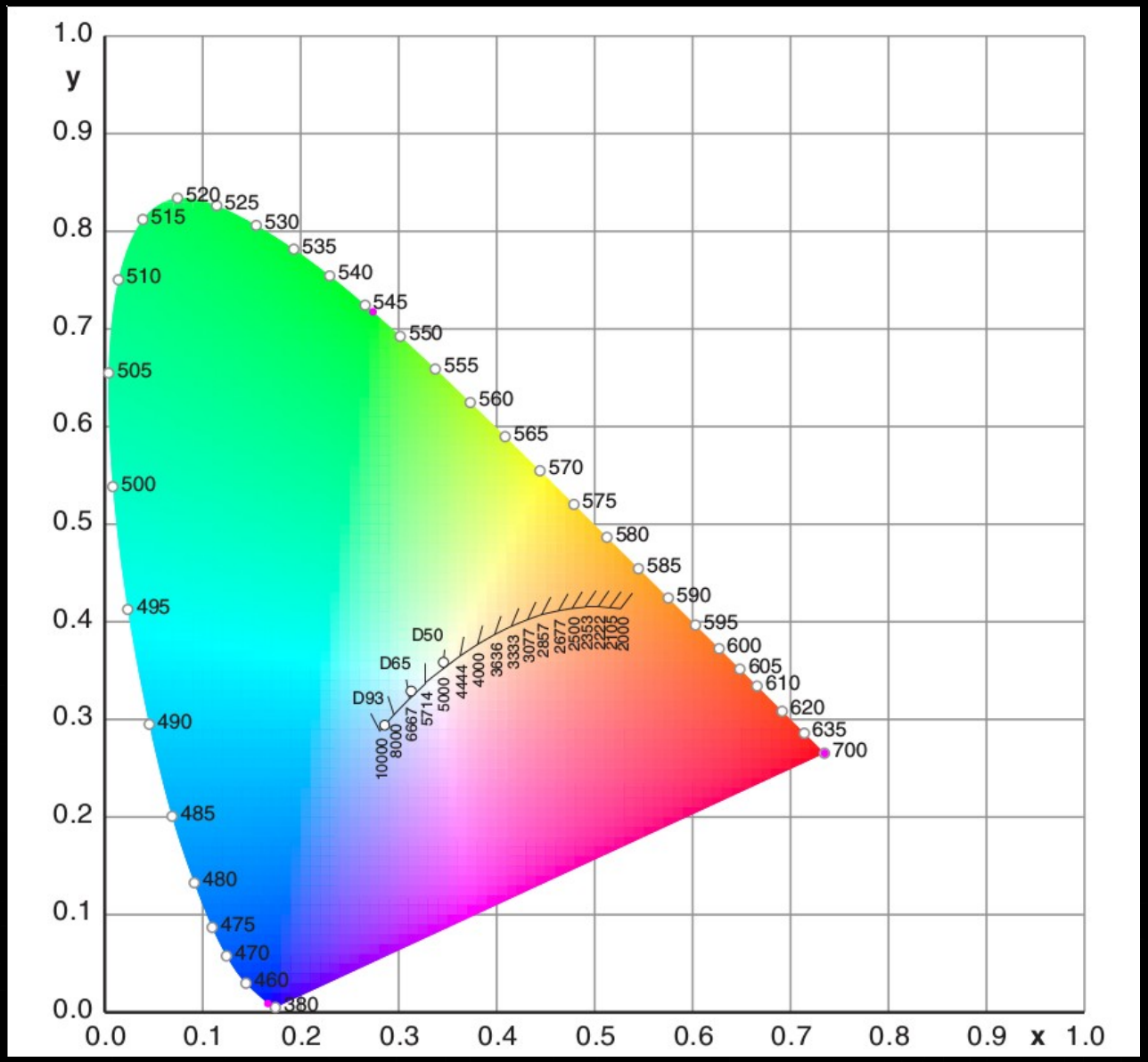

0.9
0.8
0.7
0.6
500
0.5
0.4
0.3
0.2
0.1
0.0
y
520
540
560
580
600
620
480
460
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x

gstreamer

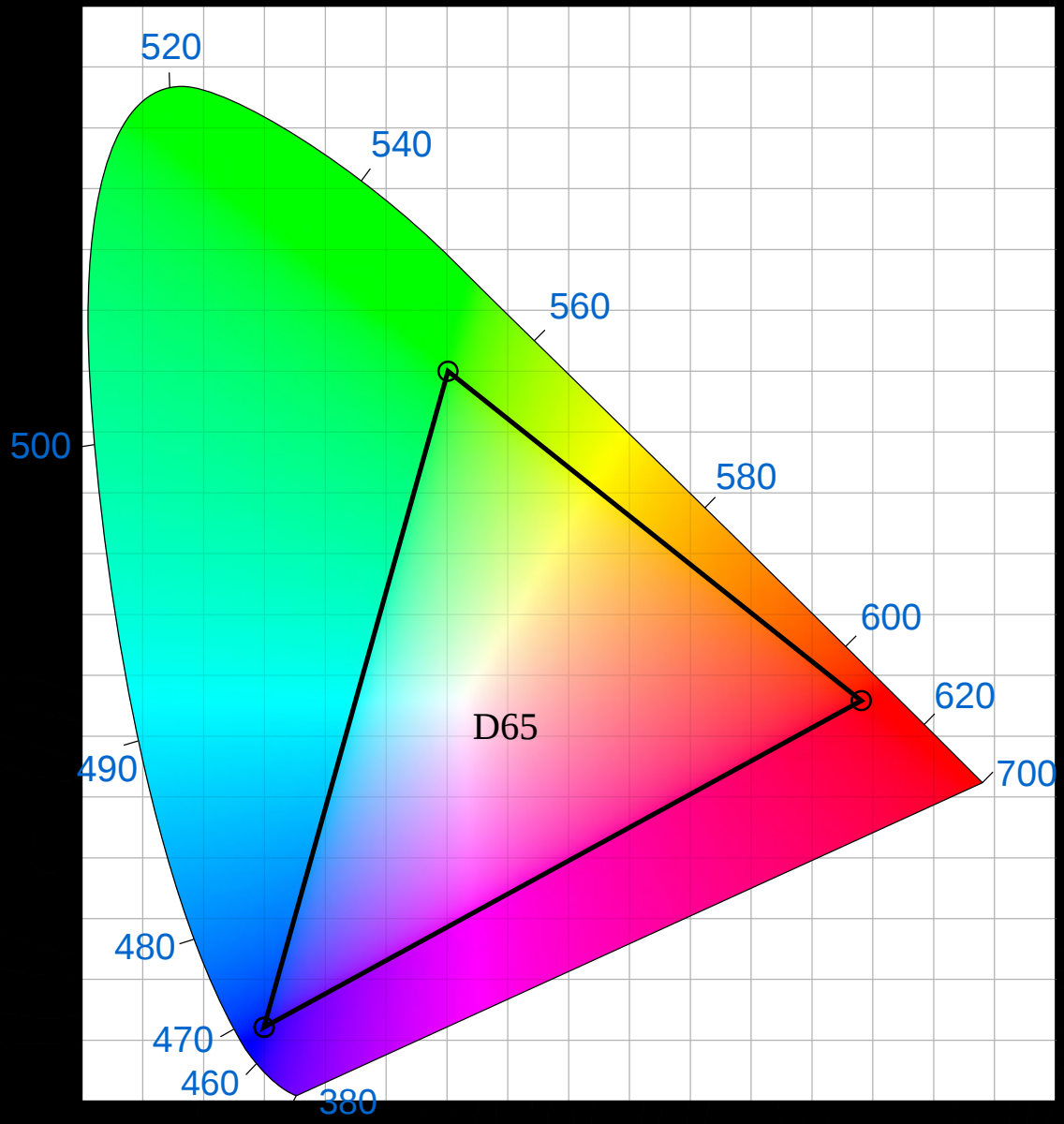



2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
7500
8000
8500
9000
9500
10000
10500
11000
11500
12000
12500
13000
infinity

gstreamer

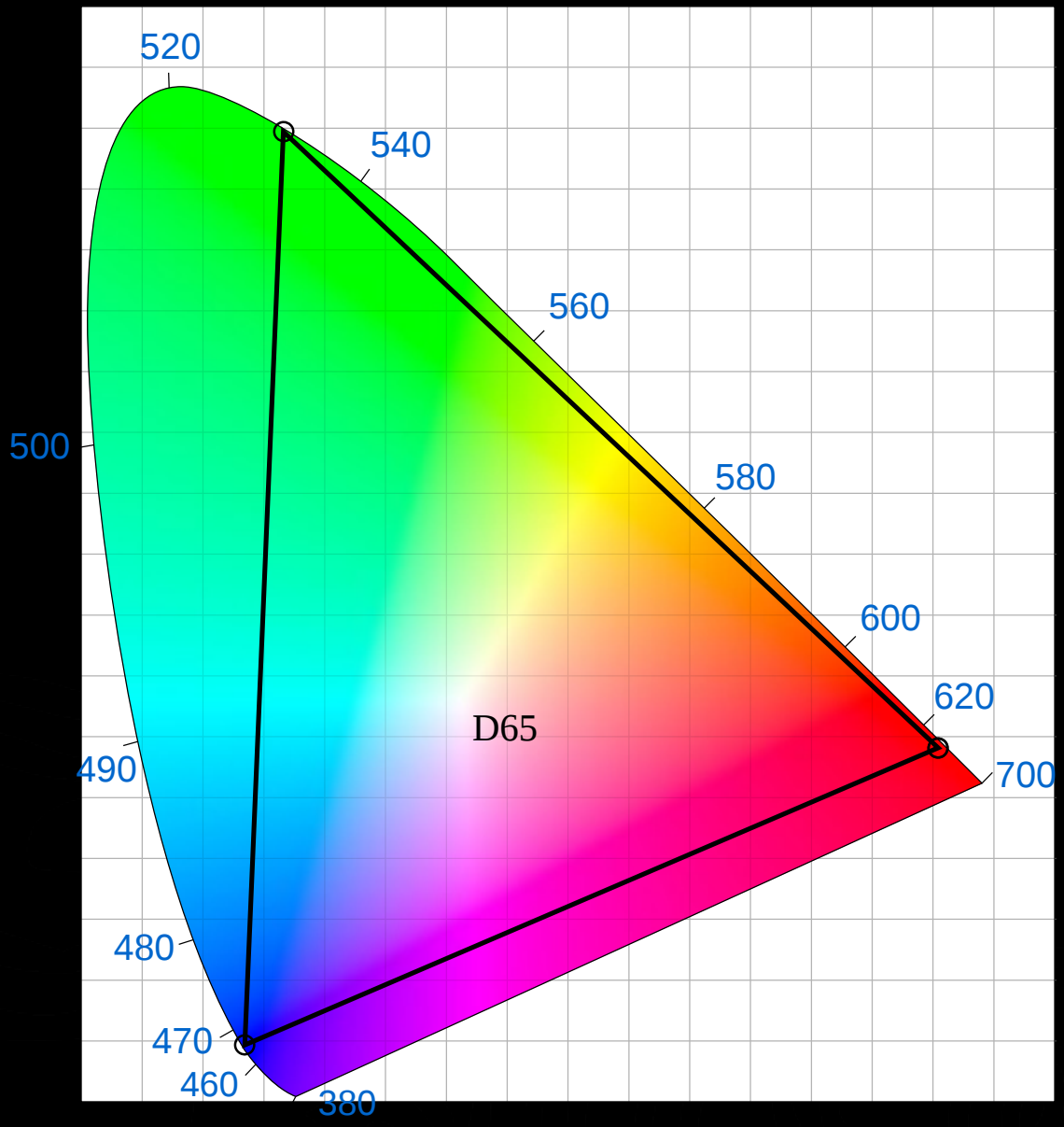

1.0
y
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
x
1.0
520
525
515
530
535
510
540
545
550
505
555
560
565
570
500
575
580
585
590
495
595
600
605
610
620
635
700
490
485
480
475
470
460
380
D50
D65
D93
10000
8000
6667
5714
5000
4444
4000
3636
3333
3077
2857
2677
2500
2353
2222
2105
2000

gstreamer



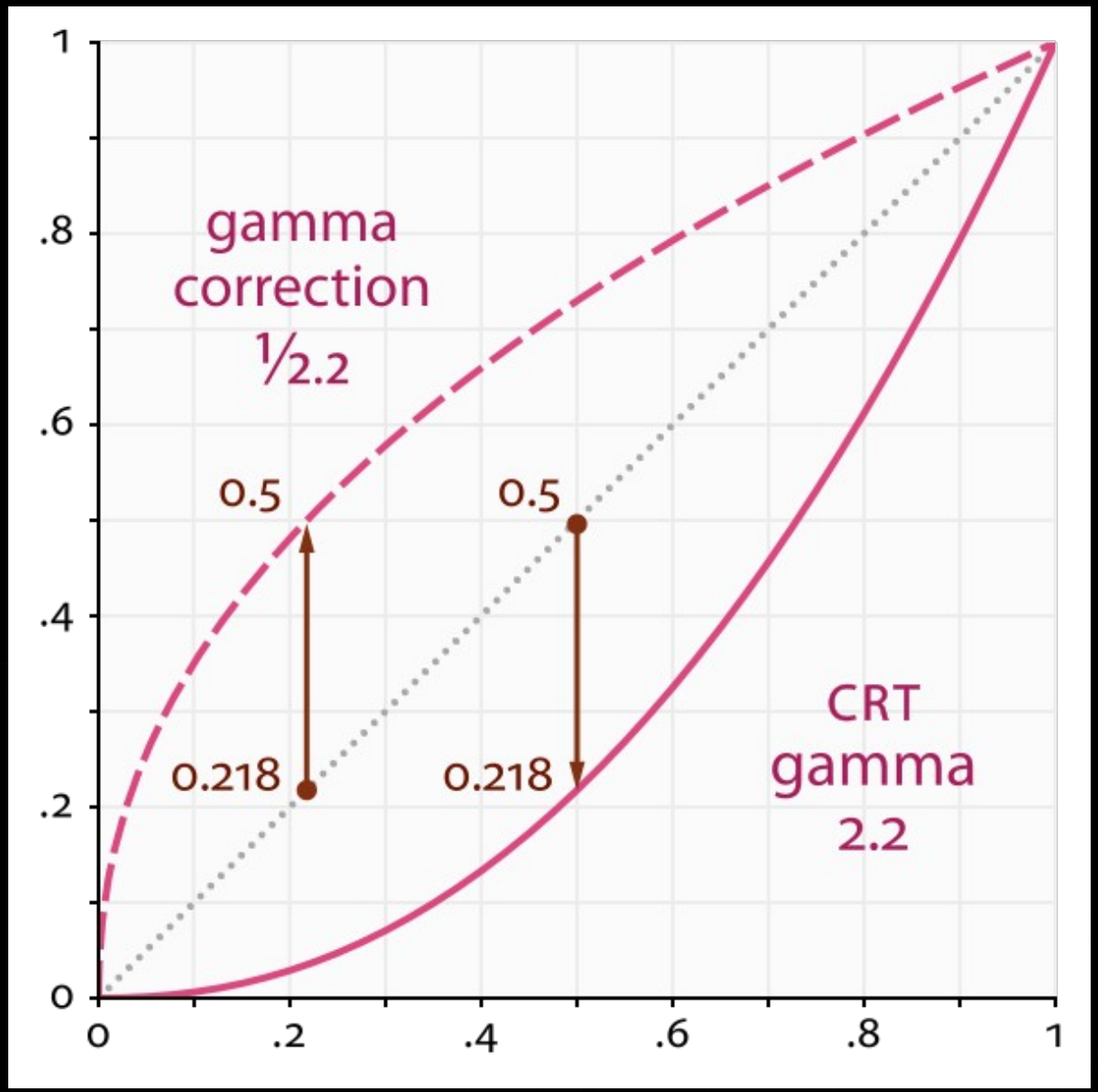
	x	y
R =	0.64	0.33
G =	0.30	0.60
B =	0.15	0.06
W=	0.3127	0.3290

gstreamer



	x	y
R =	0.708	0.292
G =	0.170	0.797
B =	0.131	0.046
W=	0.3127	0.3290

gstreamer



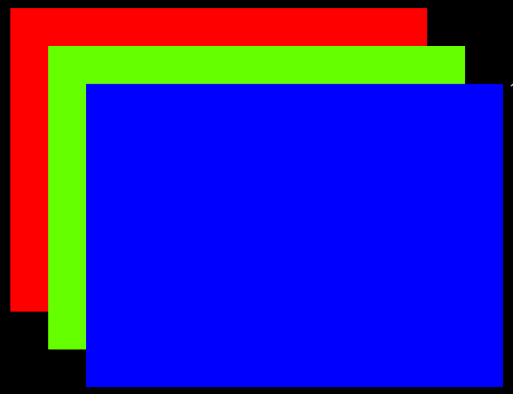
$4.5L$ $L \leq 0.018$

$1.099L^{0.45}-0.099$ $0.018 < L$

$E/4.5$ $E \leq 0.081$

$(E+0.099/1.099)^{2.2}$ $0.081 < E$

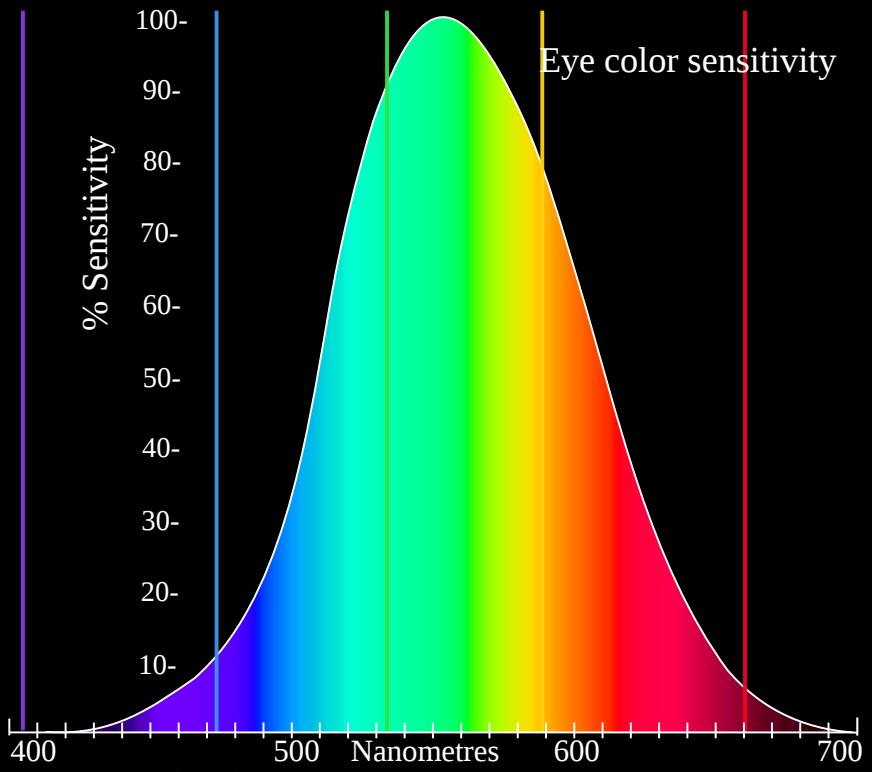
gstreamer



- Transfer
- Primaries

gstreamer

- RGB is not ideal to represent human visual system:
  - The eye is more sensitive to brightness than color
  - Green contributes a lot to our brightness perception

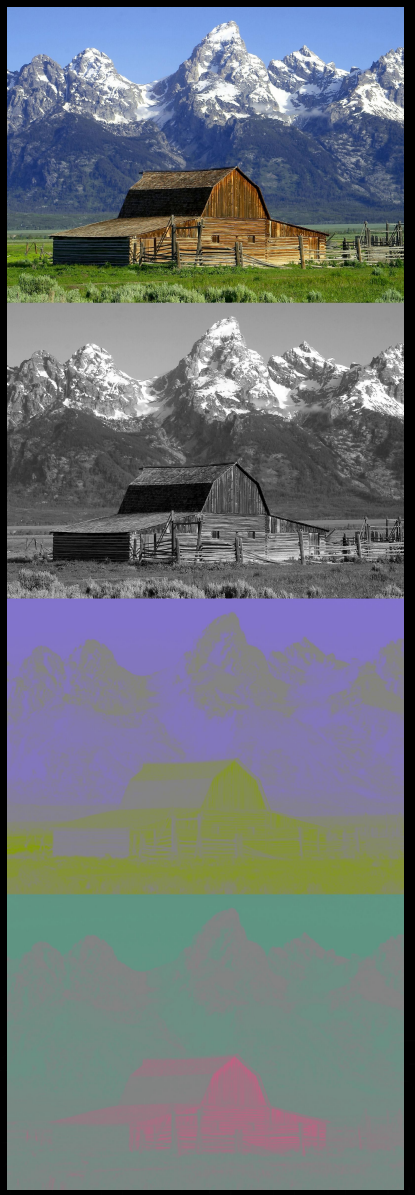


gstreamer

$$Y' = K_R \cdot R' + (1 - K_R - K_B) \cdot G' + K_B \cdot B'$$

$$P_B = \frac{1}{2} \cdot \frac{B' - Y'}{1 - K_B}$$

$$P_R = \frac{1}{2} \cdot \frac{R' - Y'}{1 - K_R}$$



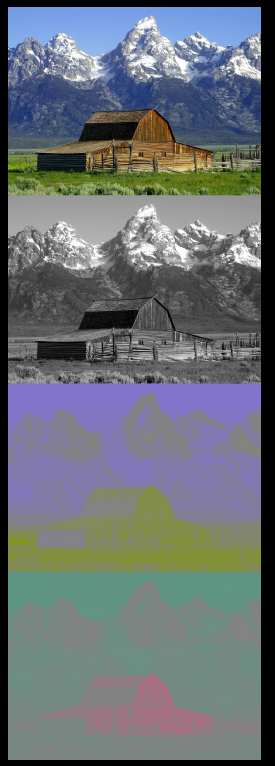
Kr and Kb as parameters

Y' in [0...1]
Pb/Pr in [-0.5..0.5]

gstreamer

Y'PbPr is then encoded to 8 bits per component

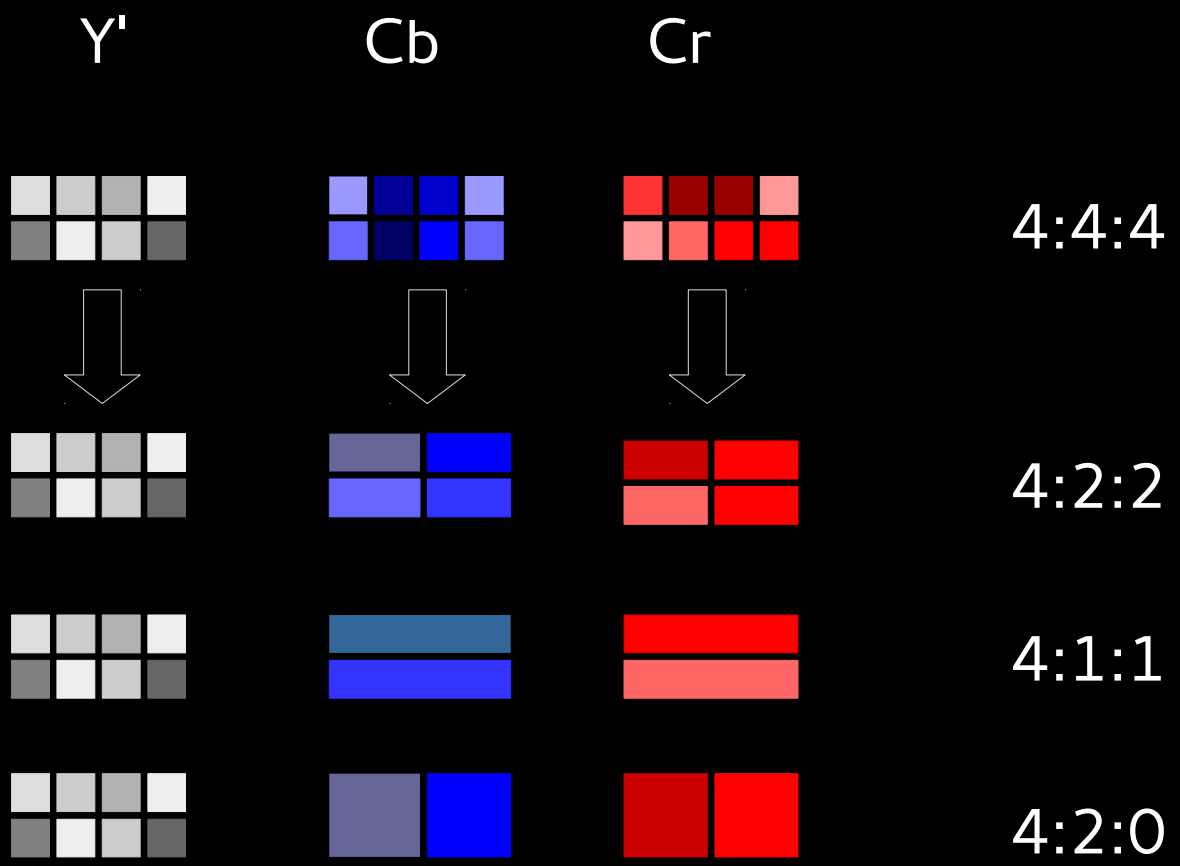
$$(Y', C_B, C_R) = (16, 128, 128) + (219 \cdot Y, 224 \cdot P_B, 224 \cdot P_R)$$



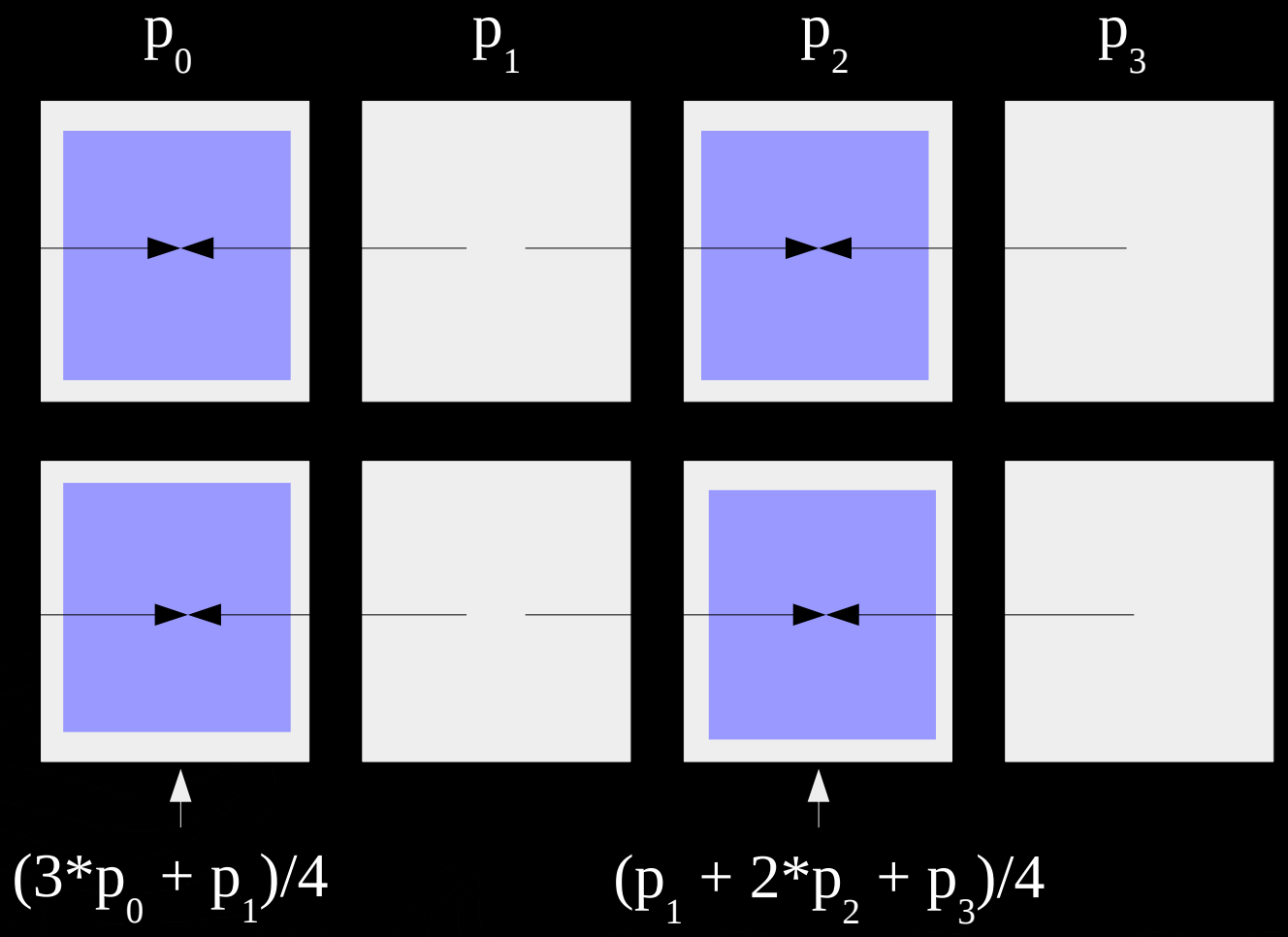
- Y' in [16..235]
- Cb/Cr in [16..240]
- digital Y'CbCr

We don't really need all that chroma info

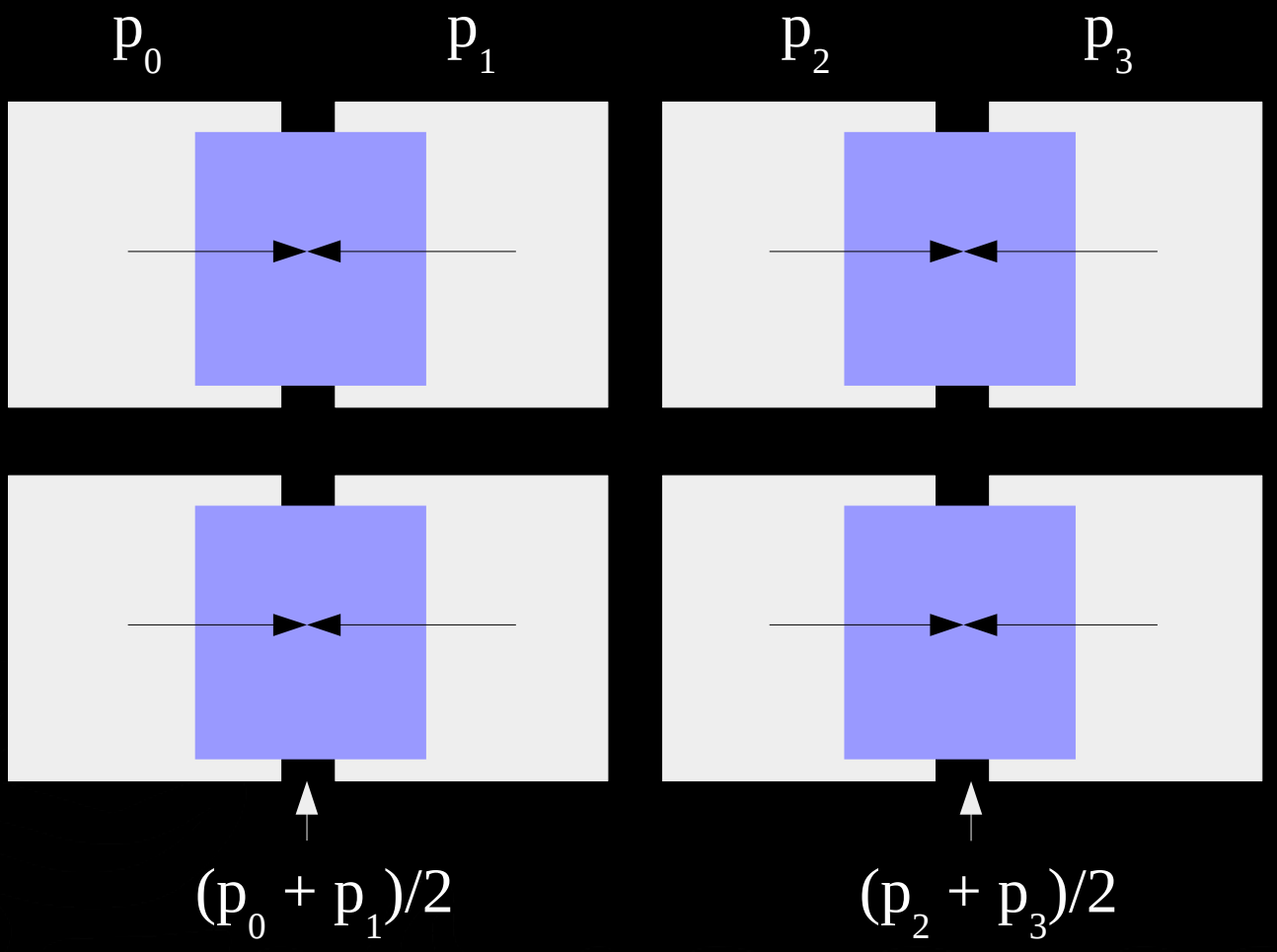
gstreamer



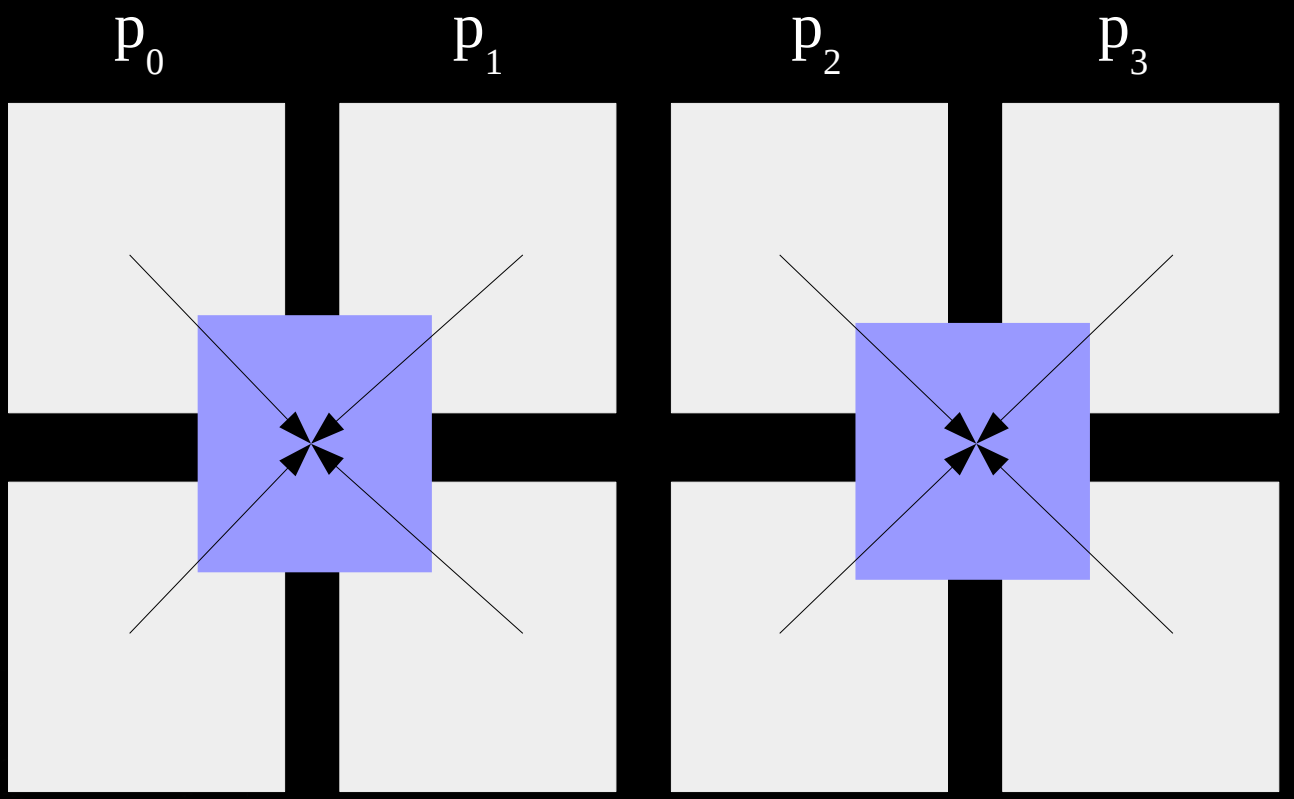
gstreamer



Cosited

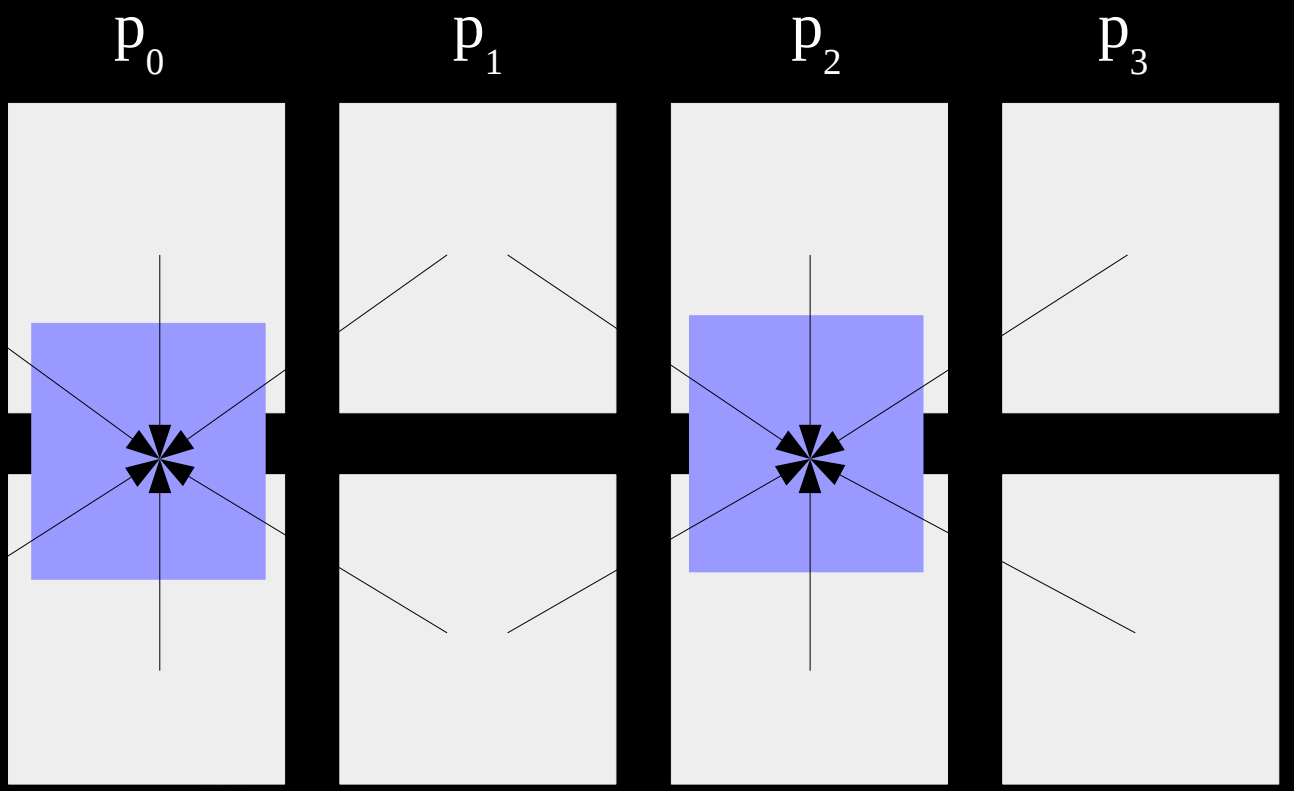


Centered

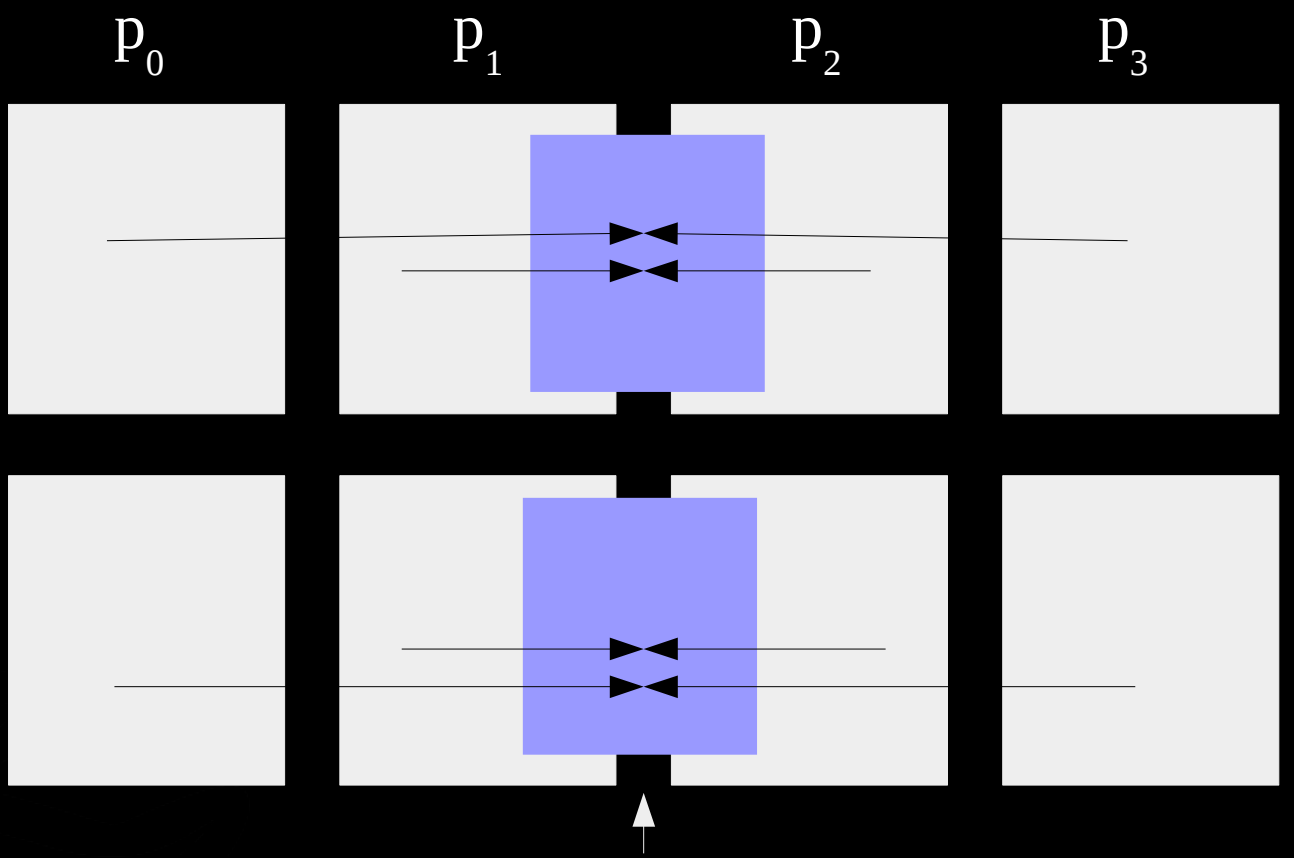


Horizontally centered

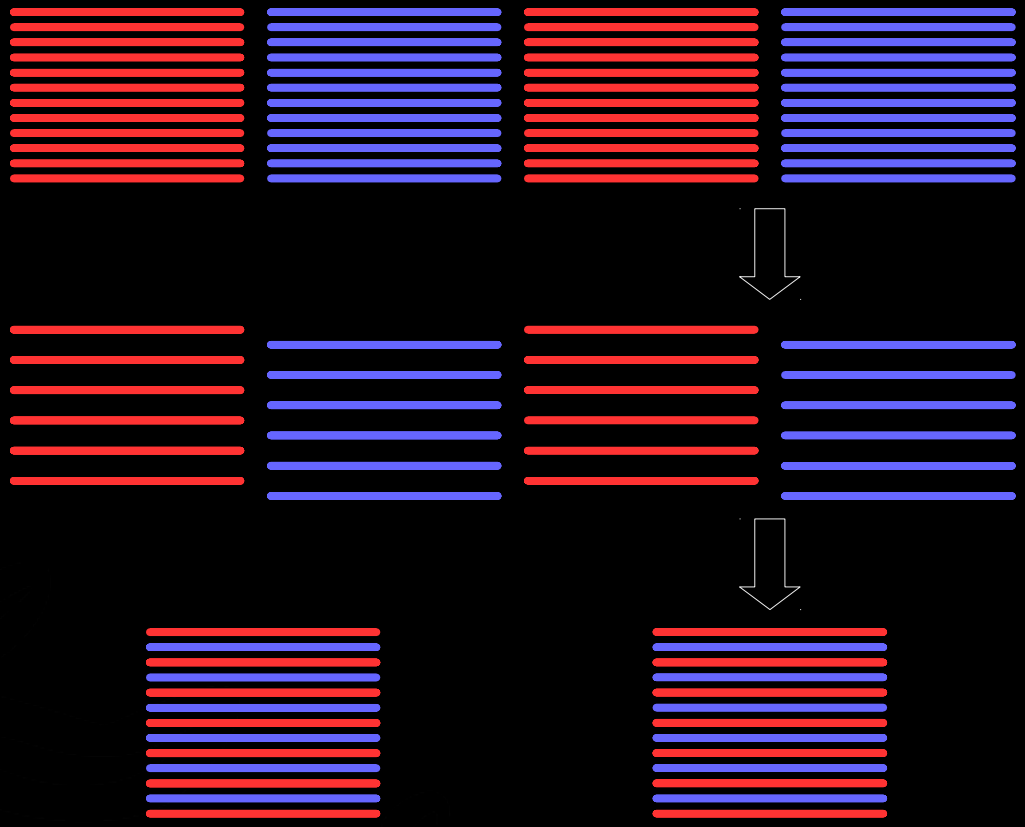
Vertically centered



Horizontally cosited
Vertically centered



4x Centered

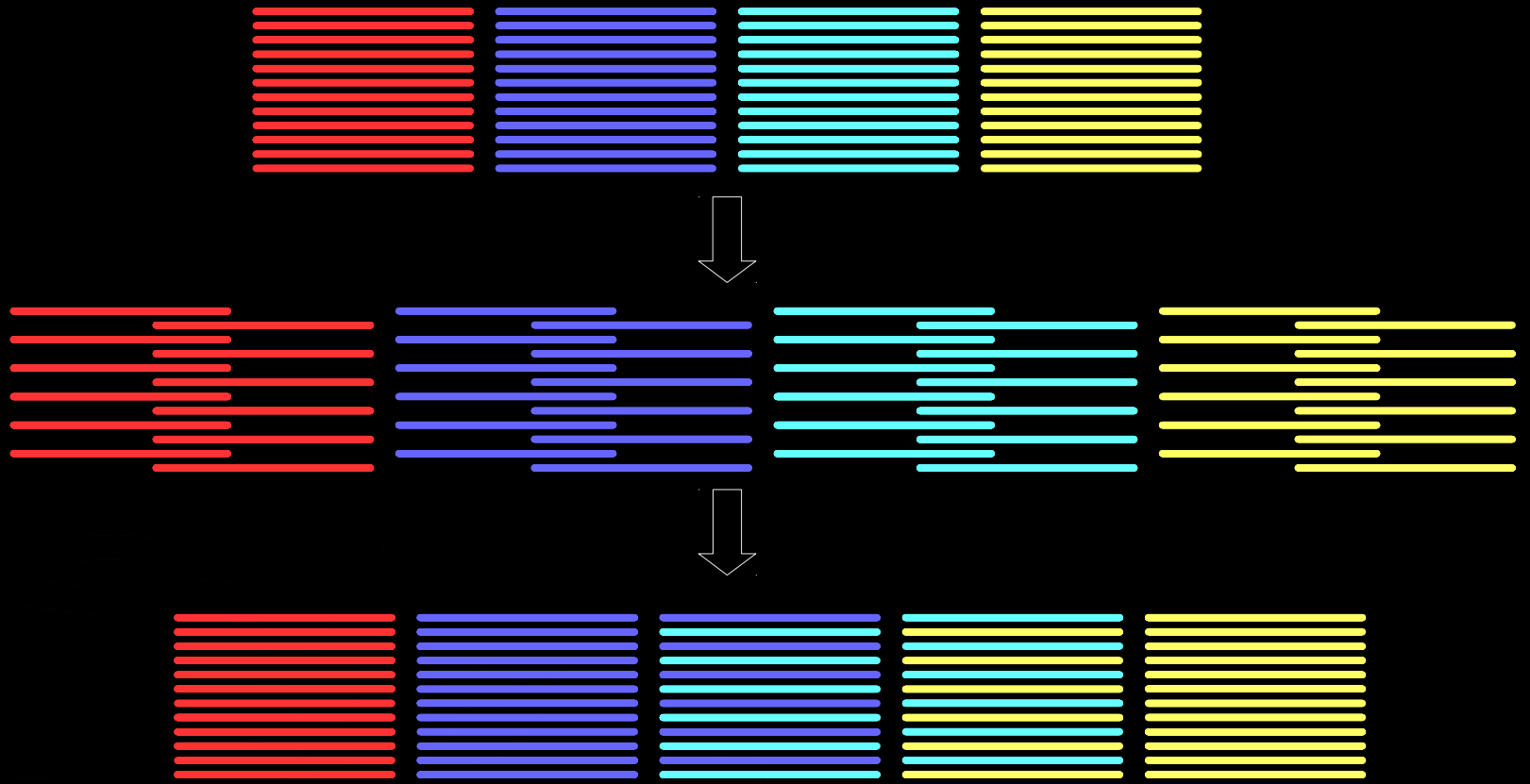


frames

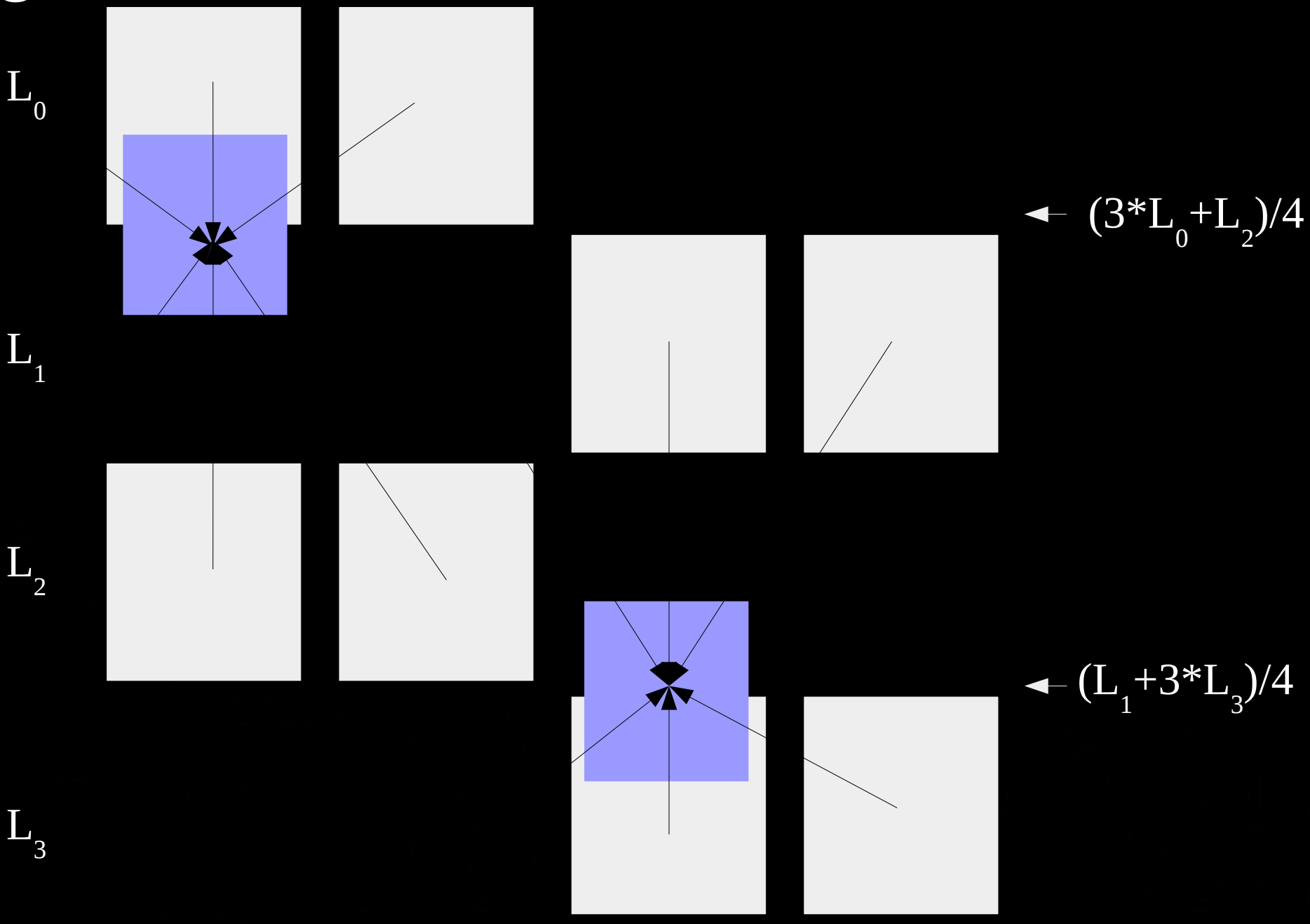
fields

interlaced frames

gstreamer



gstreamer

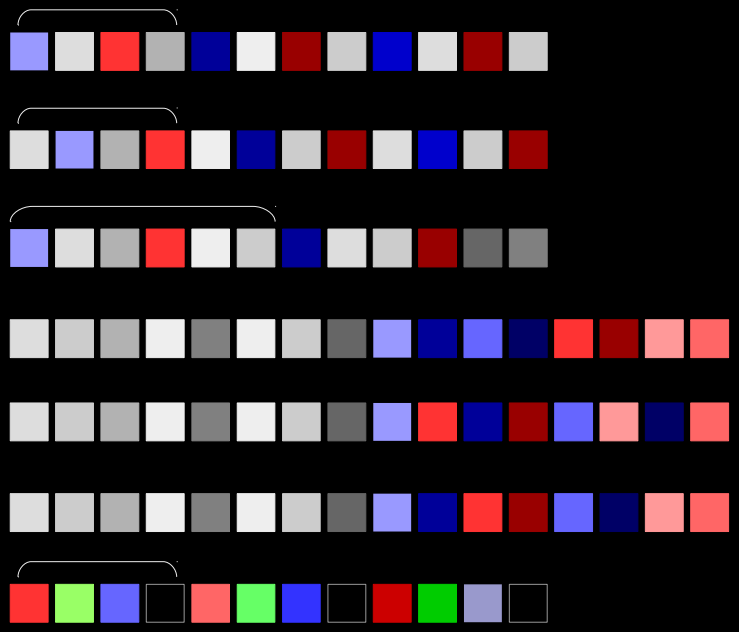


Y'	[16..235]	[64...940]
Cb/Cr	[16..240]	[64..960]
R/G/B	[0..255]	[0...1023]

But also..

Y/Cb/Cr	[0...255]
R/G/B	[16..235]

gstreamer



UYVY
YUY2
IYU1
I420
NV12
IMC2
RGBx
....

gstreamer

RGB

R'G'B'

Y'PbPr

Y'PbPr

IMC2

H.264
MPEG-4/AVC

Primaries

Transfer

Matrix

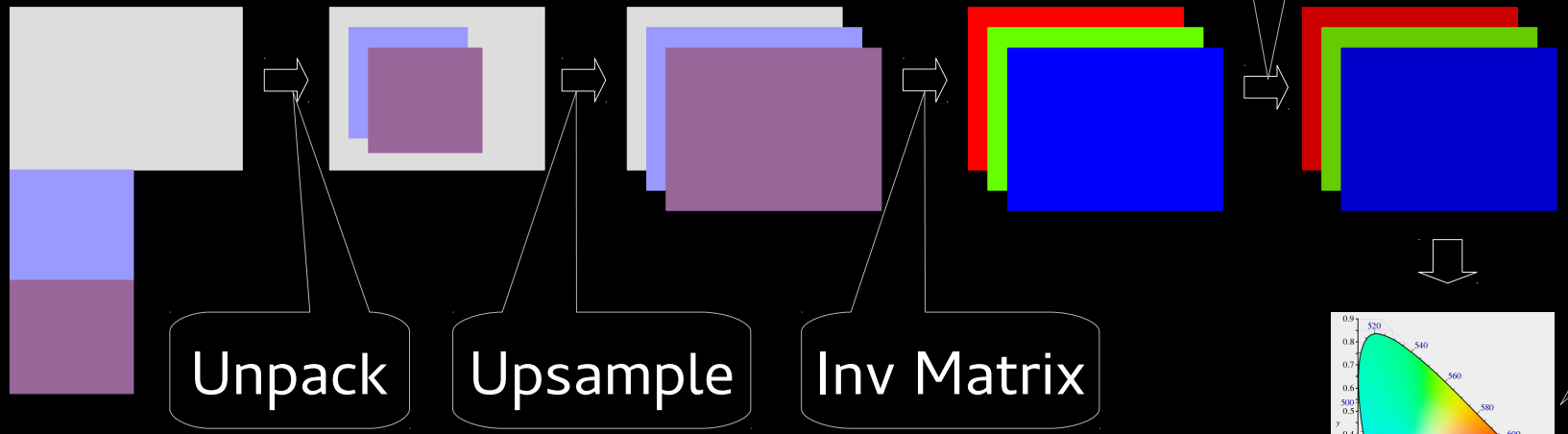
Subsampling
Siting

Range
Packing

gstreamer

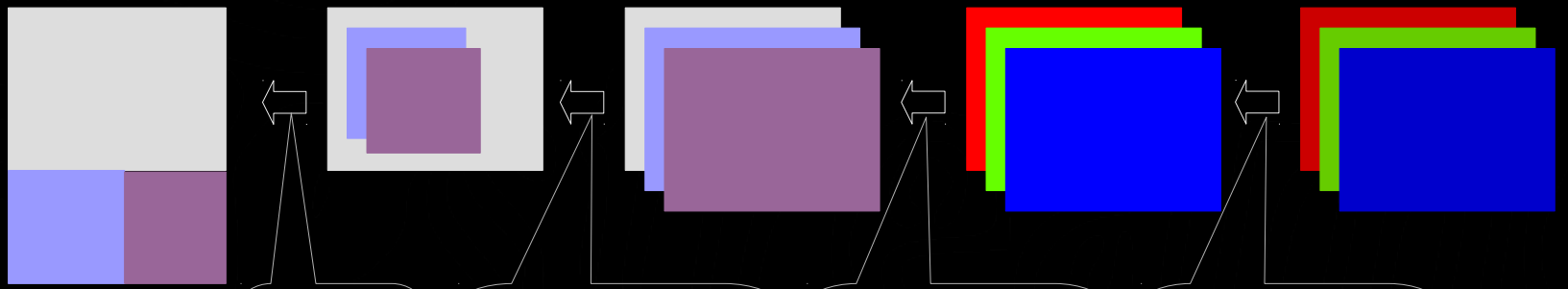
I420 4:2:0 Y'CbCr R'G'B' RGB

Inv Transfer



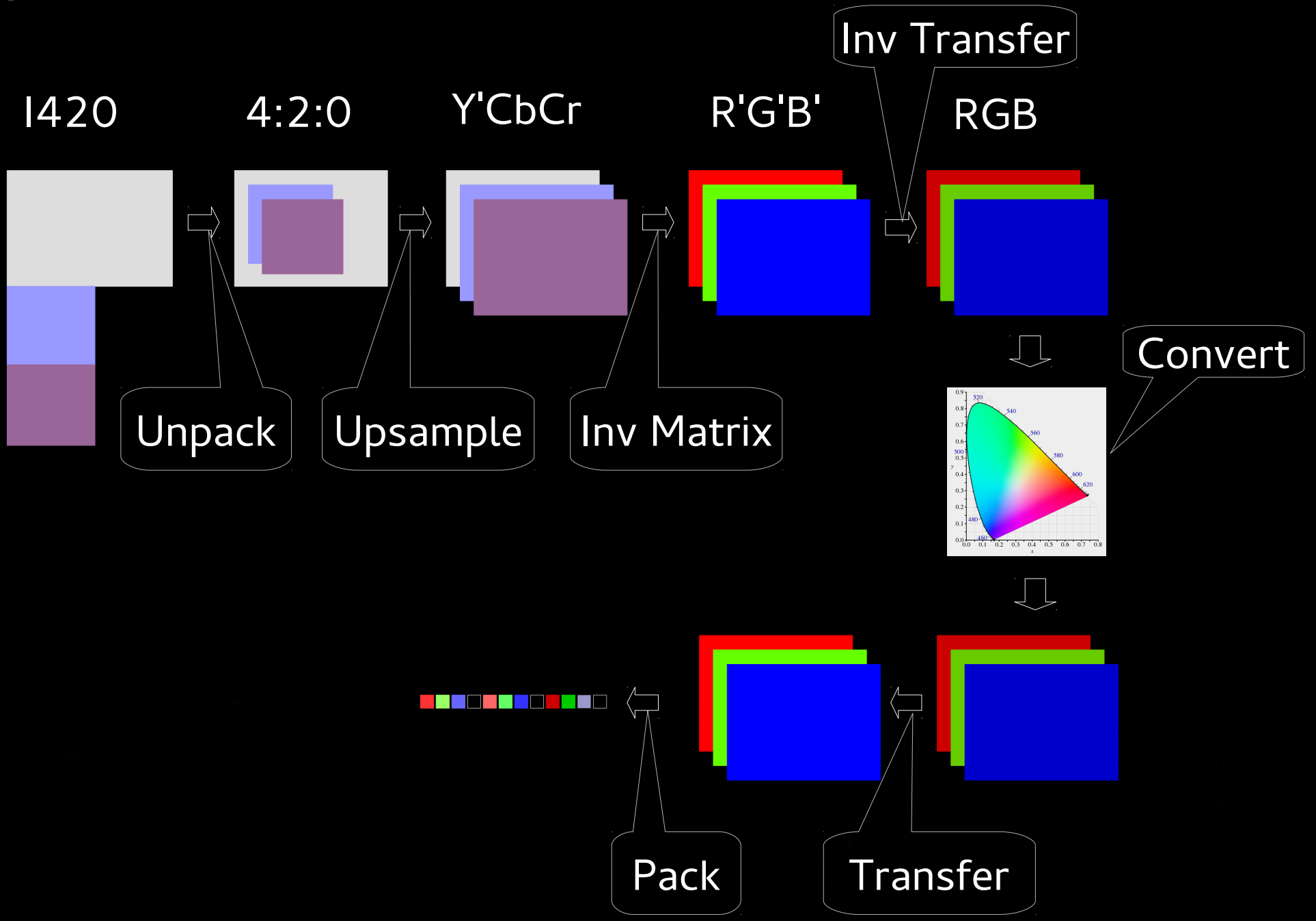
Unpack Upsample Inv Matrix

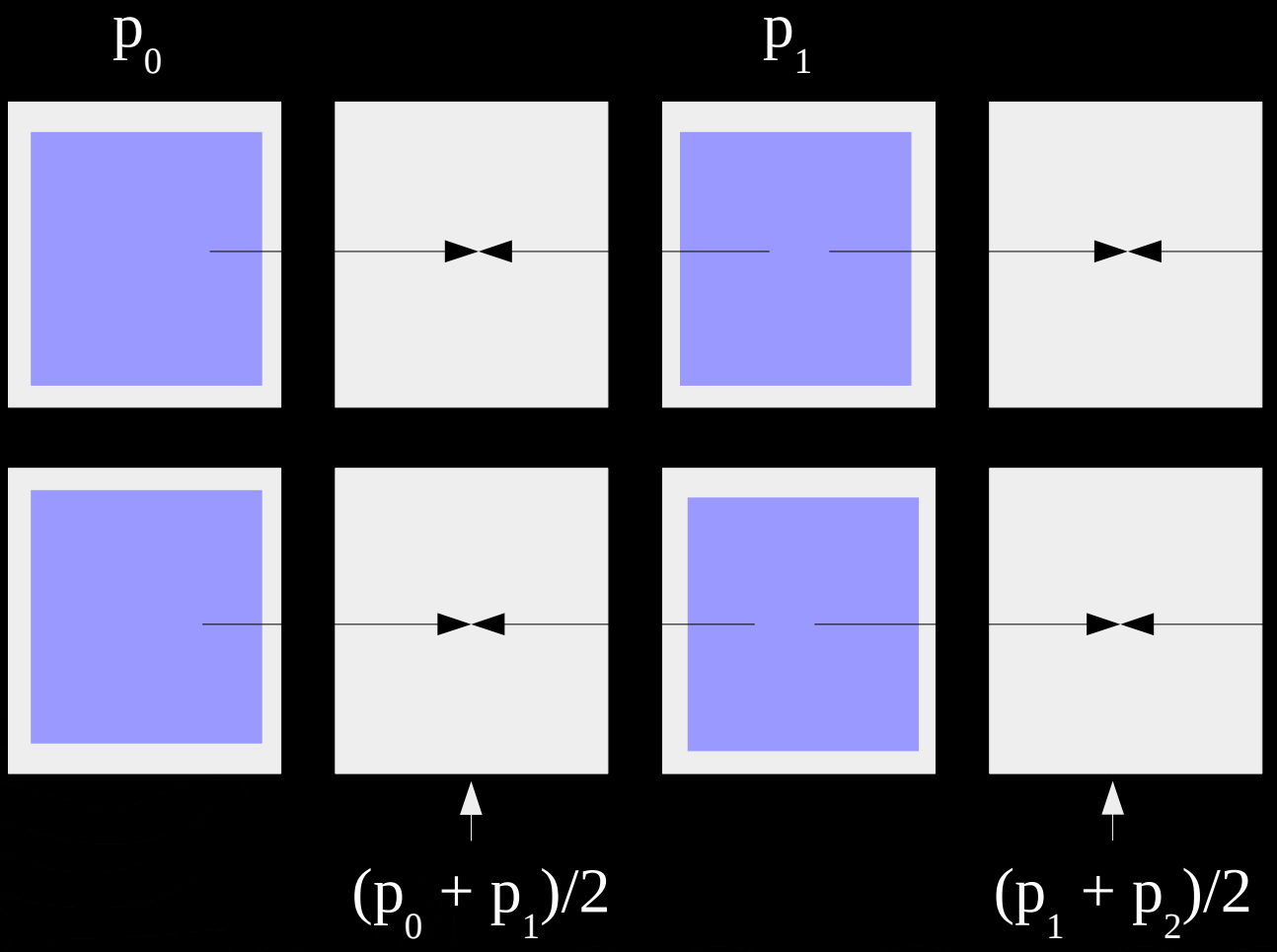
Convert



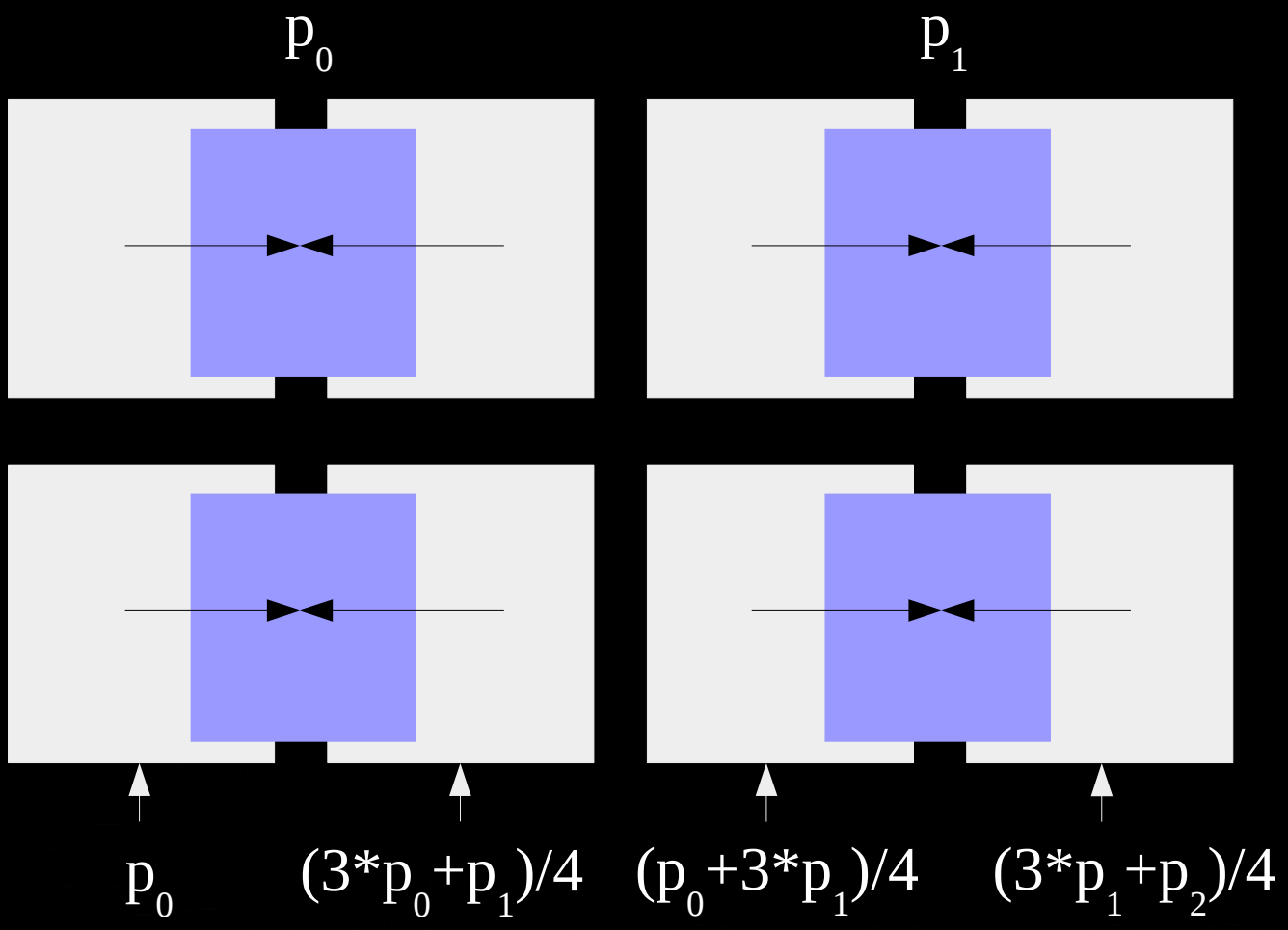
Pack Subsample Matrix Transfer

gstreamer

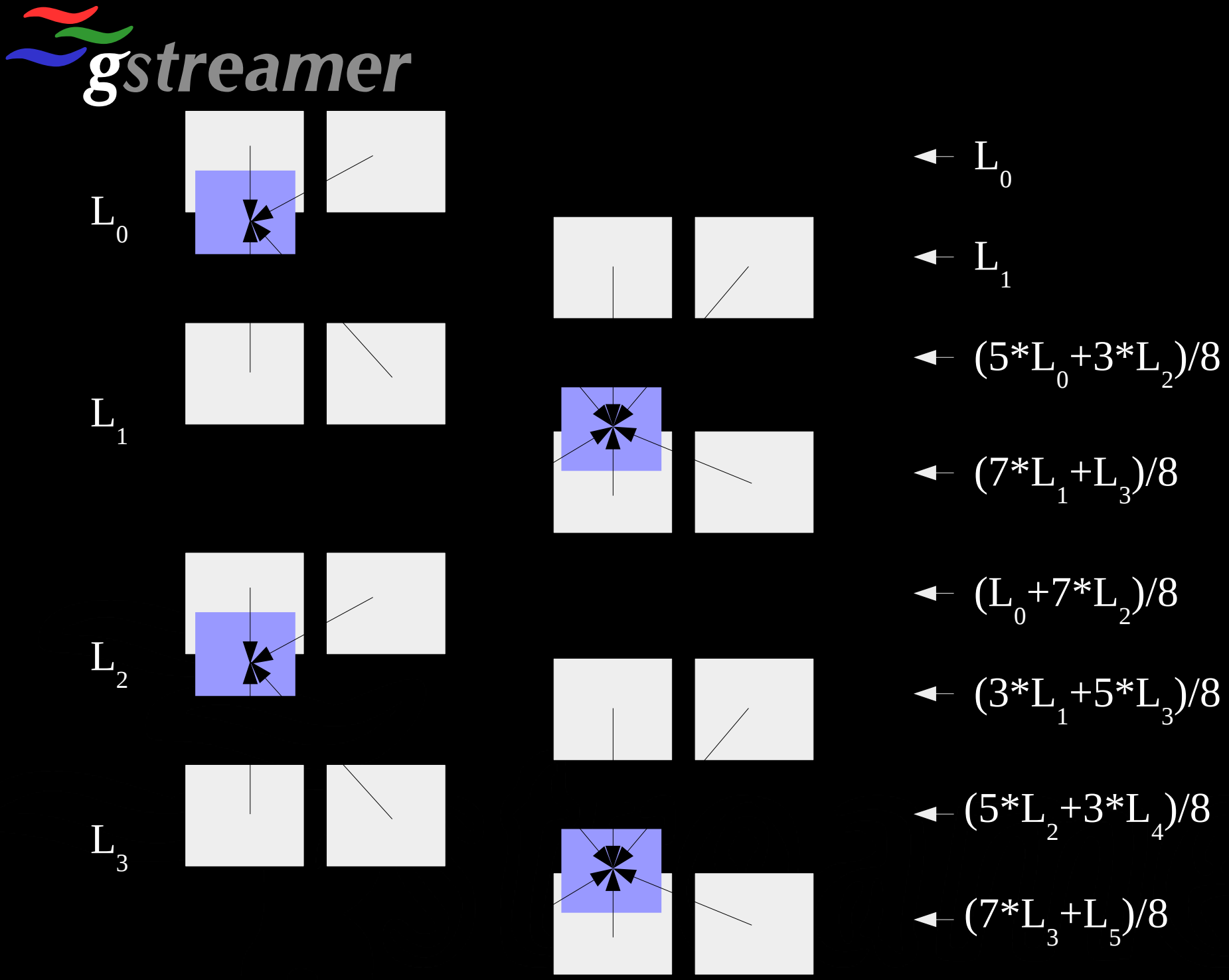

I420
4:2:0
Y'CbCr
R'G'B'
RGB
Inv Transfer
Unpack
Upsample
Inv Matrix
Convert
Pack
Transfer



Cosited

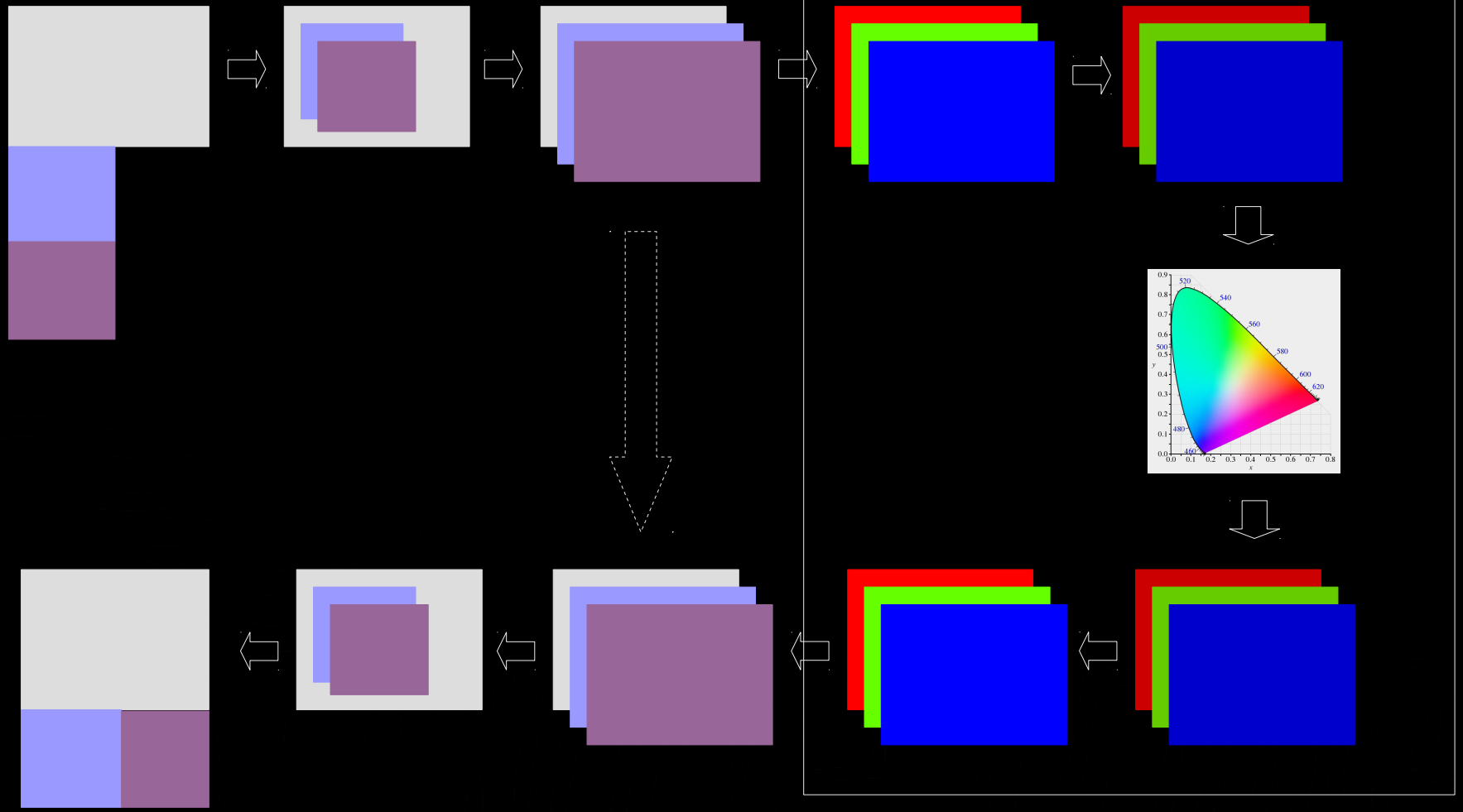


Centered

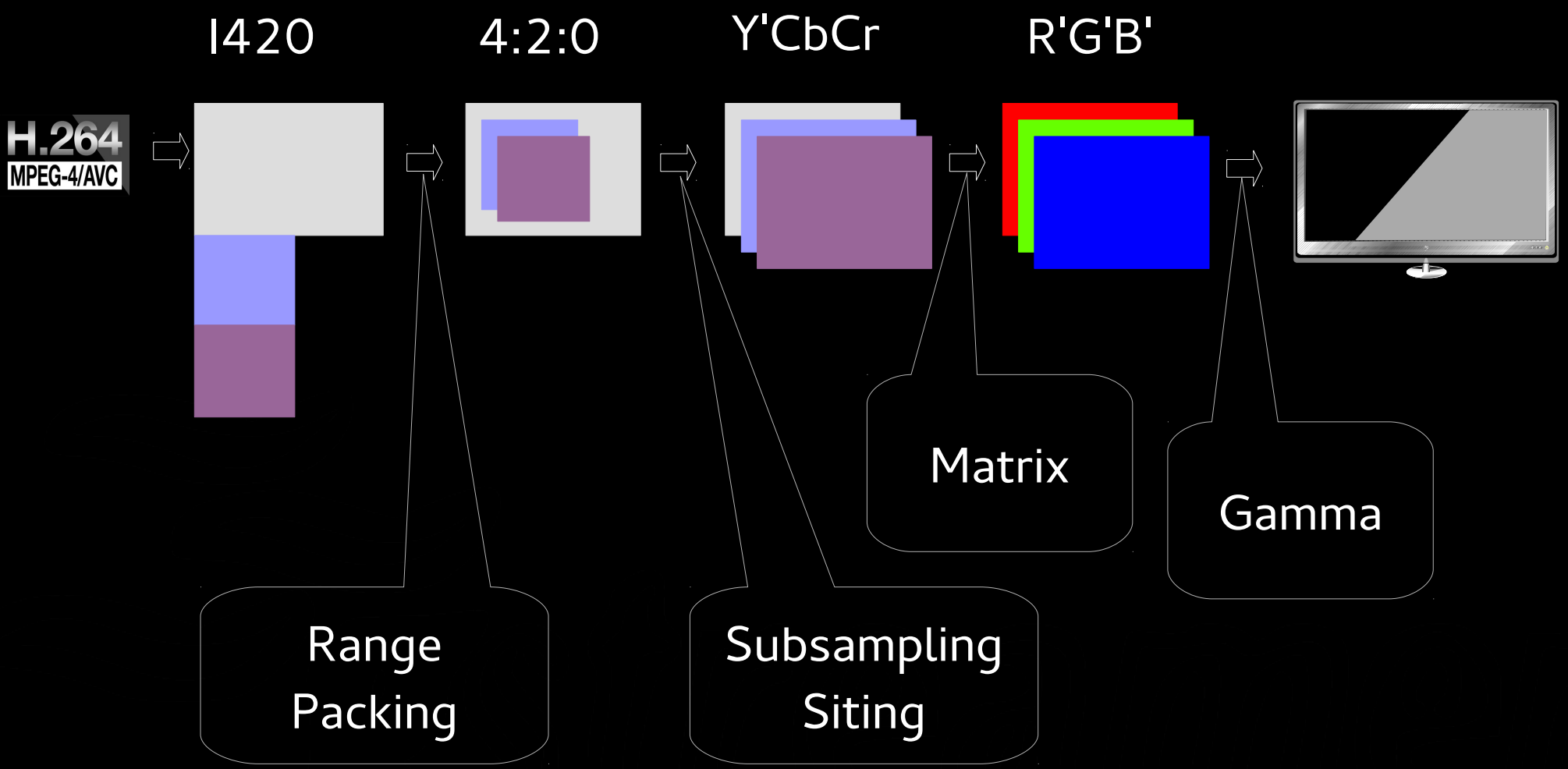

gstreamer
L0
L1
L2
L3
L0
L1
(5*L0+3*L2)/8
(7*L1+L3)/8
(L0+7*L2)/8
(3*L1+5*L3)/8
(5*L2+3*L4)/8
(7*L3+L5)/8

gstreamer

I420 4:2:0 Y'CbCr R'G'B' RGB



gstreamer


I420
4:2:0
Y'CbCr
R'G'B'
H.264
MPEG-4/AVC
Range
Packing
Subsampling
Siting
Matrix
Gamma

gst-libs/gst/video/video.h

- Packing (video-format.h)
- Range (video-color.h)
- Chroma siting (video-chroma.h)
- Matrix (video-color.h)
- Transfer (video-color.h)
- Primaries (video-color.h)

- Video-converter object that implements conversion, used by videoconvert element

caps

- Packing — format
- Range — colorimetry
- Chroma siting — chroma-site
- Matrix — colorimetry
- Transfer — colorimetry
- Primaries — colorimetry

Colorimetry :

- <range>:<matrix>:<transfer>:<primaries>
- Or bt601, bt709

What's not done:

- Conversion without bypassing transfer function
- Various (unused) down/upsampling methods. Interlaced downsampling is however useful and missing.
- More optimized paths for various common conversions.
- Only linear up/downsampling of chroma.


gstreamer

?