

[Advertise
on this
site.](#)

YUV Formats

YUV formats fall into two distinct groups, the [packed formats](#) where Y, U (Cb) and V (Cr) samples are packed together into macropixels which are stored in a single array, and the [planar formats](#) where each component is stored as a separate array, the final image being a fusing of the three separate planes.

In the diagrams below, the numerical suffix attached to each Y, U or V sample indicates the sampling position across the image line, so, for example, V0 indicates the leftmost V sample and Yn indicates the Y sample at the (n+1)th pixel from the left.

Subsampling intervals in the horizontal and vertical directions may merit some explanation. The horizontal subsampling interval describes how frequently across a line a sample of that component is taken while the vertical interval describes on which lines samples are taken. For example, UYVY format has a horizontal subsampling period of 2 for both the U and V components indicating that U and V samples are taken for every second pixel across a line. Their vertical subsampling period is 1 indicating that U and V samples are taken on each line of the image.

For YVU9, though, the vertical subsampling interval is 4. This indicates that U and V samples are only taken on every fourth line of the original image. Since the horizontal sampling period is also 4, a single U and a single V sample are taken for each square block of 16 image pixels.

Also, if you are interested in YCrCb to RGB conversion, you may find [this page](#) helpful.

People reading this page may be interested in a freeware codec from Drastic Technologies which allegedly handles the vast majority of YUV formats listed here. I've not tried it but you can find it [here](#).

Packed YUV Formats

Label	FOURCC in Hex	Bits per pixel	Description
AYUV	0x56555941	32	Combined YUV and alpha
CLJR	0x524A4C43	8	Cirrus Logic format with 4 pixels packed into a u_int32. A form of YUV 4:1:1 with less than 8 bits per Y, U and V sample.
cyuv	0x76757963	16	Essentially a copy of UYVY except that the sense of the height is reversed - the image is upside down with respect to the UYVY version.
GREY	0x59455247	8	Apparently a duplicate of Y800 (and also, presumably, "Y8")
IRAW	0x57615349	?	Intel uncompressed YUV. I have no information on this format - can you help?
IUYV	0x56595549	16	Interlaced version of UYVY (line order 0, 2, 4,...,1, 3, 5,...) registered by Silviu Brnzei of LEAD Technologies .
IY41	0x31345949	12	Interlaced version of Y41P (line order 0, 2, 4,...,1, 3, 5,...) registered by Silviu Brnzei of LEAD Technologies .
IYU1	0x31555949	12	12 bit format used in mode 2 of the IEEE 1394 Digital Camera 1.04 spec. This is equivalent to Y411

IYU2	0x32555949	24	24 bit format used in mode 0 of the IEEE 1394 Digital Camera 1.04 spec
HDYC	0x43594448	16	YUV 4:2:2 (Y sample at every pixel, U and V sampled at every second pixel horizontally on each line). A macropixel contains 2 pixels in 1 u_int32. This is a suplicate of UYVY except that the color components use the BT709 color space (as used in HD video).
UYNV	0x564E5955	16	A direct copy of UYVY registered by NVidia to work around problems in some old codecs which did not like hardware which offered more than 2 UYVY surfaces.
UYVP	0x50565955	24?	YCbCr 4:2:2 extended precision 10-bits per component in U0Y0V0Y1 order. Registered by Rich Ehlers of Evans & Sutherland. (Awaiting confirmation of component packing structure)
UYVY	0x59565955	16	YUV 4:2:2 (Y sample at every pixel, U and V sampled at every second pixel horizontally on each line). A macropixel contains 2 pixels in 1 u_int32.
V210	0x30313256	32	10-bit 4:2:2 YCrCb equivalent to the Quicktime format of the same name.
V422	0x32323456	16	I am told that this is an upside down version of UYVY.
V655	0x35353656	16?	16 bit YUV 4:2:2 format registered by Vitec Multimedia. I have no information on the component ordering or packing.
VYUY	0x59555956	?	ATI Packed YUV Data (format unknown but you can get hold of a codec supporting it here)
Y422	0x32323459	16	Direct copy of UYVY as used by ADS Technologies Pyro WebCam firewire camera.
YUY2	0x32595559	16	YUV 4:2:2 as for UYVY but with different component ordering within the u_int32 macropixel.
YUYV	0x56595559	16	Duplicate of YUY2
YUNV	0x564E5559	16	A direct copy of YUY2 registered by NVidia to work around problems in some old codecs which did not like hardware which offered more than 2 YUY2 surfaces.
YVYU	0x55595659	16	YUV 4:2:2 as for UYVY but with different component ordering within the u_int32 macropixel.
Y41P	0x50313459	12	YUV 4:1:1 (Y sample at every pixel, U and V sampled at every fourth pixel horizontally on each line). A macropixel contains 8 pixels in 3 u_int32s.
Y411	0x31313459	12	YUV 4:1:1 with a packed, 6 byte/4 pixel macroblock structure.
Y211	0x31313259	8	Packed YUV format with Y sampled at every second pixel across each line and U and V sampled at every fourth pixel.
Y41T	0x54313459	12	Format as for Y41P but the lsb of each Y component is used to signal pixel transparency .
Y42T	0x54323459	16	Format as for UYVY but the lsb of each Y component is used to signal pixel transparency .
YUVP	0x50565559	24?	YCbCr 4:2:2 extended precision 10-bits per component in Y0U0Y1V0 order. Registered by Rich Ehlers of Evans & Sutherland.
Y800	0x30303859	8	Simple, single Y plane for monochrome images.
Y8	0x20203859	8	Duplicate of Y800 as far as I can see.
Y16	0x20363159	16	16-bit uncompressed greyscale image.

AYUV

This is a 4:4:4 YUV format with 8 bit samples for each component along with an 8 bit alpha blend value per pixel. Component ordering is A Y U V (as the name suggests).

UYVY (and Y422 and UYNV and HDYC)

UYVY is probably the most popular of the various YUV 4:2:2 formats. It is output as the format of choice by the Radius Cinepak codec and is often the second choice of software MPEG codecs after YV12.

Y422 and UYNV appear to be direct equivalents to the original UYVY.

HDYC is equivalent in layout but pixels are described using the BT709 color space as used in HD video systems rather than the BT470 SD video color space typically used. Apparently there is a description in the DeckLink DirectShow SDK documentation at <http://blackmagic-design.com/support/software/archive/>, find **DeckLink SDK 5.6.2 for Windows XP** and download <http://blackmagic-design.com/support/software/register.asp?softID=108>, set product to None, serial no is not required), see "Video Formats" section.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	1
U Sample Period	2	1

Effective bits per pixel : 16

Positive biHeight implies top-down image (top line first)



IUYV

IUYV is basically the same as UYVY with the exception that the data is interlaced. Lines are ordered 0,2,4,....,1,3,5.... instead of 0,1,2,3,4,5,....

cyuv

This FOURCC, allegedly registered by Creative Labs, is essentially a duplicate of UYVY. The only difference is that the image is flipped vertically, the first u_int16 in the buffer representing the bottom line of the viewed image. Note that the FOURCC is comprised of lower case characters (so much for the upper case convention !)

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	1
U Sample Period	2	1

Effective bits per pixel : 16

Positive biHeight implies bottom-up image (bottom line first)



YUY2 (and YUNV and V422 and YUYV)

YUY2 is another in the family of YUV 4:2:2 formats and appears to be used by all the same codecs as UYVY.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	1
U Sample Period	2	1

Effective bits per pixel : 16

Positive biHeight implies top-down image (top line first)



There is a [help page here](#) which contains information on playing AVIs which include video stored in YUY2 format.

YVYU

Despite being a simple byte ordering change from YUY2 or UYVY, YVYU seems to be seen somewhat less often than the other two formats defined above.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	1
U Sample Period	2	1

Effective bits per pixel : 16

Positive biHeight implies top-down image (top line first)



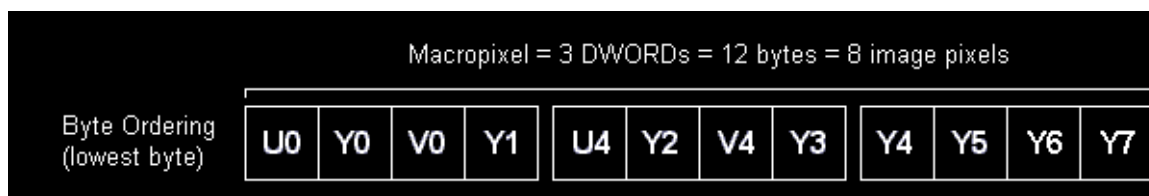
Y41P

This YUV 4:1:1 format is registered as a PCI standard format. Mediamatics MPEG 1 engine is the only codec (other than a Brooktree internal one) that I know of that can generate it.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	1
U Sample Period	4	1

Effective bits per pixel : 12

Positive biHeight implies top-down image (top line first)



Y411

I was originally told that this was a duplicate of [Y41P](#) however it seems that this is not the case after all. Y411 is a packed YUV 4:1:1 format with a 6 pixel macroblock structure containing 4 pixels. Component packing order is:

U2 Y0 Y1 V2 Y2 Y3

I have not been able to find 100% confirmation of the position for the U and V samples. I suspect that the chroma samples are probably both taken at the position of Y2 but this is a guess just now.

I have recently been informed that this format is identical to [IYU1](#).

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	1
U Sample Period	4	1

Effective bits per pixel : 12

Positive biHeight implies top-down image (top line first)

IY41

IY41 is basically the same as Y41P with the exception that the data is interlaced. Lines are ordered 0,2,4,....,1,3,5.... instead of 0,1,2,3,4,5,....

Y211

I have yet to find anything that will output Y211 ! The format looks very much like the missing YUV 4:2:2 ordering but Y samples are only taken on every second pixel. Think of it as a half width 4:2:2 image and double the width on display.

	Horizontal	Vertical
Y Sample Period	2	1
V Sample Period	4	1
U Sample Period	4	1

Effective bits per pixel : 8

Positive biHeight implies top-down image (top line first)



Y41T

This format is identical to [Y41P](#) except for the fact that the least significant bit of each Y component forms a chromakey channel. If this bit is set, the YUV image pixel is displayed, if cleared, the pixel is transparent (and the underlying graphics pixel is shown).

Positive biHeight implies top-down image (top line first)

Y42T

This format is identical to [UYVY](#) except for the fact that the least significant bit of each Y component forms a chromakey channel. If this bit is set, the YUV image pixel is displayed, if cleared, the pixel is transparent (and the underlying graphics pixel is shown).

Positive biHeight implies top-down image (top line first)

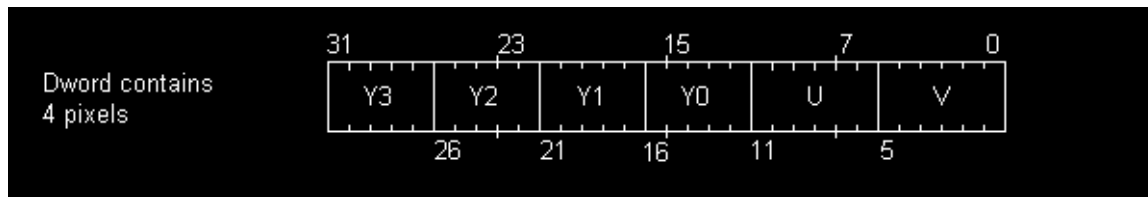
CLJR

Cirrus Logic's format packs 4 pixel samples into a single u_int32 by sacrificing precision on each sample. Y samples are truncated to 5 bits each, U and V have 6 bits per sample.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	1
U Sample Period	4	1

Effective bits per pixel : 8

Positive biHeight implies top-down image (top line first)



IYU1

The IYU1 format is a 12 bit format used in mode 2 of the IEEE 1394 Digital Camera 1.04 spec ("1394-based Digital Camera Specification, Version 1.04, August 9, 1996", page 14.). The format, a duplicate of [Y411](#), is YUV (4:1:1) according to the following pattern:

Byte	0	1	2	3	4	5
Sample	$U_{(K+0)}$	$Y_{(K+0)}$	$Y_{(K+1)}$	$V_{(K+0)}$	$Y_{(K+2)}$	$Y_{(K+3)}$

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	1
U Sample Period	4	1

IYU2

The IYU2 format is a 24 bit format used in mode 0 of the IEEE 1394 Digital Camera 1.04 spec (ibid.) The format is YUV (4:4:4) according to the following pattern:

Byte	0	1	2	3	4	5
Sample	$U_{(K+0)}$	$Y_{(K+0)}$	$V_{(K+0)}$	$U_{(K+1)}$	$Y_{(K+1)}$	$V_{(K+1)}$

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	1	1
U Sample Period	1	1

YUVP

This is another format similar to YUY2 and it's aliases. The difference here is that each Y, U and V samples is 10 bits rather than 8. I am still waiting to hear how the samples are packed - is a macropixel just 5 bytes long with all the samples packed together or is there more to it than this?

V210

[AJA Video Systems](#) have implemented this Quicktime format for Windows. It is a 10 bit per component, YCrCb 4:2:2 format in which samples for 5 pixels are packed into 4 4-byte little endian words. Rather than repeat the details here, I suggest looking at the [original definition on the Quicktime web site](#).

Supposedly there are images described as "YUV10" that are formatted similarly to this aside from the bit ordering (the correspondent mentioned having to run ntohs on the pixel data to reformat from YUV10 to V210. Despite 20 years of C, I've not heard of ntohs but I suspect it performs big-endian to little-endian conversion).

Planar YUV Formats

Label	FOURCC in Hex	Bits per pixel	Description
YVU9	0x39555659	9	8 bit Y plane followed by 8 bit 4x4 subsampled V and U planes. Registered by Intel.
YUV9	0x39565559	9?	Registered by Intel., this is the format used internally by Indeo video code
IF09	0x39304649	9.5	As YVU9 but an additional 4x4 subsampled plane is appended containing delta information relative to the last frame. (Bpp is reported as 9)
YV16	0x36315659	16	8 bit Y plane followed by 8 bit 2x1 subsampled V and U planes.
YV12	0x32315659	12	8 bit Y plane followed by 8 bit 2x2 subsampled V and U planes.
I420	0x30323449	12	8 bit Y plane followed by 8 bit 2x2 subsampled U and V planes.
IYUV	0x56555949	12	Duplicate FOURCC, identical to I420.
NV12	0x3231564E	12	8-bit Y plane followed by an interleaved U/V plane with 2x2 subsampling
NV21	0x3132564E	12	As NV12 with U and V reversed in the interleaved plane
IMC1	0x31434D49	12	As YV12 except the U and V planes each have the same stride as the Y plane
IMC2	0x32434D49	12	Similar to IMC1 except that the U and V lines are interleaved at half stride boundaries
IMC3	0x33434D49	12	As IMC1 except that U and V are swapped
IMC4	0x34434D49	12	As IMC2 except that U and V are swapped
CLPL	0x4C504C43	12	Format similar to YV12 but including a level of indirection.
Y41B	0x42313459	12?	Weitek format listed as "YUV 4:1:1 planar". I have no other information on this format.
Y42B	0x42323459	16?	Weitek format listed as "YUV 4:2:2 planar". I have no other information on this format.
Y800	0x30303859	8	Simple, single Y plane for monochrome images.
Y8	0x20203859	8	Duplicate of Y800 as far as I can see.
CXY1	0x31595843	12	Awaiting clarification of format.
CXY2	0x32595842	16	Awaiting clarification of format.

YVU9

This format dates back to the days of the ActionMedia II adapter and comprises an NxN plane of Y samples, 8 bits each, followed by (N/4)x(N/4) V and U planes.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	4
U Sample Period	4	4

Y(0,0)			Y(n-1,0)	V(0,0)		V(n-5,0)
Y(0,1)						
				V(0,m-5)		V(n-5,m-5)
Y(0,m-1)			Y(n-1,m-1)	V(0,m-1)		V(n-5,m-1)
U(0,0)			U(n-5,0)			
U(0,m-5)			U(n-5,m-5)			
U(0,m-1)			U(n-5,m-1)			

Positive biHeight implies top-down image (top line first)

ATI has a codec supporting this format that you can download from [here](#).

YUV9

[Intel's web site](#) states that YUV9 is "the color encoding scheme used in Indeo video technology. The YUV9 format stores information in 4x4 pixel blocks. Sixteen bytes of luminance are stored for every 1 byte of chrominance. For example, a 640x480 image will have 307,200 bytes of luminance and 19,200 bytes of chrominance." This sounds exactly the same as [YVU9](#) to me. Anyone know if there is any difference?

IF09

A derivative of YVU9, IF09 contains the basic 3 planes for Y, V and U followed by an additional (N/4)x(N/4) plane of "skip blocks". This final plane forms a basic delta encoding scheme which can be used by a displayer to decide which pixels in the image are unchanged from the previous displayed frame. The strange number of bits per pixel listed for the format results from the fact that an NxN image is described using $N^2 + 3(N/4)^2$ bytes.

This format is generated by Intel's Indeo codecs though users should beware - the original 32 bit Indeo 3.2 shipped with Windows 95 and the beta levels of Indeo 4.1 contain bugs which cause them to generate protection faults when using IF09. Fixed versions of these codecs are available from Intel.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	4	4

U Sample Period	4	4
------------------------	---	---

Y(0,0)			Y(n-1,0)	V(0,0)		V(n-5,0)
Y(0,1)						
				V(0,m-5)		V(n-5,m-5)
				V(0,m-1)		V(n-5,m-1)
Y(0,m-1)			Y(n-1,m-1)			
U(0,0)			U(n-5,0)			
U(0,m-5)			U(n-5,m-5)			
U(0,m-1)			U(n-5,m-1)			

Positive biHeight implies top-down image (top line first)

Delta plane definition

To be completed...

YV12

This is the format of choice for many software MPEG codecs. It comprises an NxM Y plane followed by (N/2)x(M/2) V and U planes.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	2
U Sample Period	2	2

Y(0,0)			Y(n-1,0)	V(0,0)		V(n-1,0)
Y(0,1)						
				V(0,m-3)		V(n-1,m-3)
				V(0,m-1)		V(n-1,m-1)
Y(0,m-1)			Y(n-1,m-1)			
U(0,0)			U(n-1,0)			
U(0,m-3)			U(n-1,m-3)			
U(0,m-1)			U(n-1,m-1)			

Positive biHeight implies top-down image (top line first)

ATI says they have [a download which contains this codec](#) but I can't find it on their site. If you would like something similar for Quicktime, [try here](#).

YV16

This format is basically a version of [YV12](#) with higher chroma resolution. It comprises an NxM Y plane followed by (N/2)xM U and V planes.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	1
U Sample Period	2	1

IYUV and I420

These formats are identical to YV12 except that the U and V plane order is reversed. They comprise an NxN Y plane followed by (N/2)x(N/2) U and V planes. Full marks to Intel for registering the same format twice and full marks to Microsoft for not picking up on this and rejecting the second registration.

(Note: There is some confusion over these formats thanks to the definitions on [Microsoft's site](#) which tend to suggest that the two FOURCCs are different. One is described as a 4:2:0 format while the other is described as 4:1:1. Later, however, the same page states that YV12 is the same as both of these with the U and V plane order reversed. I would consider 4:2:0 to imply 1 chroma sample for every 2x2 luma block and 4:1:1 to imply 1 chroma sample for every 4x1 luma block but it seems as if the Microsoft writer may have been using the terms interchangeably. If you know these formats, please could you [let me know](#) whether the definition here is correct or whether I need to update one or other?)

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	2
U Sample Period	2	2

Y(0,0)				Y(n-1,0)	U(0,0)			U(n-1,0)
Y(0,1)								
					U(0,m-3)			U(n-1,m-3)
Y(0,m-1)				Y(n-1,m-1)	U(0,m-1)			U(n-1,m-1)
V(0,0)				V(n-1,0)				
V(0,m-3)				V(n-1,m-3)				
V(0,m-1)				V(n-1,m-1)				

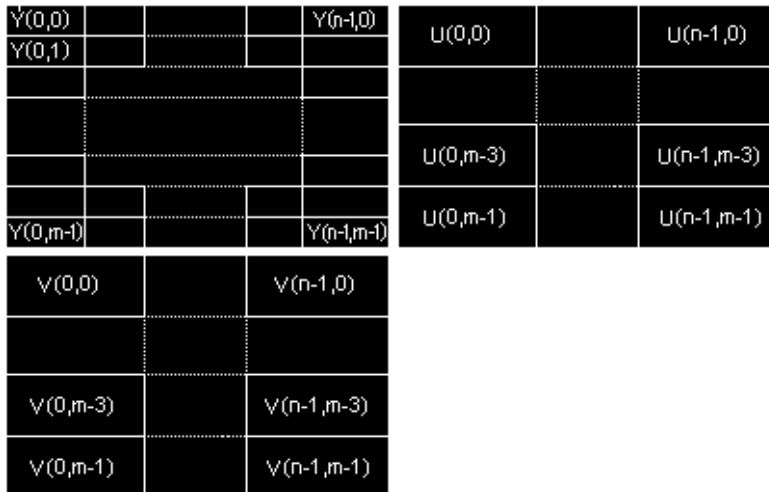
Positive biHeight implies top-down image (top line first)

CLPL

This format introduces an extra level of indirection in the process of accessing YUV pixels in

the surface. Locking the DirectDraw or DCI CLPL surface returns a pointer which itself points to three other pointers. These pointers respectively point to an $N \times N$ Y plane, an $(N/2) \times (N/2)$ U plane and an $(N/2) \times (N/2)$ V plane. The Y plane pointer retrieved is (allegedly) valid even when the surface is subsequently unlocked but the U and V pointers can only be used with a lock held (as you should be doing anyway if adhering to the DirectDraw/DCI spec).

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	2	2
U Sample Period	2	2

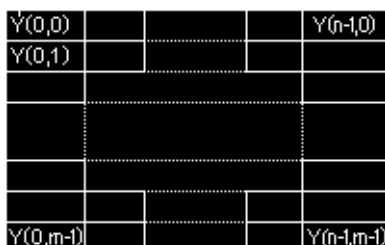


Positive biHeight implies top-down image (top line first)

Y800

This format contains only a single, 8 bit Y plane for monochrome images. Apparent duplicate FOURCCs are "Y8" and "GREY".

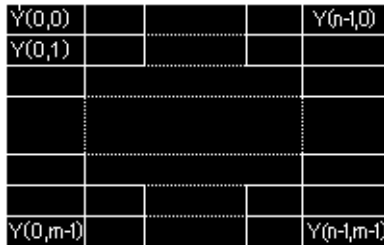
	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	N/A	N/A
U Sample Period	N/A	N/A



Y16

This format contains only a single, 16 bit Y plane for monochrome images. Each pixel is represented by a 16 bit, little endian luminance sample.

	Horizontal	Vertical
Y Sample Period	1	1
V Sample Period	N/A	N/A
U Sample Period	N/A	N/A



NV12

YUV 4:2:0 image with a plane of 8 bit Y samples followed by an interleaved U/V plane containing 8 bit 2x2 subsampled colour difference samples.

	Horizontal	Vertical
Y Sample Period	1	1
V (Cr) Sample Period	2	2
U (Cb) Sample Period	2	2

Microsoft defines this format as follows:

"A format in which all Y samples are found first in memory as an array of unsigned char with an even number of lines (possibly with a larger stride for memory alignment), followed immediately by an array of unsigned char containing interleaved Cb and Cr samples (such that if addressed as a little-endian WORD type, Cb would be in the LSBs and Cr would be in the MSBs) with the same total stride as the Y samples. This is the preferred 4:2:0 pixel format."

NV21

YUV 4:2:0 image with a plane of 8 bit Y samples followed by an interleaved V/U plane containing 8 bit 2x2 subsampled chroma samples. The same as [NV12](#) except the interleave order of U and V is reversed.

	Horizontal	Vertical
Y Sample Period	1	1
V (Cr) Sample Period	2	2
U (Cb) Sample Period	2	2

Microsoft defines this format as follows:

"The same as [NV12](#), except that Cb and Cr samples are swapped so that the chroma array of unsigned char would have Cr followed by Cb for each sample (such that if addressed as a little-endian WORD type, Cr would be in the LSBs

and Cb would be in the MSBs)."

IMC1

Similar to [YV12](#), this format comprises an $N \times N$ Y plane followed by $(N/2) \times (N/2)$ U and V planes. The U and V planes have the same stride as the Y plane and are restricted to start on 16 line boundaries.

	Horizontal	Vertical
Y Sample Period	1	1
V (Cr) Sample Period	2	2
U (Cb) Sample Period	2	2



Microsoft defines this format as follows:

"The same as [YV12](#), except that the stride of the Cb and Cr planes is the same as the stride in the Y plane. The Cb and Cr planes are also restricted to fall on memory boundaries that are a multiple of 16 lines (a restriction that has no effect on usage for the standard formats, since the standards all use 16×16 macroblocks)."

IMC2

Similar to [YV12](#), this format comprises an $N \times N$ Y plane followed by "rectangularly adjacent" $(N/2) \times (N/2)$ U and V planes. Lines of U and V pixels are interleaved at half stride boundaries below the Y plane.

	Horizontal	Vertical
Y Sample Period	1	1
V (Cr) Sample Period	2	2
U (Cb) Sample Period	2	2



Microsoft defines this format as follows:

"The same as [IMC1](#), except that Cb and Cr lines are interleaved at half-stride boundaries. In other words, each full-stride line in the chrominance area starts with a line of Cr, followed by a line of Cb that starts at the next half-stride boundary. (This is a more address-space-efficient format than [IMC1](#), cutting the chrominance address space in half, and thus cutting the total address space by 25%.) This runs a close second in preference relative to [NV12](#), but [NV12](#) appears to be more popular."

IMC3

The same as [IMC1](#) except for swapping the U and V order.

IMC4

The same as [IMC2](#) except for swapping the U and V order.

CXY1

Planar YUV 4:1:1 format registered by Conexant. Awaiting clarification of pixel component ordering.

CXY2

Planar YUV 4:2:2 format registered by Conexant. Awaiting clarification of pixel component ordering.

Google™

Search

☐ Web ☒ www.fourcc.org

*Page was last modified on: October 09 2007
22:19:17.*