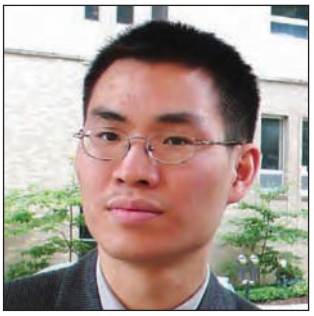
Toward Better Chroma Subsampling

By Glenn Chan

Recipient of the 2007 SMPTE Student Paper Award



Chroma subsampling is a lossy process often compounded by concatenation of dissimilar techniques. This paper surveys common subsampling applications providing examples of the advantages and disadvantages of various approaches. It also outlines a novel chroma upsampling technique that minimizes erroneous out-of-gamut colors.

Chroma subsampling is a widely used technique to reduce bandwidth in many video systems. Since the human visual system has greater acuity for luminance than color, it can be useful to reduce color resolution to lower bandwidth. Video systems approximate this via chroma subsampling. Unfortunately, chroma subsampling is not visually lossless in all situations. This paper examines sources of chroma subsampling artifacts and shows what chroma subsampling would look like if these problems were solved.

Examples throughout this paper are for 4X horizontal subsampling, which corresponds to the 4:1:1 chroma subsampling scheme used in NTSC DV. The chroma subsampling artifacts presented in this paper generalizes to other schemes, including 4:2:2 signals used in studio video (2X horizontal subsampling).

How Chroma Subsampling Works

The video signal is divided into luma and chroma; luma approximates the black-and-white portion of a signal, and chroma approximates color. Luma (Y′) is formed by the formula:

$$Y' = rR' + gG' + bB'$$

where the lowercase letters represent the *luma coefficients*. Note that the luma coefficients are different between ITU-R Rec. BT.601 and 709,[1] and SMPTE 240M.[2] For Rec. 709 video, the formula is as follows:

$$\text{Rec. 709 Luma } (Y') = 0.2126\ R' + 0.7152\ G' + 0.0722\ B'$$

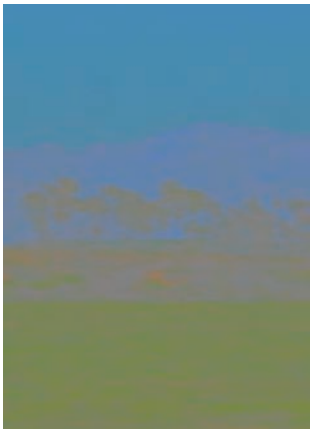
Color information is coded by subtracting Luma (Y′) from the red (R′) and blue (B′) color components to form color difference signals R′-Y′ and B′-Y′. These color difference components may have scale factors and offsets applied to them so that they can be stored or carried over video interfaces. These scale factors and offsets are reversed upon decoding.

In **Fig. 1**, luma is visualized by blanking the color difference components with neutral values. Similarly, the color difference components can be visualized by blanking the luma values.

Figure 1.

(a) Original

(b) Luma (Y′) channel only.



(c) Color difference components.

(d) Subsampled image.

Once the image is converted in this manner, the color difference components can be subsampled (i.e., reduced in resolution) to reduce bandwidth. For the majority of real-world images, chroma subsampling is visually lossless.

Problem: Out-of-Range/Gamut Colors

Chroma subsampling is not visually lossless in cases where it creates colors that cannot be reproduced by a display device (i.e., outside its gamut).

If the image consists of alternating red and black lines, all the reconstructed values will have the same chroma value (assuming typical chroma resampling). The problem here is that chroma is reconstructed onto "black" pixels (pixels where Y′ is at black level), as seen in **Fig. 2**. Logically, it is known that black pixels should not emit red, green, or blue light. However, the reconstructed black pixels have positive red, negative green, and negative blue values. Real-world monitors cannot emit negative light. The negative values are effectively clipped to zero by the monitor. So, the resulting black pixel is a reddish one that emits red light (and no green or blue light). This is clearly erroneous. A side effect of clipping is that the resulting red pixel has an effective luma value that is greater than zero. It is brighter than it should be. Similarly, the same problem occurs for white pixels. Chroma reconstructed onto white pixels can cause the red/green/blue channels to go too high and clip (or distort, as in the case of CRTs).

Non-constant Luminance

The second problem with chroma subsampling is that it does not maintain constant luminance. The luma values used in chroma subsampling are an engineering shortcut used to approximate luminance,[3] calculated with the following shortcut formula:

Rec. 709 Luma (Y′) = 0.2126 R′ + 0.7152 G′ + 0.0722 B′

To calculate luminance instead of luma, linear light processing is necessary. It is desirable to ensure that the number of photons of light emitted by the monitor stay roughly the same. To do this, the video signal can be converted into a linear light signal by removing its gamma correction. The calculations are then performed on the linear light signal, and then gamma correction is added back to the signal.

Gamma correction can be removed by applying the inverse of the Rec. 709 transfer function according to the following formula, where L is the linear-light value and E´ is the (nonlinear) gamma-corrected component:

$$L = \begin{cases} \dfrac{E'_{709}}{4.5} & E'_{709} \leqslant 0.081 \\ \left(\dfrac{E'_{709} + 0.099}{1.099}\right)^{\frac{1}{0.45}} & ,0.081 \leqslant E'_{709} \end{cases}$$

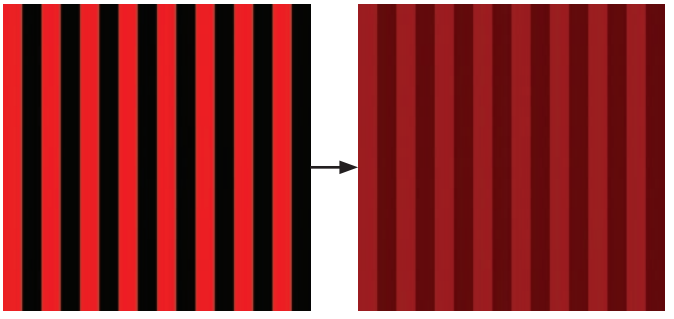
Then calculate luminance

Rec. 709 Luminance (Y) = 0.2126 R + 0.7152 G + 0.0722 B

Gamma correction can then be added back to the signal (the formula is the inverse of the previous formula).

In typical gamma-corrected processing, errors in chroma will bleed into the luminance channel. Not enough chroma will result in a drop in luminance, causing dark bands to appear (**Fig. 3**). Similarly, too much chroma will cause a rise in luminance. This effect is proportional to chroma strength; it is worst where there are fully saturated colors. In practical situations, real-world footage does not usually contain highly-saturated colors so these errors do not normally appear. Linear light processing solves the problem of chroma errors bleeding into the luminance channels and gets rid of the dark bands. For this to happen, linear light processing has to be used both in (1) forming luminance and (2) in resampling/re-scaling the chroma.

Figure 2. Calculation of resulting R´G´B´ values.



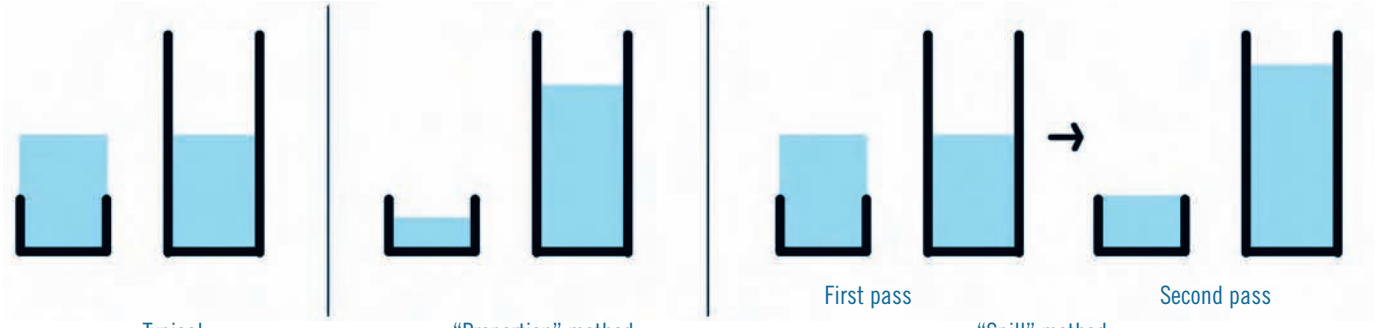
Original values		
Y′	0.2126	0
B′-Y′	-0.2126	0
R′-Y′	0.7874	0
Subsampled values		
Y′	0.2126	0
B′-Y′	-0.1063	-
R′-Y′	0.3937	-
RGB values		
R′	0.6063	0.3937
G′	0.1063	-0.1063
B′	0.1063	-0.1063

*Rec. 709 luma co-efficients are used, with values scaled for a 0-1 range.

Figure 3. Comparison of typical processing (with gamma-corrected values) versus linear light processing.

Original | Typical processing | Linear light processing

Figure 4. Diagram of chroma distribution methods.



Typical | "Proportion" method | "Spill" method

Solving the Out-of-Gamut Problem

Better results can be achieved by using the luma information to aid in reconstructing chroma information. The chroma can be distributed in such a way that it minimizes out-of-gamut colors. The author refers to this as in-range chroma reconstruction.

Picture chroma as a liquid being poured into glasses of different heights as shown in **Fig. 4**. Let the height of the glass represent the most chroma a particular pixel can hold. If too much chroma is in the glass, it will overflow and an out-of-gamut value will result. In typical chroma subsampling, the same amount of chroma is poured into each glass. If the glasses are of different heights (e.g., black pixels essentially have no height), overflow can occur. One algorithm for avoiding this problem is to distribute/pour the chroma proportional to the height of the glass. This algorithm will be referred to as the "proportion" method. A second possible algorithm is to collect any spilled chroma and repour them into remaining unfilled glasses. This is the "spill" method.

To determine the maximum chroma a pixel can hold, visualize the R´G´B´ gamut plotted in Y´, B´-Y´, and R´-Y´ coordinates in **Fig. 5**. Each B´-Y´ R´-Y´ pair corresponds to a particular color/hue and lies along a triangular slice within the R´G´B´ gamut. This triangle has corners at white, black, and some fully-saturated/pure R´G´B´ color as shown in **Fig. 5**. The height h in the figure represents the maximum chroma possible for a given Y´ value. This height *h* also corresponds to the height of the glasses in the chroma-pouring analogy.[4]

Results of in-range chroma reconstruction can be seen in **Fig. 6**. For red text on a white (or black) background, both the proportion and spill methods can achieve excellent results. For the darker red text on a grayish background, it is possible to see the differences between the two algorithms. The proportion method can exhibit some erroneous hotspots

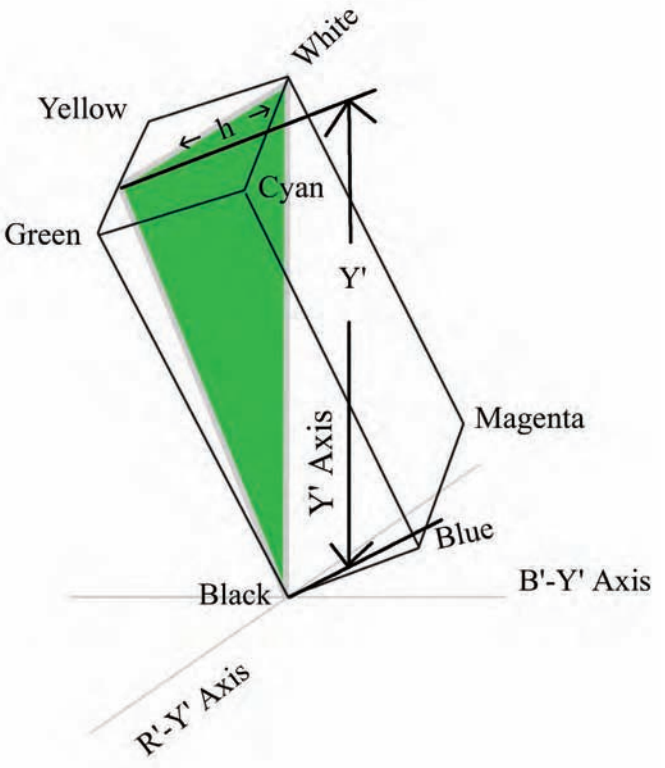


Figure 5. Visualization of how the maximum chroma for each pixel (height *h*) is determined.

of concentrated chroma, one of which can be seen near the center of the large A in the text in the dim test pattern. The spill method is not prone to such artifacts. However, it is slower because it requires a few passes repouring the spilled chroma instead of the single pass of the proportion method.

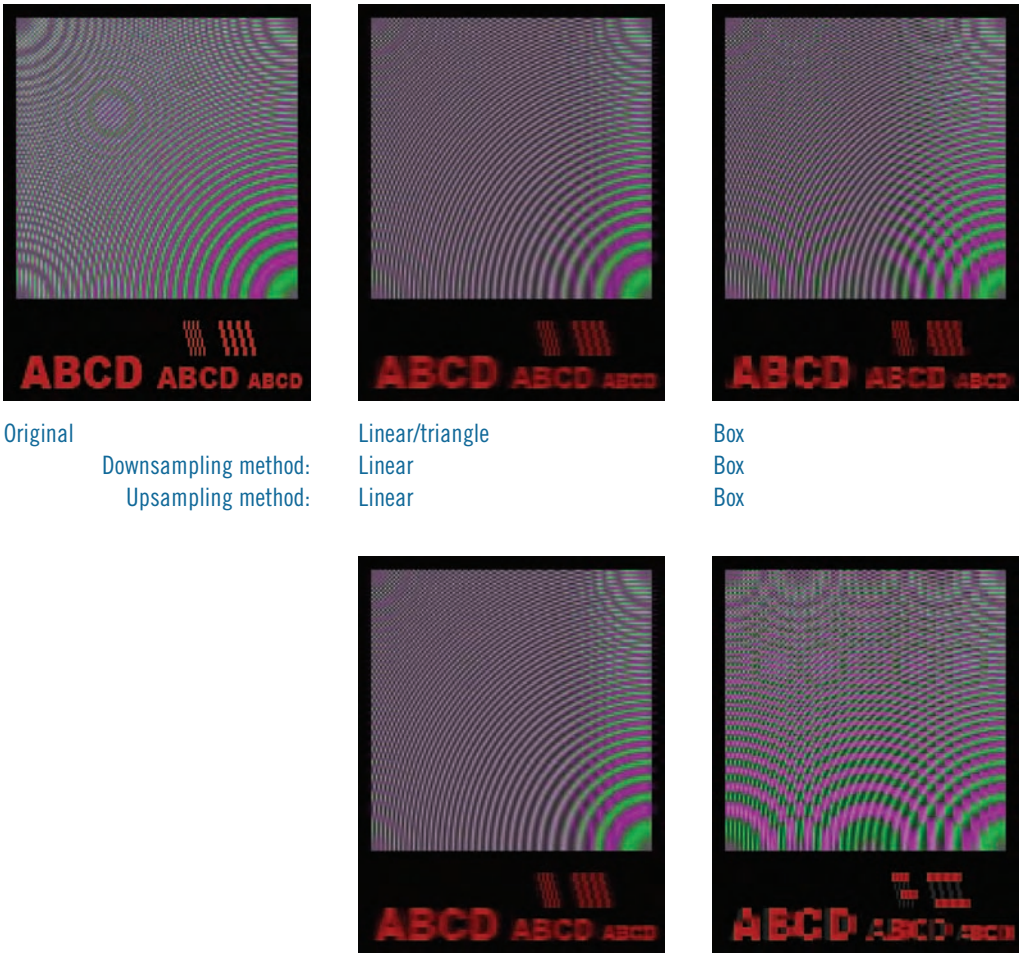
Practical Problems with In-Range Chroma Reconstruction

An underlying assumption behind in-range chroma reconstruction is that the image lies entirely within the R´G´B´

Figure 6. Comparison between typical chroma subsampling and in-range chroma reconstruction.

Original | Typical | "Proportion" method | "Spill" method

Figure 7. Resampling schemes compared.



Original		Linear/triangle	Box
	Downsampling method:	Linear	Box
	Upsampling method:	Linear	Box
		Multitap FIR	Nearest neighbor
	Downsampling method:	Multitap FIR	Nearest neighbor/point sampling
	Upsampling method:	Linear	Box

gamut and does not contain out-of-gamut colors. This can be a bad assumption for signals in a production environment. For analog material dubbed to digital, analog black level may be incorrect. For digitally-originated material, many cameras will record information above white level in the "superwhite" region.[5] Also, all sources have noise that can push legal R′G′B′ signals out of range.

If we simply apply the spill method, anomalous chroma can occur on highlight areas (not shown). Out-of-range colors can be accommodated by changing how the heights of the glasses are determined. Recall that the original function was derived from a triangle-shaped slice of the R′G′B′ cube. One method is to move the corners of this triangle-shaped function to cover the out-of-range values. Unfor-

Figure 8. Examples of inappropriate mixing of chroma siting schemes.

Method	(a) Linear	(b) (mixed)	(c) mixed	(d) mixed
Downsampling:	Linear	Point/nearest neighbor	Box	Linear
Upsampling:	Linear	Linear	Linear	Box

tunately, doing so weakens the performance of in-range chroma reconstruction.

Resampling Methods

In chroma subsampling a mishmash of different resampling schemes are also in use. The scheme that is used makes a difference in visual quality. When downsampling, any scheme has tradeoffs between:

1. *Imperfect frequency response.* Blurry images, for example, have imperfect frequency response.
2. *Aliasing.* Image detail above the Nyquist-Shannon limit can cause spurious image detail that is seen as erroneous/spurious bands in zone plate test patterns.
3. *Ringing artifacts.* These appear as ghosting or halos around high-contrast edges.

Every resampling scheme suffers from at least one of these problems or some combination of all three. The three problems can be visualized as corners on a triangle, where improving/moving along one dimension will make either or both of the other problems worse. It is impossible to solve for all three problems at once.

However, these three forms of image impairments do not tell the whole story. Image processing in the human brain also plays a role in what looks the best. Some subjective evaluation is necessary.

Figure 7 shows a test pattern run through different resampling schemes. Each scheme actually consists of two (possibly different) schemes, one for downsampling and one for upsampling. Four common pairings are shown in **Fig. 7**.

The worst-looking schemes by far are the nearest neighbor and box resampling schemes. The nearest neighbor scheme exhibits high amounts of aliasing and is also vulnerable to a form of aliasing which the author refers to as "gap aliasing." Image detail that falls in the gaps between the sampled points are discarded and completely ignored. Gap aliasing can be seen in alternating red and black lines in a test pattern, where the chroma for some sets of lines completely disappears. On top of the aliasing artifacts, the nearest neighbor and box resampling schemes suffer a boxy appearance from box upsampling.

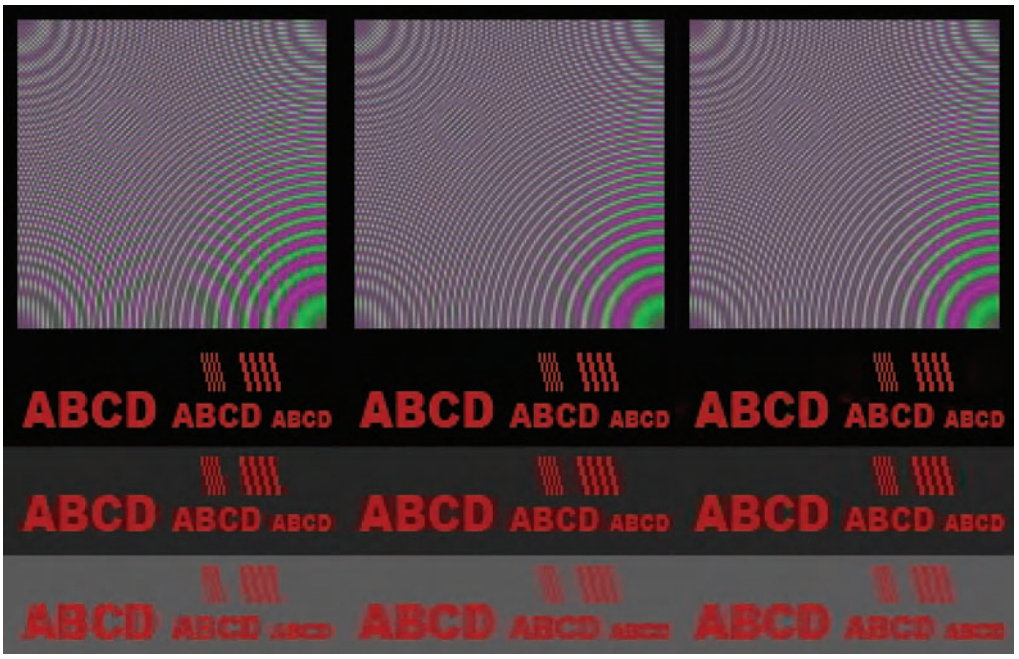
For good quality chroma subsampling, the linear/triangle or multitap finite impulse response (FIR) schemes should be used. Between these two schemes, the multitap FIR scheme is sharper and exhibits less aliasing at the expense of ringing artifacts. Rec. 601 filtering requirements and Rec. 709 filtering guidelines establish standards for filter performance. A multitap FIR filter is necessary to meet those standards.[6] This type of filter can have much better performance over multiple generations than linear/triangle resampling. However, such filters are rarely implemented in desktop applications because they are computationally expensive.

In practice, box resampling is commonly used for 4:1:1 DV despite its poor visual performance. Worse yet, using different resampling schemes for downsampling and upsampling means that different methods may be inappropriately used. This is a problem if box resampling is mixed with the linear/triangle (or multitap FIR) scheme, as shown in **Fig. 8**.

In box resampling, the chroma center lies between luma pixels. This is referred to as "interstitial siting," in this paper. In the other schemes, the chroma center lies on top of a luma pixel, which is called "co-siting." The chroma center of the linear/triangle scheme lies 1.5 pixels to the left of the chroma center of the box resampling scheme shown in **Fig. 8**. Although standards for various video formats (e.g., 4:2:2 SDI, DV, and its variants, MPEG-2, etc.) specify chroma siting,[7] these standards are not always followed. Mixing the schemes can result in the chroma being shifted in relation to the luma, as shown in **Fig. 8**. If chroma is downsampled using point sampling (i.e., nearest neighbor scheme) and upsampled with the linear/triangle scheme, the chroma center will not be shifted, but high amounts of aliasing will result. Alternatively, using the nearest neighbor scheme inherently results in chroma shifting (**Fig. 7**).

In nonlinear editing, there is a minor advantage to using box and nearest neighbor resampling because they effectively

Figure 9. Comparison of resampling schemes in ideal chroma/color subsampling.



(a) Box resampling. (b) Linear resampling. (c) Multitap FIR resampling.

pass the chroma straight through. Unlike the linear and multitap FIR schemes, there is no generation loss. Suppose the linear/triangle scheme were used? If there is a cross dissolve between two clips, most nonlinear editing systems (NLEs) will only recompress the cross dissolved section. The cross dissolve will encounter generation loss, whereas the material around it will not. At the start of the cross dissolve, a noticeable jump can occur between first- and second-generation material.

This problem could be solved by recompressing (and reapplying chroma subsampling) on all the material involved in the dissolve. This would mean that adding a 1-sec cross dissolve to an hour-long clip requires that the entire clip be recompressed. Theoretically, this is not a problem if the NLE is able to recompress the footage and output in realtime without needing to render. However, not all NLEs are capable of or designed as such. In practice, most desktop-based NLEs use box or nearest neighbor resampling for chroma subsampled formats (e.g., 4:2:2 SDI, DV, MPEG-2). This approach can backfire. Using box or nearest neighbor resampling only works well for video that has already been subsampled. It does not work well for titles, CG elements, most image processing filters, still images (or still image sequences), or downconverted material. These situations can result in the inappropriate mixing of resampling schemes, as shown in **Fig. 8**. The ideal (though not necessarily practical) solution to this dilemma is to simply avoid it. Performing acquisition and post in a non-subsampled format (e.g., 4:4:4 R′G′B′) prevents generation loss issues.

Putting It Together

For ideal quality, linear light processing and in-range chroma reconstruction should be used. Determining the ideal resampling scheme should be done subjectively. **Figure 9** shows chroma subsampling done with linear light processing, the proportion method for chroma reconstruction, and different resampling schemes.

The linear/triangle resampling scheme appears to look the best. The box scheme has a somewhat boxy appearance, and the multitap FIR scheme has objectionable ringing artifacts. Unfortunately, 4X horizontal subsampling is too much to be visually lossless, even when done ideally. In all instances the red text appears noticeably blurry against the gray background. Nonetheless, the examples do show that chroma/color subsampling is capable of higher quality.

Potential Applications

Linear light processing of chroma, although not compatible with existing systems, may (or may not) be useful in future compression schemes for delivering content. However, it is not clear if the minor improvement in quality is worth the added complexity.

In-range chroma reconstruction is potentially useful when converting 4:2:2 material to 4:4:4 R′G′B′ (e.g., many image processing tasks require this) and when upconverting subsampled SD signals to HD. In post-production, chroma quality can be improved by avoiding inappropriate mixing of chroma siting and resampling schemes.

Conclusion

In practice, chroma subsampling artifacts for 4:2:2 and progressive 4:2:0 formats are rarely noticed even when poorly implemented (e.g., with nearest neighbor or box resampling).[8] Specifically, 4:2:2 is commonly referred to (and sometimes marketed) as visually lossless, even though this

is not actually the case in all circumstances (e.g., red text on a black background). However, while chroma subsampling is not entirely visually lossless, it is often unnoticed by viewers.

On the other hand, 4:1:1 and interlaced 4:2:0 formats can be problematic as they effectively subsample the chroma by 4X in one direction (interlaced 4:2:0 effectively subsamples 4X vertically, because each interlaced field is subsampled individually). As **Fig. 9** shows, 4X subsampling is too much, even if current chroma subsampling problems are fixed. In current practice, end viewers do notice the artifacts.[9] In production, saturated colors in titles can be objectionable when working with 4:1:1 DV. Moving away from interlacing removes the need for the 4:1:1 and interlaced 4:2:0 formats. This allows the more sensible progressive 4:2:0 formats to be used and allows for higher quality.

References/Notes

1. Rec. ITU-R BT. 601, "Studio encoding parameters of digital television for standard 4:3 and widescreen aspect ratios;" Rec. ITU-R BT.709-5, "Parameter values for the HDTV standards for productions and international programme exchange.
2. SMPTE 240M-1999 (Archived 2004) Television—"1125-Line High-Definition Production Systems–Signal Parameters," www.smpte.org.
3. Per SMPTE EG 28, luma and luminance have different meanings. See Poynton, Charles. "YUV and luminance considered harmful: A plea for precise terminology in video," http://poynton.com/PDFs/YUV_and_luminance_harmful.pdf.
4. For a more complete description of the algorithms and source code, please contact the author at glennchan@gmail.com.
5. Multitap FIR chroma filtering can also cause out of gamut colors. Although this behavior has been extensively tested, it does not seem overly problematic.
6. For the multitap FIR filter, the author tried to keep the filter characteristics in the spirit of ITU-R Rec. 601. The template guidelines with the passband frequency divided by 2 (because Rec. 601 defines filter performance for 2X subsampling and not 4X) were used.
7. For an overview of various standards in regards to chroma siting, see Poynton, Charles. "Merging computing with studio video: Converting between R'G'B' and 4:2:2." www.poynton.com/PDFs/Merging_RGB_and_422.pdf. Please note that the resampling schemes presented in this article represent what many NLE codecs do (for NTSC DV) and does not reflect the actual NTSC DV standard. Those implementing NTSC DV systems should refer to original standards documents.
8. A good website with comparisons of different production codecs (especially 4:2:2 codecs) can be found at www.codecs.onerivermedia.com.
9. See the discussion of the "interlaced chroma problem" in Don Munsil and Stacey Spears' article "The Chroma Upsampling Error and The 4:2:0 Interlaced Chroma Problem," http://www.hometheaterhifi.com/volume_8_2/dvd-benchmark-special-report-chroma-bug-4-2001.html.

The author of this paper is the recipient of the 2007 SMPTE Student Paper Award.

The Author

Glenn Chan graduated from the Radio and Television Arts program at Ryerson University. He is self-taught in image processing and C++ programming. His main interest is in color grading and in developing better color grading tools. Chan is currently working on color grading filters that move beyond traditional color spaces (e.g., HSL, Y'CbCr), and their shortcomings, such as the failure to maintain constant luminance. His goal is to provide more intuitive and novel ways of manipulating color to add life and depth to images.