Subject: What (or if) format of video buffer would be needed for video support?
Posted by fudadmin on Mon, 23 Jul 2007 21:02:27 GMT
View Forum Message <> Reply to Message

Just some research to feed some imagination for the future...
Quote:DVD quality MPEG2 is in YCbCr 420. Many broadcast quality mpeg2 videos are coded in YCbCr422 or even YCbCr444.

from: http://thread.gmane.org/gmane.comp.graphics.agg/2466/focus=2 486
Quote:
after various tests, I have implemented the YCbCr support in my application to my satisfaction. It turned out that it didn't make much sense to support YCbCr422 directly by a pixel format class. It is better to use a YCbCr444 buffer in main memory, rendering any vector stuff on top of that and then converting to YCbCr422 on the fly while blitting to graphics memory for use by the overlay engine. Since writing to graphics memory is slow, the additional filtering overhead is neglectable. (If bandwidth is not your concern, your graphics card overlay engine is likely to support YCbCr444 directly.) This is all much faster then trying to use YCbCr422 directly paying attention to odd and even pixels let alone trying to filter in the pixel format implementation. The pixel format implementation for YCbCr444 is virtually the same as the implementation for RGB24. The only difference is in the color representation. One could use a fake agg::ycca8 color type that uses "r g b a" as the names for "y cb cr a" so that one can directly use the rgb pixel format and image filters. In the "ycca8" color type implementation, one would convert into YCbCr color space like so:

```
inline void
rgb_to_ycbcr(value_type r, value_type g, value_type b,
             value_type& y, value_type& cb, value_type& cr)
{
   y = (8432 * r + 16425 * g + 3176 * b) / 32768 + 16;
   cb = (-4818 * r - 9527 * g + 14345 * b) / 32768 + 128;
   cr = (14345 * r - 12045 * g - 2300 * b) / 32768 + 128;
}

// ...

ycca8::ycca8(const rgba8& c) :
    a(c.a)
{
    rgb_to_ycbcr(c.r, c.g, c.b, r, g, b);
}

void ycca8::clear()
{
    y = 16;
    cb = 128;
```

```
    cr = 128;
    a = 0;
}

// ... and so on
```

r, g, b and a being members of ycca8 really meaning y, cb, cr and a.

Given these findings, is there still interest in my YCbCr stuff... or do you have any comments or further insights?

Best regards,
-Stephan

So, my questions is: Would YCbCr444 (or any other) video buffer(s) needed as a starting point for U++ video support?
