

YUV sampling in Vulkan

VK_KHR_sampler_ycbcr_conversion

What is YUV?

A color model

1. What is YUV?

1. is describe colorspace that are encoded using YCbCr
2. encodes color image or video taking human perception into account
3. allows reduced bandwidth for chrominance components, compared to RGB
4. Y — Luminance component
 1. Physical linear-space brightness
5. U — blue projection
6. V — red projection

Why YUV is some kind of complicated?

Video compression

1. Why YUV is some kind of complicated?
 1. Planar: Each color component is packed in different 2D images
 2. Luma: Y refers to luminance
 3. UV(CbCr) refers to chrominance(color)
 4. Downsampled chroma. Less bandwidth on color is an easy way to save space.
 5. Different various of YUV format...
 1. How many planes? 2 or 3
 2. Which color component comes first?
 3. How many bit per component? 8-bit or 10-bit?
 4. How much is chroma downsampled? 2x?
 5. Where is the telex center for the chroma samples?
 6. What is exact color space conversion matrix from YUV to RGB?
 7. How is chroma reconstructed to full resolution?

Dealing with YUV without fancy extensions

How many formats you need to deal with

Shader variants may quickly get out hand if too many formats

```
layout(binding = 0) uniform TexLuma;
layout(binding = 1) uniform TexCb;
layout(binding = 2) uniform TexCr;

layout(location = 0) out vec3 FragColor;
layout(location = 0) in vec2 TexCoord;

const mat3 yuv_to_rgb_matrix = mat3(...);

void main()
{
    float Luma = textureLod(TexLuma, TexCoord, 0.0).x;
    float Cb = textureLod(TexCb, TexCoord, 0.0).x; // For mid-point chroma
    float Cr = textureLod(TexCr, TexCoord, 0.0).x;
    vec3 yuv = vec3(Luma, Cb, Cr);
    // Possibly expand range here if using TV YUV range and not PC YUV range.
    yuv = rescale_yuv(yuv);
    FragColor = yuv_to_rgb_matrix * yuv;
}
```