

Using Ycbcr Samplers with Bindless Vulkan

Sergey Kosarevsky, Meta



Agenda

- ▶ What are Ycbcr samplers?
- ▶ Vulkan immutable samplers
- ▶ What is "bindless"?
- ▶ Bindless Ycbcr samplers

What are Ycbcr samplers?

- ▶ A niche and complicated feature...
- ▶ [VK_KHR_sampler_ycbcr_conversion](#)
- ▶ Now a part of Vulkan 1.1

The use of $Y' C_b C_r$ sampler conversion is an area in 3D graphics not used by most Vulkan developers. It is mainly used for processing inputs from video decoders and cameras.

This extension provides the ability to perform specified color space conversions during texture sampling operations for the $Y' C_b C_r$ color space natively. It also adds a selection of multi-planar formats, image aspect plane, and the ability to bind memory to the planes of an image collectively or separately.

https://registry.khronos.org/vulkan/specs/1.3-extensions/man/html/VK_KHR_sampler_ycbcr_conversion.html

What are Ycbcr samplers?

- ▶ Create a conversion object

```
const VkSamplerYcbcrConversionCreateInfo ci = {  
    .sType = VK_STRUCTURE_TYPE_SAMPLER_YCBCR_CONVERSION_CREATE_INFO,  
    .format = vkFormat,  
    .ycbcrModel = VK_SAMPLER_YCBCR_MODEL_CONVERSION_YCBCR_709,  
    .ycbcrRange = VK_SAMPLER_YCBCR_RANGE_ITU_FULL,  
    .components = { VK_COMPONENT_SWIZZLE_IDENTITY,  
                    VK_COMPONENT_SWIZZLE_IDENTITY,  
                    VK_COMPONENT_SWIZZLE_IDENTITY,  
                    VK_COMPONENT_SWIZZLE_IDENTITY },  
    .xChromaOffset = midpoint ? VK_CHROMA_LOCATION_MIDPOINT :  
                      VK_CHROMA_LOCATION_COSITED_EVEN,  
    .yChromaOffset = midpoint ? VK_CHROMA_LOCATION_MIDPOINT :  
                      VK_CHROMA_LOCATION_COSITED_EVEN,  
    .chromaFilter = VK_FILTER_LINEAR,  
    .forceExplicitReconstruction = VK_FALSE,  
};  
VkSamplerYcbcrConversionInfo info = {  
    .sType = VK_STRUCTURE_TYPE_SAMPLER_YCBCR_CONVERSION_INFO,  
    .pNext = nullptr,  
};  
vkCreateSamplerYcbcrConversion(vkDevice_, &ci, nullptr, &info.conversion);
```

What are Ycbcr samplers?

- ▶ Create a conversion object
- ▶ Create **VkSampler**

```
... Ycbcr conversion

VkSamplerCreateInfo cinfo = { ... };

if (yuvFormat != Format_Invalid) {
    cinfo.pNext = getOrCreateYcbcrConversionInfo(yuvFormat);
    // must be CLAMP_TO_EDGE
    cinfo.addressModeU = VK_SAMPLER_ADDRESS_MODE_CLAMP_TO_EDGE;
    cinfo.addressModeV = VK_SAMPLER_ADDRESS_MODE_CLAMP_TO_EDGE;
    cinfo.addressModeW = VK_SAMPLER_ADDRESS_MODE_CLAMP_TO_EDGE;
    cinfo.anisotropyEnable = VK_FALSE;
    cinfo.unnormalizedCoordinates = VK_FALSE;
}

VkSampler sampler = VK_NULL_HANDLE;
vkCreateSampler(vkDevice_, &cinfo, nullptr, &sampler);
```

What are Ycbcr samplers?

- ▶ Create a conversion object
- ▶ Create **VkSampler**
- ▶ Create **VkImageView**

```
... Ycbcr image view

const VkImageViewCreateInfo ci = {
    .sType = VK_STRUCTURE_TYPE_IMAGE_VIEW_CREATE_INFO,
    .pNext = ycbcr,
    .image = vkImage_,
    .viewType = type,
    .format = format,
    .components = mapping,
    .subresourceRange = {aspectMask, baseLevel, numLevels, baseLayer, numLayers},
};
VkImageView vkView = VK_NULL_HANDLE;
vkCreateImageView(device, &ci, nullptr, &vkView);
```

Vulkan immutable samplers

- ▶ Create a conversion object
- ▶ Create **VkSampler**
- ▶ Create **VkImageView**
- ▶ Add immutable samplers to **VkDescriptorSetLayoutBinding**

```
● ● ● Descriptor Set Layout

VkDescriptorSetLayoutBinding getDSLBinding(uint32_t binding,
                                            VkDescriptorType descriptorType,
                                            uint32_t descriptorCount,
                                            VkShaderStageFlags stageFlags,
                                            const VkSampler* immutableSamplers)
{
    return VkDescriptorSetLayoutBinding{
        .binding = binding,
        .descriptorType = descriptorType,
        .descriptorCount = descriptorCount,
        .stageFlags = stageFlags,
        .pImmutableSamplers = immutableSamplers,
    };
}
```

What is "bindless"?

*Bindless design is a technique that allows for efficient management of resources in modern graphics APIs. This technique **eliminates the need for binding resources like textures, buffers, and samplers to specific slots**, instead allowing the application to **access resources directly through their unique handles**.*

<https://dev.to/gasim/implementing-bindless-design-in-vulkan-34no>

Bindless Ycbcr samplers

- ▶ Make Ycbcr samplers available in GLSL...
- ▶ ...but how?

```
••• GLSL

layout (set = 0, binding = 0) uniform texture2D kTextures2D[];
layout (set = 0, binding = 1) uniform sampler kSamplers[];

layout (set = 0, binding = 3) uniform sampler2D kSamplerYUV[];
```

Bindless Ycbcr samplers

- ▶ Right?

```
••• GLSL

layout (set = 0, binding = 0) uniform texture2D kTextures2D[];
layout (set = 0, binding = 1) uniform sampler kSamplers[];

layout (set = 0, binding = 3) uniform sampler2D kSamplerYUV[];
```

```
•••
layout (push_constant) uniform {
    uint textureId;
}

void main() {
    out_FragColor = texture(kSamplerYUV[textureId], uv);
}
```

Bindless Ycbcr samplers

- ▶ Specialization constants!

```
•••  
  
layout (location=0) in vec2 uv;  
layout (location=0) out vec4 out_FragColor;  
  
layout (constant_id = 0) const uint textureId = 0;  
  
void main() {  
    out_FragColor = texture(kSamplerYUV[textureId], uv);  
}
```

Check out the demo!

YUV 420p
Press any key to change



https://github.com/corporateshark/lightweightvk/blob/master/samples/004_YUV.cpp

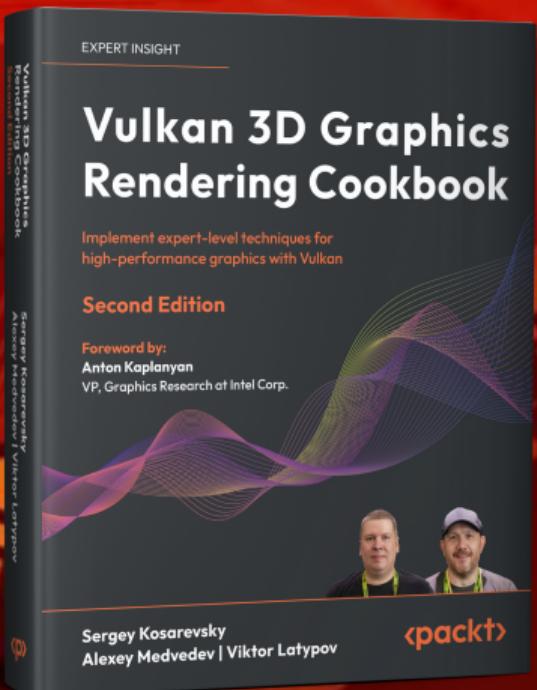
Questions?

Thank you!

Sergey Kosarevsky

<https://github.com/facebook/igl>

<https://github.com/corporateshark/lightweightvk> ← lots of bindless Vulkan stuff in this fork



Vulkan 3D Graphics Rendering Cookbook



Packt Direct



Amazon

Use the discount code VULKANF2025
on the Packt website to get 20% off

packt