

Core Image

Photo Editing in Color and Depth

SwiftAveiro – June 20, 2019

by Kalle Kabell and Tobias Due Munk

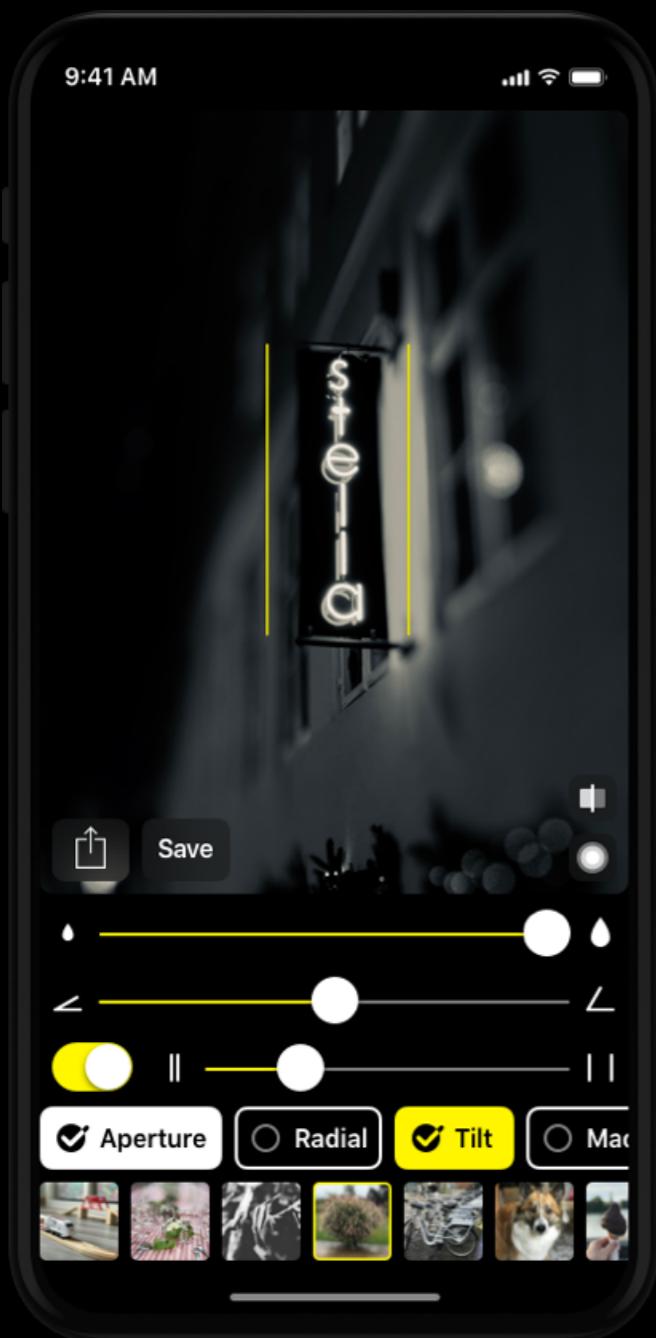
KABELL & MUNK

kabellmunk.dk

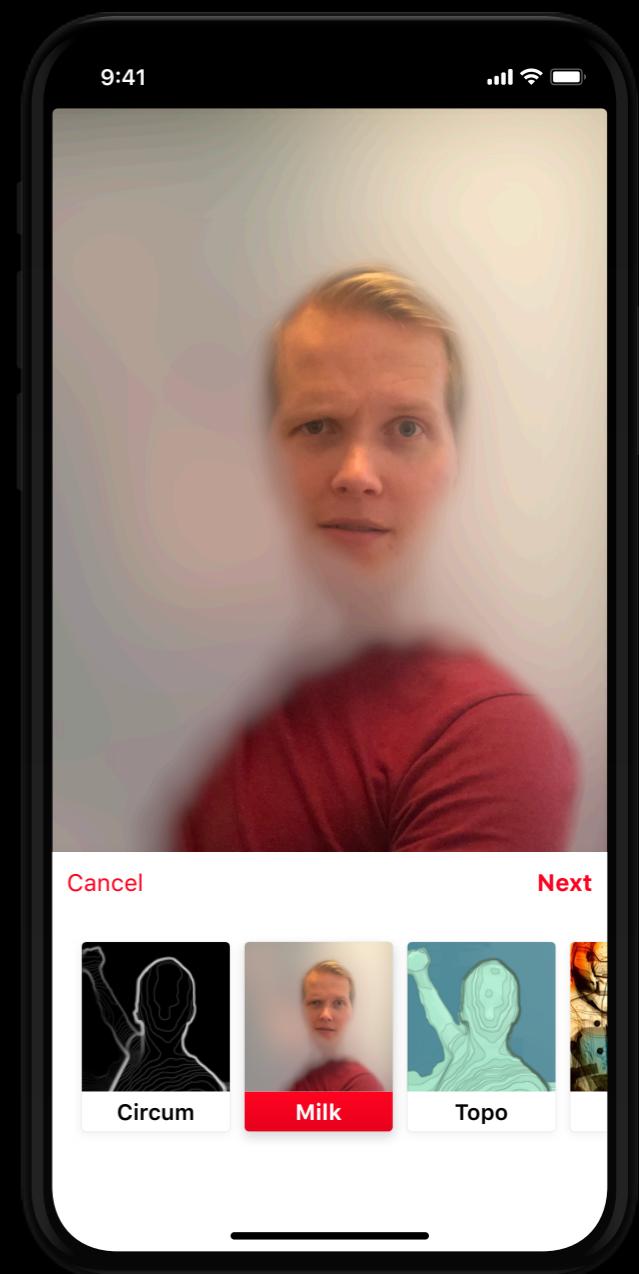
Kalle Kabell
@kkabell

Tobias Due Munk
@tobiasdm

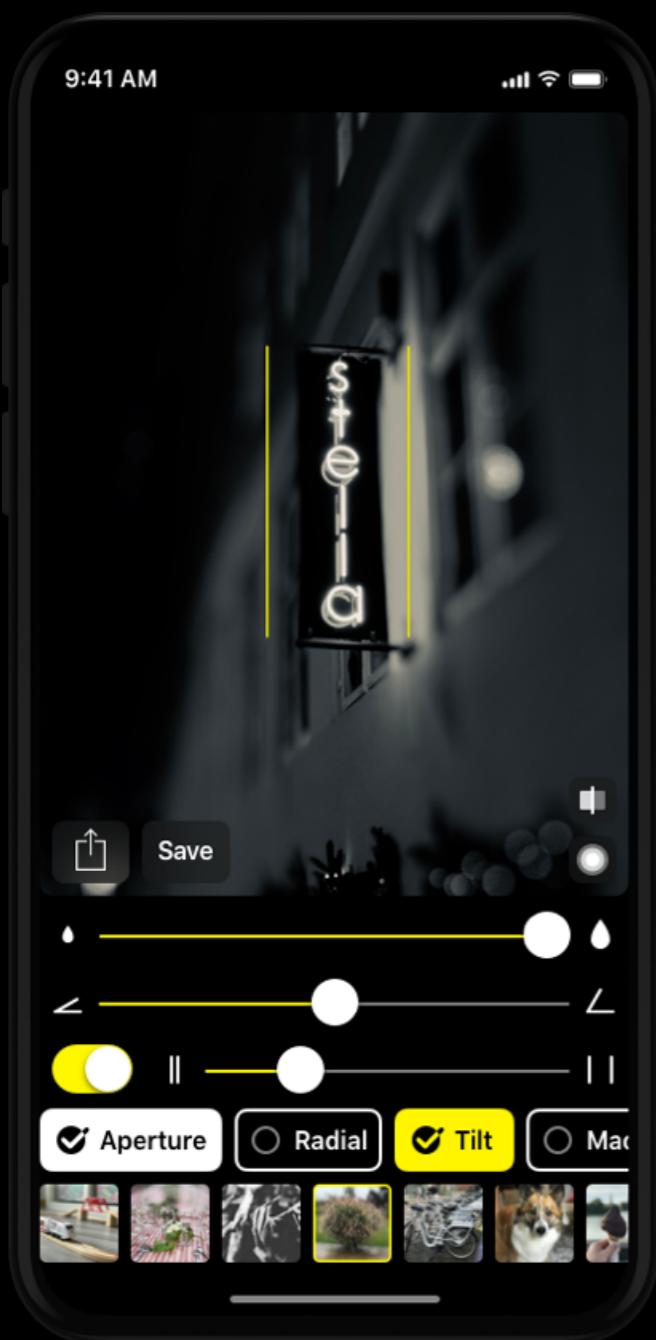
Piculet



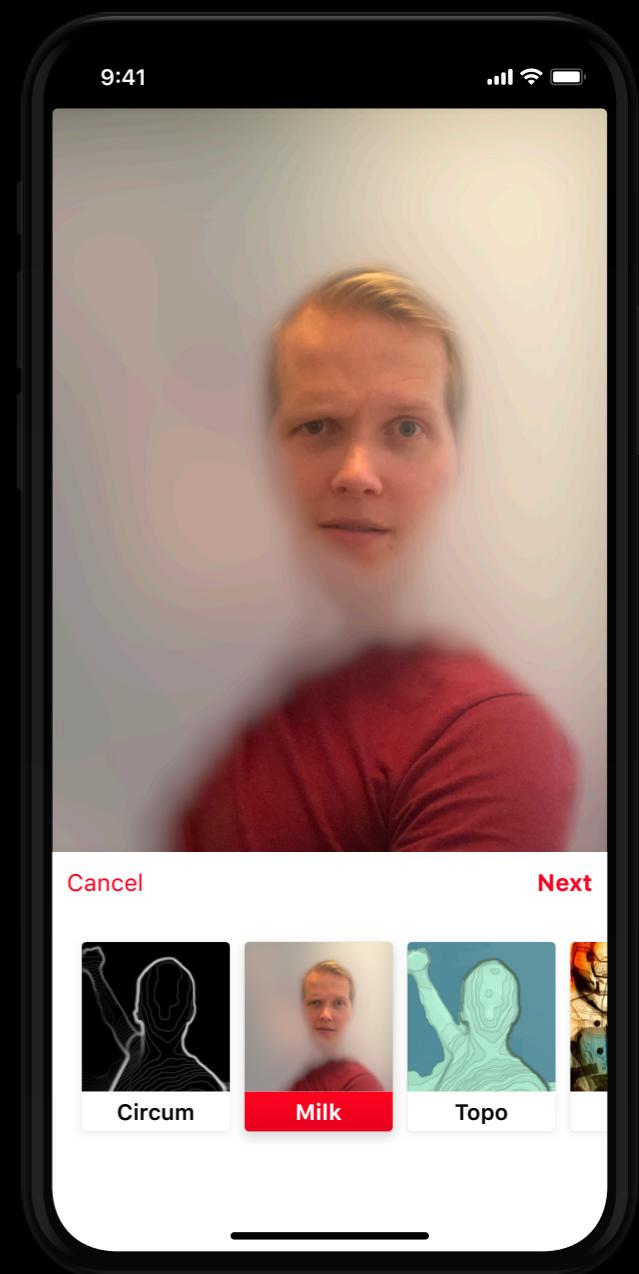
Sler



Piculet



Sler



Core Image

- Image Processing and Analysis
- iOS, macOS, and tvOS
- GPU
- Recipe based
- 207 built-in filters

Extensible

- Core Graphics
- Accelerate vImage
- Metal – Metal Shading Language
- CPU-based processing, e.g. Swift

Cllimage



CIFilter

Sepia
→

Cllimage



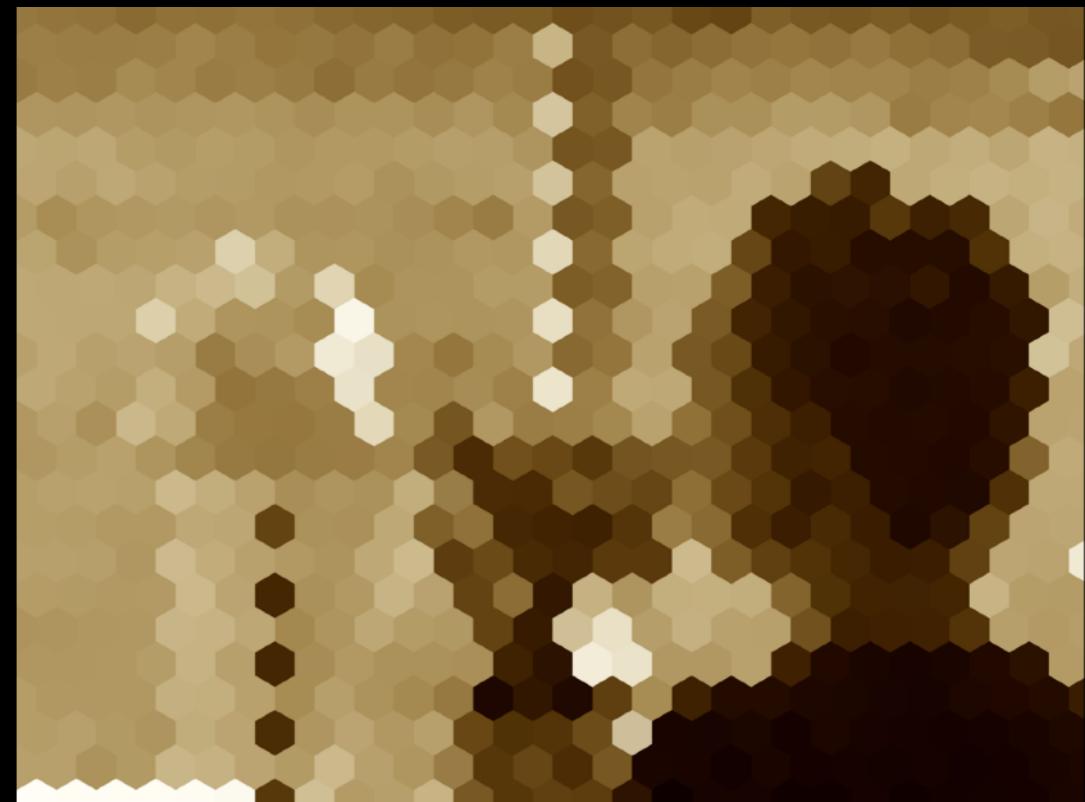
Cllimage



CIFilter

Hex
→

Cllimage



Cllimage



CIFilter

Instant
→

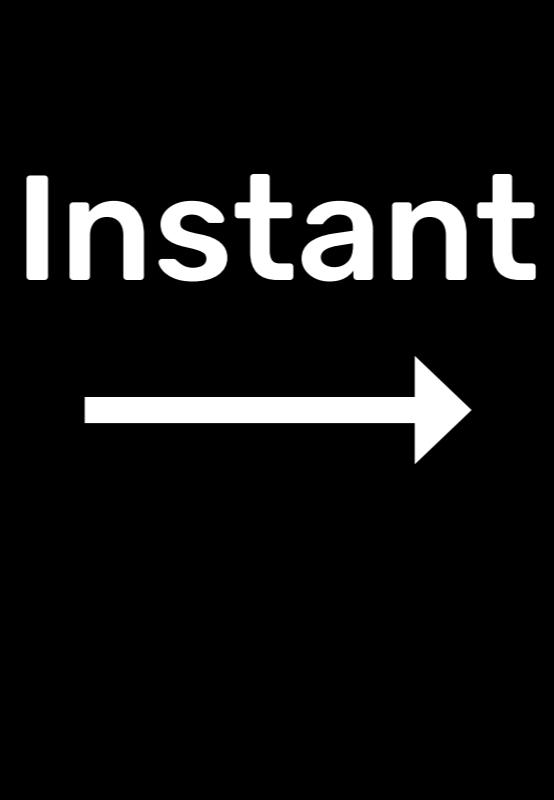
Cllimage



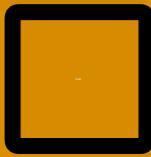


Instant

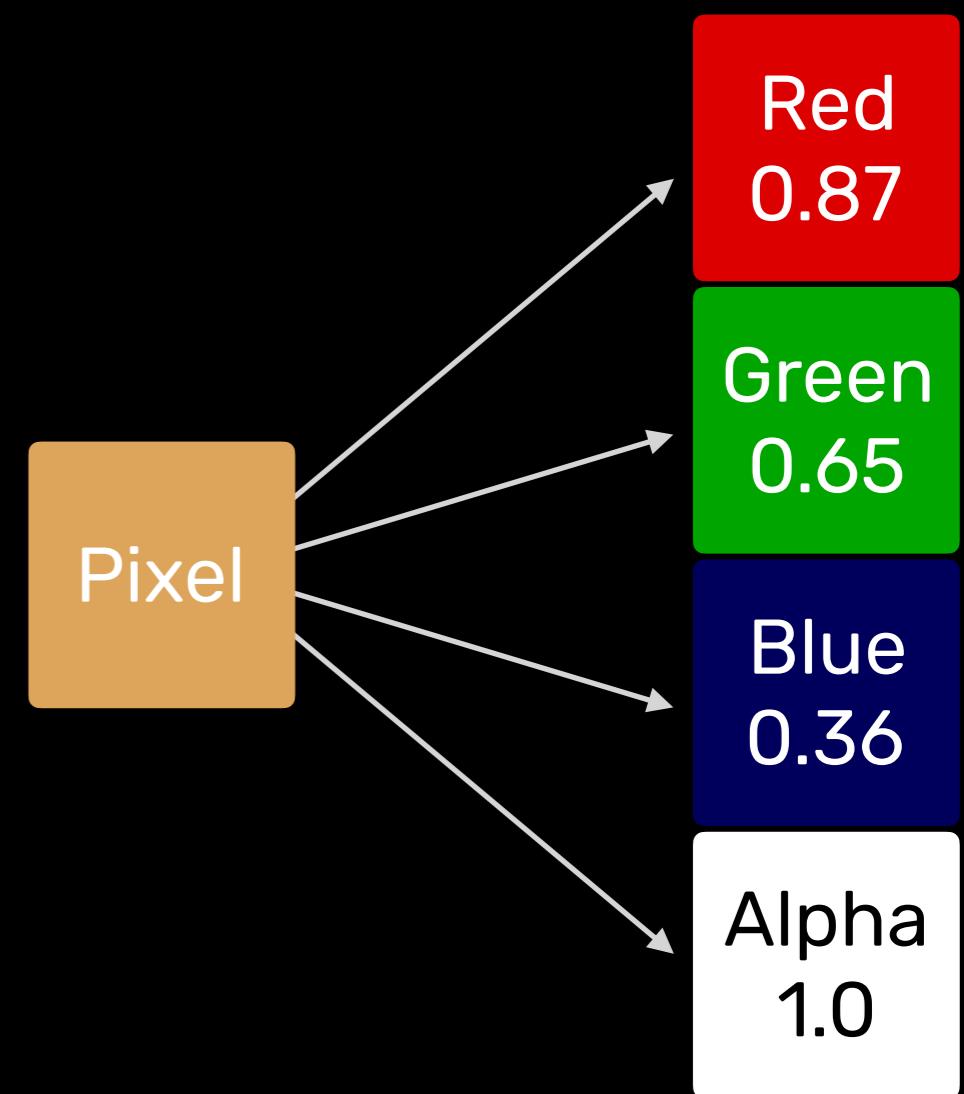
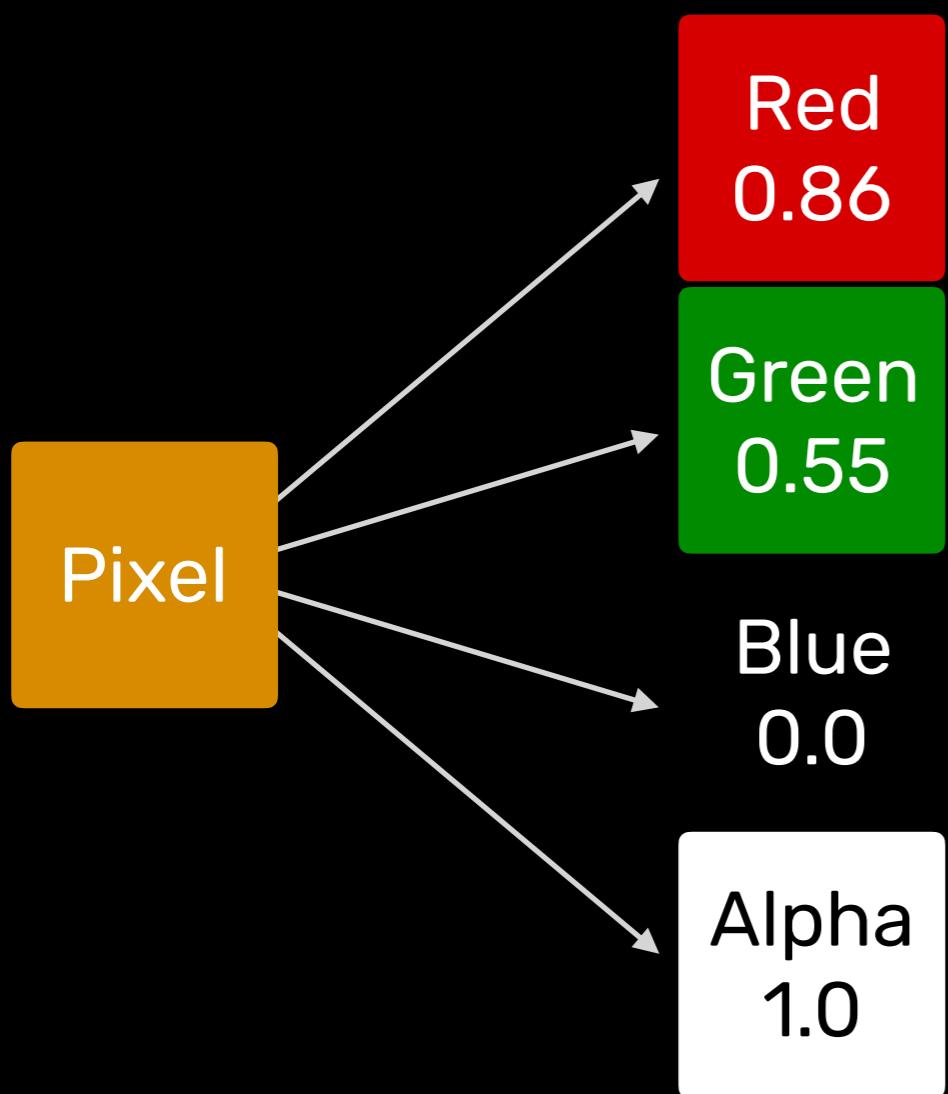




Instant



Instant



Color

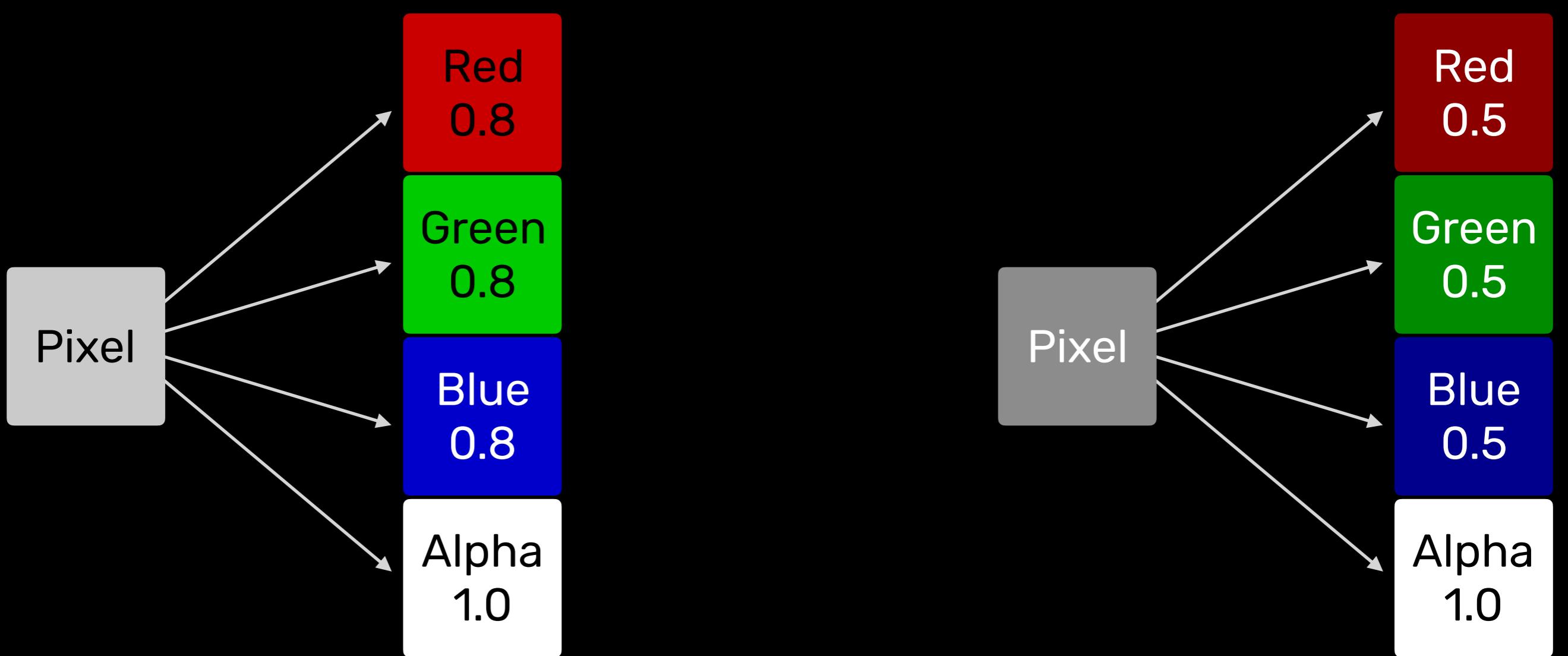
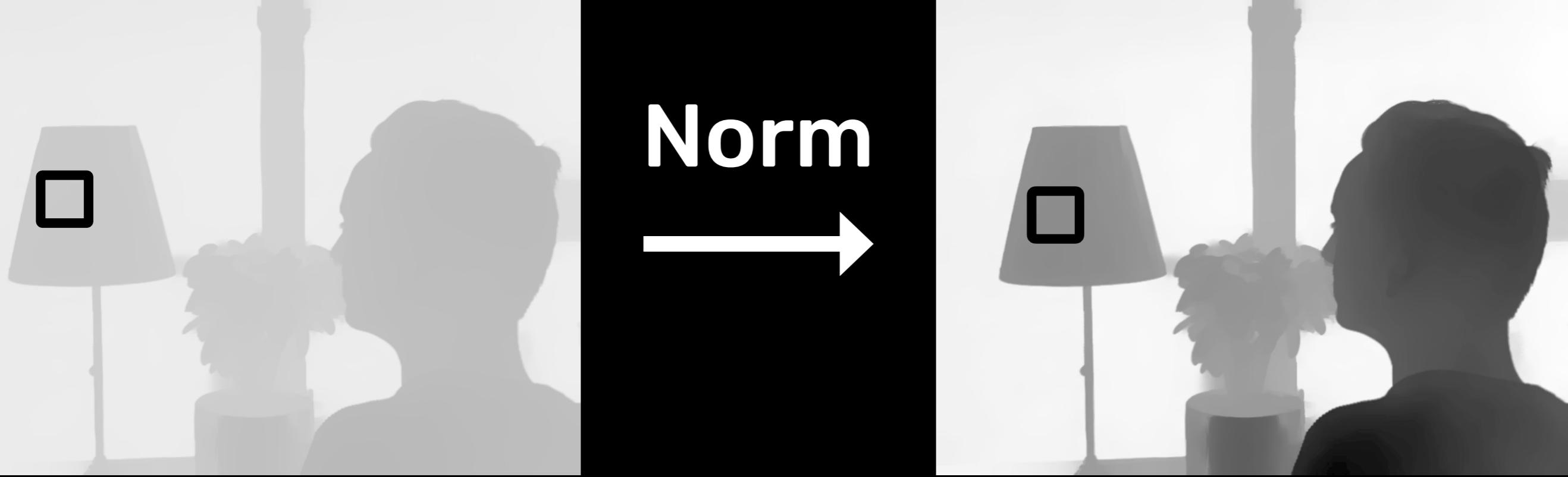


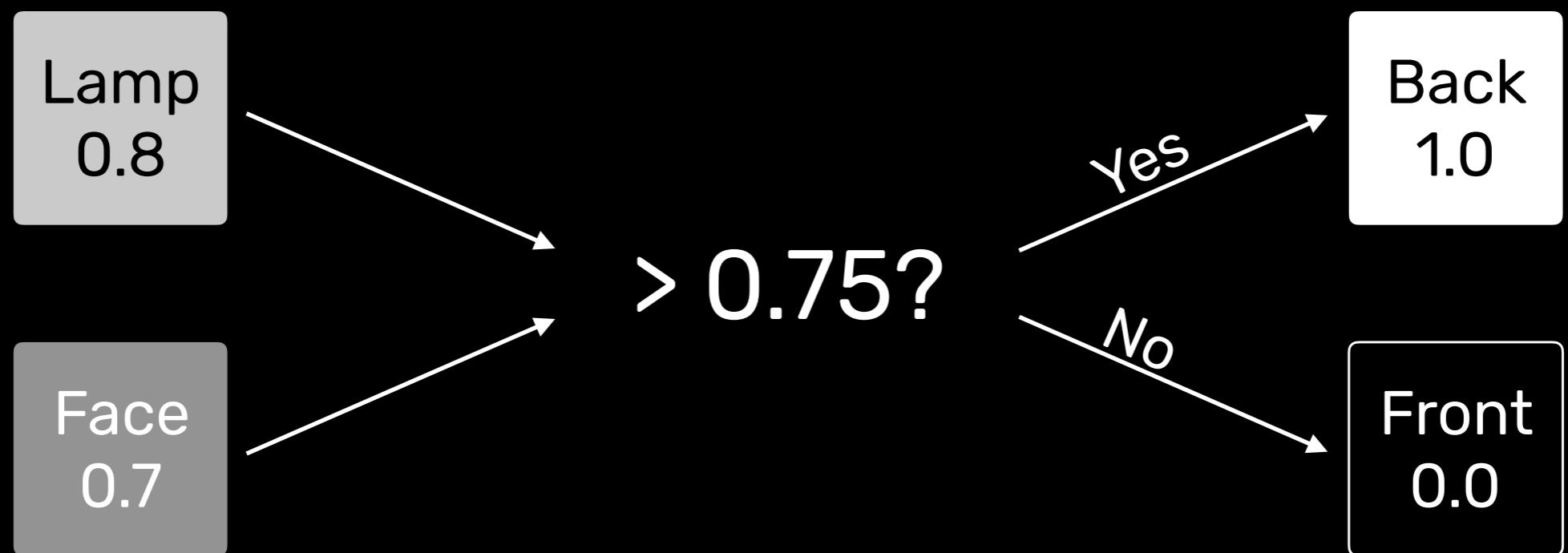
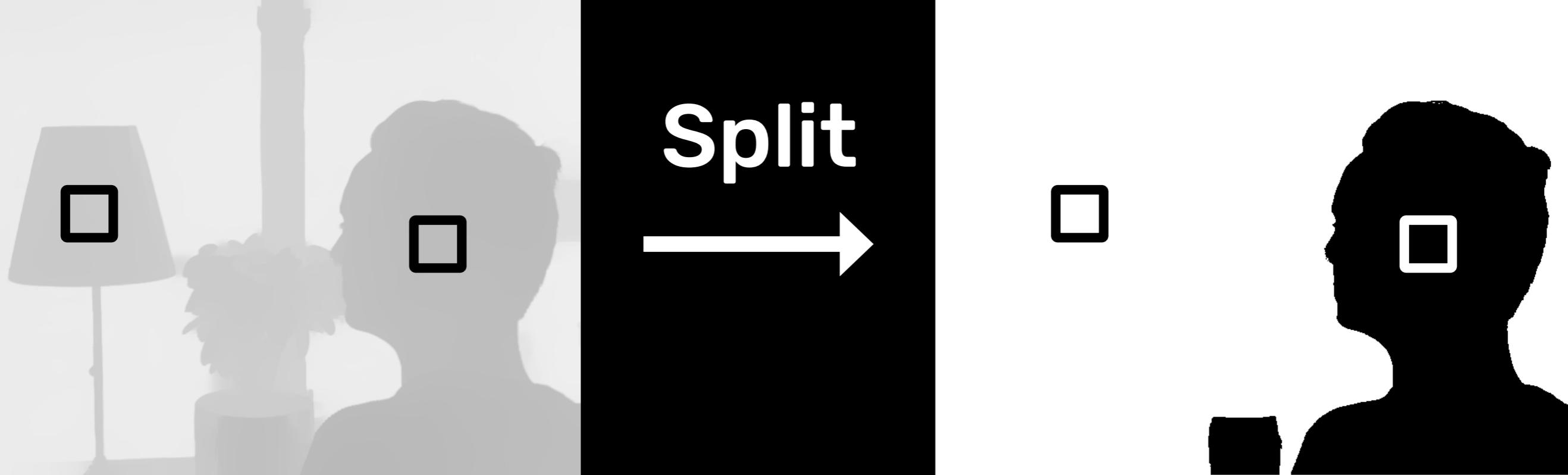
Depth Map



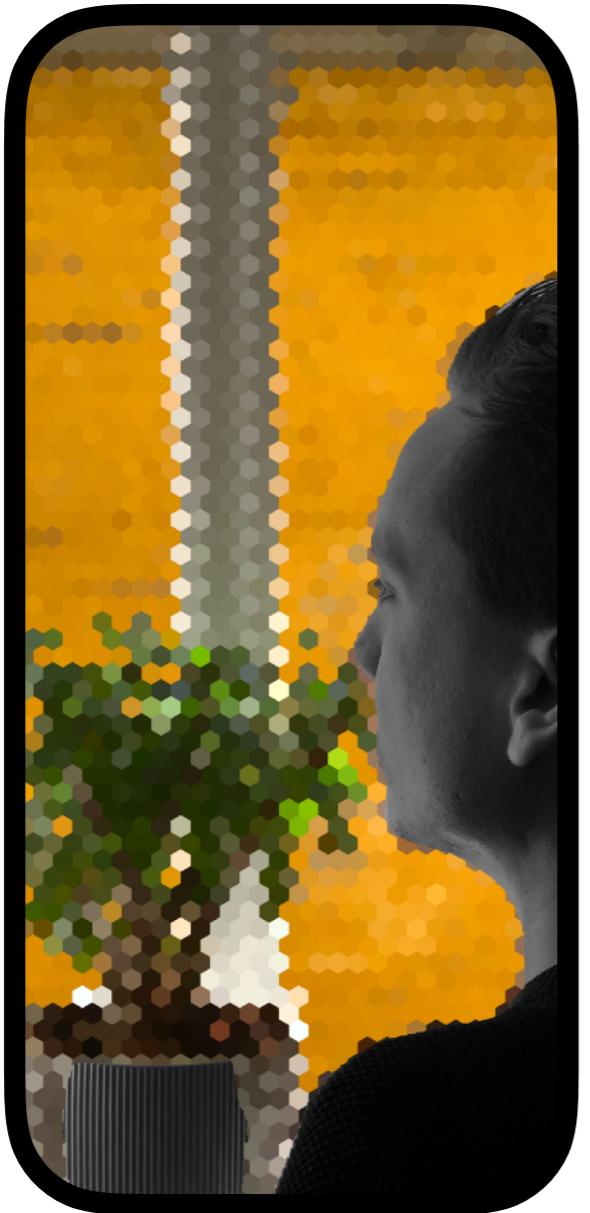
Cllimage

Cllimage

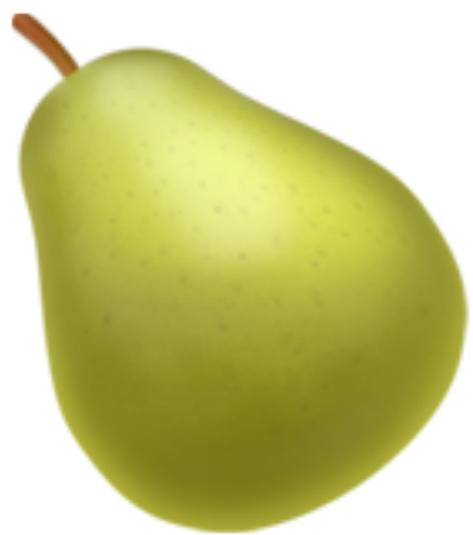




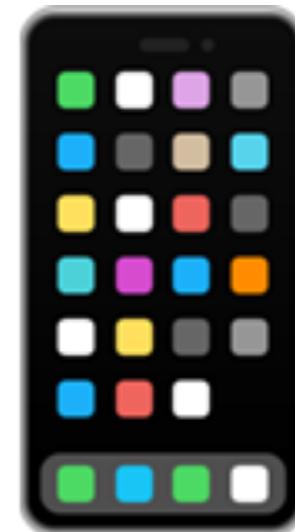
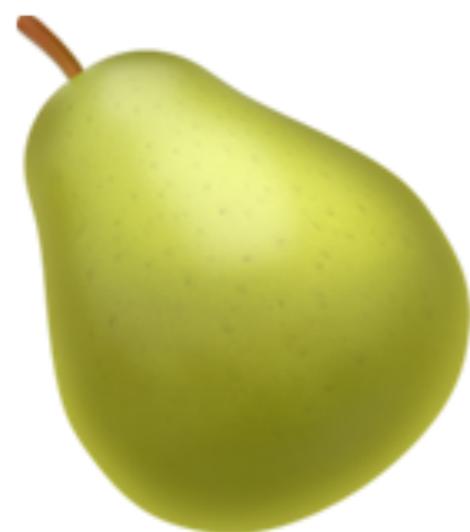
Outcome



Pair Up



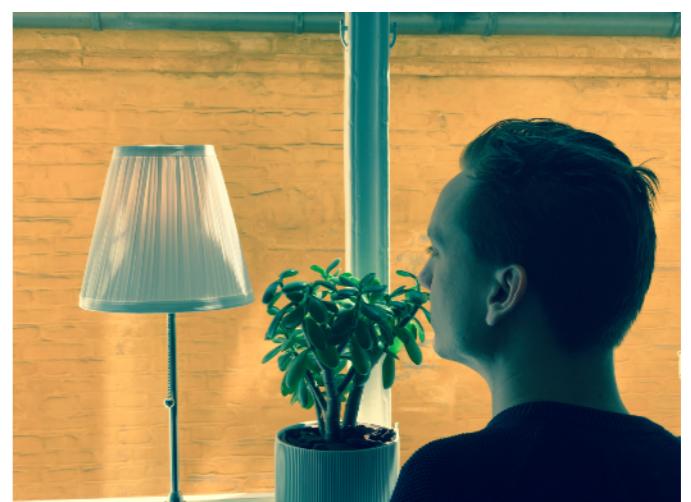
Pair Up



git.kabellmunk.dk/
coreimage-color-and-depth/
photo-editor



Task #1



Task #1



[CIFilter.io](#)

CICategoryColorEffect

Task #1



Task1.swift

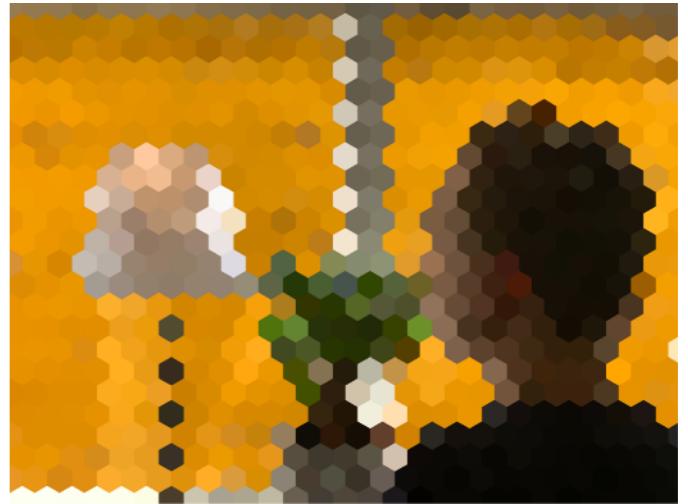
Task #1



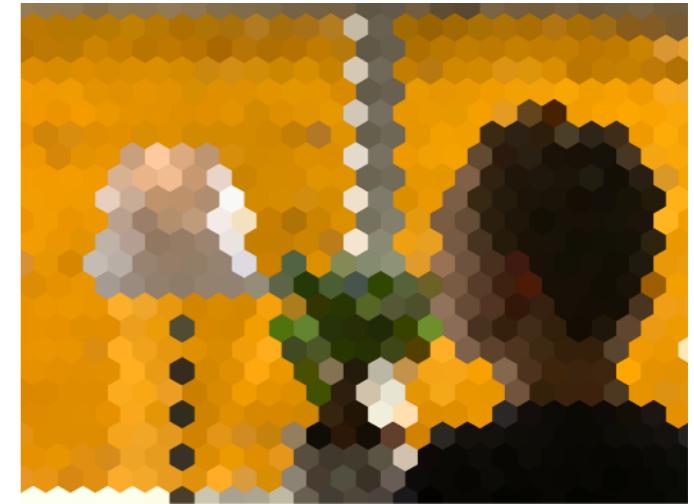
```
let filter = CIFilter(name: "CIPhotoEffectNoir")
filter?.setValue(originalImage,
                 forKey: "inputImage")
let result = filter?.outputImage
```



Task #2



Task #2

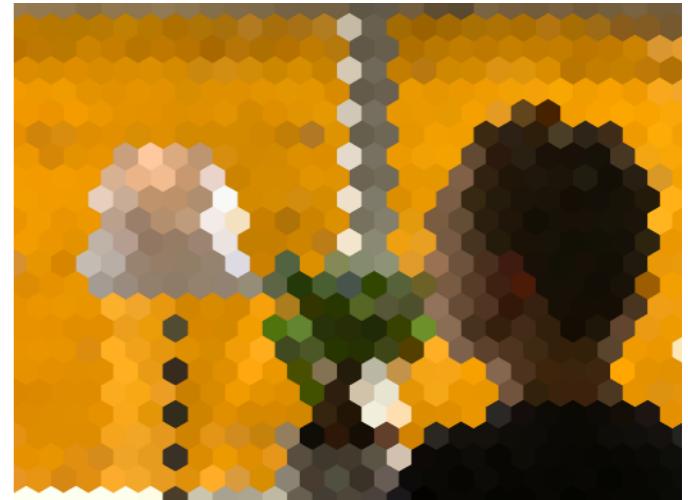


Blur / Distortion / Stylize

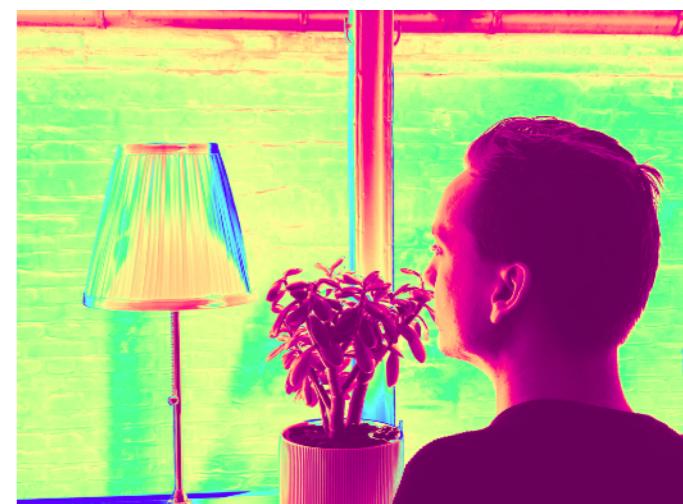
Task2.swift



Task #2



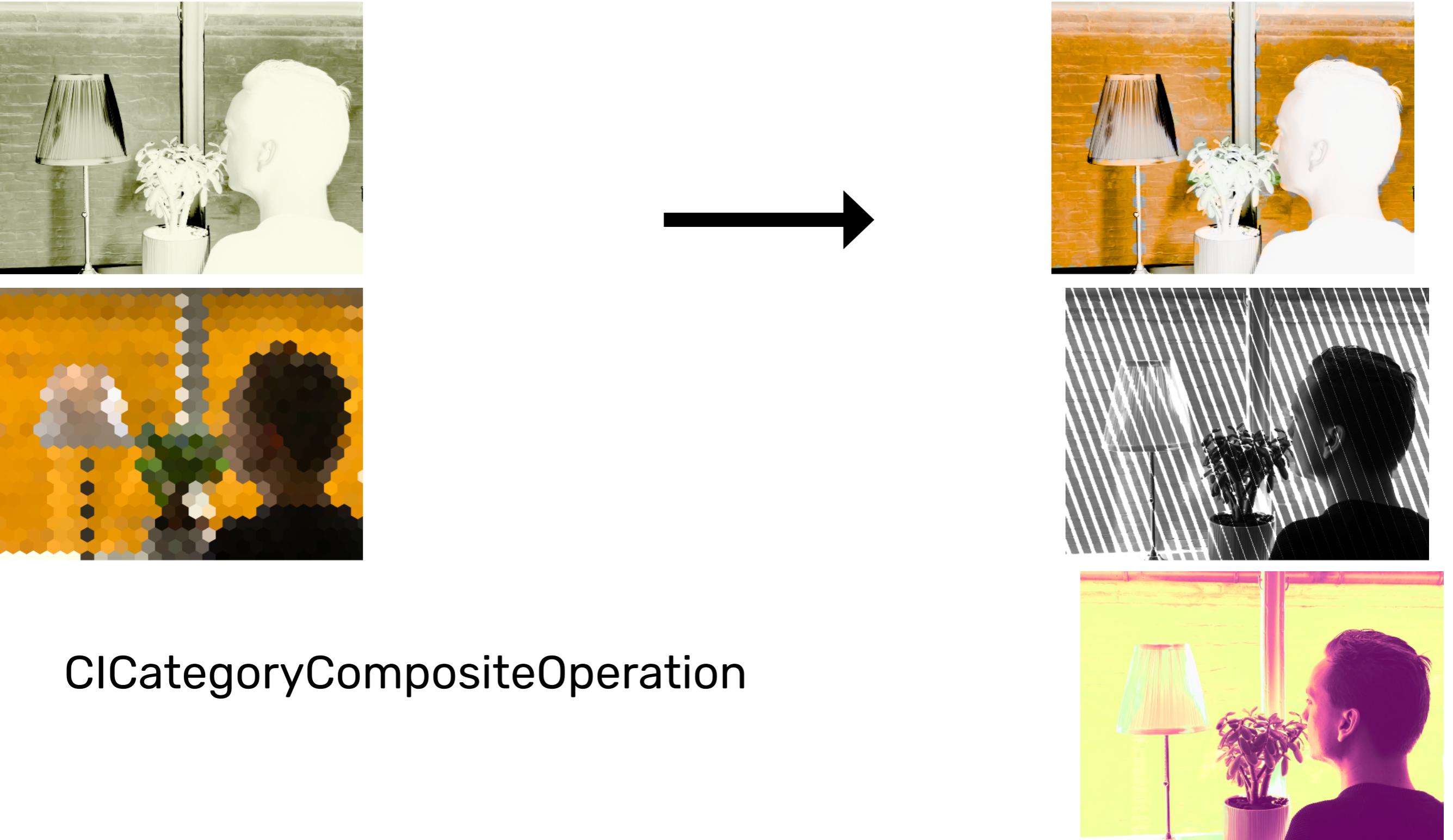
```
filter.setValue(10, forKey: "inputRadius")
```



Task #3



Task #3



Task #3



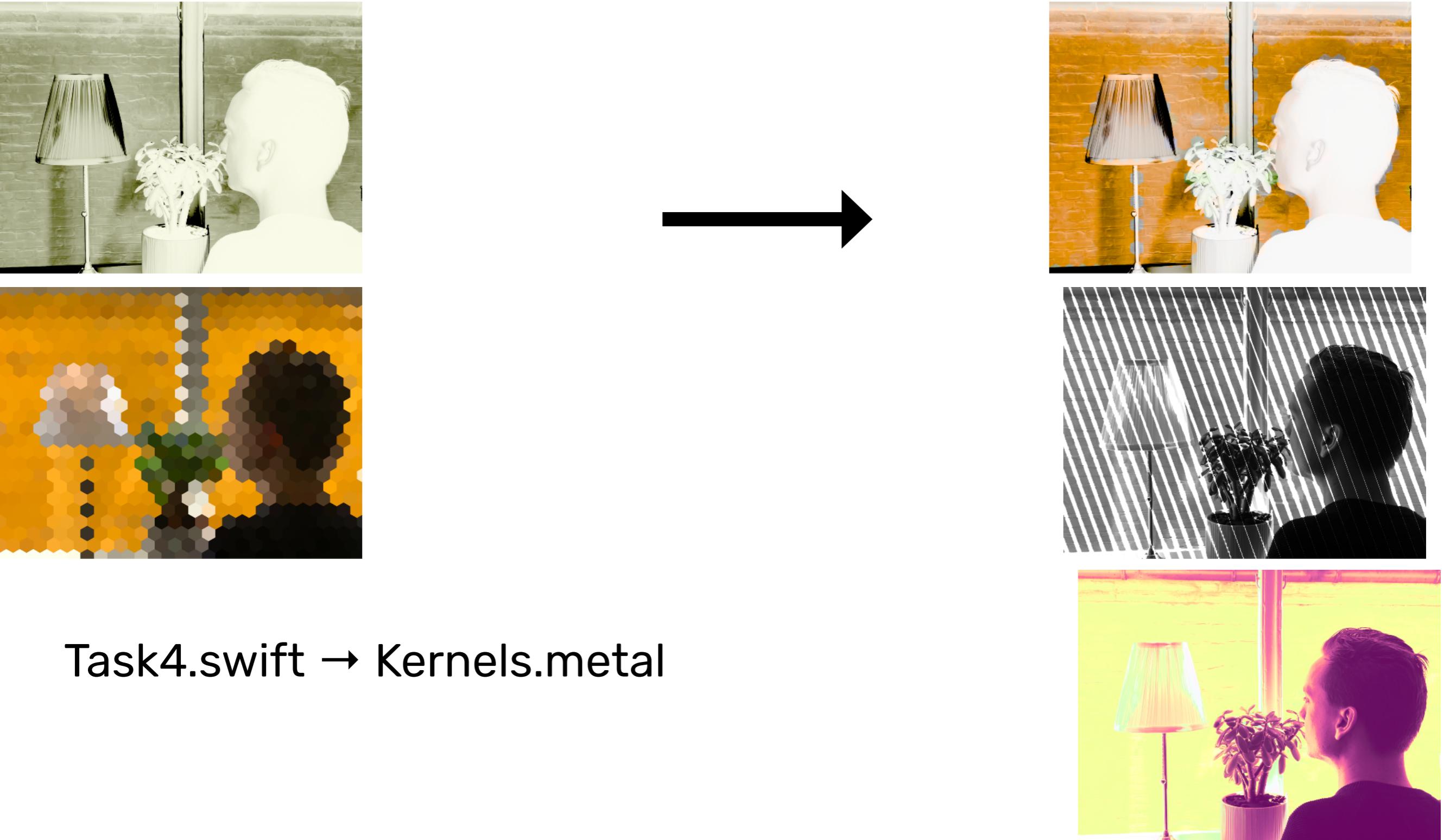
```
let filter = CIFilter(name: "CIMultiplyCompositing")
filter.setValue(task1, forKey: "inputBackgroundImage")
filter.setValue(task2, forKey: "inputImage")
let blendedImage = filter.outputImage
```



Task #4



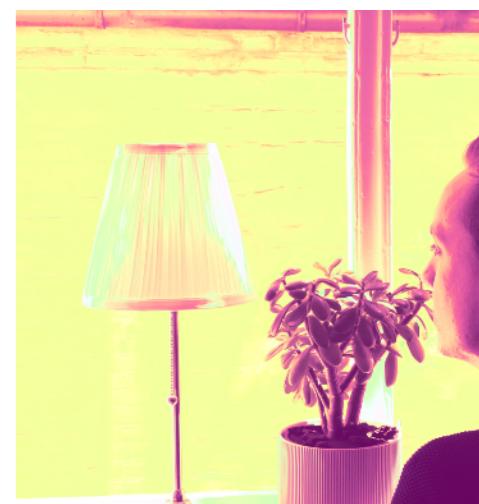
Task #4



Task #4



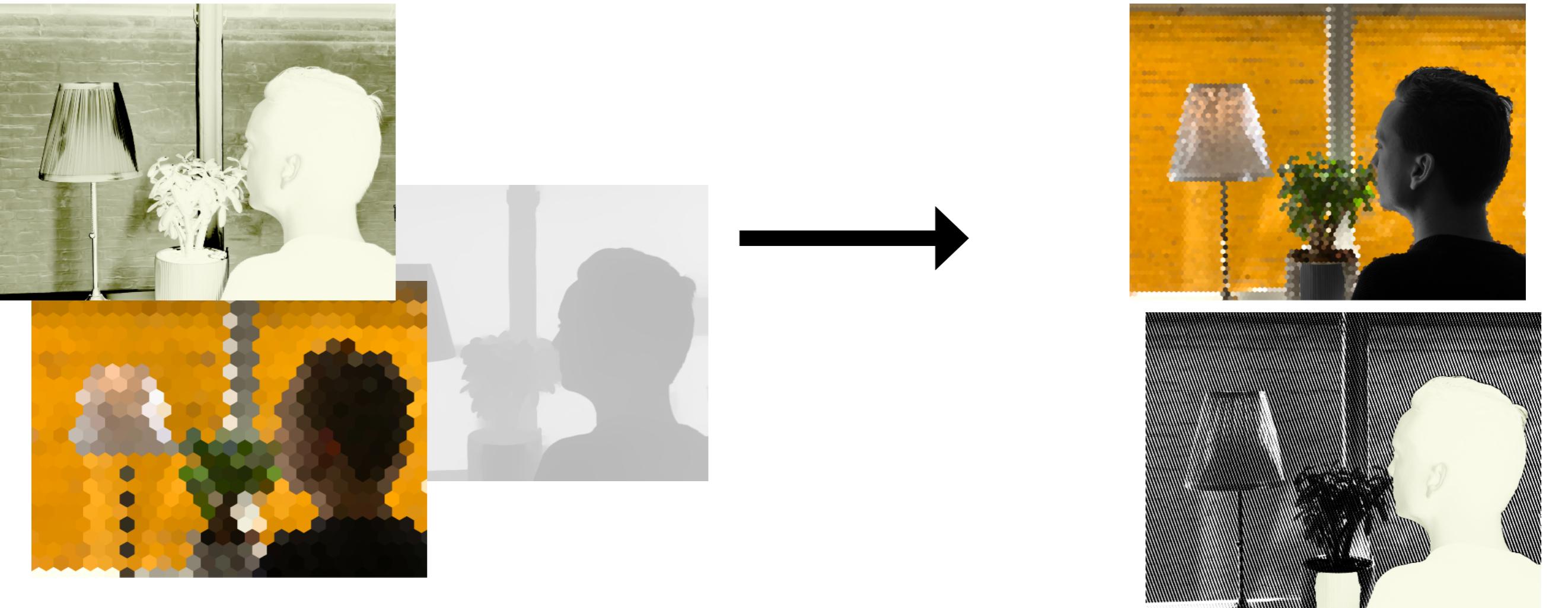
```
float4 kernel(sample_t foreground, sample_t background) {  
    return foreground + background;  
}
```



Task #5



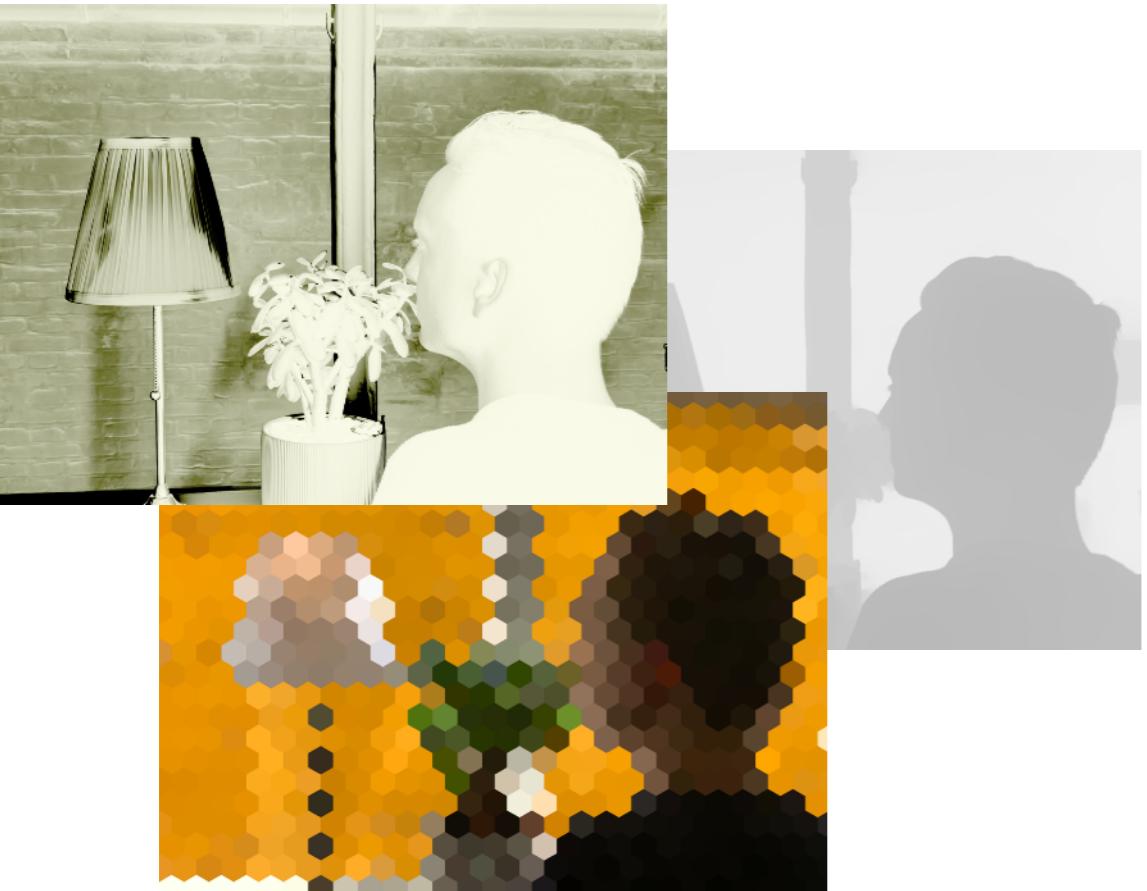
Task #5



Task5.swift → Kernels.metal



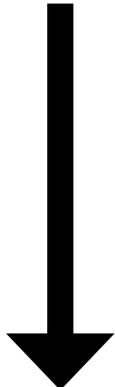
Task #5



```
float4 task5(sample_t foreground,  
            sample_t background,  
            sample_t depthMap) {  
    return /* float4 */;  
}
```



Appendix #1



```
float2 appendix1(destination dest) {
    float2 c = dest.coord();

    float columnWidth = 150.0;
    float columnIndex = floor(c.x / columnWidth);

    float2 sampleCoord;
    if (int(columnIndex) % 2 == 0) {
        sampleCoord = float2(c.x, c.y + 30.0);
    } else {
        sampleCoord = float2(c.x, c.y - 30.0);
    }

    return sampleCoord;
}
```

Appendix #2



```
float4 appendix2(sampler originalImage,
                  sampler depthMap,
                  destination dest) {
    float2 c = dest.coord();

    float depth = depthMap.sample(depthMap.coord()).r;
    float distortAmount = (1.0 - min(depth, 1.0)) * 100.0;

    float columnWidth = 150.0;
    float columnIndex = floor(c.x / columnWidth);

    float2 sampleCoord;
    if (int(columnIndex) % 2 == 0) {
        sampleCoord = float2(c.x, c.y + distortAmount);
    } else {
        sampleCoord = float2(c.x, c.y - distortAmount);
    }

    float2 samplerSpaceCoord = originalImage.transform(sampleCoord);
    return originalImage.sample(samplerSpaceCoord);
}
```

Appendix #3

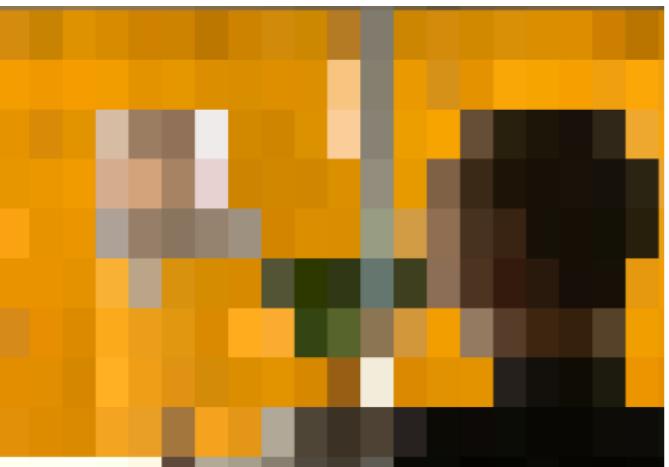


```
float2 appendix3(destination dest) {
    float2 c = dest.coord();

    const float2 tileSize = float2(200.0, 300.0);

    float2 sampleCoord = float2(
        floor(c.x / tileSize.x) * tileSize.x,
        floor(c.y / tileSize.y) * tileSize.y
    );

    return sampleCoord;
}
```



Core Image

Photo Editing in Color and Depth

SwiftAveiro – June 20, 2019

by Kalle Kabell and Tobias Due Munk