Instance Method

applyGamma(linearParameters: exponentialParameters:boundary: destination:)

Applies a piecewise gamma calculation to a 32-bit pixel buffer.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst | macOS 13.0+ | tvOS 16.0+ | visionOS | watchOS 9.0+

```
func applyGamma(
    linearParameters: (scale: Float, bias: Float),
    exponentialParameters: (scale: Float, preBias: Float, gamma: Float, postBias: F
    boundary: Float,
    destination: vImage.PixelBuffer<Format>
)
```

Available when `Format` conforms to `StaticPixelFormat` and `Format.ComponentType` is `Float`.

Parameters

`linearParameters`
The scale and bias applied to pixels with a value below `boundary`.

`exponentialParameters`
The boundary value that defines whether the function transforms pixels with the linear or the exponential calculation.

`boundary`
The parameters that the function uses for the exponential calculation.

`destination`
The destination pixel buffer.

Discussion

This function computes the output value of each pixel using the following pseudocode:

```
For each source pixel value x:
 if x < boundary:
     scale = linearParameters.scale
     bias = linearParameters.bias

     r = x * scale + bias
 else:
     scale = exponentialParameters.scale
     preBias = exponentialParameters.preBias
     gamma = exponentialParameters.gamma
     postBias = exponentialParameters.postBias

     t = x * scale + preBias
     r = pow(t, gamma) + postBias
 output pixel value = r
```

For example, the following code applies piecewise gamma to a one-pixel, two-channel pixel buffer.

The first channel's value is below the boundary and the second channel's value is above the boundary.

The operation mutliplies the first channel's value by two and squares the second channel's value.

```
let buffer = vImage.PixelBuffer<vImage.InterleavedFx2>(
    pixelValues: [0.25, 0.75],
    size: vImage.Size(width: 1,
                      height: 1))

buffer.applyGamma(linearParameters: (scale: 2,
                                     bias: 0),
                  exponentialParameters: (scale: 1,
                                          preBias: 0,
                                          gamma: 2,
                                          postBias: 0),
```

```
                    boundary: 0.5,
                    destination: buffer)

// Prints "[0.5, 0.5625]" = [ 0.25 * 2, 0.75² ].
print(buffer.array)
```

See Also

Applying piecewise gamma

`func applyGamma(linearParameters: (scale: Float, bias: Float), exponentialParameters: (scale: Float, preBias: Float, gamma: Float, postBias: Float), boundary: Pixel_8, destination: vImage.PixelBuffer<Format>)`

Applies a piecewise gamma calculation to an 8-bit pixel buffer.