

[Accelerate](#) / [...](#) / [vImage.PixelBuffer](#) / `applyGamma(_:destination:)`

Instance Method

applyGamma(_:destination:)

Applies a gamma function 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(
    _ gamma: vImage.Gamma,
    destination: vImage.PixelBuffer<Format>
)
```

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

Parameters

`gamma`

An enumeration that specifies either a used-defined or constant gamma.

`destination`

The destination pixel buffer.

Discussion

For example, the following code applies a gamma of `2.0` to a one-pixel pixel buffer:

```
let buffer = vImage.PixelBuffer<vImage.PlanarF>(
    pixelValues: [0.5],
    size: vImage.Size(width: 1,
                       height: 1))
```

```
buffer.applyGamma(.fullPrecision(2),  
                 destination: buffer)  
  
// Prints "[0.25]" = [0.5²].  
print(buffer.array)
```

See Also

Applying gamma

```
func applyGamma(vImage.Gamma, intermediateBuffer: vImage.PixelBuffer<vImage.PlanarF>?, destination: vImage.PixelBuffer<vImage.Planar8>)
```

Applies a gamma function to an 8-bit planar pixel buffer.

```
func applyGamma(vImage.Gamma, intermediateBuffer: vImage.PixelBuffer<vImage.InterleavedFx2>?, destination: vImage.PixelBuffer<vImage.Interleaved8x2>)
```

Applies a gamma function to an 8-bit-per-channel, 2-channel interleaved pixel buffer.

```
func applyGamma(vImage.Gamma, intermediateBuffer: vImage.PixelBuffer<vImage.InterleavedFx3>?, destination: vImage.PixelBuffer<vImage.Interleaved8x3>)
```

Applies a gamma function to an 8-bit-per-channel, 3-channel interleaved pixel buffer.

```
func applyGamma(vImage.Gamma, intermediateBuffer: vImage.PixelBuffer<vImage.InterleavedFx4>?, destination: vImage.PixelBuffer<vImage.Interleaved8x4>)
```

Applies a gamma function to an 8-bit-per-channel, 4-channel interleaved pixel buffer.

```
enum Gamma
```

Describes either a used-defined or constant gamma.