

[Accelerate](#) / [... \(1\)](#) / [vImage.PixelBuffer](#) / `separableConvolve(horizontalKernel:verticalKernel:bias:edgeMode:destination:)`

Instance Method

separableConvolve(horizontalKernel: verticalKernel:bias:edgeMode: destination:)

Performs separable convolution on an 8-bit planar pixel buffer.

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

```
func separableConvolve(  
    horizontalKernel: [Float],  
    verticalKernel: [Float],  
    bias: Float = 0,  
    edgeMode: vImage.EdgeMode<Pixel_16U>,  
    destination: vImage.PixelBuffer<Format>  
)
```

Available when Format is `vImage.Planar8`.

Parameters

horizontalKernel

The 1D horizontal convolution kernel.

verticalKernel

The 1D vertical convolution kernel.

bias

A value that the operation adds to each element in the convolution result, before performing any clipping.

edgeMode

The convolution edge mode. The background color must be a single `Pixel_16U` value.

destination

The destination pixel buffer.

Discussion

The following code shows how to apply a Gaussian blur using separable convolution to a planar buffer:

```
let srcImage = imageLiteral(resourceName: "...").cgImage(
    forProposedRect: nil,
    context: nil,
    hints: nil)!

var cgImageFormat = vImage_CGImageFormat(
    bitsPerComponent: 8,
    bitsPerPixel: 8 * 1,
    colorSpace: CGColorSpaceCreateDeviceGray(),
    bitmapInfo: CGBitmapInfo(rawValue: CGImageAlphaInfo.none.rawValue))!

let src = try vImage.PixelBuffer(
    cgImage: srcImage,
    cgImageFormat: &cgImageFormat,
    pixelFormat: vImage.Planar8.self)

let dest = vImage.PixelBuffer(
    size: src.size,
    pixelFormat: vImage.Planar8.self)

src.separableConvolve(
    horizontalKernel: vImage.ConvolutionKernel.gaussian1Dx7,
    verticalKernel: vImage.ConvolutionKernel.gaussian1Dx7,
    edgeMode: .truncateKernel,
    destination: dest)

let outputImage = dest.makeCGImage(cgImageFormat: cgImageFormat)
```

See Also

Related Documentation

{} Blurring an image

Filter an image by convolving it with custom and high-speed kernels.

Separable convolution

```
func separableConvolve(horizontalKernel: [Float], verticalKernel: [Float], bias: Float, edgeMode: vImage.EdgeMode<Pixel_16F>, useFloat16Accumulator: Bool, destination: vImage.PixelBuffer<Format>)
```

Performs separable convolution on a 16-bit planar pixel buffer.

```
func separableConvolve(horizontalKernel: [Float], verticalKernel: [Float], bias: Float, edgeMode: vImage.EdgeMode<Pixel_F>, destination: vImage.PixelBuffer<Format>)
```

Performs separable convolution on a 32-bit planar pixel buffer.

```
func separableConvolve(horizontalKernel: [Float], verticalKernel: [Float], bias: Float, edgeMode: vImage.EdgeMode<Pixel_16U>, destination: vImage.PixelBuffer<Format>)
```

Performs separable convolution on a multiple plane 8-bit pixel buffer.

```
func separableConvolve(horizontalKernel: [Float], verticalKernel: [Float], bias: Float, edgeMode: vImage.EdgeMode<Pixel_F>, destination: vImage.PixelBuffer<Format>)
```

Performs separable convolution on a multiple plane 32-bit pixel buffer.

```
struct ConvolutionKernel
```

Constants that describe 1D convolution kernels.