

[Accelerate](#) / Convolution

API Collection

Convolution

Apply a convolution kernel to an image.

Overview

Convolution is a common image-processing technique that changes the value of a pixel according to the values of its surrounding pixels. Many common image filters, such as blurring, detecting edges, sharpening, and embossing, derive from convolution.

Kernels form the basis of convolution operations. Kernels are arrays or matrices of weights that indicate the influence of a pixel's neighbors on its final value. To calculate the value of each transformed pixel, a convolution operation adds the products of each surrounding pixel value with the corresponding kernel value. During a convolution operation, the kernel passes over every pixel in the image, repeating this procedure, and then applies the effect to the entire image.

Topics

Convolving an 8-bit image with 32-bit weights

```
func vImageConvolveFloatKernel_ARGB8888(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Float,
UnsafePointer<UInt8>!, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit-per-channel, 4-channel interleaved image using 32-bit weights.

Convolving with separable filter kernels

```
func vImageSepConvolve_Planar8(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<Float>!,
UInt32, Float, Pixel_16U, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit planar image by separate horizontal and vertical separable kernels.

```
func vImageSepConvolve_Planar16U(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<Float>!,
UInt32, Float, Pixel_16U, vImage_Flags) -> vImage_Error
```

Convolve an unsigned 16-bit planar image by separate horizontal and vertical separable kernels.

```
func vImageSepConvolve_Planar16F(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<Float>!,
UInt32, Float, Pixel_16F, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 16-bit planar image by separate horizontal and vertical separable kernels.

```
func vImageSepConvolve_PlanarF(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<Float>!,
UInt32, Float, Pixel_F, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 32-bit planar image by separate horizontal and vertical separable kernels.

```
func vImageSepConvolve_Planar8to16U(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<
Float>!, UInt32, Float, Float, Pixel_8, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit planar image by separate horizontal and vertical separable kernels, and writes the result to an unsigned 16-bit planar destination.

```
func vImageSepConvolve_ARGB8888(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UnsafePointer<Float>!, UInt32, UnsafePointer<Float>!,
UInt32, Float, UnsafePointer<UInt8>!, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit-per-channel, 4-channel interleaved image by separate horizontal and vertical separable kernels.

Convoluting without bias

```
func vImageConvolve_Planar8(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Int16>!, UInt32, UInt32, Int32, Pixel_8, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit planar image by a 2D kernel and divide the pixel values by a divisor.

```
func vImageConvolve_Planar16F(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Pixel_16F, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 16-bit planar image by a 2D kernel.

```
func vImageConvolve_PlanarF(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Pixel_F, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 32-bit planar image by a 2D kernel.

```
func vImageConvolve_ARGB8888(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Int16>!, UInt32, UInt32, Int32, UnsafePointer<UInt8>!, vImage_Flags) -> vImage_Error
```

Convolve an 8-bit-per-channel, 4-channel interleaved image by a 2D kernel and divide the pixel values by a divisor.

```
func vImageConvolve_ARGB16F(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, UnsafePointer<UInt16>!, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 16-bit-per-channel, 4-channel interleaved image by a 2D kernel, then divide the pixel values by a divisor.

```
func vImageConvolve_ARGBFFFF(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, UnsafePointer<Float>!, vImage_Flags) -> vImage_Error
```

Convolve a floating-point 32-bit-per-channel, 4-channel interleaved image by a 2D kernel, then divide the pixel values by a divisor.

Convoluting with bias

```
func vImageConvolveWithBias_Planar8(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Int16>!, UInt32, UInt32, Int32,
Int32, Pixel_8, vImage_Flags) -> vImage_Error
```

Convolves an 8-bit planar image by a 2D kernel and adds a bias.

```
func vImageConvolveWithBias_Planar16F(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Float,
Pixel_16F, vImage_Flags) -> vImage_Error
```

Convolves a floating-point 16-bit planar image by a 2D kernel and adds a bias.

```
func vImageConvolveWithBias_PlanarF(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Float,
Pixel_F, vImage_Flags) -> vImage_Error
```

Convolves a floating-point 32-bit planar image by a 2D kernel and adds a bias.

```
func vImageConvolveWithBias_ARGB8888(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Int16>!, UInt32, UInt32, Int32,
Int32, UnsafePointer<UInt8>!, vImage_Flags) -> vImage_Error
```

Convolves an 8-bit-per-channel, 4-channel interleaved image by a 2D kernel and adds a bias.

```
func vImageConvolveWithBias_ARGB16F(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Float,
UnsafePointer<UInt16>!, vImage_Flags) -> vImage_Error
```

Convolves a floating-point 16-bit-per-channel, 4-channel interleaved image by a 2D kernel and adds a bias.

```
func vImageConvolveWithBias_ARGBFFFF(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafePointer<Float>!, UInt32, UInt32, Float,
UnsafePointer<Float>!, vImage_Flags) -> vImage_Error
```

Convolves a floating-point 32-bit-per-channel, 4-channel interleaved image by a 2D kernel and adds a bias.

Convolving with multiple kernels

```
func vImageConvolveMultiKernel_ARGB8888(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafeMutablePointer<UnsafePointer<Int16>?>!,
UInt32, UInt32, UnsafePointer<Int32>!, UnsafePointer<Int32>!, Unsafe
Pointer<UInt8>!, vImage_Flags) -> vImage_Error
```

Convolve each channel of an 8-bit-per-channel, 4-channel interleaved image by one of the four 2D kernels.

```
func vImageConvolveMultiKernel_ARGBFFFF(UnsafePointer<vImage_Buffer>,
UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixel
Count, vImagePixelCount, UnsafeMutablePointer<UnsafePointer<Float>?>,
UInt32, UInt32, UnsafePointer<Float>, UnsafePointer<Float>, vImage
_Flags) -> vImage_Error
```

Convolve each channel of a floating-point 32-bit-per-channel, 4-channel interleaved image by one of the four 2D kernels.

Convoluting with high-speed box and tent filters

```
func vImageBoxConvolve_Planar8(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UInt32, UInt32, Pixel_8, vImage_Flags) -> vImage_Error
```

Applies a box filter to an 8-bit planar source image.

```
func vImageBoxConvolve_ARGB8888(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UInt32, UInt32, UnsafePointer<UInt8>!, vImage_Flags) -
> vImage_Error
```

Applies a box filter to an 8-bit-per-channel, 4-channel interleaved source image.

```
func vImageTentConvolve_Planar8(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UInt32, UInt32, Pixel_8, vImage_Flags) -> vImage_Error
```

Applies a tent filter to an 8-bit planar source image.

```
func vImageTentConvolve_ARGB8888(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, v
ImagePixelCount, UInt32, UInt32, UnsafePointer<UInt8>!, vImage_Flags) -
> vImage_Error
```

Applies a tent filter to an 8-bit-per-channel, 4-channel interleaved source image.

Deconvoluting

```
func vImageRichardsonLucyDeConvolve_Planar8(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Int16>!, UnsafePointer<Int16>!, UInt32, UInt32, UInt32, UInt32, Int32, Int32, Pixel_8, UInt32, vImage_Flags) -> vImage_Error
```

Deconvolves an 8-bit planar image.

```
func vImageRichardsonLucyDeConvolve_PlanarF(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UnsafePointer<Float>!, UInt32, UInt32, UInt32, UInt32, Pixel_F, UInt32, vImage_Flags) -> vImage_Error
```

Deconvolves a floating-point 32-bit planar image.

```
func vImageRichardsonLucyDeConvolve_ARGB8888(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Int16>!, UnsafePointer<Int16>!, UInt32, UInt32, UInt32, UInt32, Int32, Int32, UnsafePointer<UInt8>!, UInt32, vImage_Flags) -> vImage_Error
```

Deconvolves an 8-bit-per-channel, 4-channel interleaved image.

```
func vImageRichardsonLucyDeConvolve_ARGBFFFF(UnsafePointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRawPointer!, vImagePixelCount, vImagePixelCount, UnsafePointer<Float>!, UnsafePointer<Float>!, UInt32, UInt32, UInt32, UInt32, UnsafePointer<Float>!, UInt32, vImage_Flags) -> vImage_Error
```

Deconvolves a floating-point 32-bit-per-channel, 4-channel interleaved image.

See Also

Convolution and Morphology



Blurring an image

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



Adding a bokeh effect to images

Simulate a bokeh effect by applying dilation.



Morphology

Dilate and erode images.