Documentation

Accelerate / vDSP / 2D convolution

API Collection

2D convolution

Perform convolution operations on matrices of real data.

Overview

Use the functions in this group to apply a convolution kernel to a 2D matrix. The vDSP library provides functions to convolve with fixed-sized kernels and kernels with an arbitrary size.

> **Important**
>
> The functions in this group don’t support in-place operation.

Topics

Fixed-Size Kernel

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int,
with3x3Kernel: U) -> [Double]
```

Returns the 2D convolution of a double-precision vector with a 3 x 3 kernel.

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int,
with3x3Kernel: U) -> [Float]
```

Returns the 2D convolution of a single-precision vector with a 3 x 3 kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int,
with3x3Kernel: U, result: inout V)
```

Calculates the 2D convolution of a double-precision vector with a 3 x 3 kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int, with3x3Kernel: U, result: inout V)
```

Calculates the 2D convolution of a single-precision vector with a 3 x 3 kernel.

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int, with5x5Kernel: U) -> [Double]
```

Returns the 2D convolution of a double-precision vector with a 5 x 5 kernel.

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int, with5x5Kernel: U) -> [Float]
```

Returns the 2D convolution of a single-precision vector with a 5 x 5 kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int, with5x5Kernel: U, result: inout V)
```

Calculates the 2D convolution of a double-precision vector with a 5 x 5 kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int, with5x5Kernel: U, result: inout V)
```

Calculates the 2D convolution of a single-precision vector with a 5 x 5 kernel.

```
vDSP_f3x3
```

Filters a single-precision image by performing a 2D convolution with a 3 x 3 kernel.

```
vDSP_f3x3D
```

Filters a double-precision image by performing a 2D convolution with a 3 x 3 kernel.

```
vDSP_f5x5
```

Filters a single-precision image by performing a 2D convolution with a 5 x 5 kernel.

```
vDSP_f5x5D
```

Filters a double-precision image by performing a 2D convolution with a 5 x 5 kernel.

Arbitrary-Size Kernel

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int, withKernel: U, kernelRowCount: Int, kernelColumnCount: Int) -> [Double]
```

Returns the 2D convolution of a double-precision vector with an arbitrarily sized kernel.

```
static func convolve<T, U>(T, rowCount: Int, columnCount: Int, withKernel: U, kernelRowCount: Int, kernelColumnCount: Int) -> [Float]
```

Returns the 2D convolution of a single-precision vector with an arbitrarily sized kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int, with
Kernel: U, kernelRowCount: Int, kernelColumnCount: Int, result: inout V
)
```

Calculates the 2D convolution of a double-precision vector with an arbitrarily sized kernel.

```
static func convolve<T, U, V>(T, rowCount: Int, columnCount: Int, with
Kernel: U, kernelRowCount: Int, kernelColumnCount: Int, result: inout V
)
```

Calculates the 2D convolution of a single-precision vector with an arbitrarily sized kernel.

`vDSP_imgfir`

Filters a single-precision image by performing a 2D convolution with an arbitrarily sized kernel.

`vDSP_imgfirD`

Filters a double-precision image by performing a 2D convolution with an arbitrarily sized kernel.

See Also

Vector and matrix correlation and convolution

1D correlation and convolution

Use correlation to compare and convolution to combine vectors of real or complex data.