API Collection

# Transforming with matrix multiplication

Use matrix multiplication to apply color transformations to images.

## Overview

Matrix multiplication functions treat source pixels as `m`-element vectors, with the number of vector elements corresponding to the number of channels. The functions multiply each source value by an `n x m` matrix to produce an `n`-element destination pixel. You can use matrix multiplication functions for tasks like converting between color spaces. For example, you can multiply three-channel RGB pixels by a 4 x 3 matrix to generate four-channel CMYK pixels.

## Topics

### Multiplying multiple-plane pixels by a matrix

{} Adjusting saturation and applying tone mapping

Convert an RGB image to discrete luminance and chrominance channels, and apply color and contrast treatments.

```
func vImageMatrixMultiply_Planar8(UnsafeMutablePointer<UnsafePointer<vImage_Buffer>?>, UnsafeMutablePointer<UnsafePointer<vImage_Buffer>?>, UInt32, UInt32, UnsafePointer<Int16>, Int32, UnsafePointer<Int16>!, UnsafePointer<Int32>!, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in a set of 8-bit source image planes by a matrix to produce a set of 8-bit destination image planes.

```
func vImageMatrixMultiply_Planar16S(UnsafeMutablePointer<UnsafePointer<
vImage_Buffer>?>, UnsafeMutablePointer<UnsafePointer<vImage_Buffer>?>,
UInt32, UInt32, UnsafePointer<Int16>, Int32, UnsafePointer<Int16>!,
UnsafePointer<Int32>!, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in a set of 16-bit source image planes by a matrix to produce a set of 8-bit destination image planes.

```
func vImageMatrixMultiply_PlanarF(UnsafeMutablePointer<UnsafePointer<v
Image_Buffer>?>, UnsafeMutablePointer<UnsafePointer<vImage_Buffer>?>,
UInt32, UInt32, UnsafePointer<Float>, UnsafePointer<Float>!, Unsafe
Pointer<Float>!, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in a set of 32-bit source image planes by a matrix to produce a set of 32-bit destination image planes.

## Multiplying interleaved pixels by a matrix

```
func vImageMatrixMultiply_ARGB8888(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafePointer<Int16>, Int32, UnsafePointer<
Int16>!, UnsafePointer<Int32>!, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in an interleaved four-channel, 8-bit source image by a matrix to produce an interleaved four-channel, 8-bit destination image.

```
func vImageMatrixMultiply_ARGBFFFF(UnsafePointer<vImage_Buffer>, Unsafe
Pointer<vImage_Buffer>, UnsafePointer<Float>, UnsafePointer<Float>!,
UnsafePointer<Float>!, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in an interleaved four-channel, 32-bit source image by a matrix to produce an interleaved four-channel, 32-bit destination image.

```
func vImageMatrixMultiply_ARGB8888ToPlanar8(UnsafePointer<vImage_Buffer
>, UnsafePointer<vImage_Buffer>, UnsafePointer<Int16>, Int32, Unsafe
Pointer<Int16>!, Int32, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in an interleaved four-channel, 8-bit source image by a matrix to produce a planar 8-bit destination image.

```
func vImageMatrixMultiply_ARGBFFFFToPlanarF(UnsafePointer<vImage_Buffer
>, UnsafePointer<vImage_Buffer>, UnsafePointer<Float>, UnsafePointer<
Float>!, Float, vImage_Flags) -> vImage_Error
```

Multiplies each pixel in an interleaved four-channel, 32-bit source image by a matrix to produce a planar 32-bit destination image.

# See Also

## Applying color transforms to images

- Transforming with lookup tables
  Use lookup tables to apply color transformations to images.
- Transforming with polynomials
  Use polynomials to apply color transformations to images.
- Transforming with a gamma function
  Use gamma functions to apply color transformations to images.
- Applying a flood fill to an image
  Fill connected components of an image with a new color.