Sample Code

# Adding a bokeh effect to images

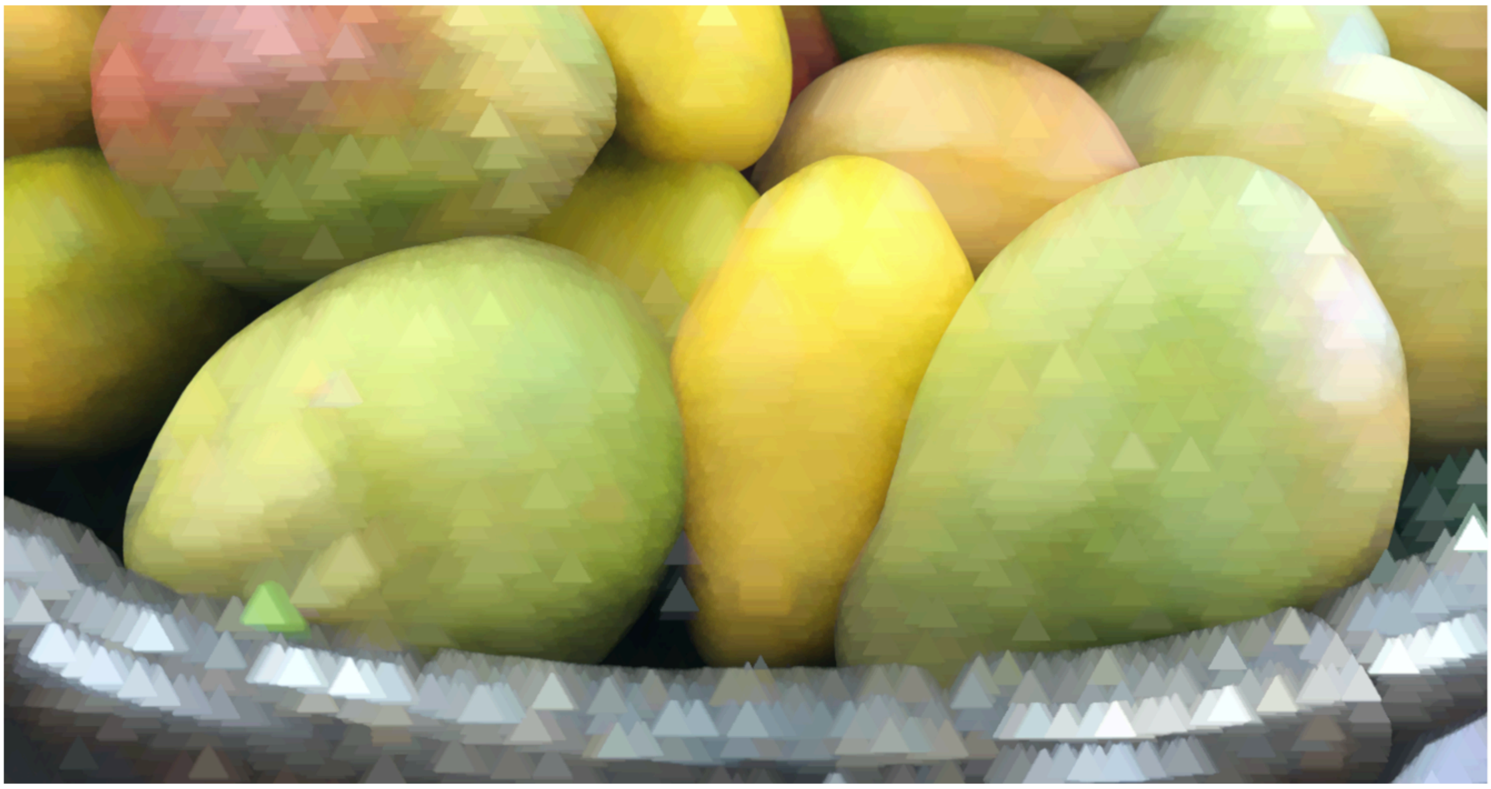
Simulate a bokeh effect by applying dilation.

Download

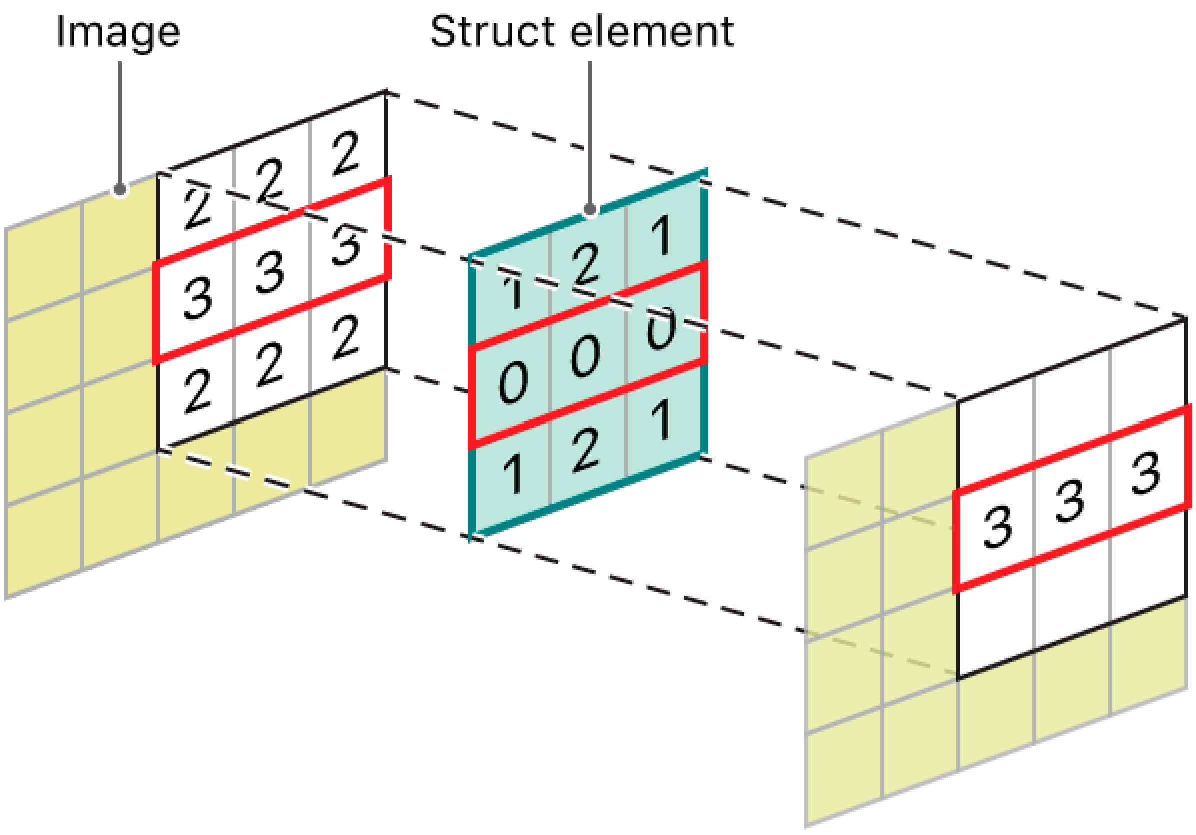
macOS 13.3+ | Xcode 14.0+

## Overview

This sample app creates a bokeh effect, where parts of an image that are out of focus adopt the shape of the lens's aperture. The app dynamically generates a polygon-shaped kernel — also known as a *structuring element* — and applies a morphology operation to an image based on that kernel. The following sample shows a photograph after the app has applied dilation with a triangular kernel:



Kernels are 1D or 2D matrices of values that the morphology operation subtracts from a corresponding pixel value in the image. The final value of each transformed pixel is either the lightest result (for dilation) or darkest result (for erosion) of each subtraction.



The following formula shows how a dilation operation calculates the value for the pixel at the center of the grid. The operation subtracts each of the nine kernel values from the image's corresponding pixel and returns the maximum value.

$$\begin{bmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 2 & 2 & 2 \end{bmatrix} \oplus \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \max \begin{pmatrix} (2-1) & (2-2) & (2-1) \\ (3-0) & (3-0) & (3-0) \\ (2-1) & (2-2) & (2-1) \end{pmatrix} = 3$$

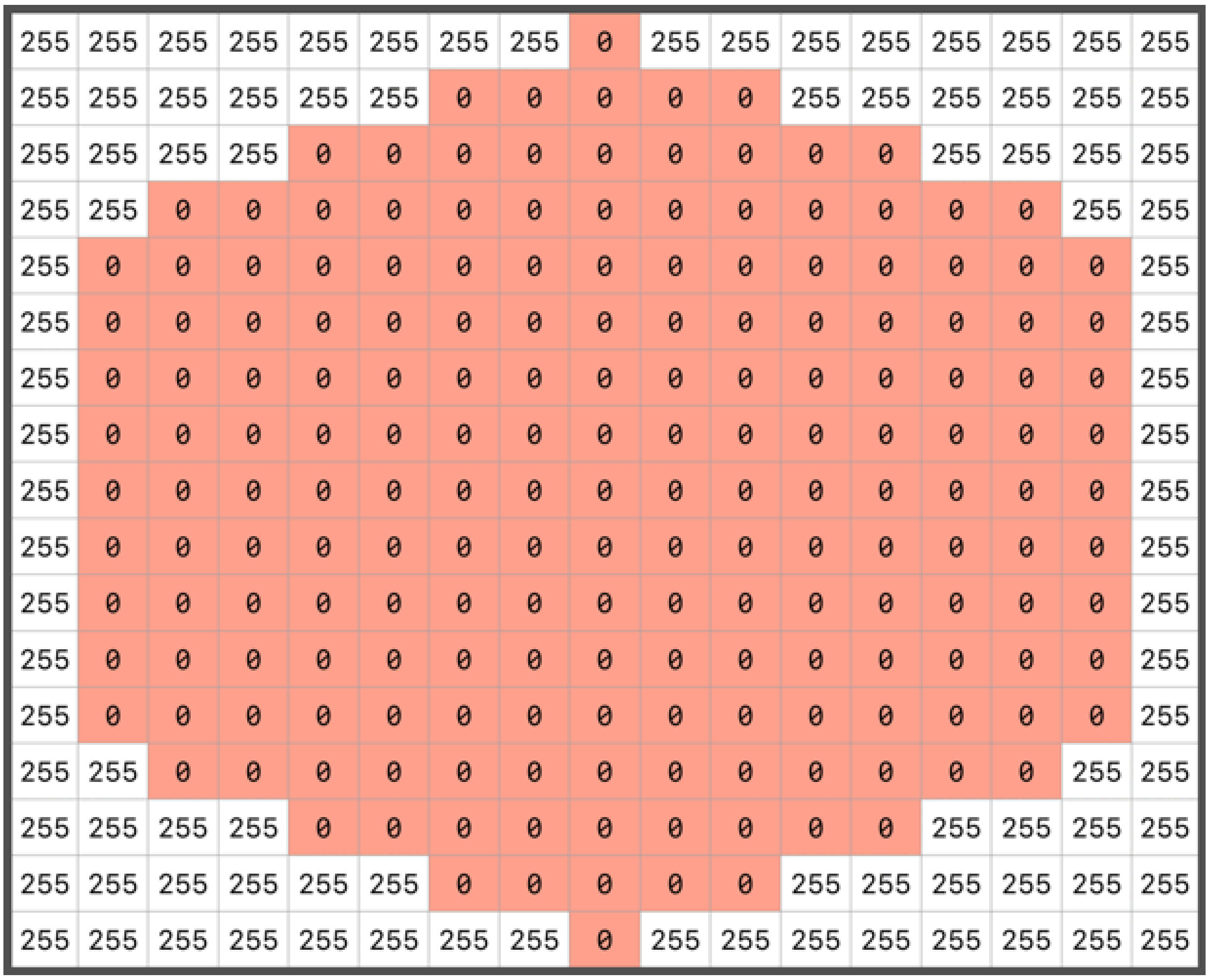
## Generate the structuring element

The `MorphologyTransformer.makeBokehStructuringElement(ofRadius:atAngle:withSides:)` method returns a `vImage.StructuringElement` structure. Within that structure, the `diaphragmBladeCount` variable defines the number of sides. For example, to create a hexagon-shaped bokeh effect, the sample app calls the `MorphologyTransformer.makeBokehStructuringElement(ofRadius:atAngle:withSides:)` method with the number of sides set to 6.

```swift
/// The number of edges on the bokeh polygon.
@Published var diaphragmBladeCount = 6.0 {
    didSet {
        Task(priority: .userInitiated) {
            await applyBokeh()
        }
    }
}
```

```swift
let bokeh = Self.makeBokehStructuringElement(ofRadius: Int(bokehRadius),
                                             atAngle: angle,
                                             withSides: Int(diaphragmBladeCount))
```

On return, `bokeh` contains the following values:



| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 0 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 0 | 0 | 0 | 0 | 0 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 |
| 255 | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 | 255 |
| 255 | 255 | 255 | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 0 | 0 | 0 | 0 | 0 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 0 | 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |

## Apply the dilation

To optimize the dilation operations, the sample app calls the planar morphology function, `applyMorphology(operation:destination:)`, concurrently on the three planar pixel buffers that represent the individual red, green, and blue channels.

To learn more about improving your app’s performance by converting image buffer formats from interleaved to planar, see Optimizing image-processing performance.

The following code calls the three functions inside a `withtaskgroup(of:returning:isolation:body:)` closure:

```
/// Apply dilation to the red channel.
group.addTask(priority: .userInitiated) { [self] in
    sourceRedBuffer.applyMorphology(operation: .dilate(structuringElement: bokeh),
                                    destination: destinationRedBuffer)
}

/// Apply dilation to the green channel.
group.addTask(priority: .userInitiated) { [self] in
    sourceGreenBuffer.applyMorphology(operation: .dilate(structuringElement: bokeh),
                                      destination: destinationGreenBuffer)
}

/// Apply dilation to the blue channel.
group.addTask(priority: .userInitiated) { [self] in
    sourceBlueBuffer.applyMorphology(operation: .dilate(structuringElement: bokeh),
                                     destination: destinationBlueBuffer)
}
```

On return, the destination buffer contains the dilation result:

## See Also

### Convolution and Morphology

{} Blurring an image

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

Convolution

Apply a convolution kernel to an image.

Morphology

Dilate and erode images.