Getting Started with Core Image

Session 510

David Hayward
Advanced Imaging Team

What We Will Discuss Today

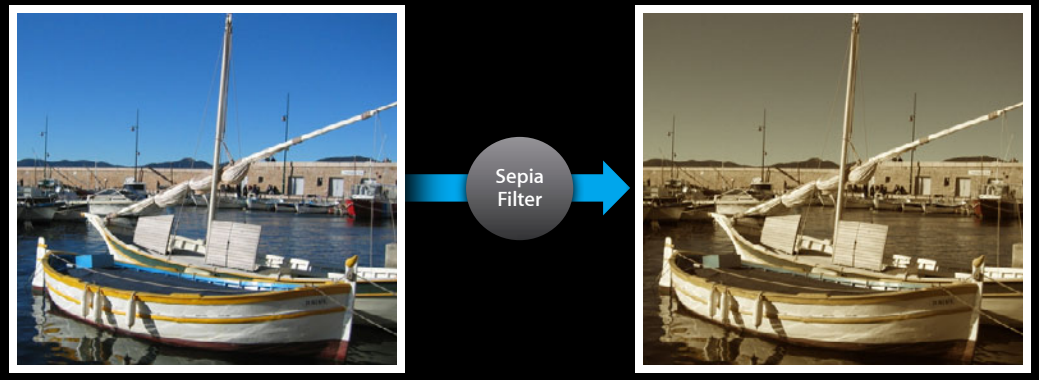
- Introduction to Core Image
  - Key concepts
  - Built-in filters
- Using Core Image API
  - Key classes
  - Platform specifics
  - Creating a CIImage, applying CIFilters, rendering through a CIContext
- Filter recipes

Introduction to Core Image

Key concepts

Basic Concept

Filters perform per pixel operations on an image



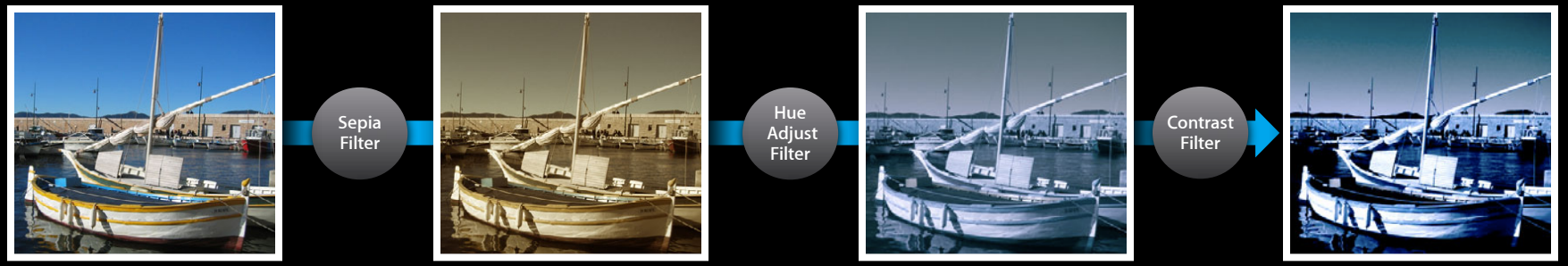
Original

Result

The final result is a new image

Basic Concept

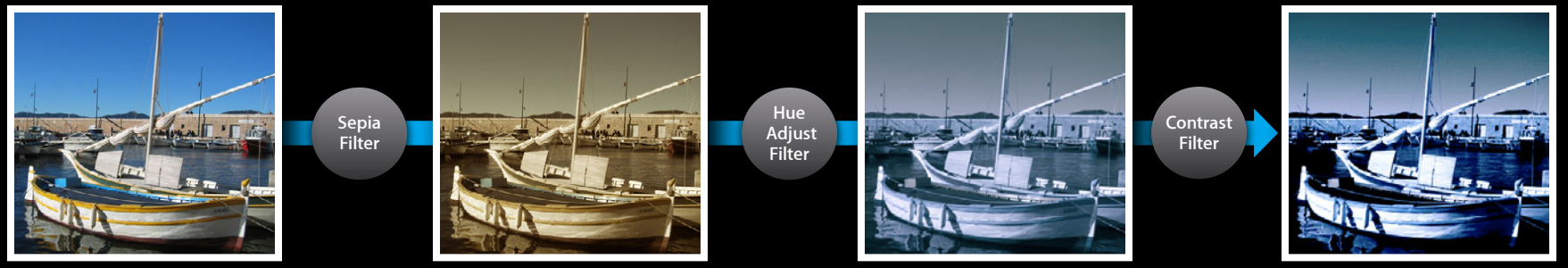
Filters can be chained together



This allows for complex effects

Basic Concept

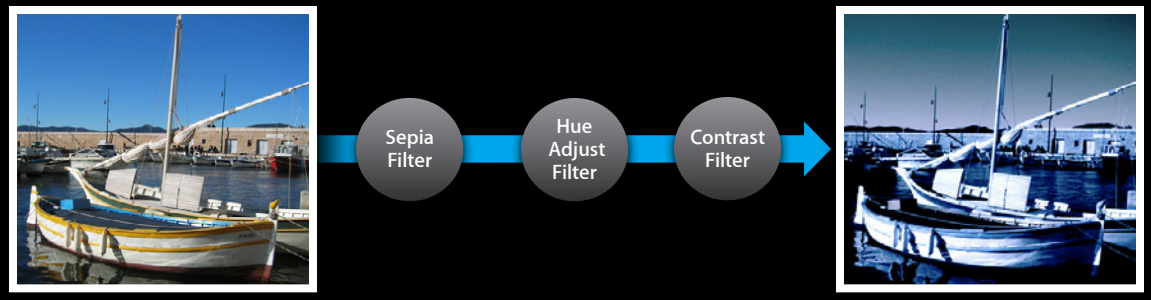
Filters can be chained together



This allows for complex effects

Basic Concept

Filter chains are concatenated



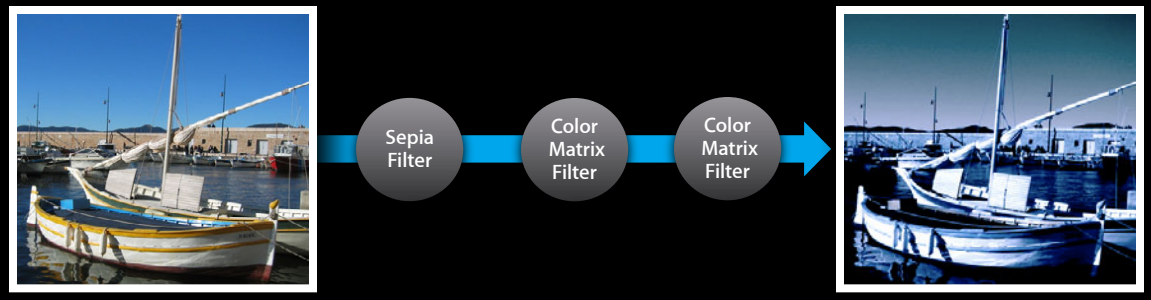
Original

Result

This eliminates intermediate buffers

Basic Concept

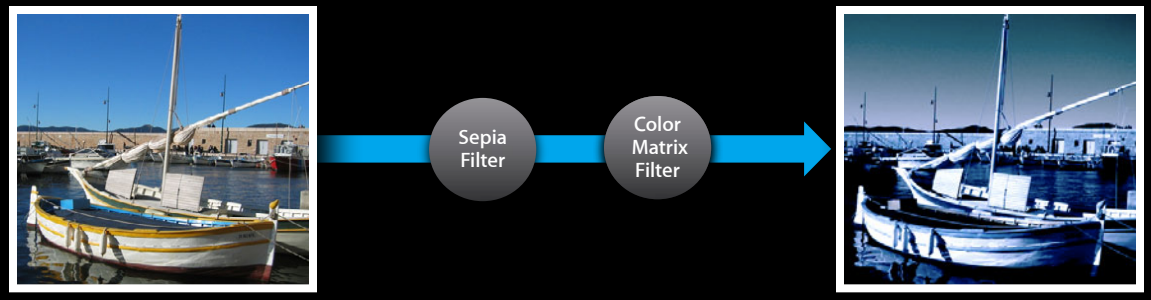
Filter chains are optimized at render time



This further improves performance

Basic Concept

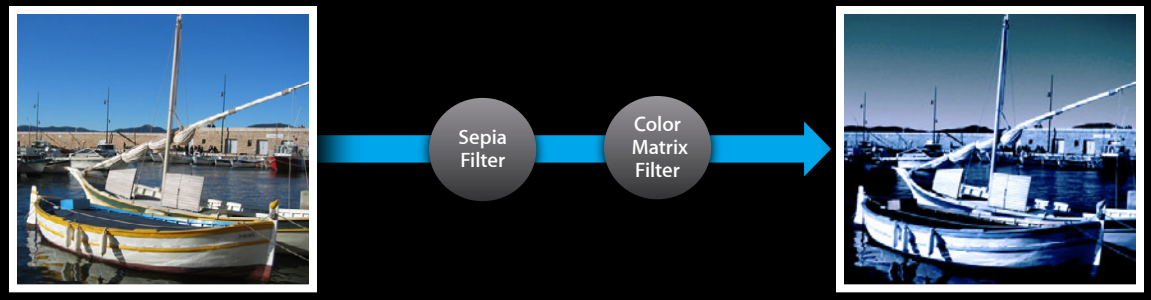
Filter chains are optimized at render time



This further improves performance

Basic Concept

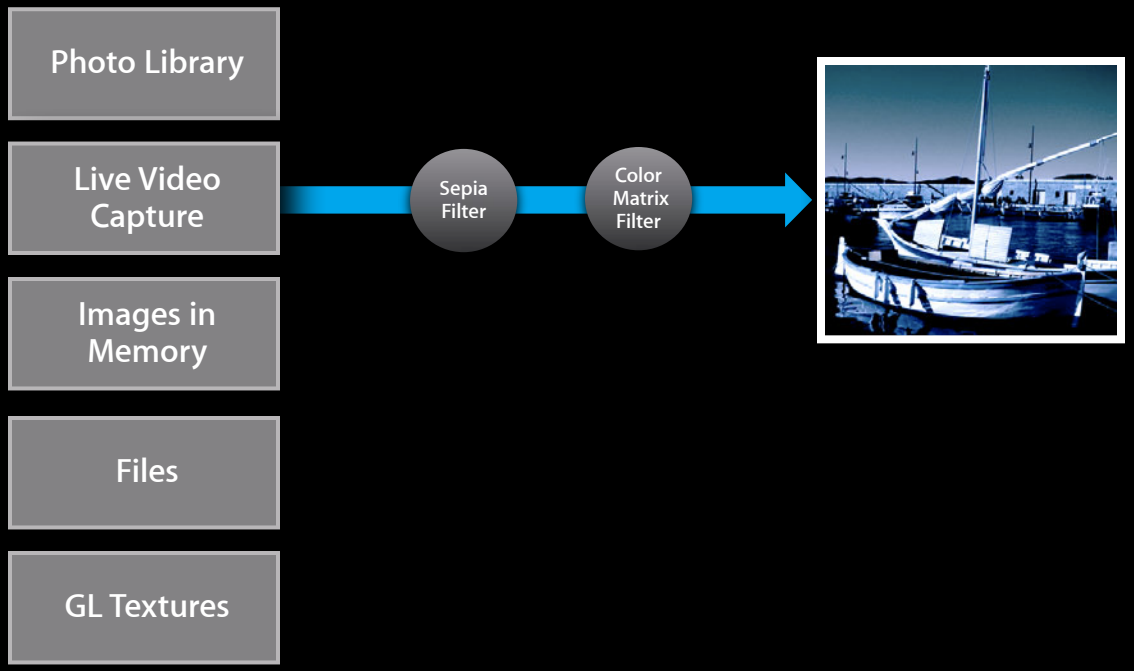
Filter chains are optimized at render time



This further improves performance

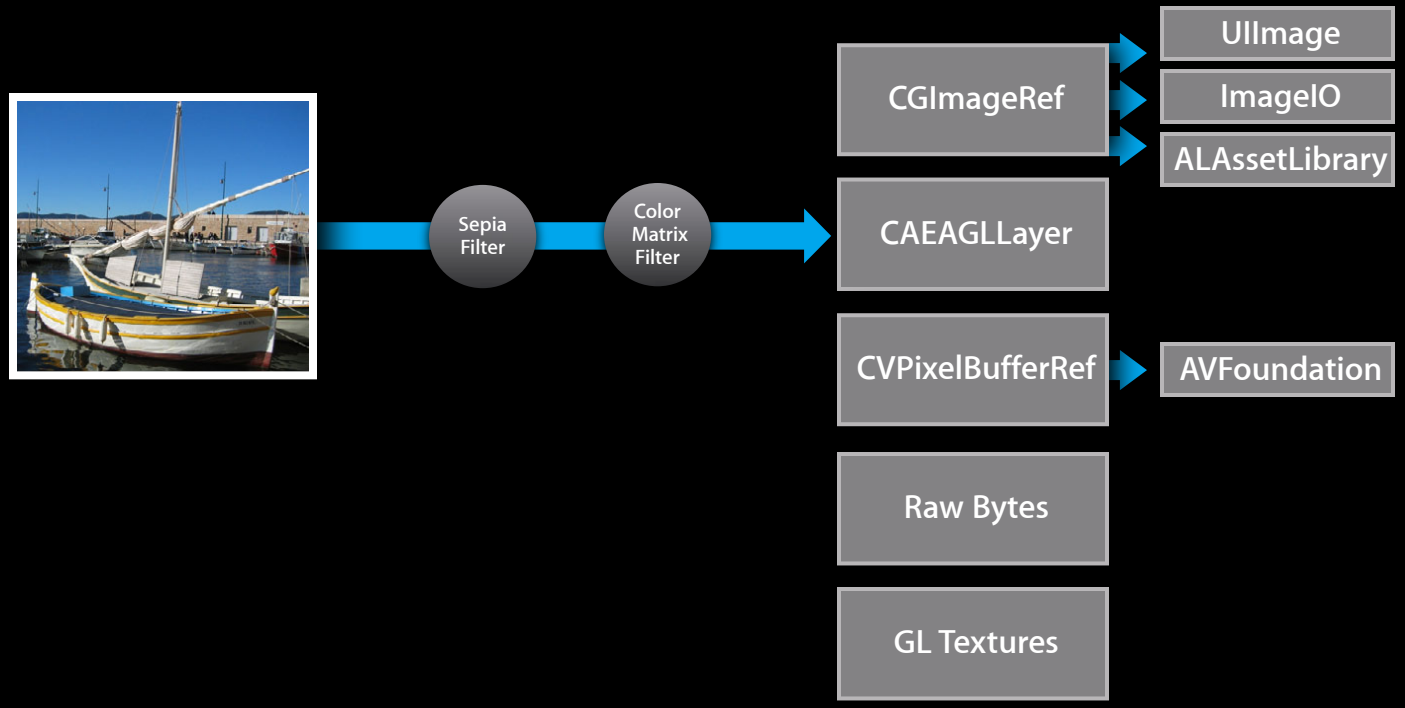
Basic Concept

Flexible inputs



Basic Concept

Flexible outputs



Introduction to Core Image

Built-in filters on iOS

93 Filters in iOS 6

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

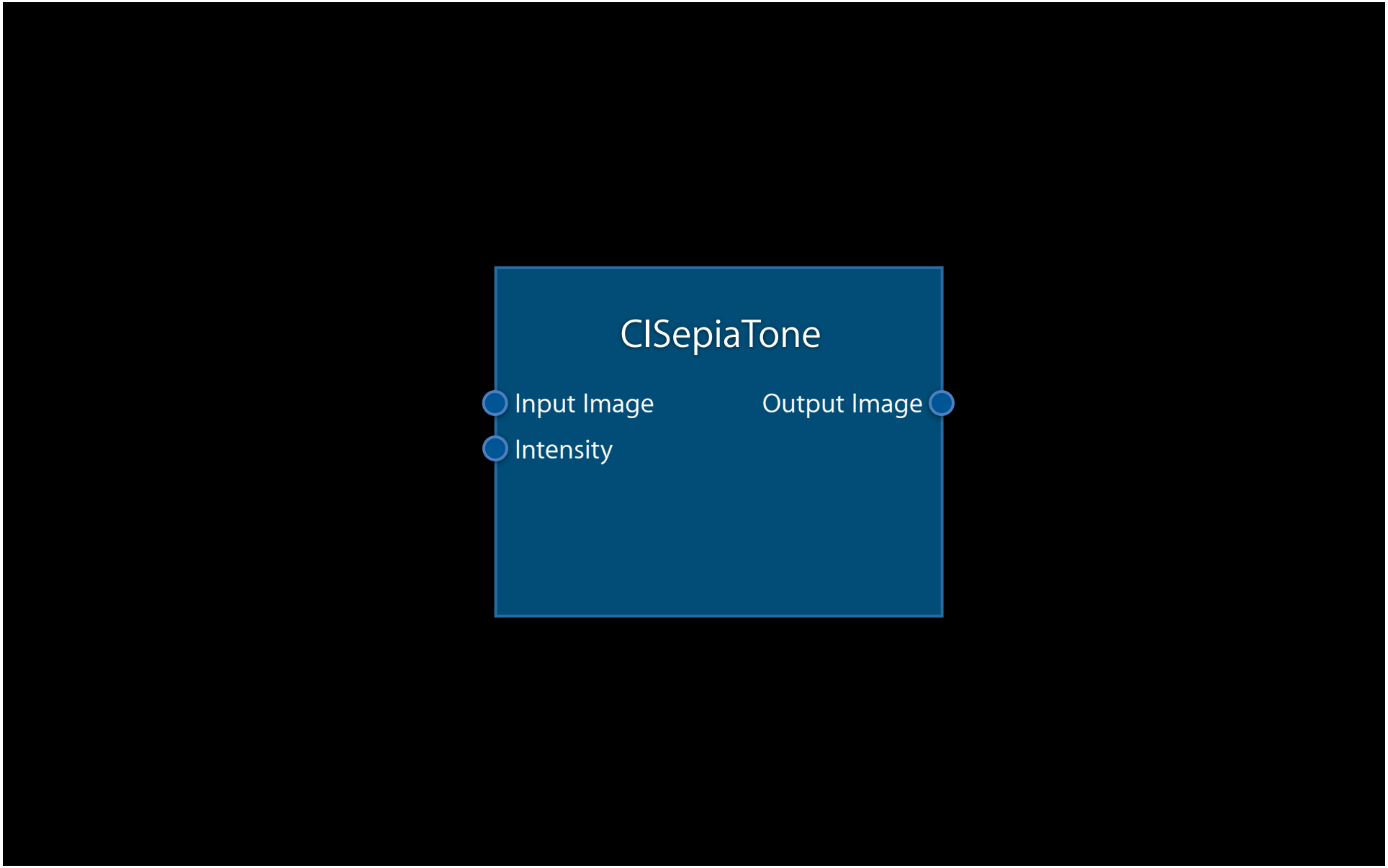
Color effects and adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Color effects and adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust


CISepiaTone
Input Image
Intensity
Output Image

93 Filters in iOS 6

Color effects and adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

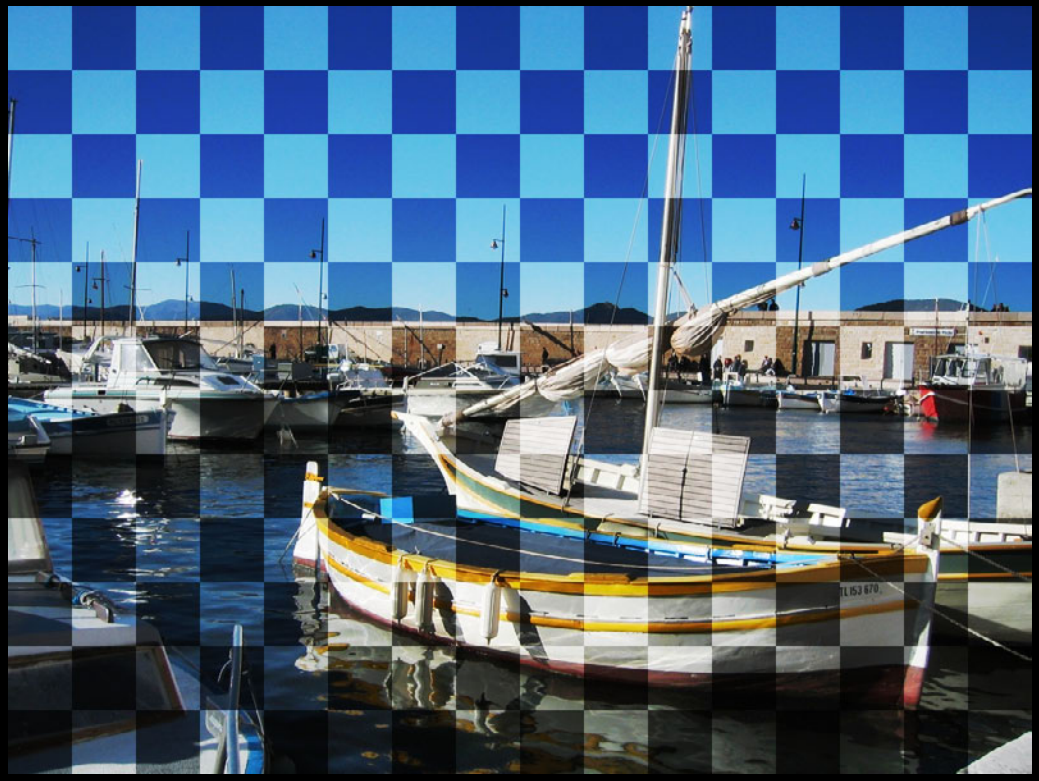
Compositing operations

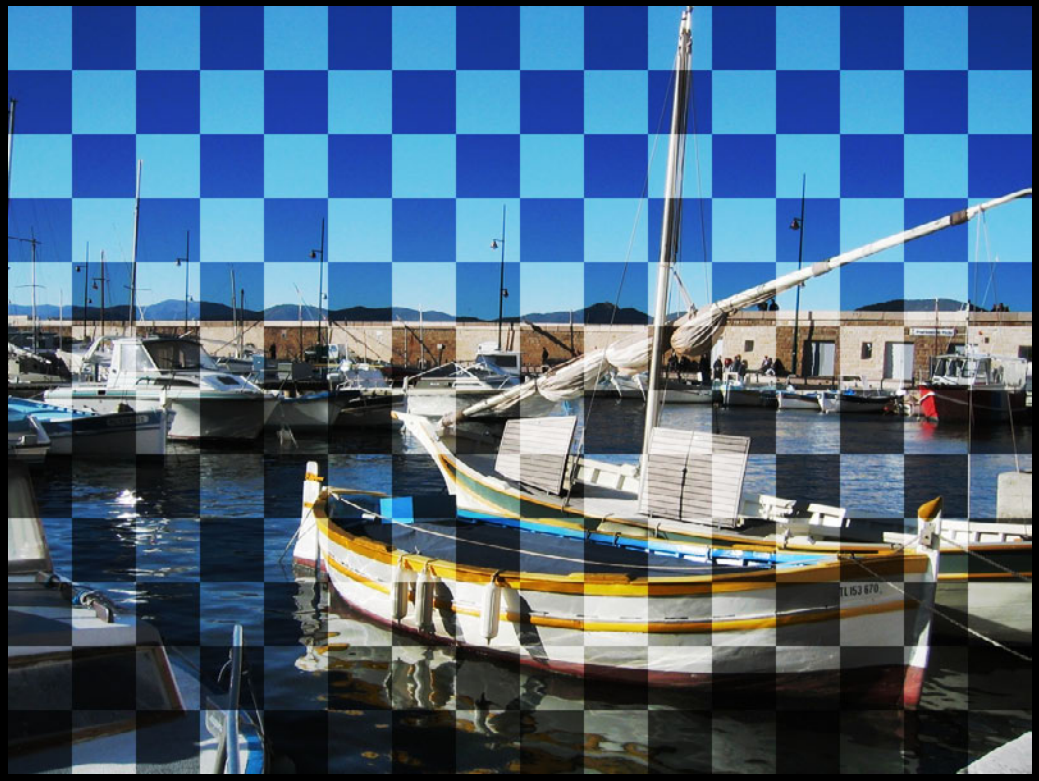
CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

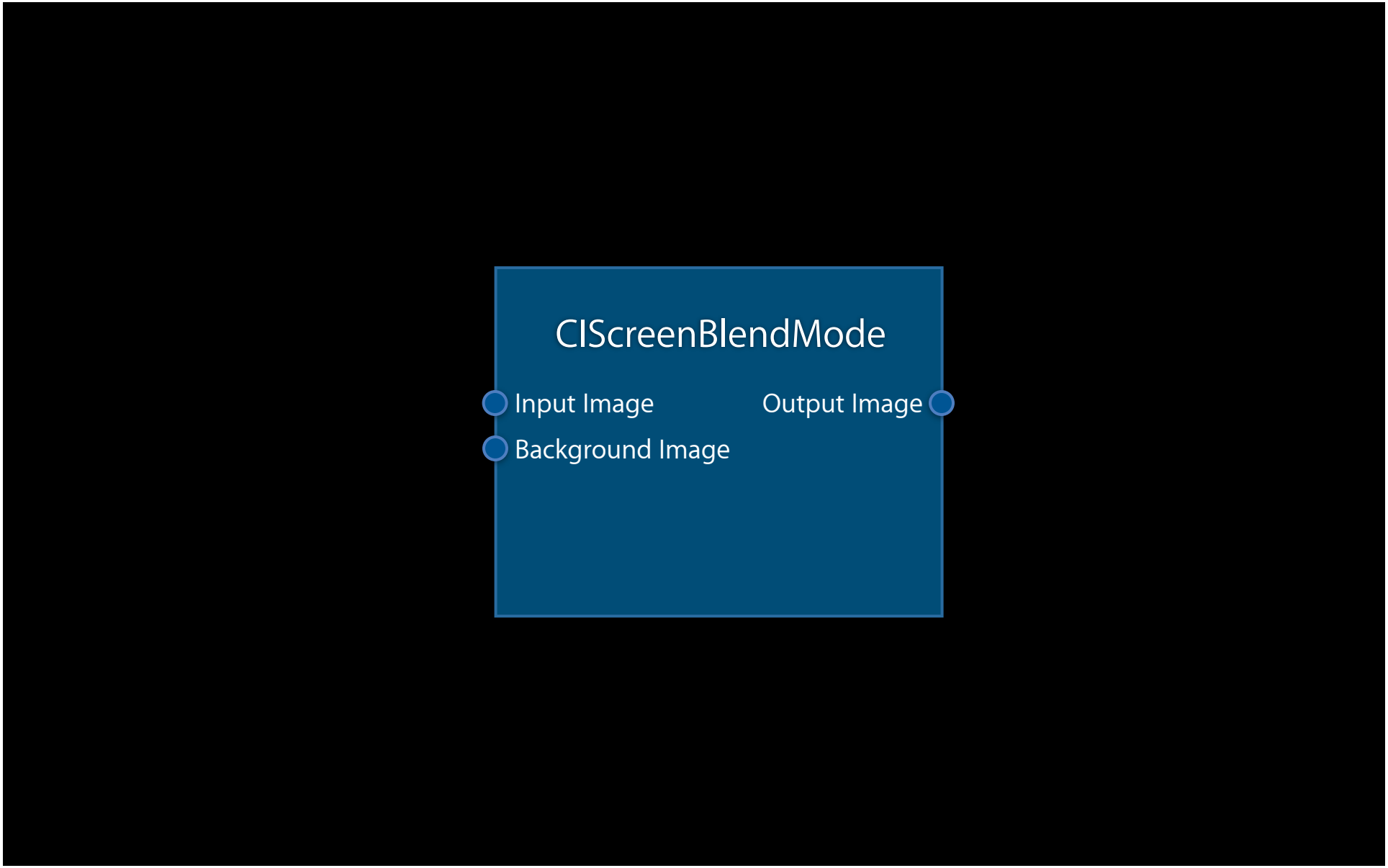
93 Filters in iOS 6

Compositing operations

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CIScreenBlendMode
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust






CIScreenBlendMode
Input Image
Background Image
Output Image

93 Filters in iOS 6

Compositing operations

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Geometry adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Geometry adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Geometry adjustments

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Tile effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Tile effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Tile effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Distortion effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Distortion effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Distortion effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Blur and sharpen effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Blur and sharpen effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Blur and sharpen effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Stylize filters

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Stylize filters

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHighlightShadowAdjust
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Stylize filters

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Halftone effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Halftone effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Halftone effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

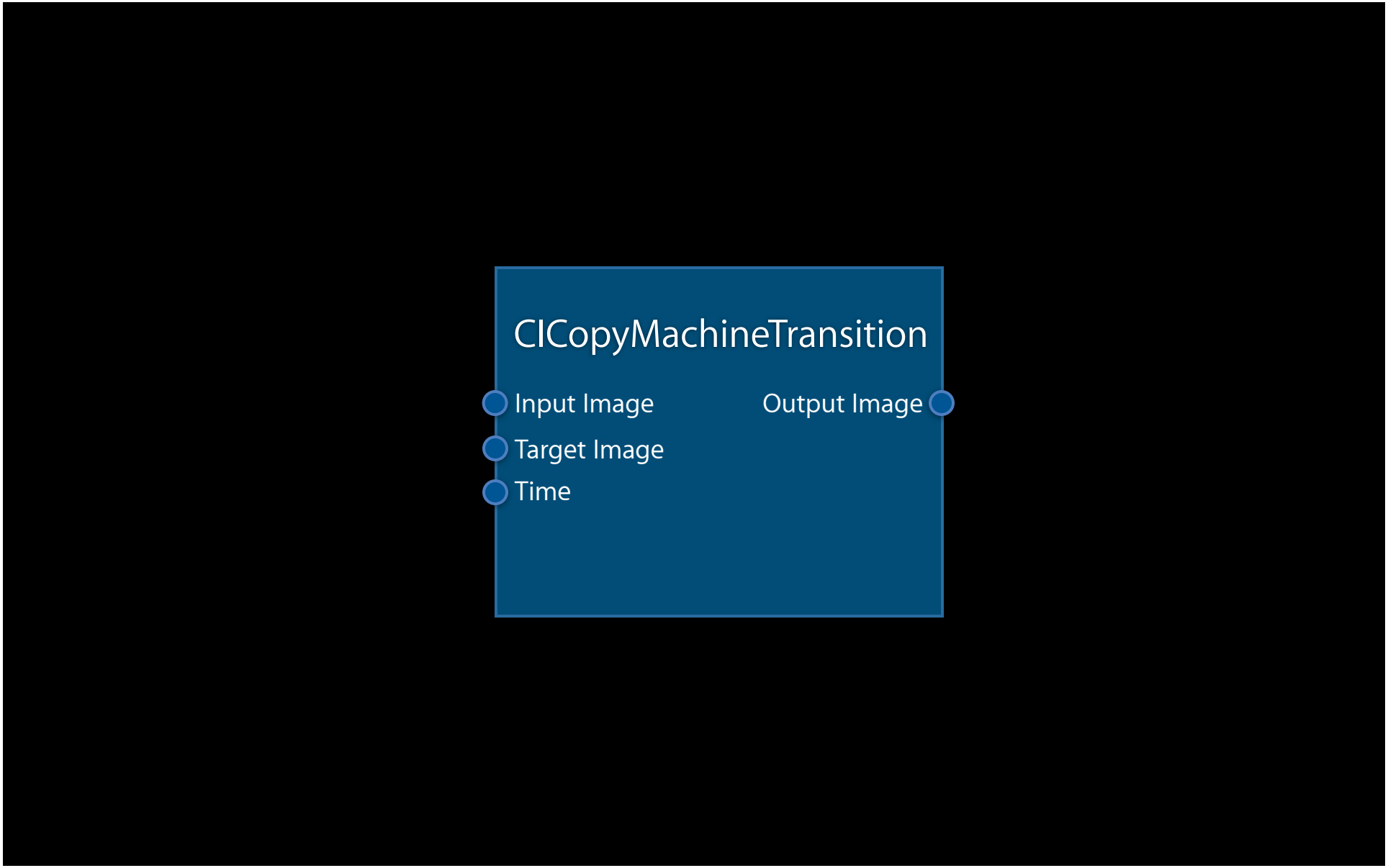
Transition effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

Transition effects

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CICopyMachineTransition
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust


CICopyMachineTransition
Input Image
Target Image
Time
Output Image

93 Filters in iOS 6

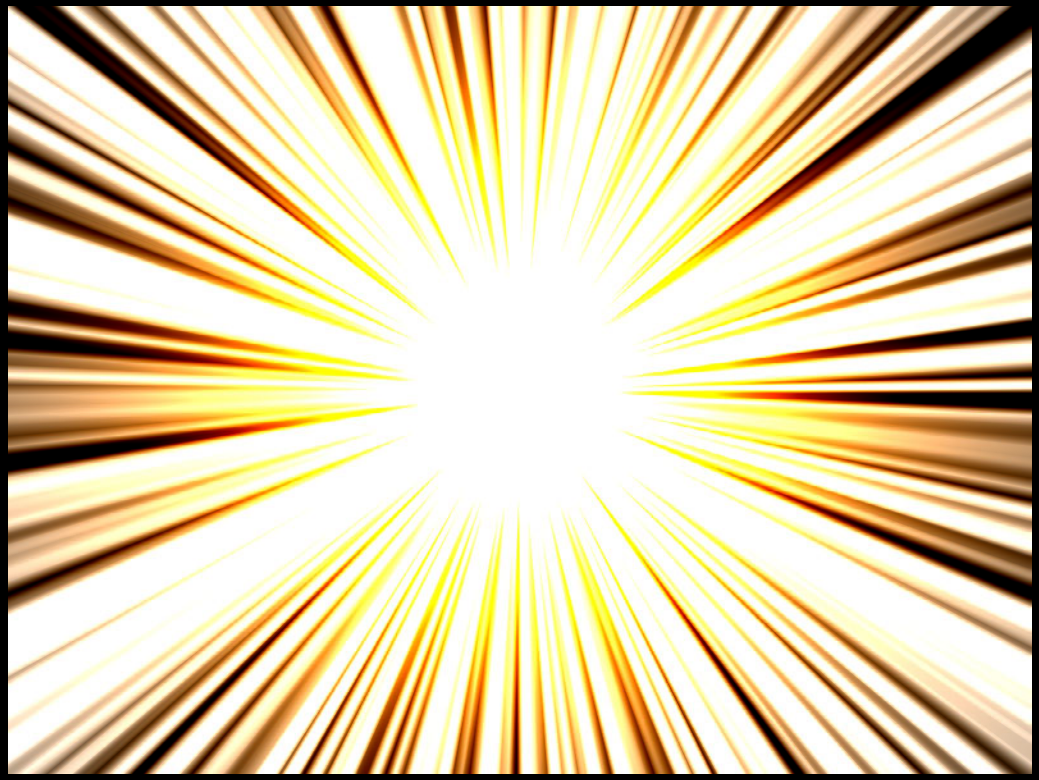
Transition effects

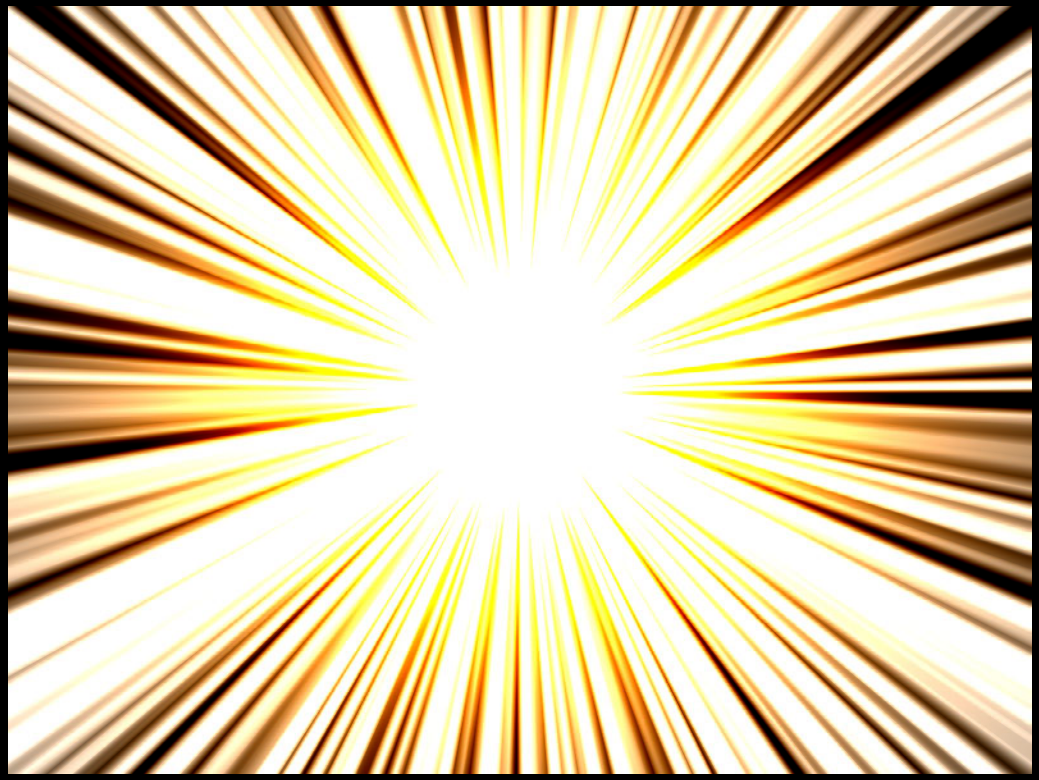
CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

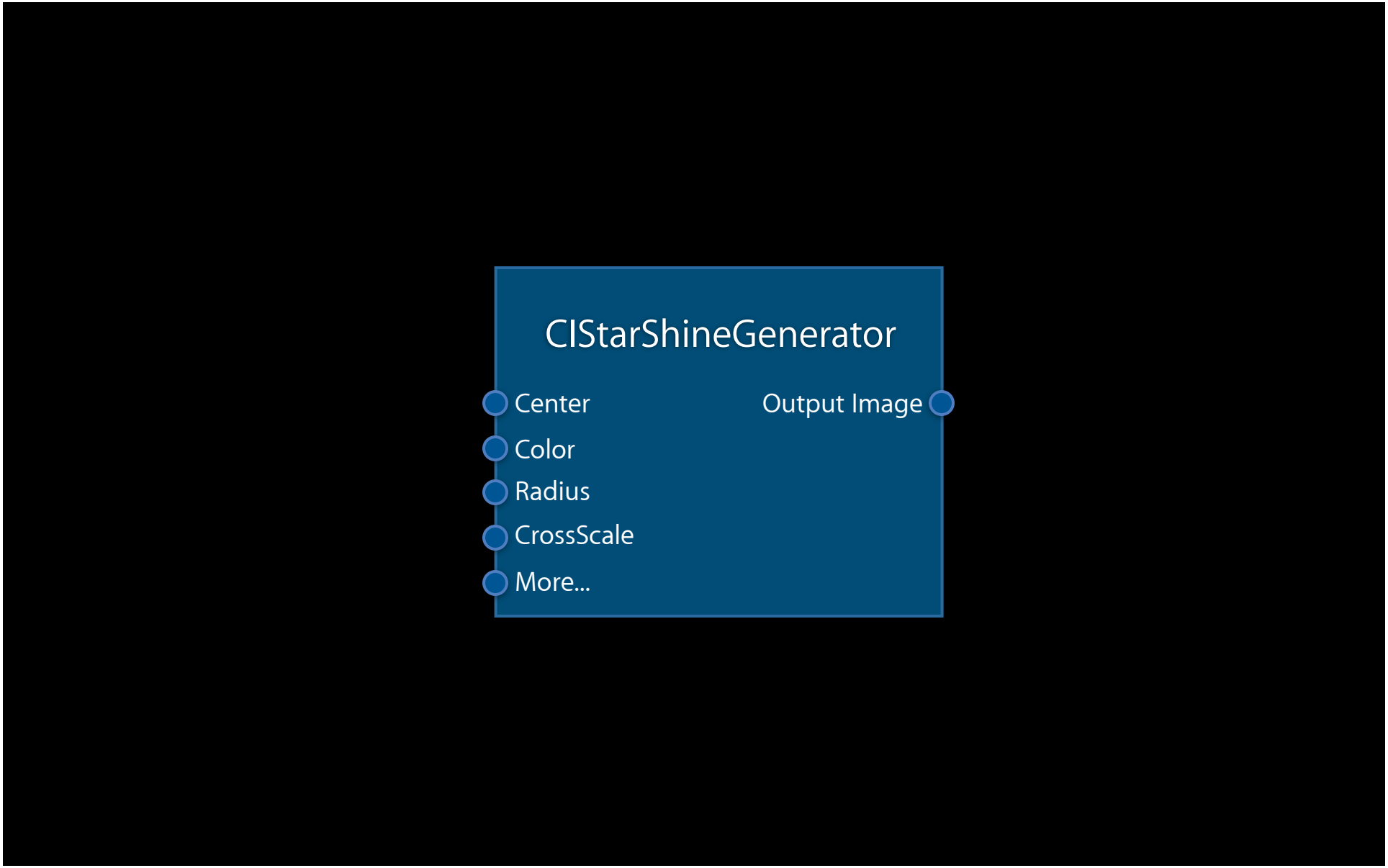
93 Filters in iOS 6

Generators

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust






CIStarShineGenerator
Center
Color
Radius
CrossScale
More...
Output Image

93 Filters in iOS 6

Generators

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

93 Filters in iOS 6

CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

Demo

Core Image in action

Using the Core Image API

Core Image Classes

- CIFilter
  - A mutable object that represents an effect
  - Has image or numeric input parameters
  - Produces one output image based on current inputs

Core Image Classes

- CIFilter
  - A mutable object that represents an effect
  - Has image or numeric input parameters
  - Produces one output image based on current inputs
- CIImage
  - An immutable object that represents the recipe for an image
  - Can represent a file from disk or the output of a CIFilter

Core Image Classes

- CIFilter
  - A mutable object that represents an effect
  - Has image or numeric input parameters
  - Produces one output image based on current inputs
- CIImage
  - An immutable object that represents the recipe for an image
  - Can represent a file from disk or the output of a CIFilter
- CIContext
  - An object through which Core Image draws results
  - Can be based on CPU or GPU

Platform Specifics

iOS	Mac OS X

Platform Specifics

	iOS	Mac OS X
Filters	93 built-in filters	130 built-in filters + developer extendable

Platform Specifics

	iOS	Mac OS X
Filters	93 built-in filters	130 built-in filters + developer extendable
Core API	CIFilter CIImage CIContext	CIFilter CIImage CIContext CIKernel CIFilterShape

Platform Specifics

	iOS	Mac OS X
Filters	93 built-in filters	130 built-in filters + developer extendable
Core API	CIFilter CIImage CIContext	CIFilter CIImage CIContext CIKernel CIFilterShape
Performance	Render-time optimizations of filter graph	

Platform Specifics

	iOS	Mac OS X
Filters	93 built-in filters	130 built-in filters + developer extendable
Core API	CIFilter CIImage CIContext	CIFilter CIImage CIContext CIKernel CIFilterShape
Performance	Render-time optimizations of filter graph	
Rendering	CPU or OpenGL ES 2.0	CPU or OpenGL

Core Image Usage

```
// Create a CIImage object
CIImage *image = [CIImage imageWithContentsOfURL:myURL];

// Create a CIFilter object and set input values
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"];
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8f] forKey:@"inputIntensity"];

// Create a CIContext object
CIContext *context = [CIContext contextWithOptions:nil];

// Render the filter output image into a CGImage
CIImage *result = [filter valueForKey:kCIOutputImageKey];
CGImageRef cgImage = [context createCGImage:result fromRect:[result extent]];
```

Core Image Usage

```
// Create a CIImage object
CIImage *image = [CIImage imageWithContentsOfURL:myURL];

// Create a CIFilter object and set input values
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"];
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8f] forKey:@"inputIntensity"];

// Create a CIContext object
CIContext *context = [CIContext contextWithOptions:nil];

// Render the filter output image into a CGImage
CIImage *result = [filter valueForKey:kCIOutputImageKey];
CGImageRef cgImage = [context createCGImage:result fromRect:[result extent]];
```

Core Image Usage

```
// Create a CIImage object
CIImage *image = [CIImage imageWithContentsOfURL:myURL];

// Create a CIFilter object and set input values
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"];
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8f] forKey:@"inputIntensity"];

// Create a CIContext object
CIContext *context = [CIContext contextWithOptions:nil];

// Render the filter output image into a CGImage
CIImage *result = [filter valueForKey:kCIOutputImageKey];
CGImageRef cgImage = [context createCGImage:result fromRect:[result extent]];
```

Core Image Usage

```
// Create a CIImage object
CIImage *image = [CIImage imageWithContentsOfURL:myURL];

// Create a CIFilter object and set input values
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"];
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8f] forKey:@"inputIntensity"];

// Create a CIContext object
CIContext *context = [CIContext contextWithOptions:nil];

// Render the filter output image into a CGImage
CIImage *result = [filter valueForKey:kCIOutputImageKey];
CGImageRef cgImage = [context createCGImage:result fromRect:[result extent]];
```

Core Image Usage

```
// Create a CIImage object
CIImage *image = [CIImage imageWithContentsOfURL:myURL];

// Create a CIFilter object and set input values
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"];
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8f] forKey:@"inputIntensity"];

// Create a CIContext object
CIContext *context = [CIContext contextWithOptions:nil];

// Render the filter output image into a CGImage
CIImage *result = [filter valueForKey:kCIOutputImageKey];
CGImageRef cgImage = [context createCGImage:result fromRect:[result extent]];
```

1 Creating a CIImage

Initializing a CIImage

- A CIImage can be initialized from:
  - ImageIO supported formats

```
+imageWithURL:options:
+imageWithData:options:
```

  - Other image types

```
+imageWithCGImage:options:

+imageWithCVPixelBuffer:options:
```

Only on iOS

```
+imageWithCVImageBuffer:options:
+imageWithIOSurface:options:
```

Only on Mac OS

```
+imageWithTexture:size:flipped:colorSpace:
```

  - Raw Data

```
+imageWithBitmapData:bytesPerRow:size:format:colorSpace:
```

Initializing a CIImage's Colorspace

- On Mac OS
  - A CIImage can be tagged with any colorspace
    - If tagged, pixels are converted to linear working space before filtering
- On iOS
  - A CIImage can be tagged with "Device RGB" colorspace
    - If tagged, pixels are gamma corrected to linear before filtering
- Use the `kCIImageColorSpace` option to override the default colorspace
  - Set value of this key to `[NSNull null]` to leave image unmanaged

Initializing a CIImage's Metadata

- New `[filter properties]` method to get metadata properties from an image
  - Returns dictionary with same key/values as `CGImageSourceCopyPropertiesAtIndex`
  - One notable key is `kCGImagePropertyOrientation`
- Properties are automatic if you use `imageWithURL:` or `imageWithData:`
  - Otherwise properties can be specified using `kCIImageProperties` option

2 Applying CIFilters

Filter Application

- Query Core Image for the list of built-in filters

```
NSArray *list = [CIFilter filterNamesInCategory:kCICategoryBuiltIn];
```

- Filters are instantiated by name

```
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone"]
```

- Calling `[filter attributes]` will tell you about the filter's inputs
  - The key for each input
  - The expected data type of each input
    - NSNumber, CIVector, CIImage, etc.
  - Common values of each input
    - Default, identity, minimum, and maximum

Filter Application

- Input on filters are set using key value conventions

```
[filter setValue:image forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithFloat:0.8] forKey:@"inputIntensity"];
```

- Output of filter is through outputImage property:

```
output = [filter valueForKey:kCIOutputImageKey];
```

```
output = [filter outputImage];
output = filter.outputImage;
```

Only on iOS

- Shortcut:

```
output = [CIFilter filterWithName:@"CISepiaTone" keysAndValues:
            kCIInputImageKey, image,
            @"inputIntensity", [NSNumber numberWithFloat: 0.8f],
            nil].outputImage;
```

Filter Application

Chaining multiple filters

Filter Application

Chaining multiple filters

Filter Application

Chaining multiple filters

- Apply first filter to input image

```
output = [CIFilter filterWithName:@"CISepiaTone" keysAndValues:
         kCIInputImageKey, image,
         @"inputIntensity", [NSNumber numberWithFloat:0.8],
         nil].outputImage;
```

Filter Application

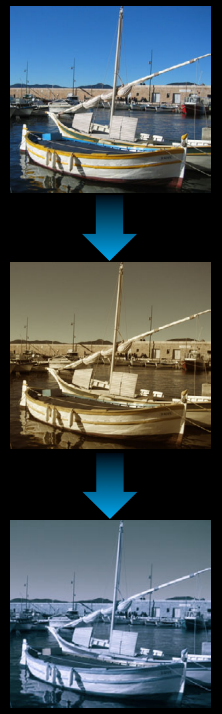
Chaining multiple filters

- Apply first filter to input image

```
output = [CIFilter filterWithName:@"CISepiaTone" keysAndValues:
         kCIInputImageKey, image,
         @"inputIntensity", [NSNumber numberWithFloat:0.8],
         nil].outputImage;
```

- Apply next filter

```
output = [CIFilter filterWithName:@"CIHueAdjust" keysAndValues:
         kCIInputImageKey, output,
         @"inputAngle", [NSNumber numberWithFloat:0.8],
         nil].outputImage;
```



Filter Application

Chaining multiple filters

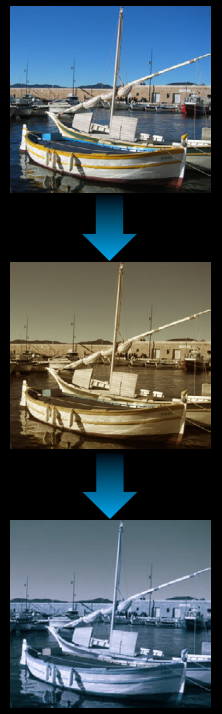
- Apply first filter to input image

```
output = [CIFilter filterWithName:@"CISepiaTone" keysAndValues:
         kCIInputImageKey, image,
         @"inputIntensity", [NSNumber numberWithFloat:0.8],
         nil].outputImage;
```

- Apply next filter

```
output = [CIFilter filterWithName:@"CIHueAdjust" keysAndValues:
         kCIInputImageKey, output,
         @"inputAngle", [NSNumber numberWithFloat:0.8],
         nil].outputImage;
```

- No pixel processing is performed while building the chain
  - That work is deferred until render is requested



3 Rendering Through a CIContext

Rendering Core Image Output

CIContext

- Renders a CIImage into its destination
  - CGImageRef
  - EAGLContext
  - CVPixelBufferRef
  - void*
- Common use cases
  - Displaying in a UIImageView
  - Save the result into the photo library
  - Displaying in a CAEAGLLayer-backed view
  - Passing results back to Core Video

Rendering Core Image Output

Displaying in a UIImageView

- Create CIContext
- Get outputImage from the last filter in chain
- Render outputImage to a CGImageRef
- Tell the UIImageView to use a UIImage for the CGImage

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                        orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                                 orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                                 orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                                 orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                                 orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

```
// Create the CIContext to render into
CIContext *context = [CIContext context];

// Get outputImage from the last filter in chain
CIImage *ciimage = [filter outputImage];

// Render the CIImage into a CGImageRef
CGImageRef cgimg = [context createCGImage:ciimage fromRect:[ciimage extent]];

// Create a UIImage from the CGImageRef
UIImage *uiimage = [UIImage imageWithCGImage:cgimg scale:1.0f
                                 orientation:ui_orientation([ciimage properties])];
CGImageRelease(cgimg);

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

- Shortcut: UIImage has built-in support for CIImage

```
// Create a UIImage using the filter output
UIImage *image = [UIImage imageWithCIImage:filter.outputImage];

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

- Shortcut: UIImage has built-in support for CIImage

```
// Create a UIImage using the filter output
UIImage *image = [UIImage imageWithCIImage:filter.outputImage];

// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Displaying in a UIImageView

- Shortcut: UIImage has built-in support for CIImage

```
// Create a UIImage using the filter output
UIImage *image = [UIImage imageWithCIImage:filter.outputImage];
```

```
// Use the UIImage in an UIImageView
imageView.image = uiimage;
```

Rendering Core Image Output

Photo library

- Create a CPU context
  - Why?
    - CPU context supports larger input and output images
    - Will allow your app to do processing in the background
  - Render the CIImage into a CGImageRef
  - Add the CGImageRef to the photo library

Rendering Core Image Output

Photo library

```
// Create a CPU context
NSDictionary *options = @{ kCIContextUseSoftwareRenderer : @YES };
CIContext *context = [CIContext contextWithOptions:options];

// Create a CGImage from the CIImage
CIImage *outputImage = [filter outputImage];
CGImageRef cgimage = [cpu_context createCGImage:outputImage
                                       fromRect:[outputImage extent]];

// Add the CGImage to the photo library
ALAssetsLibrary *library = [ALAssetsLibrary new];
[library writeImageToSavedPhotosAlbum:cgimage
                             metadata:[outputImage properties]
     completionBlock:^(NSURL *assetURL NSError *error) {
                CGImageRelease(cgimg);
      }];
```

Rendering Core Image Output

Photo library

```
// Create a CPU context
NSDictionary *options = @{ kCIContextUseSoftwareRenderer : @YES };
CIContext *context = [CIContext contextWithOptions:options];
```

```
// Create a CGImage from the CIImage
CIImage *outputImage = [filter outputImage];
CGImageRef cgimage = [cpu_context createCGImage:outputImage
                                       fromRect:[outputImage extent]];

// Add the CGImage to the photo library
ALAssetsLibrary *library = [ALAssetsLibrary new];
[library writeImageToSavedPhotosAlbum:cgimage
                             metadata:[outputImage properties]
     completionBlock:^(NSURL *assetURL NSError *error) {
                CGImageRelease(cgimg);
      }];
```

Rendering Core Image Output

Photo library

```
// Create a CPU context
NSDictionary *options = @{ kCIContextUseSoftwareRenderer : @YES };
CIContext *context = [CIContext contextWithOptions:options];

// Create a CGImage from the CIImage
CIImage *outputImage = [filter outputImage];
CGImageRef cgimage = [cpu_context createCGImage:outputImage
                                       fromRect:[outputImage extent]];

// Add the CGImage to the photo library
ALAssetsLibrary *library = [ALAssetsLibrary new];
[library writeImageToSavedPhotosAlbum:cgimage
                             metadata:[outputImage properties]
     completionBlock:^(NSURL *assetURL NSError *error) {
                CGImageRelease(cgimg);
      }];
```

Rendering Core Image Output

Photo library

```
// Create a CPU context
NSDictionary *options = @{ kCIContextUseSoftwareRenderer : @YES };
CIContext *context = [CIContext contextWithOptions:options];

// Create a CGImage from the CIImage
CIImage *outputImage = [filter outputImage];
CGImageRef cgimage = [cpu_context createCGImage:outputImage
                                       fromRect:[outputImage extent]];
```

```
// Add the CGImage to the photo library
ALAssetsLibrary *library = [ALAssetsLibrary new];
[library writeImageToSavedPhotosAlbum:cgimage
                             metadata:[outputImage properties]
     completionBlock:^(NSURL *assetURL NSError *error) {
                CGImageRelease(cgimg);
      }];
```

Tips and Best Practices

- CIImages and CIFilter are autoreleased
  - Use autorelease pools to reduce memory pressure
- Don’t create a CIContext every time you render
- Core Animation and Core Image both can use the GPU
  - Avoid CA animations while rendering CIImages with a GPU context

Tips and Best Practices

- CPU and GPU CIContexts have limits on image sizes
  - Check the context limits by using:

```
-(CGSize) inputImageMaximumSize;
-(CGSize) outputImageMaximumSize;
```

- Use smaller images when possible
  - Performance scales with the number of output pixels
  - You can use Core Graphics or ImageIO APIs to crop or down-sample

```
CGImageCreateWithImageInRect
CGImageSourceCreateThumbnailAtIndex
```

Filter Recipes

Alexandre Naaman
Employee, Advanced Imaging Team

Filter Recipes

- Core Image on iOS 93 filters
  - You can creatively combine filters to achieve many other effects
  - CoreImage will efficiently combine the filter graph
  - Custom kernels are not supported
    - But subclassing CIFilter is allowed

Filter Recipes

Subclassing CIFilter

- A filter recipe can be coded as a subclass of CIFilter
- Your CIFilter subclass will need
  - Declare `@properties` for its input parameters such as `inputImage`
  - Override `-(void) setDefaults`
  - Override `-(CIImage*) outputImage`
  - Core Image implements some of its built-in CIFilters using this technique

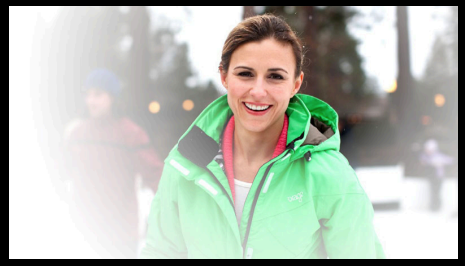
Filter Recipes

Subclassing CIFilter: An example from CI's source

```
@interface CIColorInvert: CIFilter {
    CIImage *inputImage;
}
@property (retain, nonatomic) CIImage *inputImage;
@end

@implementation CIColorInvert
@synthesize inputImage;
- (CIImage *)outputImage {
    return [CIFilter filterWithName:@"CIColorMatrix" keysAndValues:
        kCIInputImageKey, inputImage,
        @"inputRVector", [CIVector vectorWithX:-1 Y:0 Z:0],
        @"inputGVector", [CIVector vectorWithX:0 Y:-1 Z:0],
        @"inputBVector", [CIVector vectorWithX:0 Y:0 Z:-1],
        @"inputBiasVector", [CIVector vectorWithX:1 Y:1 Z:1],
        nil].outputImage;
}
```

Let’s Get Cooking

Let's Get Cooking



Recipe One: Chroma Key

Recipe One: Chroma Key

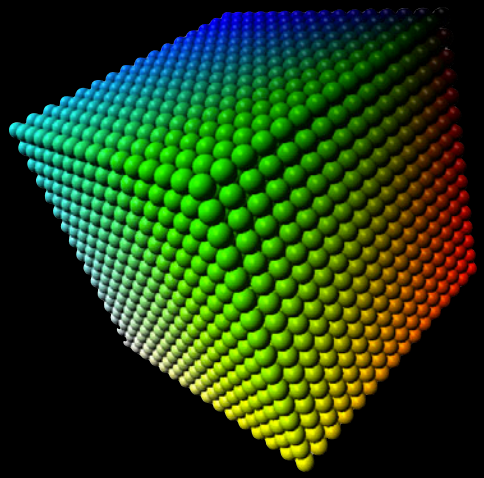
Recipe One: Chroma Key

Overview

- Create a cube of data that maps certain RGB colors to transparent
- Set the cube data and input image as params to CIColorCube
- Use source over compositing to blend over a background

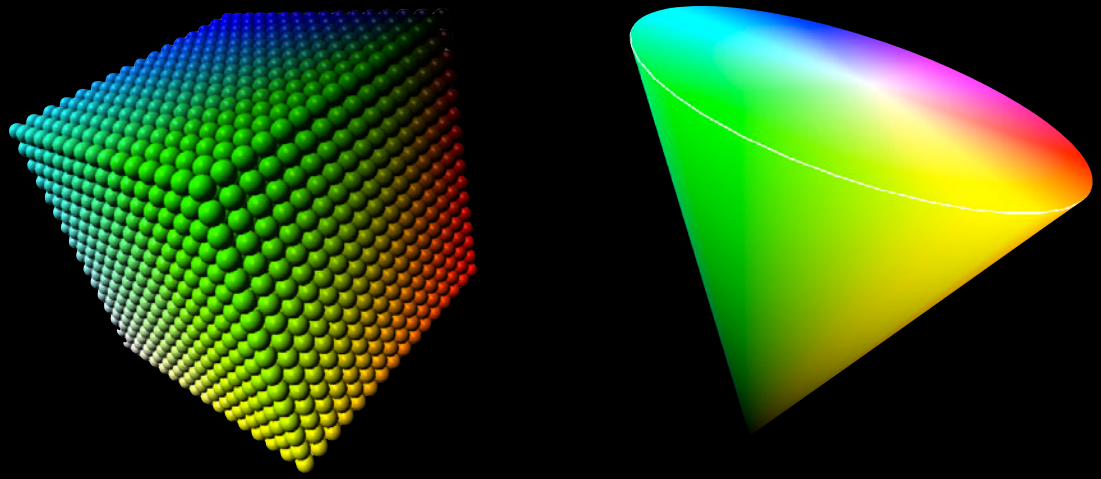
Color Cube

RGB



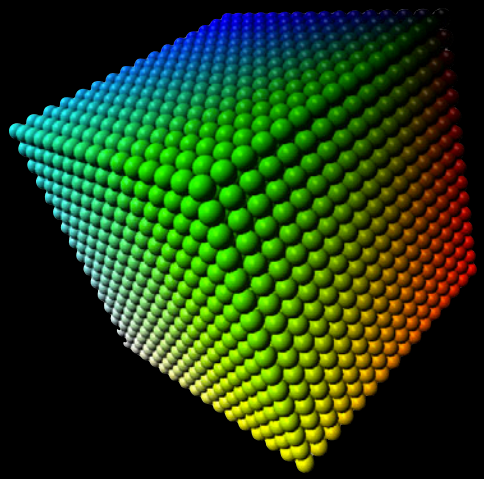
Color Cube

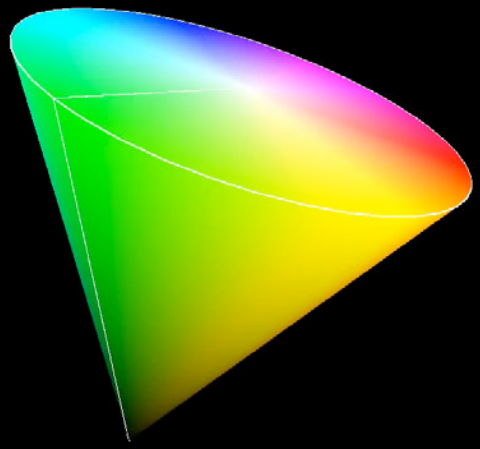
RGB → HSV



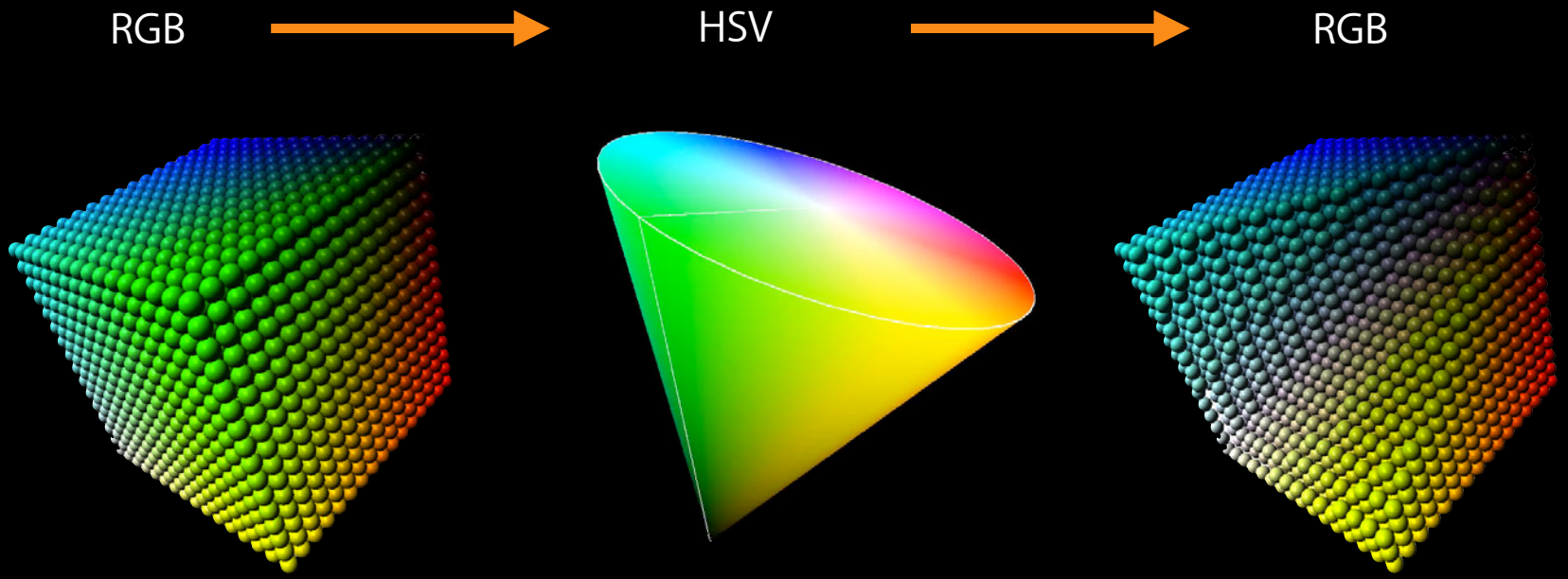
Color Cube

RGB → HSV





Color Cube



Color Cube, in Code

```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

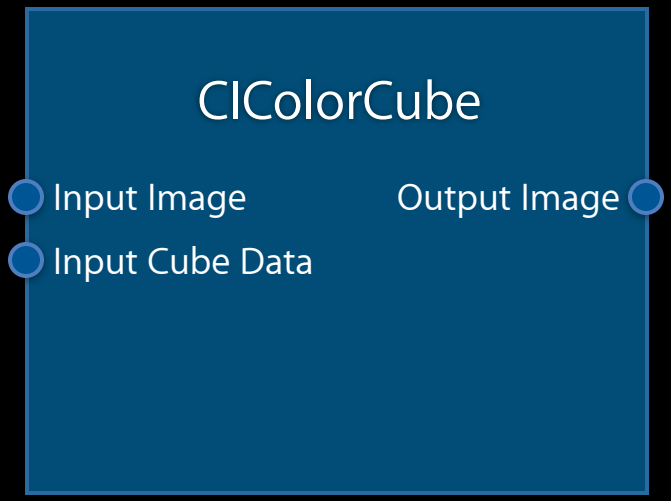
```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

Color Cube, in Code

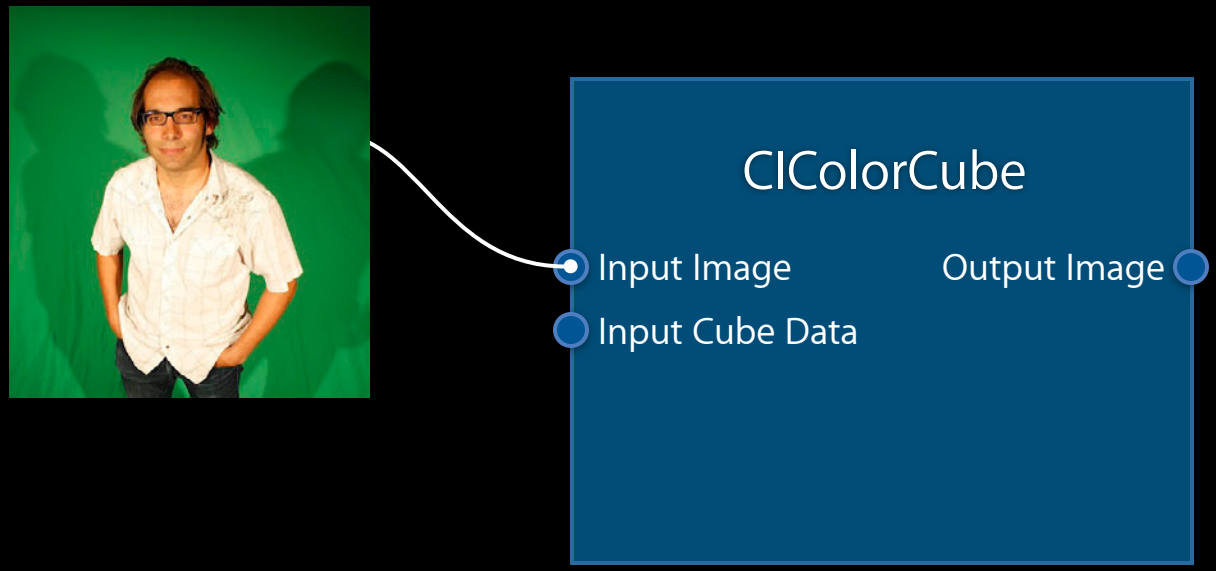
```
const unsigned int size = 64;
float *cubeData = (float *) malloc ( size * size * size * sizeof ( float ) * 4 );
float rgb[3], hsv[3], *c = cubeData;

for ( int z = 0; z < size; z++ ) {
    rgb[2] = ( (double)z ) / ( size - 1 ); // blue value
    for ( int y = 0; y < size; y++ ) {
        rgb[1] = ( (double)y ) / ( size - 1 ); // green value
        for ( int x = 0; x < size; x++ ) {
            rgb[0] = ( (double)x ) / ( size - 1 ); // red value
            rgbToHSV ( rgb, hsv );
            float alpha = ( hsv[0] > minHueAngle && hsv[0] < maxHueAngle ) ? 0.0f : 1.0f;
            c[0] = rgb[0] * alpha; c[1] = rgb[1] * alpha; c[2] = rgb[2] * alpha; c[3] = alpha;
        }
    }
}
```

```
NSData *data = [NSData dataWithBytesNoCopy:cubeData length:cubeDataSize freeWhenDone:YES];
CIColorCube *colorCube = [CIFilter filterWithName:@"CIColorCube"];
[colorCube setValue:[NSNumber numberWithInt:size] forKey:@"inputCubeDimension"];
[colorCube setValue:data forKey:@"inputCubeData"];
```

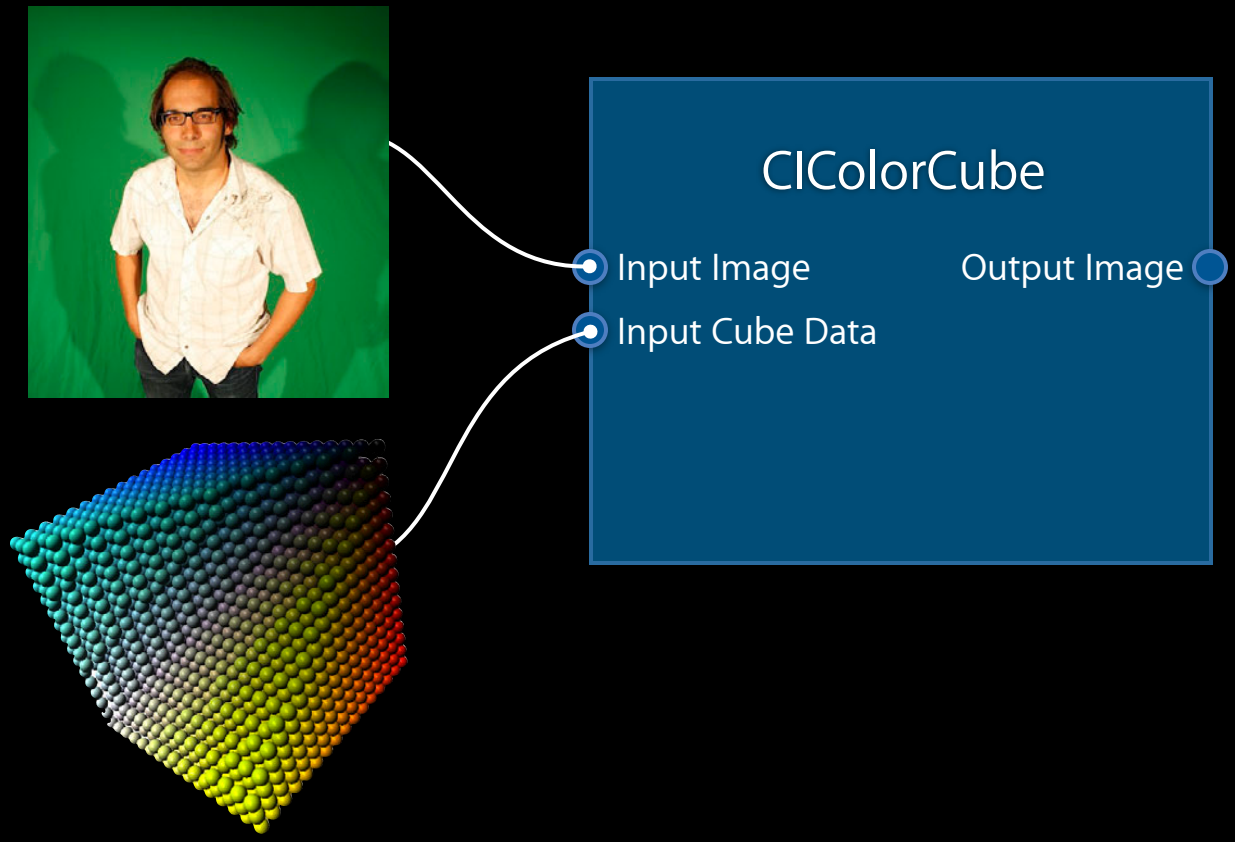
Bye Bye Background



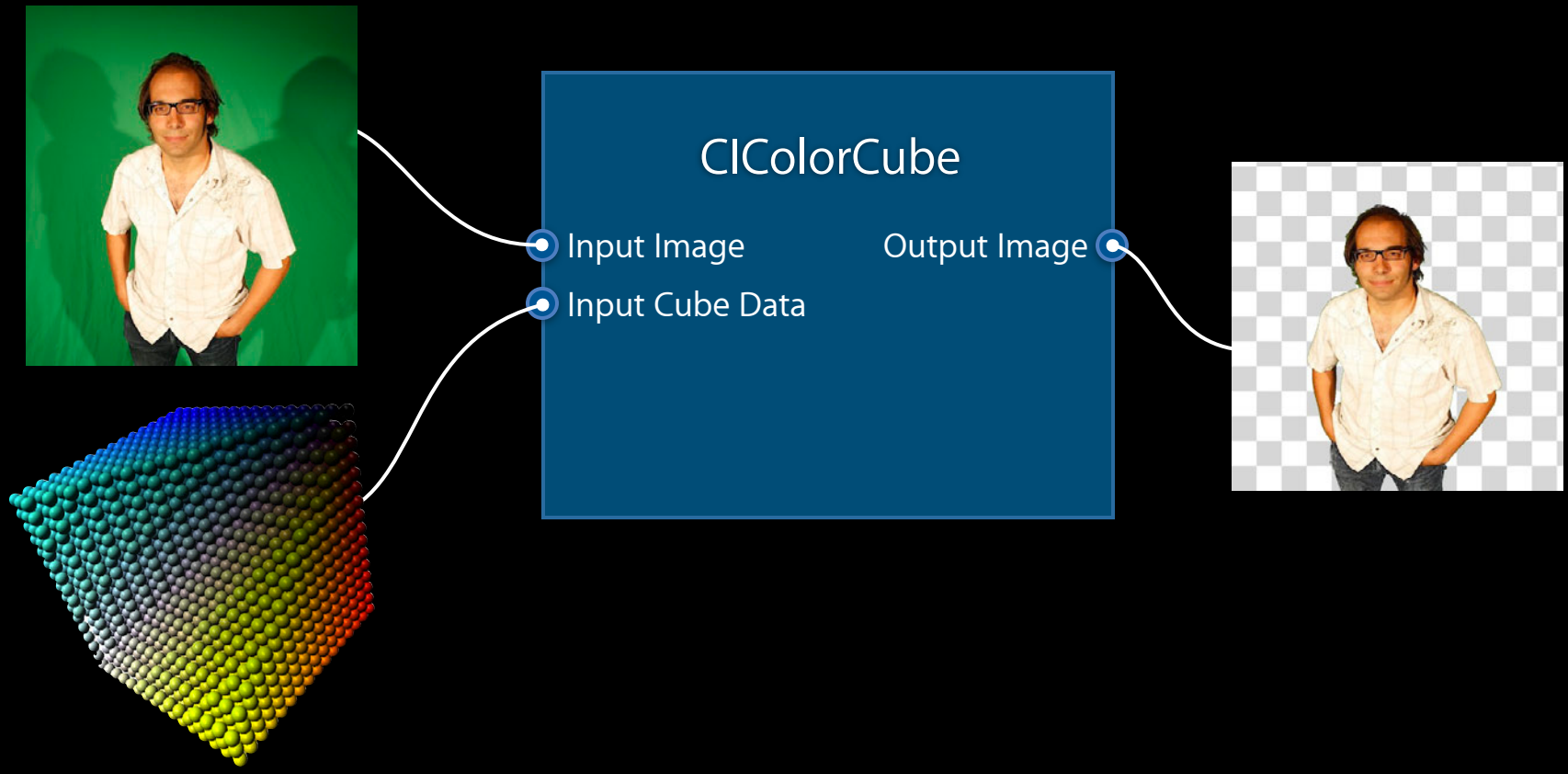
Bye Bye Background



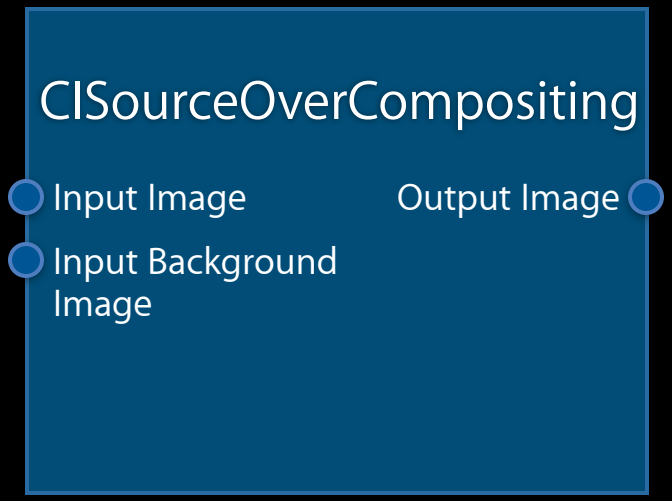
Bye Bye Background



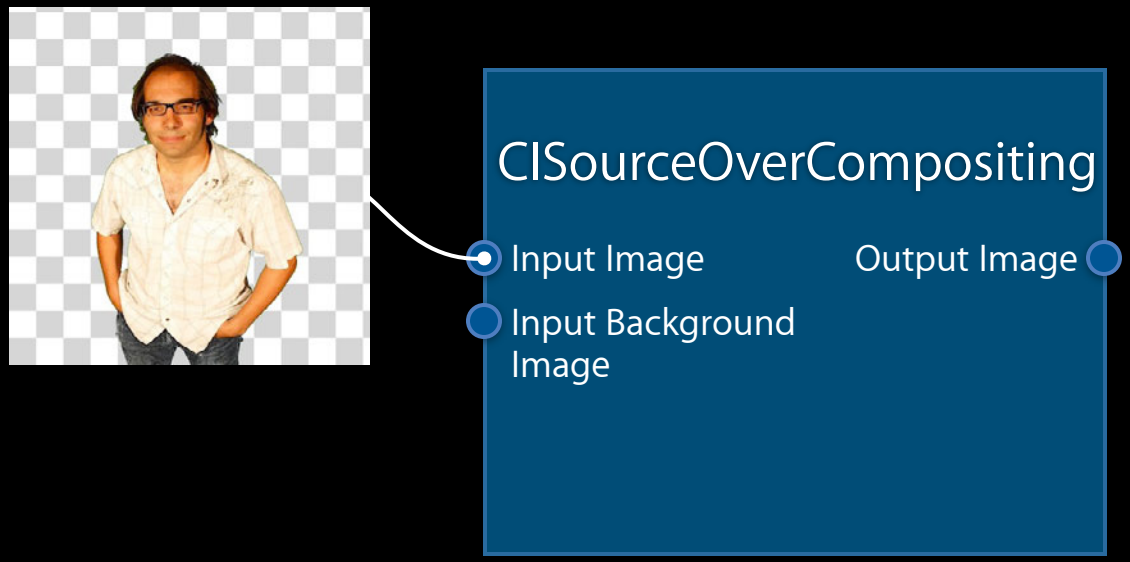
Bye Bye Background



Make it Look Like I've Been Places



Make it Look Like I've Been Places



Make it Look Like I've Been Places

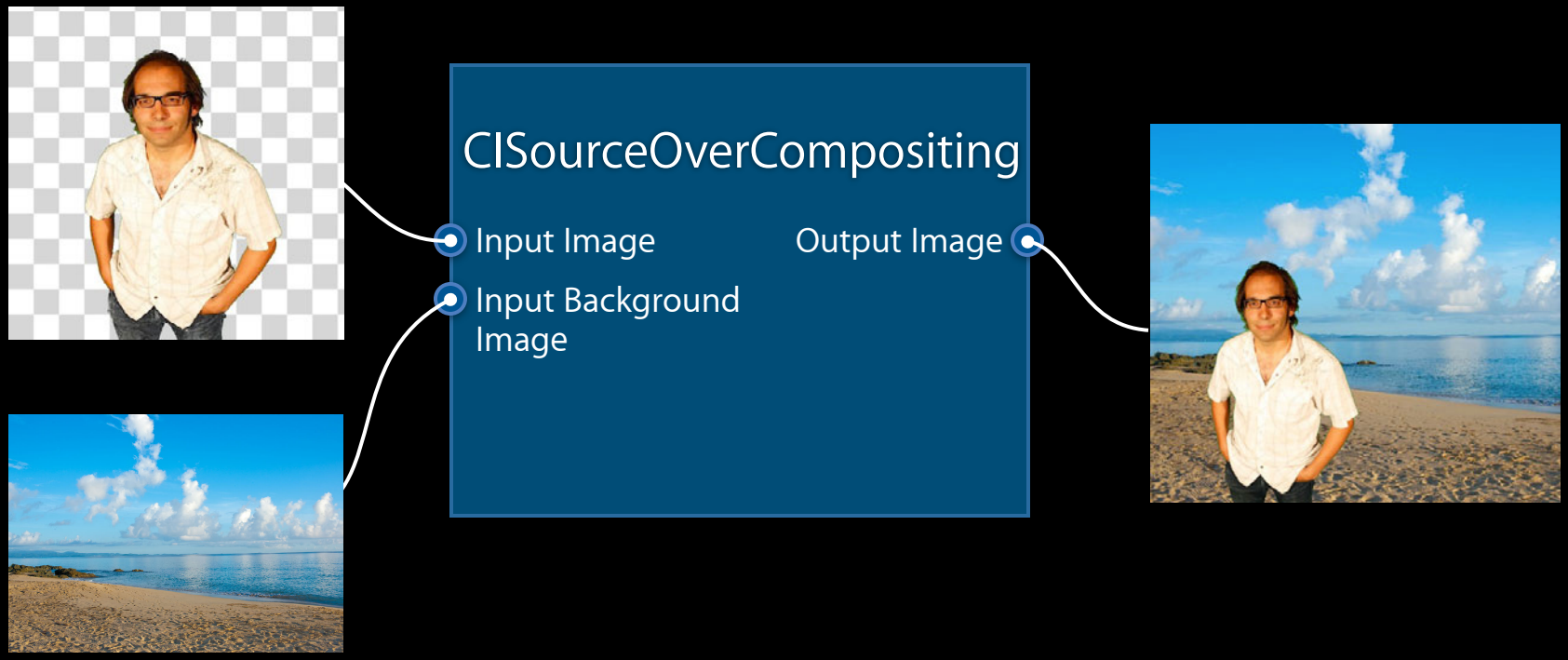
CISourceOverCompositing

Input Image

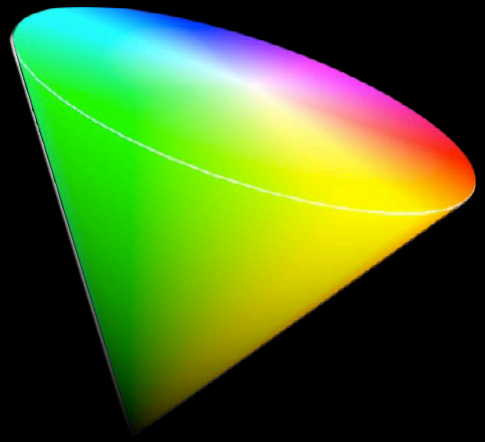
Input Background Image

Output Image

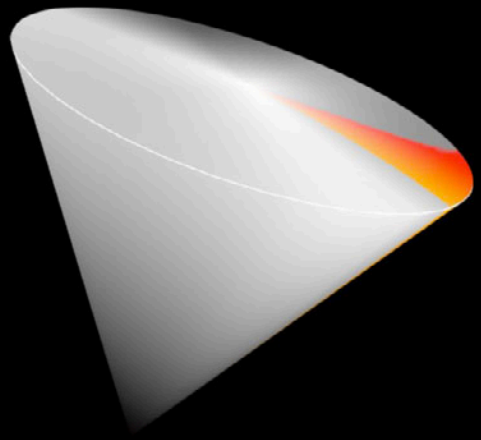
Make it Look Like I've Been Places



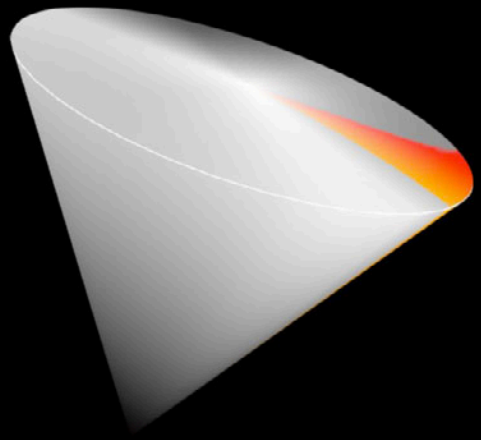
Another Application: Color Accent Mode



Another Application: Color Accent Mode

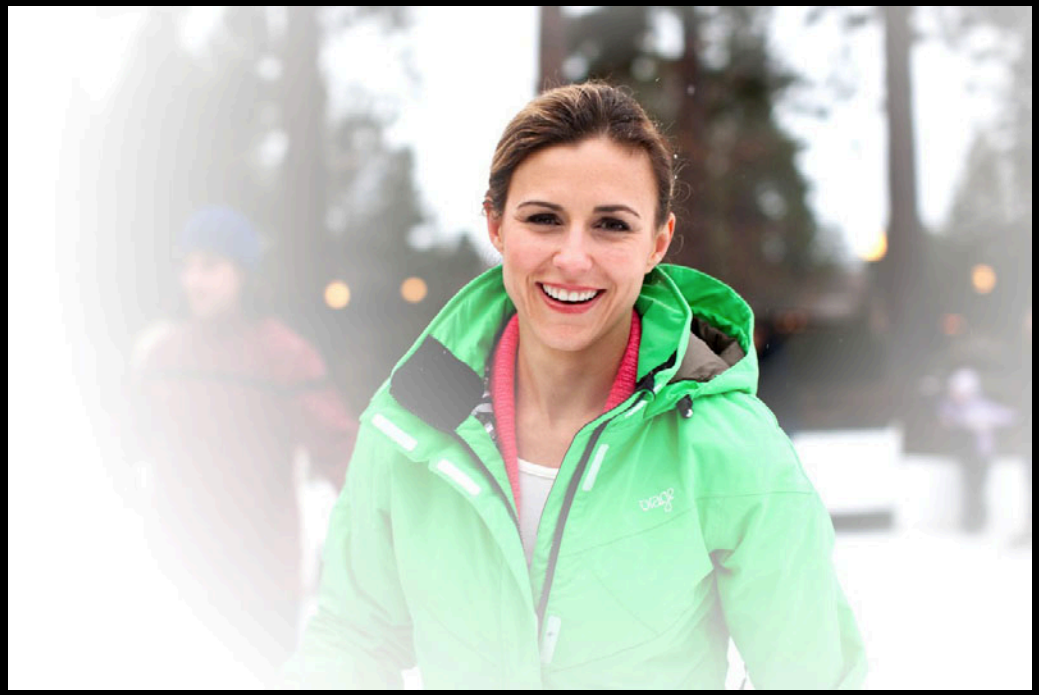


Another Application: Color Accent Mode



Recipe Two: White Vignette

Recipe Two: White Vignette



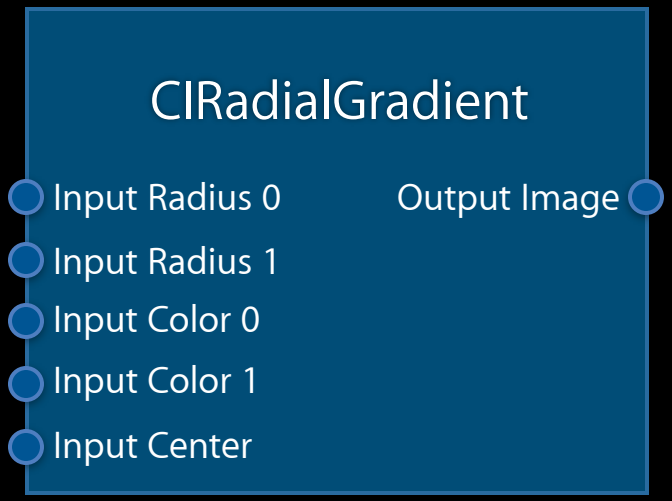
Overview

- Find face(s)
- Create a base shade map using CIRadialGradient centered on face
- Blend gradient image with base image

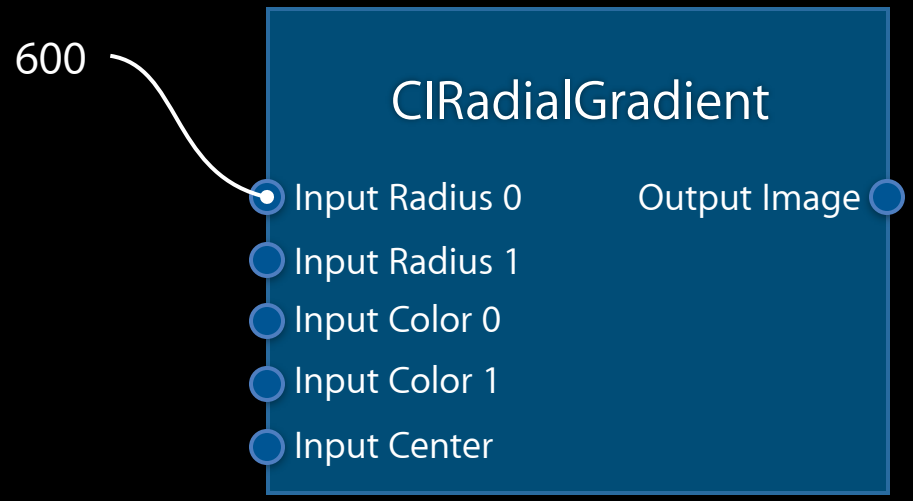
Find Face(s)

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];

CIFeature *face = (CIFeature *)[faceArray objectAtIndex:0];

CGFloat xCenter = face.bounds.origin.x + face.bounds.size.width/2.0;
CGFloat yCenter = face.bounds.origin.y + face.bounds.size.height/2.0;

CIVector *center = [CIVector vectorWithX:xCenter Y:yCenter];
```

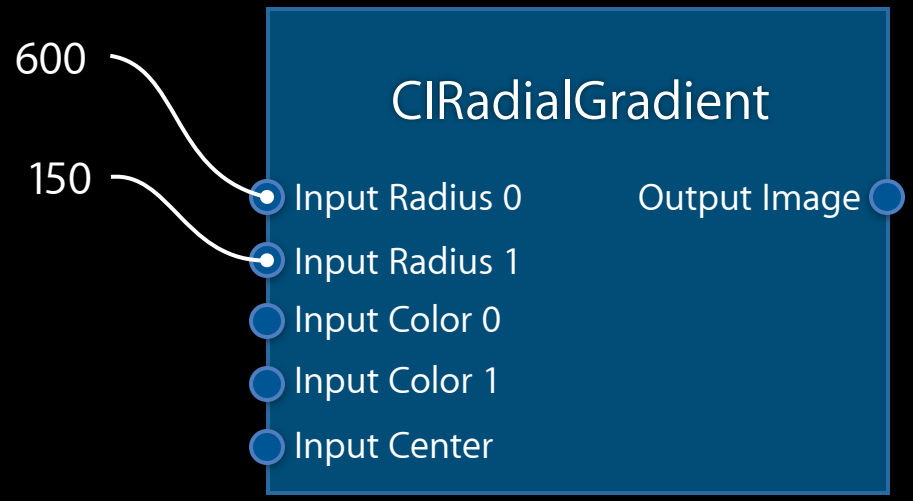
CIRadialGradient: Image Generator



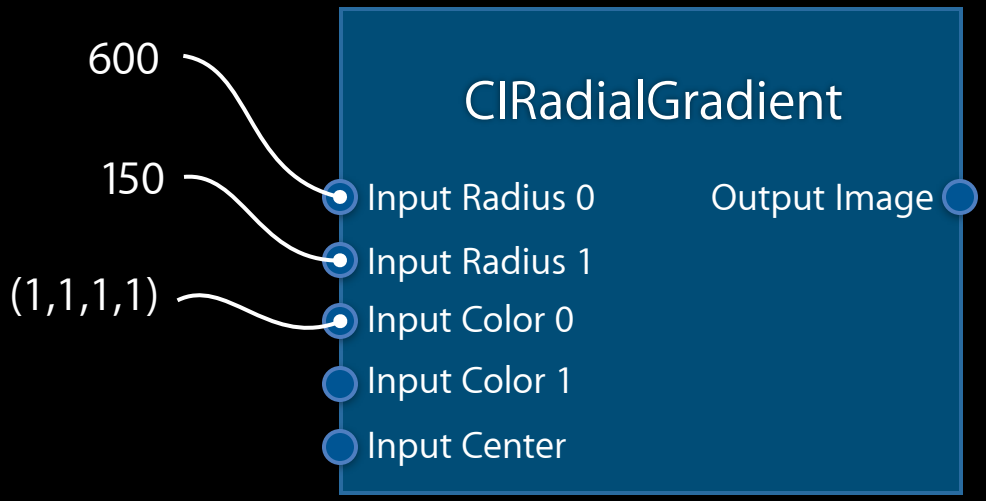
CIRadialGradient: Image Generator



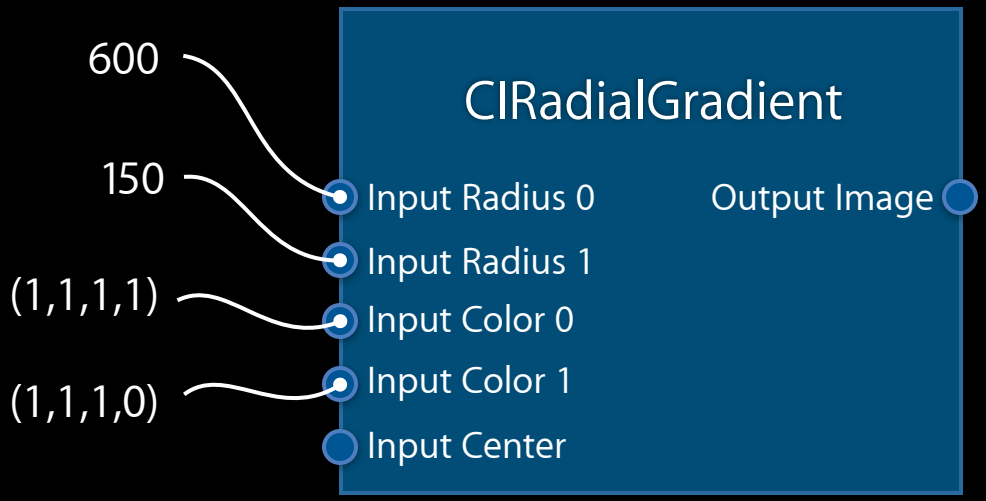
CIRadialGradient: Image Generator



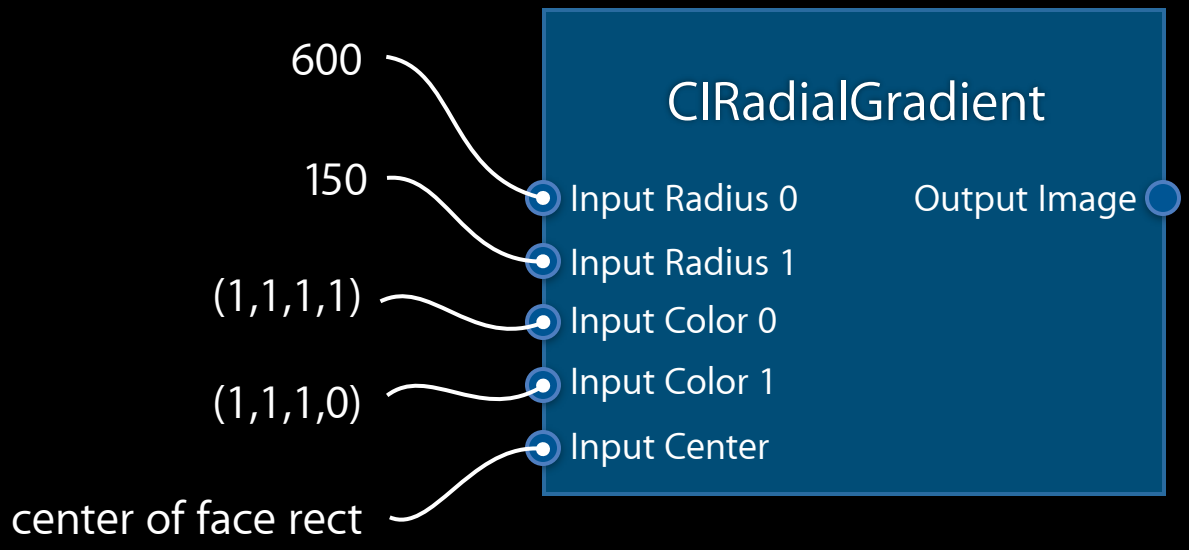
CIRadialGradient: Image Generator



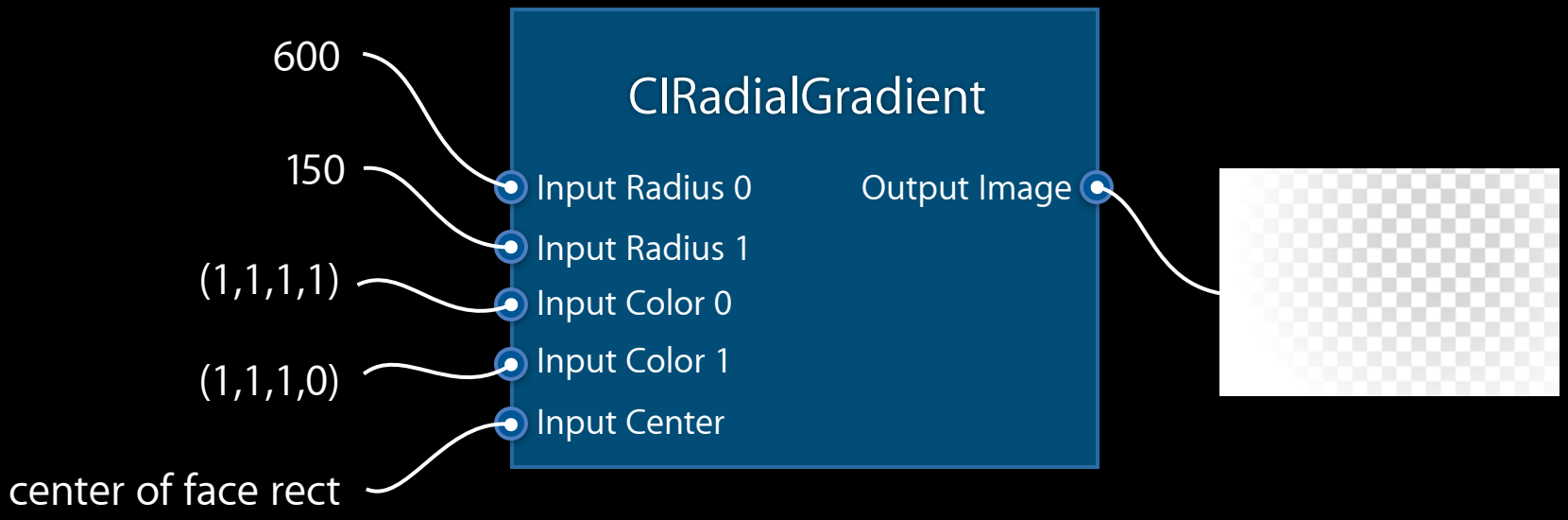
CIRadialGradient: Image Generator



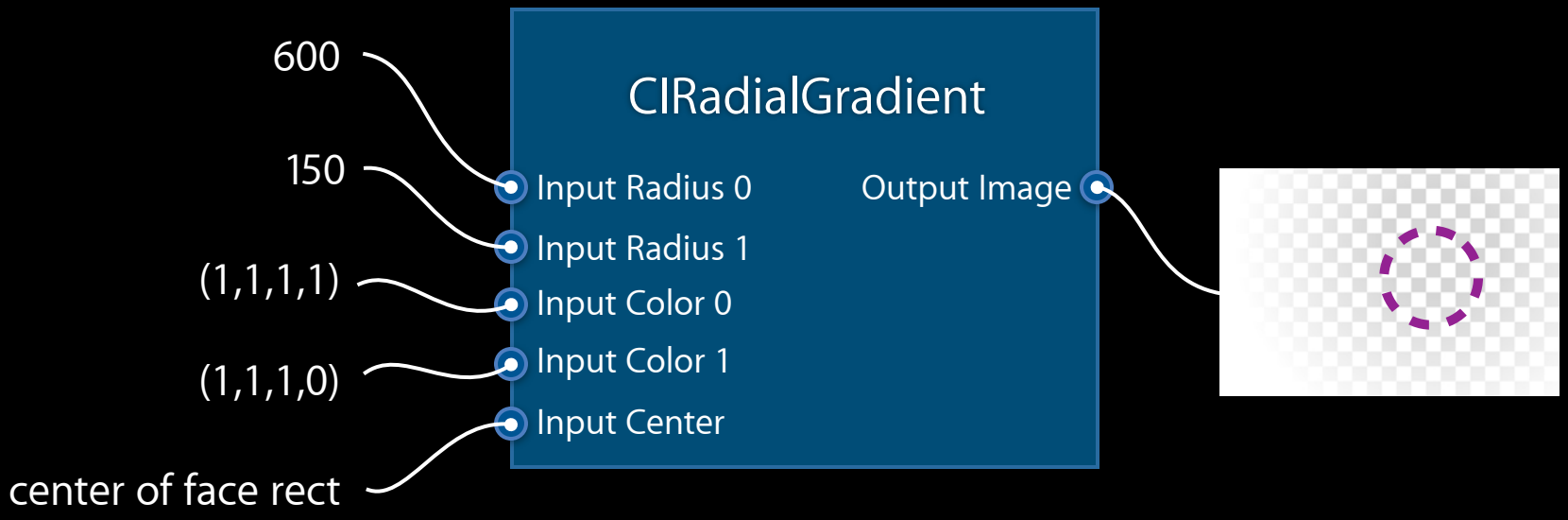
CIRadialGradient: Image Generator



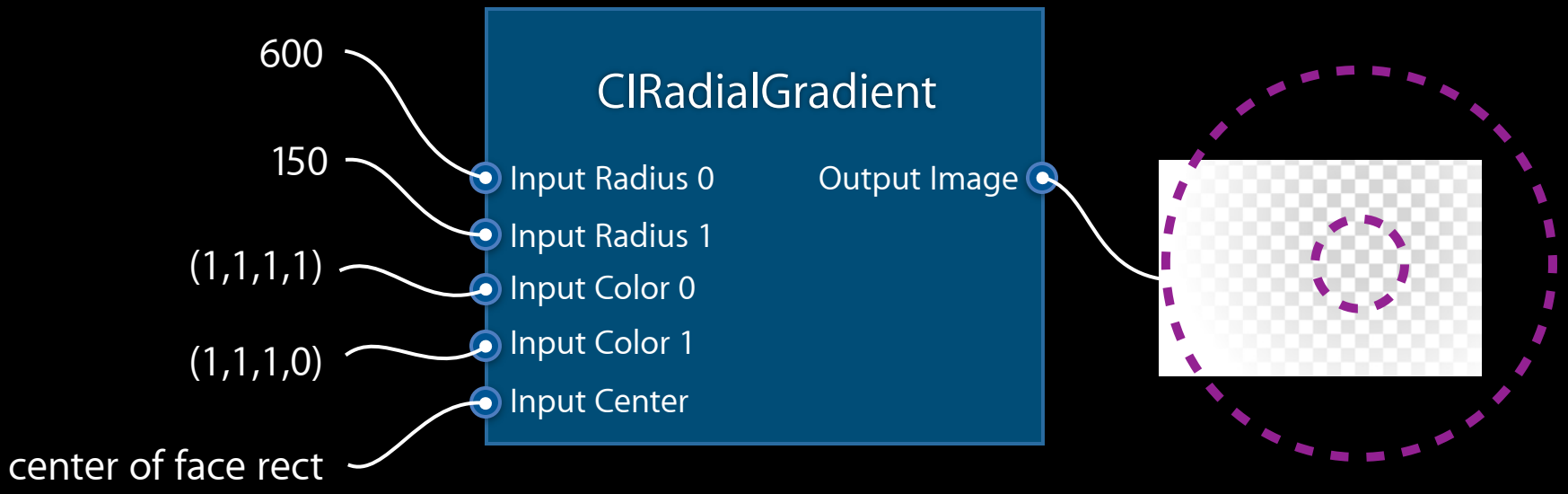
CIRadialGradient: Image Generator



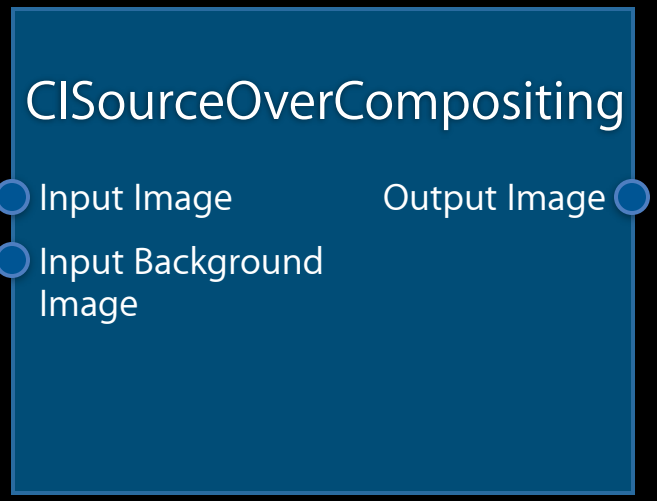
CIRadialGradient: Image Generator



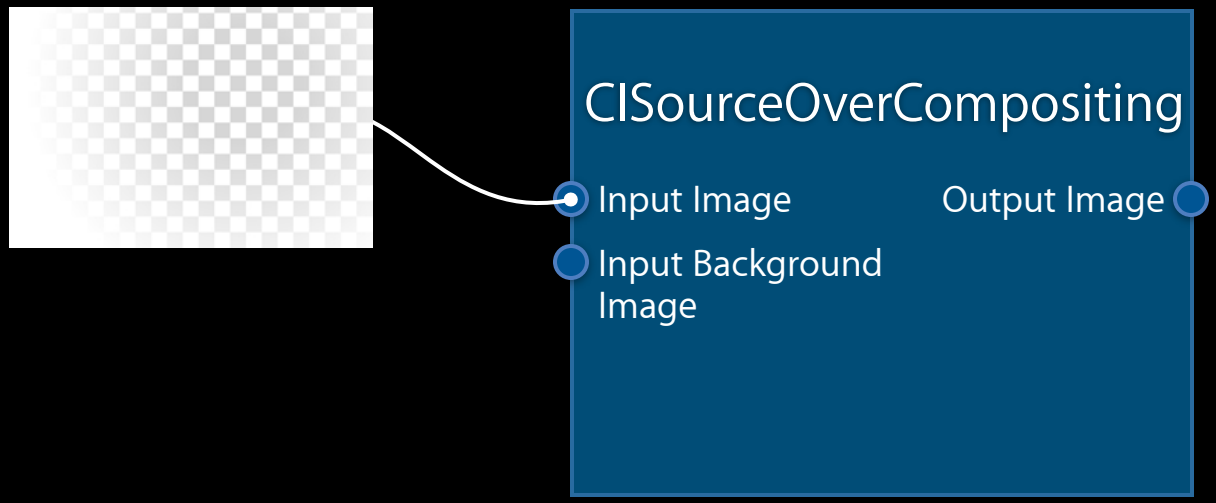
CIRadialGradient: Image Generator



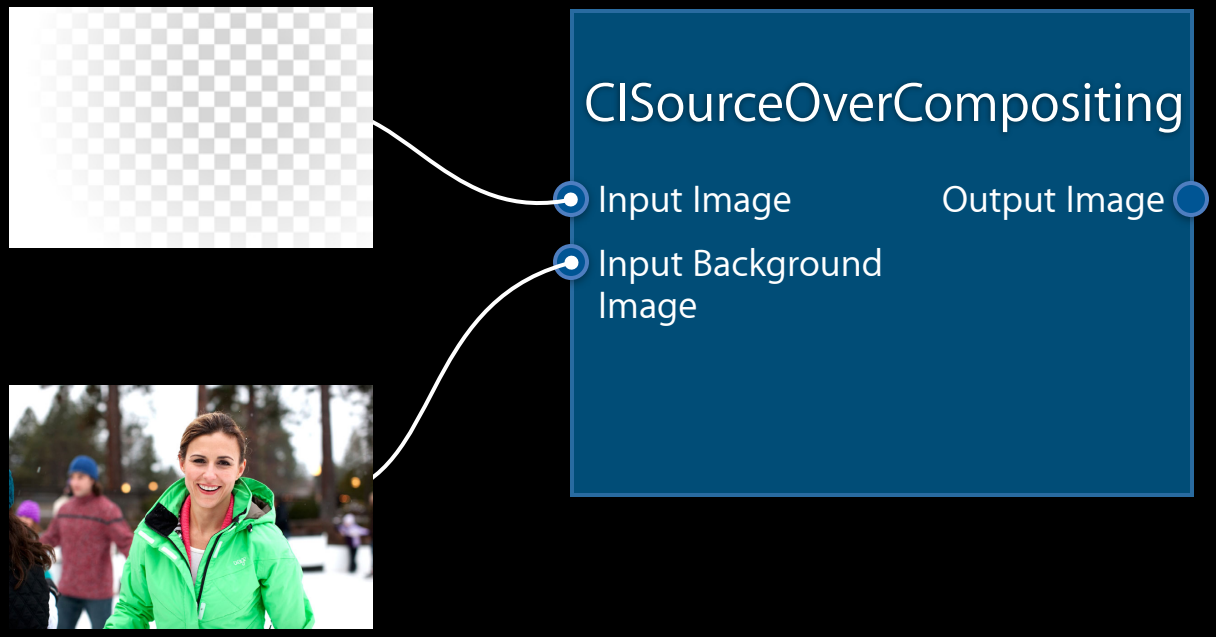
Final Compositing



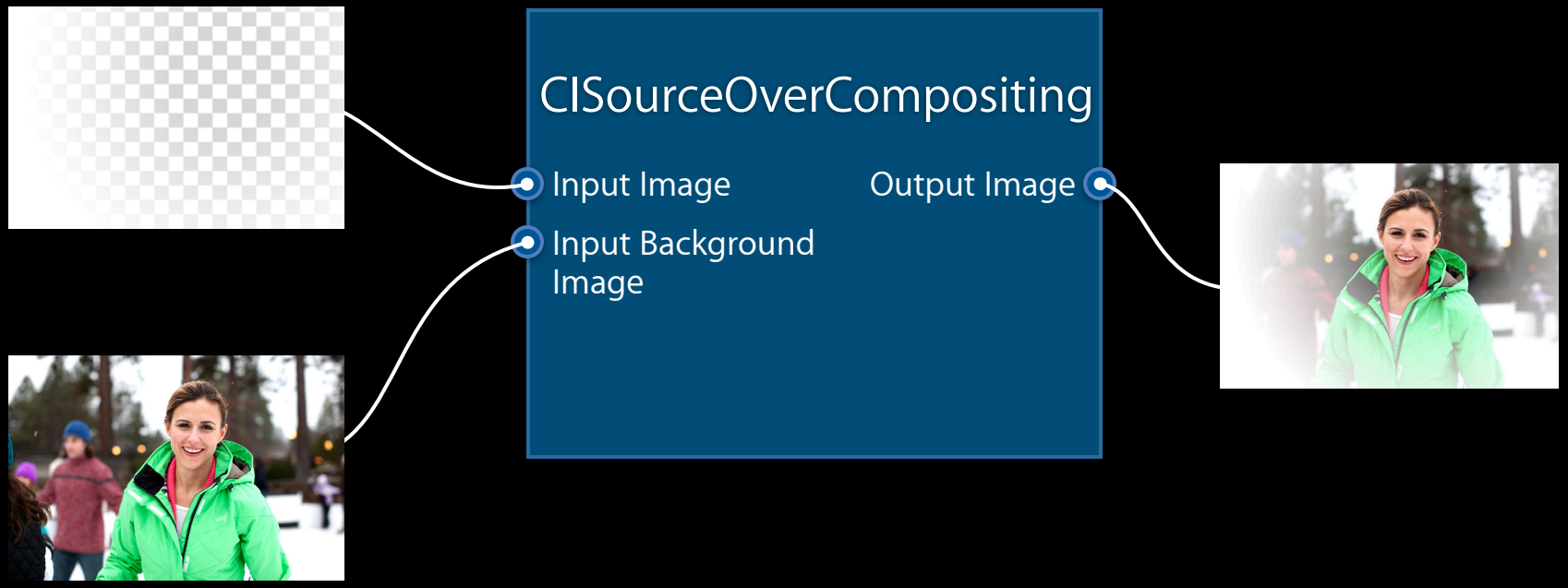
Final Compositing



Final Compositing

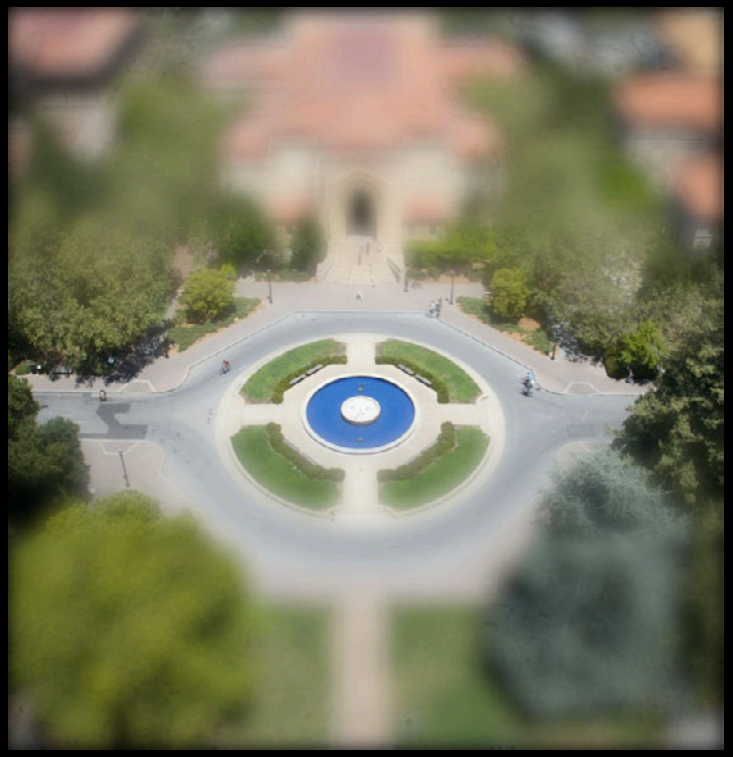


Final Compositing



Recipe Three: Tilt Shift Look

Recipe Three: Tilt Shift Look



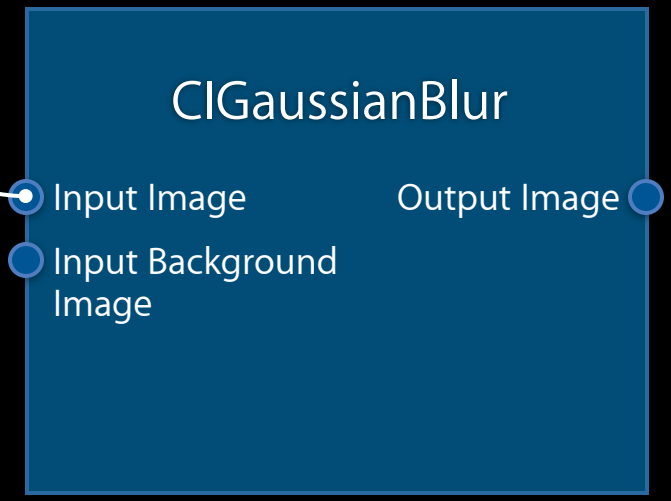
Overview

- Create a blurred version of the image
- Create two linear gradients and blend together with addition compositing
- Composite blurred image with mask using linear gradients

Blurrrr


CIGaussianBlur
Input Image
Input Background Image
Output Image

Blurrrr


CIGaussianBlur
Input Image
Input Background Image
Output Image

Blurrrr


CIGaussianBlur
Input Image
Output Image
Input Background Image
10

Blurrrr

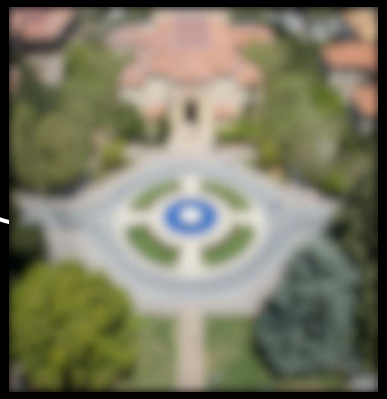
CIGaussianBlur

Input Image

Input Background Image

Output Image

10



Blurrrr

CIGaussianBlur

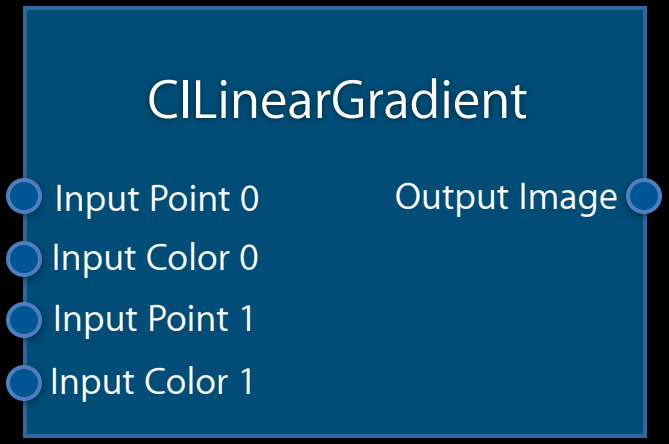
Input Image

Input Background Image

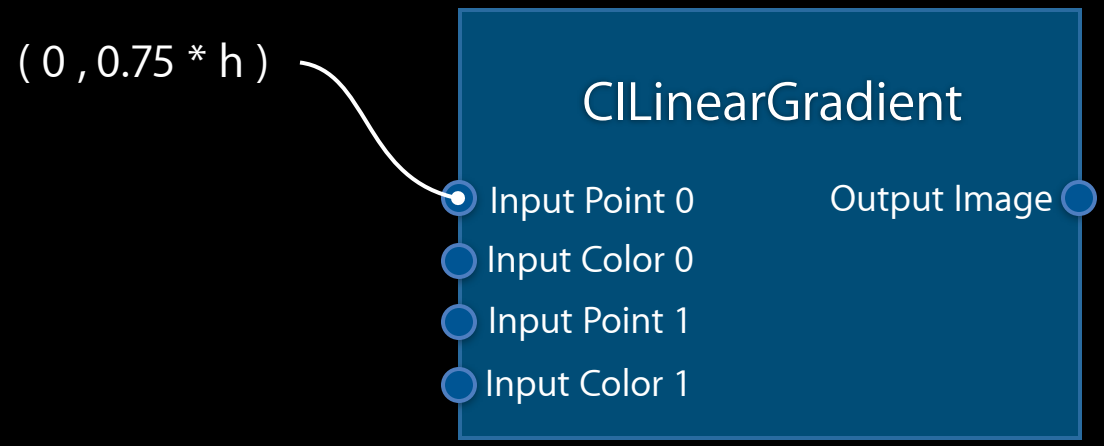
Output Image

10

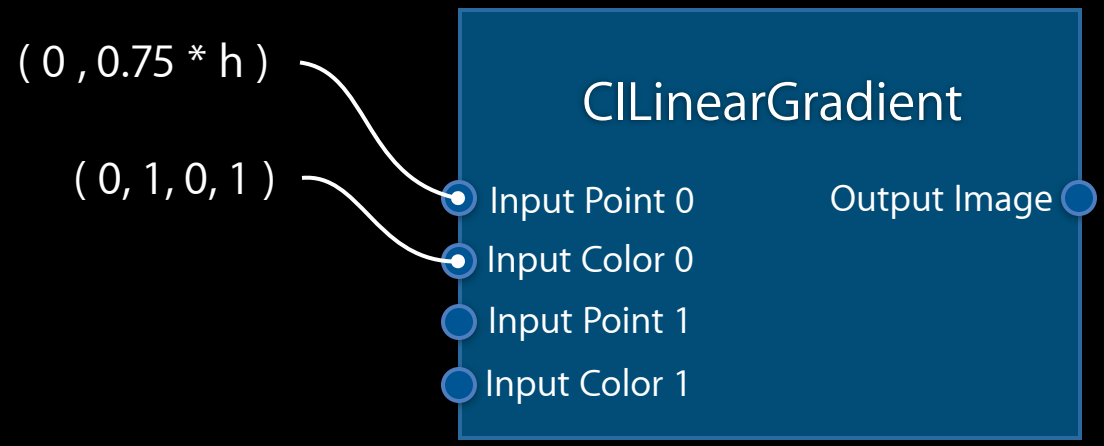
Create (Green) Linear Gradients



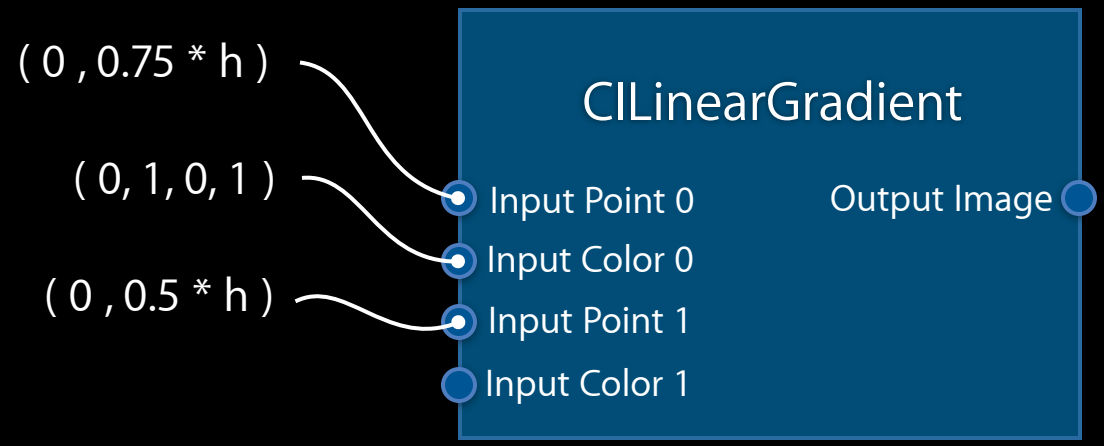
Create (Green) Linear Gradients



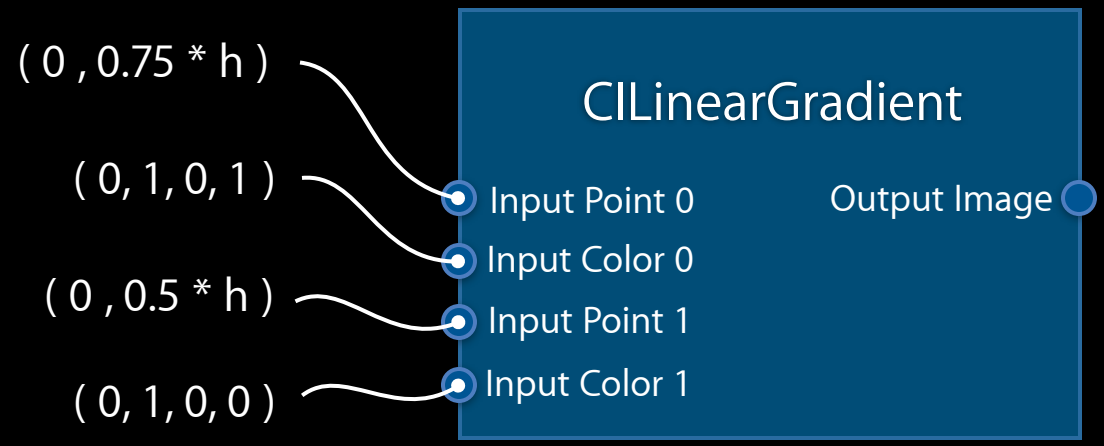
Create (Green) Linear Gradients



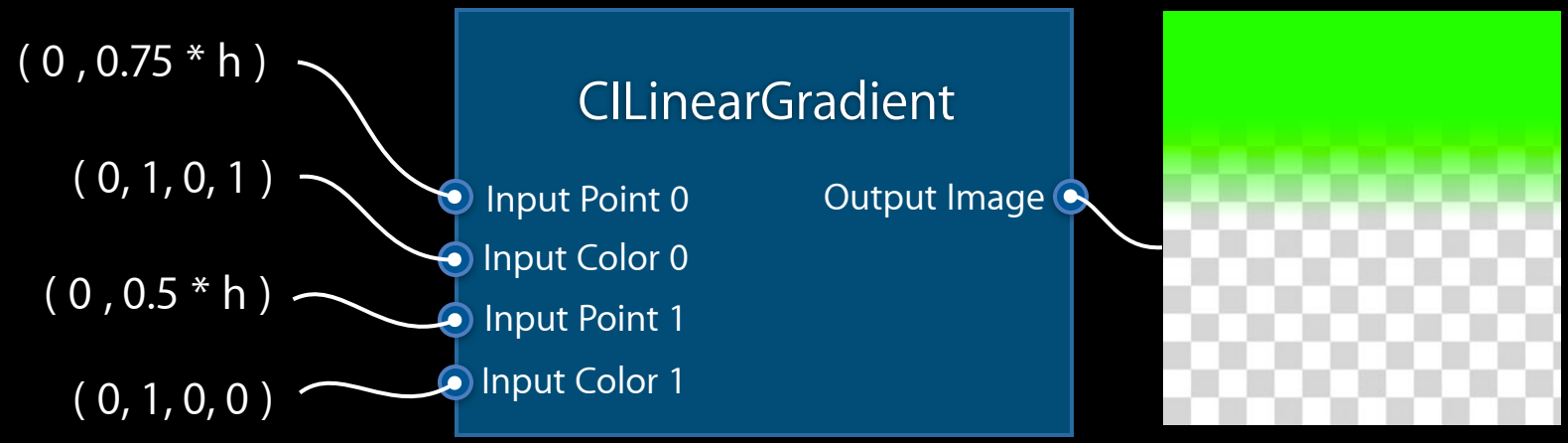
Create (Green) Linear Gradients



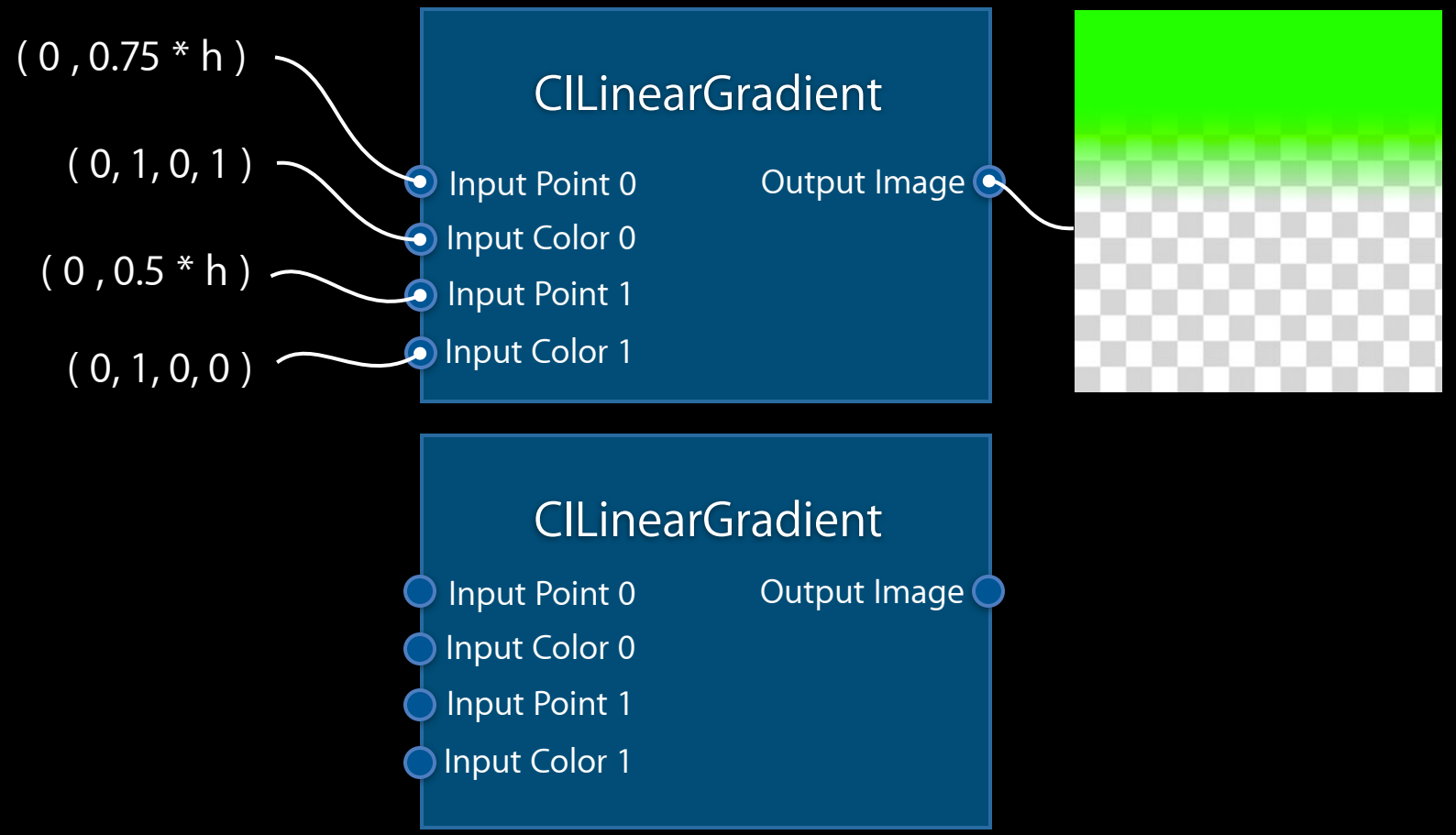
Create (Green) Linear Gradients



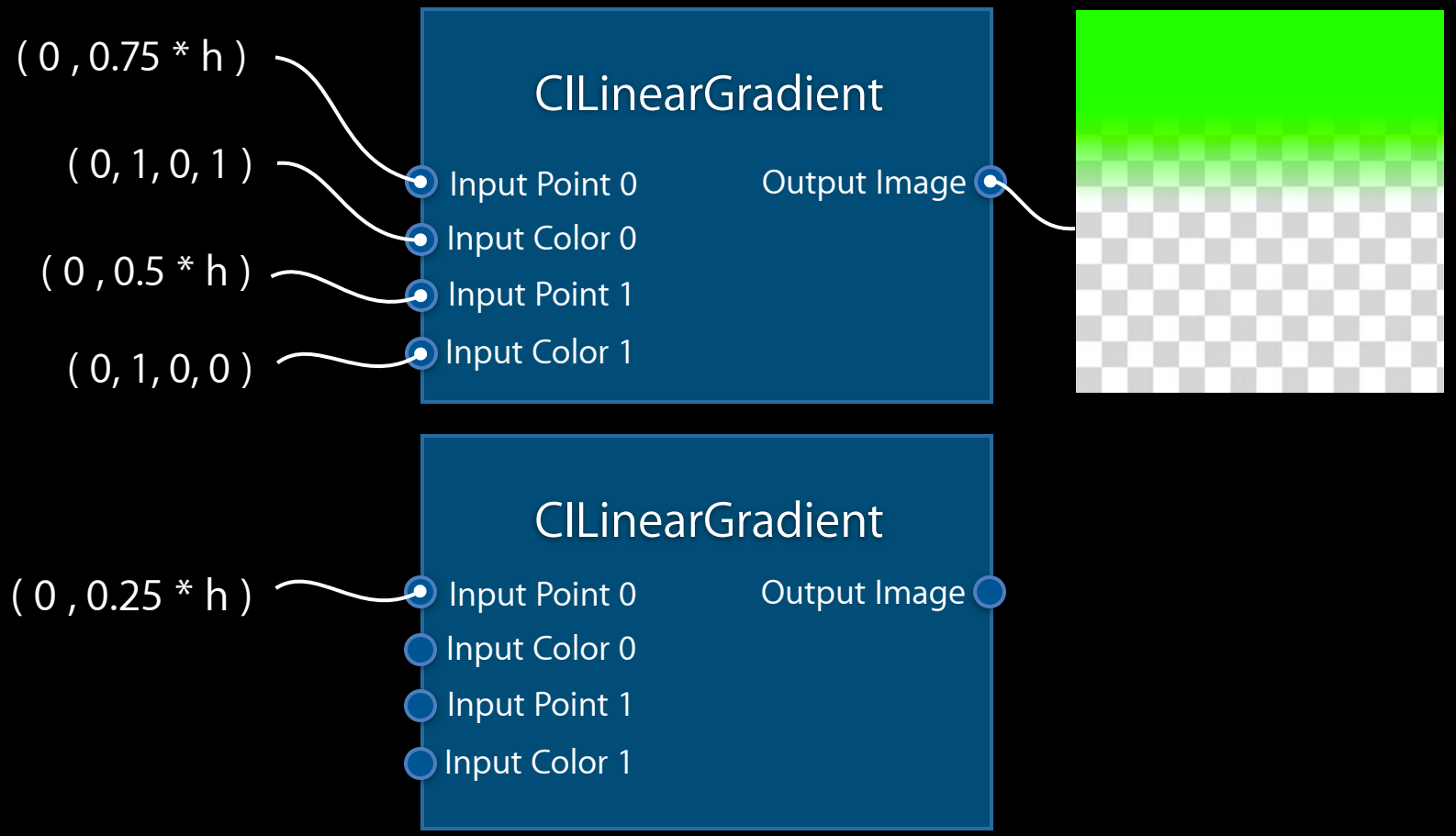
Create (Green) Linear Gradients



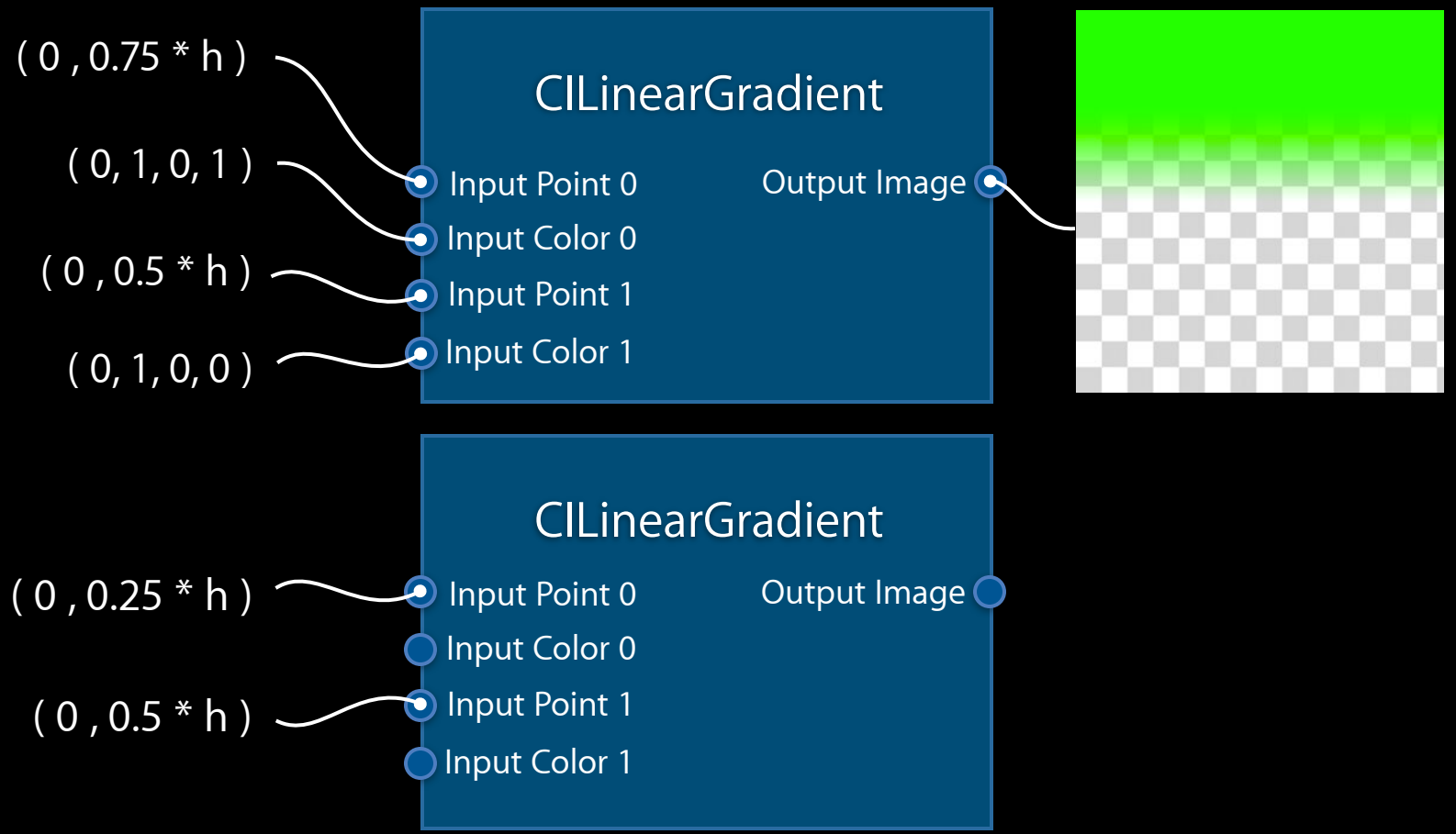
Create (Green) Linear Gradients



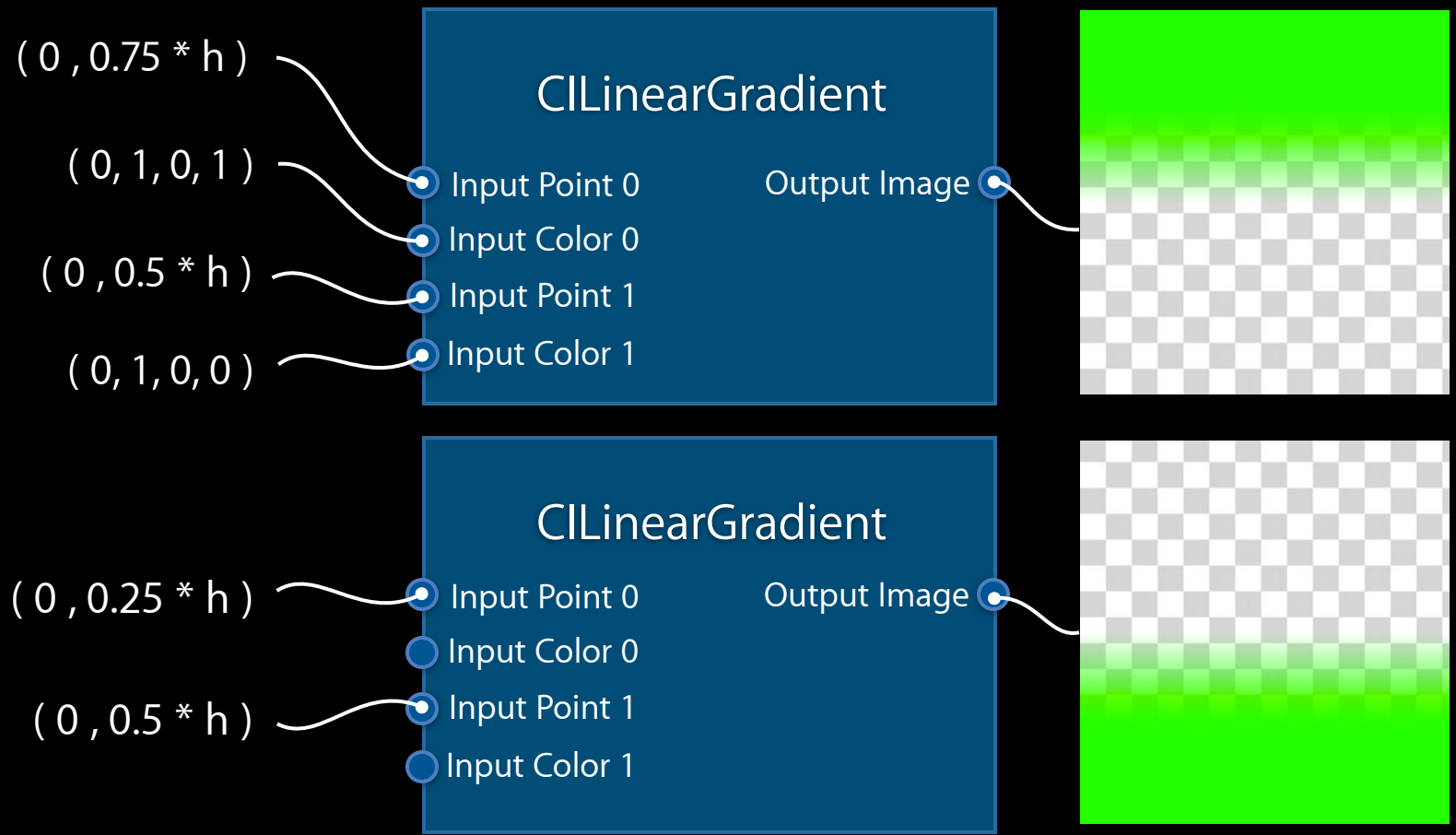
Create (Green) Linear Gradients



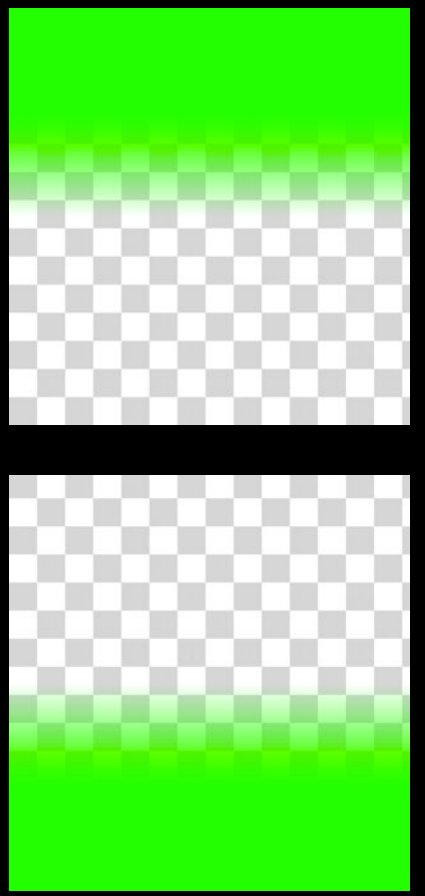
Create (Green) Linear Gradients



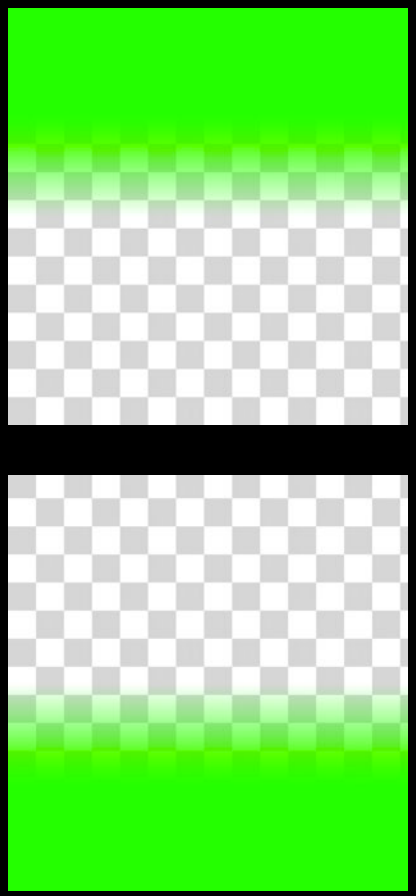
Create (Green) Linear Gradients



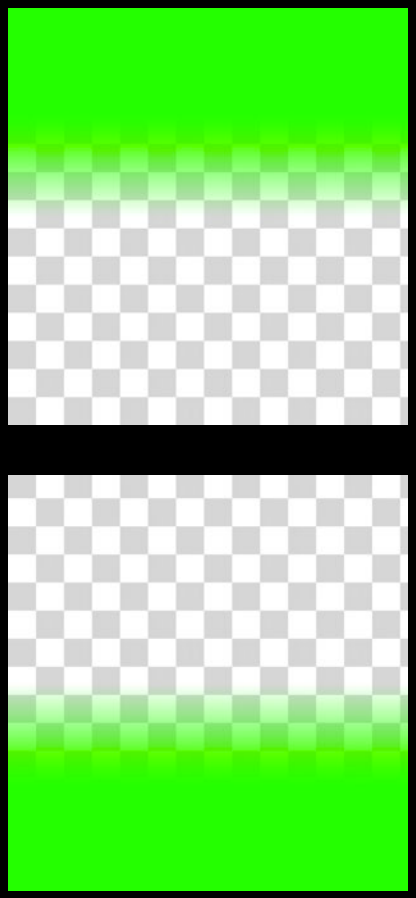
Create (Green) Linear Gradients

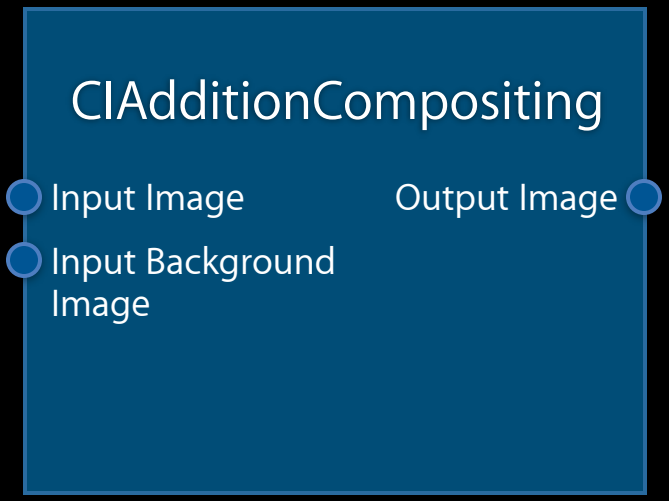


Create (Green) Linear Gradients

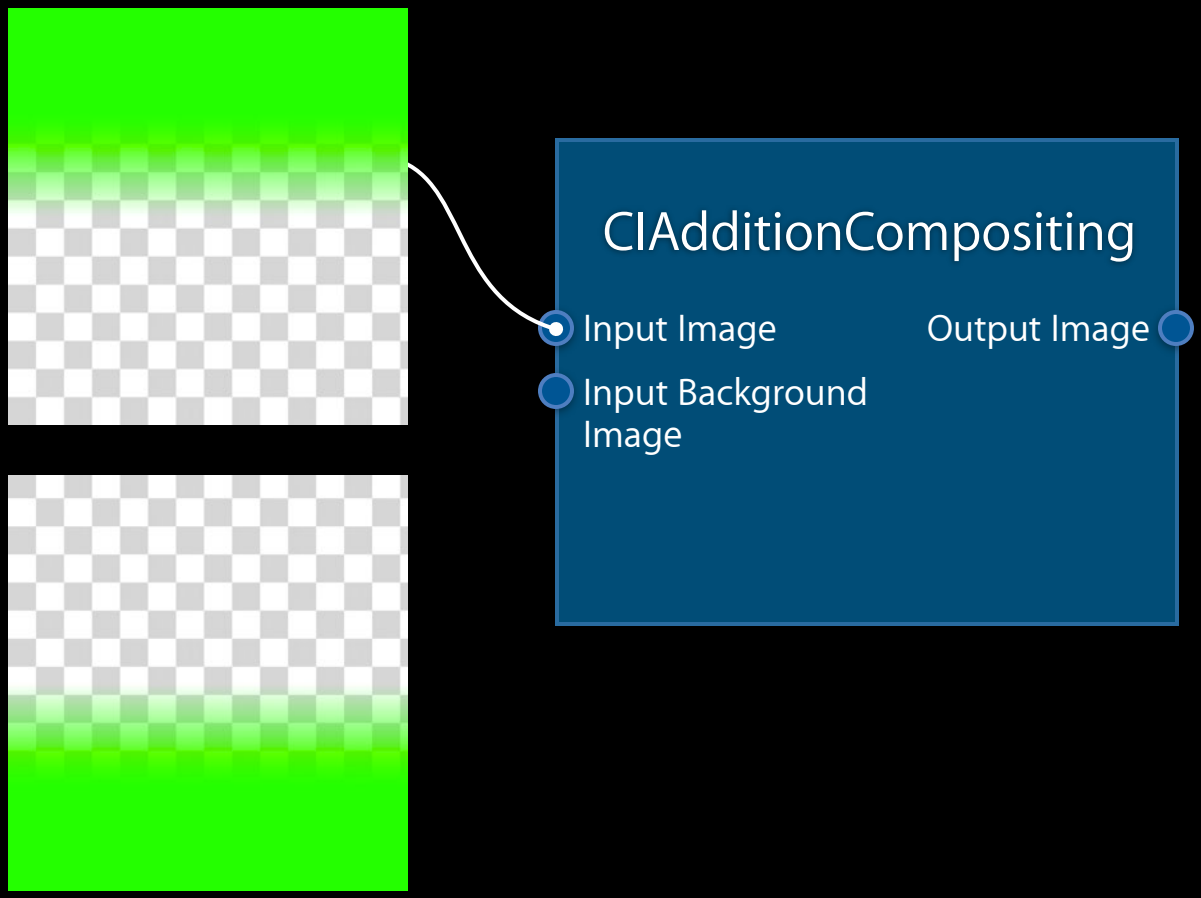


Create (Green) Linear Gradients

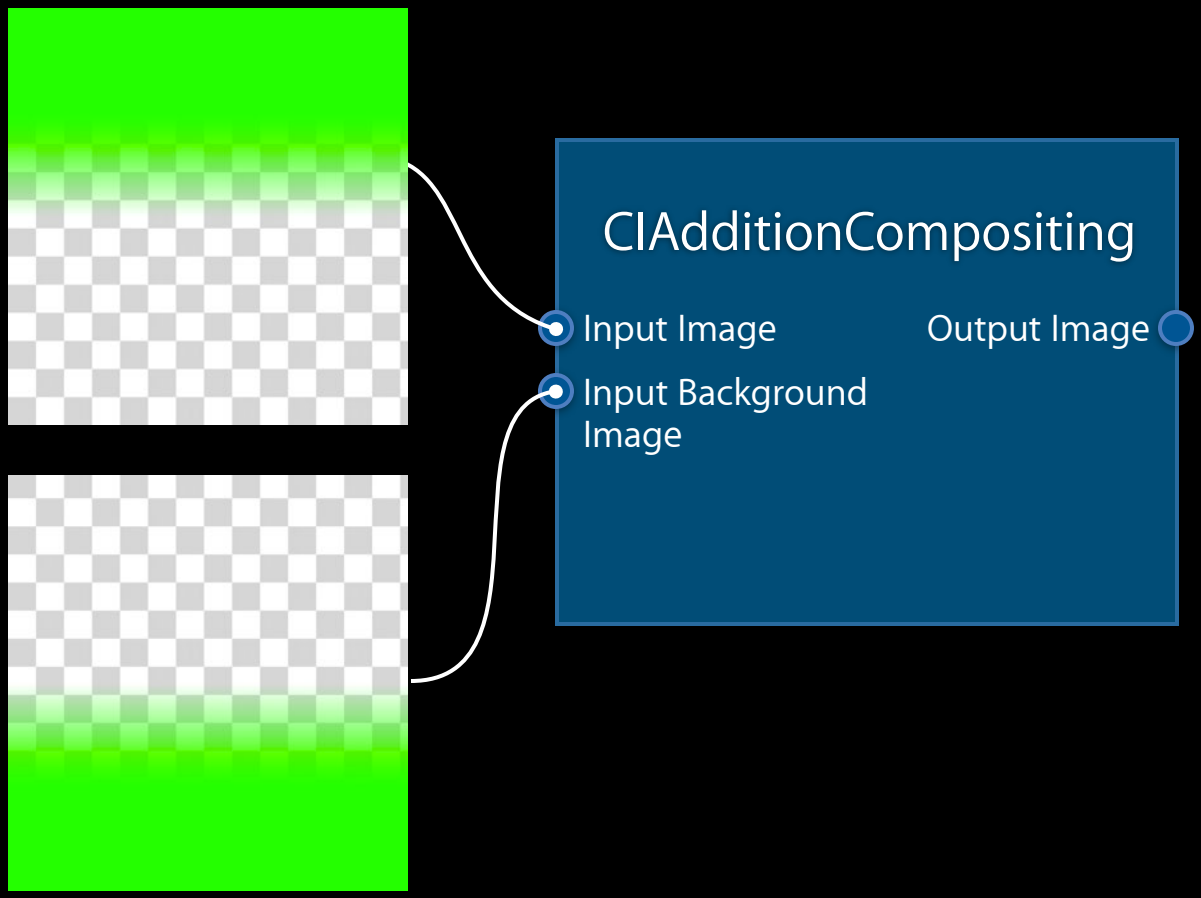




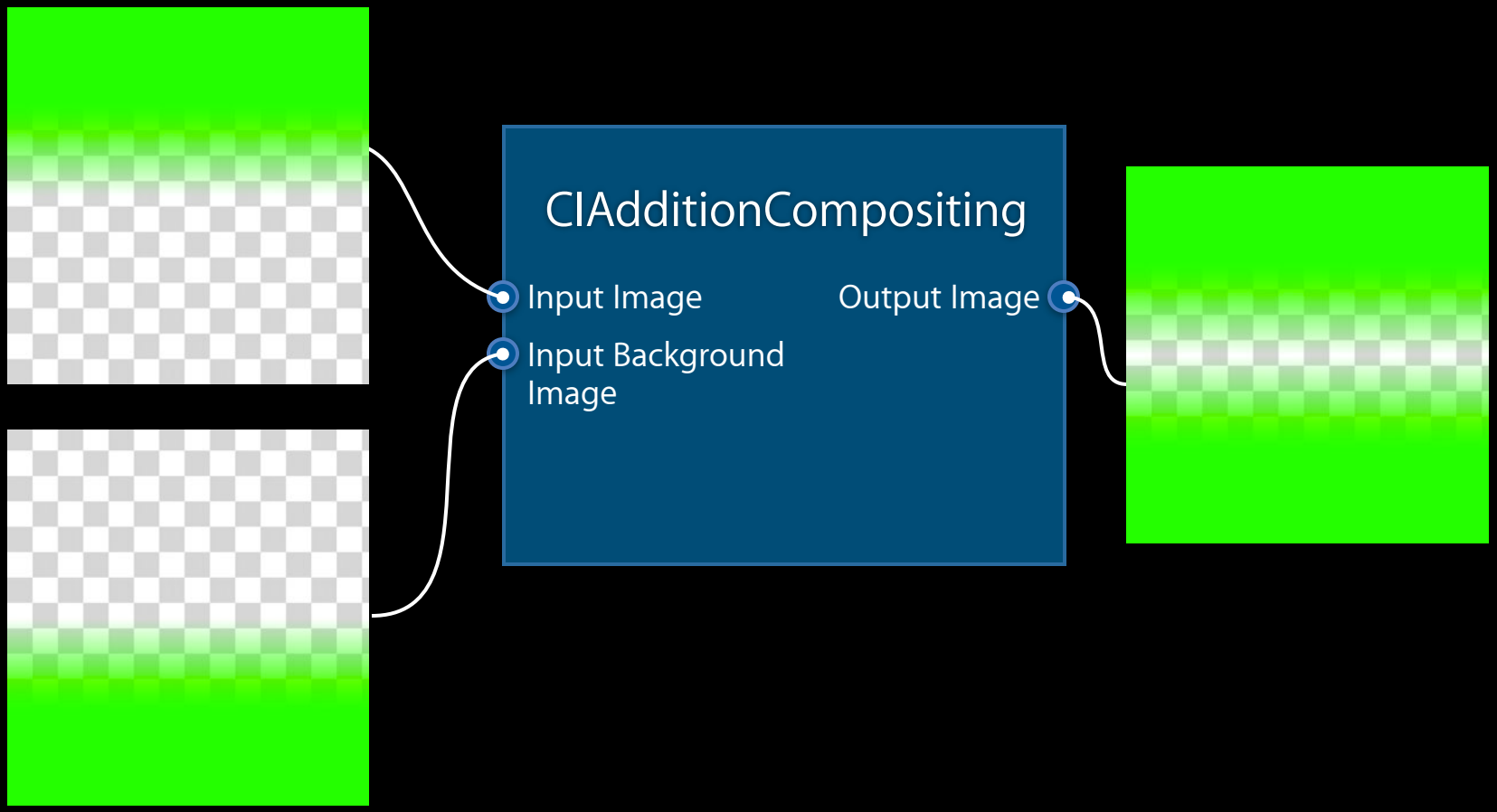
Create (Green) Linear Gradients



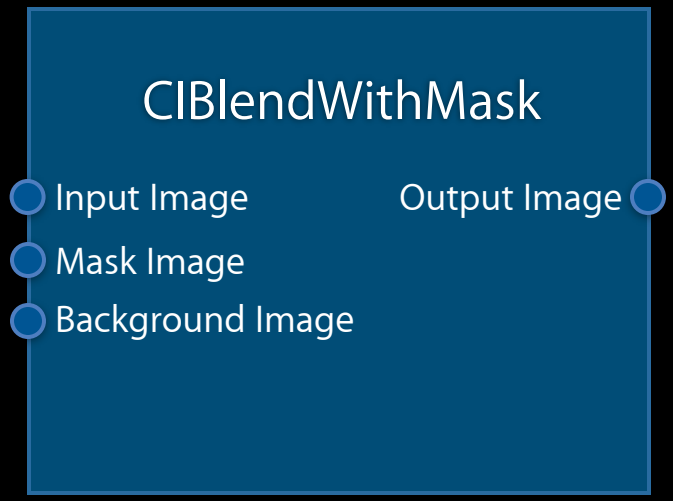
Create (Green) Linear Gradients



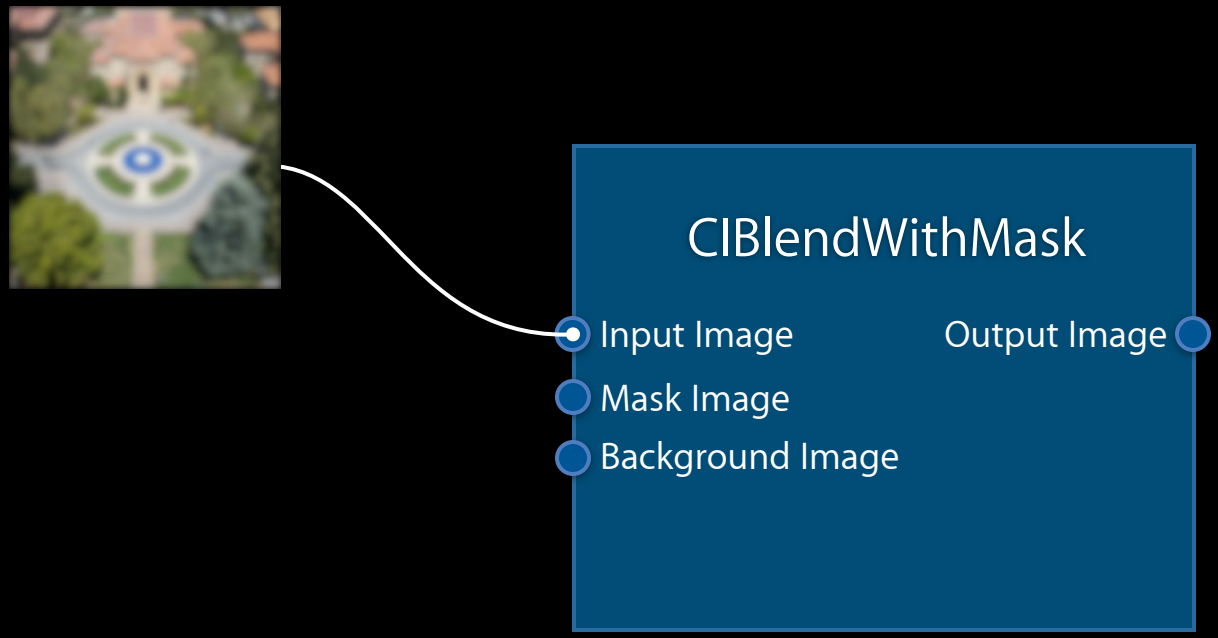
Create (Green) Linear Gradients



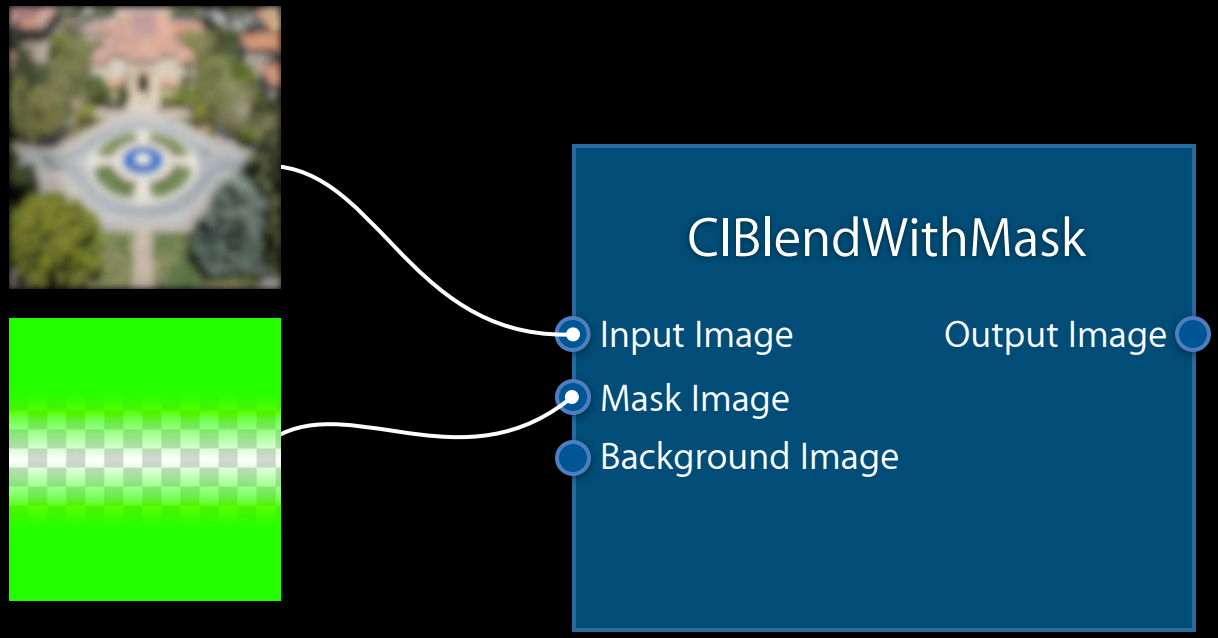
Final Blending



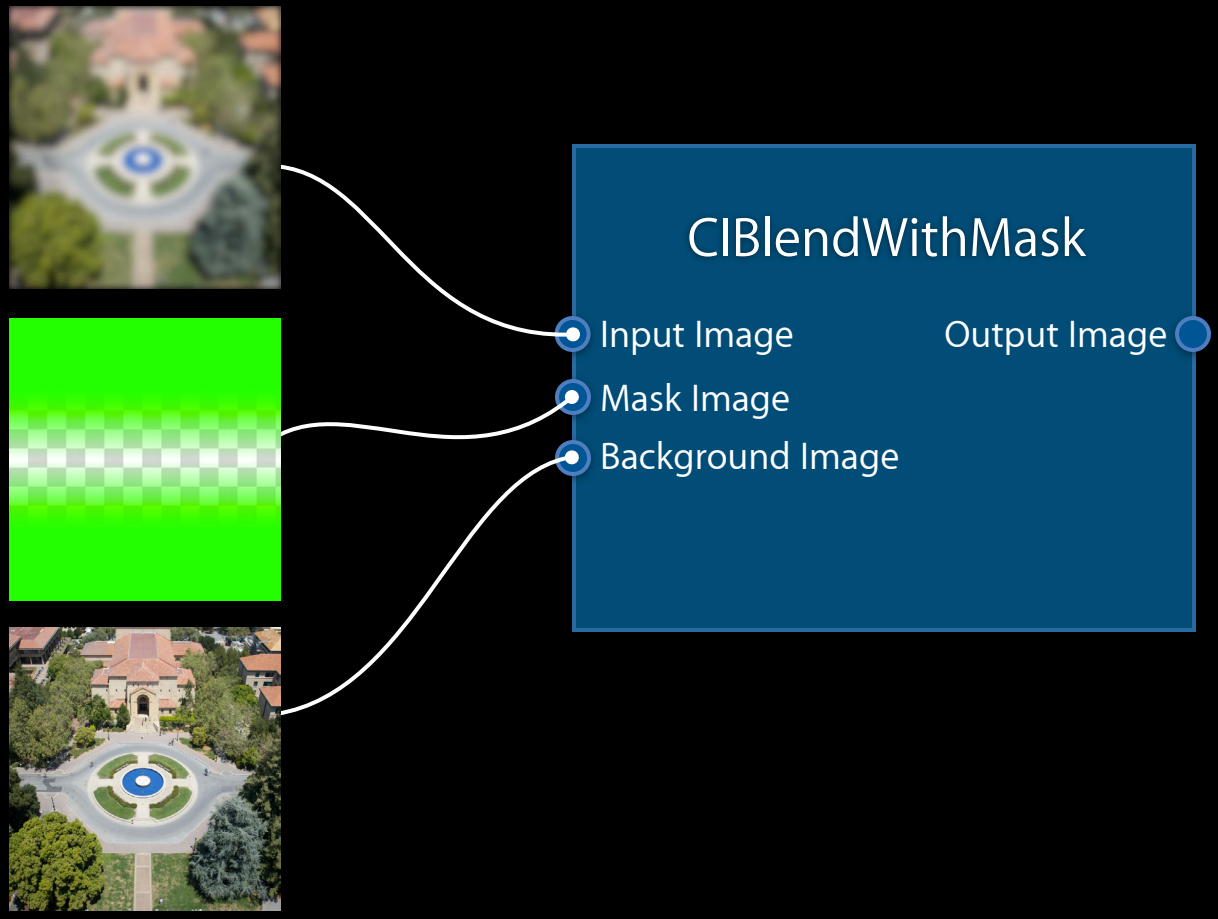
Final Blending



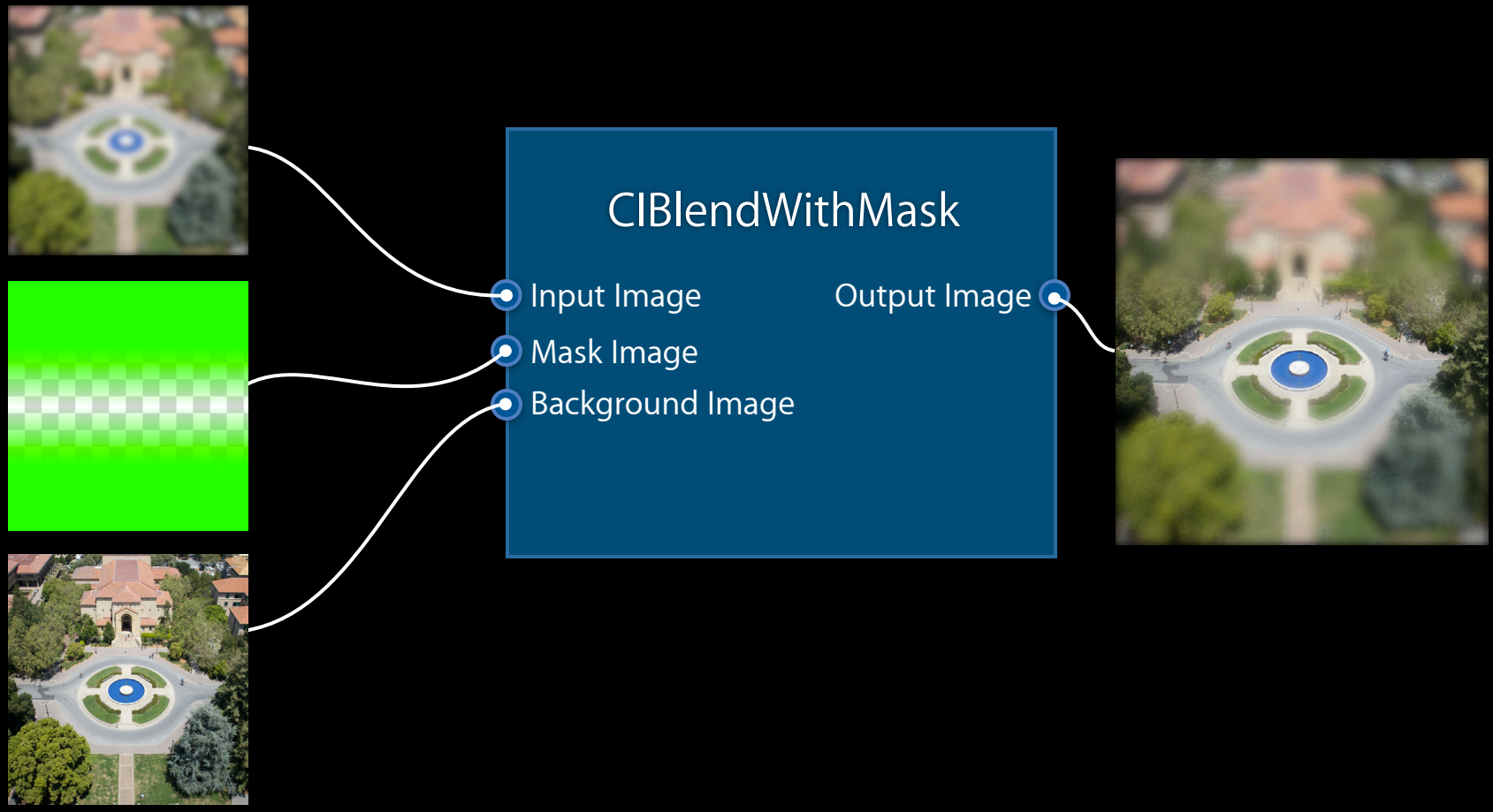
Final Blending



Final Blending



Final Blending



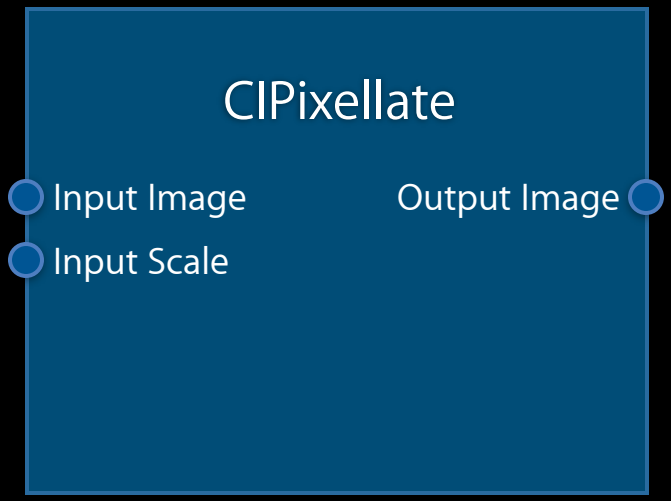
Recipe Four: Anonymize Faces

Recipe Four: Anonymize Faces

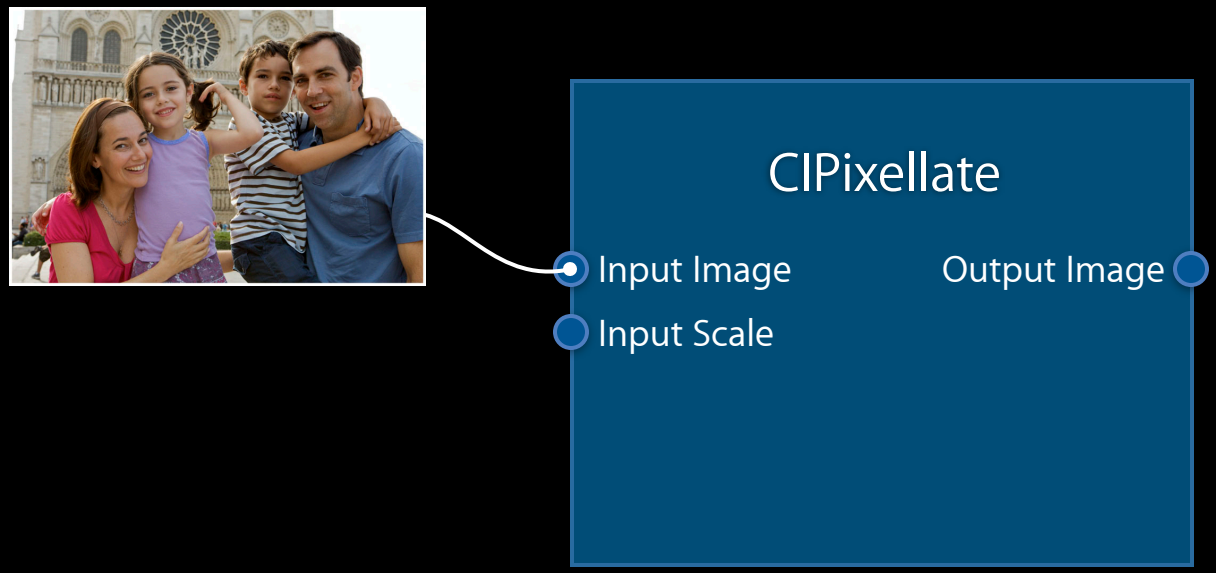
Overview

- Create pixellated image
- Build mask by using face detector and find all the faces in input image
  - For each face
    - Use CIRadialGradient to create a circle
    - Use source over compositing to add it to the mask
- Blend pixellated image with original using mask

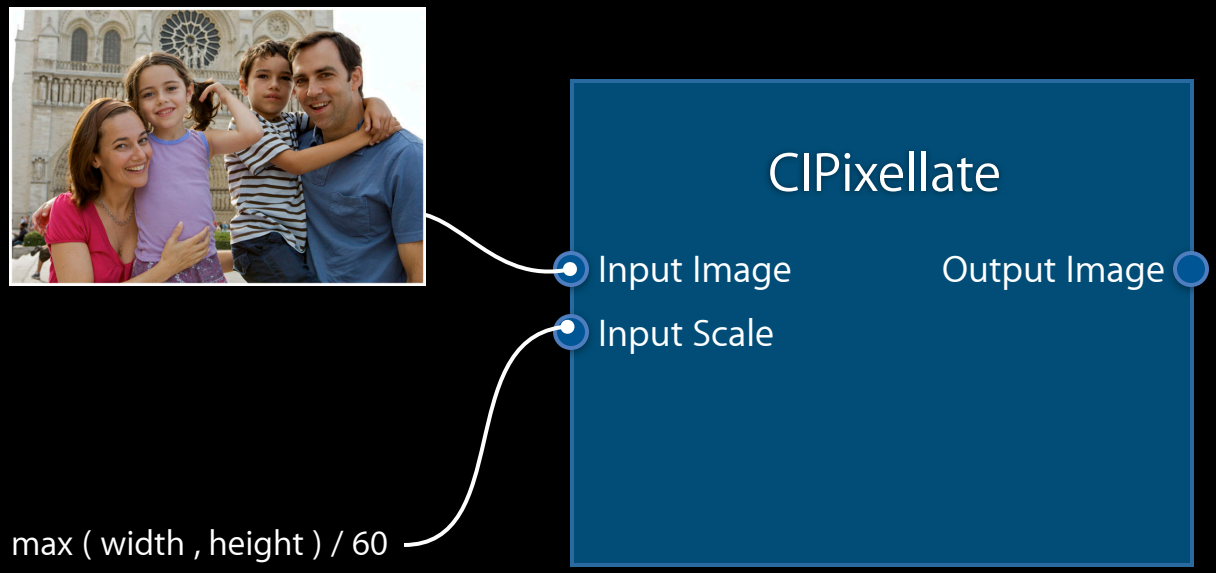
Pixellate Entire Image



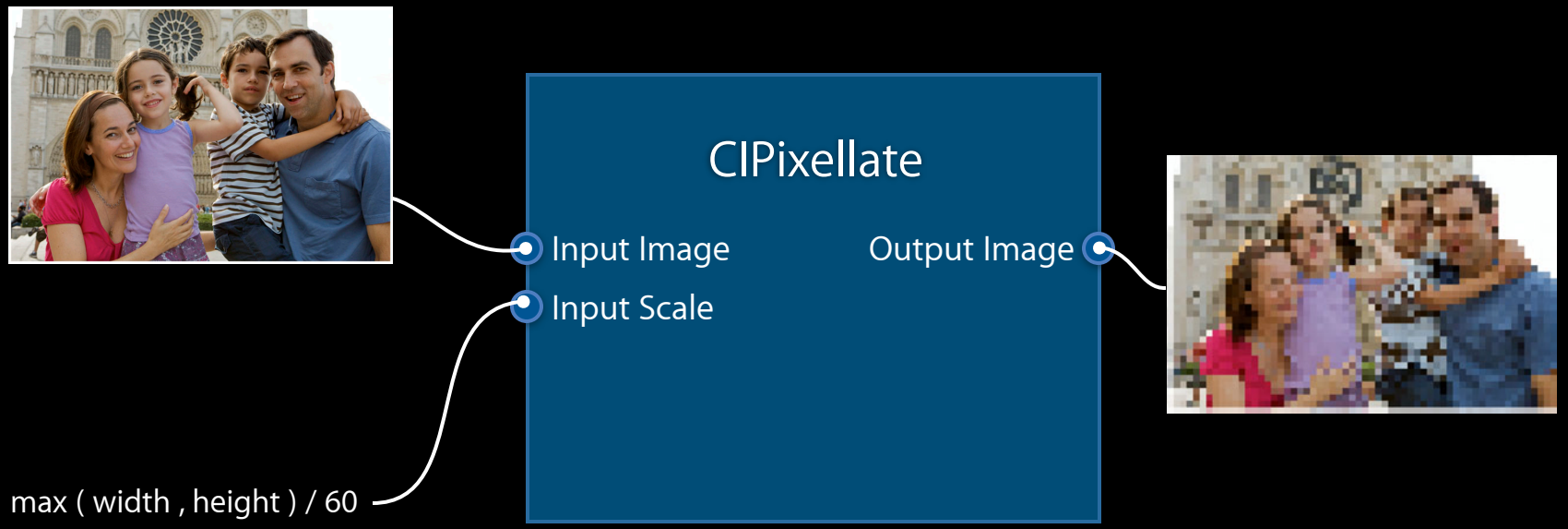
Pixellate Entire Image



Pixellate Entire Image



Pixellate Entire Image



Find Faces and Create Mask

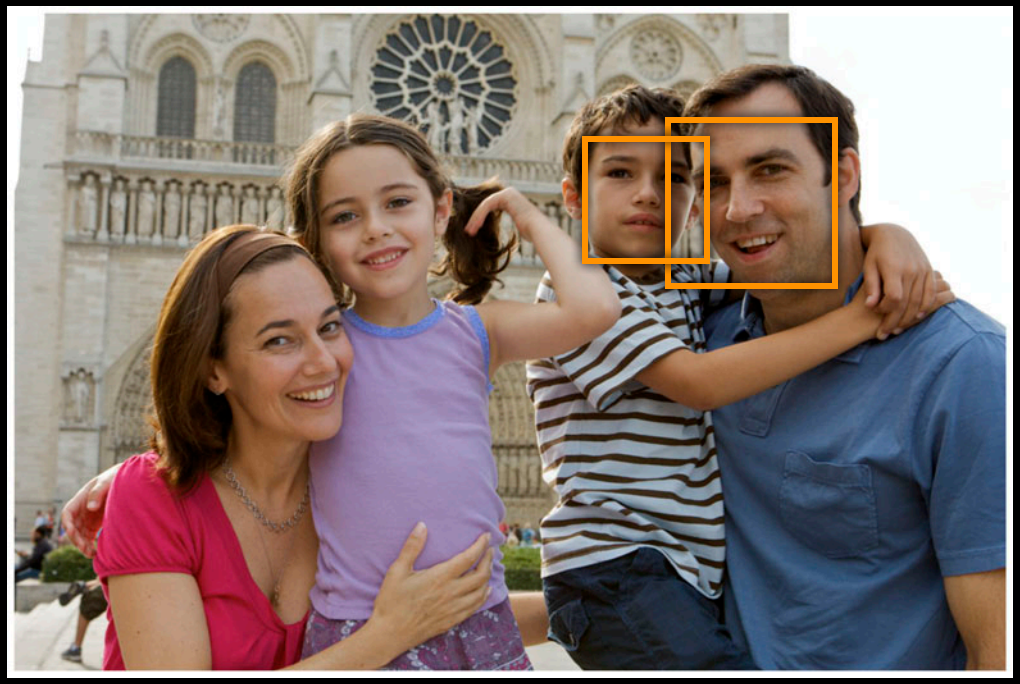
```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```

Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```

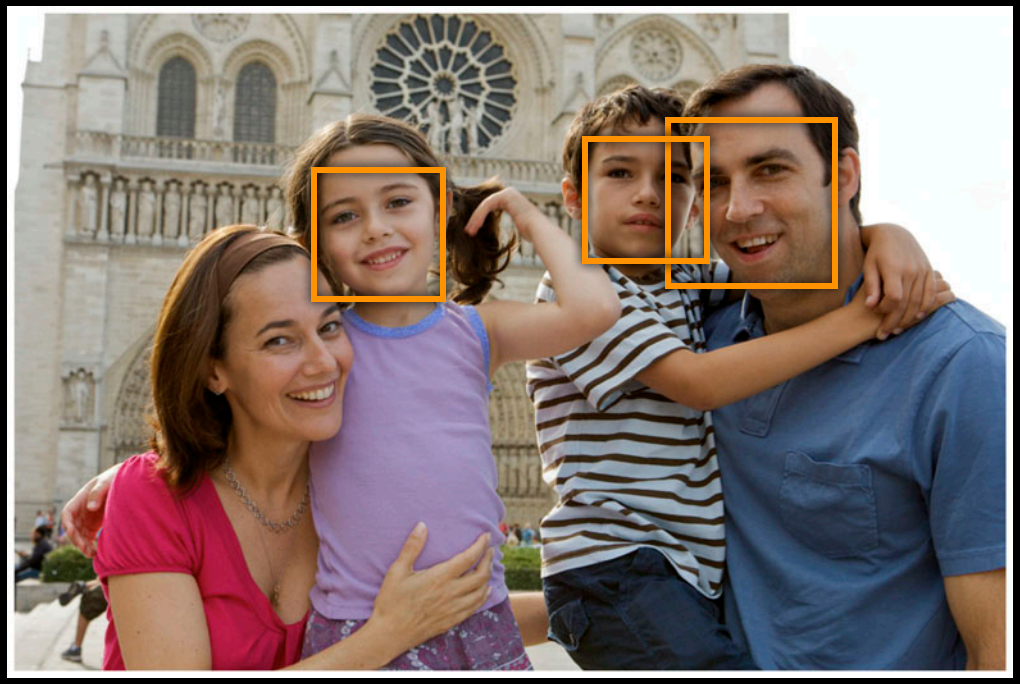
Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



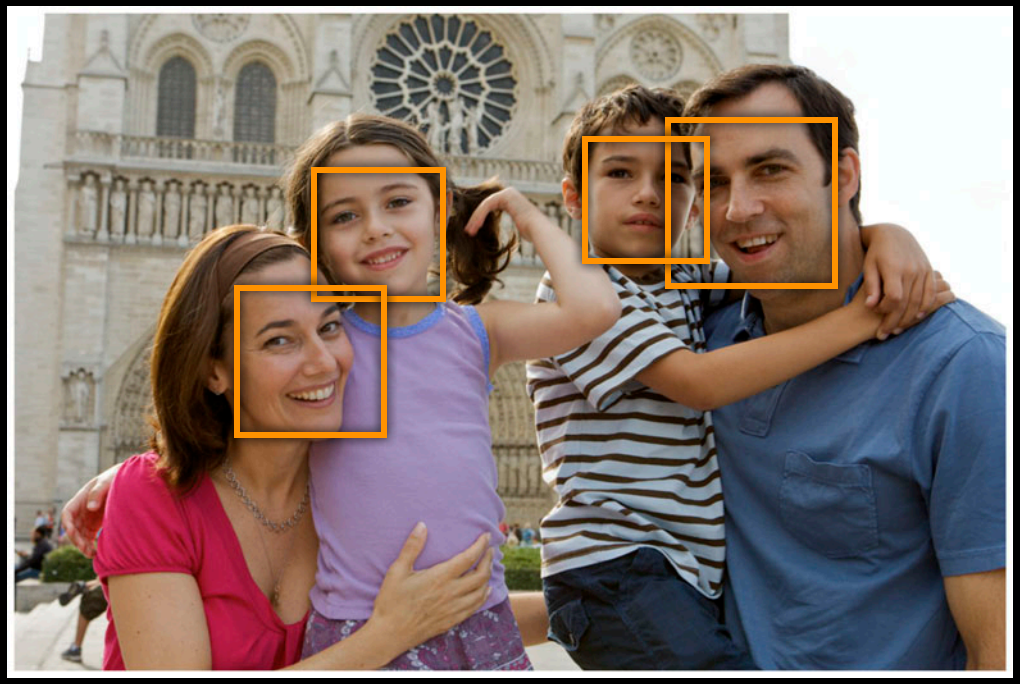
Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



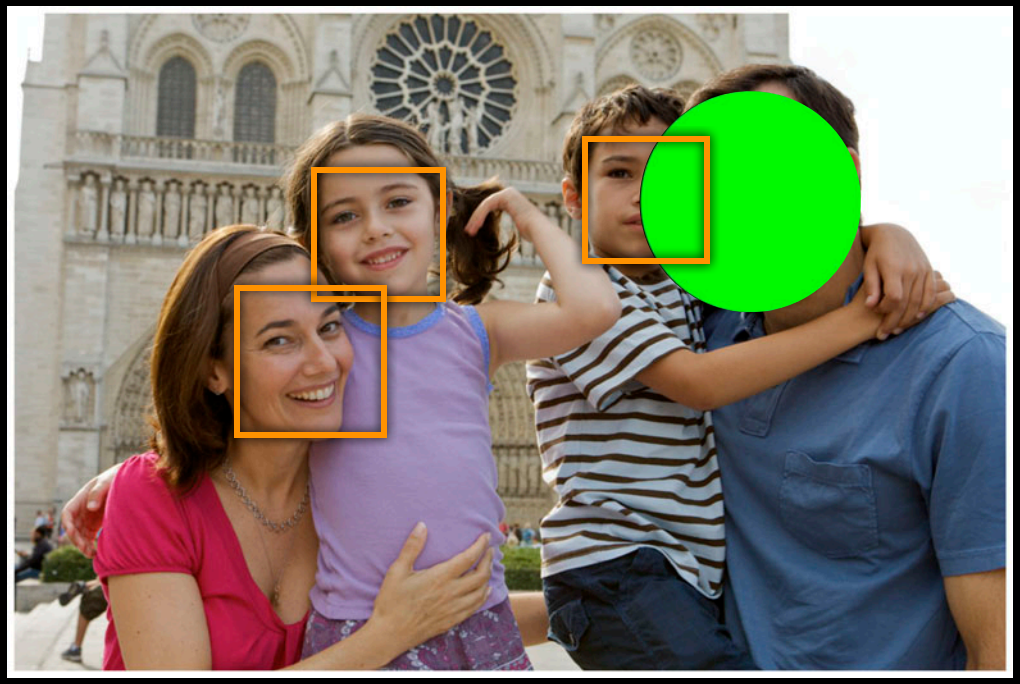
Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



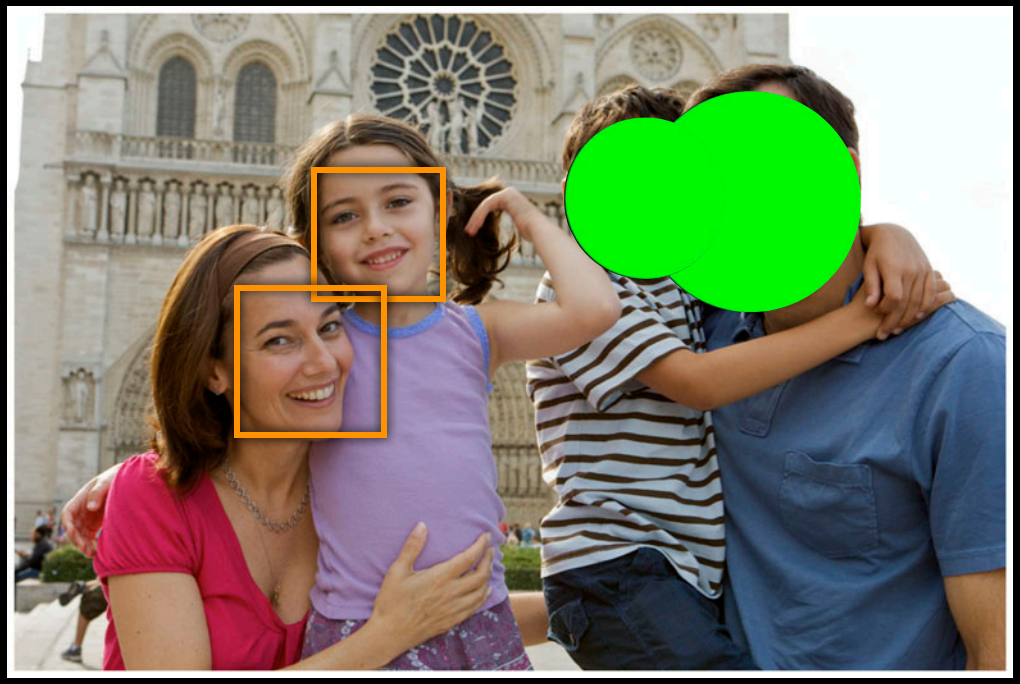
Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



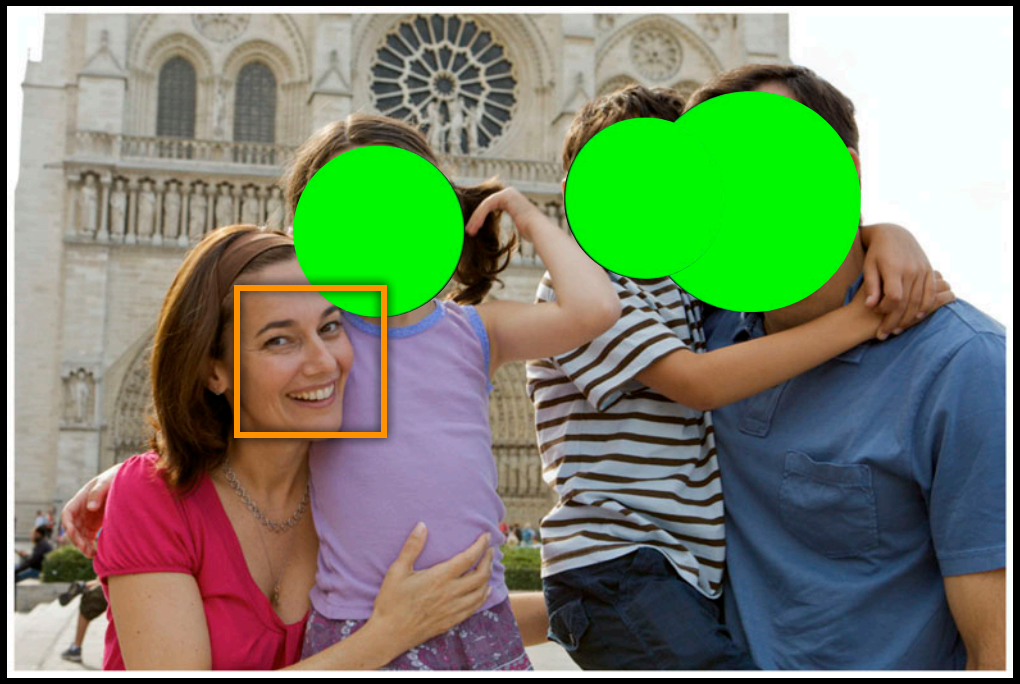
Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```

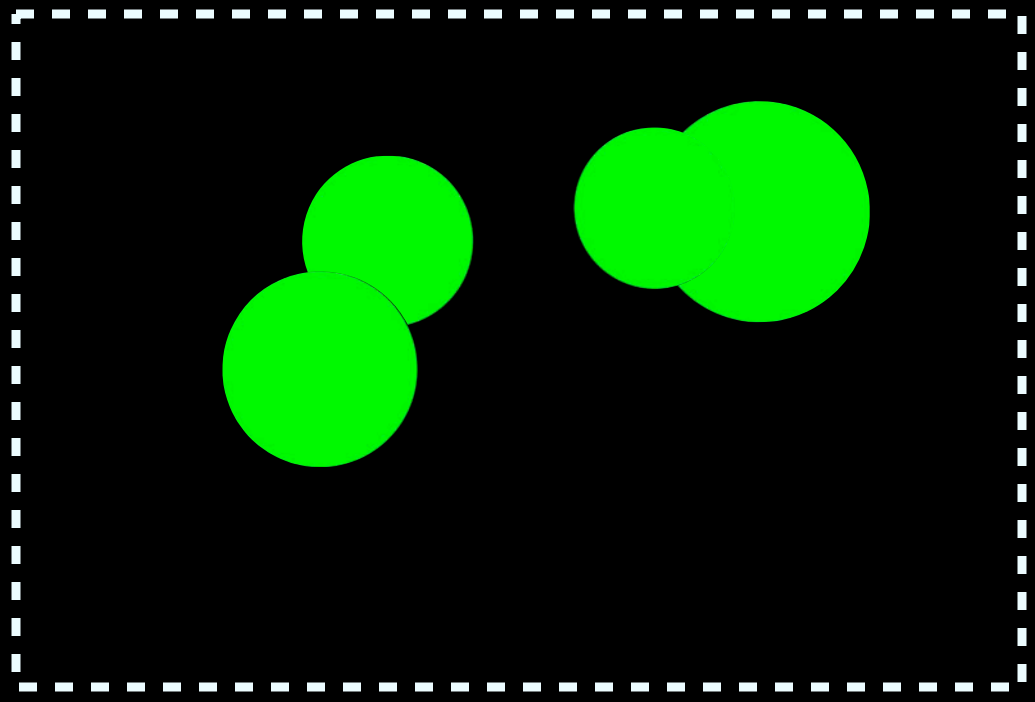


Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```

Find Faces and Create Mask

```
CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:nil
                                          options:nil];
NSArray* faceArray = [detector featuresInImage:image options:nil];
```



Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

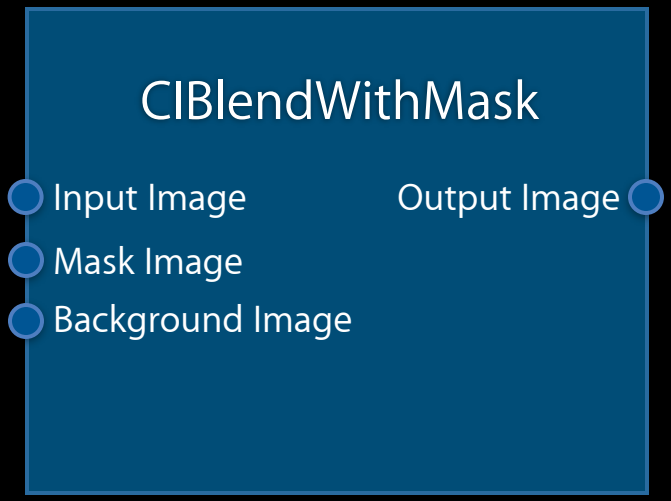
Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

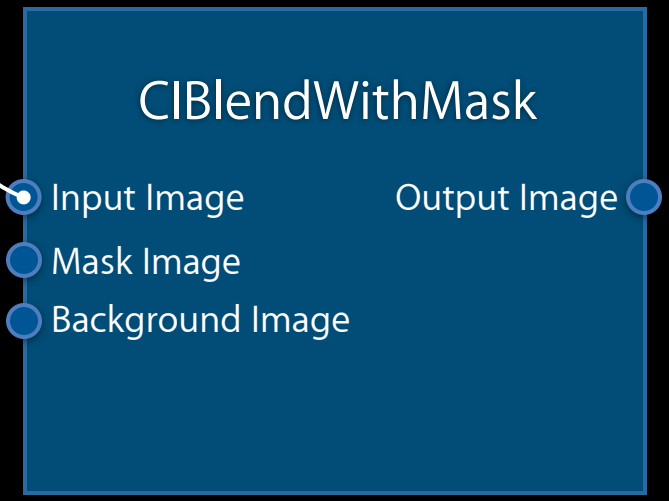
Creating the Mask, in Code

```
CIImage *maskImage = nil;

for ( CIFeature *f in faces ) {
   CIVector *cen = [CIVector vectorWithX:f.bounds.origin.x+f.bounds.size.width/2.0 Y: ....];
   CGFloat radius = MIN ( [f bounds].size.width, [f bounds].size.height ) / 1.5;
   CIFilter *radialGradient = [CIFilter filterWithName:@"CIRadialGradient" keysAndValues:
      @”inputRadius0”, [NSNumber numberWithFloat:radius],
      @”inputRadius1”, [NSNumber numberWithFloat:radius+1.0f],
      @”inputColor0”, [CIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0],
      @”inputColor1”, [CIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.0],
      @”inputCenter”, cen, nil];

   CIImage *circleImage =[radialGradient valueForKey:kCIOutputImageKey];

   if ( nil == maskImage )
     maskImage = circleImage;
   else
     maskImage = [[CIFilter filterWithName:@"CISourceOverCompositing" keysAndValues:
                kCIInputImageKey, circleImage, kCIInputBackgroundImageKey, maskImage,
                nil] valueForKey:kCIOutputImageKey];
}
```

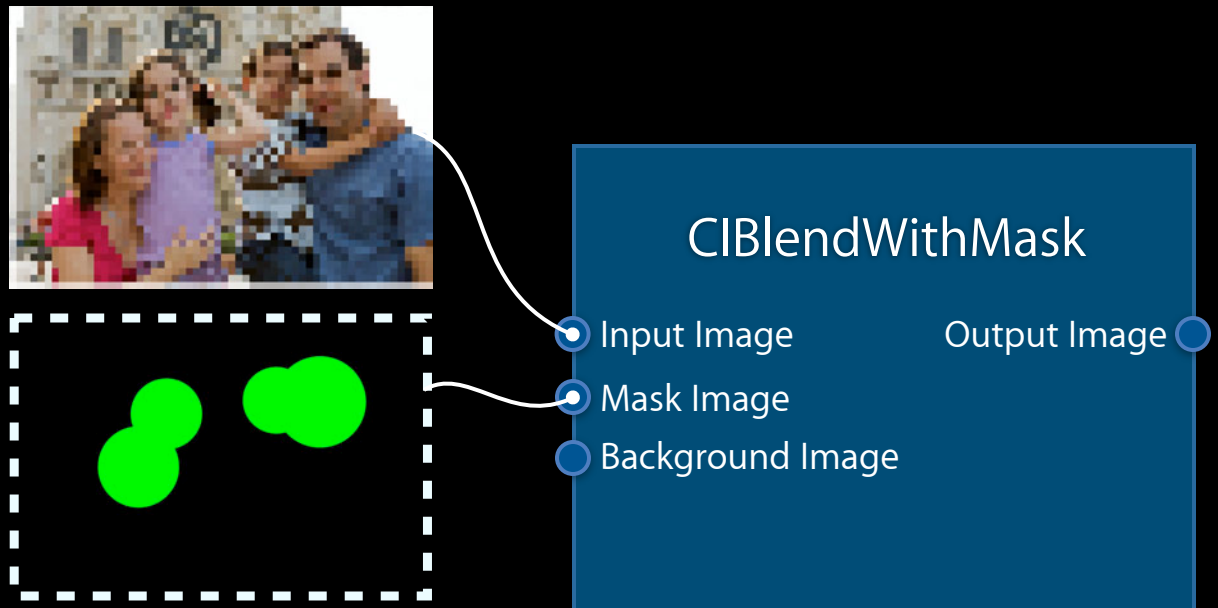
Pixel People



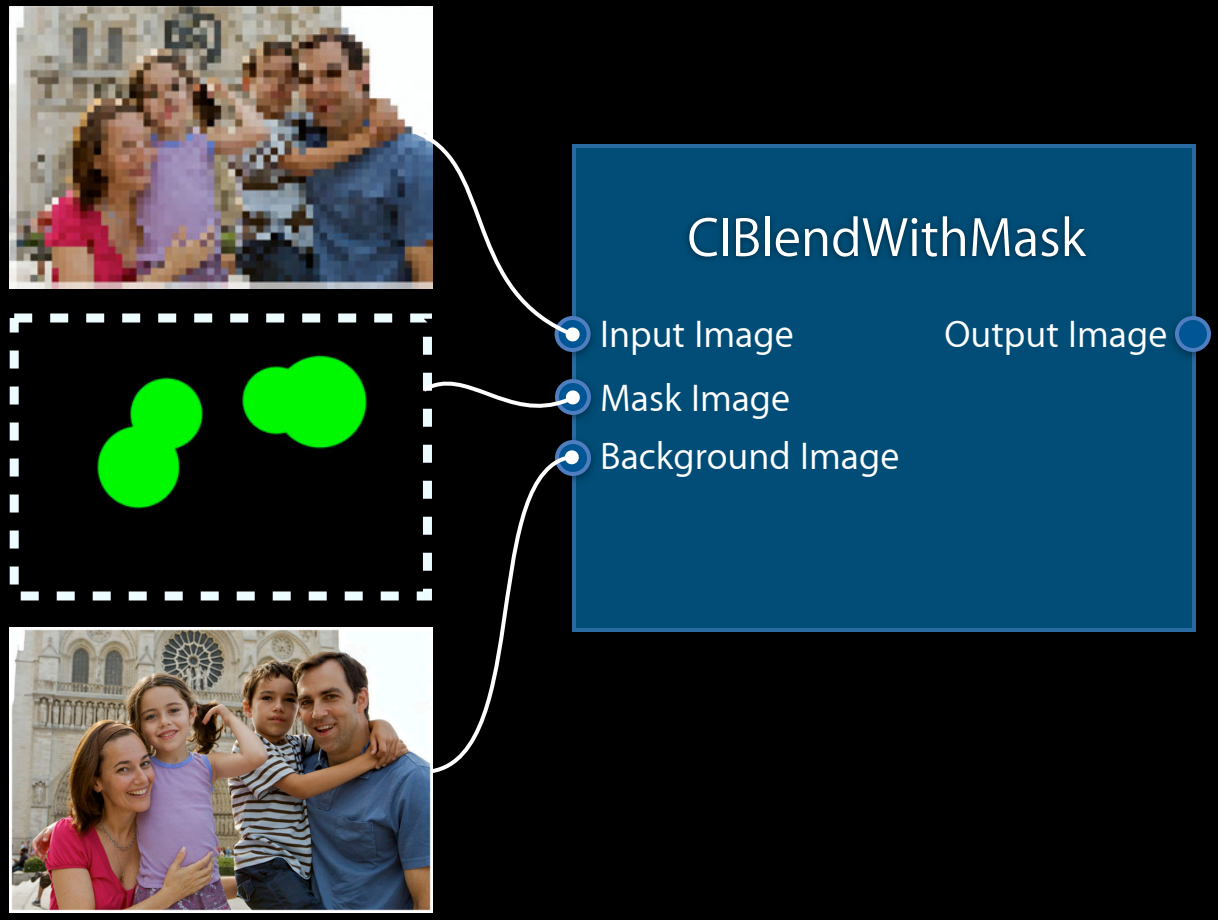
Pixel People



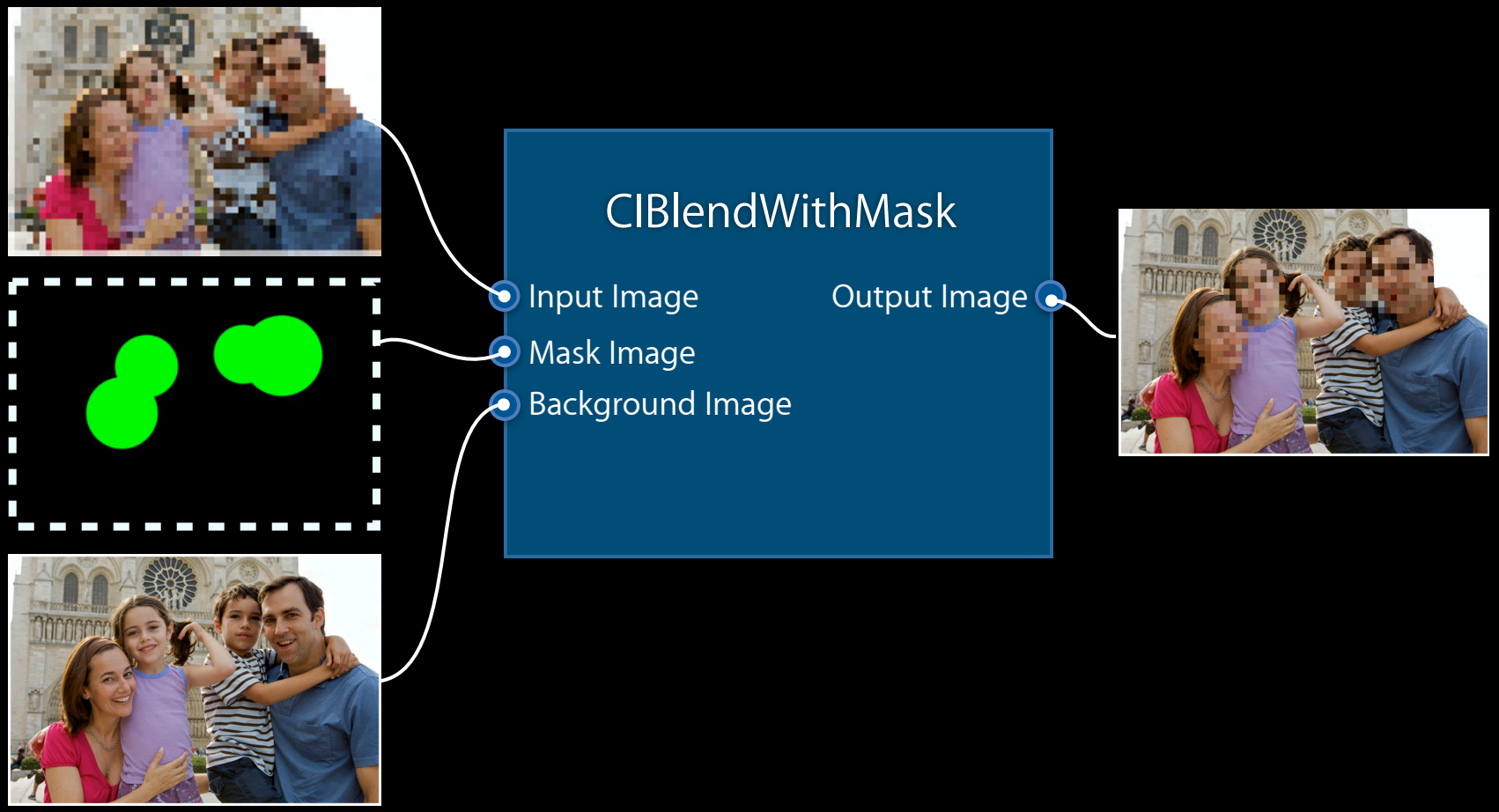
Pixel People



Pixel People



Pixel People



Recipe Five: Pixellate Transition

Recipe Five: Pixellate Transition

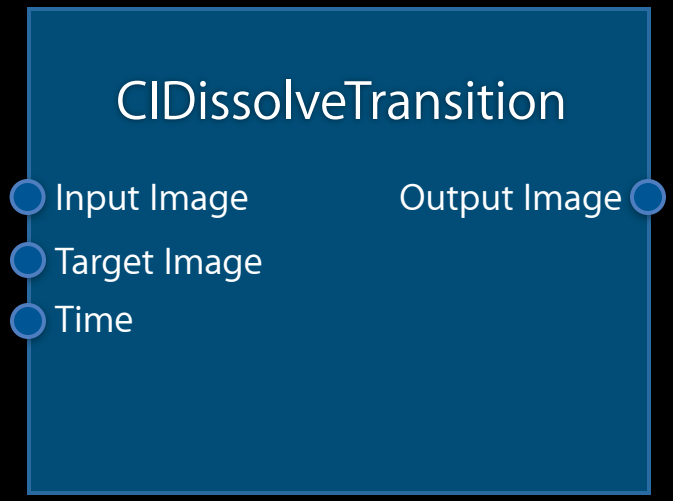
Recipe Five: Pixellate Transition

Overview

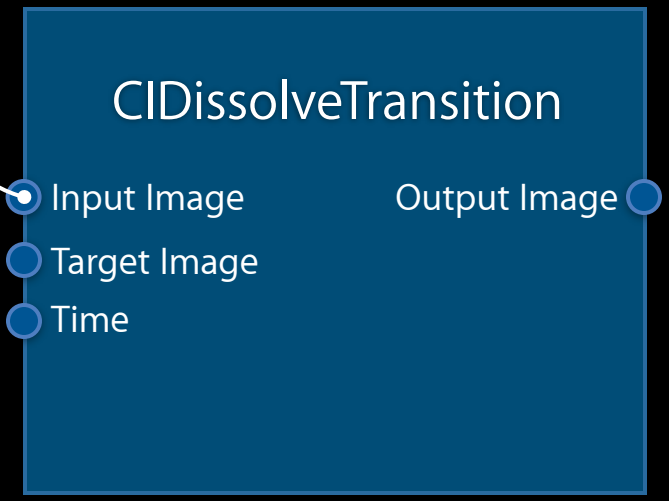
Overview

- Use CIDissolveTransition to blend in between both input images
- Pixellate result of transition filter and vary scale over time

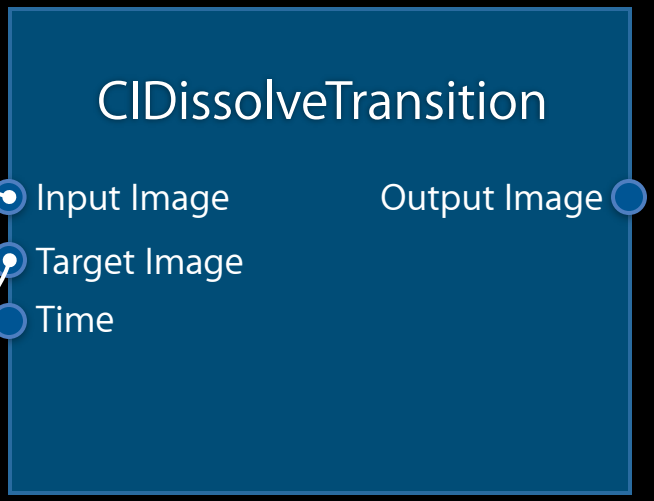
Dissolve Transition



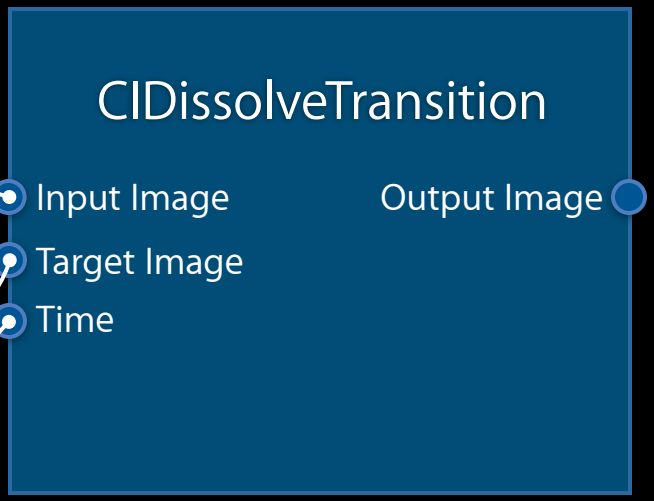
Dissolve Transition



Dissolve Transition

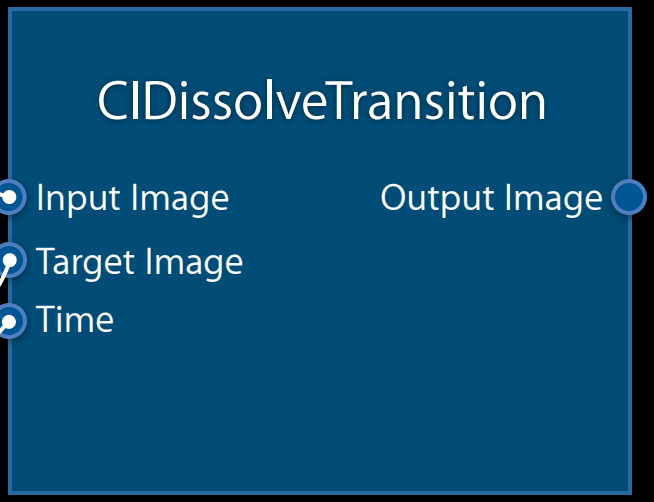


Dissolve Transition

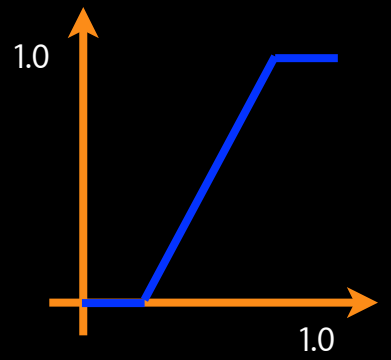


min (max (2 * (time - 0.25) , 0) ,1)

Dissolve Transition



min (max (2 * (time - 0.25) , 0) ,1)



Dissolve Transition

CIDissolveTransition

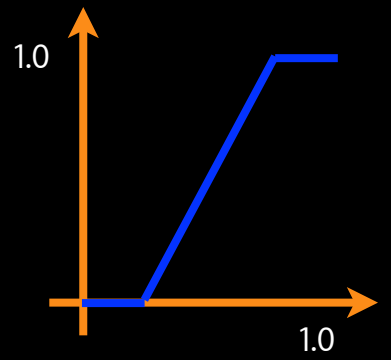
Input Image

Target Image

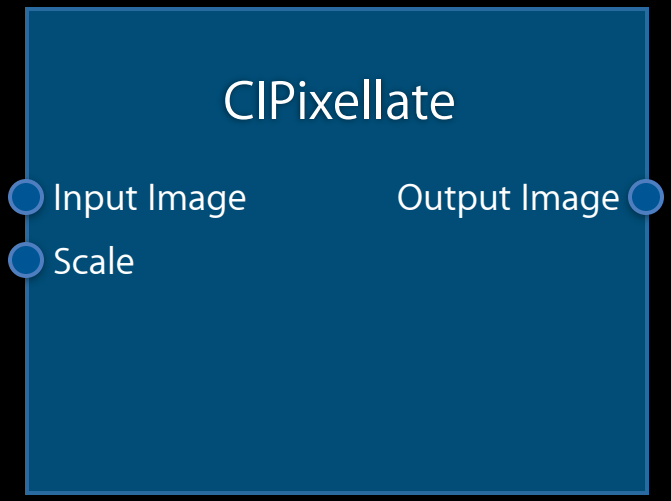
Time

Output Image

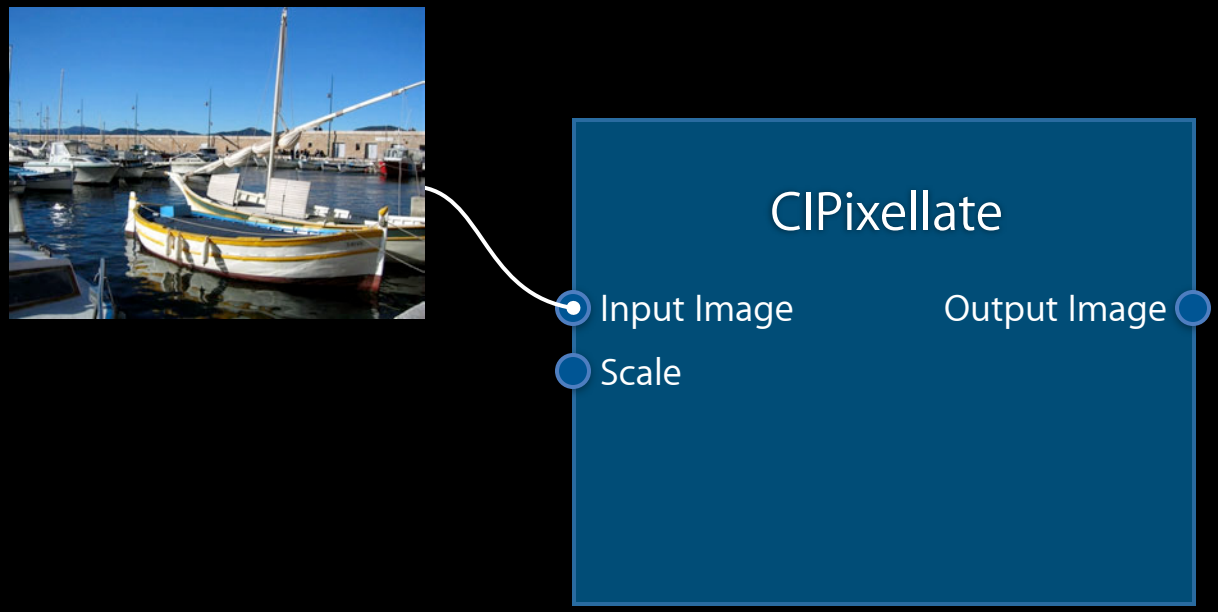
min (max (2 * (time - 0.25) , 0) ,1)



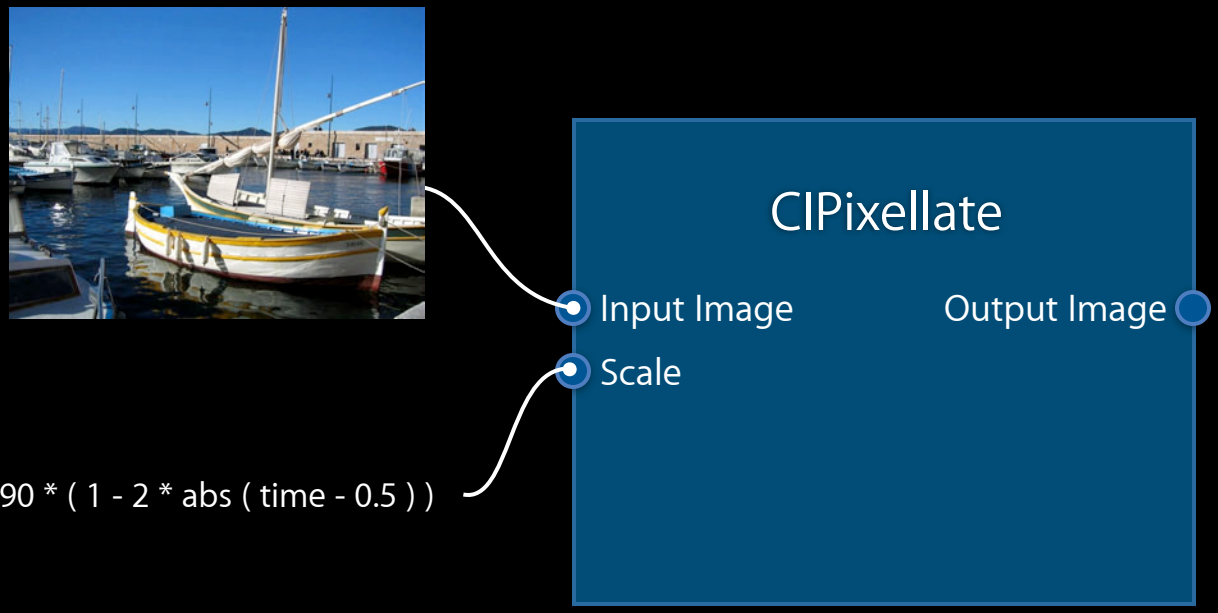
Pixellate the Result of the Transition



Pixellate the Result of the Transition



Pixellate the Result of the Transition



Pixellate the Result of the Transition

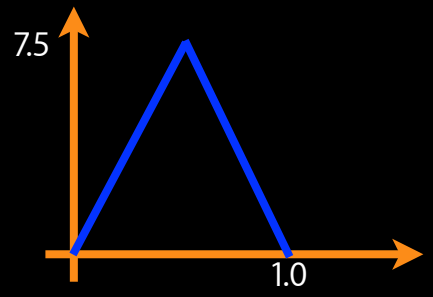
CIPixellate

Input Image

Output Image

Scale

90 * (1 - 2 * abs (time - 0.5))



Pixellate the Result of the Transition

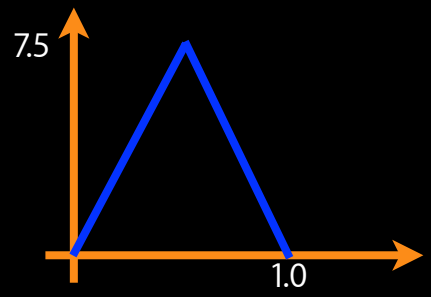
CIPixellate

Input Image

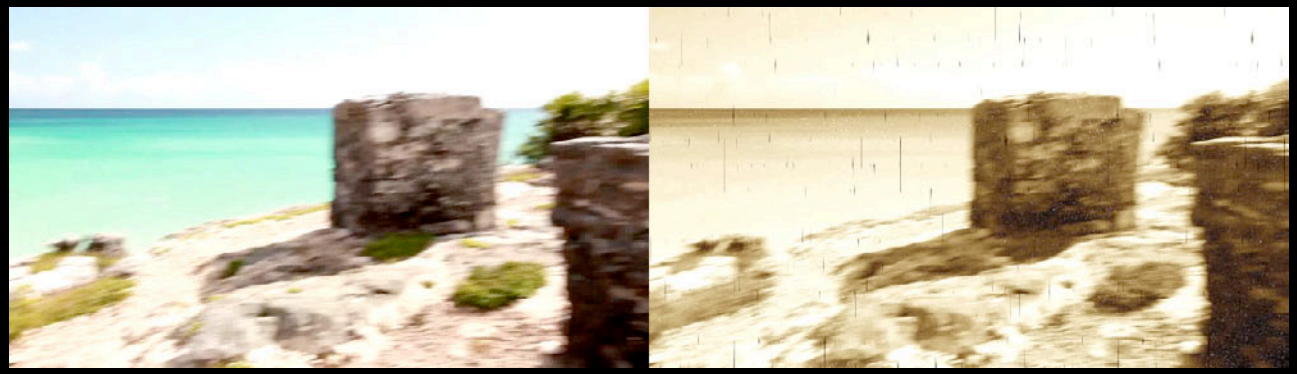
Scale

Output Image

90 * (1 - 2 * abs (time - 0.5))



Recipe Six: Olde Film

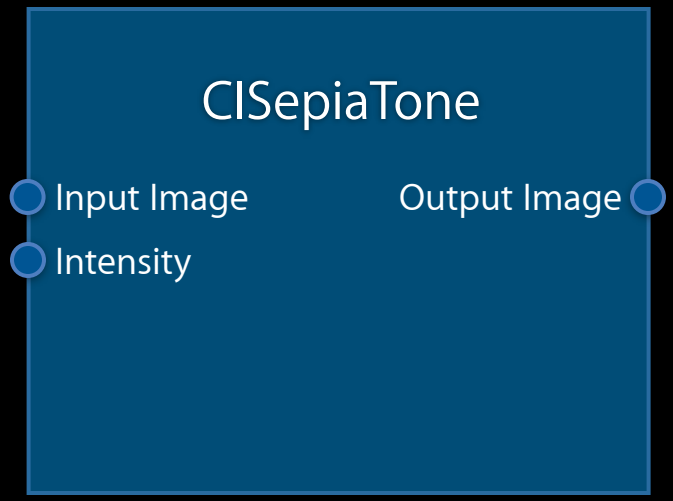


Overview

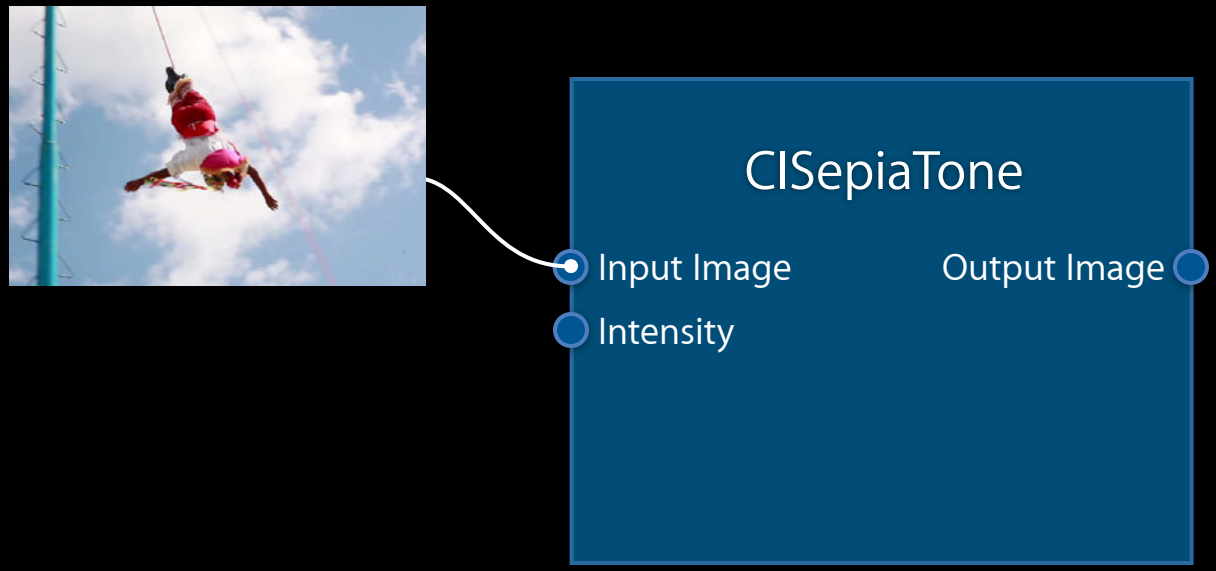
Overview

- Apply sepia
- Add white specks
- Add dark scratches
- Composite everything together

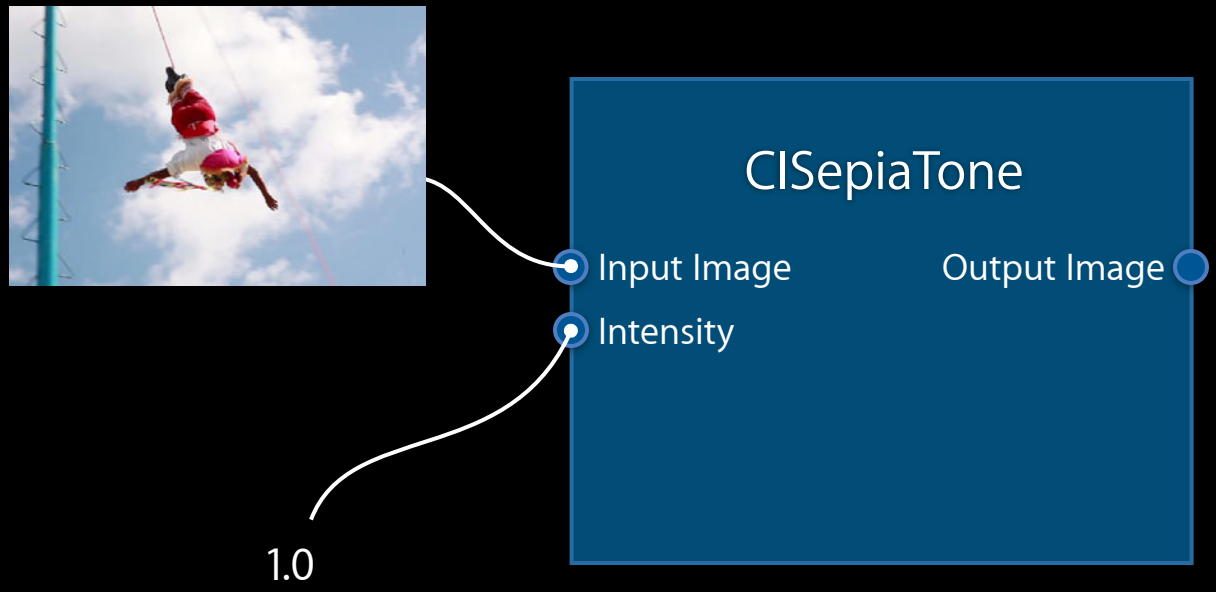
Sepia Image



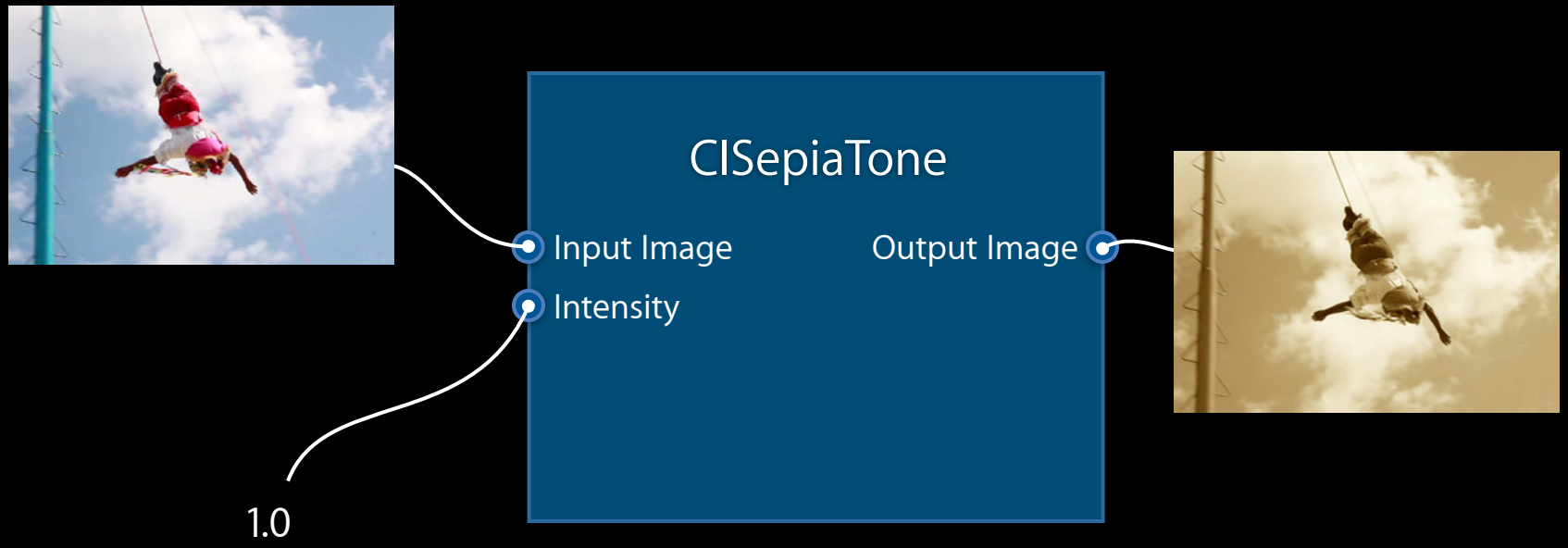
Sepia Image



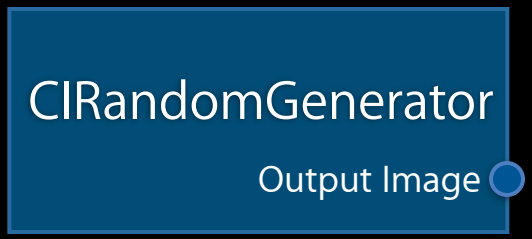
Sepia Image



Sepia Image

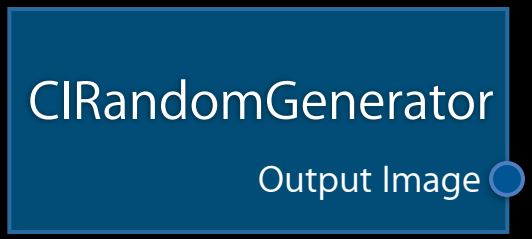


Adding the White Specks

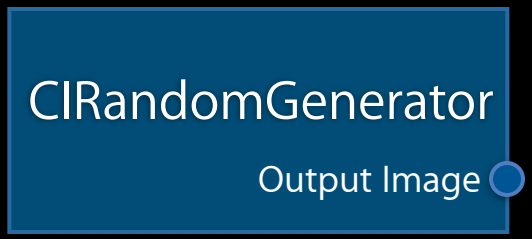


Adding the White Specks

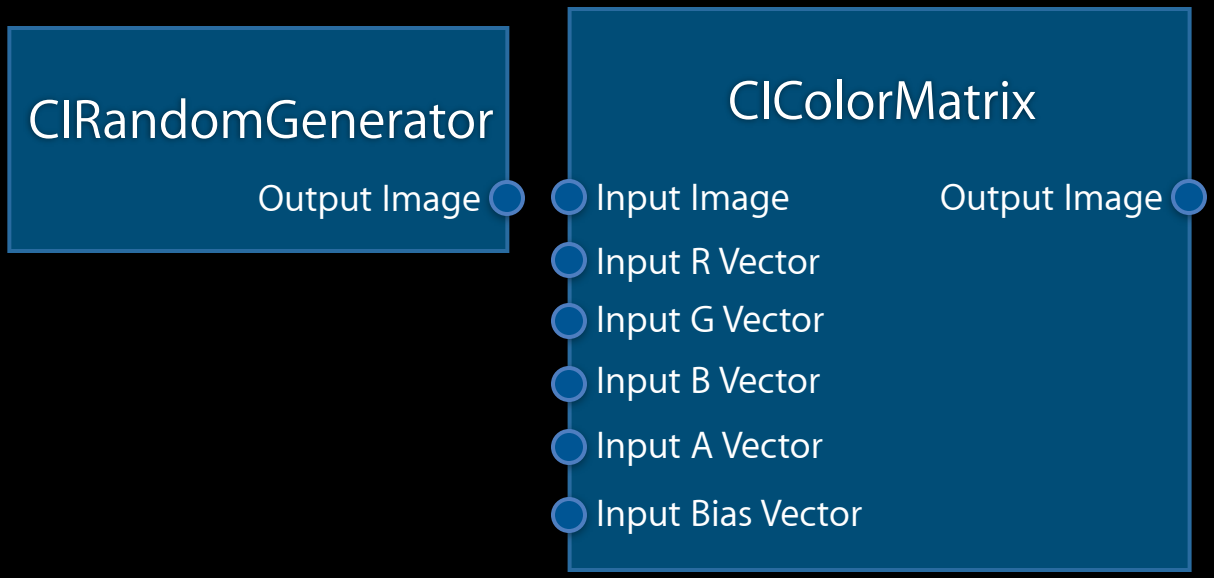
Adding the White Specks



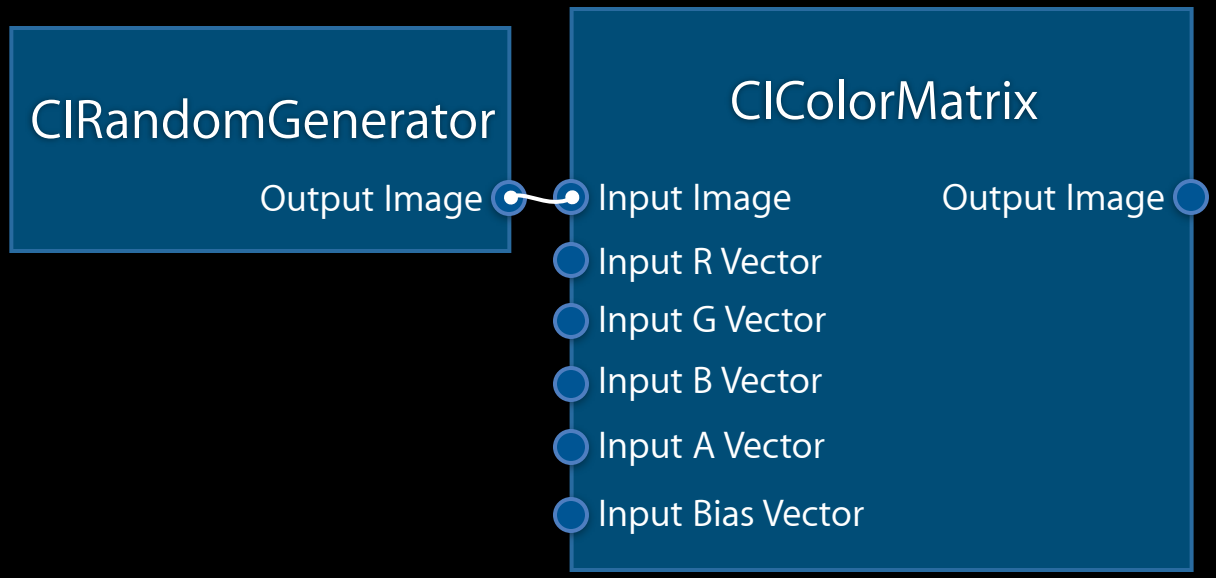
Adding the White Specks



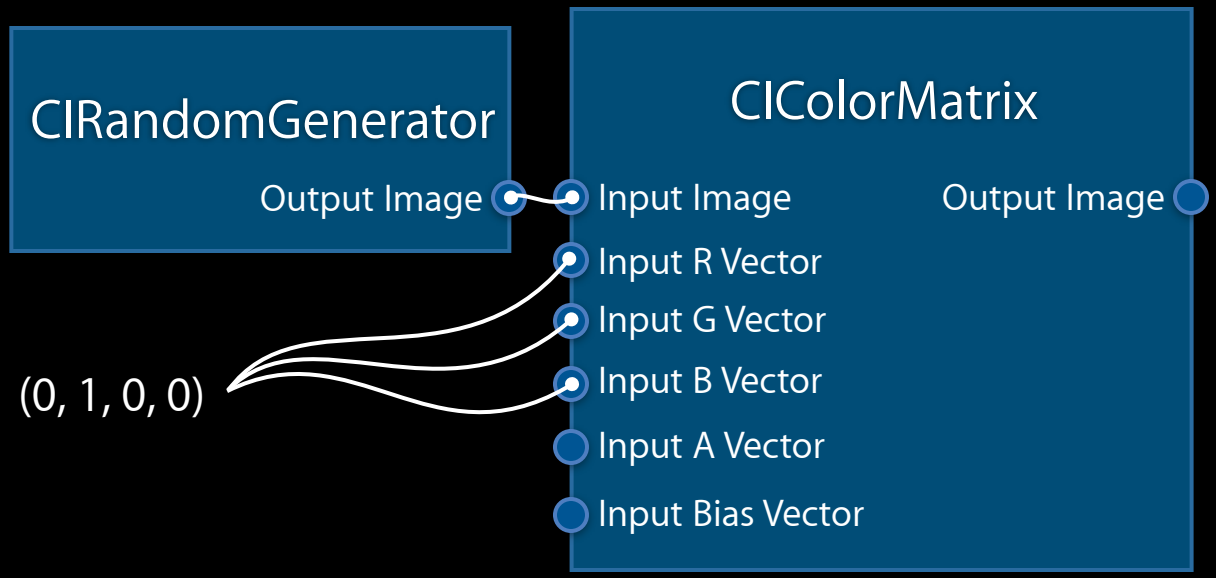
Adding the White Specks



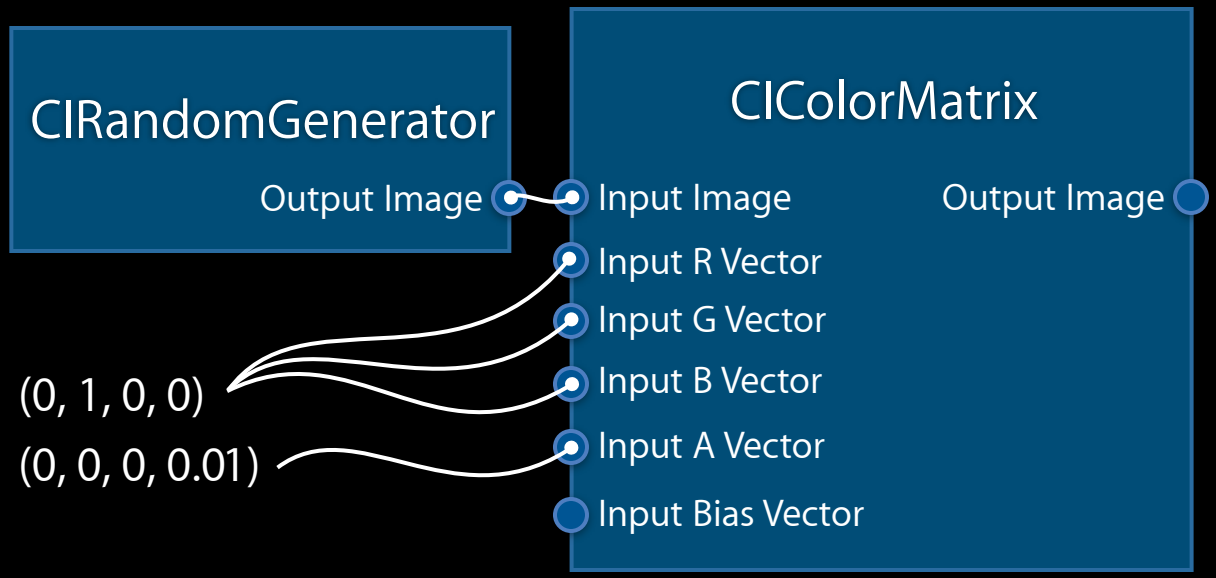
Adding the White Specks



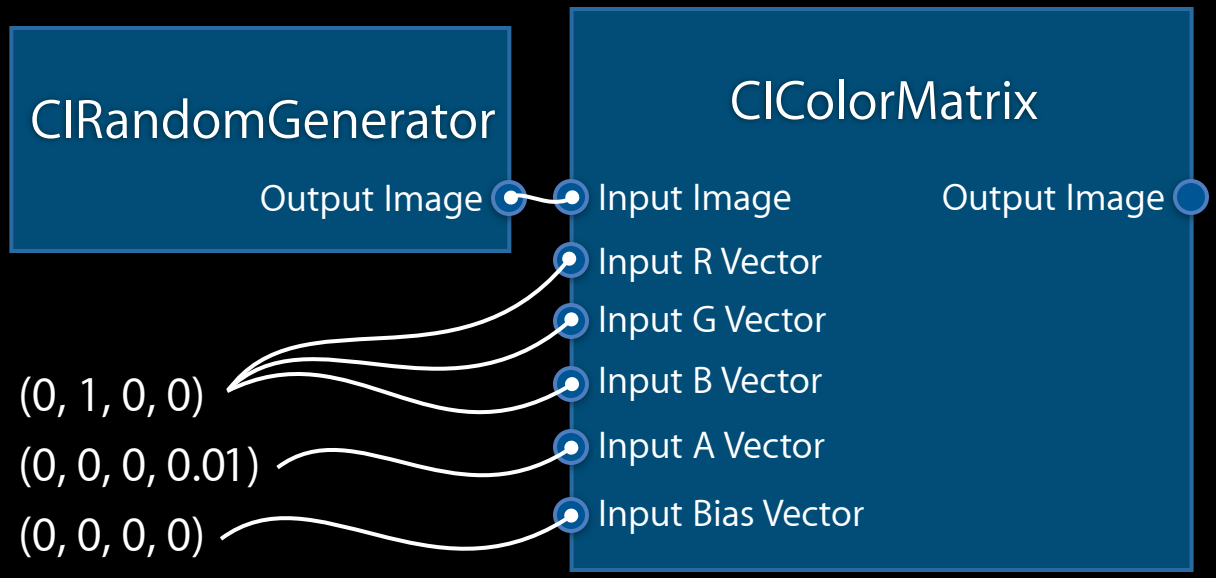
Adding the White Specks



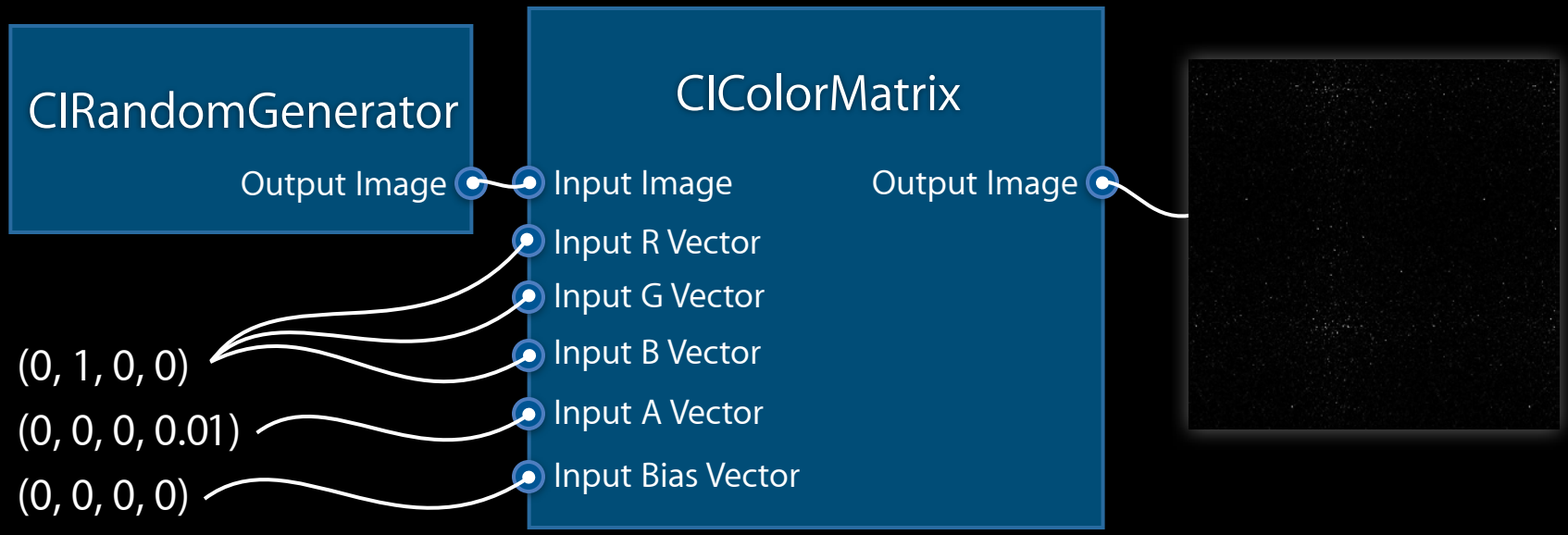
Adding the White Specks



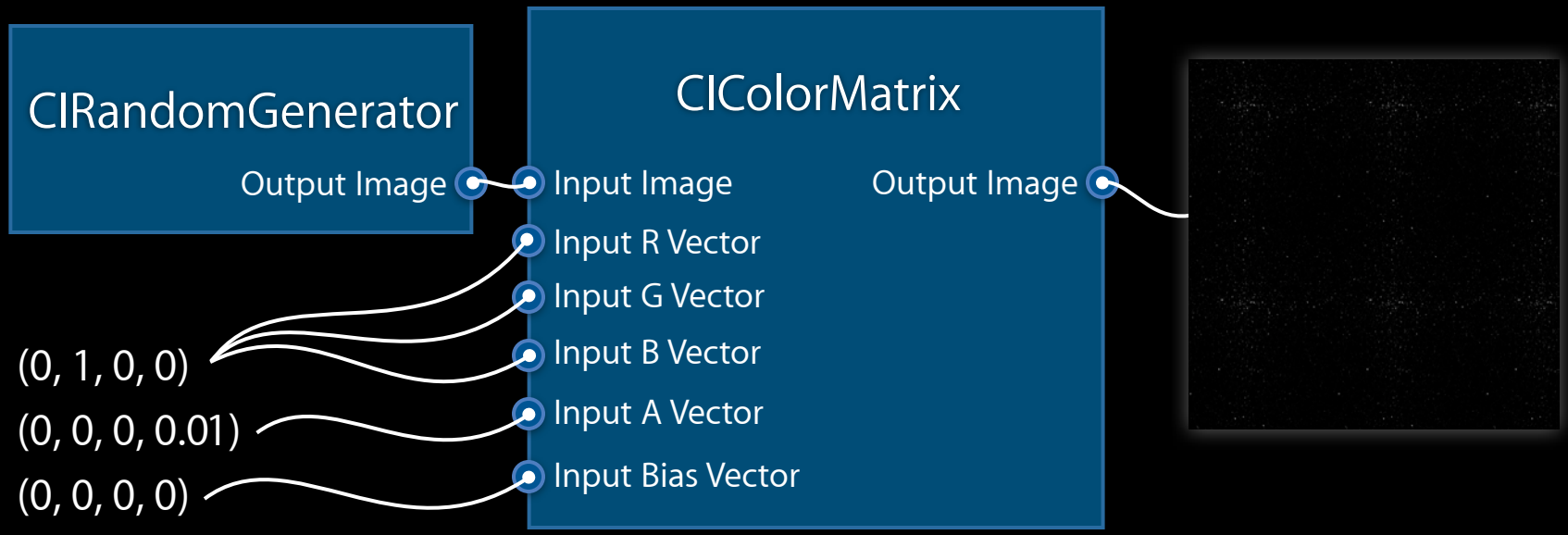
Adding the White Specks



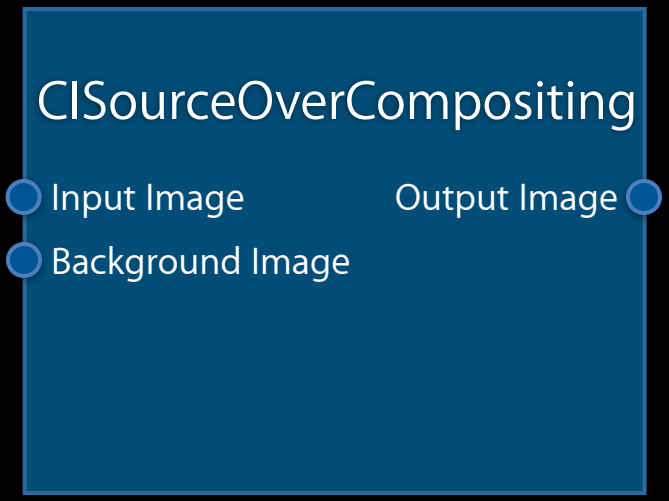
Adding the White Specks



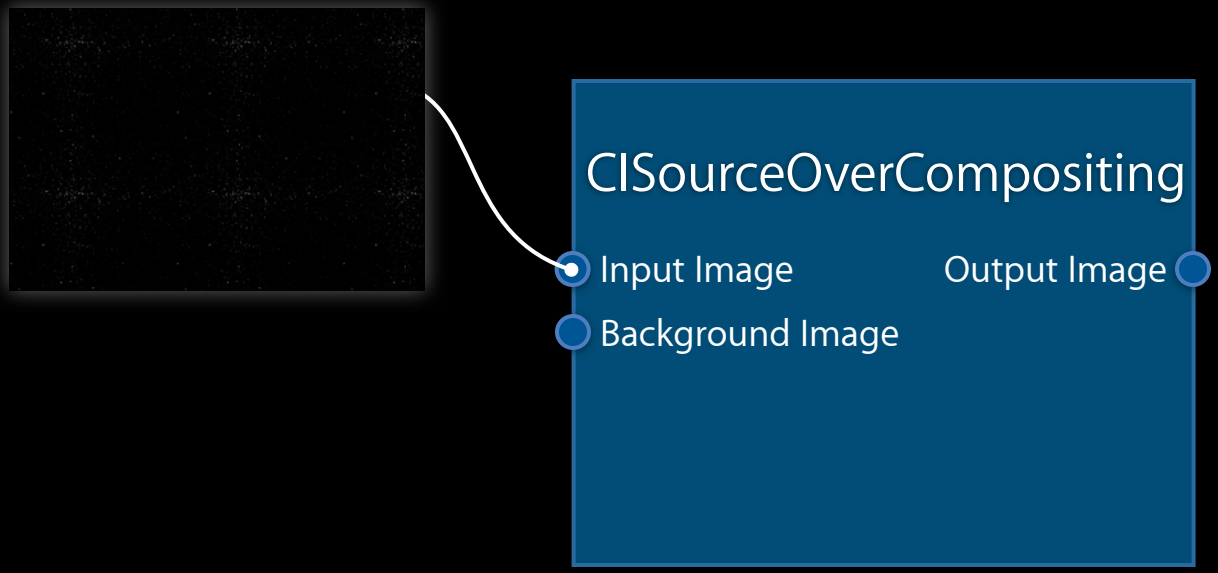
Adding the White Specks



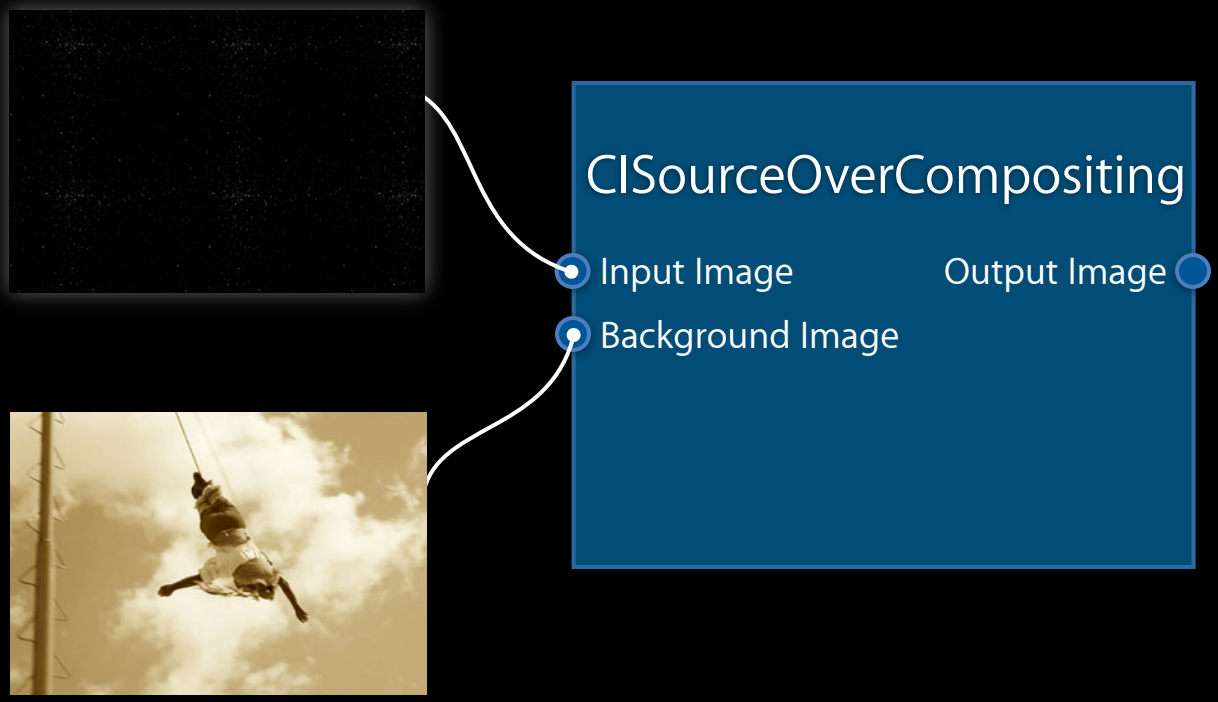
Blend Noise Image with Sepia Image



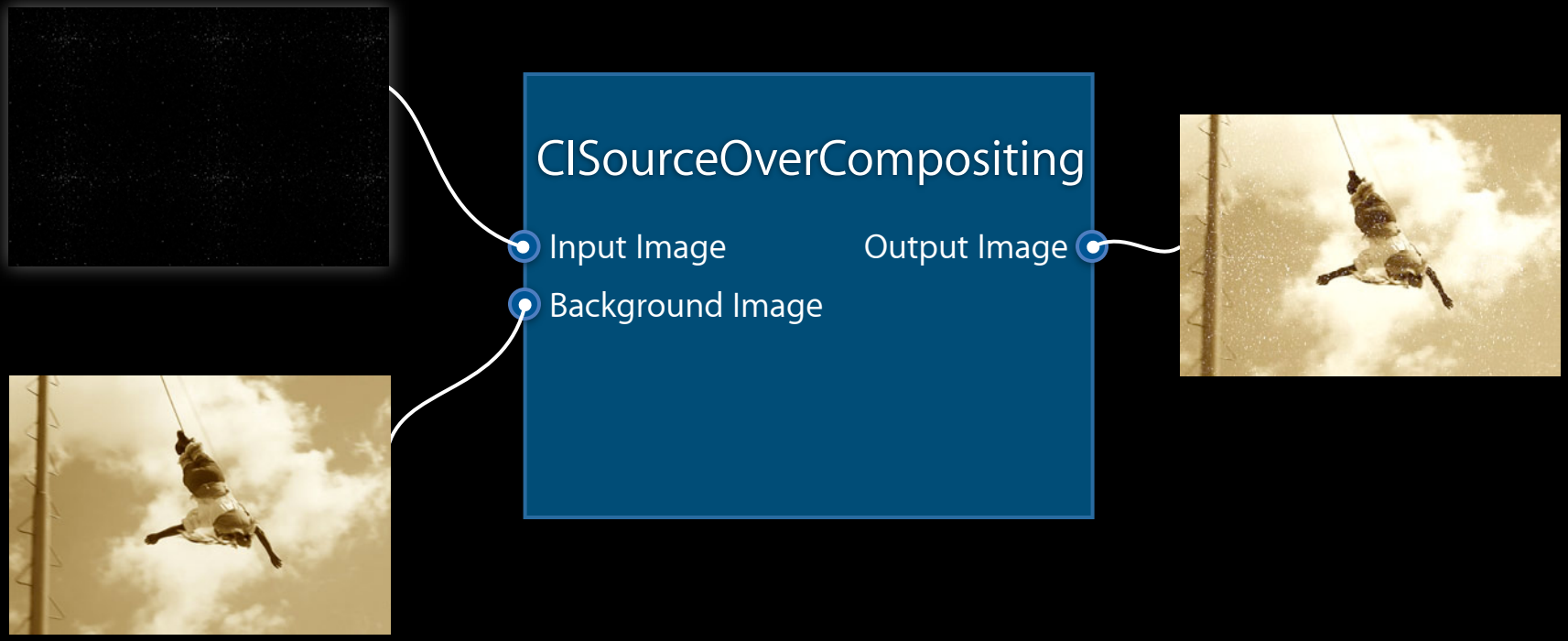
Blend Noise Image with Sepia Image



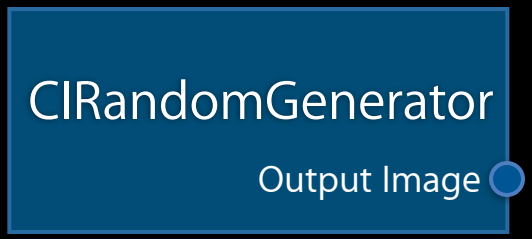
Blend Noise Image with Sepia Image



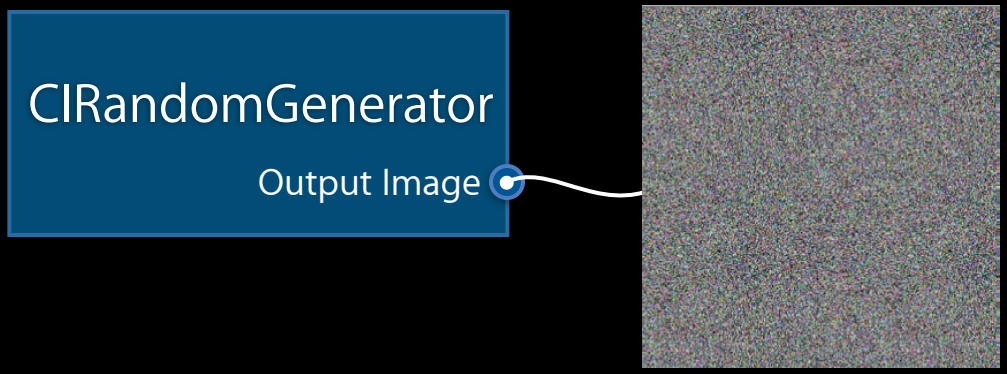
Blend Noise Image with Sepia Image



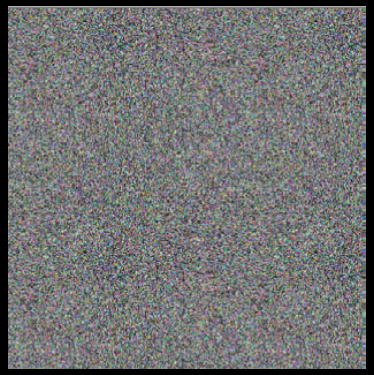
Add Dark Scratches



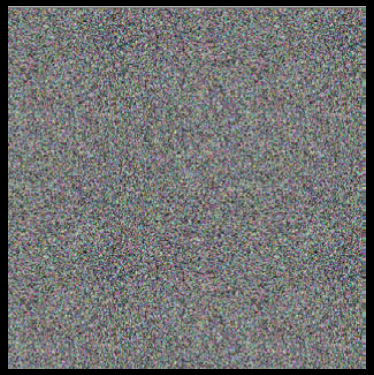
Add Dark Scratches



Add Dark Scratches

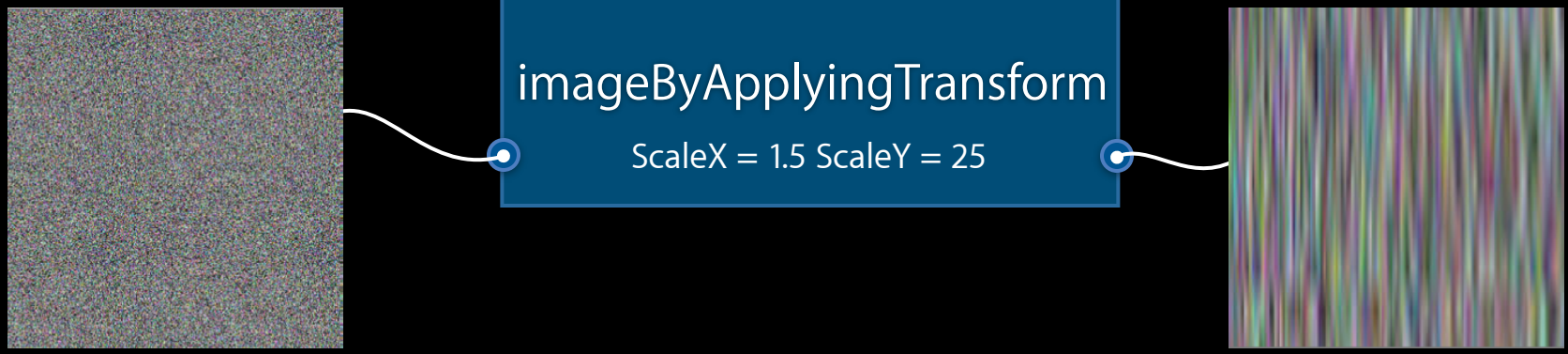


Add Dark Scratches

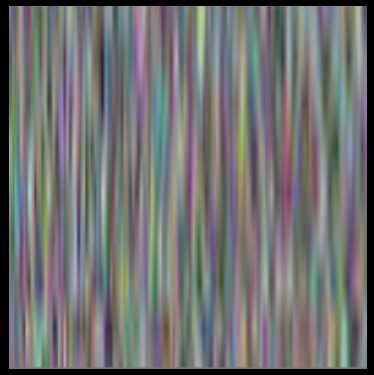


Add Dark Scratches

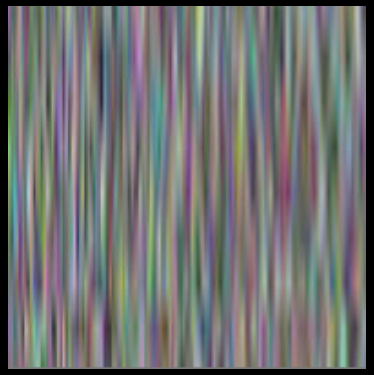
Add Dark Scratches



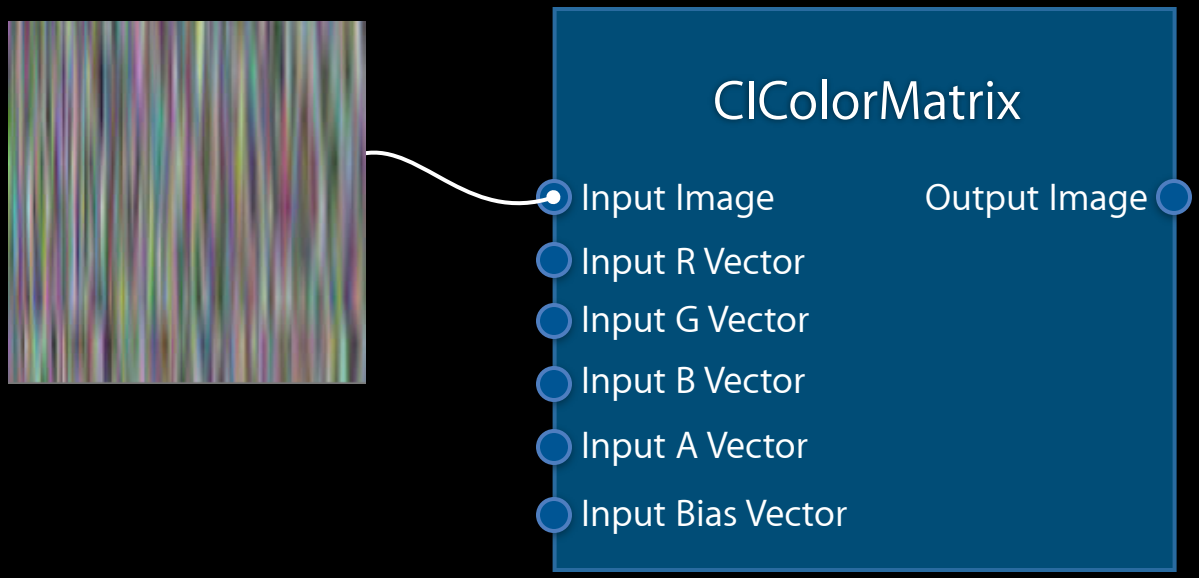
Add Dark Scratches



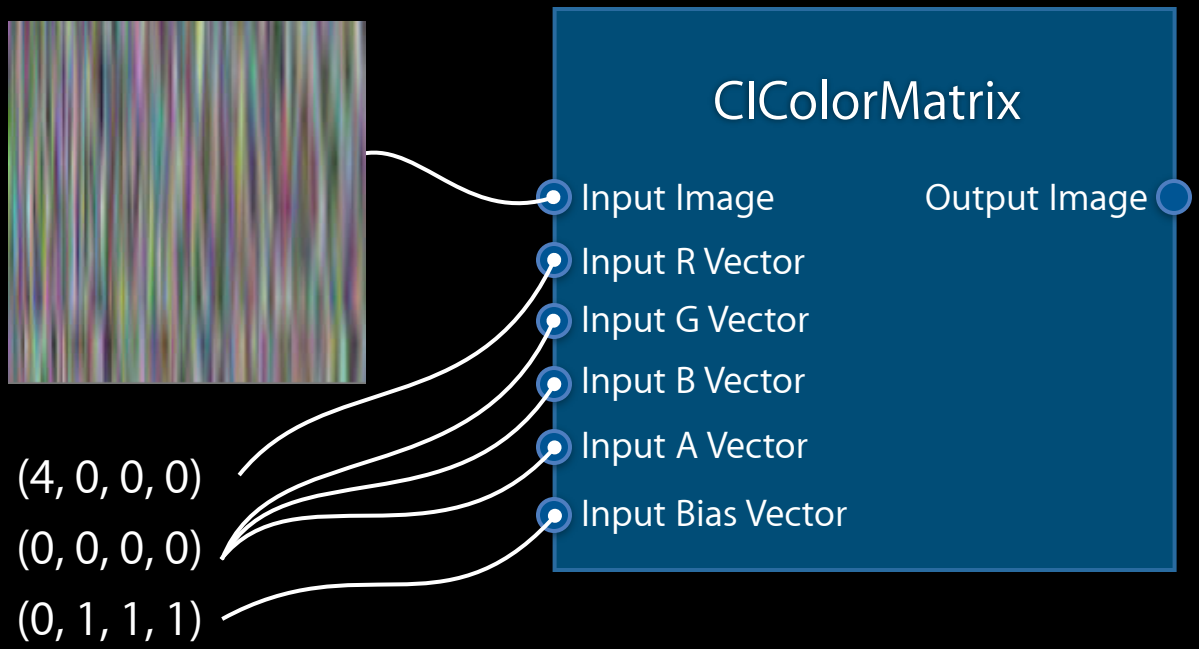
Add Dark Scratches



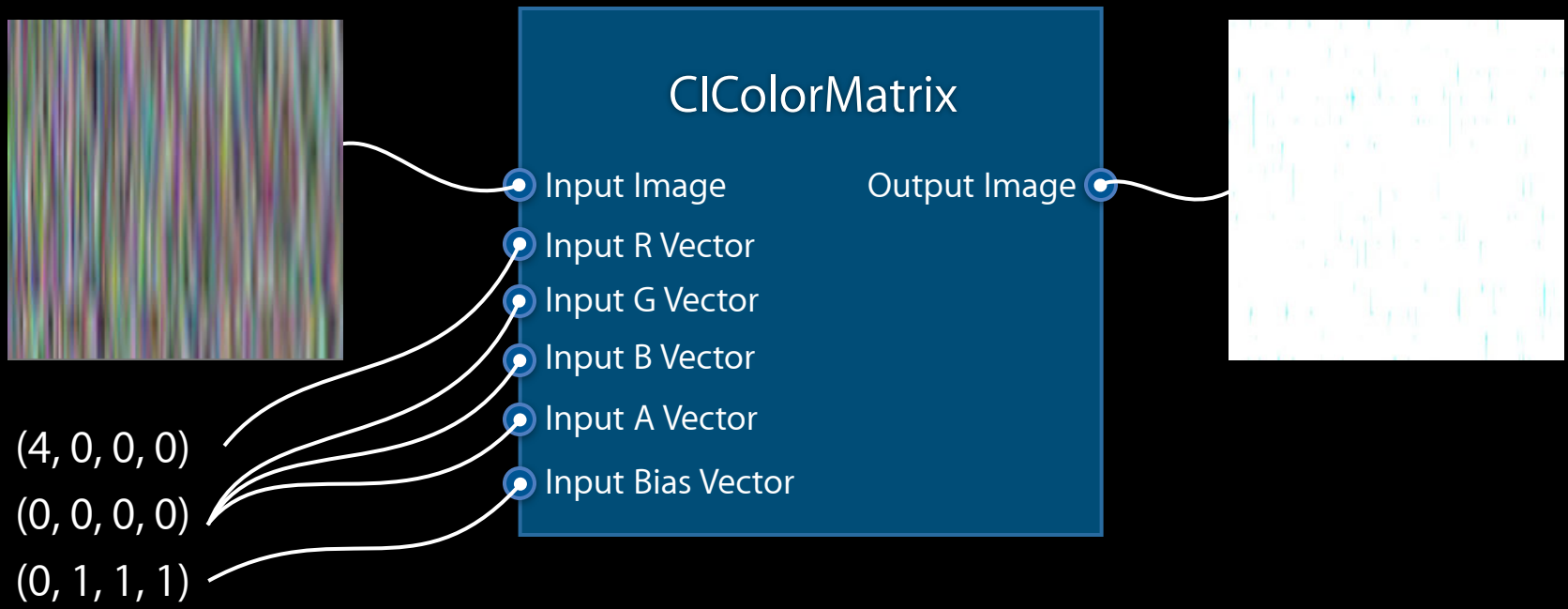
Add Dark Scratches



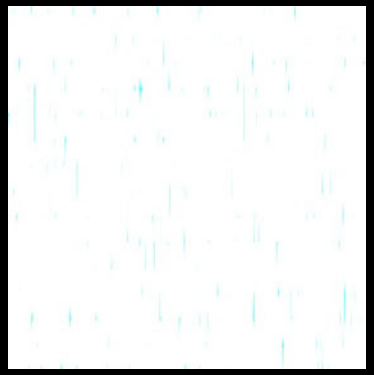
Add Dark Scratches



Add Dark Scratches



Add Dark Scratches

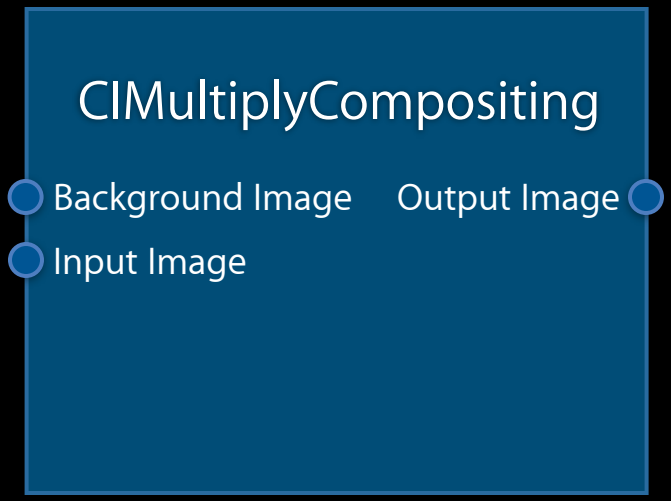


Add Dark Scratches

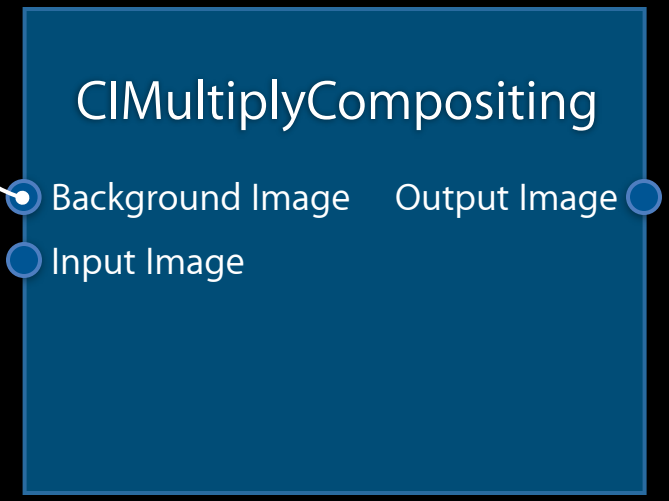
Add Dark Scratches

Add Dark Scratches

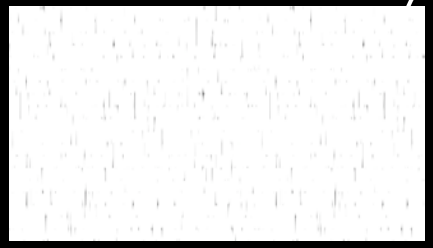
Final Compositing

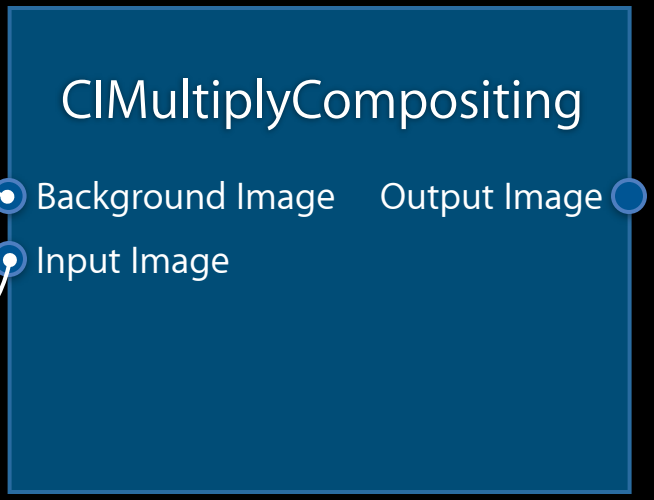


Final Compositing

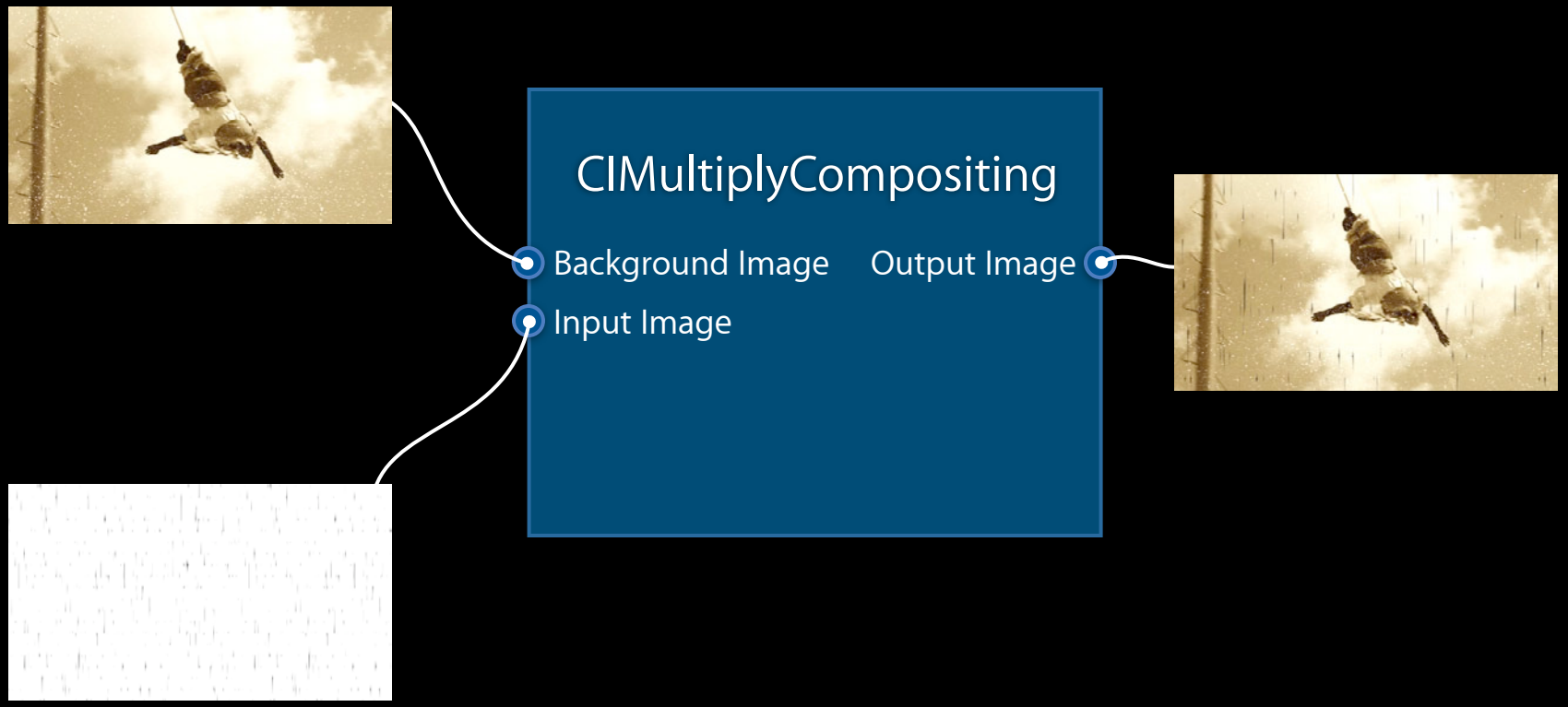


Final Compositing





Final Compositing



More Information

Allan Schaffer
Graphics and Imaging Evangelist
aschaffer@apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Core Image Techniques	Pacific Heights Wednesday 11:30AM

Labs

Core Image	Graphics, Media & Games Lab A Wednesday 2:00PM


WWDC2012