

#WWDC18

What's New in ARKit 2

Session 602

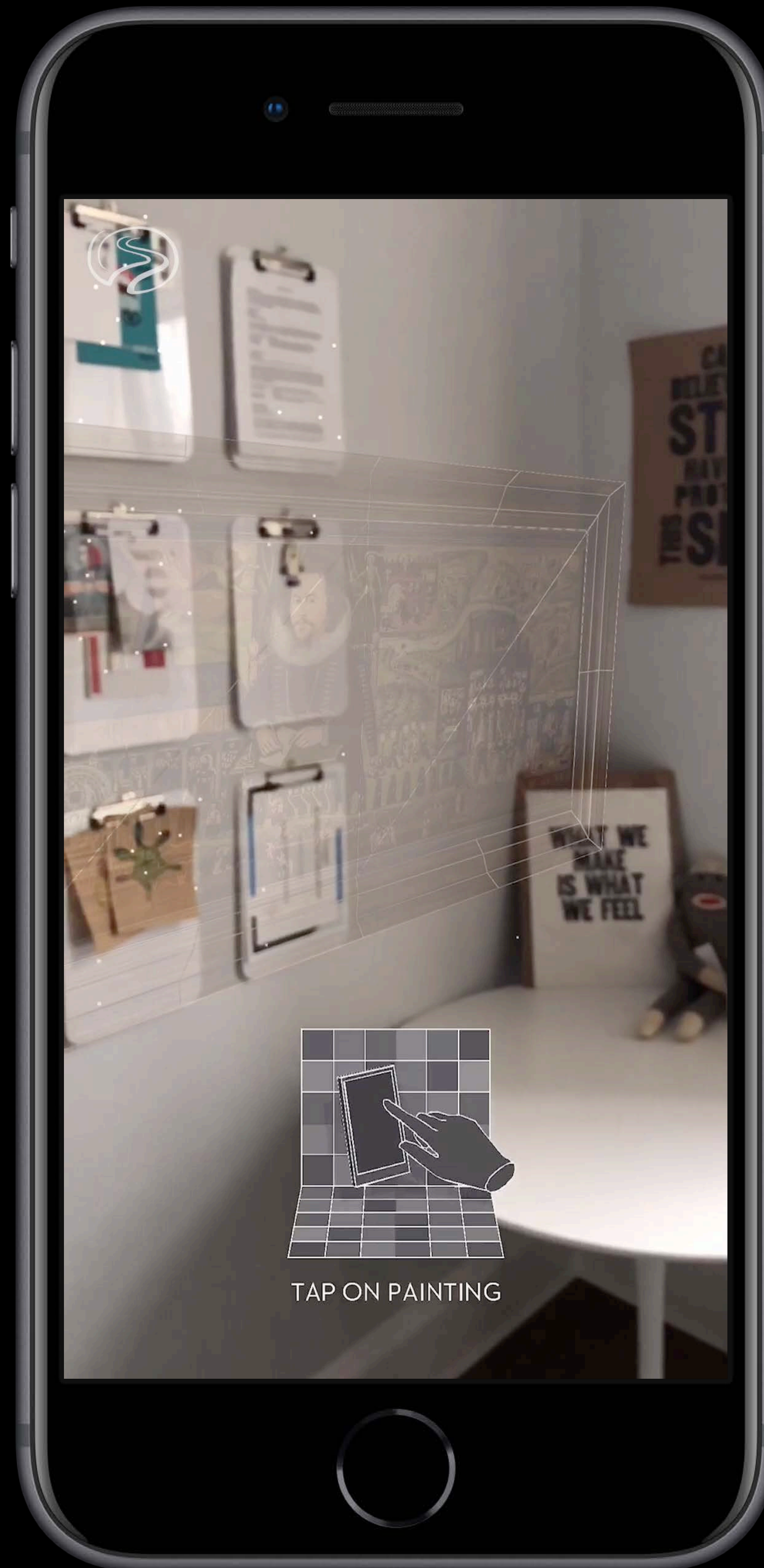
Arsalan Malik, ARKit Engineer
Reinhard Klapfer, ARKit Engineer

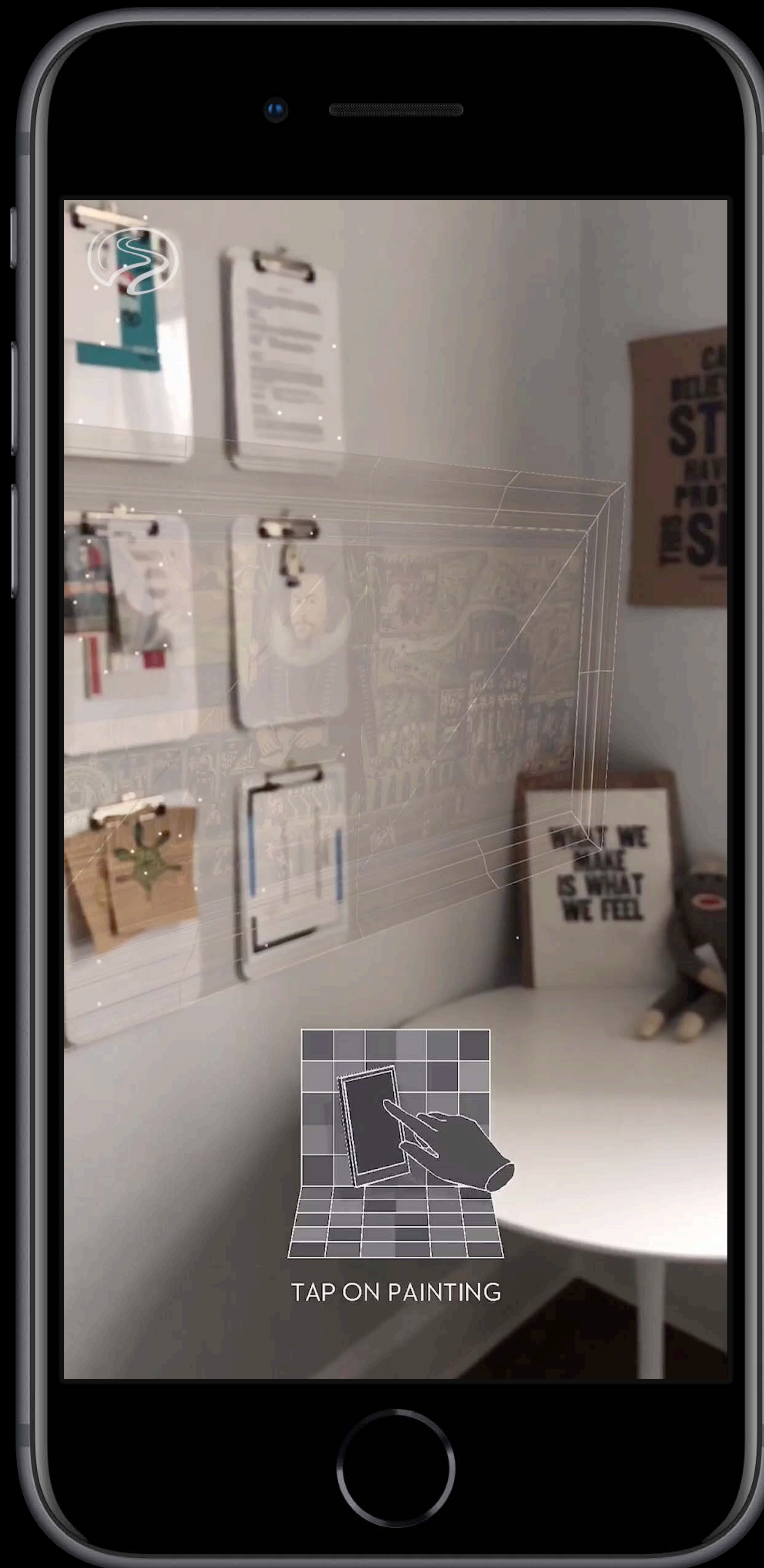


ARKit

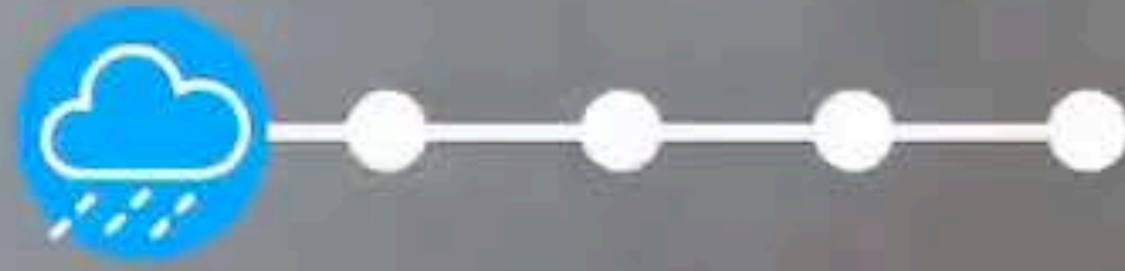








TAP ON PAINTING



FOLLOW WATER'S JOURNEY

Ocean water evaporates into clouds, which fall as rain. As rainwater flows over the landscape, it collects



0/9



FOLLOW WATER'S JOURNEY

Ocean water evaporates into clouds, which fall as rain. As rainwater flows over the landscape, it collects



ARKit Recap



Tracking



Tracking



Scene Understanding



Tracking



Scene Understanding



Rendering



What's New in ARKit 2

Agenda



NEW

What's New in ARKit 2

Agenda



NEW

Saving and Loading Maps

What's New in ARKit 2

Agenda



NEW

Saving and Loading Maps

Environment Texturing

What's New in ARKit 2

Agenda



NEW

Saving and Loading Maps

Environment Texturing

Image Tracking

What's New in ARKit 2

Agenda



NEW

Saving and Loading Maps

Environment Texturing

Image Tracking

Object Detection

What's New in ARKit 2

Agenda



NEW

Saving and Loading Maps

Environment Texturing

Image Tracking

Object Detection

Face Tracking Enhancements

Saving and Loading Maps

World Tracking

Recap



World Tracking

Recap

Position and orientation



World Tracking

Recap

Position and orientation

Physical scale



World Tracking

Recap

Position and orientation

Physical scale

3D feature points



World Tracking

Recap

Position and orientation

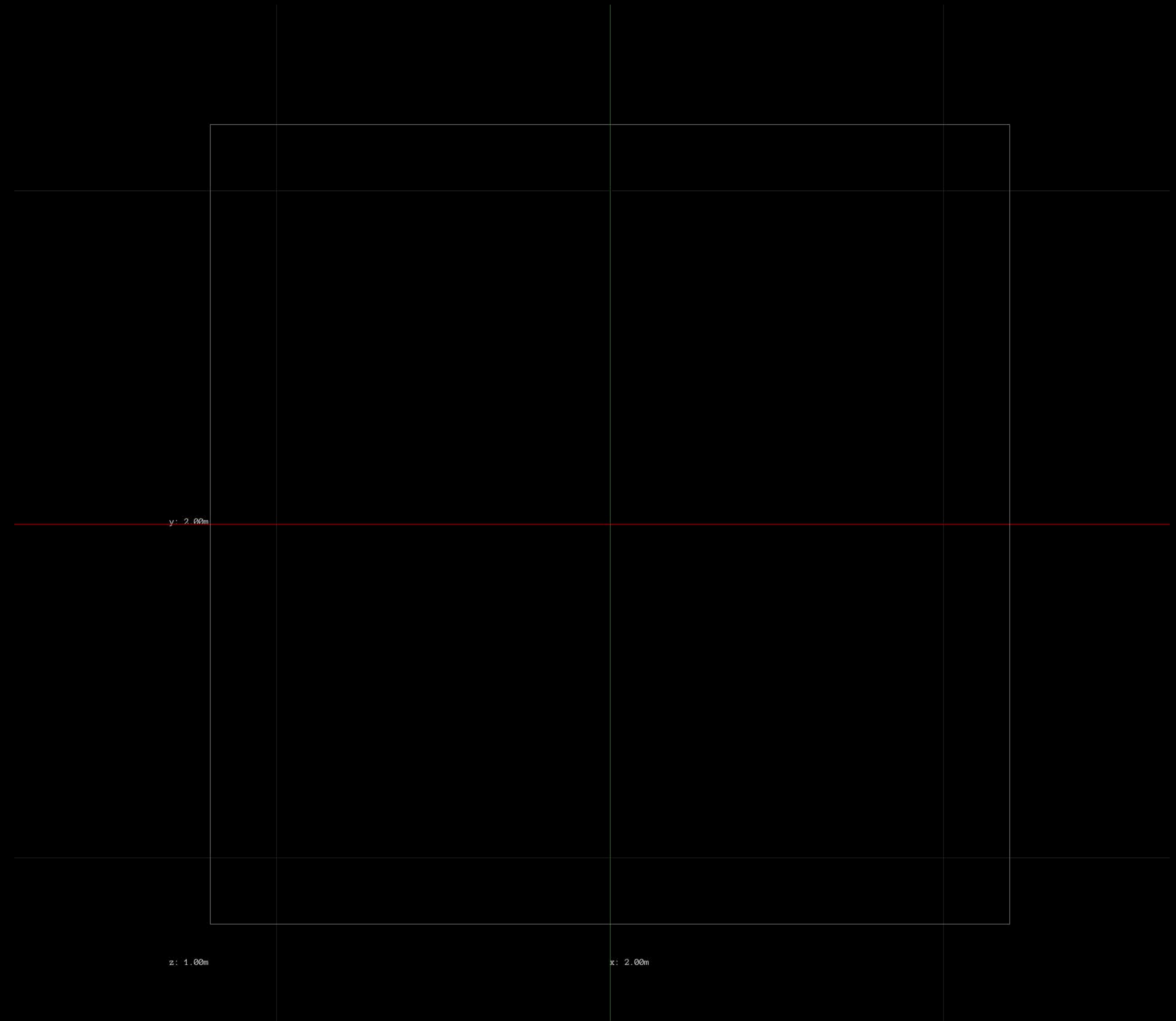
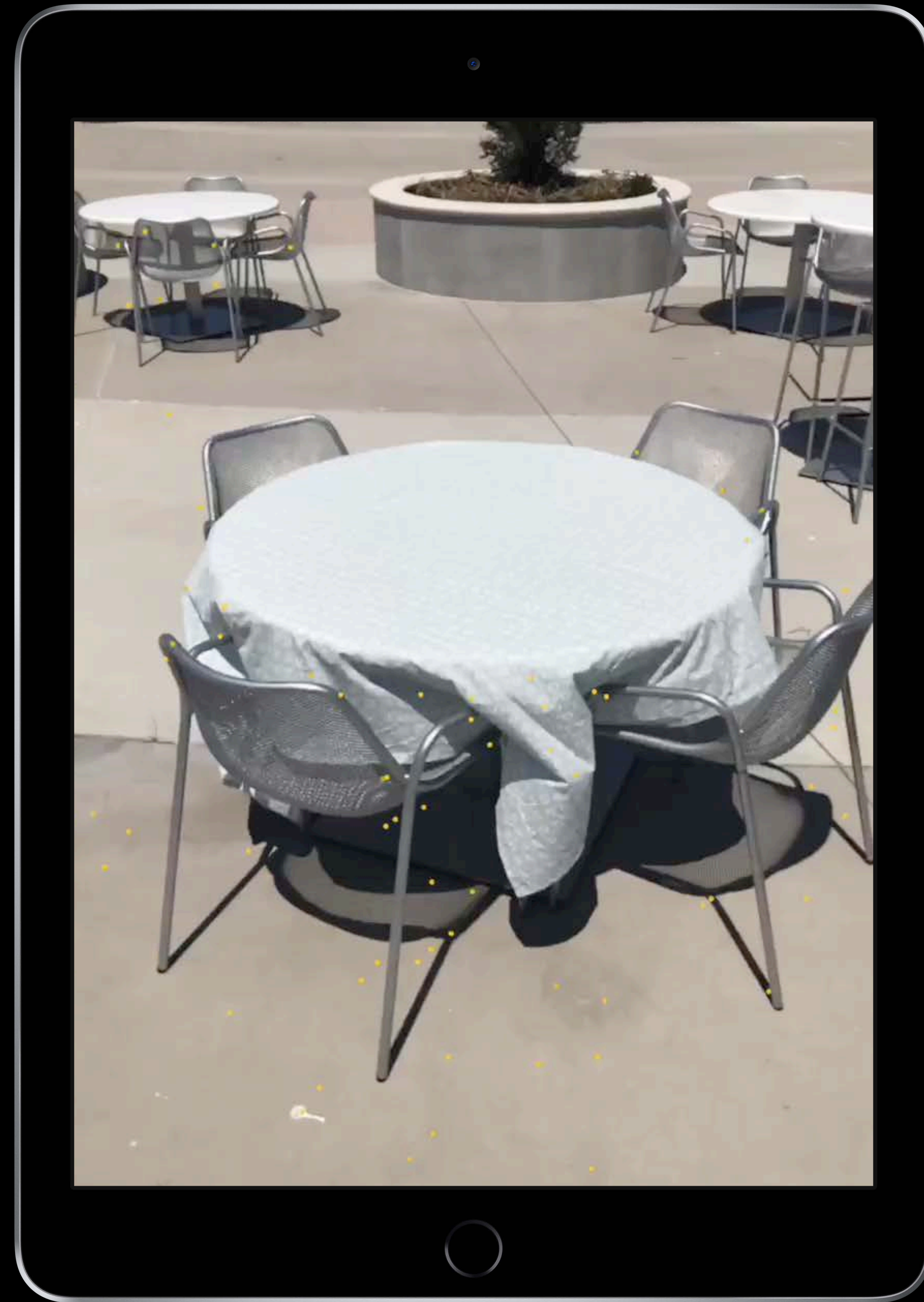
Physical scale

3D feature points

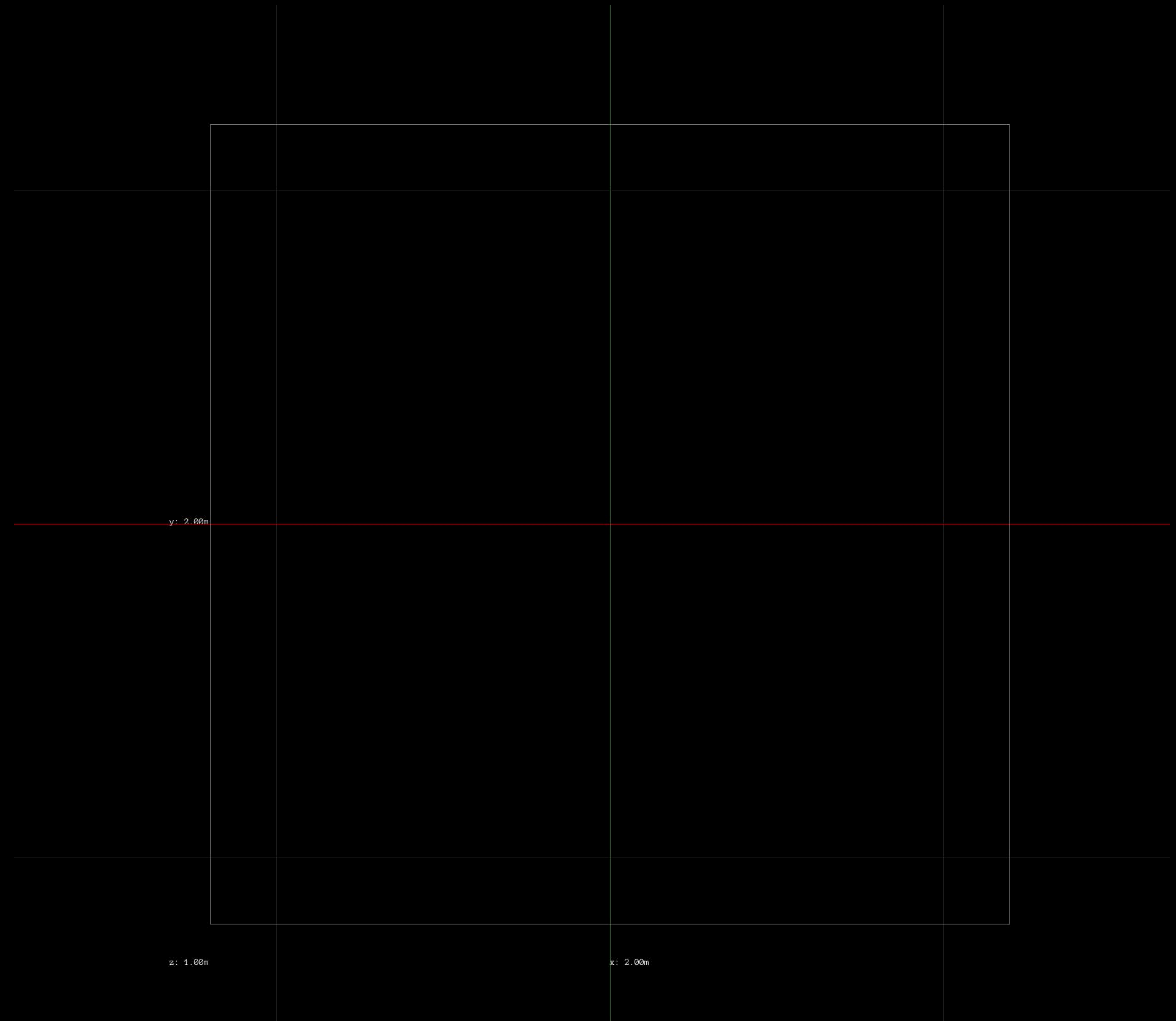
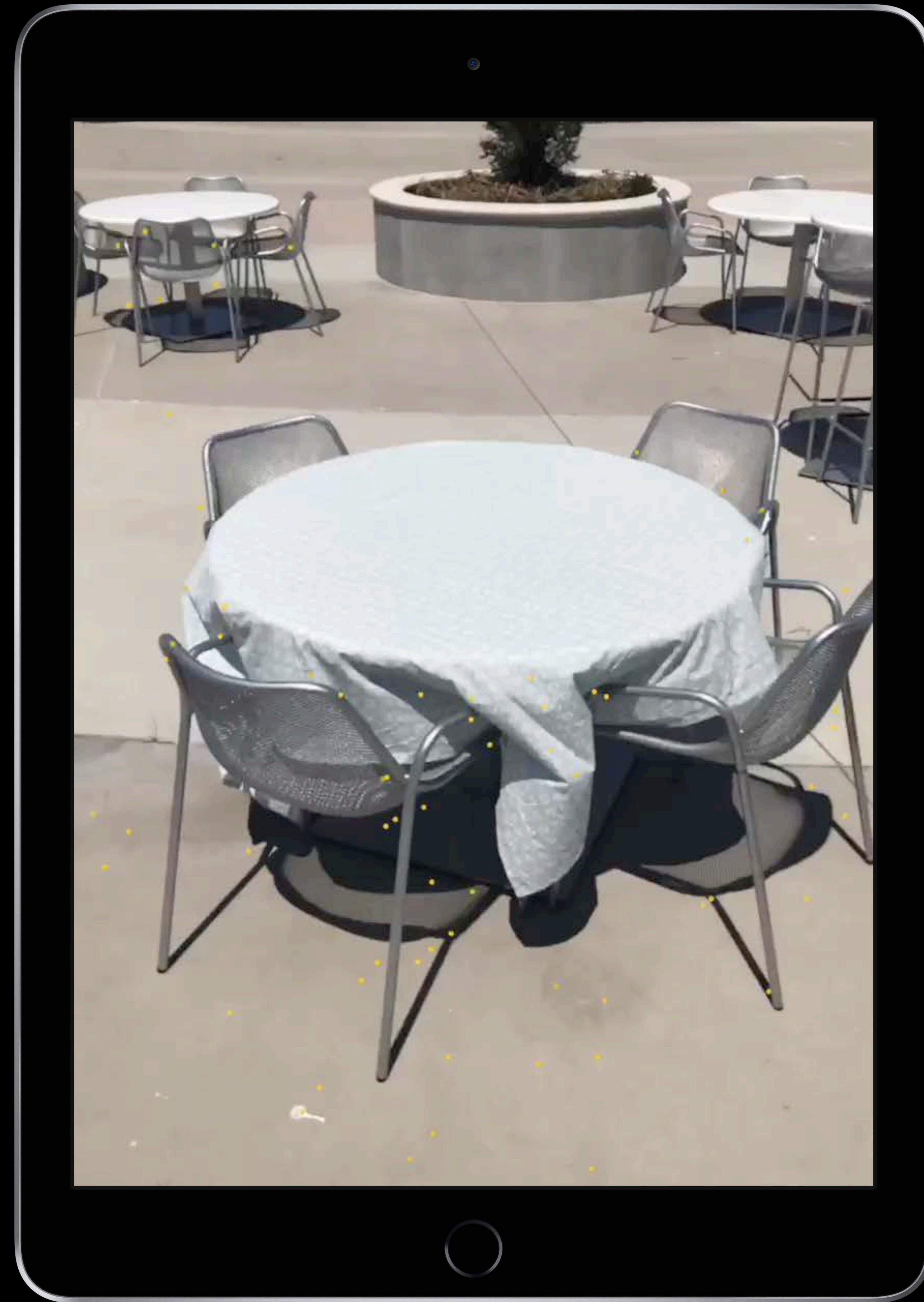
Relocalization



Mapping

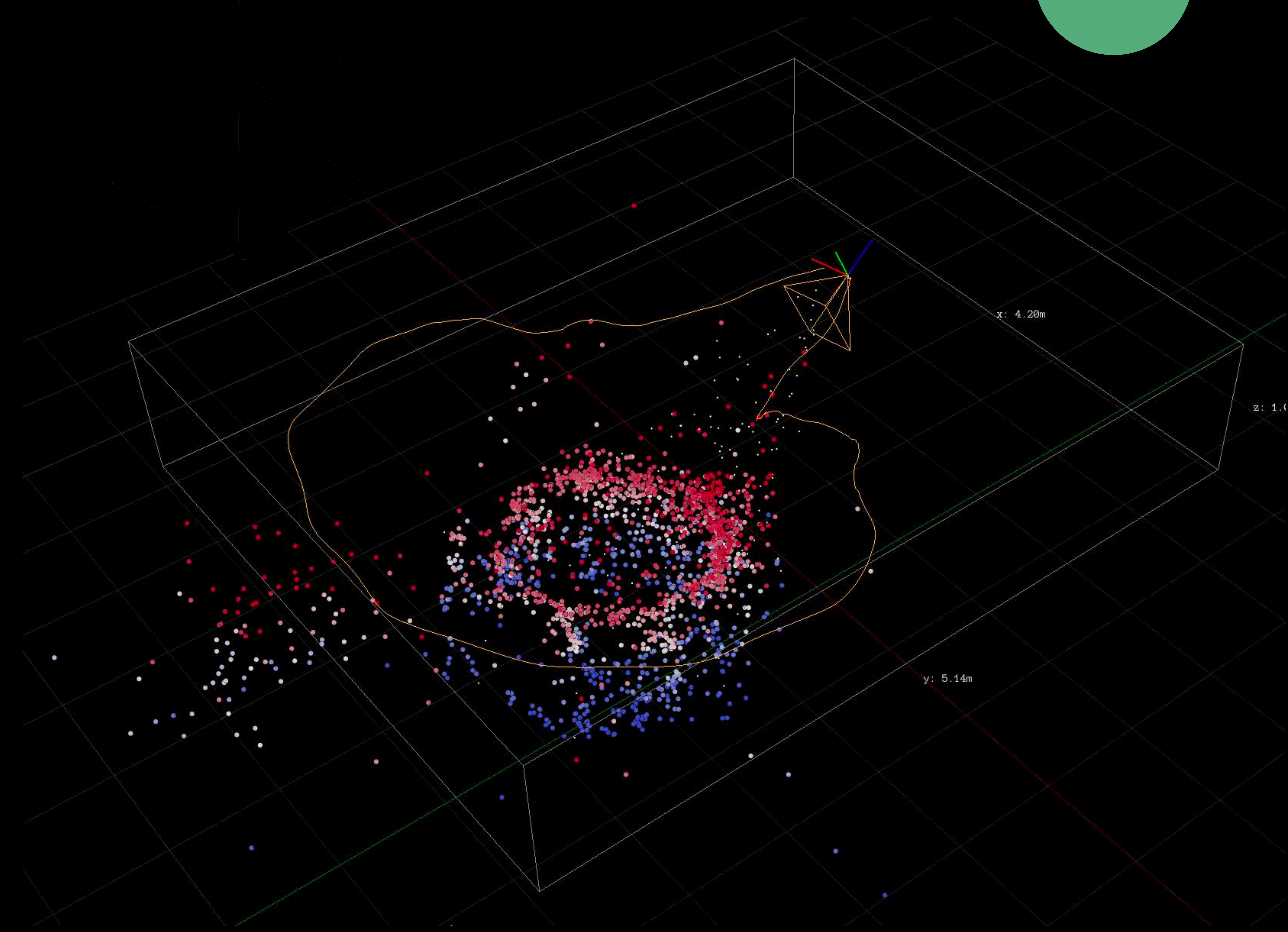


Mapping



ARWorldMap

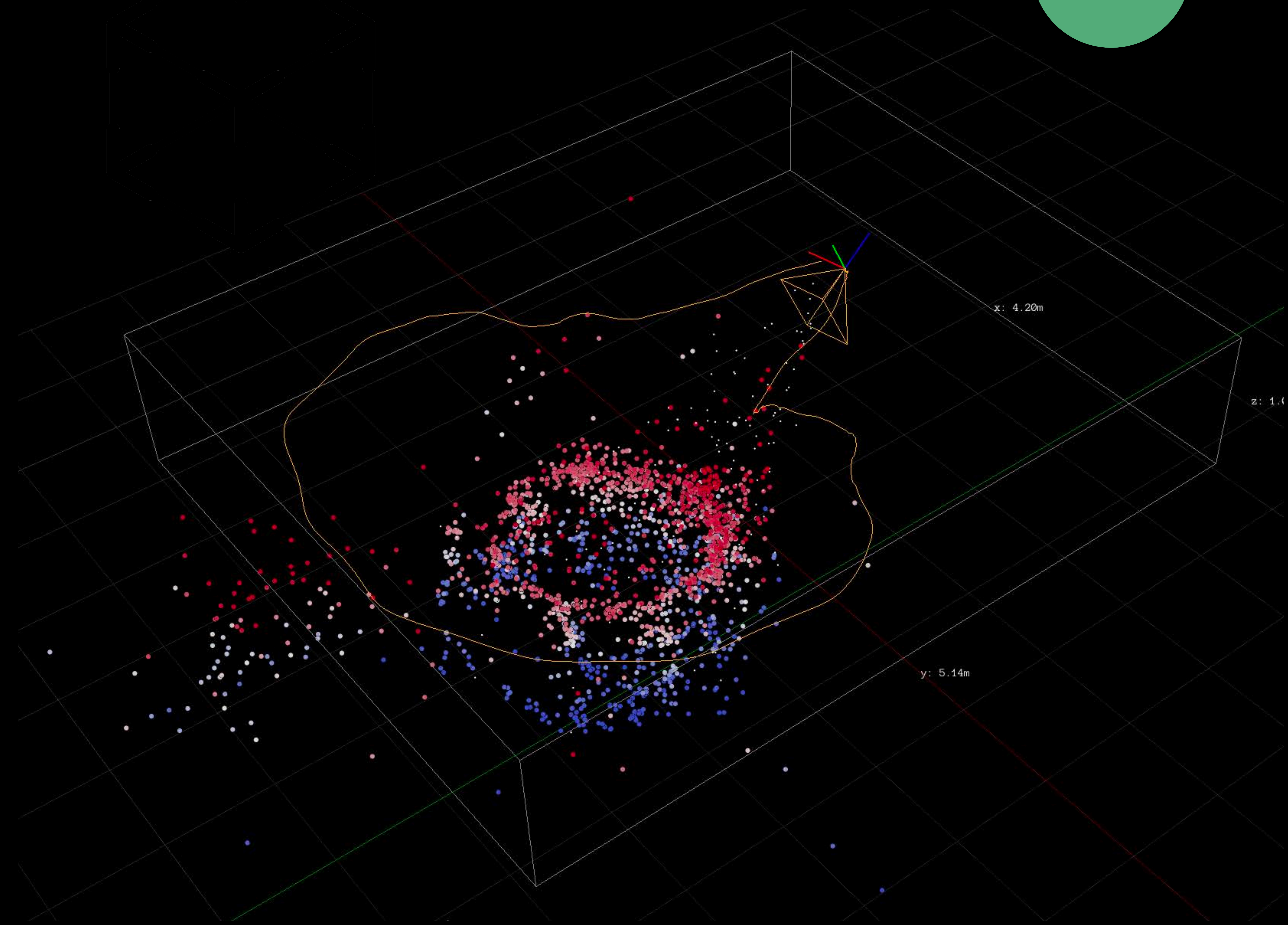
NEW



ARWorldMap

NEW

Mapping of physical 3D space



ARWorldMap

NEW

Mapping of physical 3D space

Mutable list of named anchors



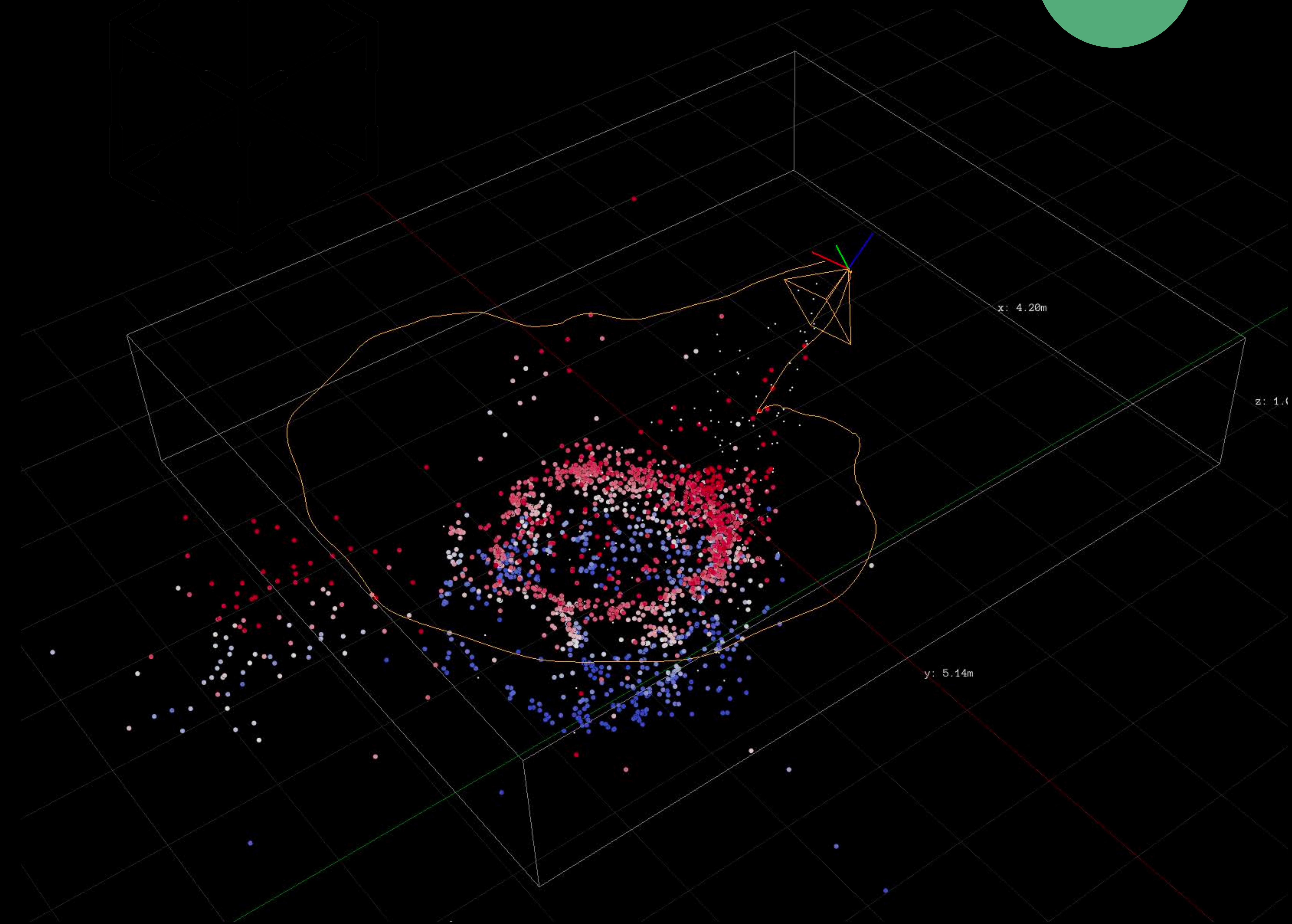
ARWorldMap

NEW

Mapping of physical 3D space

Mutable list of named anchors

Raw feature points and extent



ARWorldMap

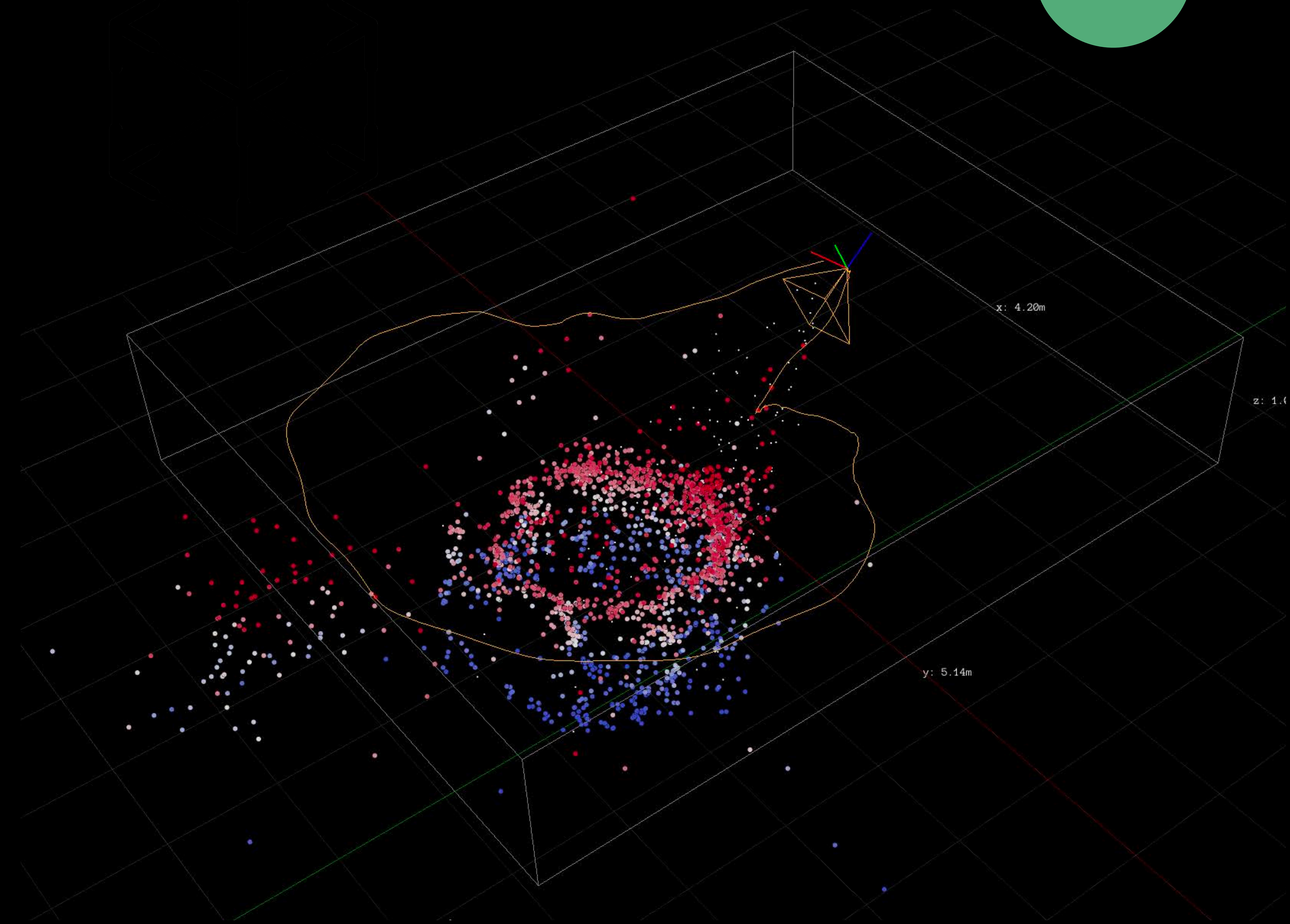
NEW

Mapping of physical 3D space

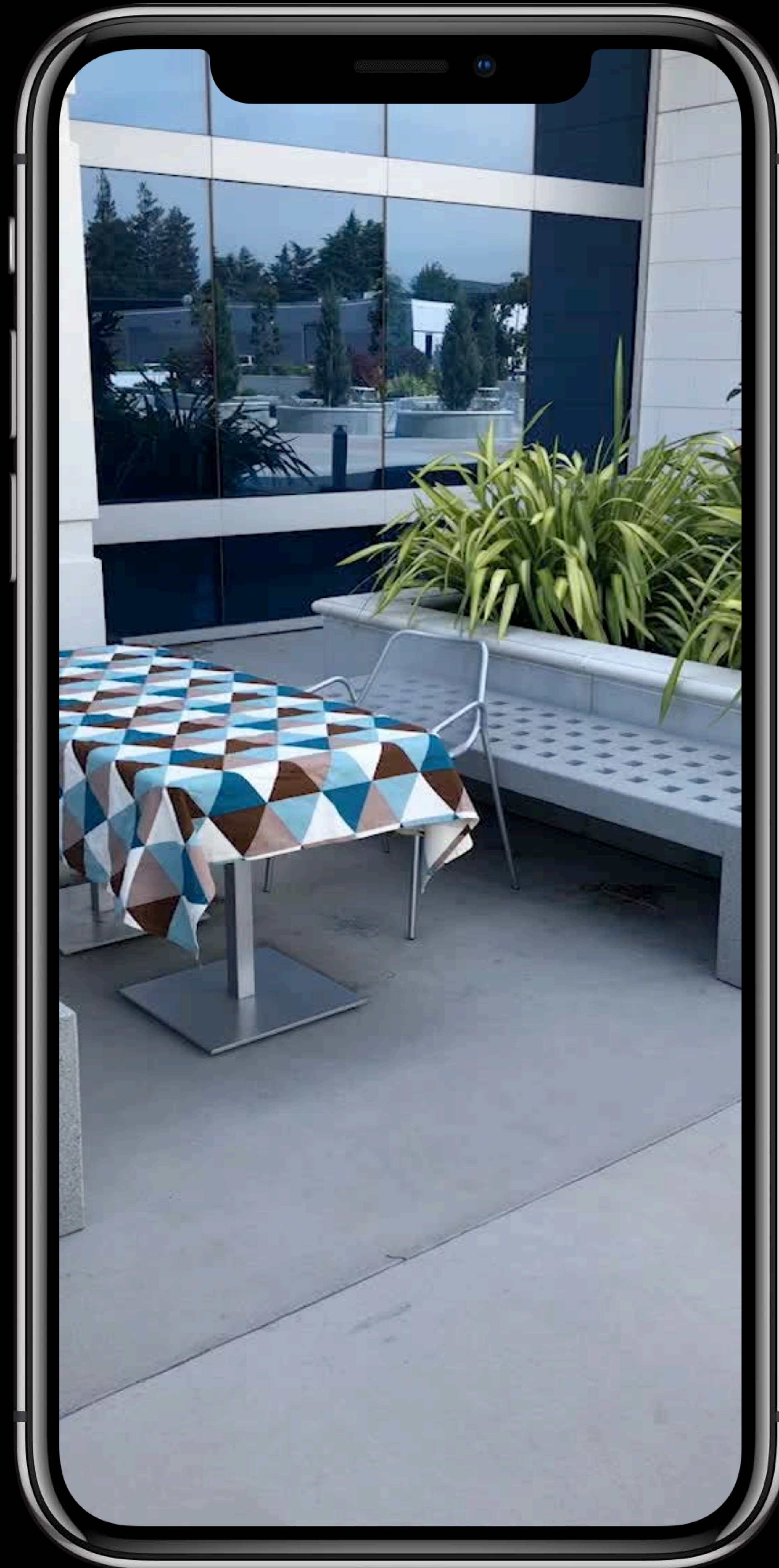
Mutable list of named anchors

Raw feature points and extent

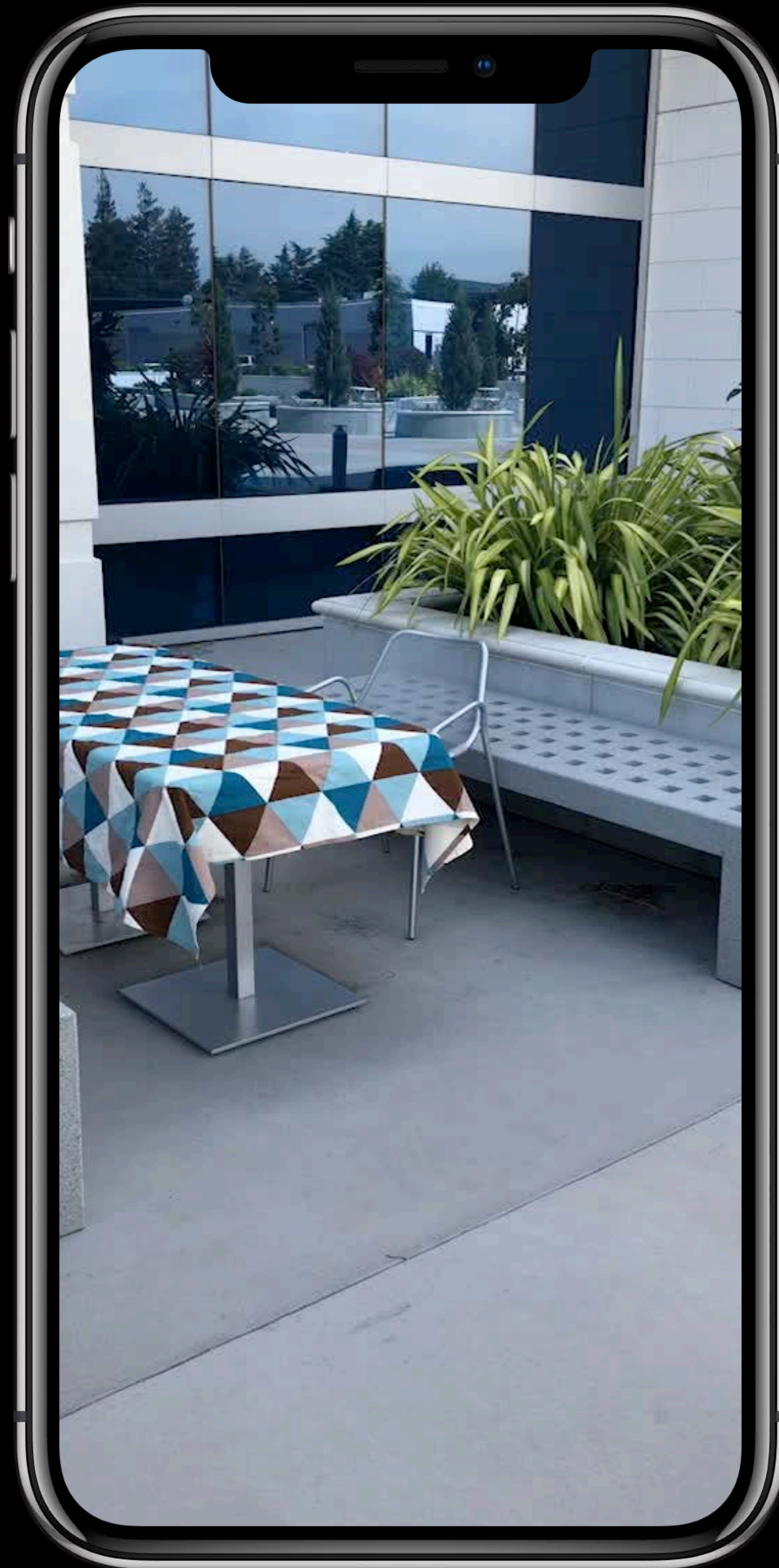
Serialization



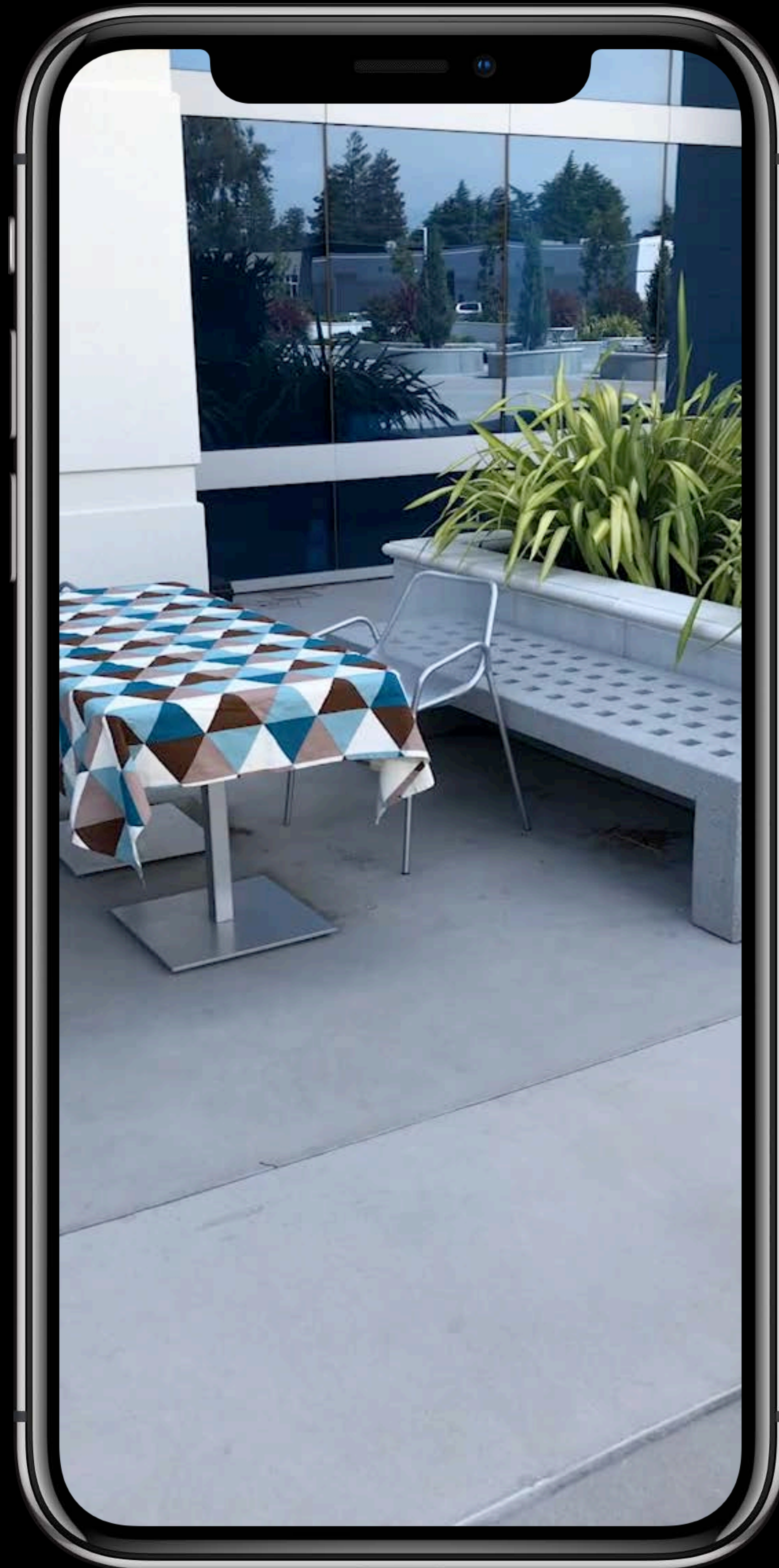
Persistence



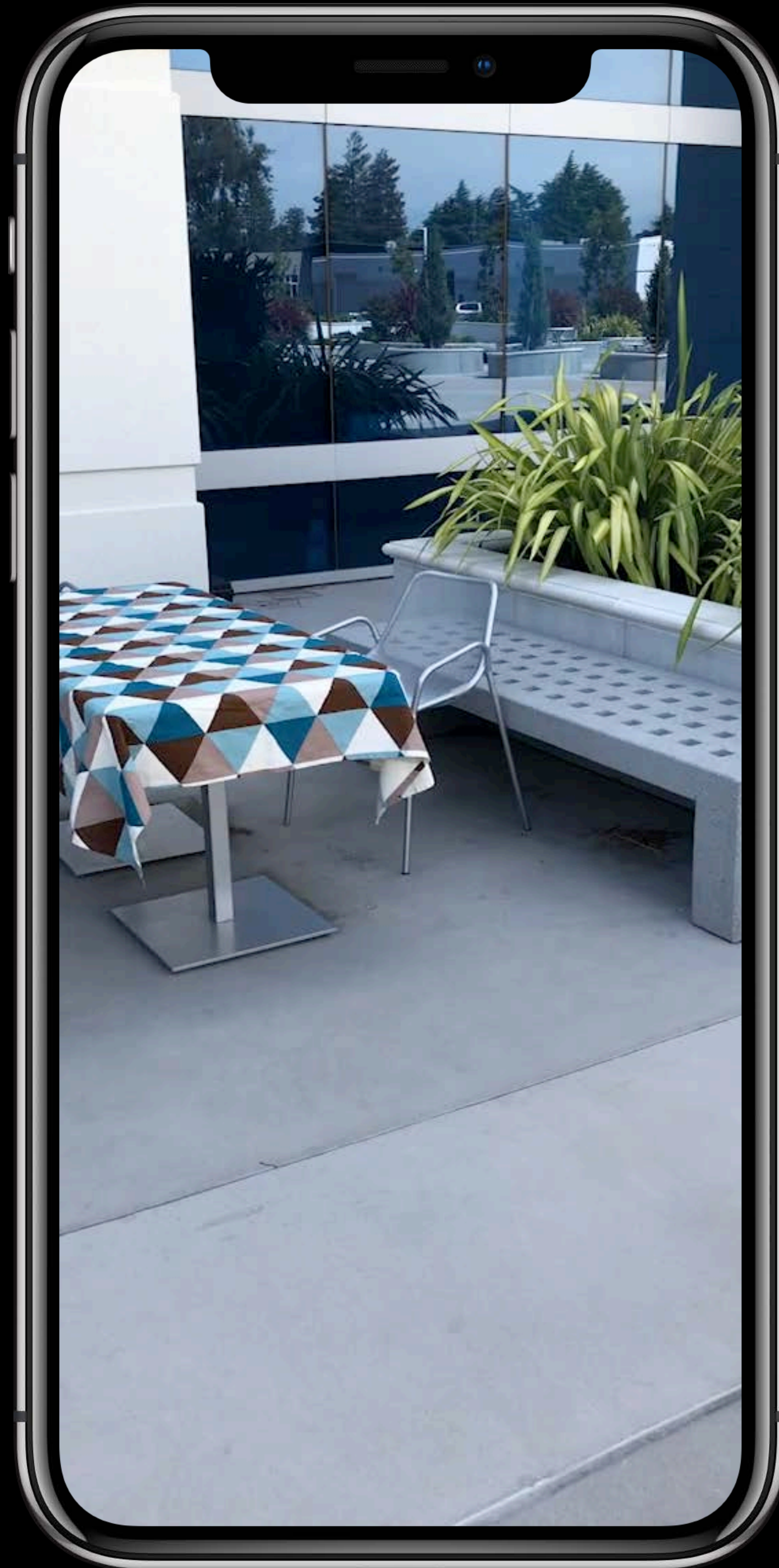
Persistence



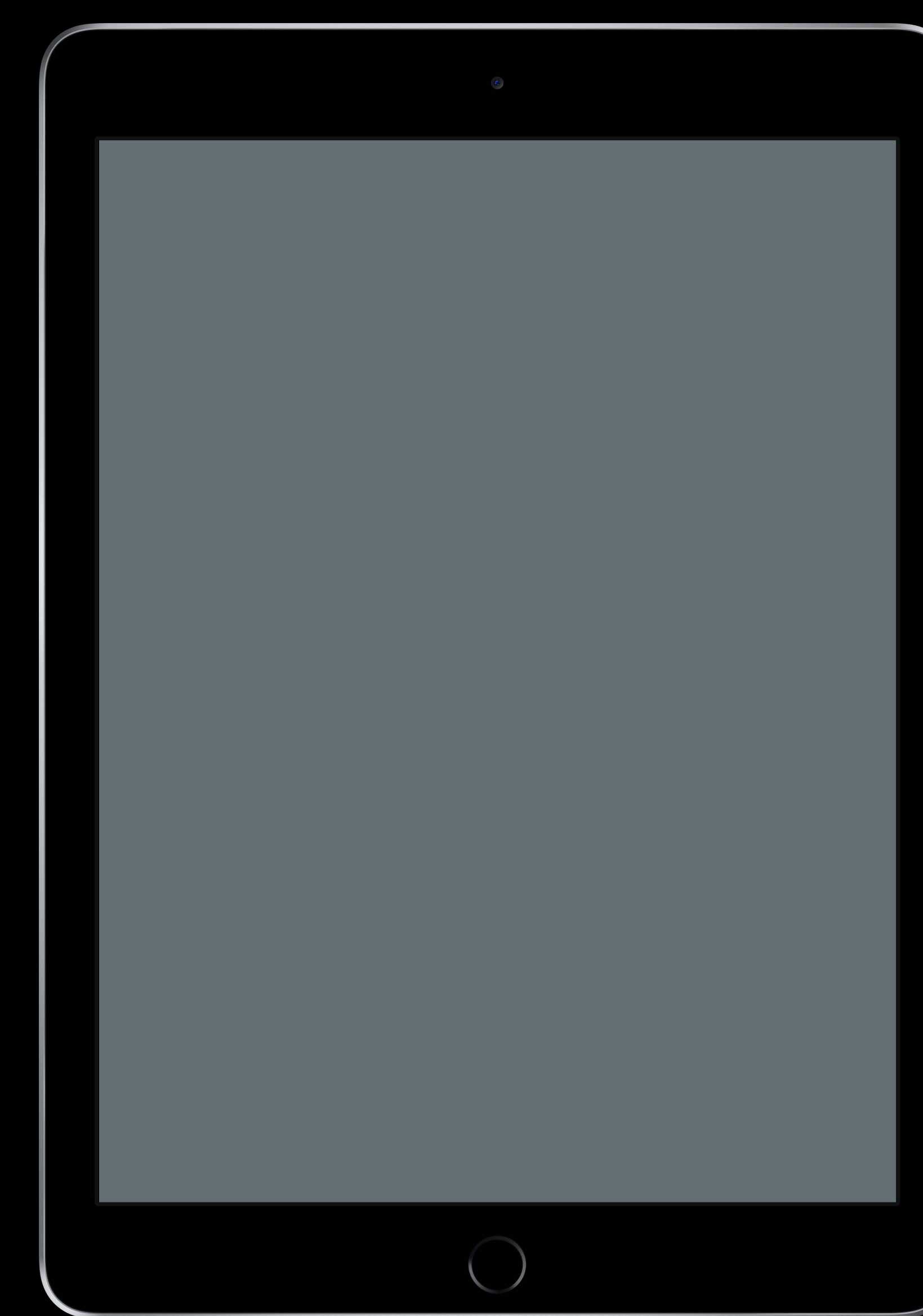
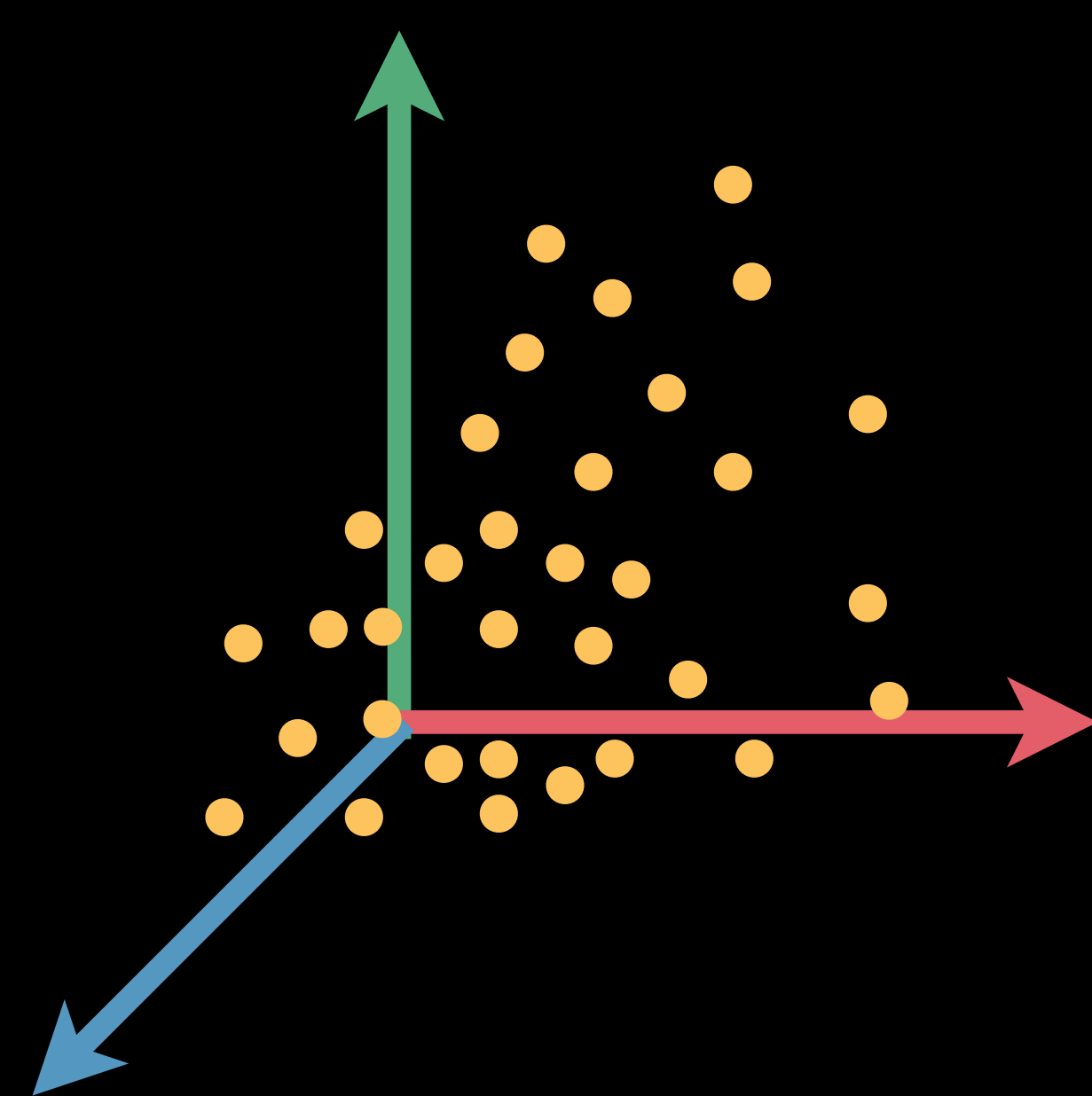
Persistence



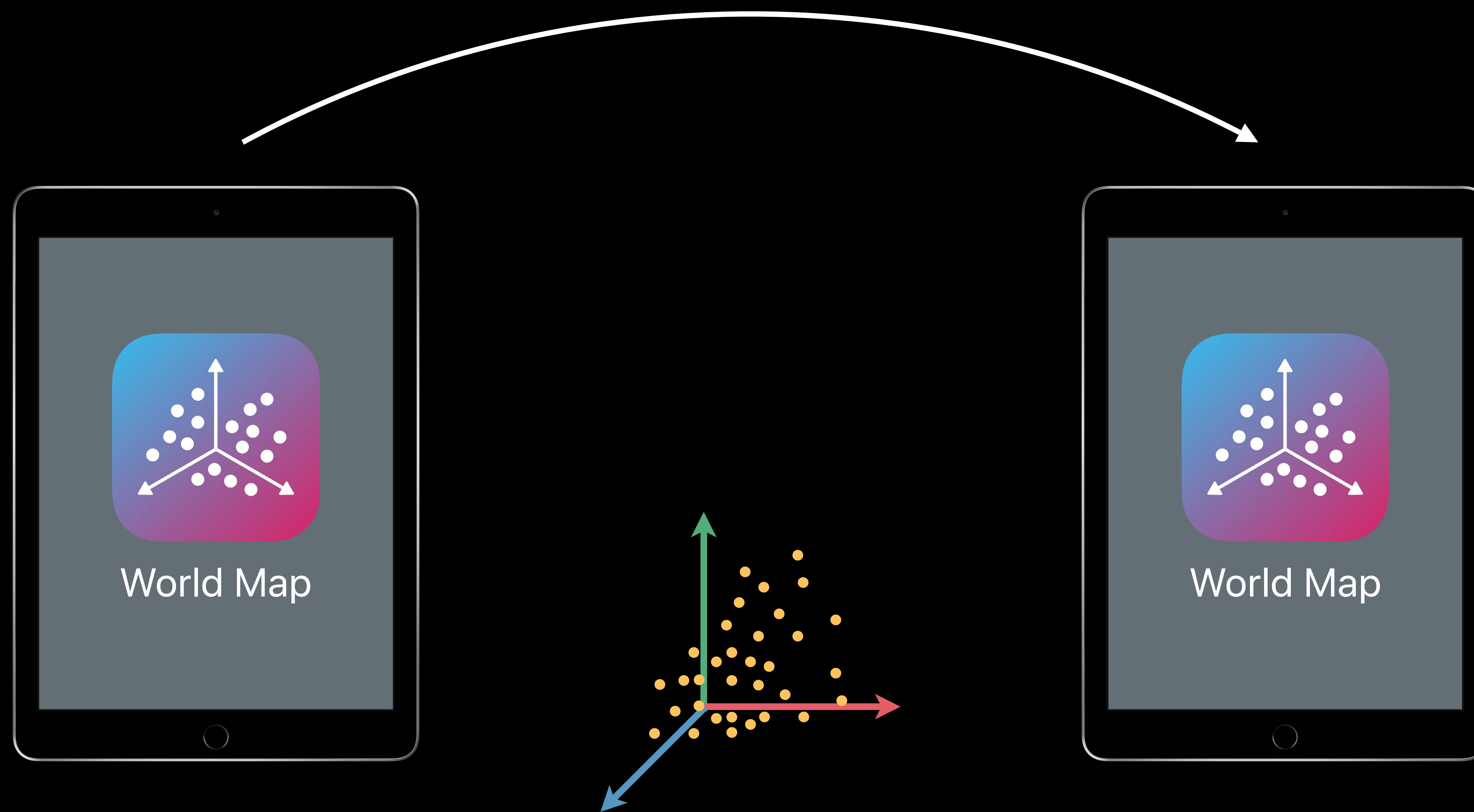
Persistence



Multi-User Experiences



Multi-User Experiences



```
// Saving and Loading World Maps

// Retrieve world map from session object
session.getCurrentWorldMap { worldMap, error in
    guard let worldMap = worldMap else {
        showAlert(error)
        return
    }
}
```

```
// Saving and Loading World Maps
```

```
// Retrieve world map from session object
```

```
session.getCurrentWorldMap { worldMap, error in
```

```
    guard let worldMap = worldMap else {
```

```
        showAlert(error)
```

```
        return
```

```
    }
```

```
}
```



```
// Saving and Loading World Maps

// Retrieve world map from session object
session.getCurrentWorldMap { worldMap, error in
    guard let worldMap = worldMap else {
        showAlert(error)
        return
    }
}
```

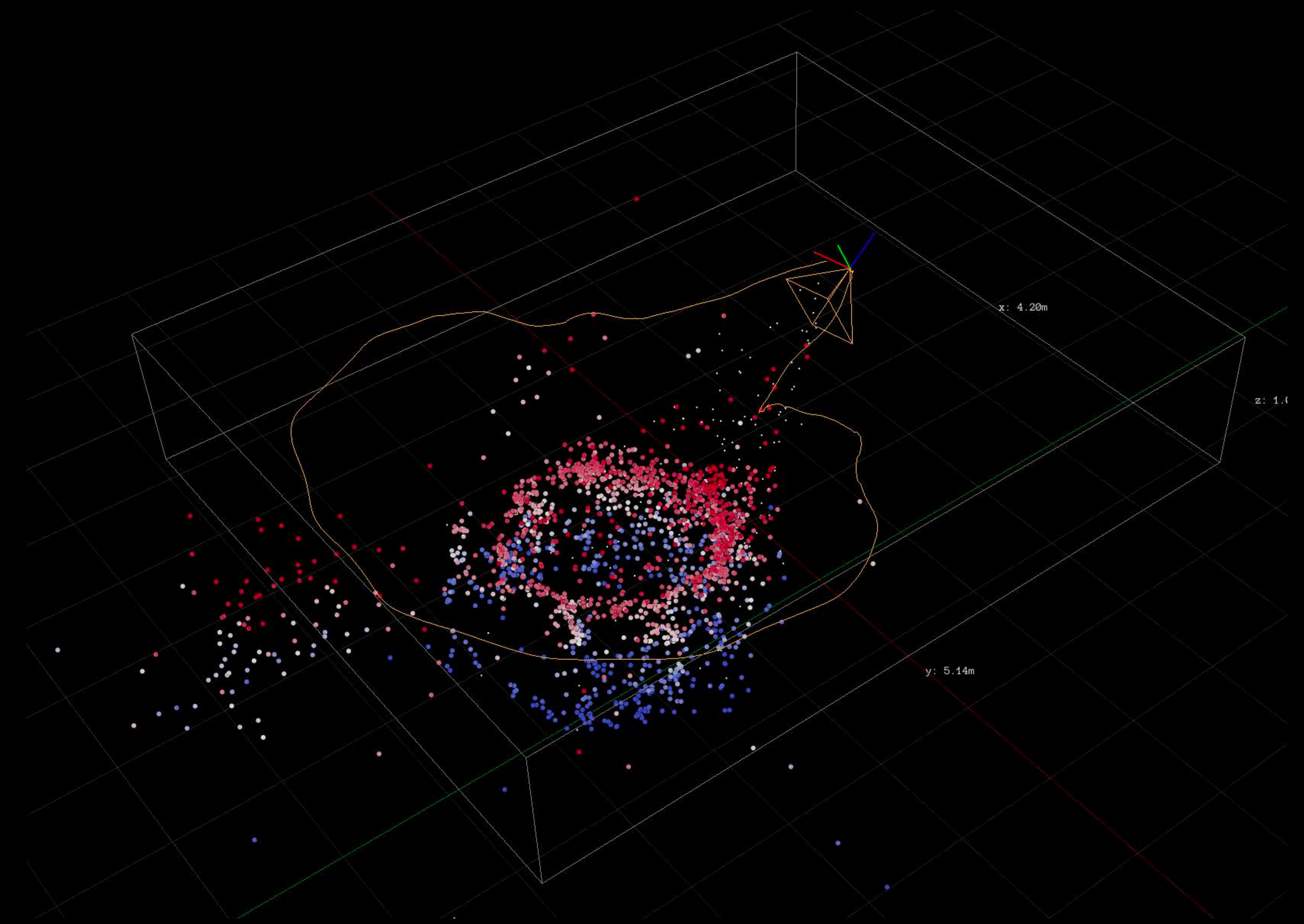
```
// Load world map and run the configuration
let configuration = ARWorldTrackingConfiguration()
configuration.initialWorldMap = worldMap
session.run(configuration)
```

Acquiring Good World Maps



Acquiring Good World Maps

Important for relocalization



Acquiring Good World Maps

Important for relocalization

Multiple points of view



Acquiring Good World Maps

Important for relocalization

Multiple points of view

Static, well-textured environment



Acquiring Good World Maps

Important for relocalization

Multiple points of view

Static, well-textured environment

Dense feature points on the map



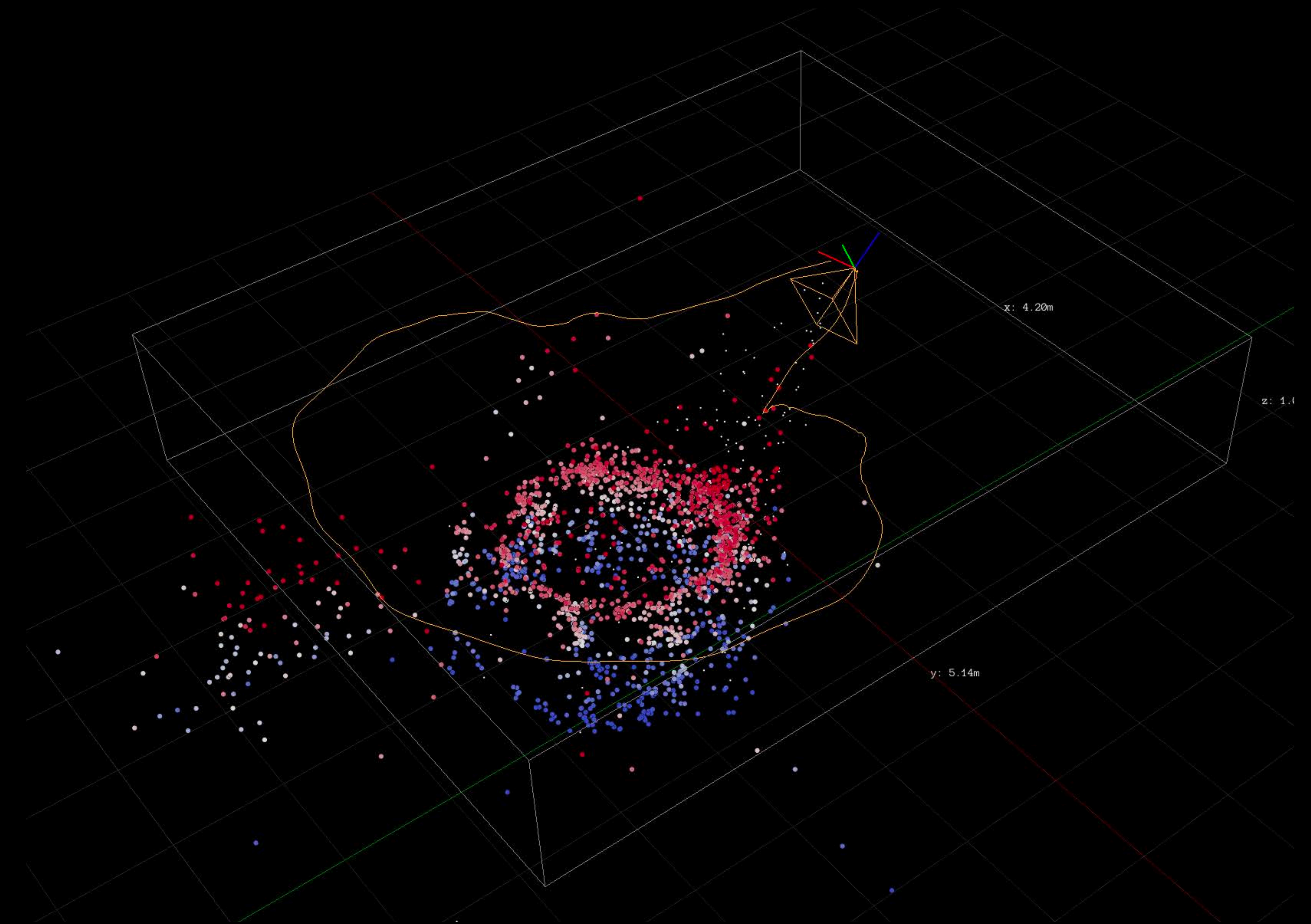
Acquiring Good World Maps

Important for relocalization

Multiple points of view

Static, well-textured environment

Dense feature points on the map

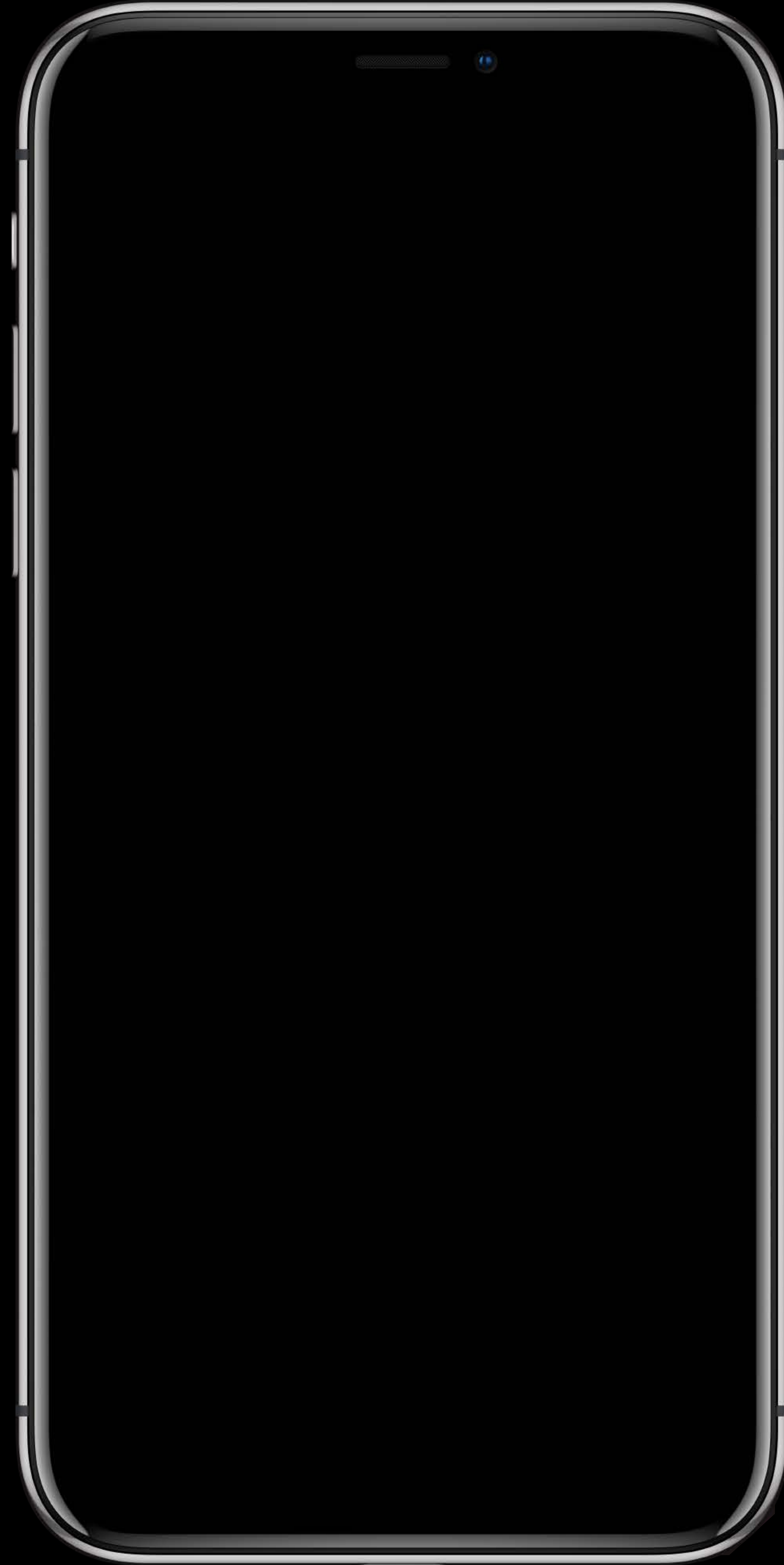


```
open class ARFrame : NSObject, NSCopying {
```

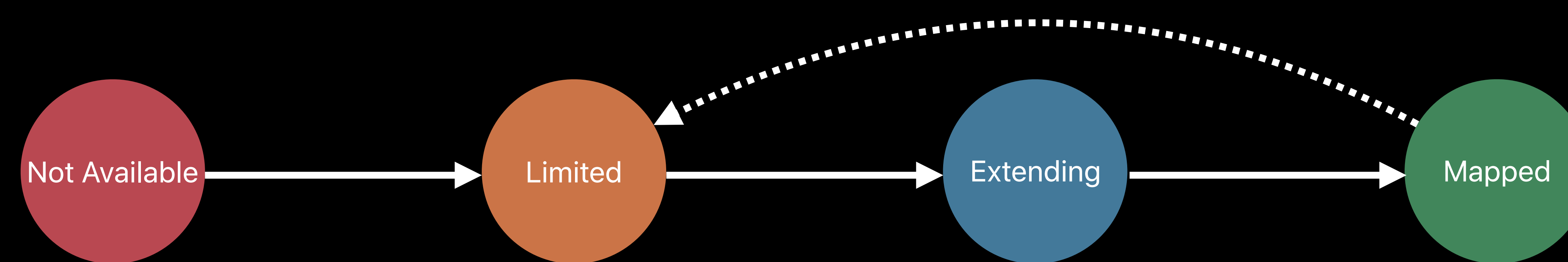
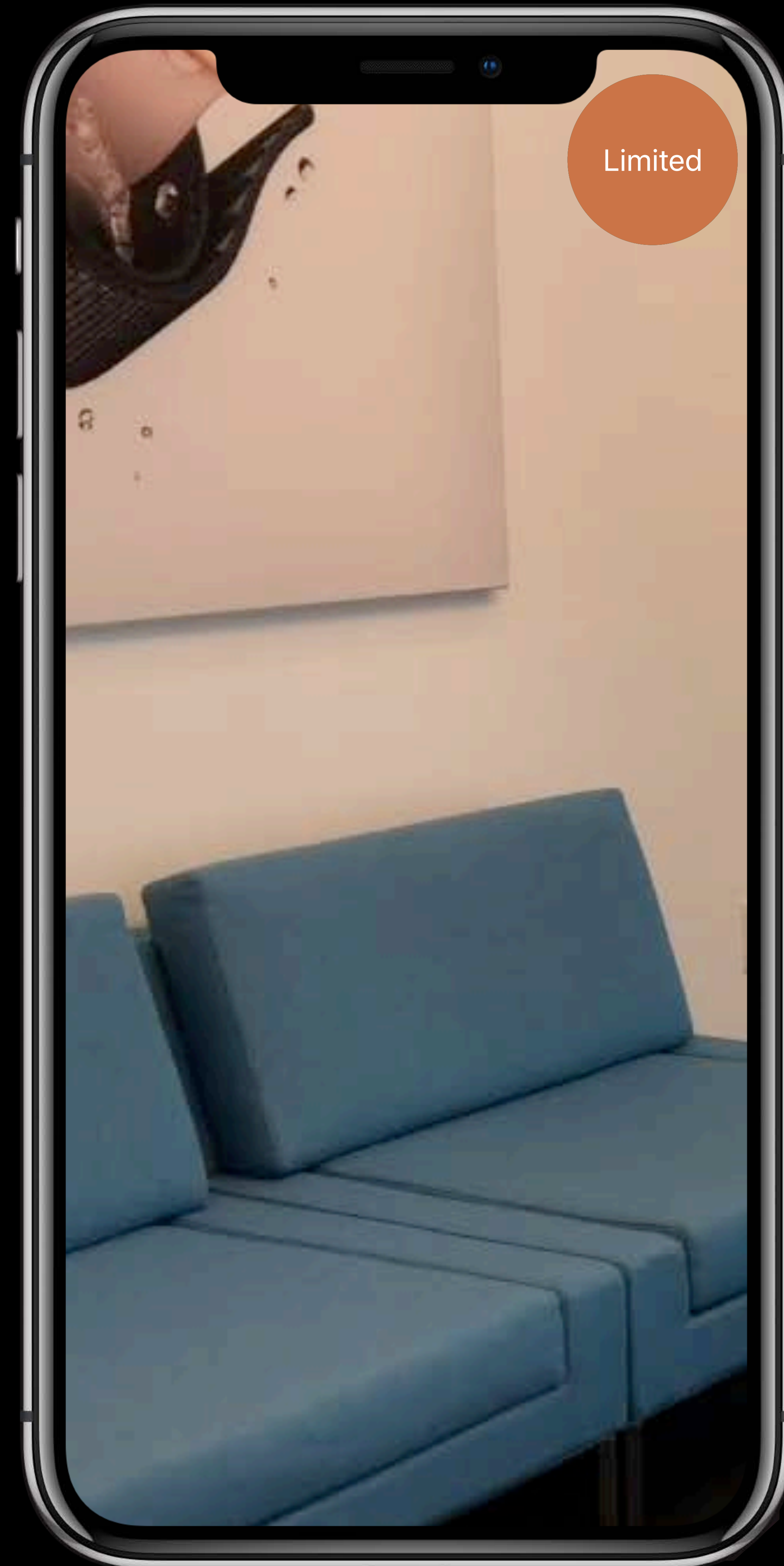
```
    open var worldMappingStatus: ARFrame.WorldMappingStatus
```

```
}
```

World Mapping Status



World Mapping Status



Using WorldMappingStatus

```
public enum WorldMappingStatus : Int {  
    case notAvailable  
  
    case limited  
  
    case extending  
  
    case mapped  
  
}
```



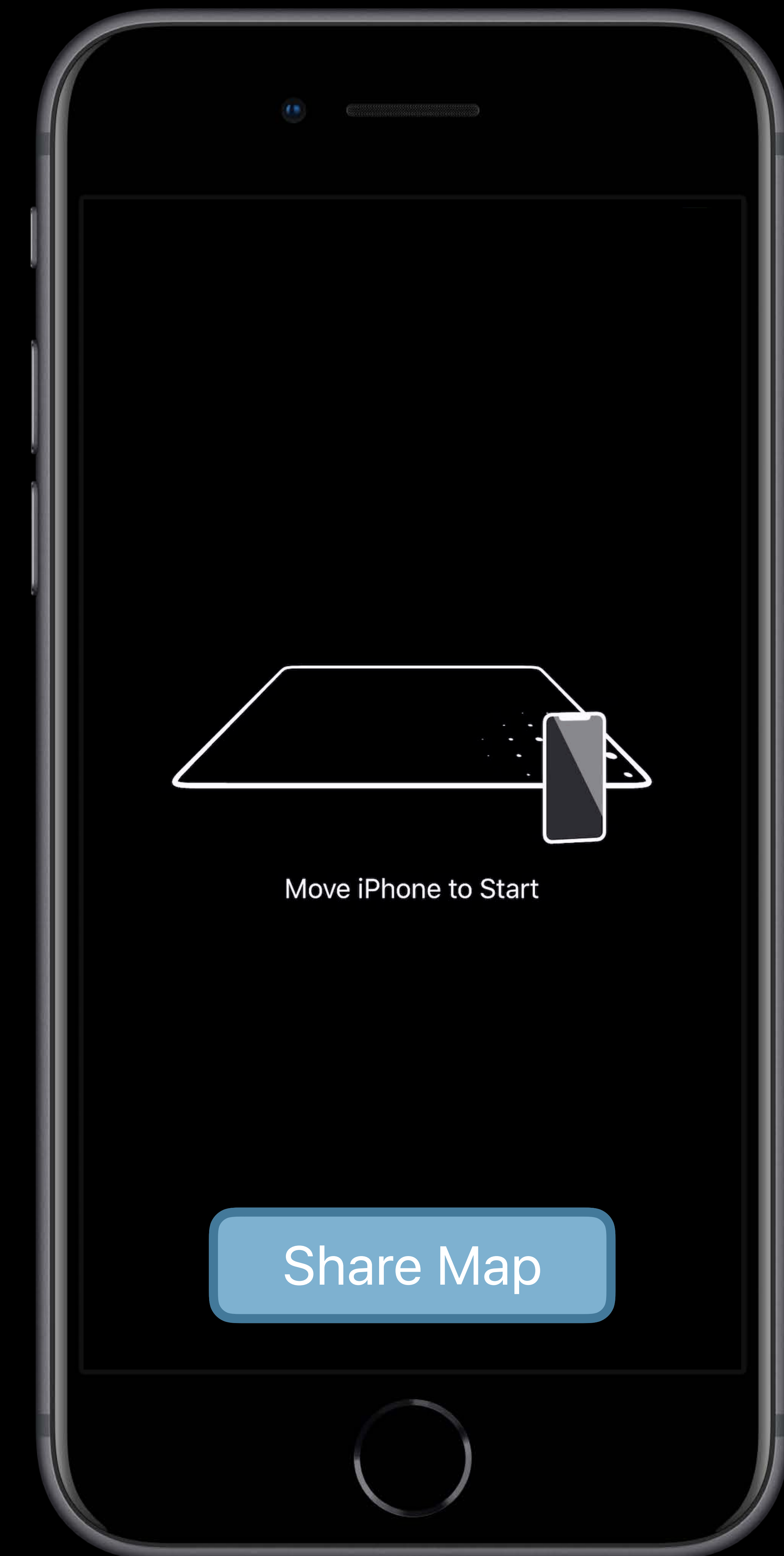
Using WorldMappingStatus

```
public enum WorldMappingStatus : Int {  
  
    case notAvailable  
  
    case limited  
  
    case extending  
  
    case mapped  
  
}
```



Using WorldMappingStatus

```
public enum WorldMappingStatus : Int {  
  
    case notAvailable  
  
    case limited  
  
    case extending  
  
    case mapped  
  
}
```



Using WorldMappingStatus

```
public enum WorldMappingStatus : Int {  
  
    case notAvailable  
  
    case limited  
  
    case extending  
  
    case mapped  
  
}
```



Demo

Saving and Loading Map

Arsalan Malik, ARKit Engineer

World Tracking Enhancements

NEW

World Tracking Enhancements

NEW

Saving and loading maps

World Tracking Enhancements



NEW

Saving and loading maps

Faster initialization and plane detection

World Tracking Enhancements



NEW

Saving and loading maps

Faster initialization and plane detection

Robust tracking and plane detection

World Tracking Enhancements



NEW

Saving and loading maps

Faster initialization and plane detection

Robust tracking and plane detection

More accurate extent and boundary

World Tracking Enhancements



NEW

Saving and loading maps

Faster initialization and plane detection

Robust tracking and plane detection

More accurate extent and boundary

Continuous autofocus

World Tracking Enhancements



NEW

Saving and loading maps

Faster initialization and plane detection

Robust tracking and plane detection

More accurate extent and boundary

Continuous autofocus

New 4:3 video formats

NEW

Environment Texturing

Realistic Rendering



Realistic Rendering

Position and orientation



Realistic Rendering

Position and orientation

Scale



Realistic Rendering

Position and orientation

Scale

Lighting



Realistic Rendering

Position and orientation

Scale

Lighting

Shadow



Realistic Rendering

Position and orientation

Scale

Lighting

Shadow

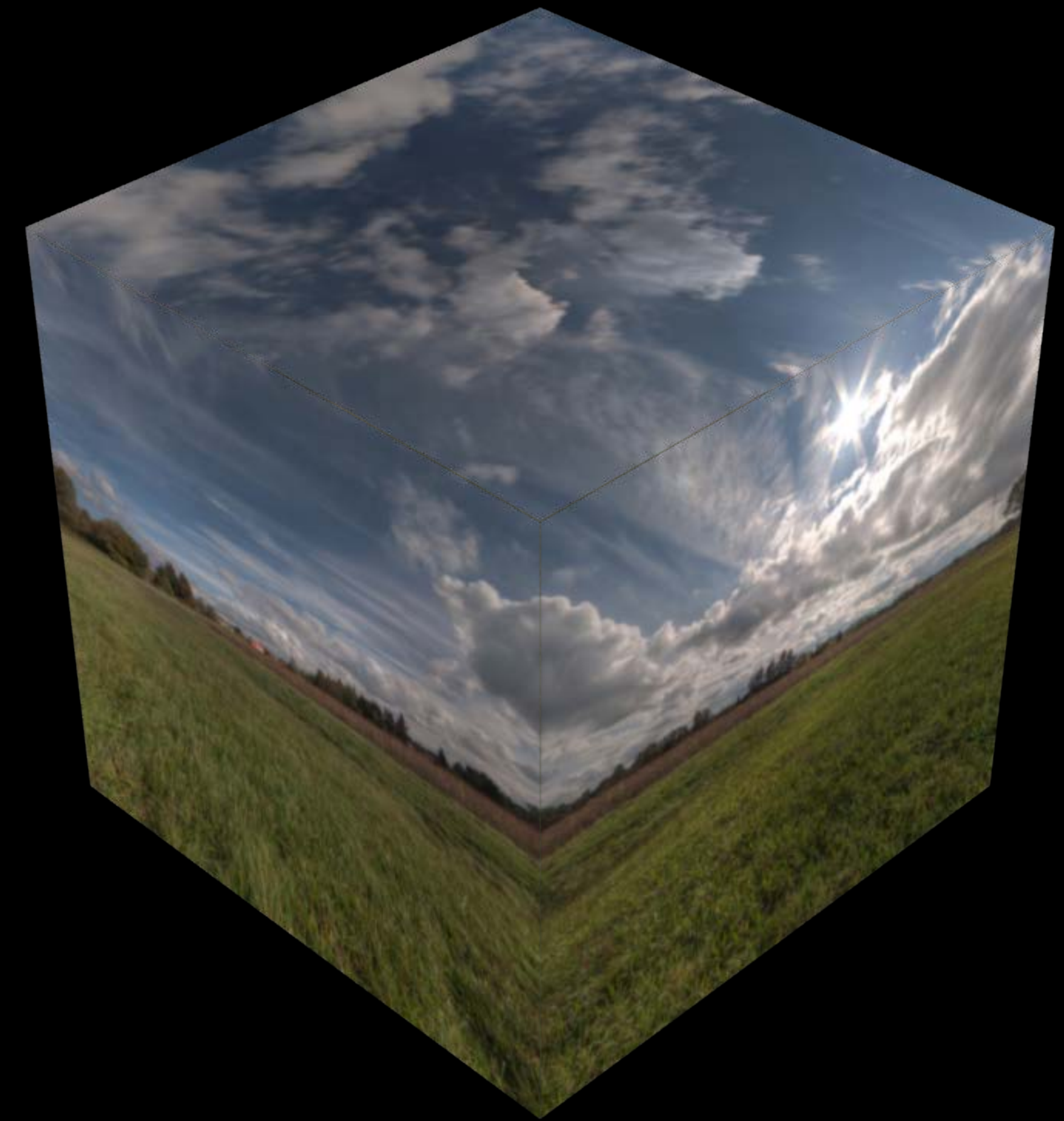
Reflection of textures





Environment Texturing

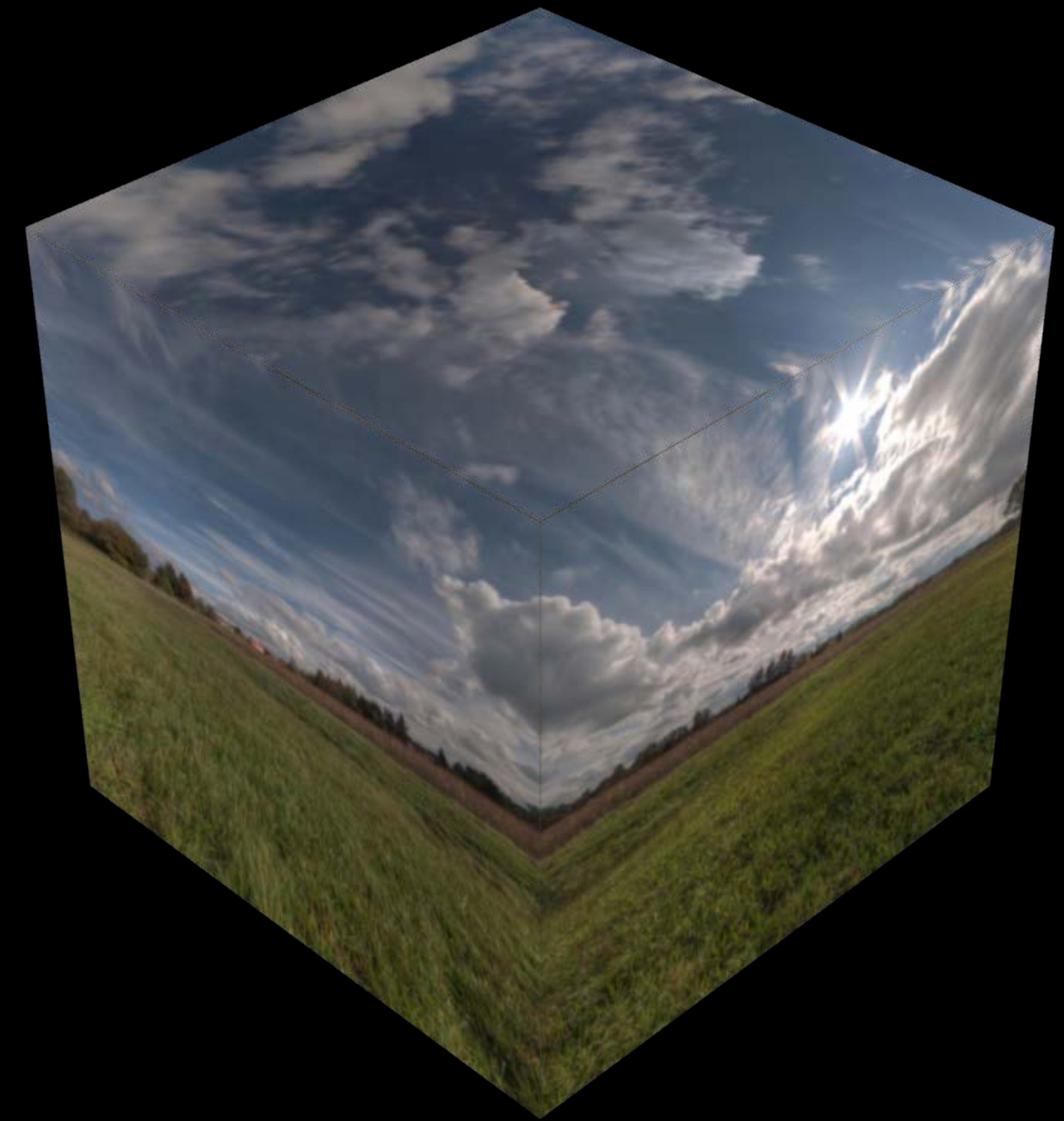
NEW



Environment Texturing

NEW

Gathers scene texture

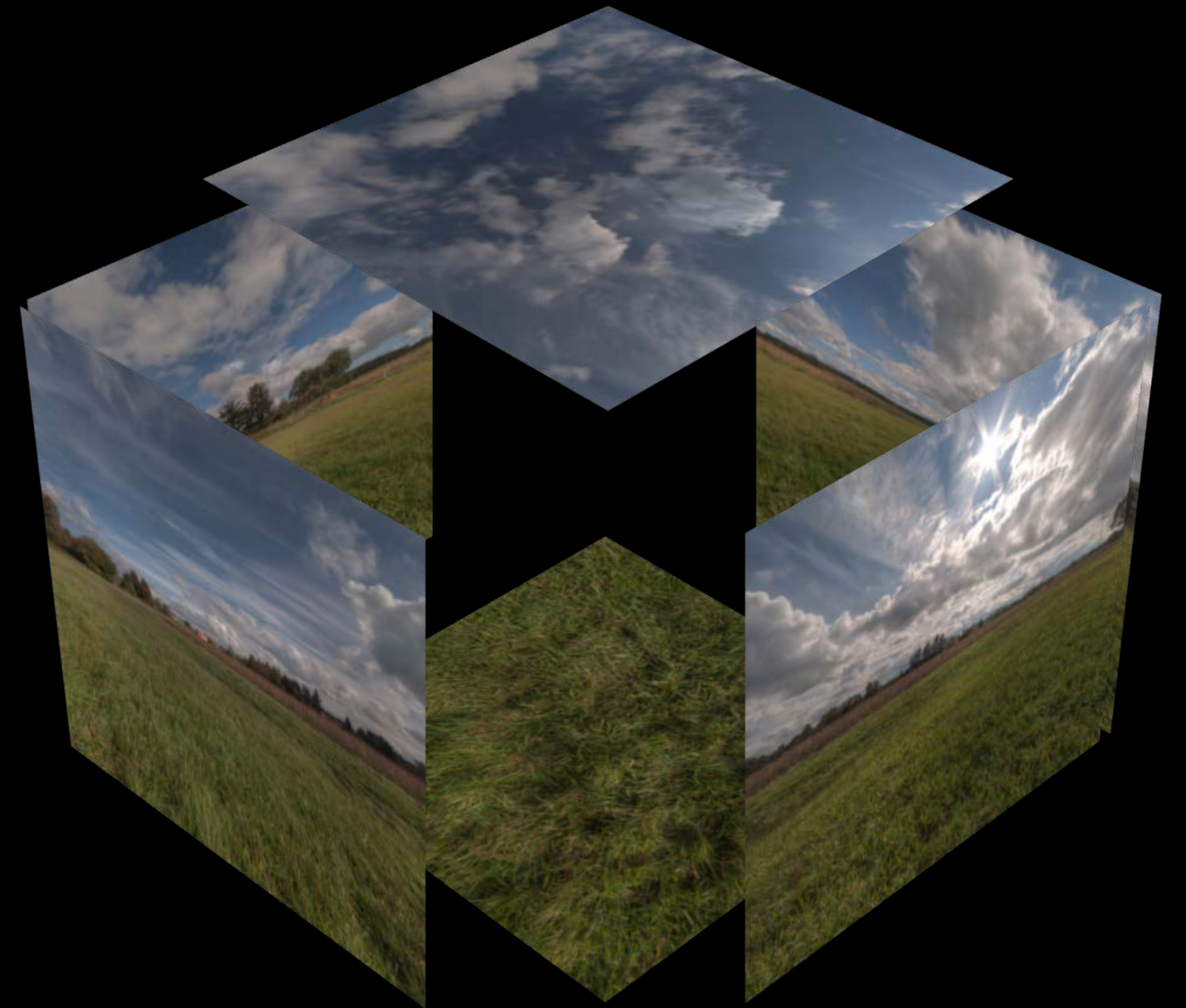


Environment Texturing

NEW

Gathers scene texture

Cube map representation



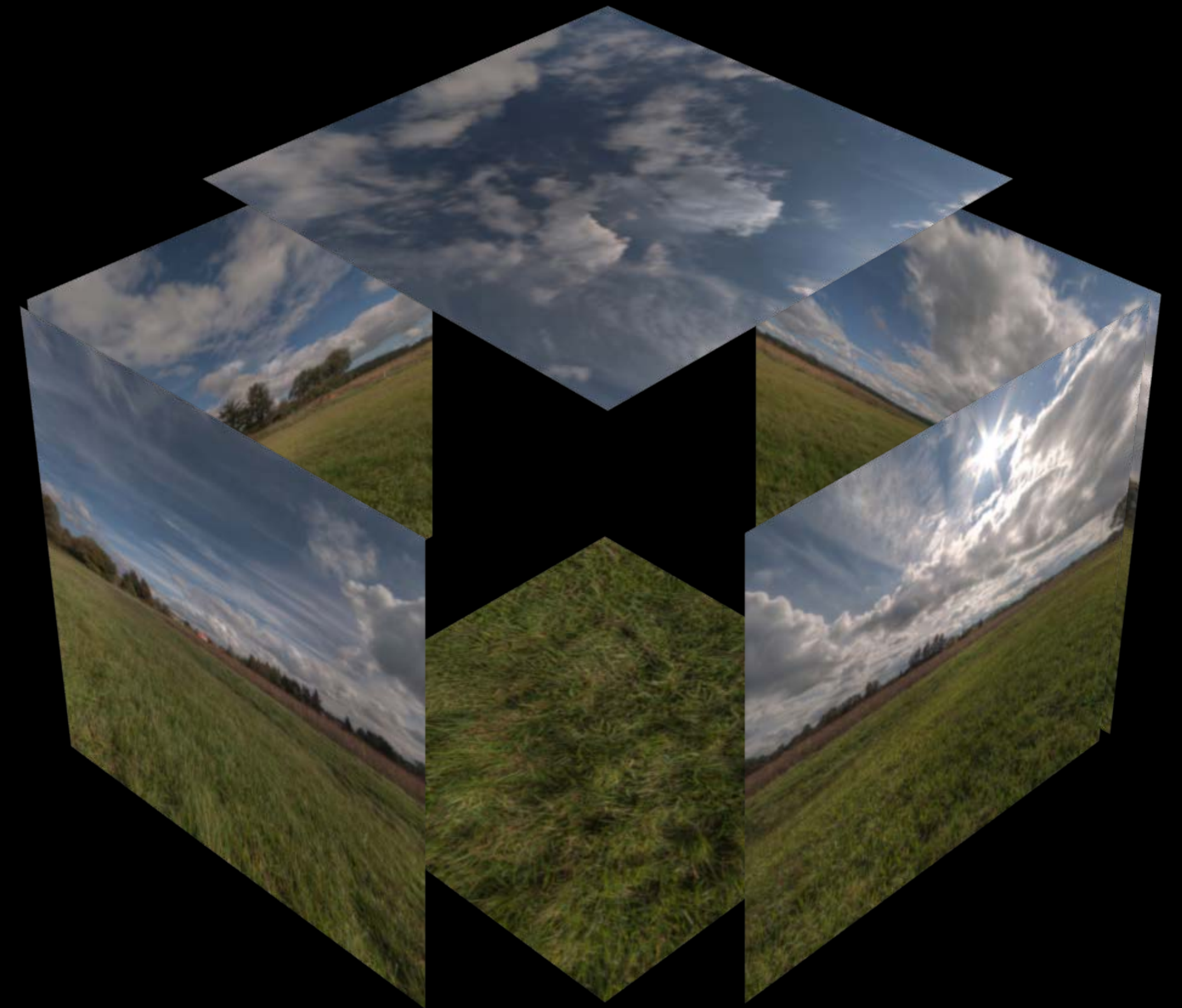
Environment Texturing

NEW

Gathers scene texture

Cube map representation

Used as reflection probe



Environment Texturing

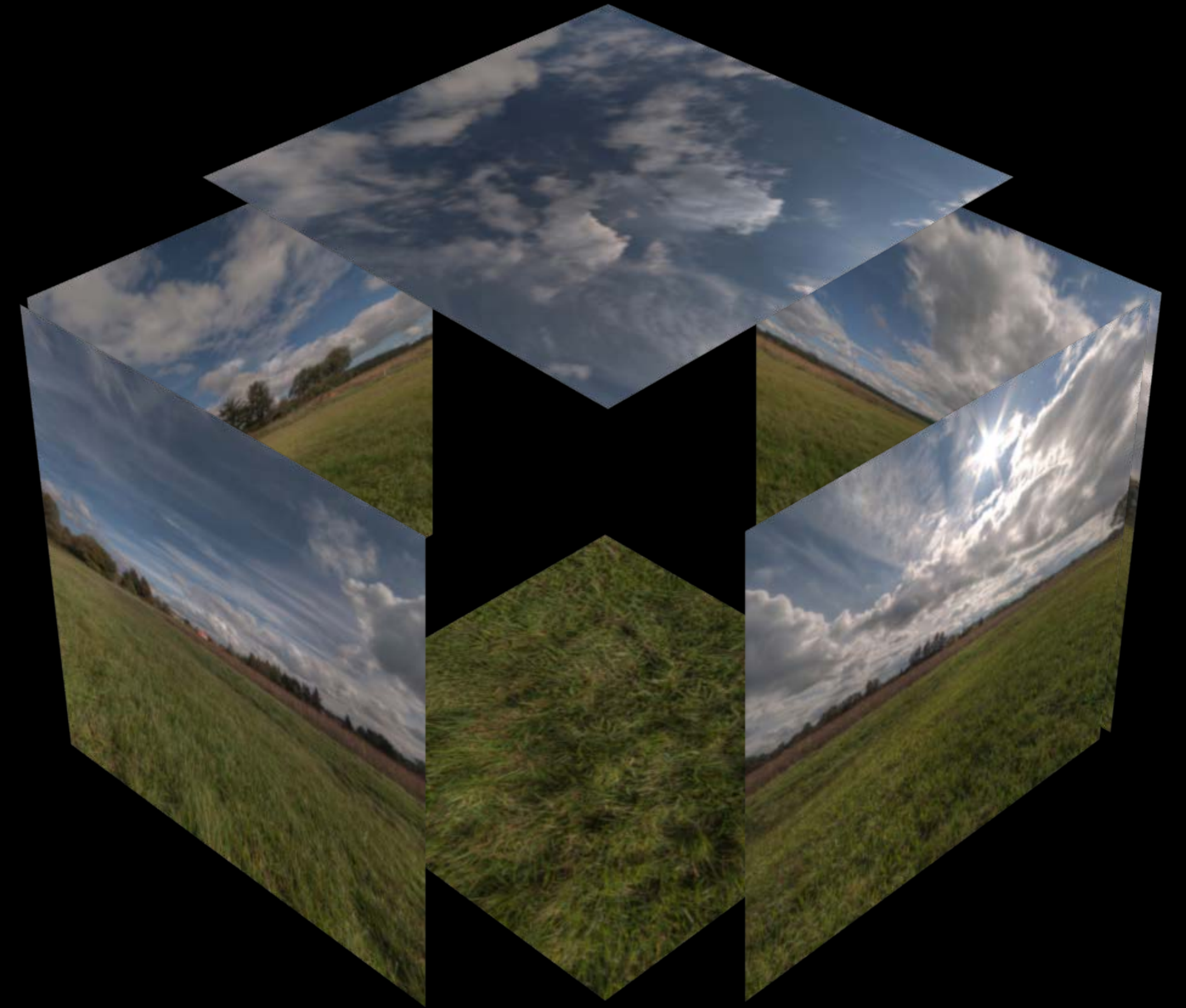
NEW

Gathers scene texture

Cube map representation

Used as reflection probe

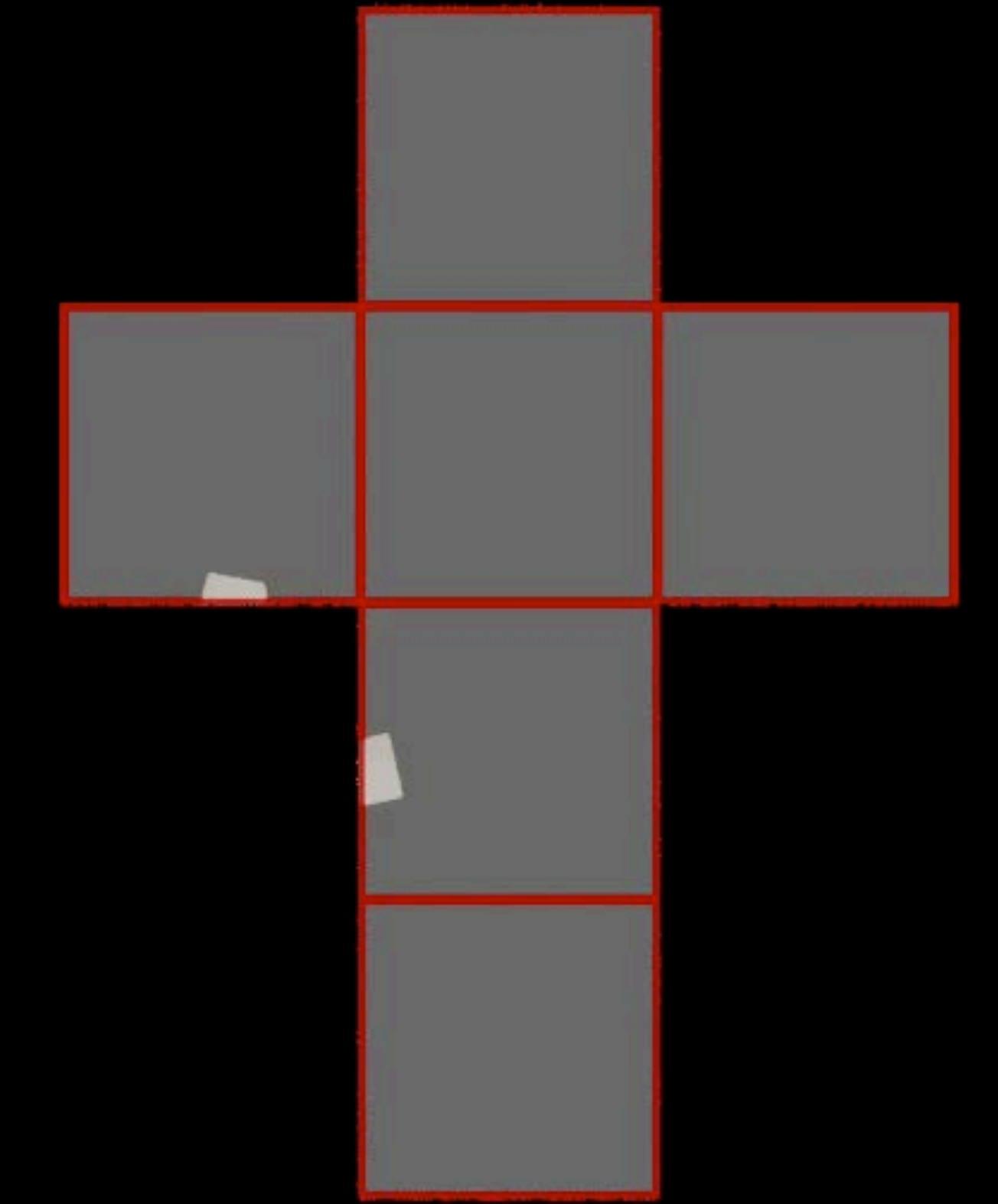
Reflective objects



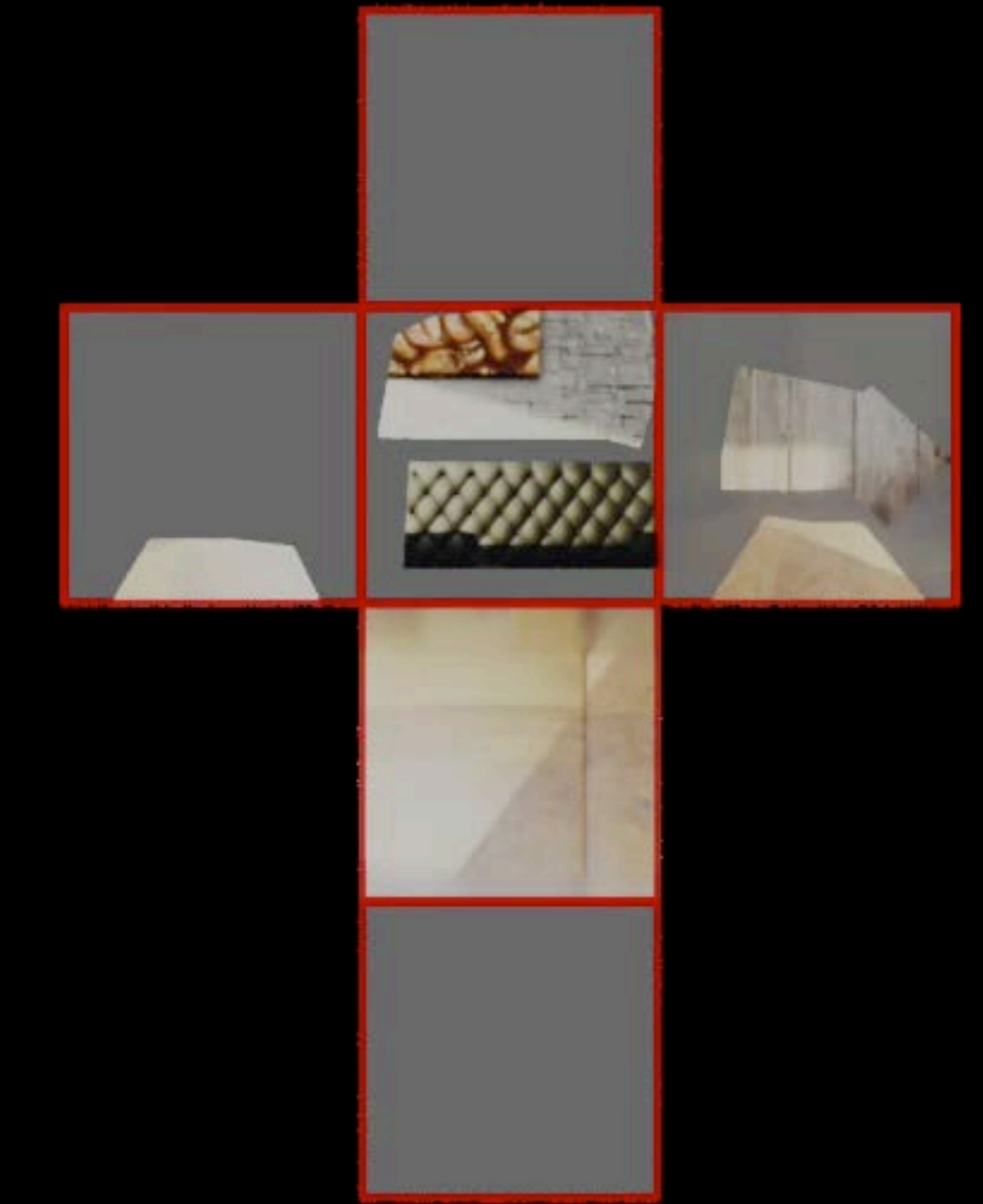
Environment Texturing



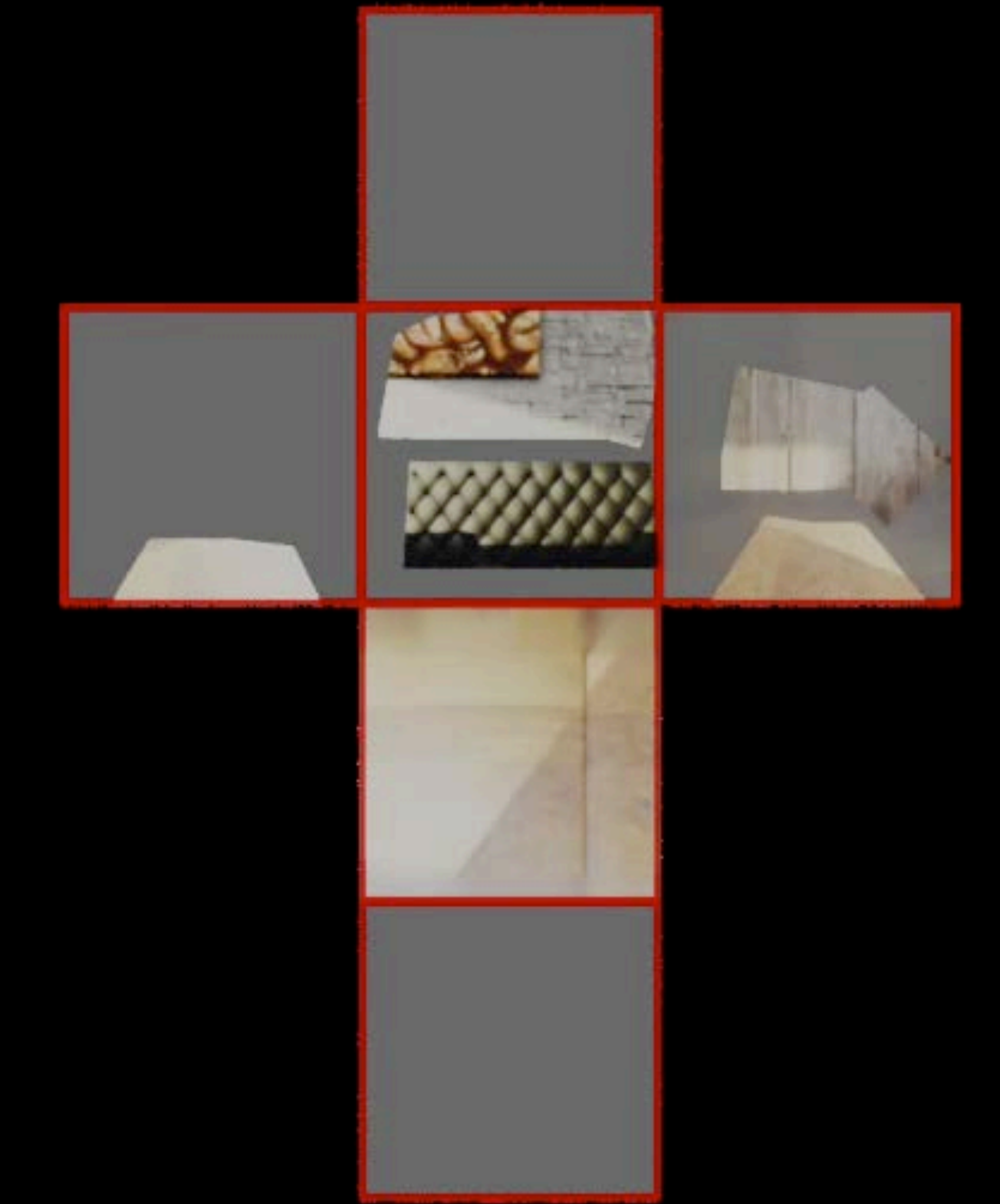
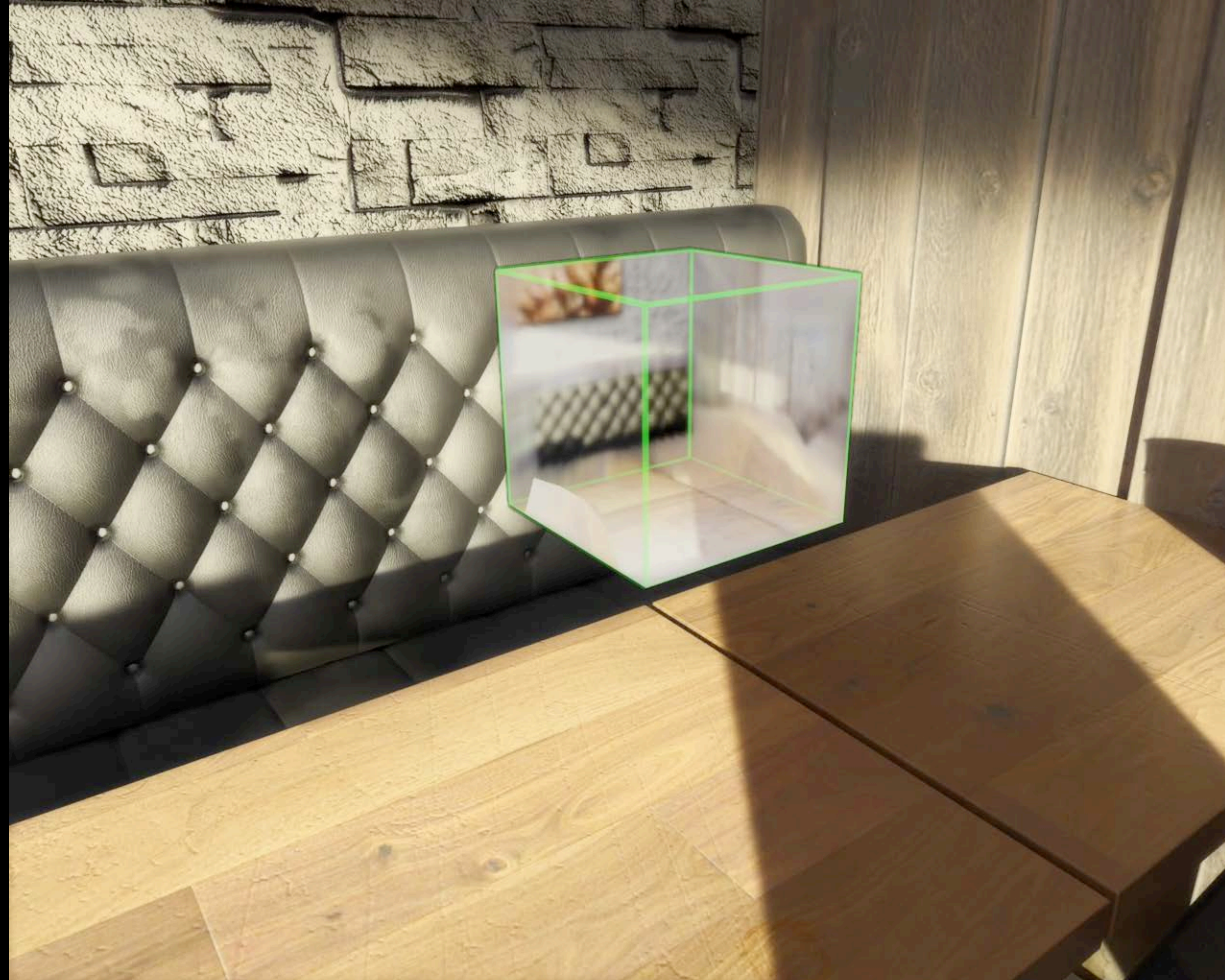
Environment Texturing



Environment Texturing



Environment Texturing



```
// Environment Texturing

// Create world tracking configuration
let configuration = ARWorldTrackingConfiguration()

// Enable automatic environment texturing
configuration.environmentTexturing = .automatic

// Run the configuration
session.run(configuration)
```

```
// Environment Texturing
```

```
// Create world tracking configuration
```

```
let configuration = ARWorldTrackingConfiguration()
```

```
// Enable automatic environment texturing
```

```
configuration.environmentTexturing = .automatic
```

```
// Run the configuration
```

```
session.run(configuration)
```



```
// Environment Texturing

// Create world tracking configuration
let configuration = ARWorldTrackingConfiguration()

// Enable automatic environment texturing
configuration.environmentTexturing = .automatic

// Run the configuration
session.run(configuration)
```

```
// Environment Texturing

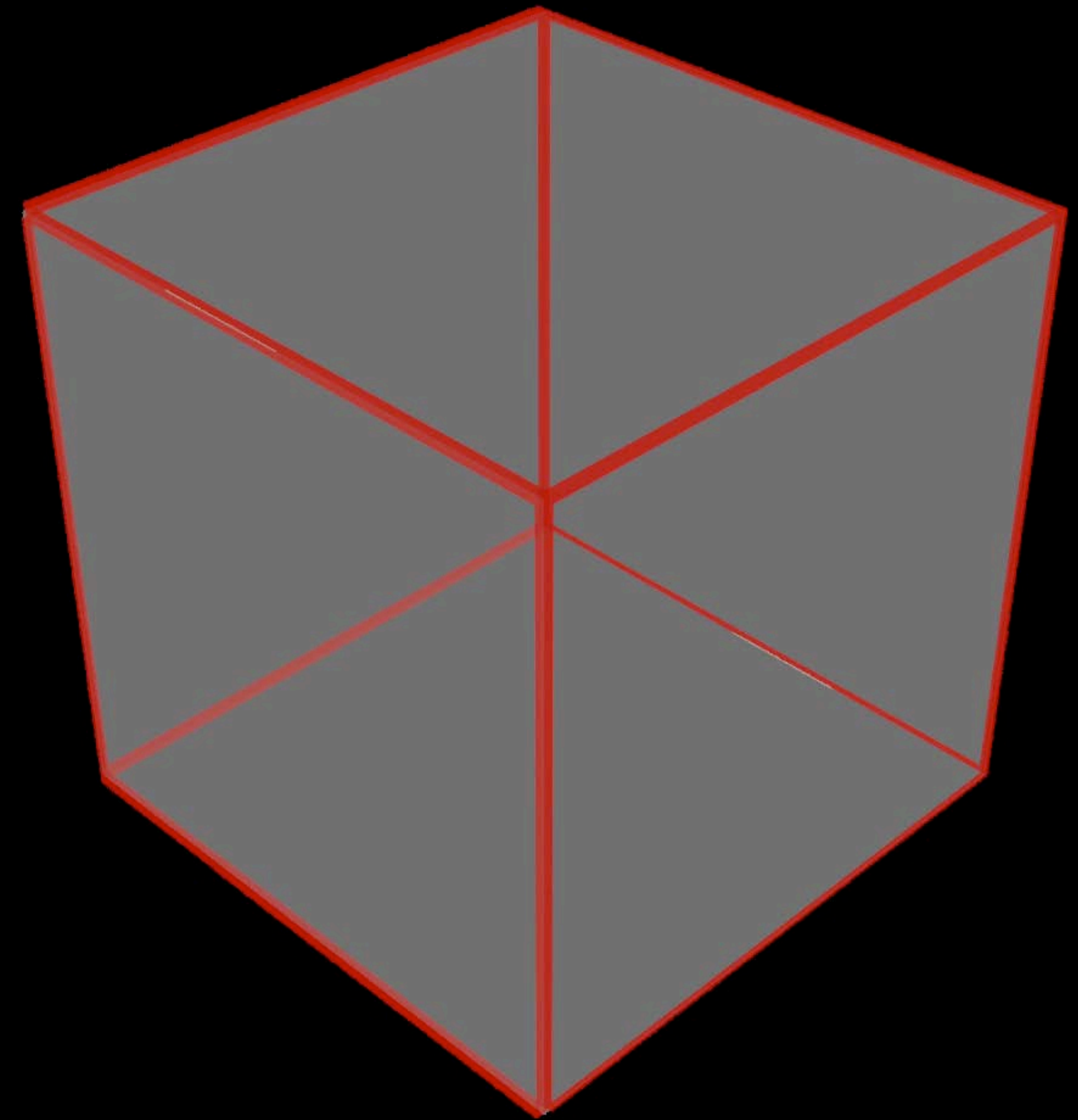
// Create world tracking configuration
let configuration = ARWorldTrackingConfiguration()

// Enable automatic environment texturing
configuration.environmentTexturing = .automatic

// Run the configuration
session.run(configuration)
```

AREnvironmentProbeAnchor

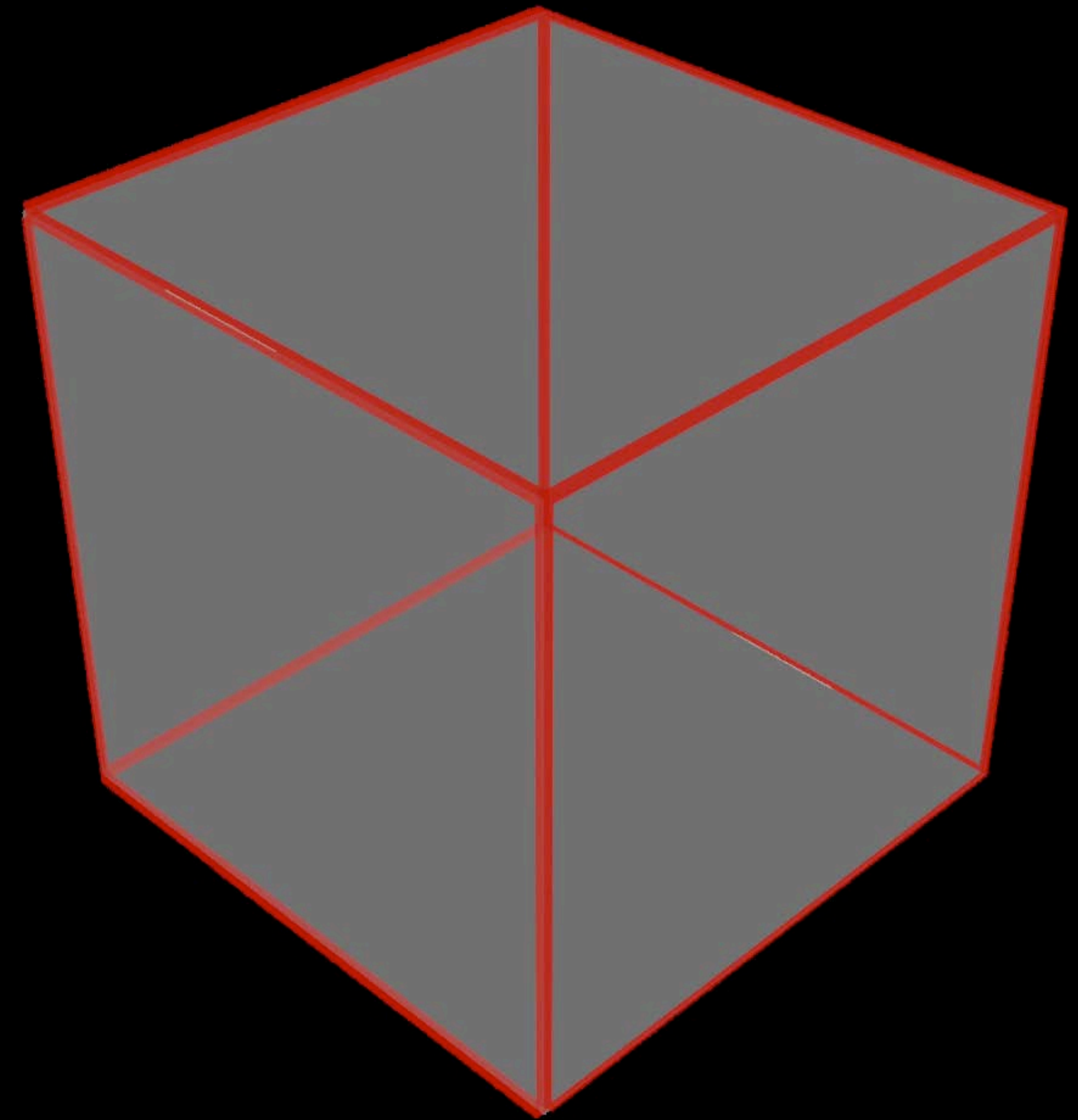
NEW



AREnvironmentProbeAnchor

NEW

Cube map (MTLTexture)

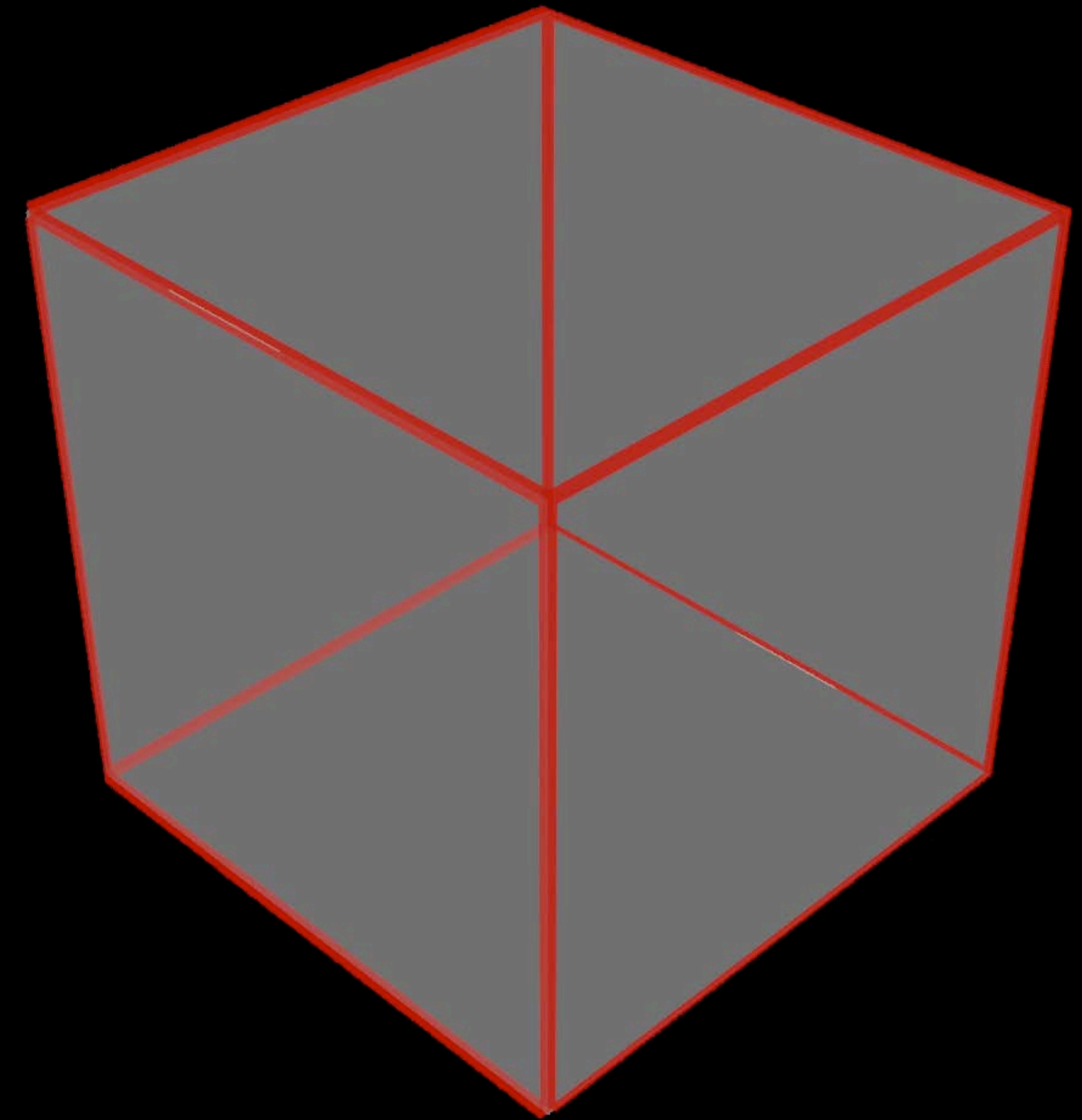


AREnvironmentProbeAnchor

NEW

Cube map (MTLTexture)

Physical extent



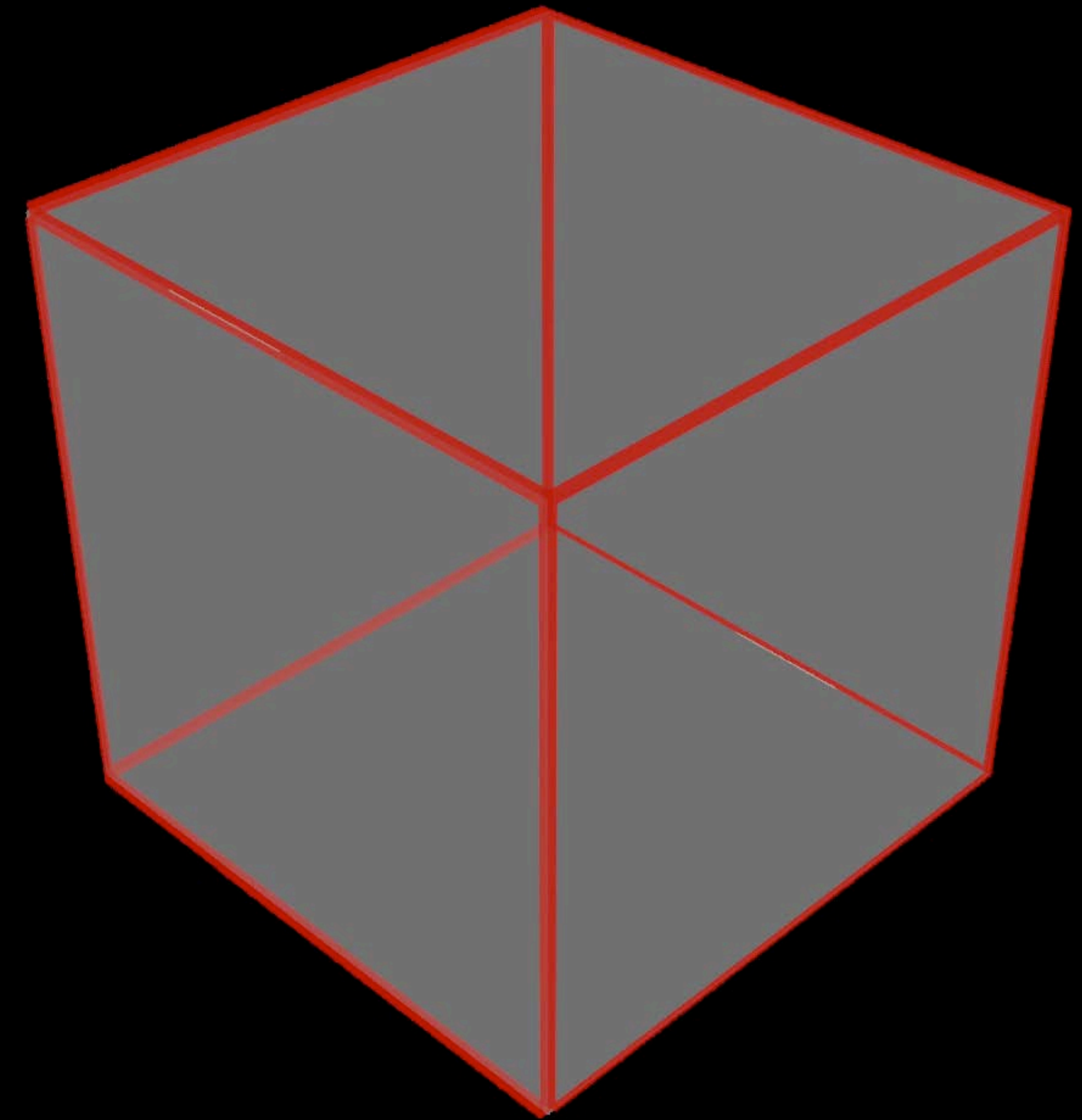
AREnvironmentProbeAnchor

NEW

Cube map (MTLTexture)

Physical extent

Same lifecycle as other anchors



AREnvironmentProbeAnchor

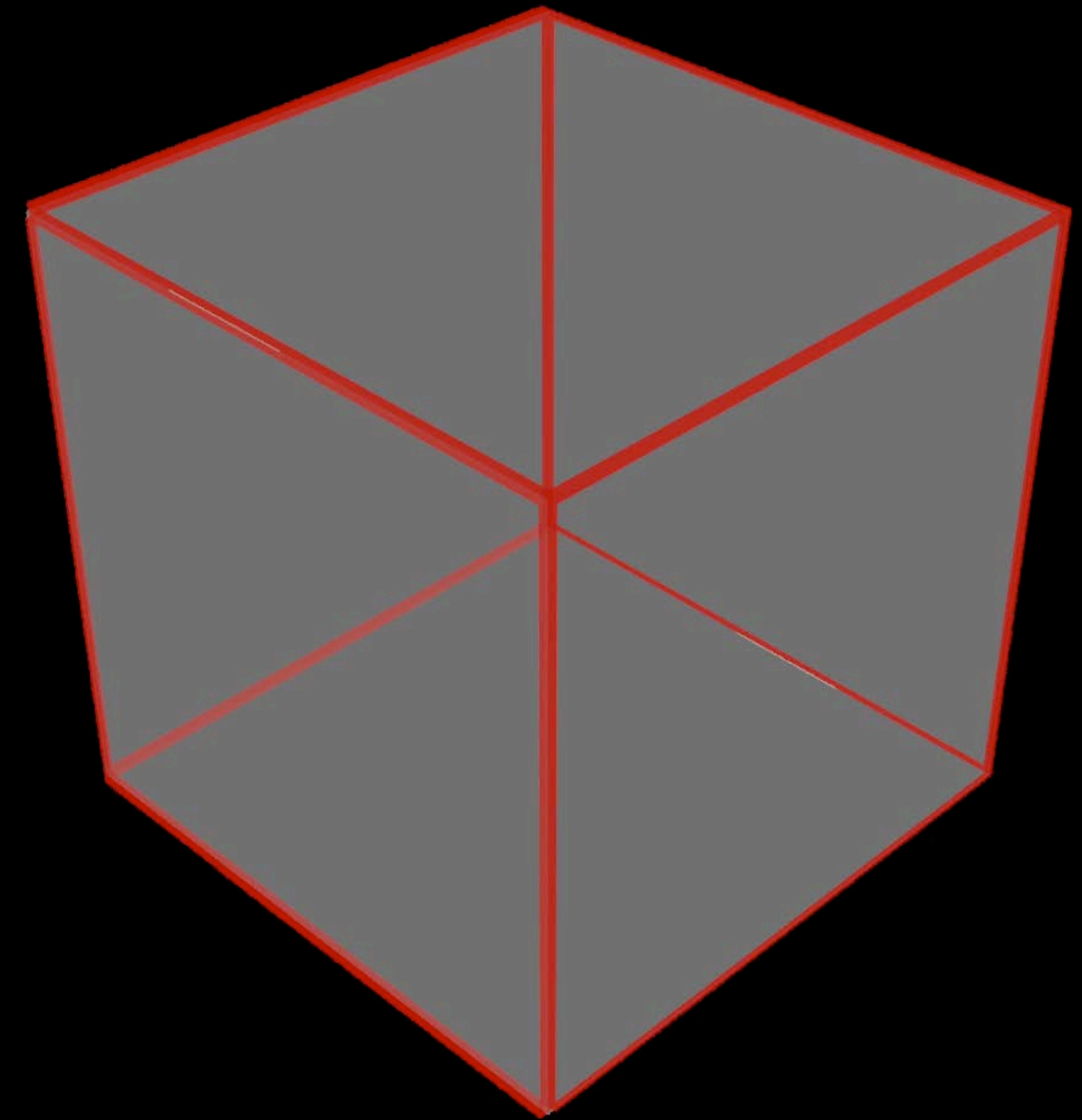
NEW

Cube map (MTLTexture)

Physical extent

Same lifecycle as other anchors

Integrated in ARSCNView



AREnvironmentProbeAnchor

NEW

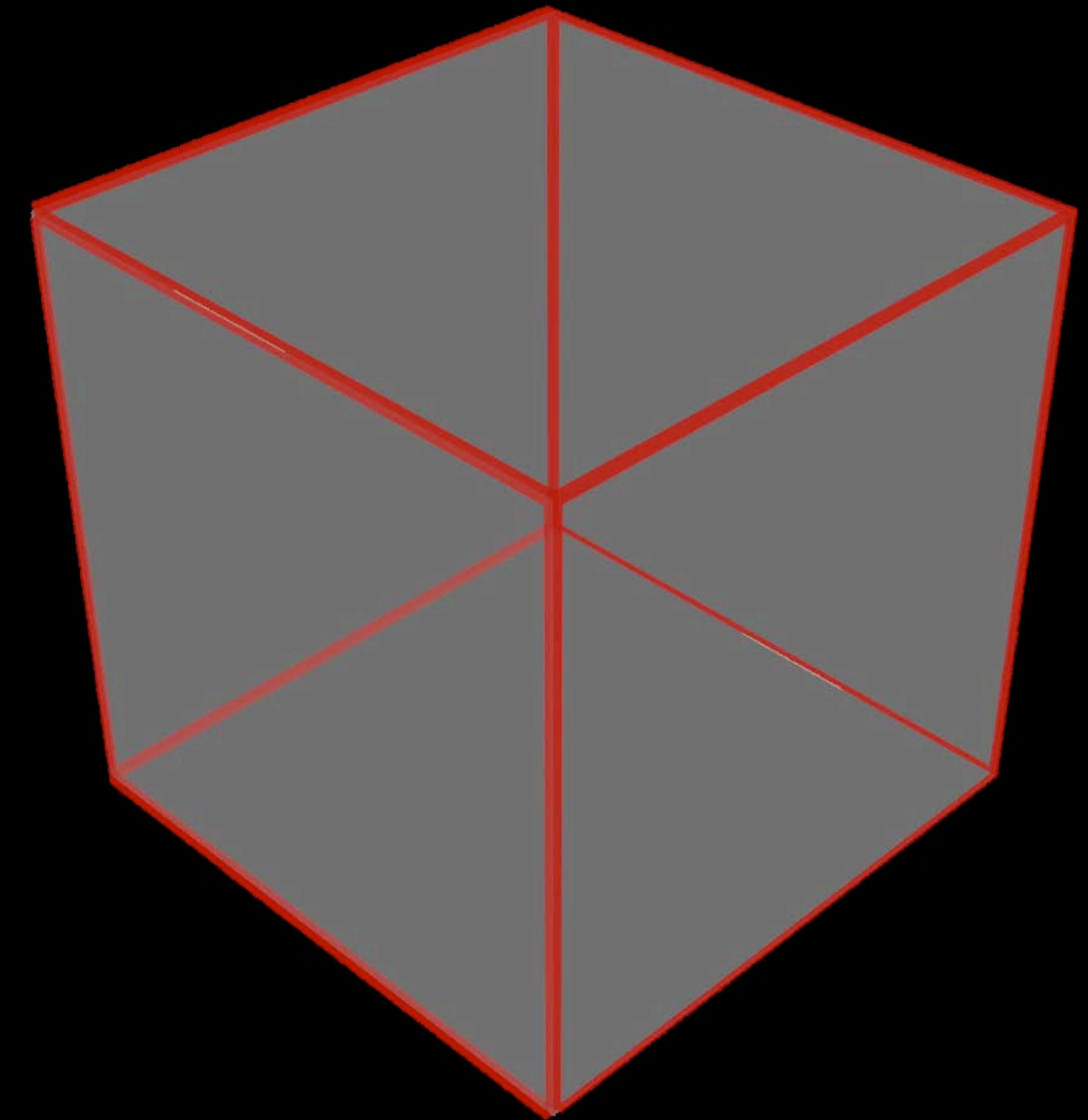
Cube map (MTLTexture)

Physical extent

Same lifecycle as other anchors

Integrated in ARSCNView

Manual placement



```
// Enable manual placement  
configuration.environmentTexturing = .manual
```


Demo

Environment Texturing

Arsalan Malik, ARKit Engineer

Image Tracking

Reinhard Klapfer, ARKit Engineer

Image Detection

Recap



Image Detection

Recap

Recognition of known static 2D images



Image Detection

Recap

Recognition of known static 2D images

Position and orientation



Image Detection

Recap

Recognition of known static 2D images

Position and orientation

Integrated into world tracking



Image Detection

Recap

Recognition of known static 2D images

Position and orientation

Integrated into world tracking

Supported by Xcode asset catalog



Image Tracking

NEW

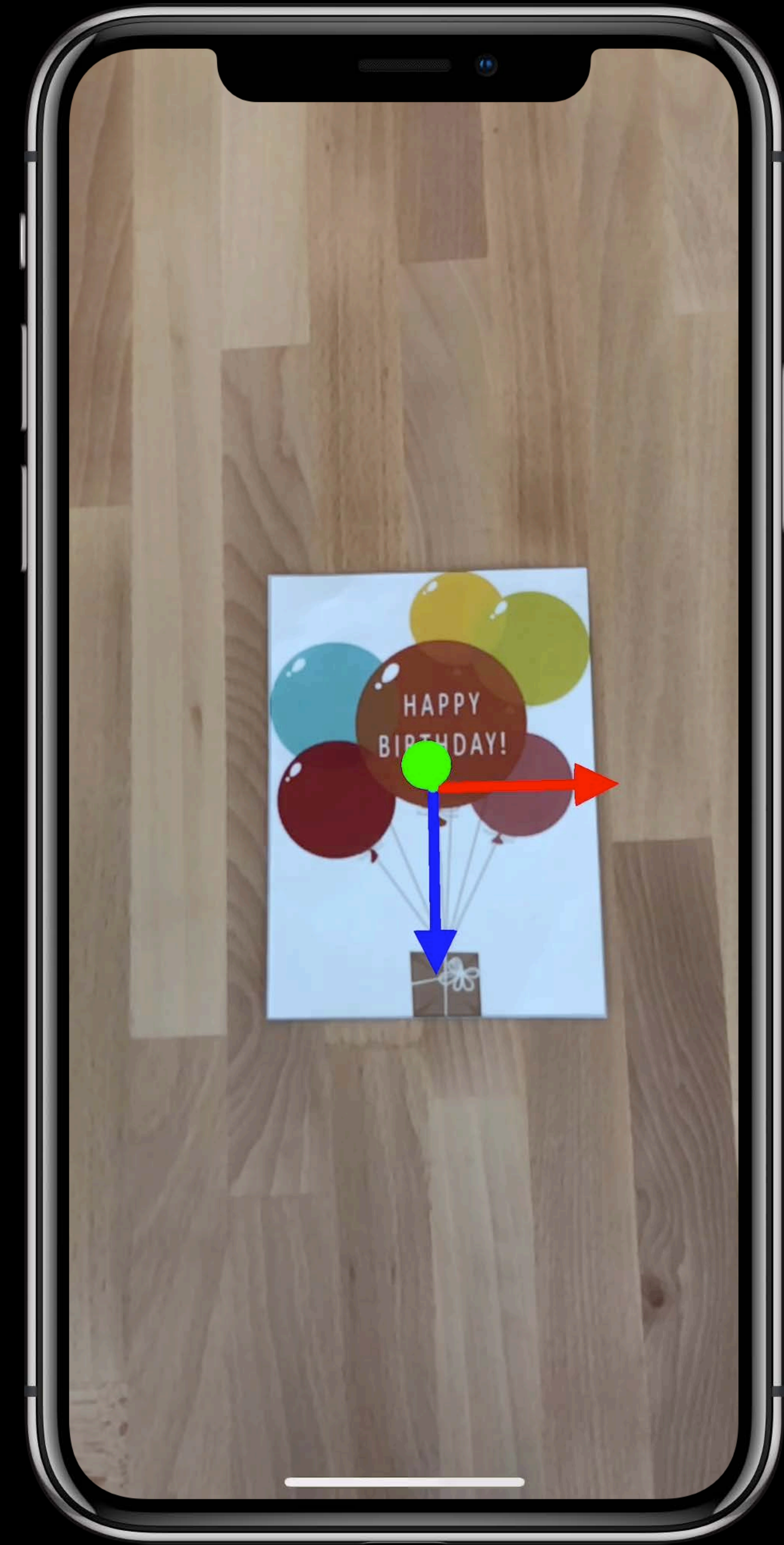


Image Tracking

Images don't need to be static

NEW

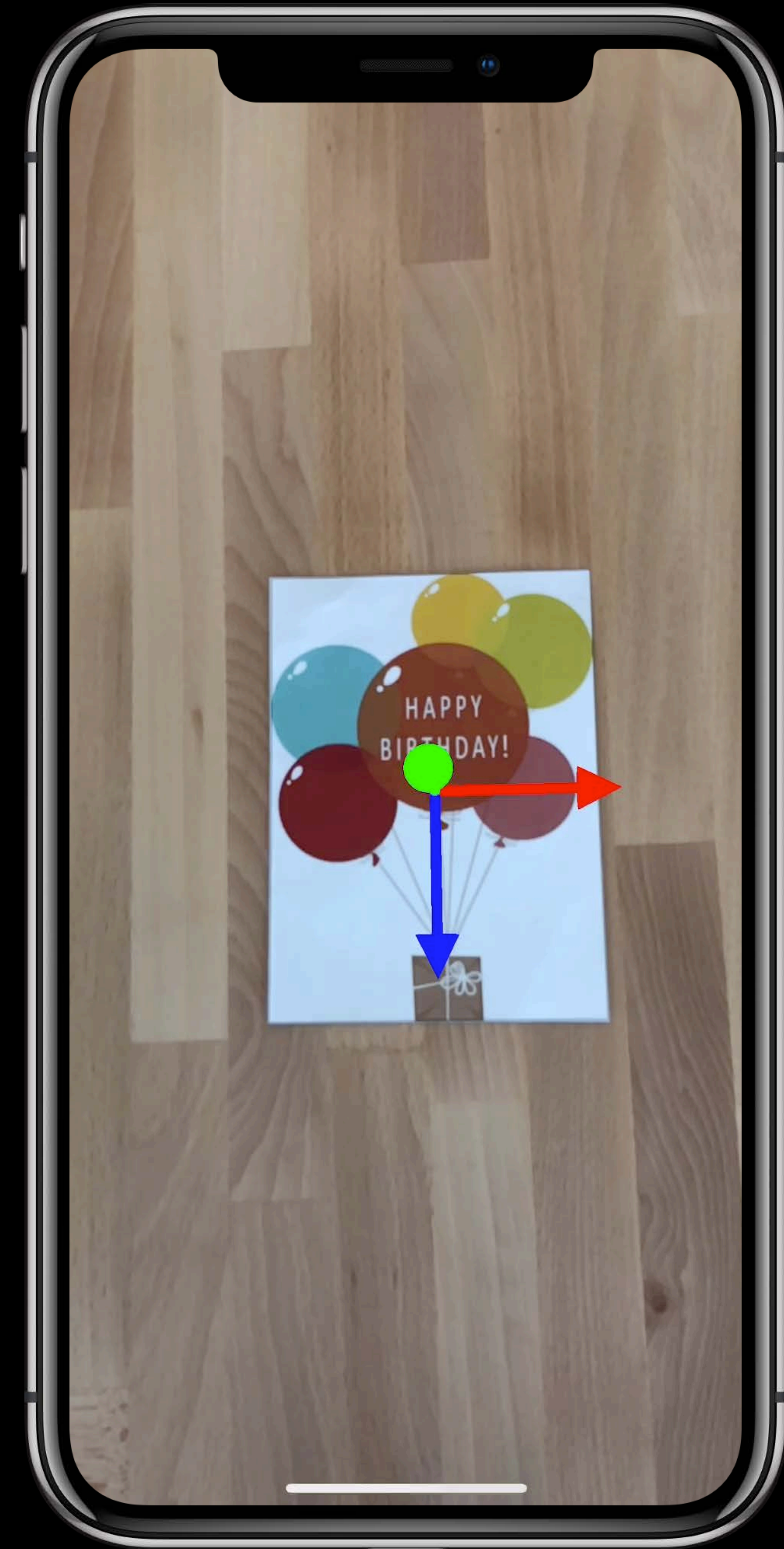


Image Tracking

Images don't need to be static

NEW

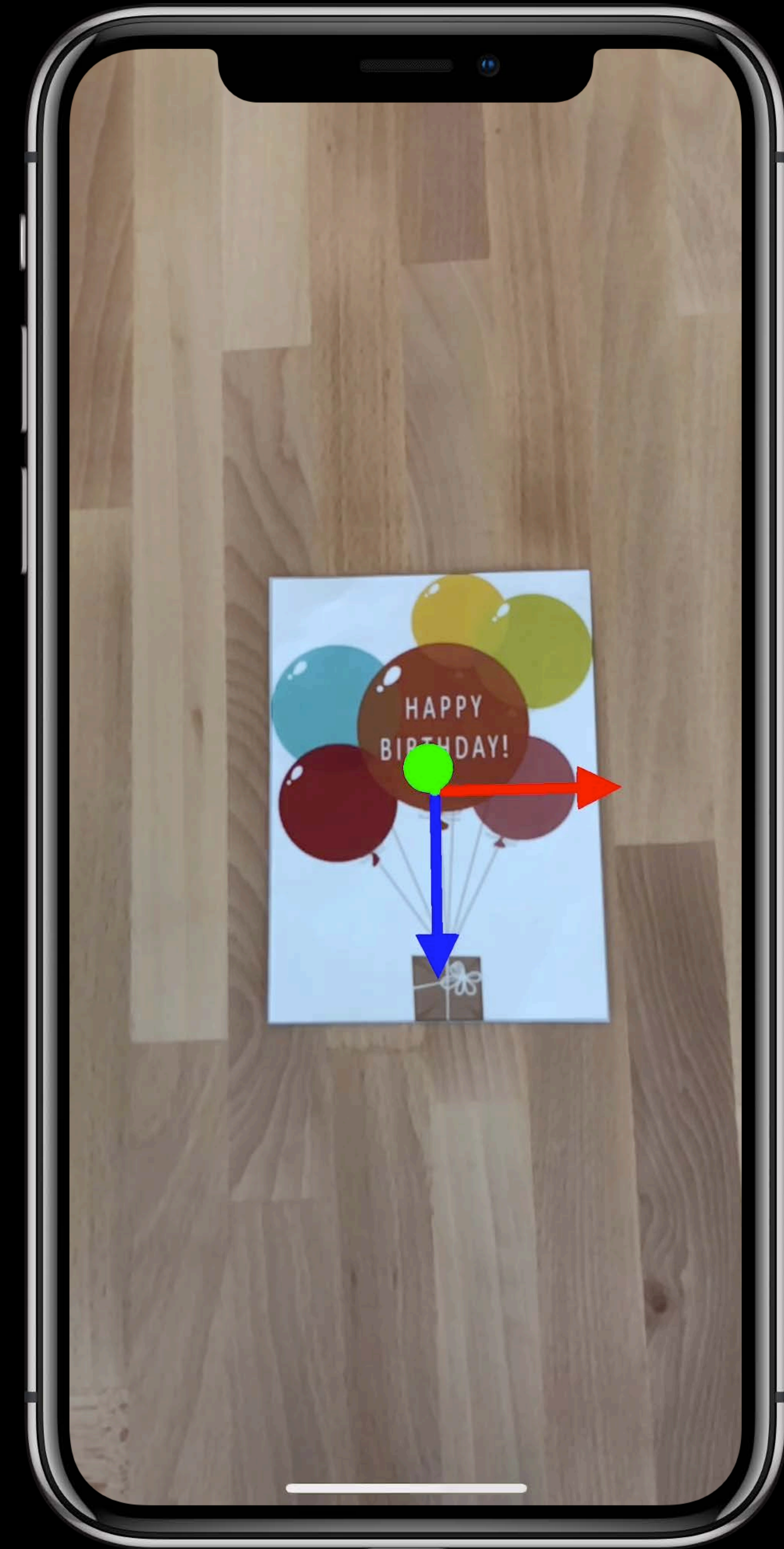
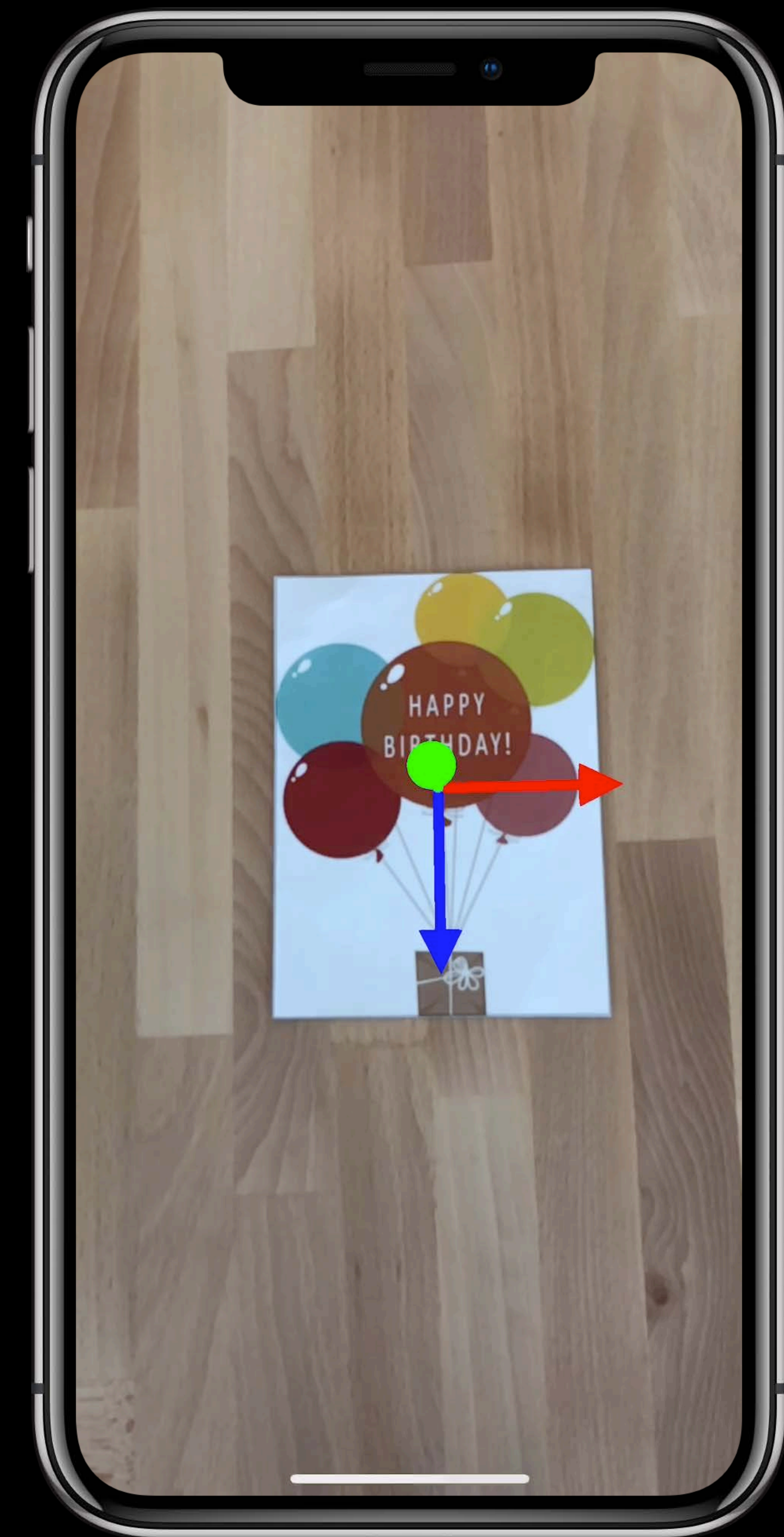


Image Tracking

Images don't need to be static

Position and orientation for every frame

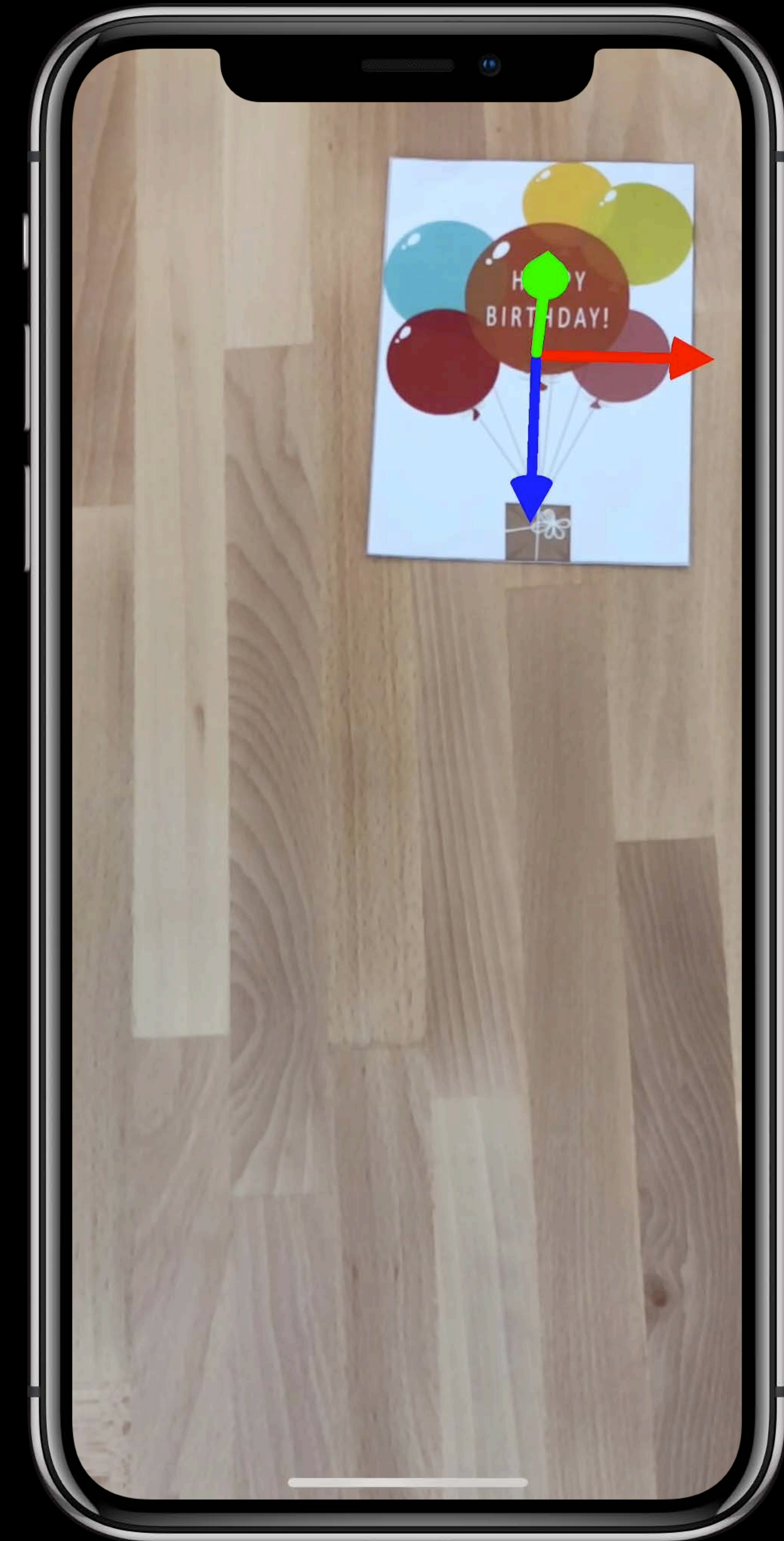


NEW

Image Tracking

Images don't need to be static

Position and orientation for every frame



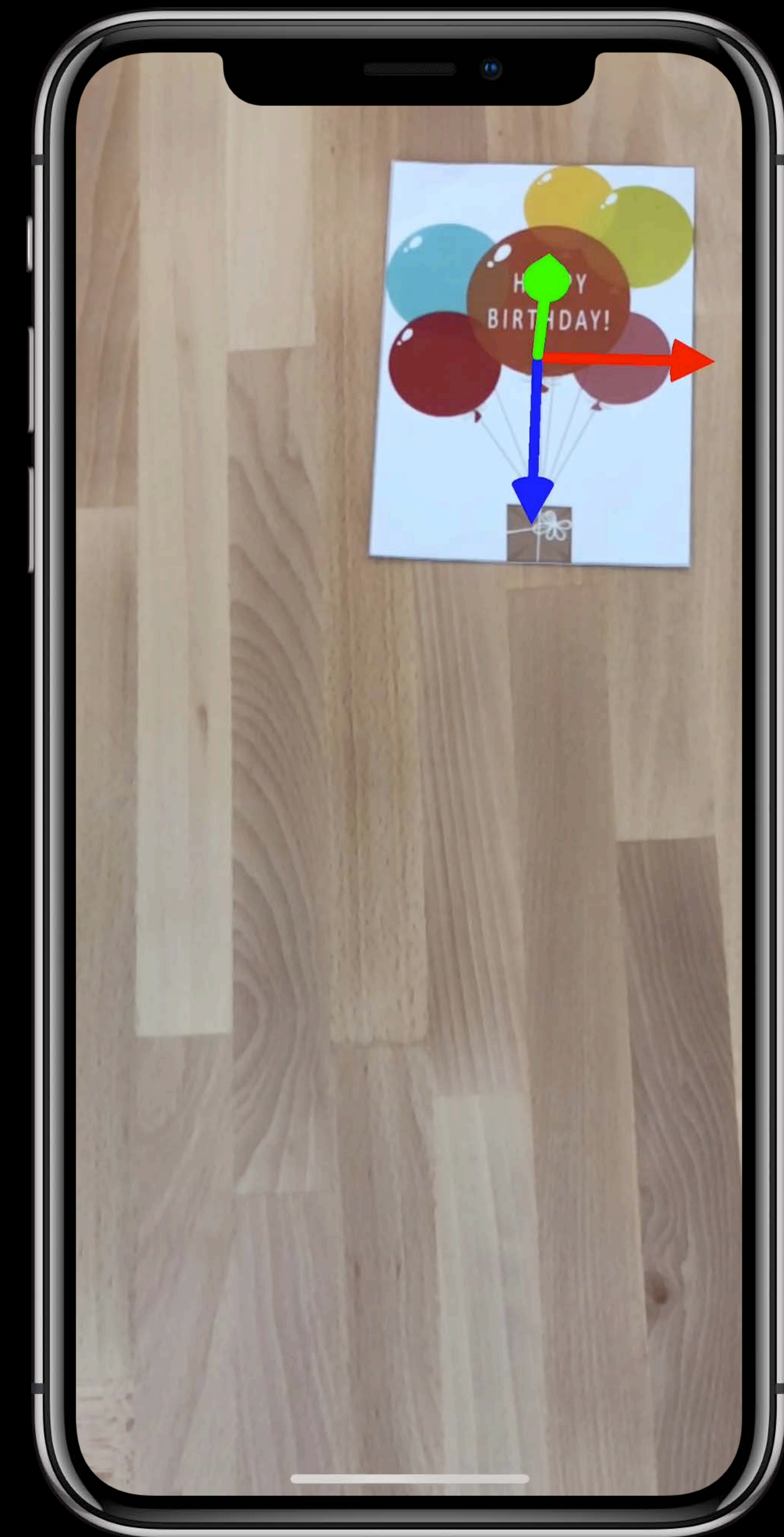
NEW

Image Tracking

Images don't need to be static

Position and orientation for every frame

Track multiple images simultaneously



NEW

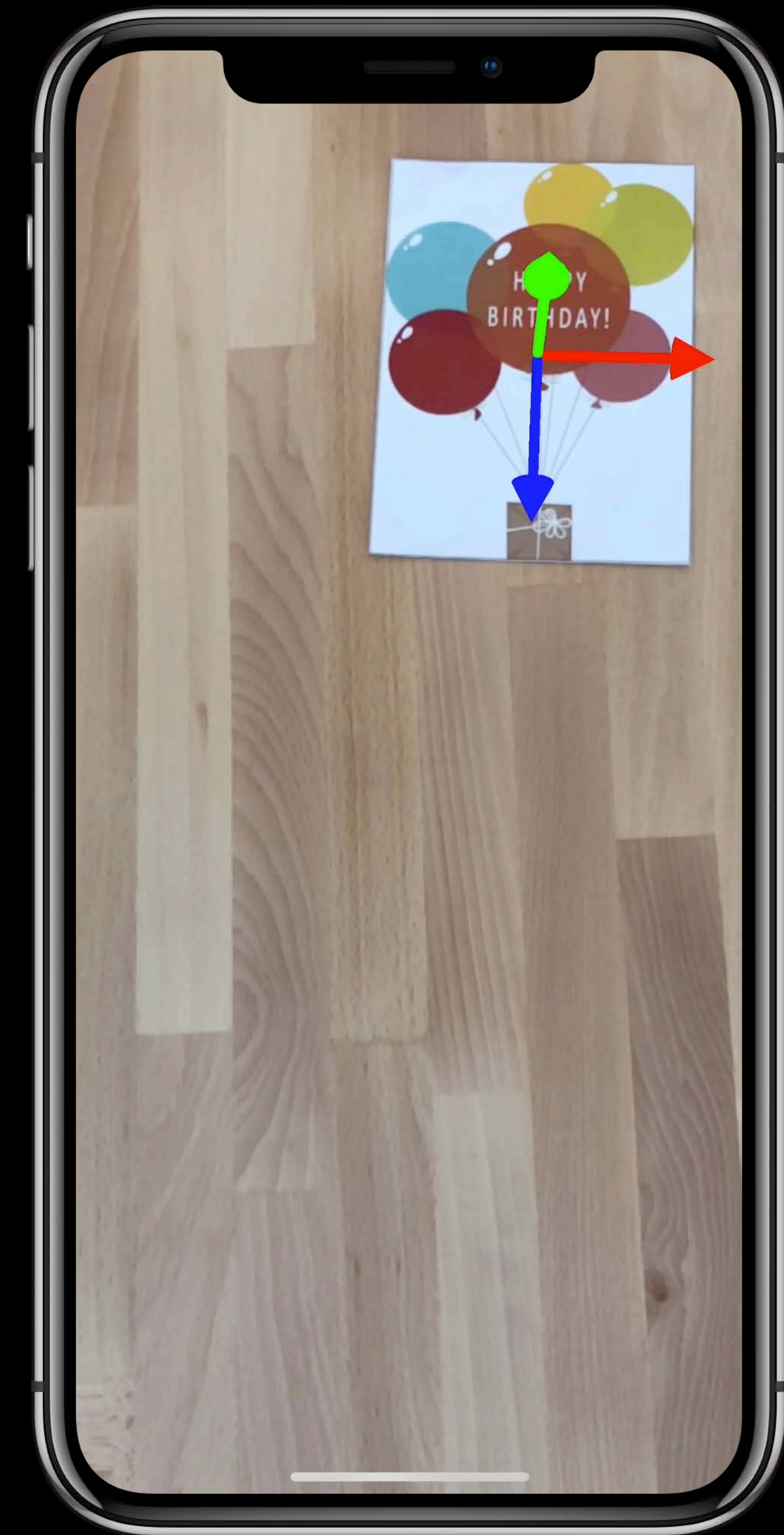
Image Tracking

Images don't need to be static

Position and orientation for every frame

Track multiple images simultaneously

`ARImageTrackingConfiguration`



NEW

Image Tracking

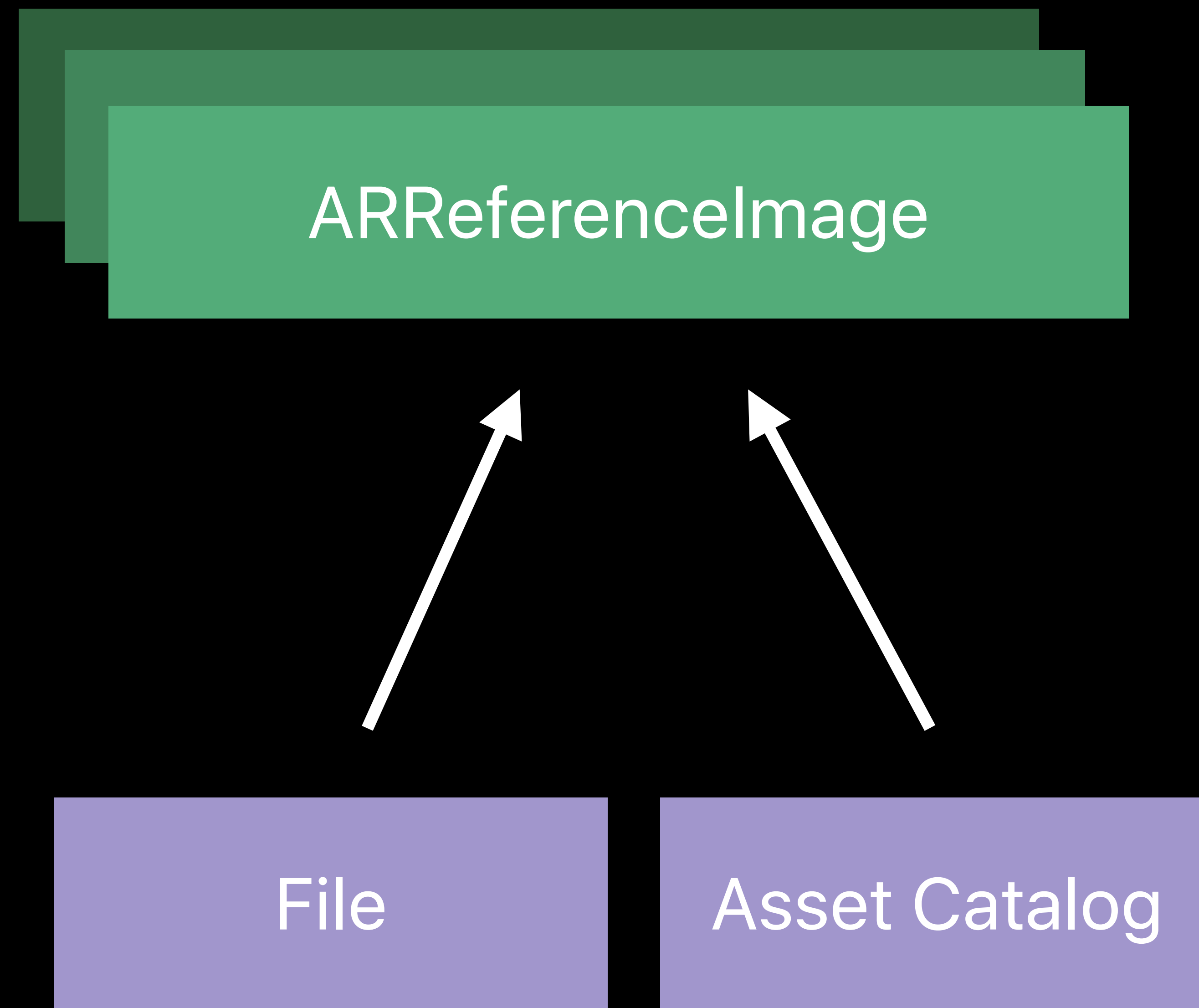


Image Tracking

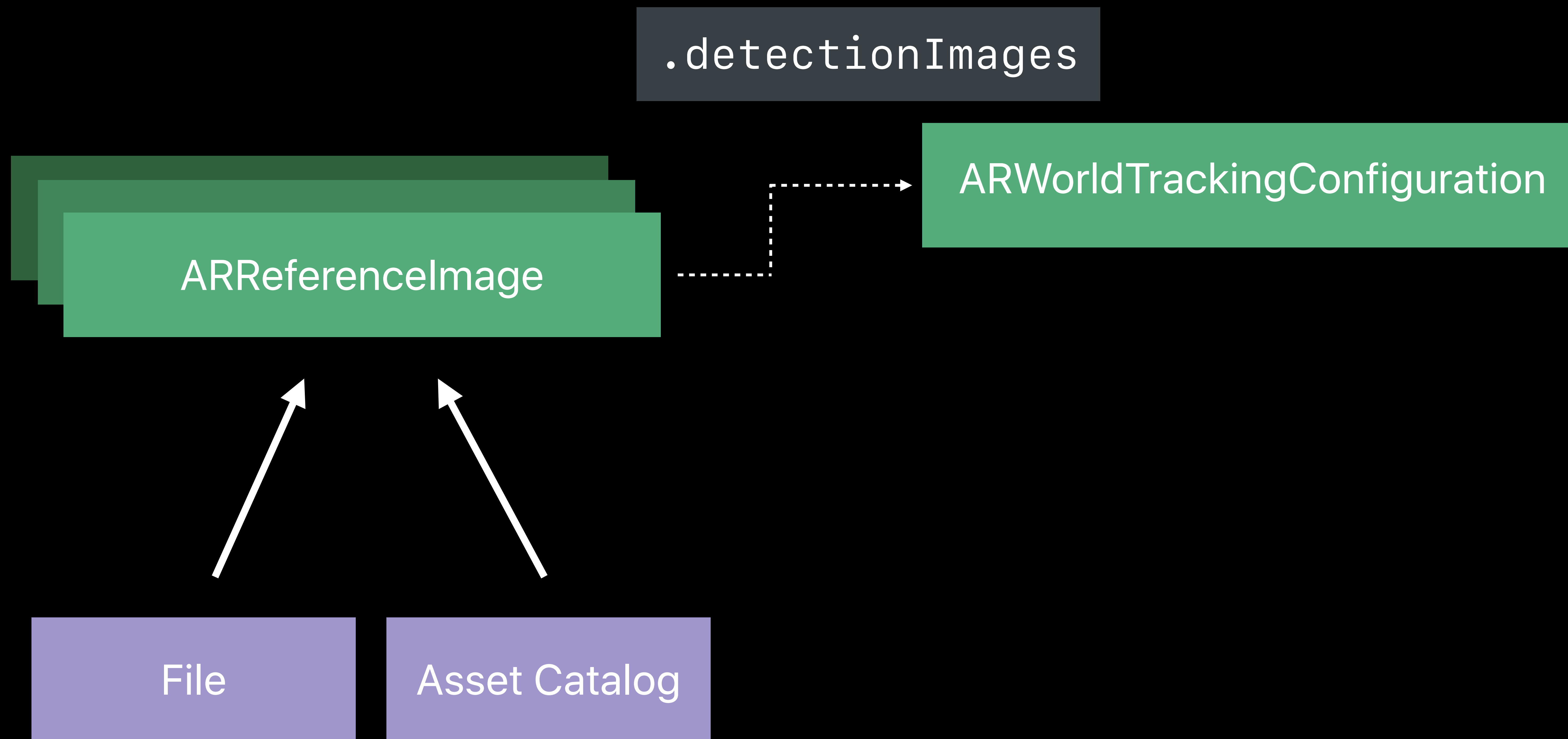


Image Tracking

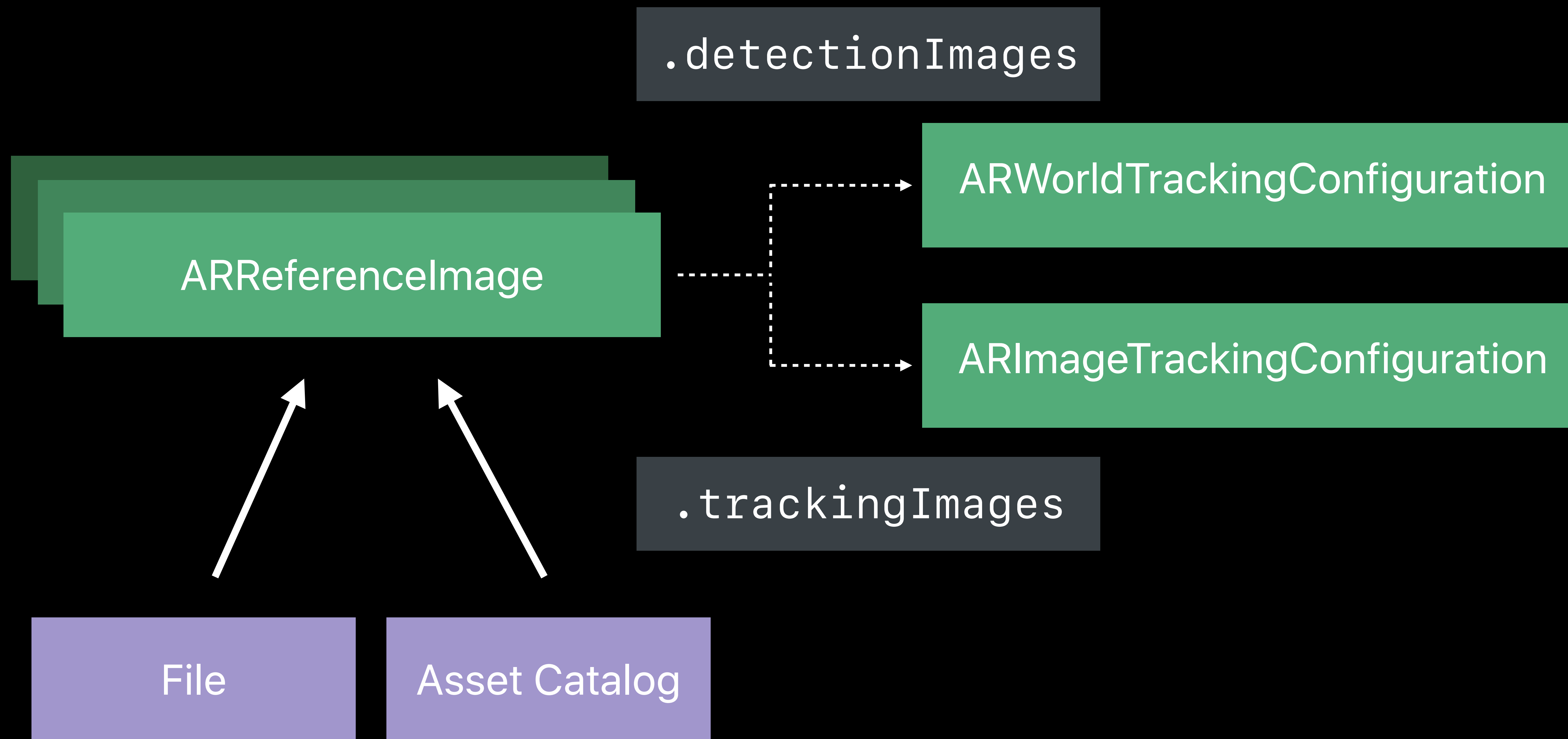


Image Tracking

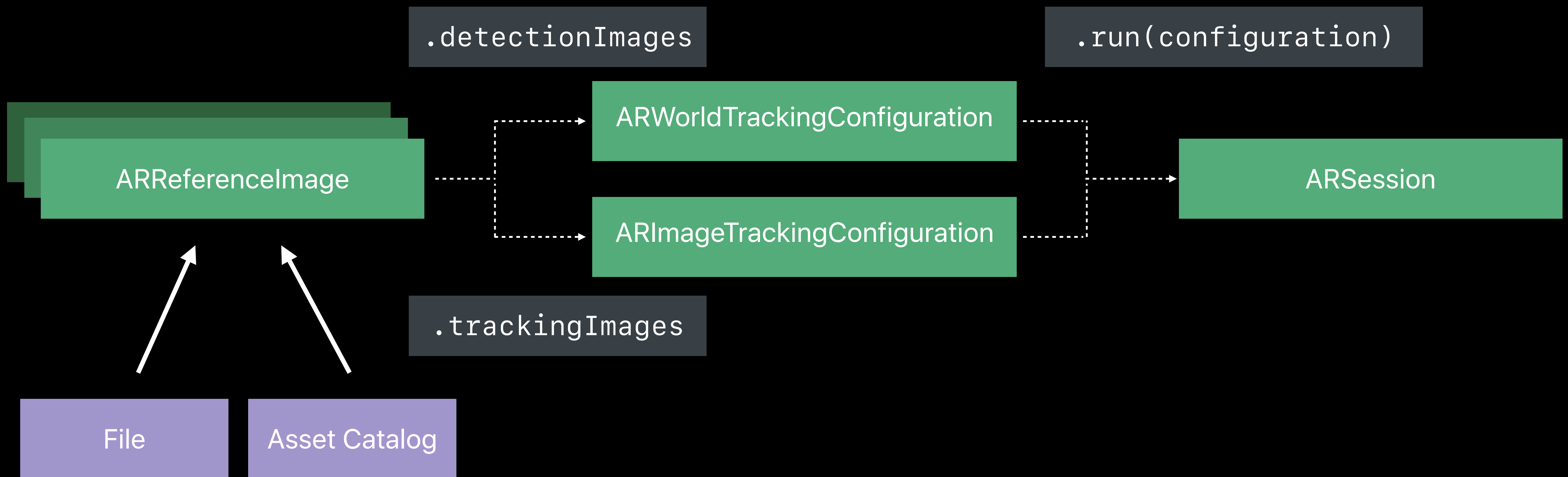


Image Tracking

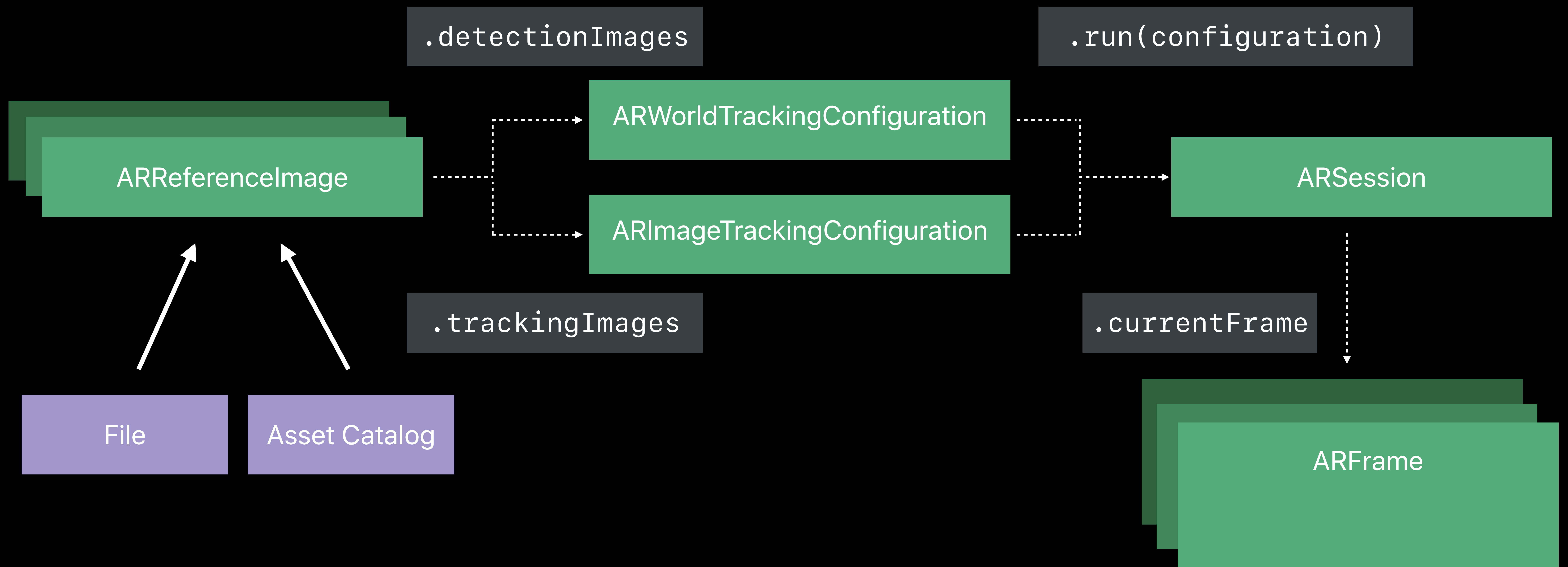
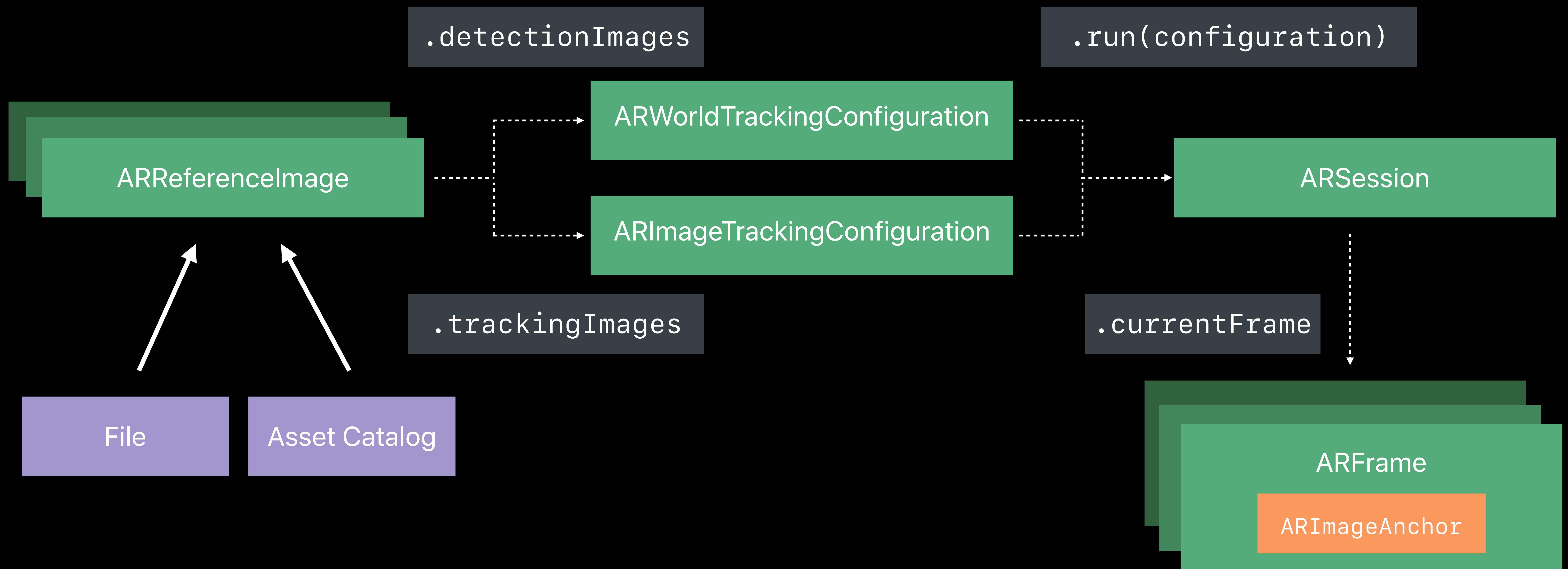
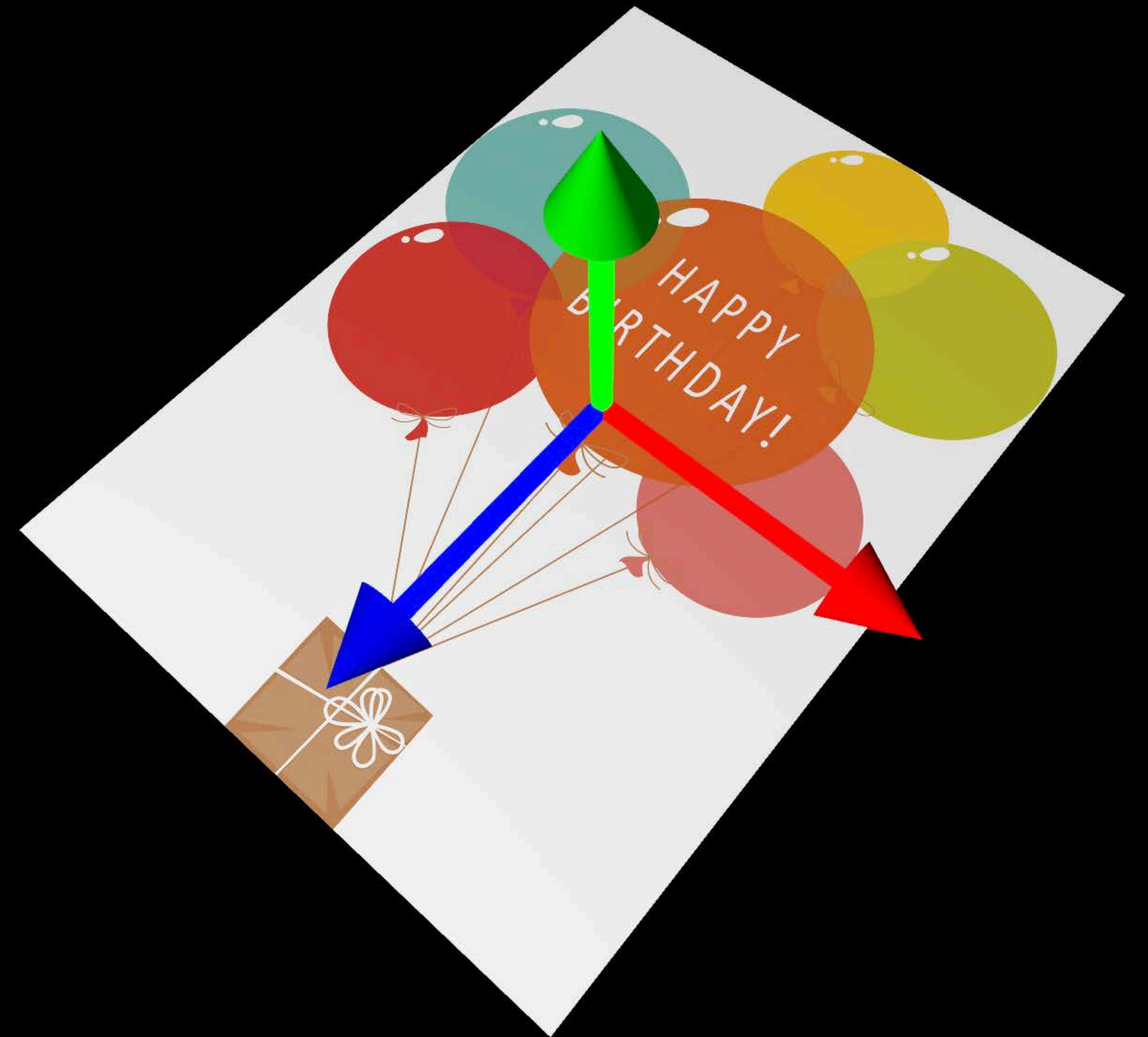


Image Tracking



ARImageAnchor

```
open class ARImageAnchor : ARAnchor, ARTrackable {  
    public var isTracked: Bool { get }  
  
    open var transform: simd_float4x4 { get }  
  
    open var referenceImage: ARReferenceImage { get }  
  
}
```



Using Good Images

Using Good Images



Using Good Images



Distinct features

Well textured

Good contrast

Using Good Images



Distinct features

Well textured

Good contrast

Using Good Images



Distinct features

Well textured

Good contrast

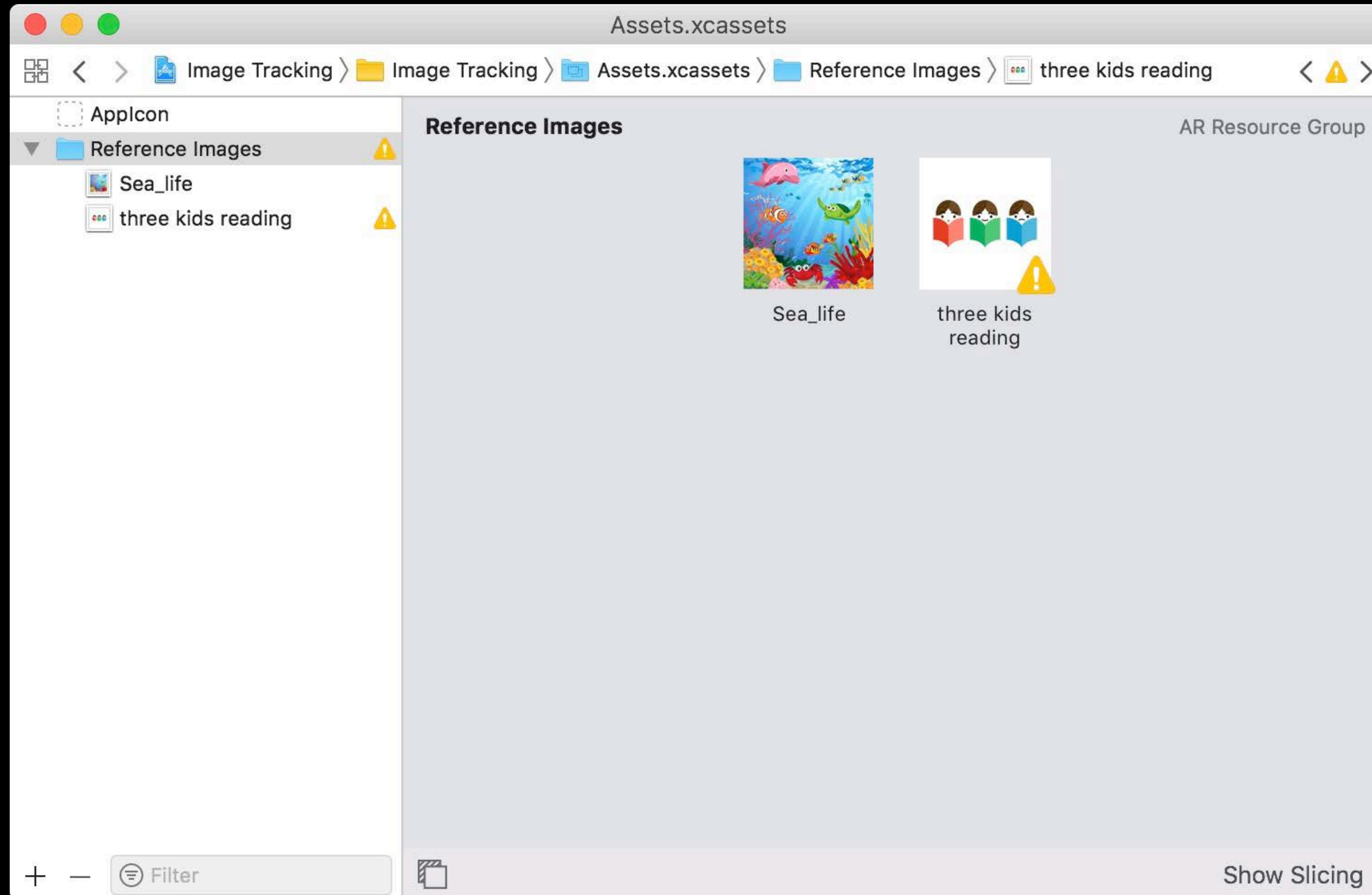


Repetitive structures

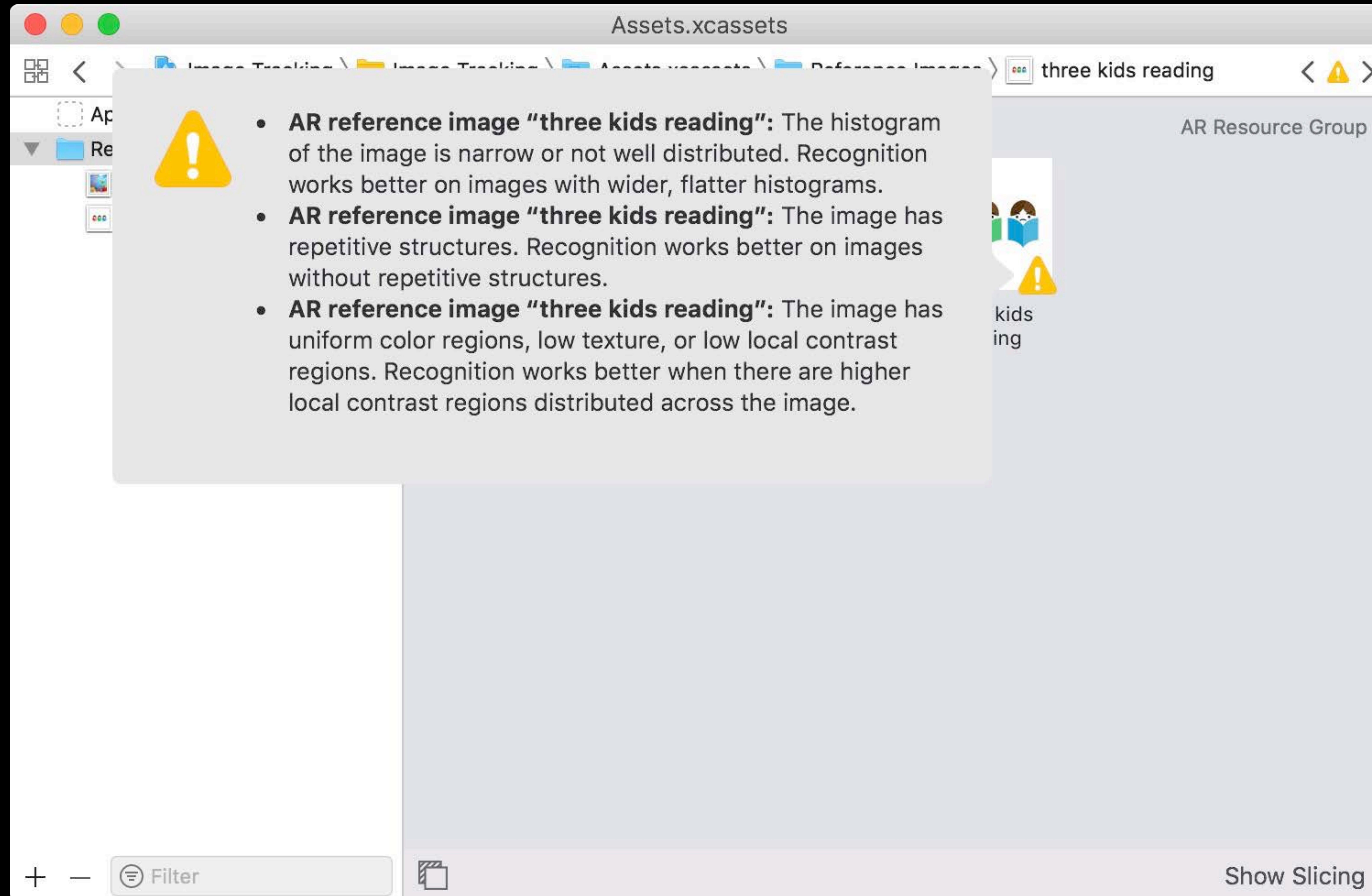
Uniform color regions

Narrow histogram

Asset Catalog in Xcode



Asset Catalog in Xcode



Configurations for Image Tracking

Configurations for Image Tracking

ARWorldTrackingConfiguration



Configurations for Image Tracking

ARWorldTrackingConfiguration

- Image anchors in world reference frame



Configurations for Image Tracking

ARWorldTrackingConfiguration

- Image anchors in world reference frame
- Detected images can be tracked NEW



Configurations for Image Tracking

ARWorldTrackingConfiguration

- Image anchors in world reference frame
- Detected images can be tracked NEW



ARImageTrackingConfiguration



Configurations for Image Tracking

ARWorldTrackingConfiguration

- Image anchors in world reference frame
- Detected images can be tracked NEW



ARImageTrackingConfiguration

- Independent from world tracking



Configurations for Image Tracking

ARWorldTrackingConfiguration

- Image anchors in world reference frame
- Detected images can be tracked NEW



ARImageTrackingConfiguration

- Independent from world tracking
- Position and orientation for every frame



```
// Image Tracking

// Create an image tracking configuration
let configuration = ARImageTrackingConfiguration()

// Set of images to be tracked
configuration.trackingImages = [catPhoto, dogPhoto, birdPhoto]

// Optionally specify the maximum number of images to track in parallel
configuration.maximumNumberOfTrackedImages = 2

// Run the session
session.run(configuration)
```

```
// Image Tracking

// Create an image tracking configuration
let configuration = ARImageTrackingConfiguration()

// Set of images to be tracked
configuration.trackingImages = [catPhoto, dogPhoto, birdPhoto]

// Optionally specify the maximum number of images to track in parallel
configuration.maximumNumberOfTrackedImages = 2

// Run the session
session.run(configuration)
```

```
// Image Tracking

// Create an image tracking configuration
let configuration = ARImageTrackingConfiguration()

// Set of images to be tracked
configuration.trackingImages = [catPhoto, dogPhoto, birdPhoto]

// Optionally specify the maximum number of images to track in parallel
configuration.maximumNumberOfTrackedImages = 2

// Run the session
session.run(configuration)
```

```
// Image Tracking

// Create an image tracking configuration
let configuration = ARImageTrackingConfiguration()

// Set of images to be tracked
configuration.trackingImages = [catPhoto, dogPhoto, birdPhoto]

// Optionally specify the maximum number of images to track in parallel
configuration.maximumNumberOfTrackedImages = 2

// Run the session
session.run(configuration)
```

```
// Image Tracking

// Create an image tracking configuration
let configuration = ARImageTrackingConfiguration()

// Set of images to be tracked
configuration.trackingImages = [catPhoto, dogPhoto, birdPhoto]

// Optionally specify the maximum number of images to track in parallel
configuration.maximumNumberOfTrackedImages = 2

// Run the session
session.run(configuration)
```



```
// Image Tracking
```

```
// Create a world tracking configuration
```

```
let configuration = ARWorldTrackingConfiguration()
```

```
// Set of images to be detected
```

```
configuration.detectionImages = [catPhoto, dogPhoto, birdPhoto]
```

```
// Optionally specify the maximum number of images to track in parallel
```

```
configuration.maximumNumberOfTrackedImages = 2
```

```
// Run the session
```

```
session.run(configuration)
```



NEW

Demo

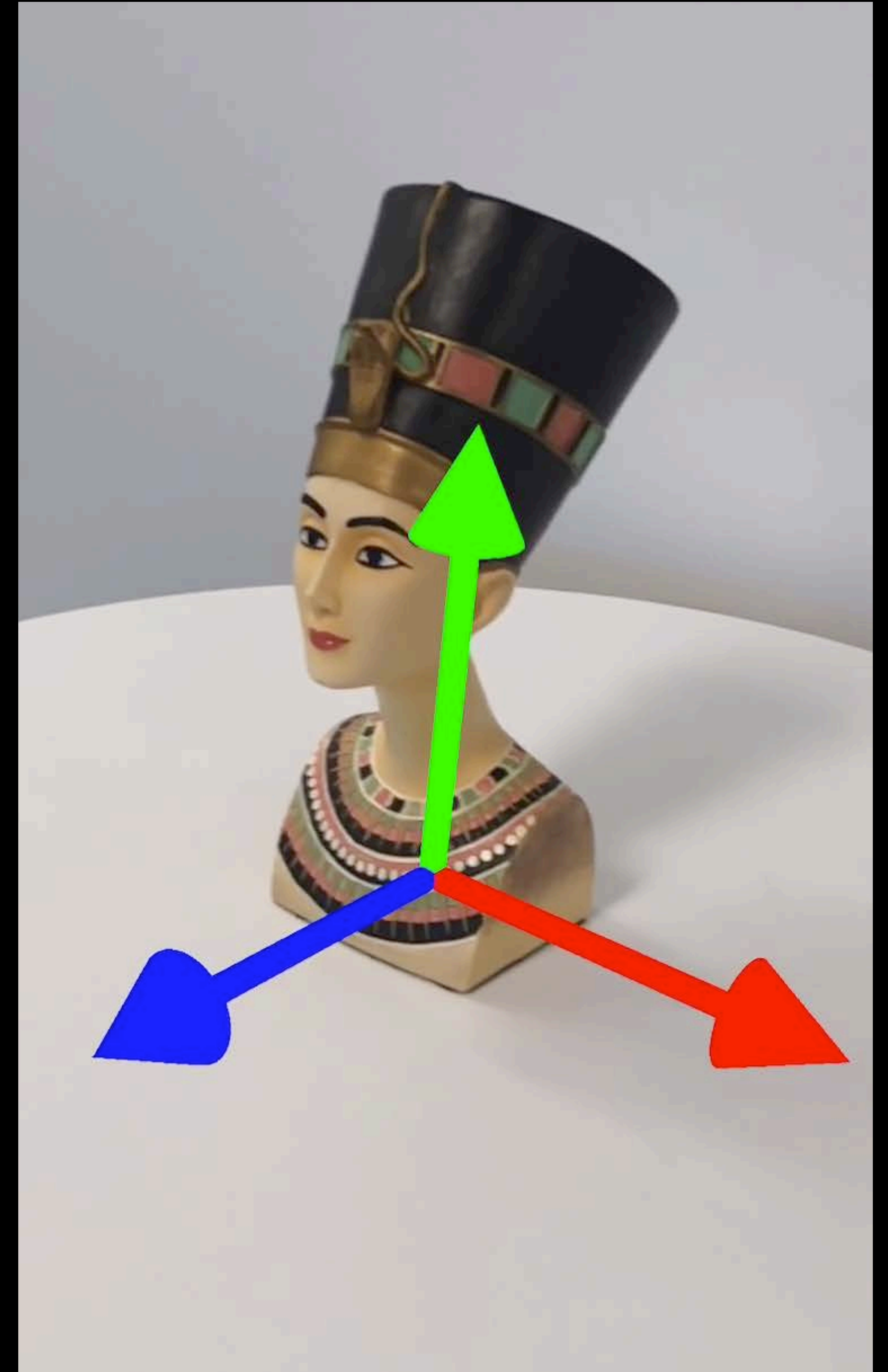
Augmented reality photo frame

Reinhard Klapfer, ARKit Engineer

NEW

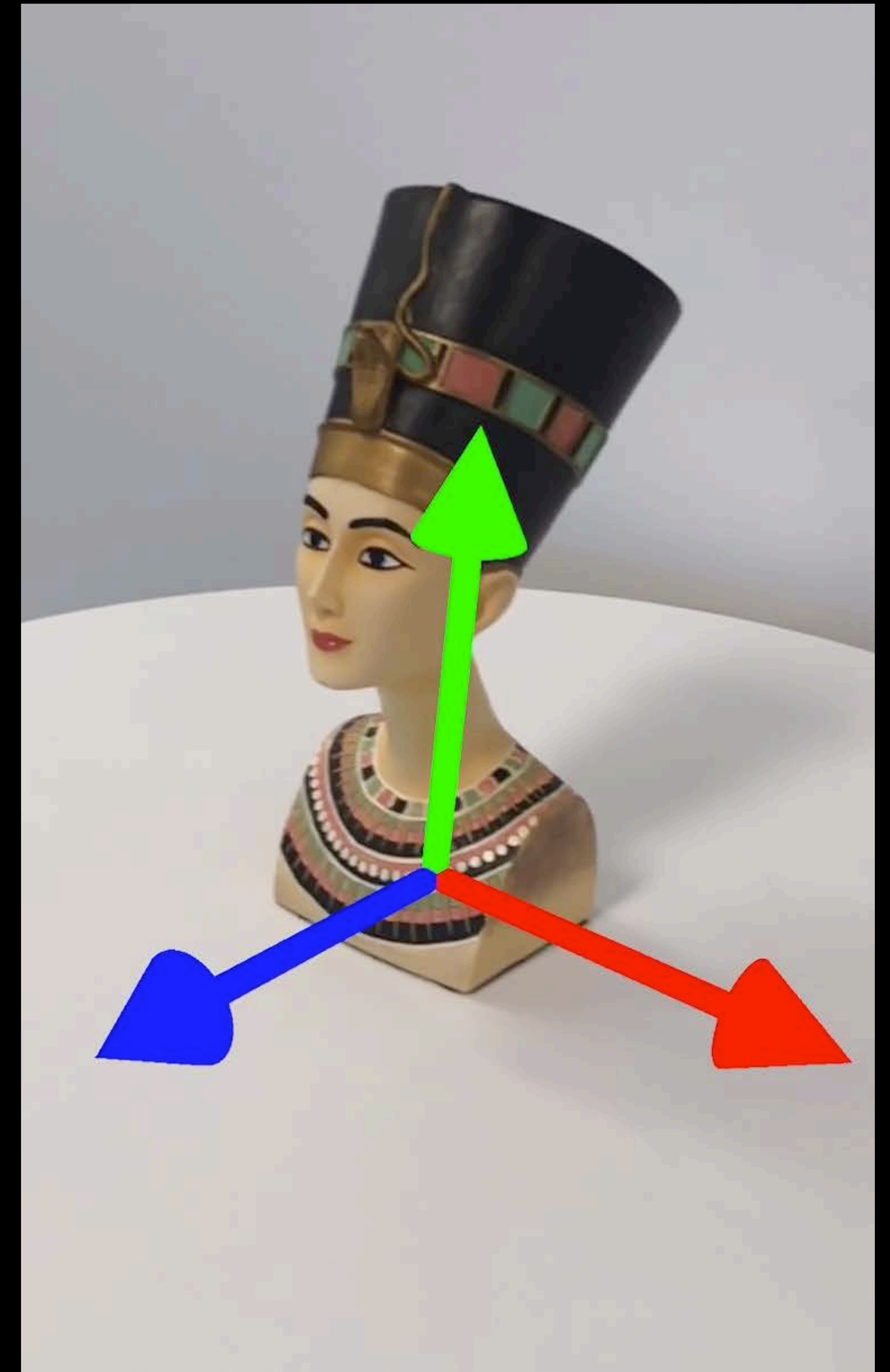
Object Detection

Object Detection



Object Detection

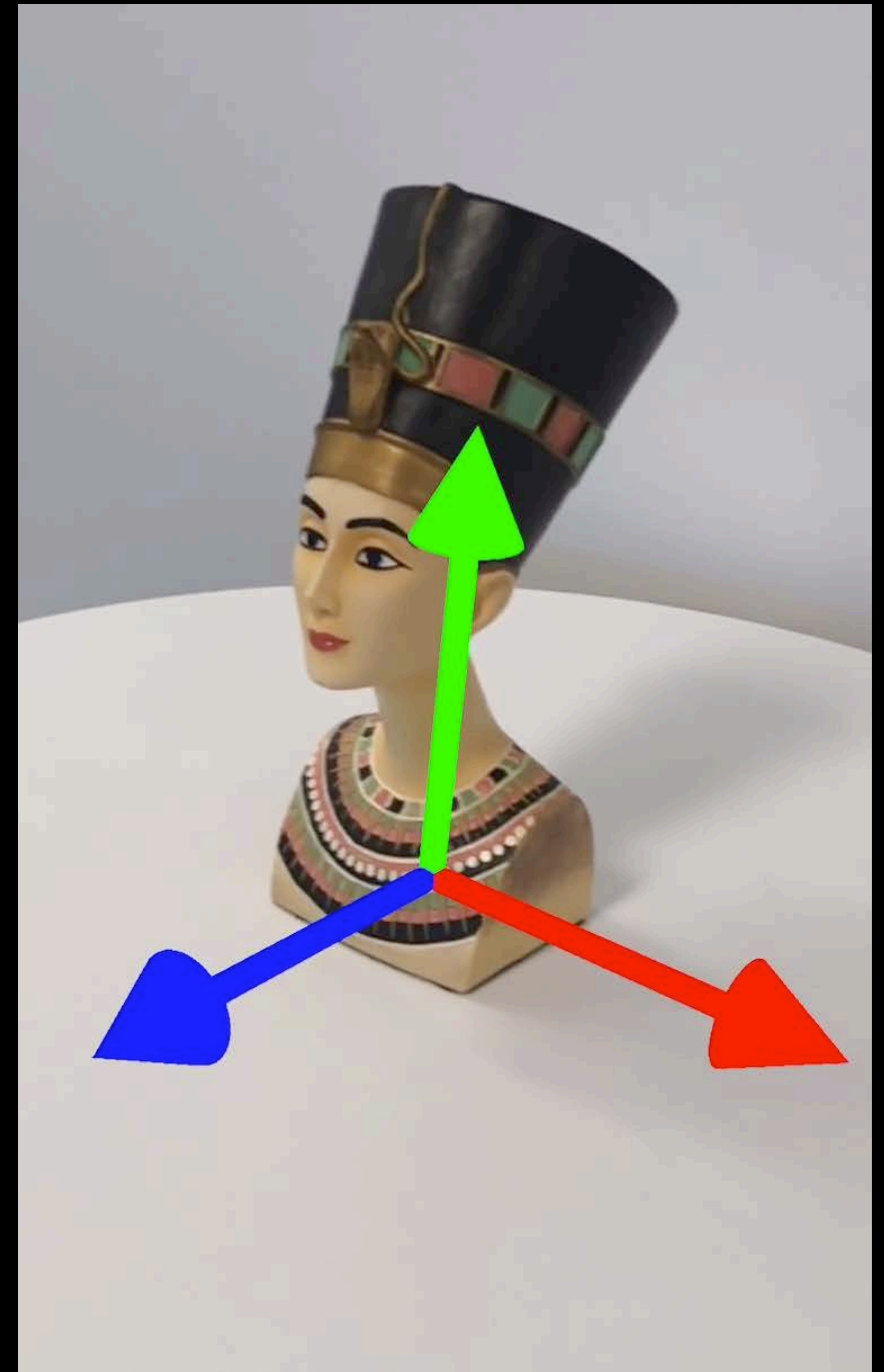
Detection of a known static 3D object



Object Detection

Detection of a known static 3D object

Objects need to be scanned first

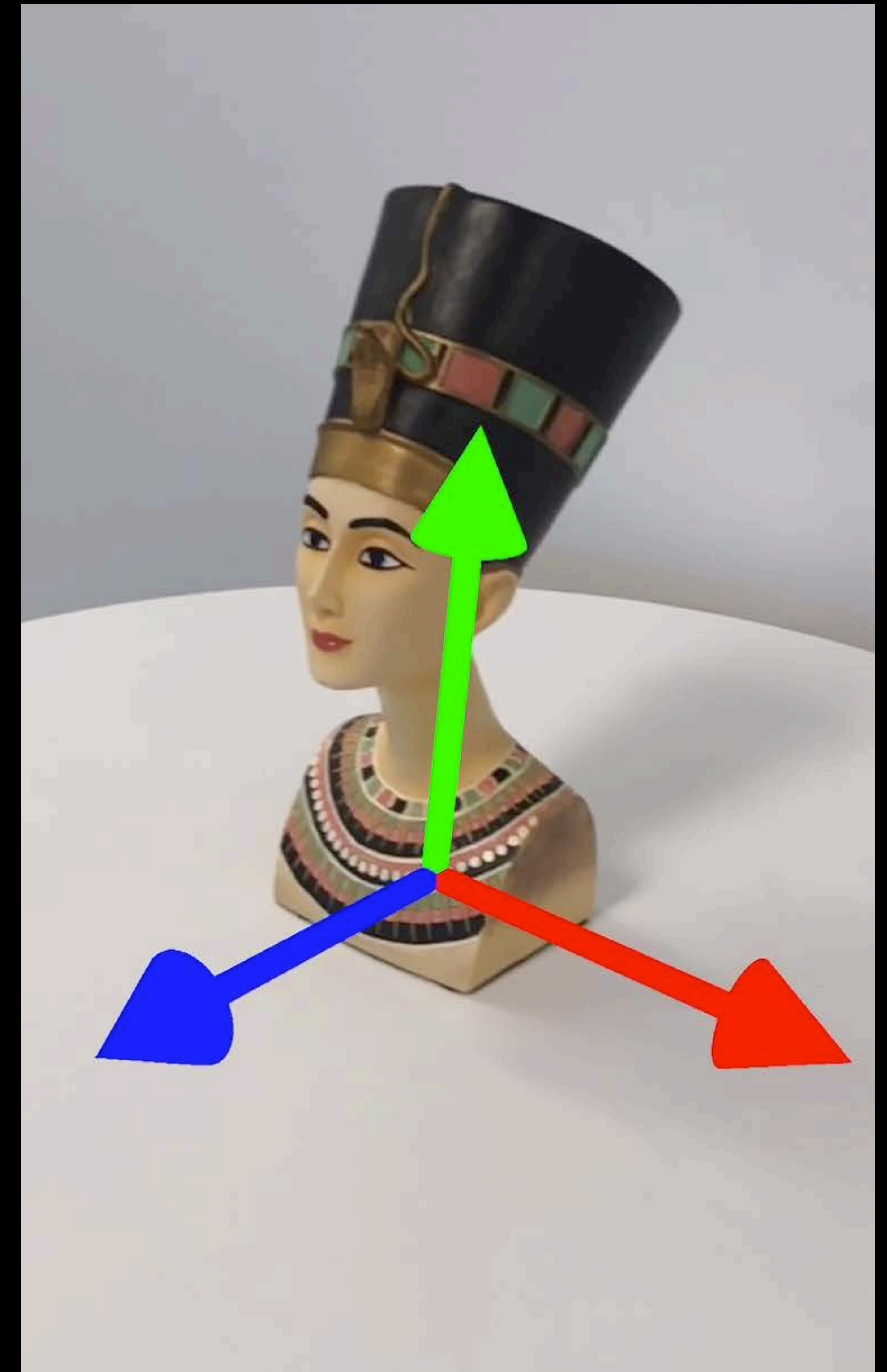


Object Detection

Detection of a known static 3D object

Objects need to be scanned first

Well-textured, rigid, non-reflective



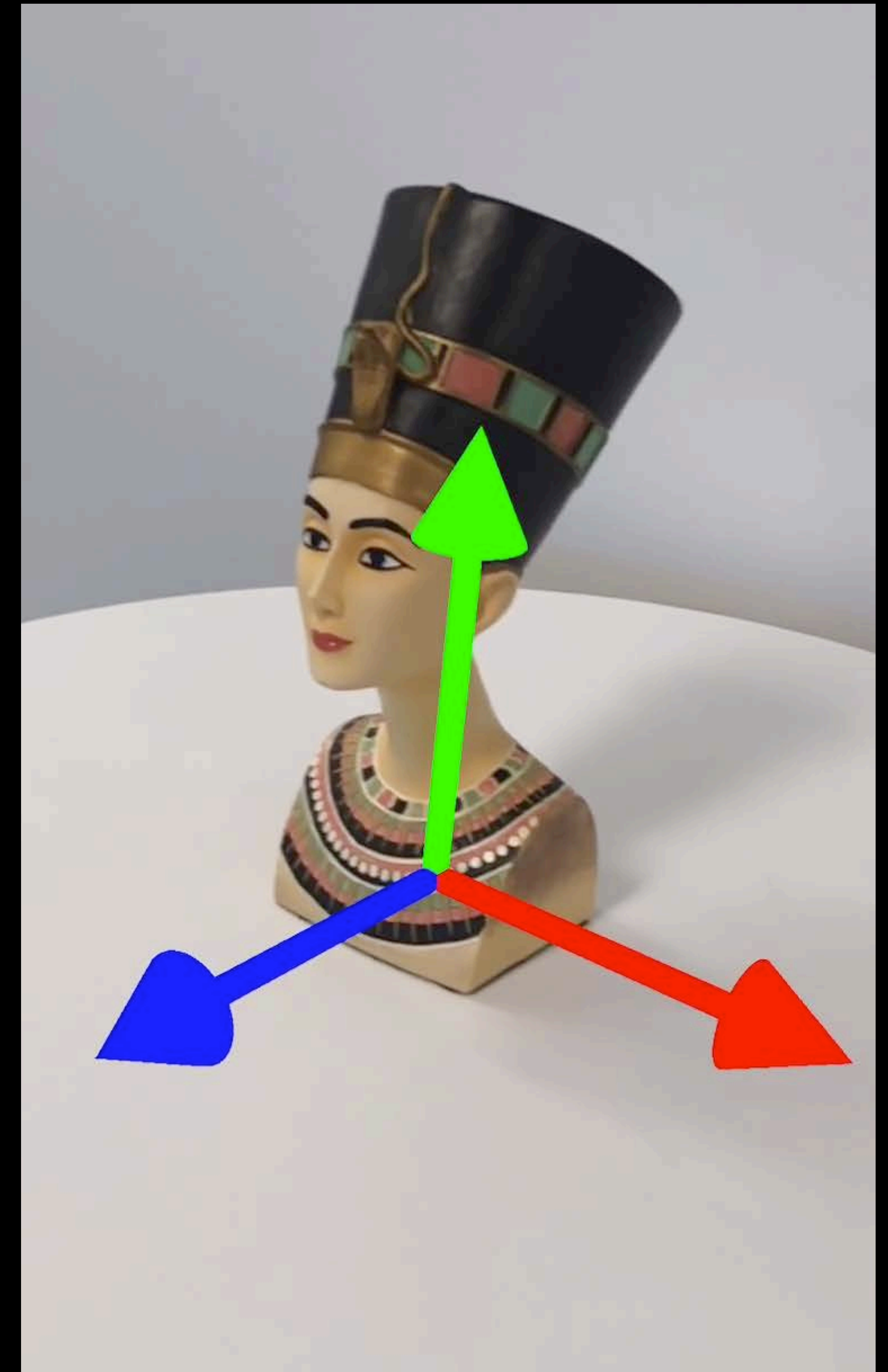
Object Detection

Detection of a known static 3D object

Objects need to be scanned first

Well-textured, rigid, non-reflective

Position and orientation



Object Detection

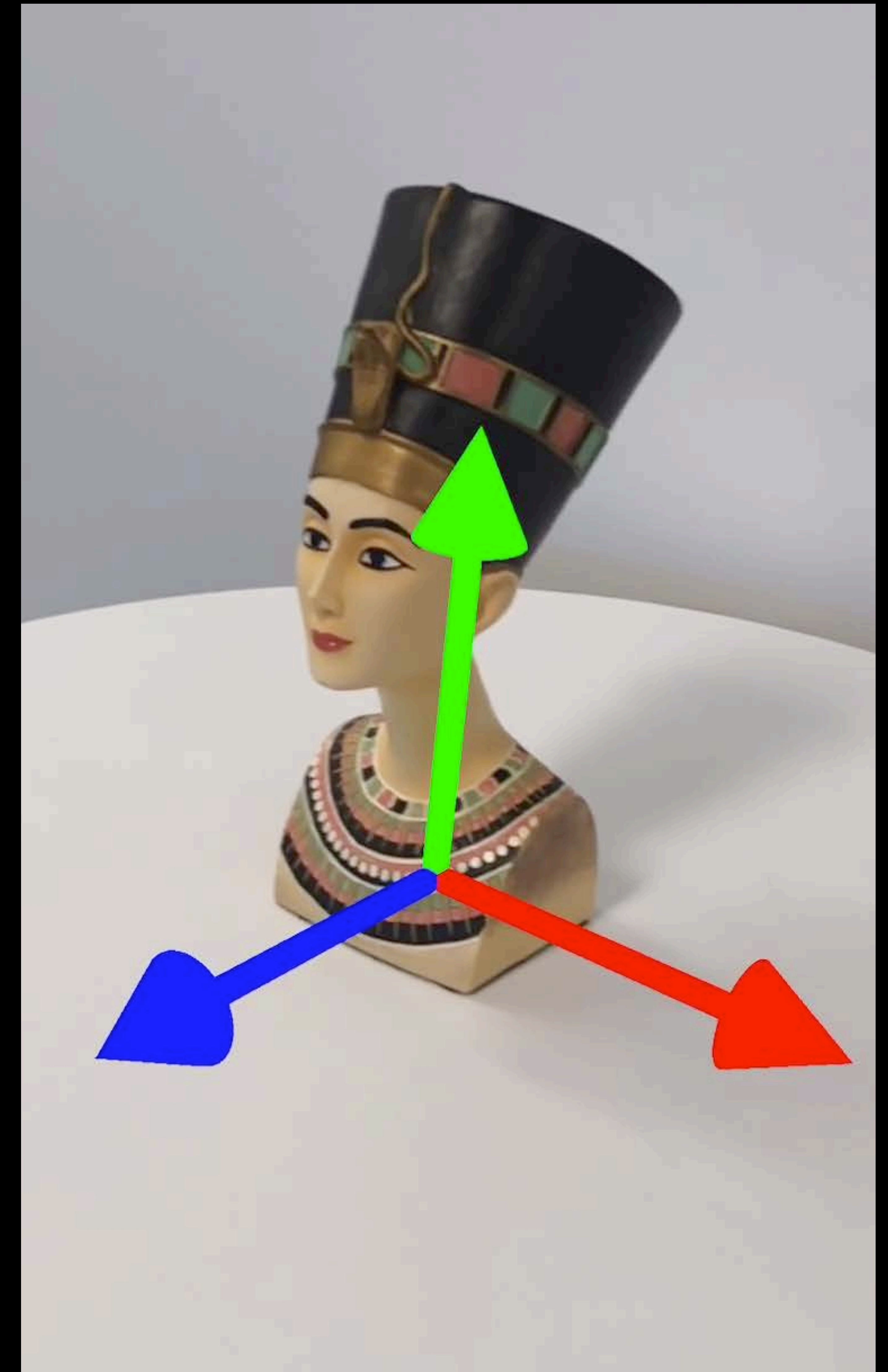
Detection of a known static 3D object

Objects need to be scanned first

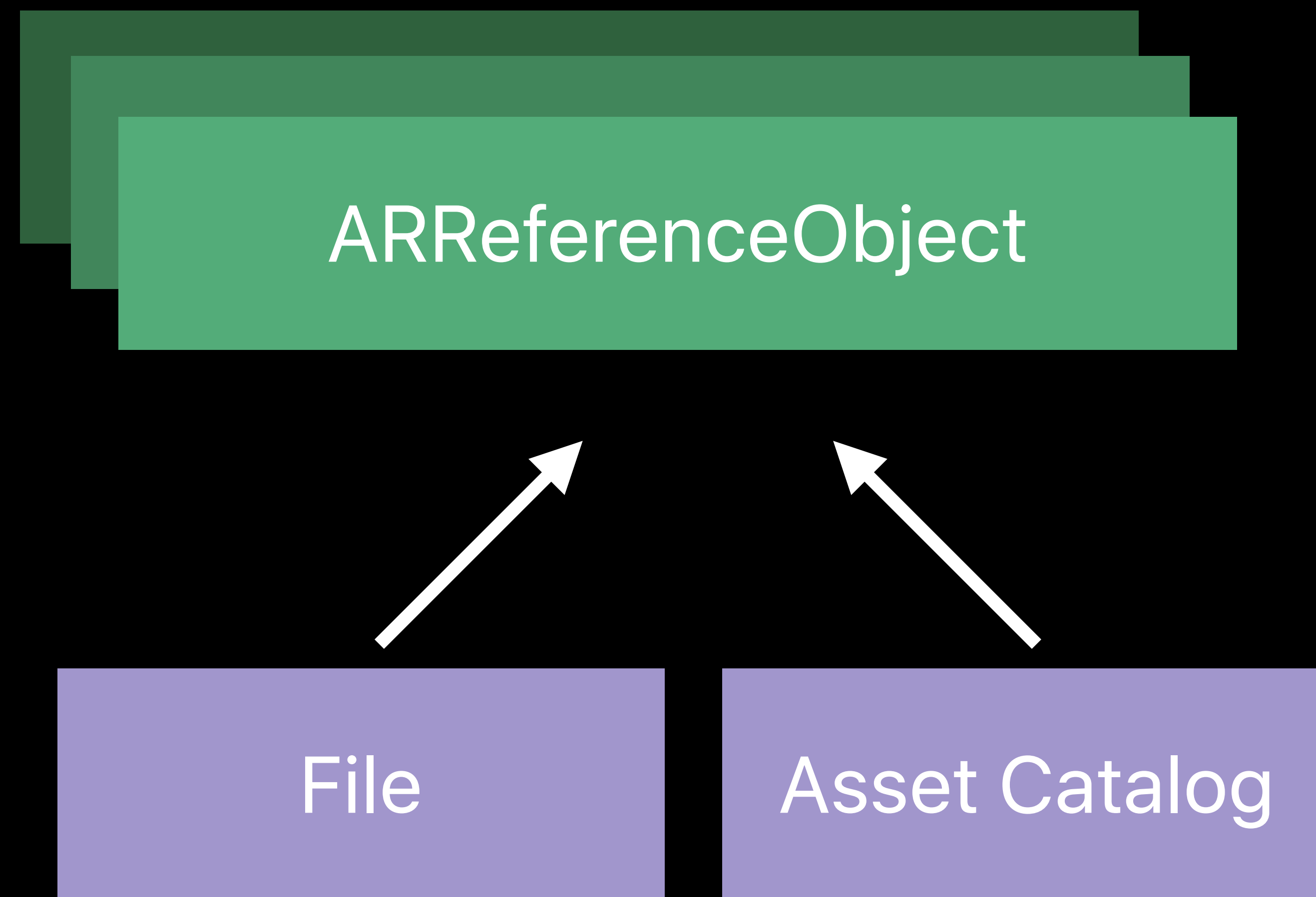
Well-textured, rigid, non-reflective

Position and orientation

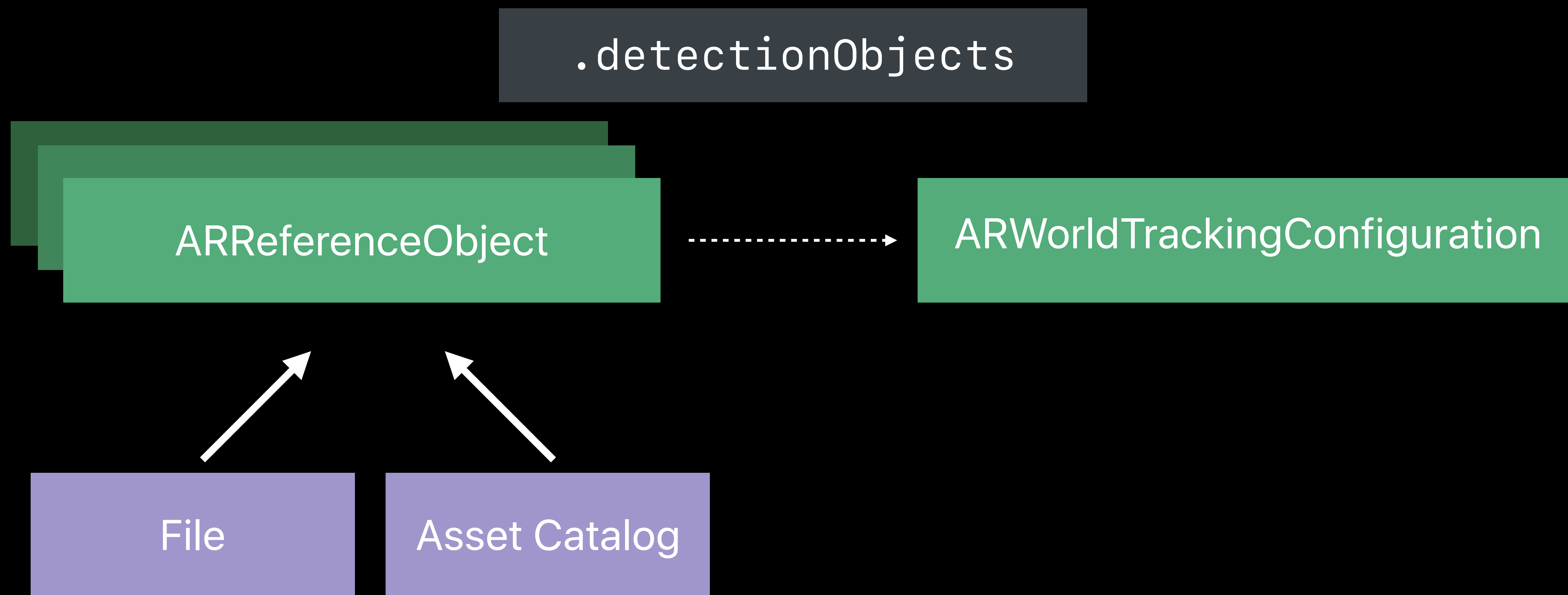
Integrated into world tracking



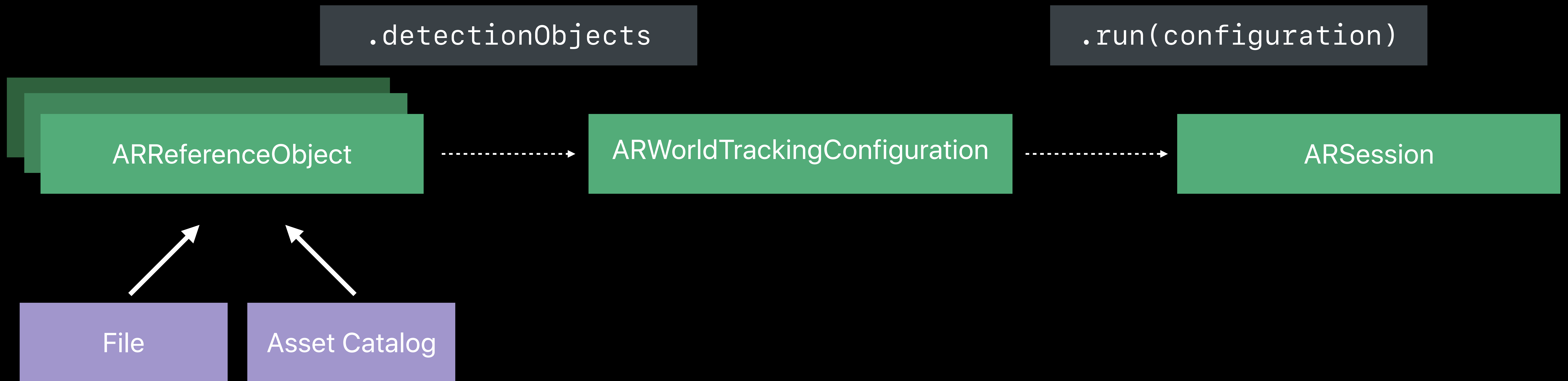
Object Detection



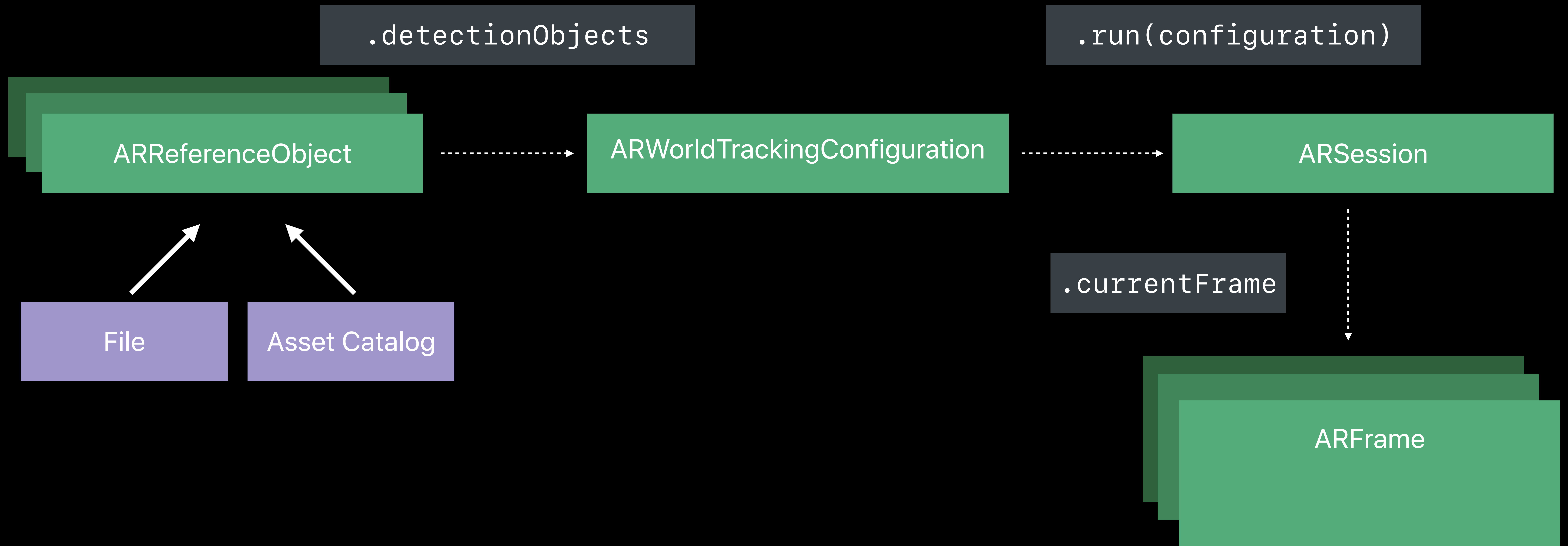
Object Detection



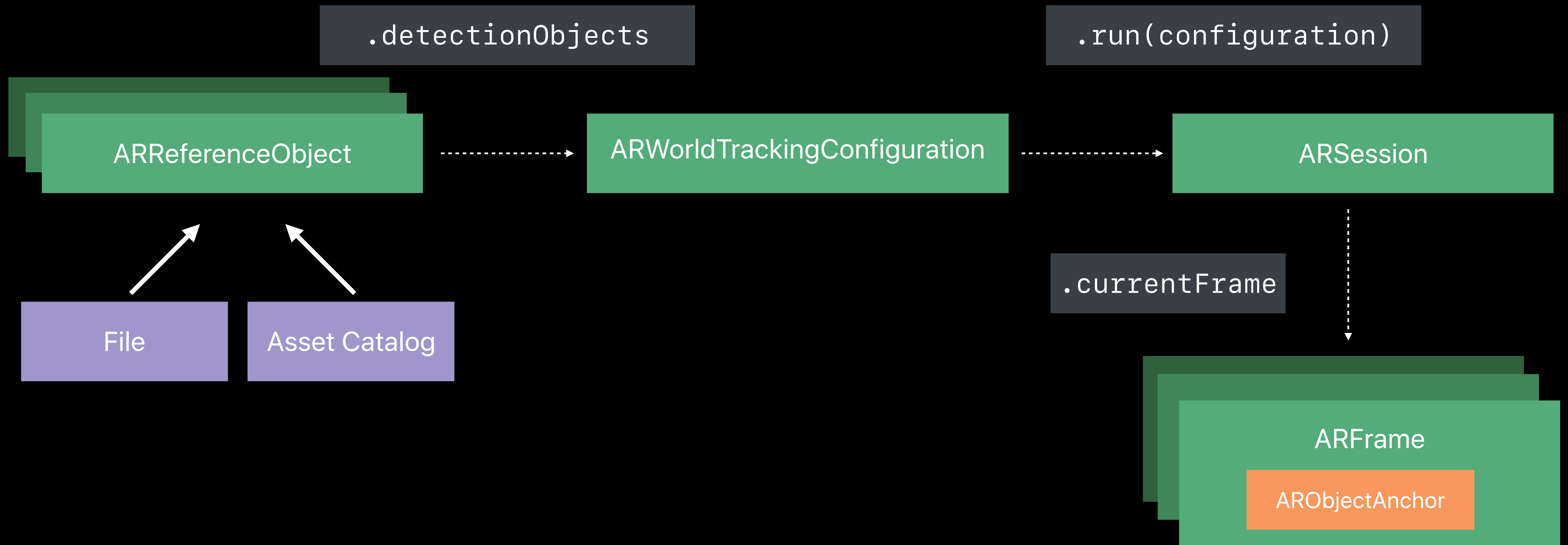
Object Detection



Object Detection

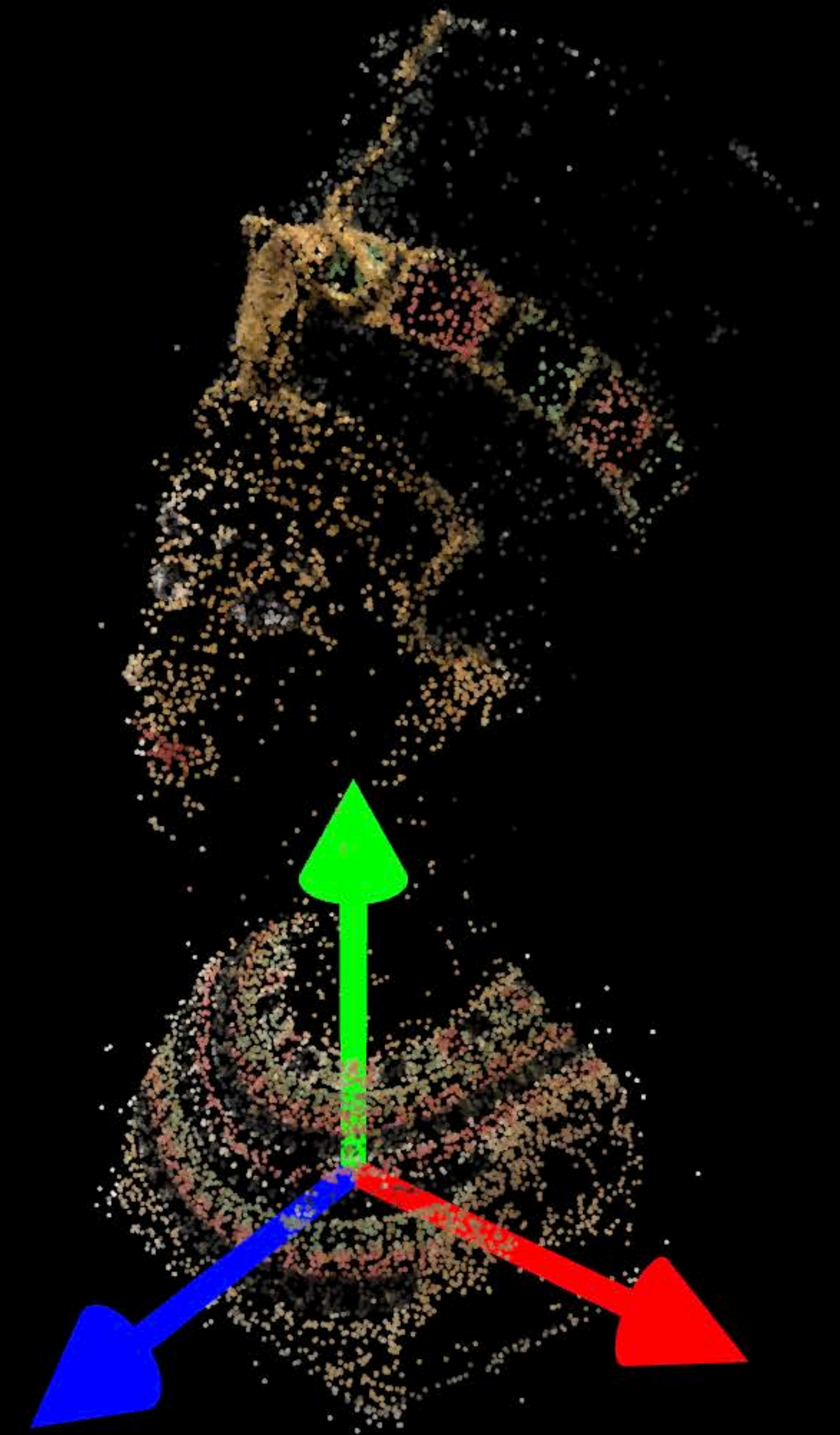


Object Detection



ARObjectAnchor

```
open class ARObjectAnchor : ARAnchor {  
  
    open var transform: simd_float4x4 { get }  
  
    open var referenceObject: ARReferenceObject { get }  
  
}
```



```
// Object Detection

// Create a world tracking configuration
let configuration = ARWorldTrackingConfiguration()

// Set of objects to be detected
configuration.detectionObjects = [ancientBust, clayPot]

// Run the session
session.run(configuration)
```



```
// Object Detection
```

```
// Create a world tracking configuration
```

```
let configuration = ARWorldTrackingConfiguration()
```

```
// Set of objects to be detected
```

```
configuration.detectionObjects = [ancientBust, clayPot]
```

```
// Run the session
```

```
session.run(configuration)
```

```
// Object Detection

// Create a world tracking configuration
let configuration = ARWorldTrackingConfiguration()

// Set of objects to be detected
configuration.detectionObjects = [ancientBust, clayPot]

// Run the session
session.run(configuration)
```

```
// Object Detection

// Create a world tracking configuration
let configuration = ARWorldTrackingConfiguration()

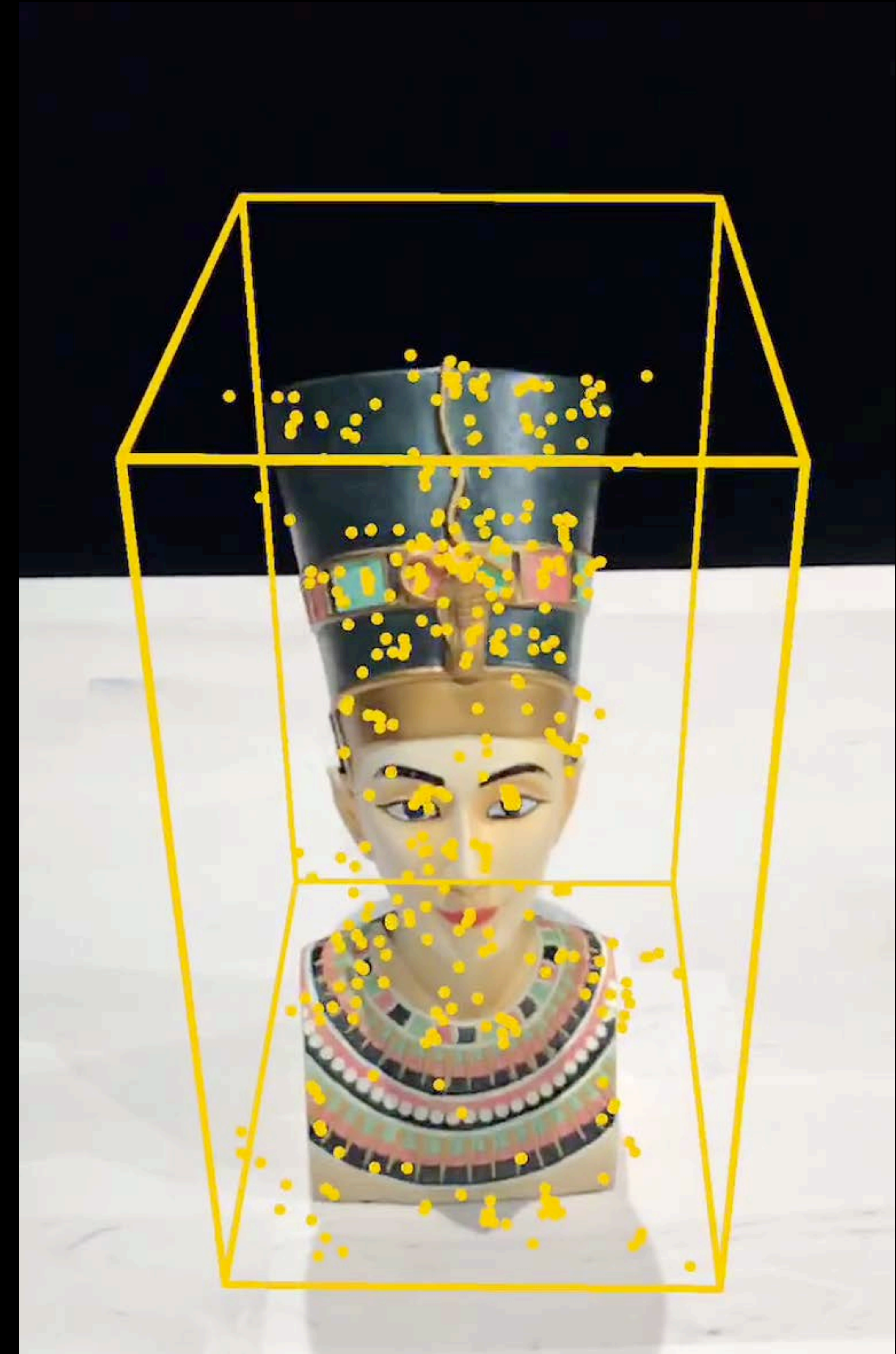
// Set of objects to be detected
configuration.detectionObjects = [ancientBust, clayPot]

// Run the session
session.run(configuration)
```



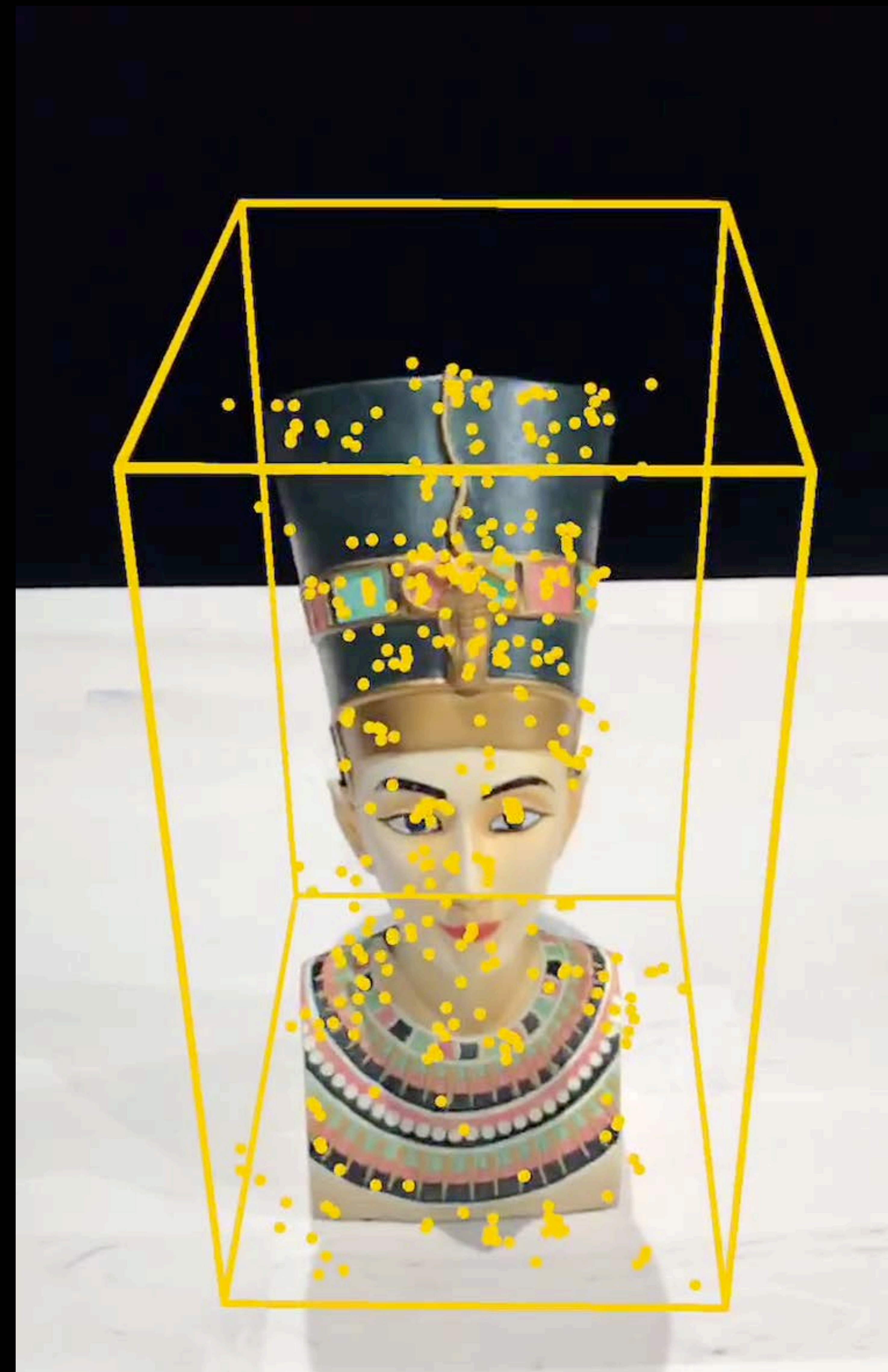


Object Scanning



Object Scanning

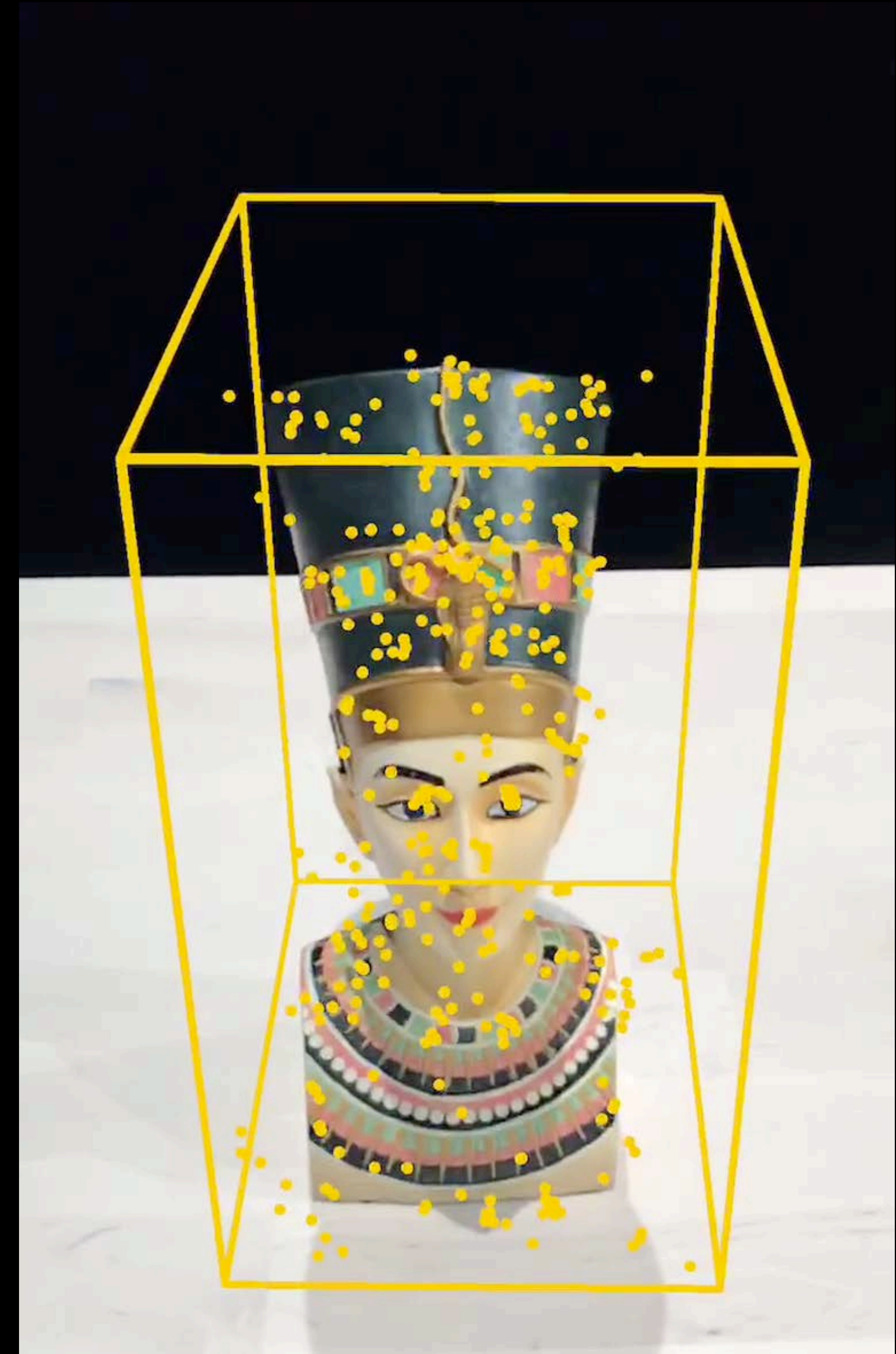
Accumulated scene information



Object Scanning

Accumulated scene information

Transform, extent, center

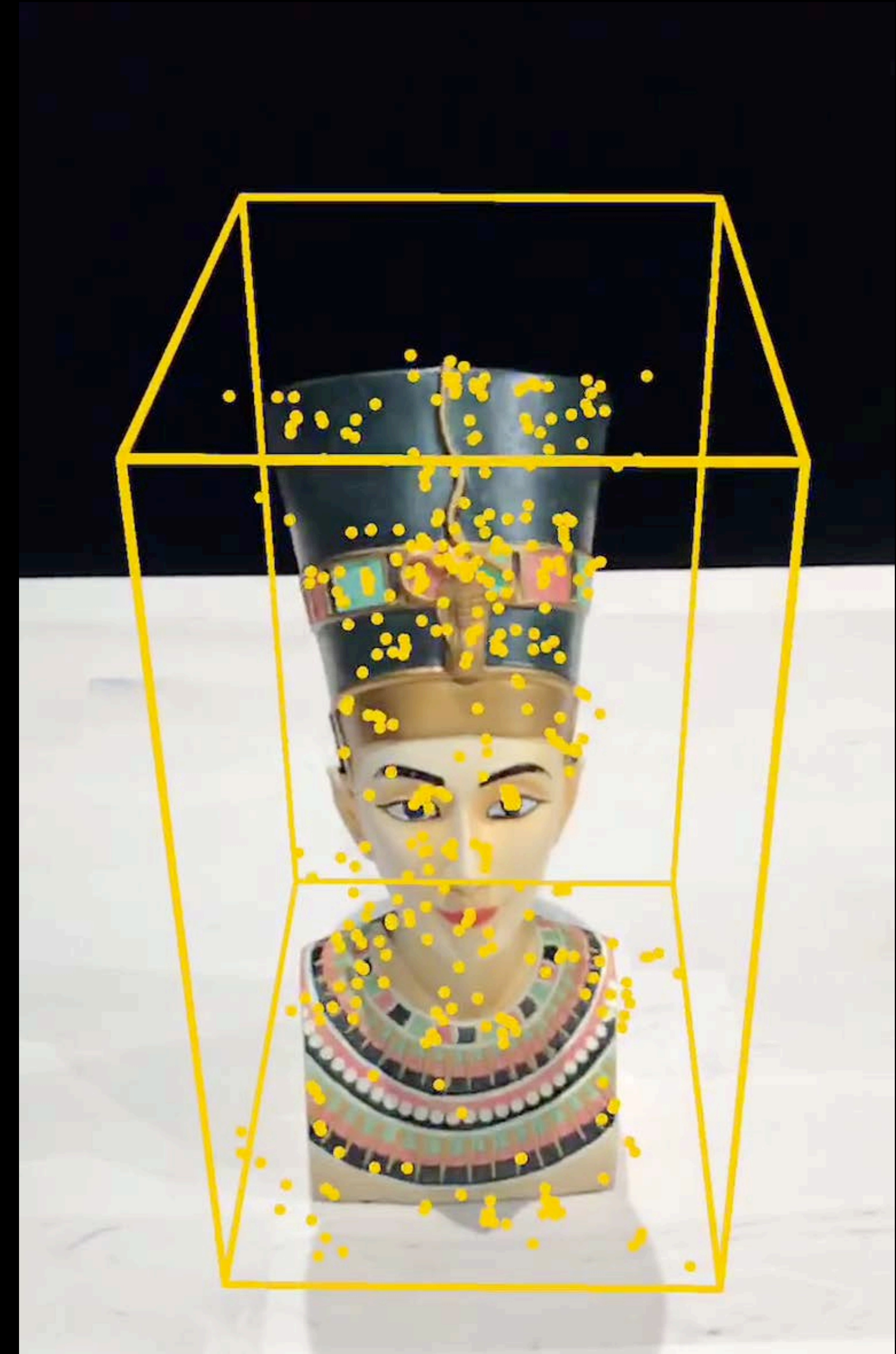


Object Scanning

Accumulated scene information

Transform, extent, center

Supported by Xcode asset catalog



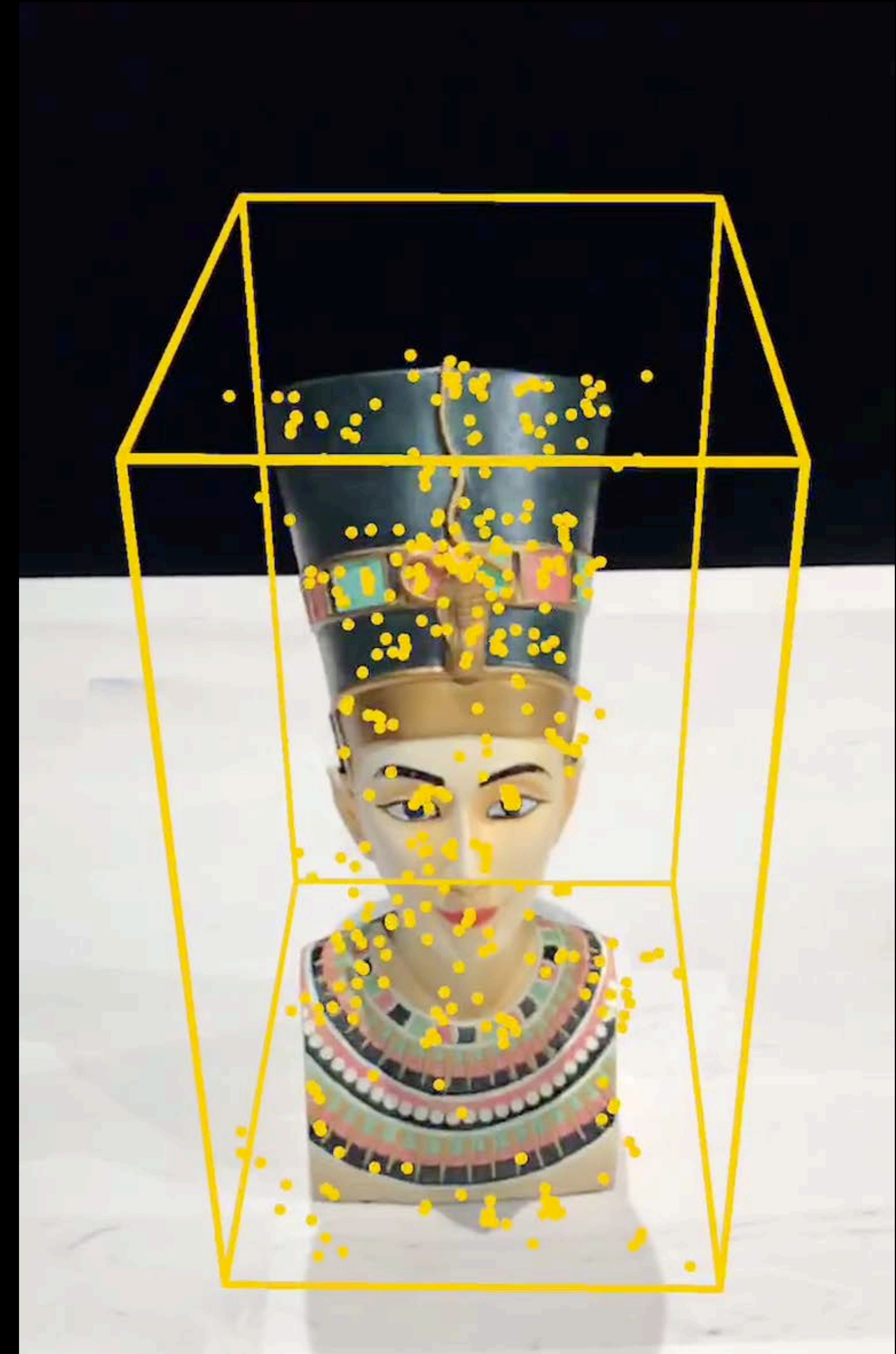
Object Scanning

Accumulated scene information

Transform, extent, center

Supported by Xcode asset catalog

`ARObjectScanningConfiguration`



Object Scanning

Accumulated scene information

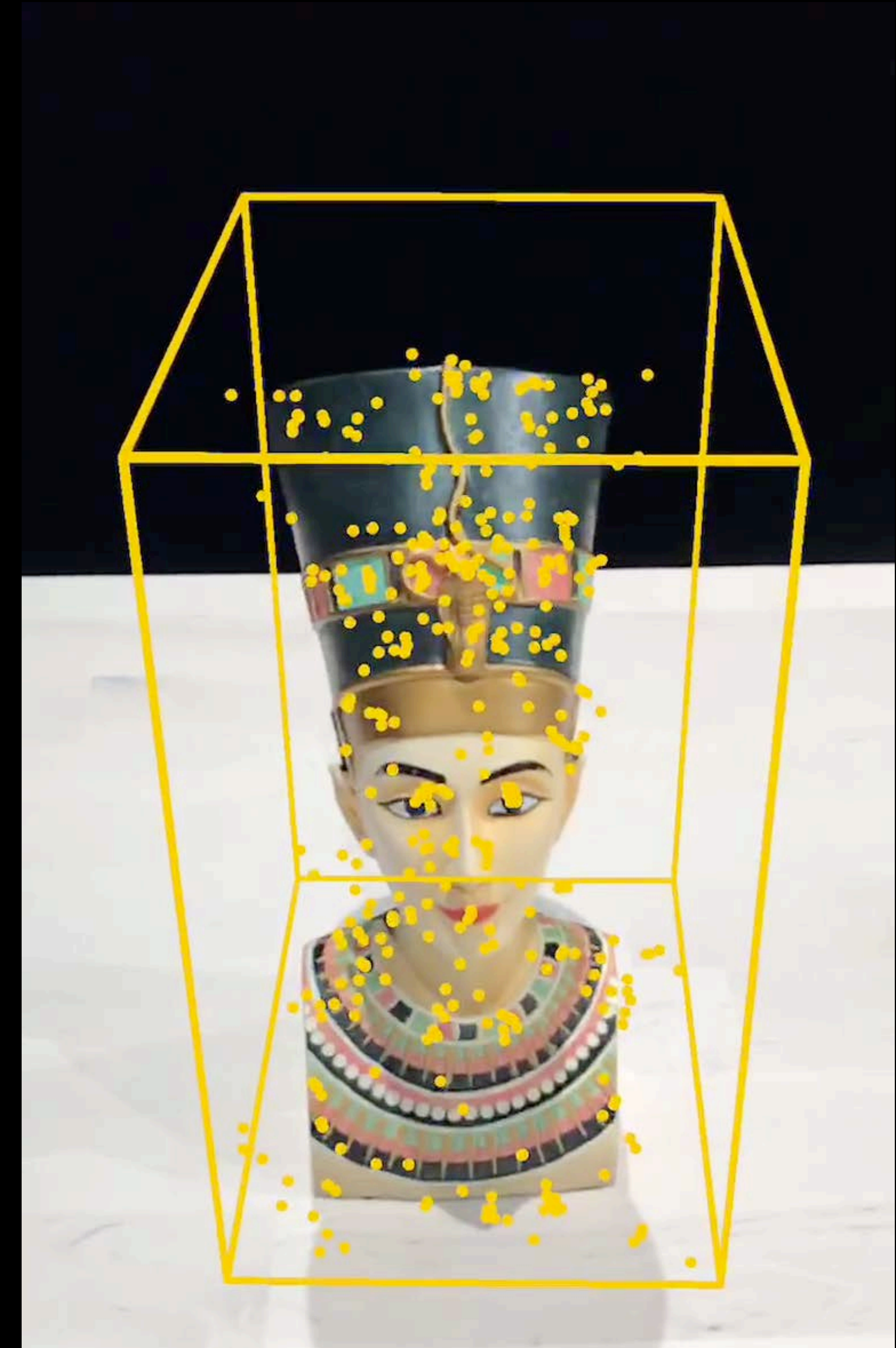
Transform, extent, center

Supported by Xcode asset catalog

`ARObjectScanningConfiguration`

Sample code available

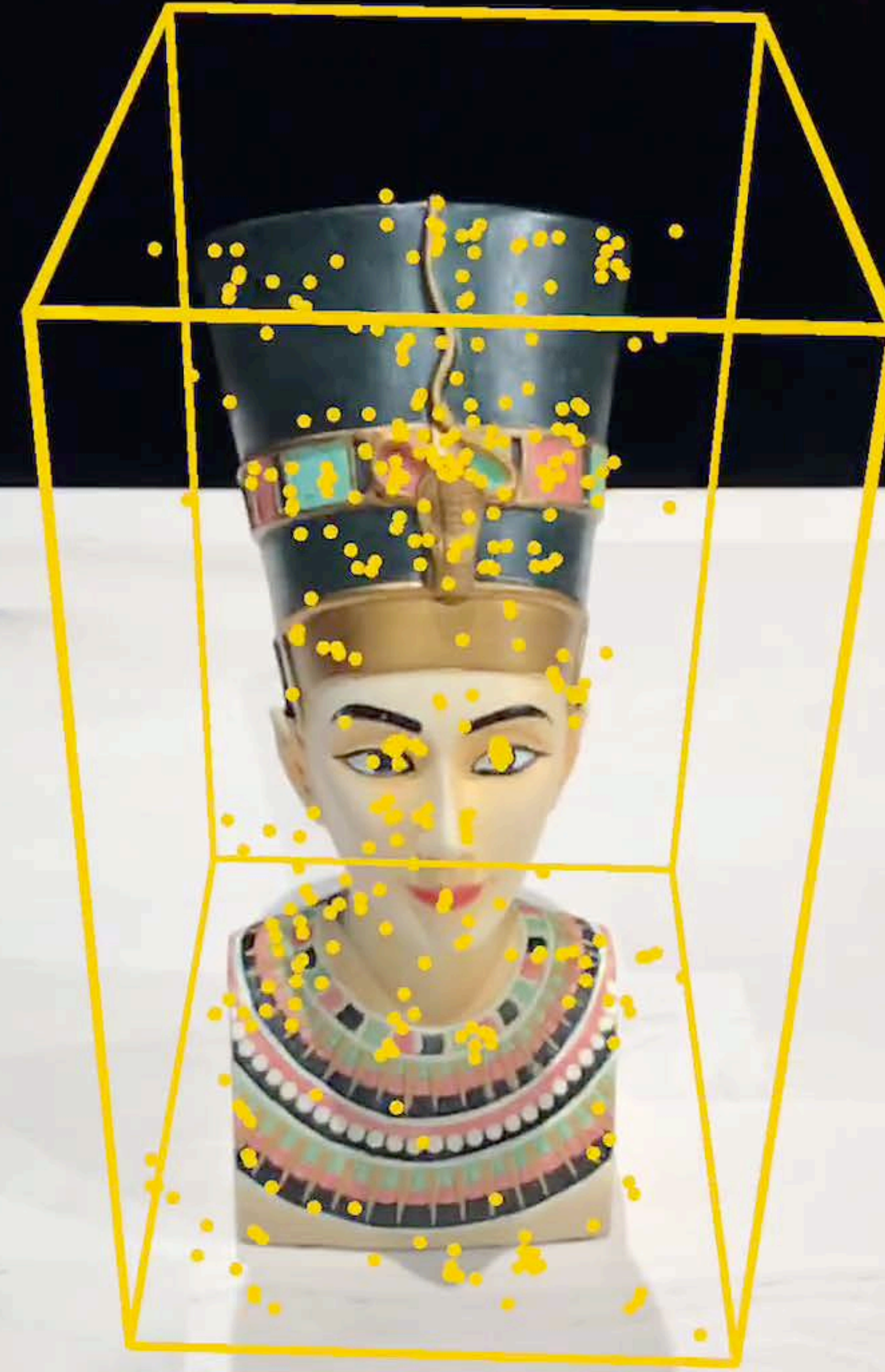
- Scanning and detecting 3D objects



Back

Define bounding box

Restart



Define bounding box of object

Tap the screen to position the bounding box. Rotate, scale or move with two fingers. Long press sides of the box to push/pull them in or out. Touch the box to move it on an axis.

When the bounding box contains the entire object, tap 'Scan'.

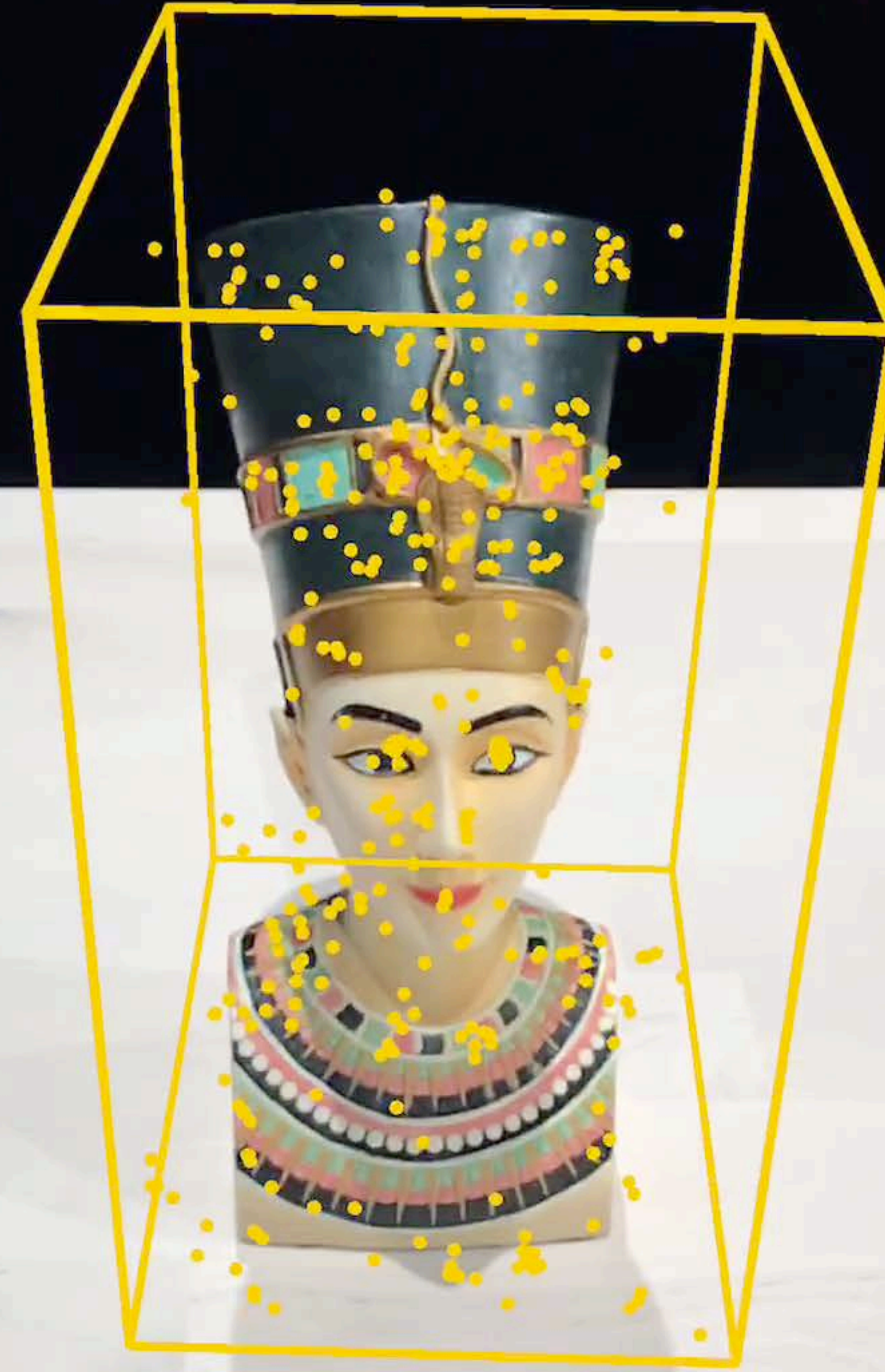
Scan

?

Back

Define bounding box

Restart



Define bounding box of object

Tap the screen to position the bounding box. Rotate, scale or move with two fingers. Long press sides of the box to push/pull them in or out. Touch the box to move it on an axis.
When the bounding box contains the entire object, tap 'Scan'.

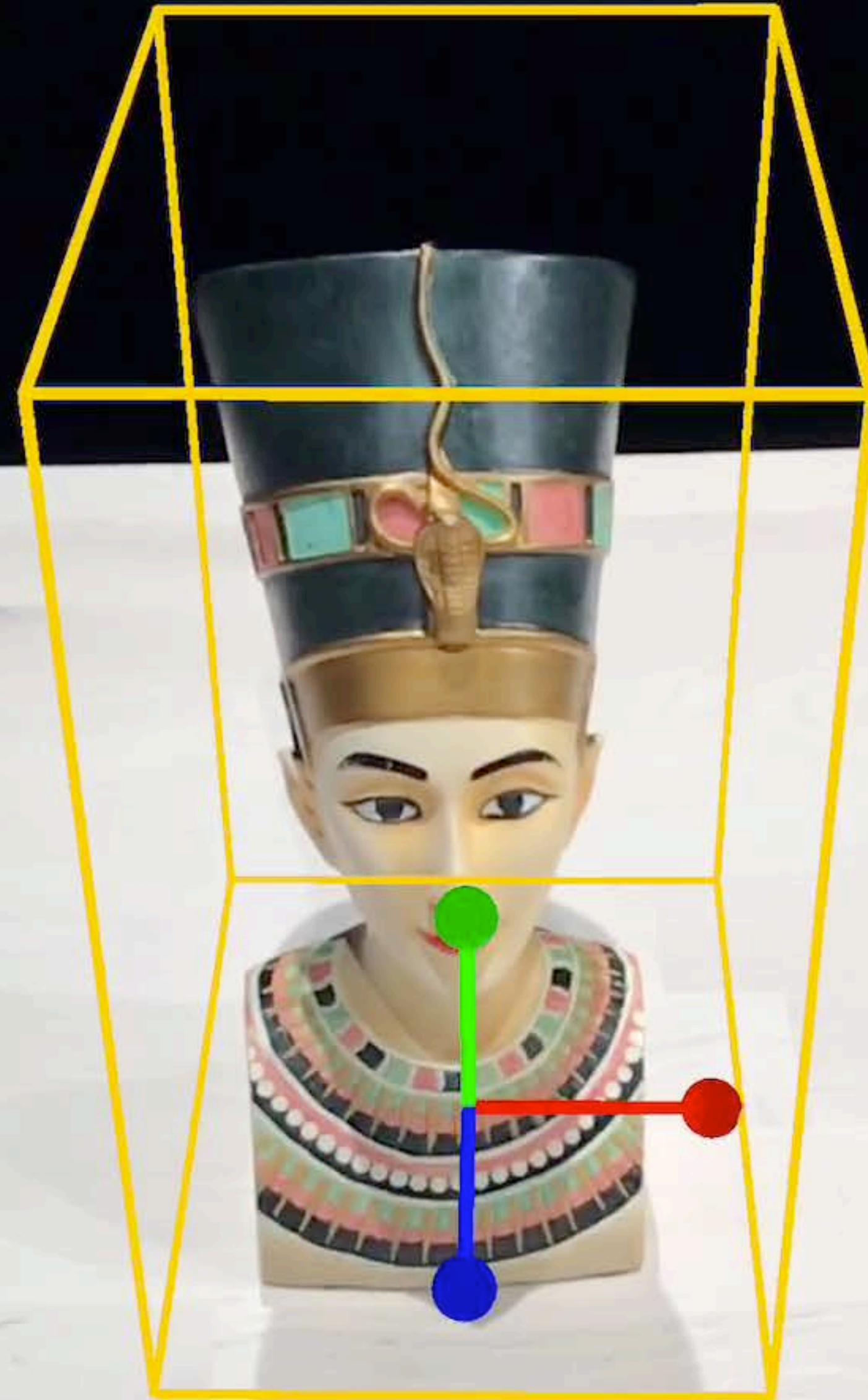
Scan

?

Back

Adjust origin

Restart



Adjust origin

Touch one of the spheres and pan to move the origin along the selected axis, or rotate and move with two fingers. When satisfied, tap 'Test'. You can load a *.usdz 3D model overlay by tapping the 'Load Model' button.

Load Model

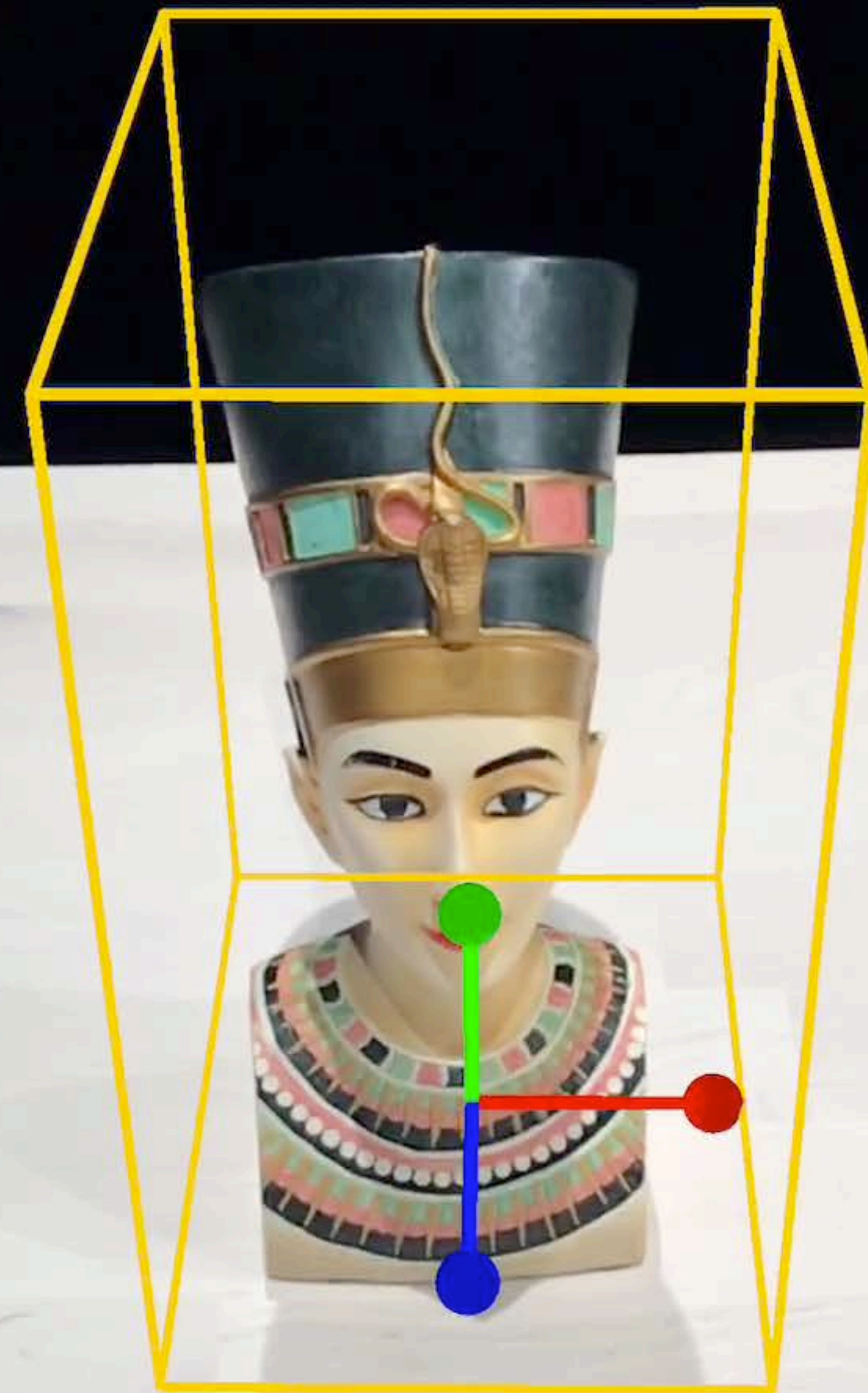
Test

?

Back

Adjust origin

Restart



Adjust origin

Touch one of the spheres and pan to move the origin along the selected axis, or rotate and move with two fingers. When satisfied, tap 'Test'. You can load a *.usdz 3D model overlay by tapping the 'Load Model' button.

Load Model

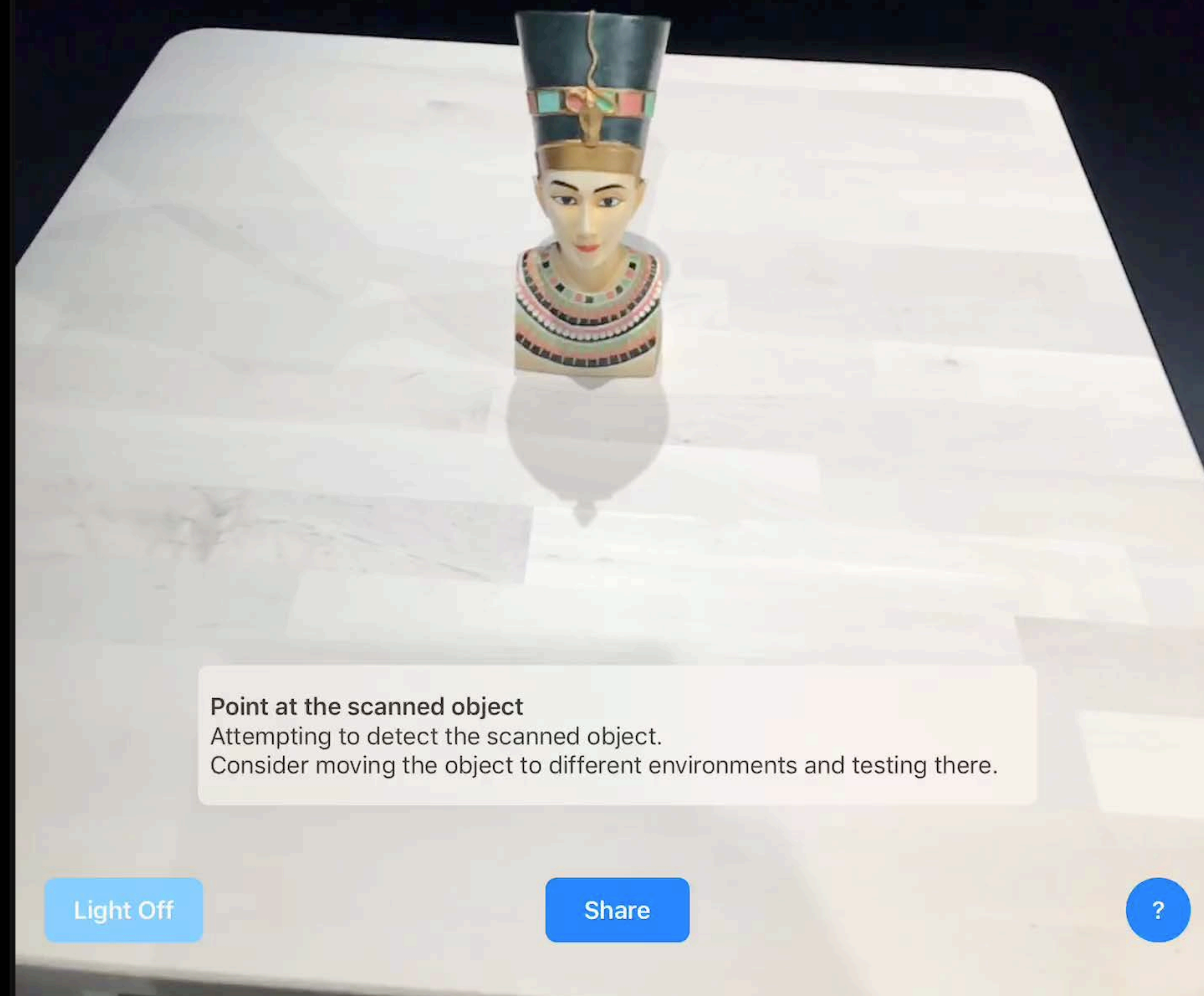
Test

?

Test

Restart

Scanning & detecting might not work:
ARKit is initializing



Point at the scanned object
Attempting to detect the scanned object.
Consider moving the object to different environments and testing there.

Light Off

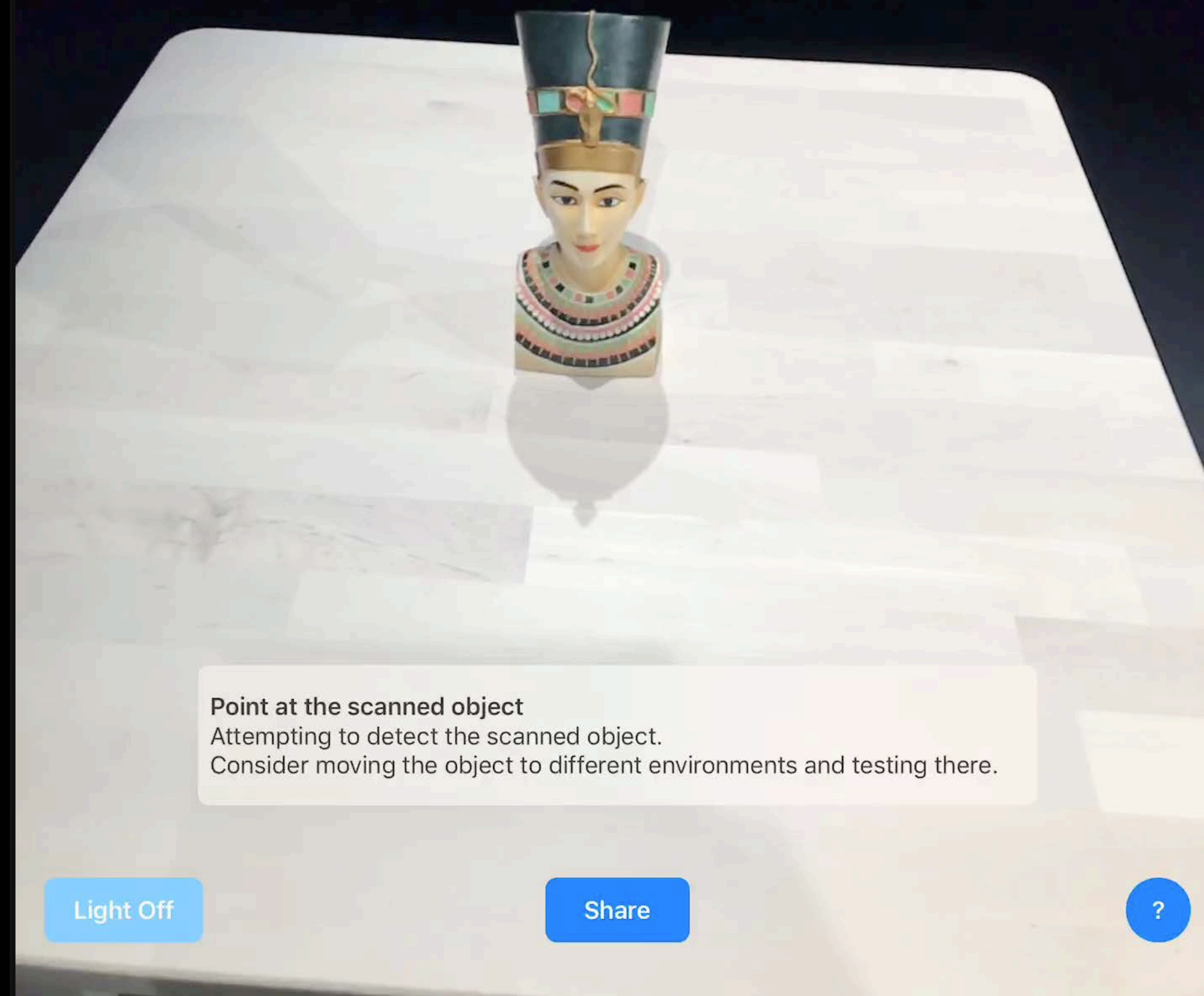
Share

?

Test

Restart

Scanning & detecting might not work:
ARKit is initializing



Point at the scanned object
Attempting to detect the scanned object.
Consider moving the object to different environments and testing there.

Light Off

Share

?

Test

Restart



Point at the scanned object
Attempting to detect the scanned object.
Consider moving the object to different environments and testing there.

Light Off

Share

?

Test

Restart



Point at the scanned object
Attempting to detect the scanned object.
Consider moving the object to different environments and testing there.

Light Off

Share

?

ARReferenceObject

```
open class ARReferenceObject
  : NSObject, NSCopying, NSSecureCoding {

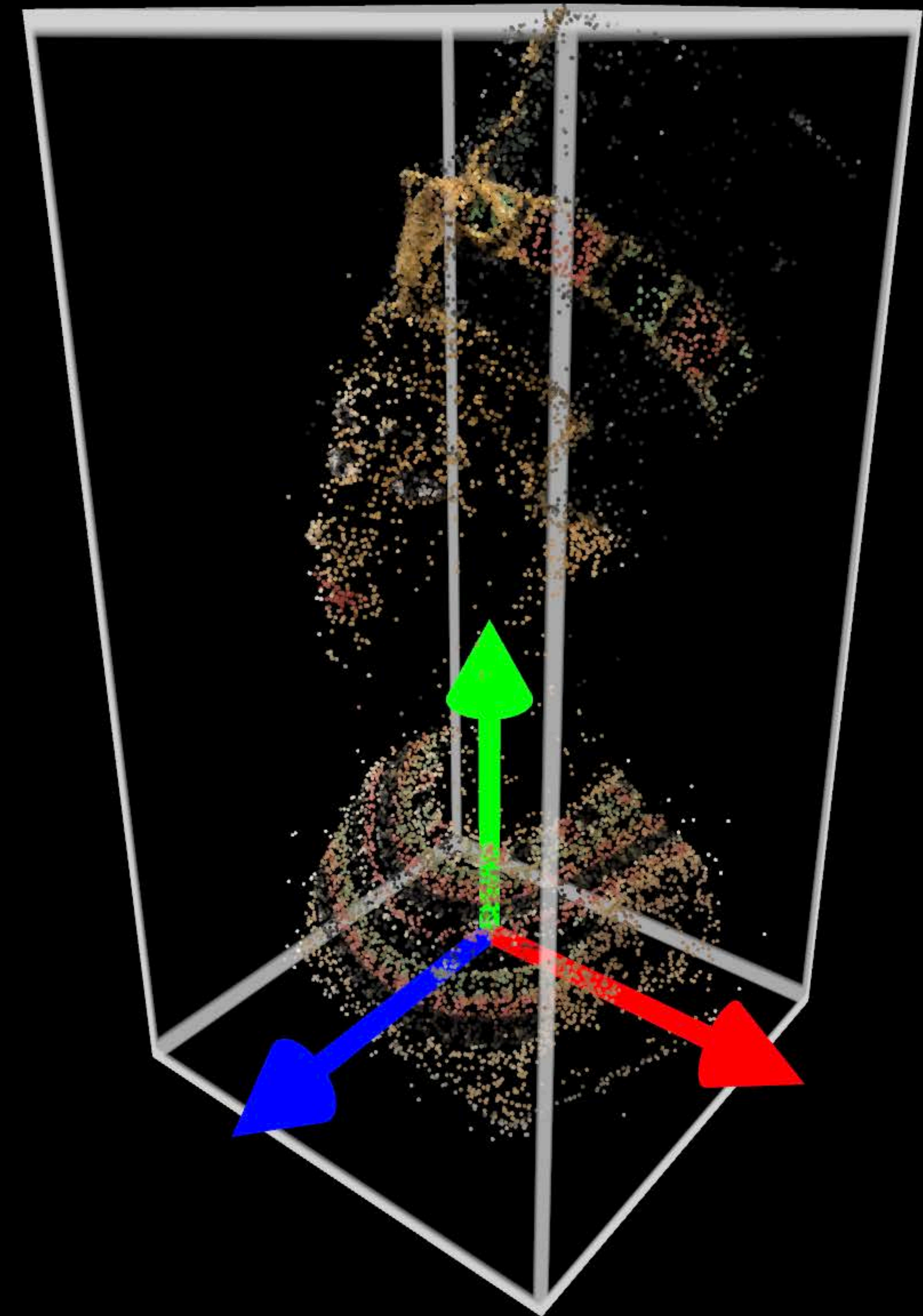
  open var name: String?

  open var center: simd_float3 { get }

  open var extent: simd_float3 { get }

  open var rawFeaturePoints: ARPointCloud { get }

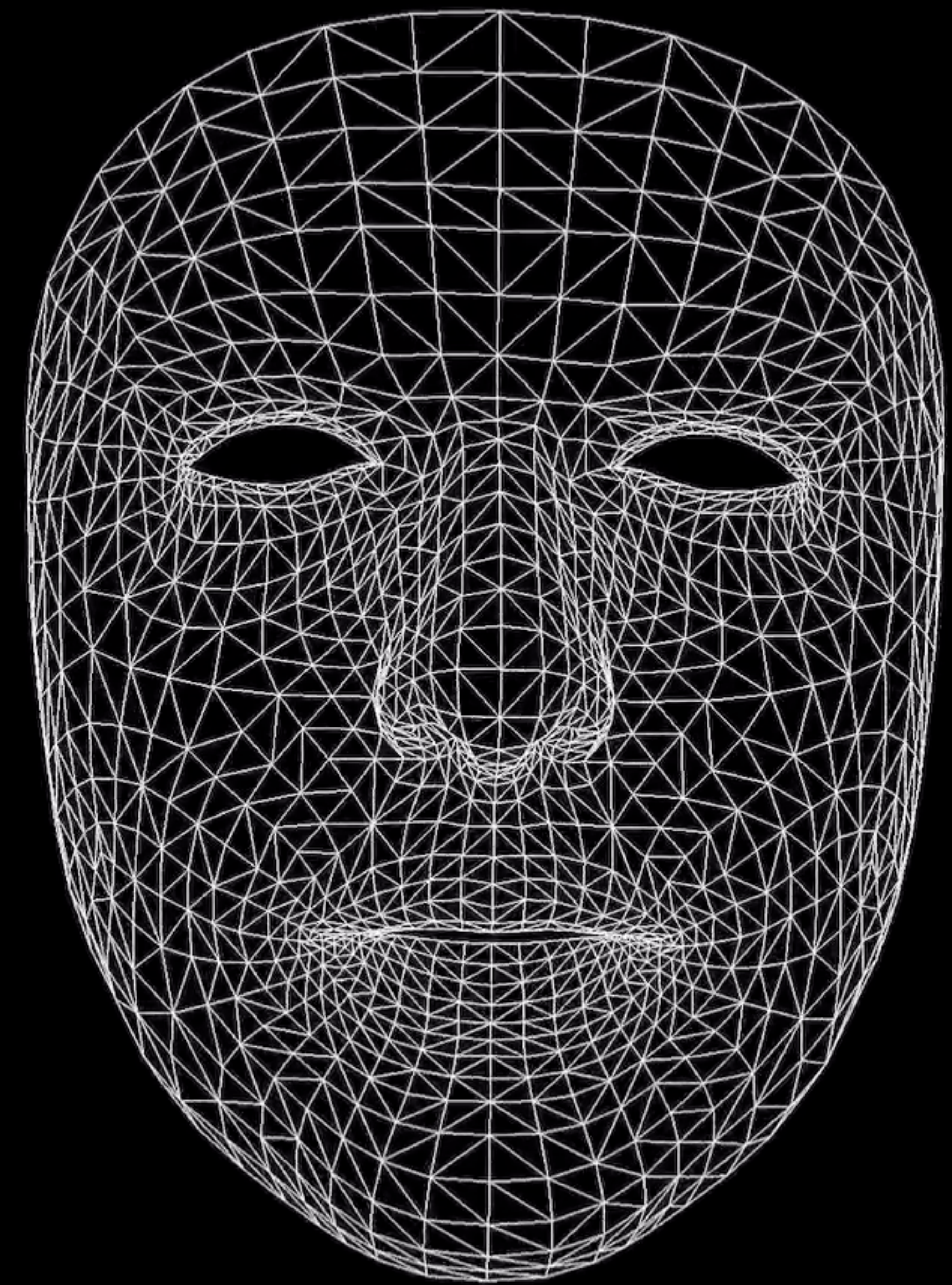
}
```



Face Tracking

Face Tracking

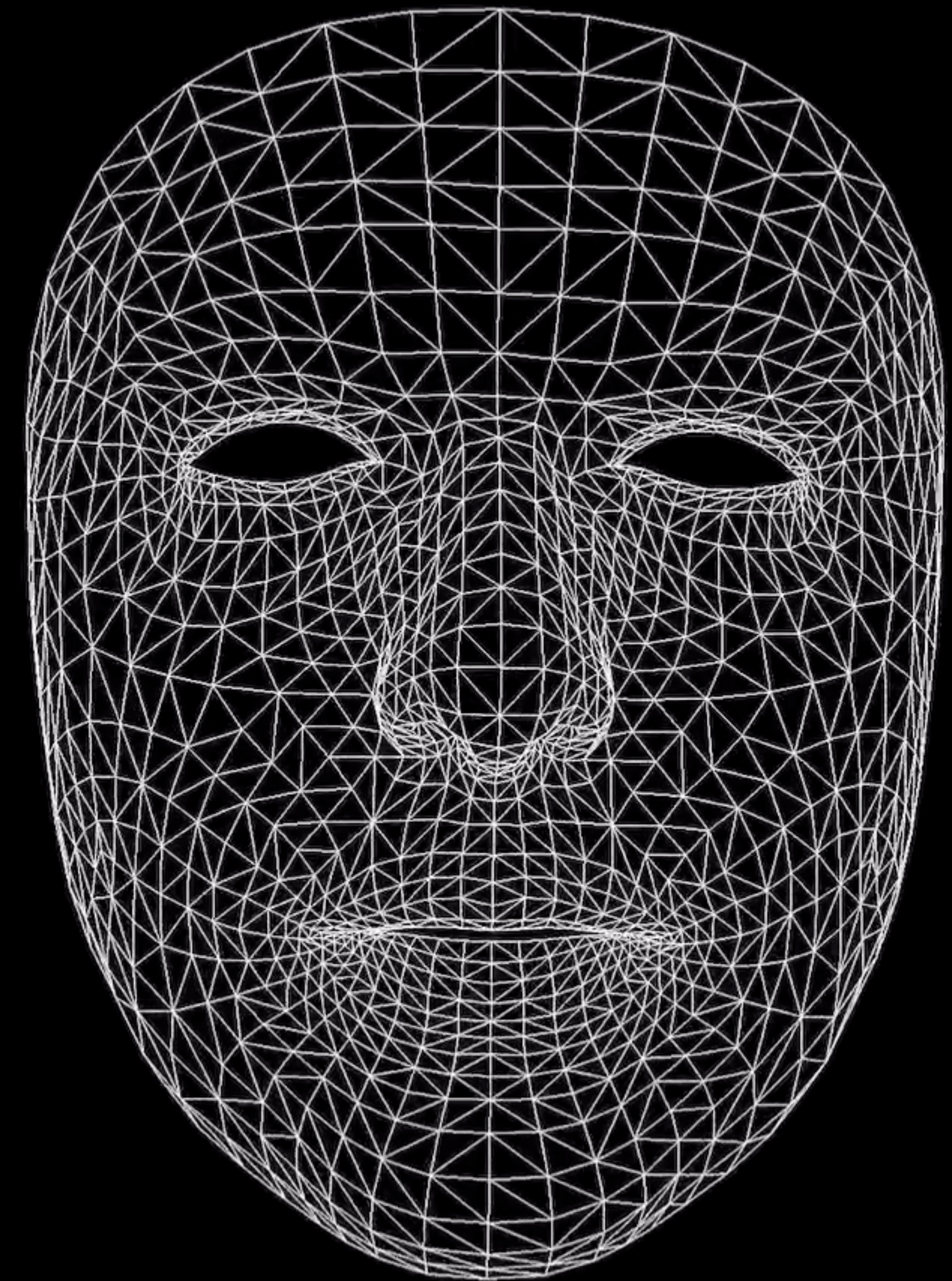
Recap



Face Tracking

Recap

Robust face detection and tracking

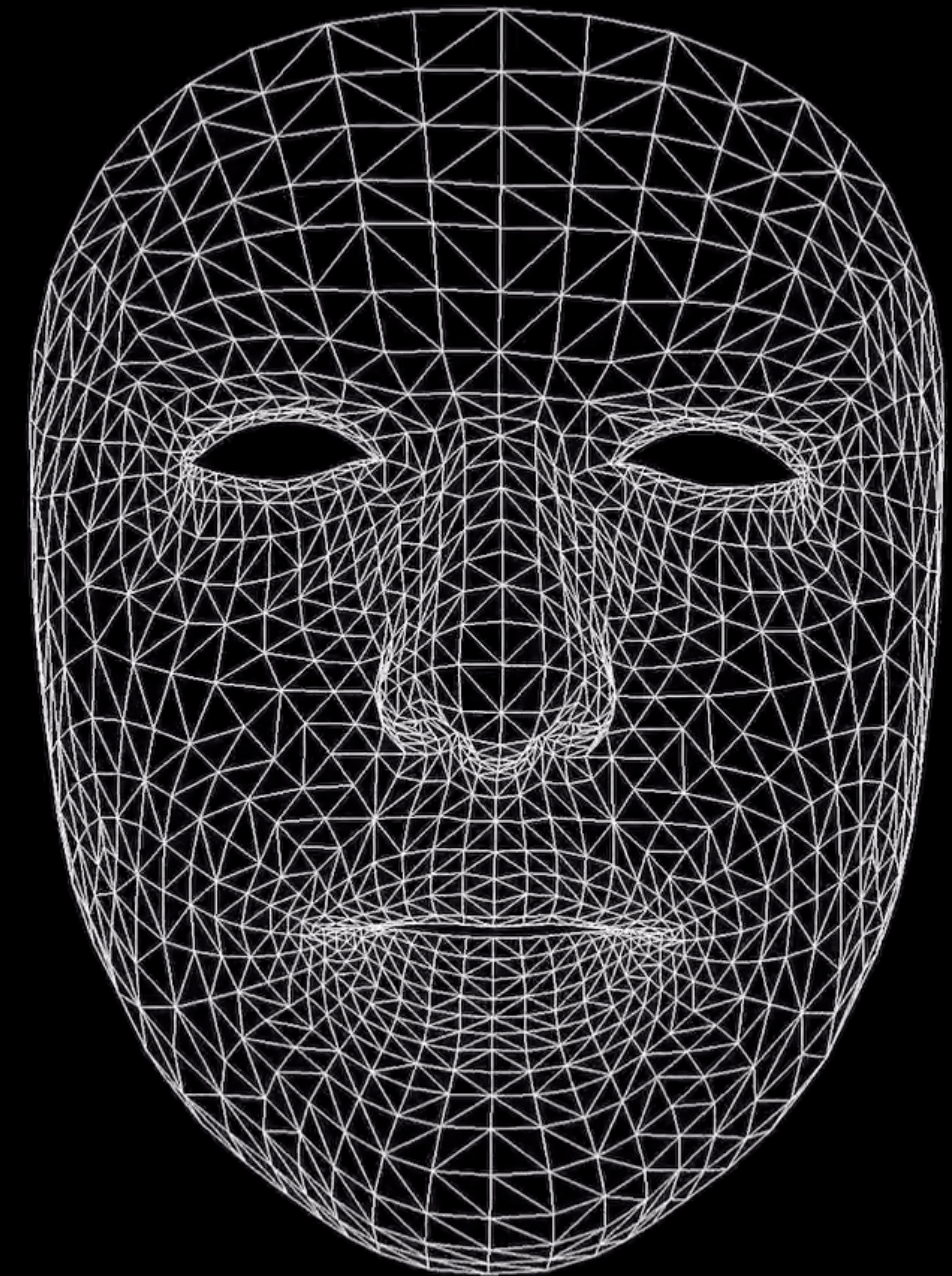


Face Tracking

Recap

Robust face detection and tracking

Position and orientation for every frame



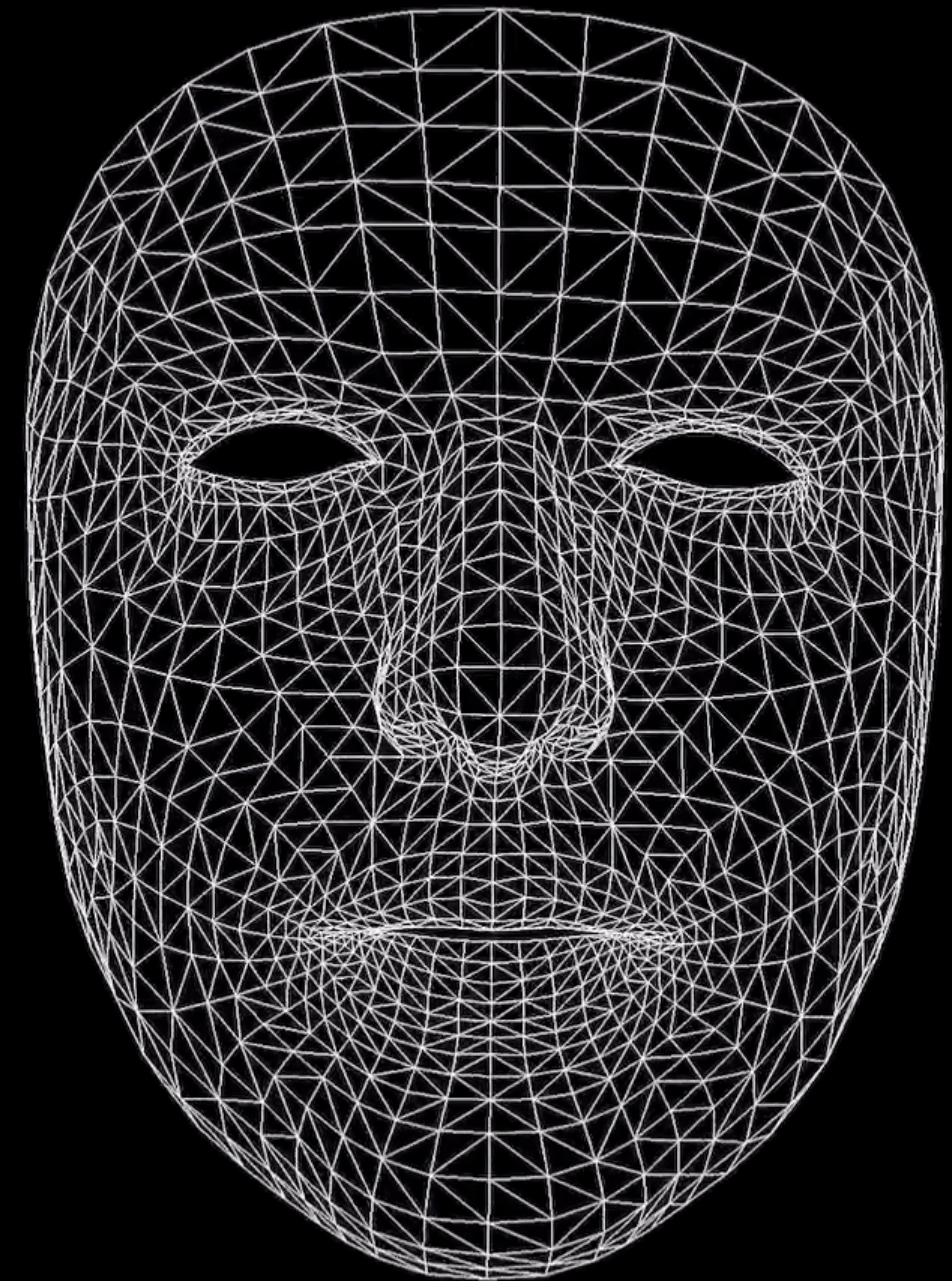
Face Tracking

Recap

Robust face detection and tracking

Position and orientation for every frame

Fitted triangle mesh/ARFaceGeometry



Face Tracking

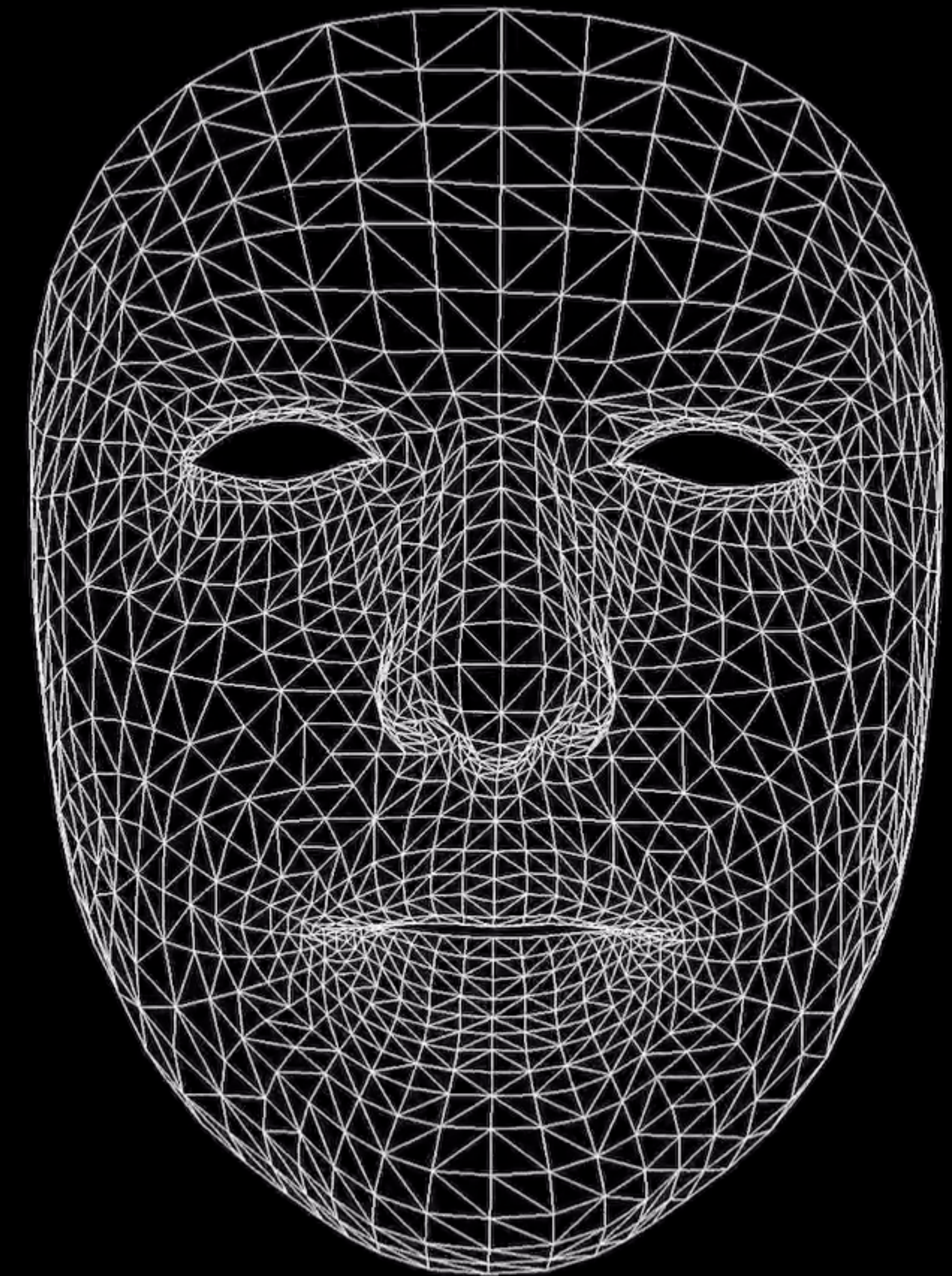
Recap

Robust face detection and tracking

Position and orientation for every frame

Fitted triangle mesh/ARFaceGeometry

ARFaceAnchor



Directional Light Estimation



Directional Light Estimation

Uses face as light probe



Directional Light Estimation

Uses face as light probe

Light intensity, direction, color temperature



Directional Light Estimation

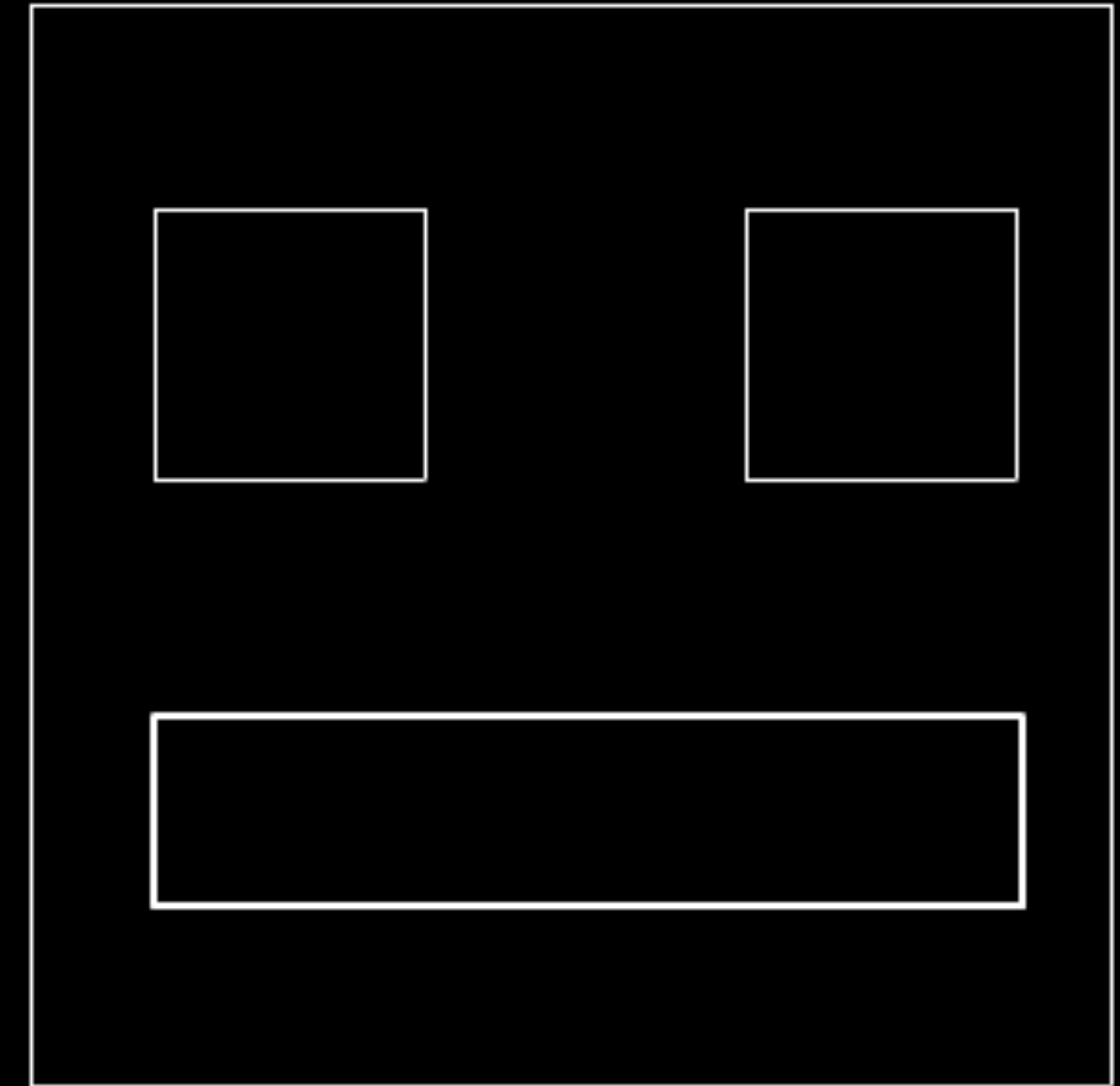
Uses face as light probe

Light intensity, direction, color temperature

Spherical harmonics coefficients

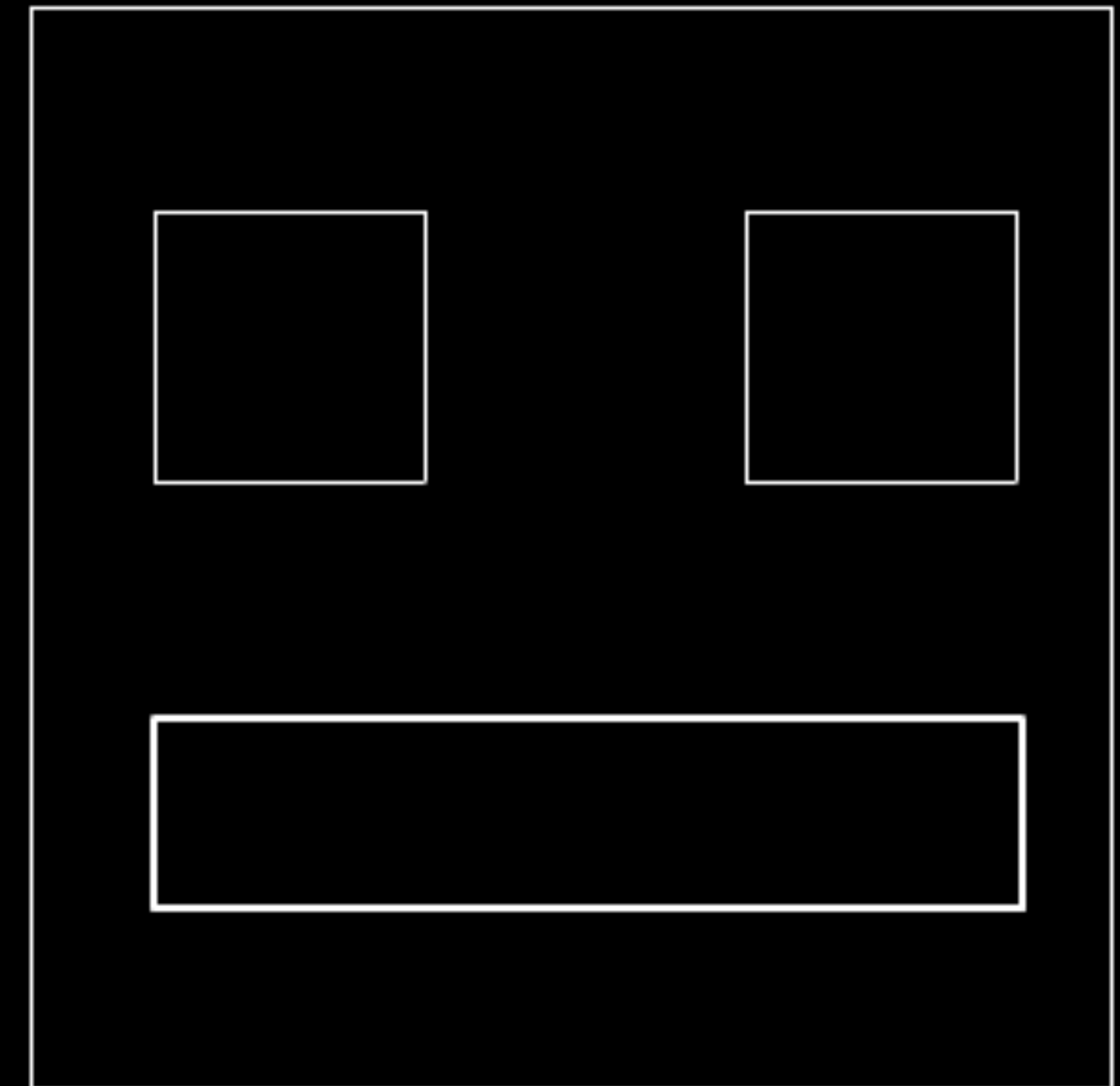


Blendshapes Recap



Blendshapes Recap

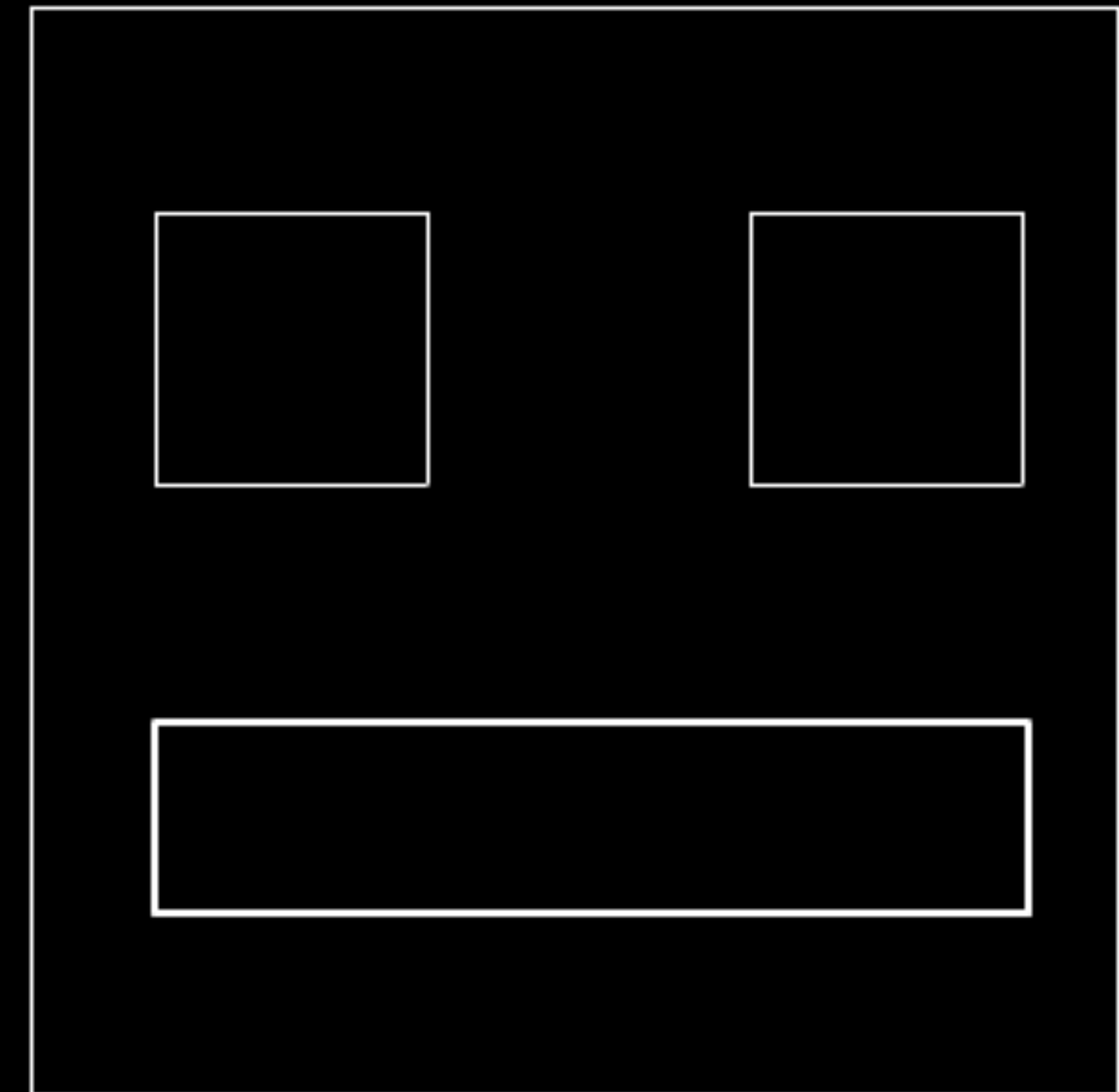
Expressions tracked in real time



Blendshapes Recap

Expressions tracked in real time

Recognizes 50+ specific features

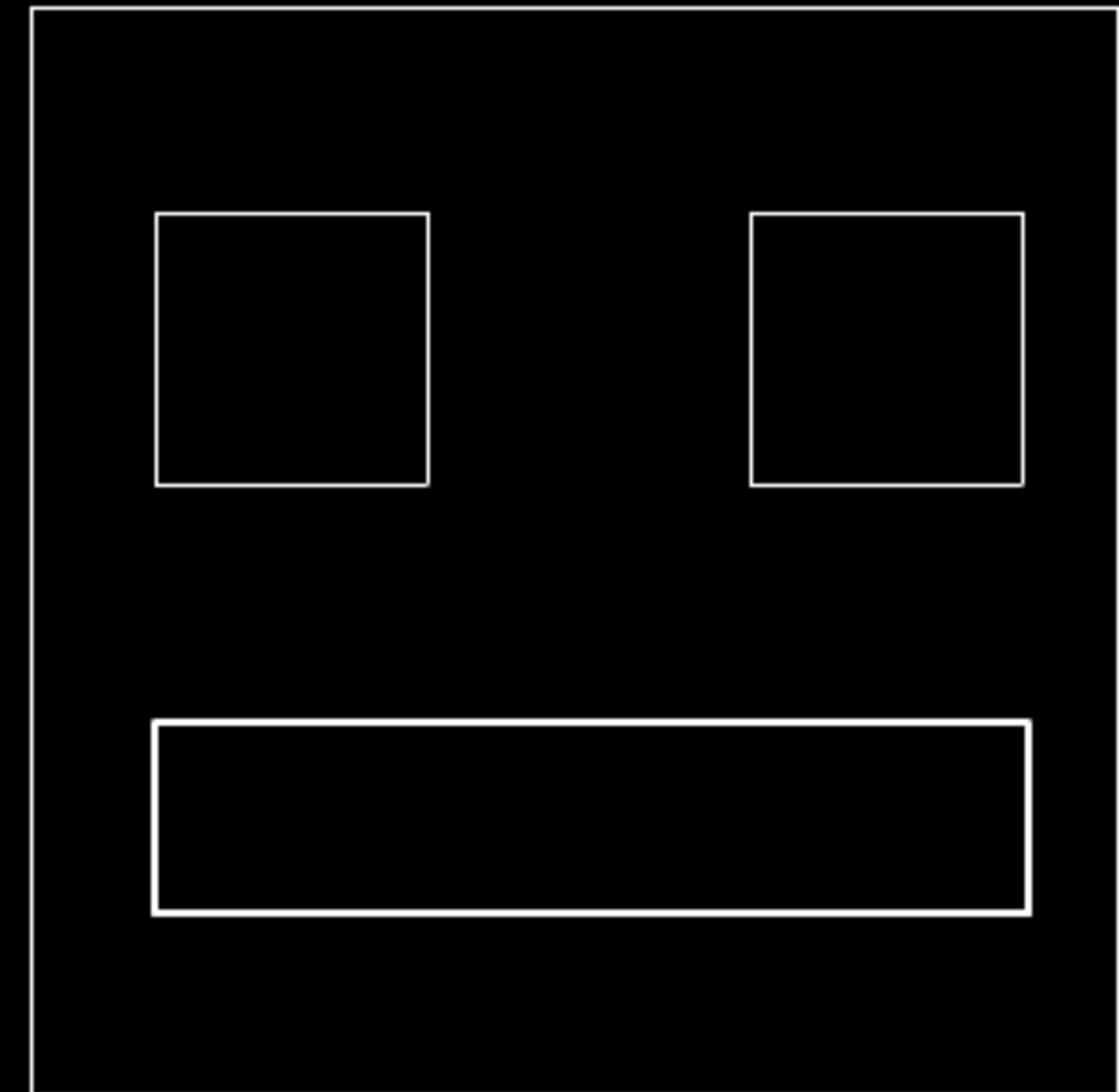


Blendshapes Recap

Expressions tracked in real time

Recognizes 50+ specific features

Animation of virtual character

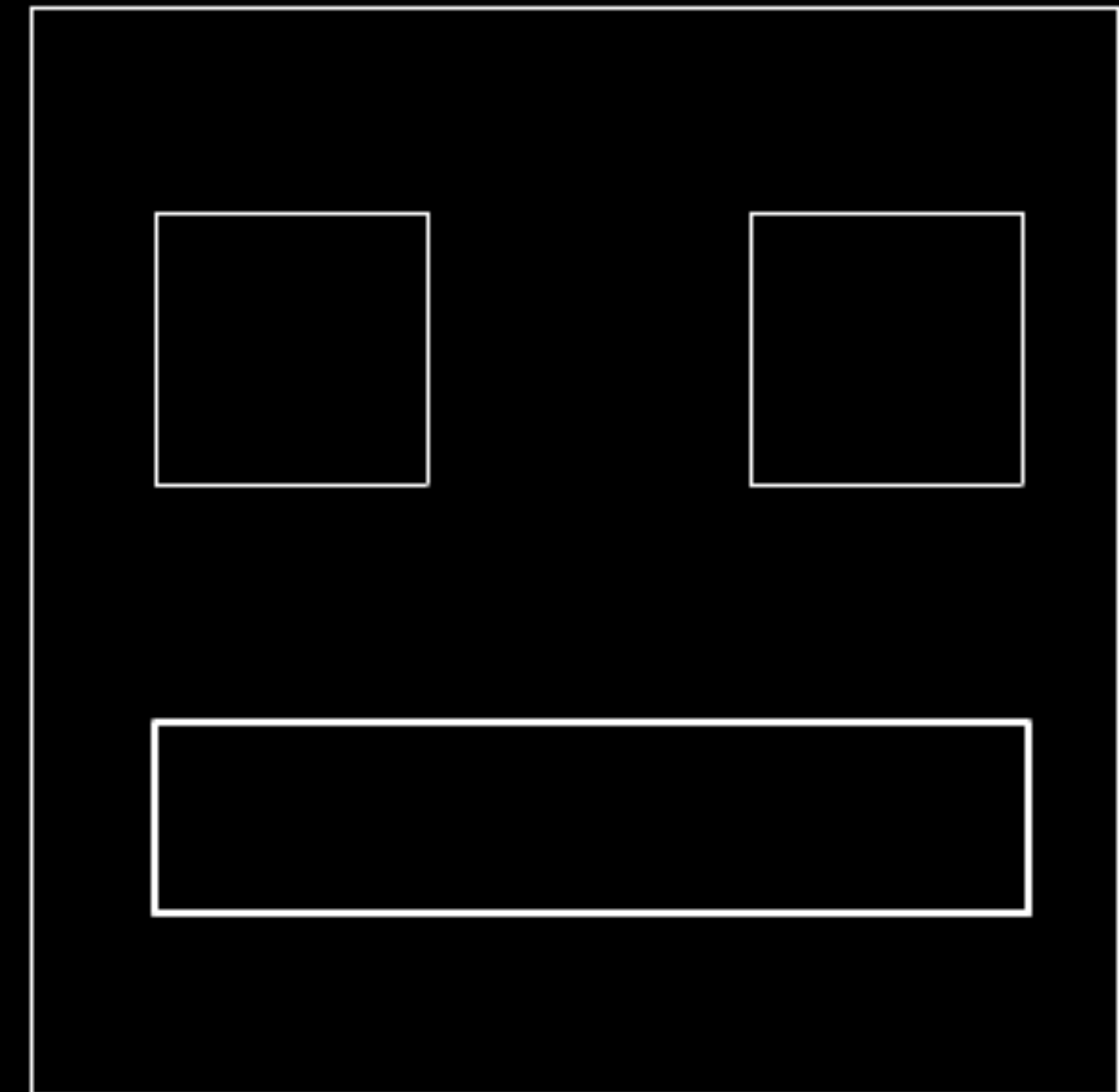


Blendshapes Recap

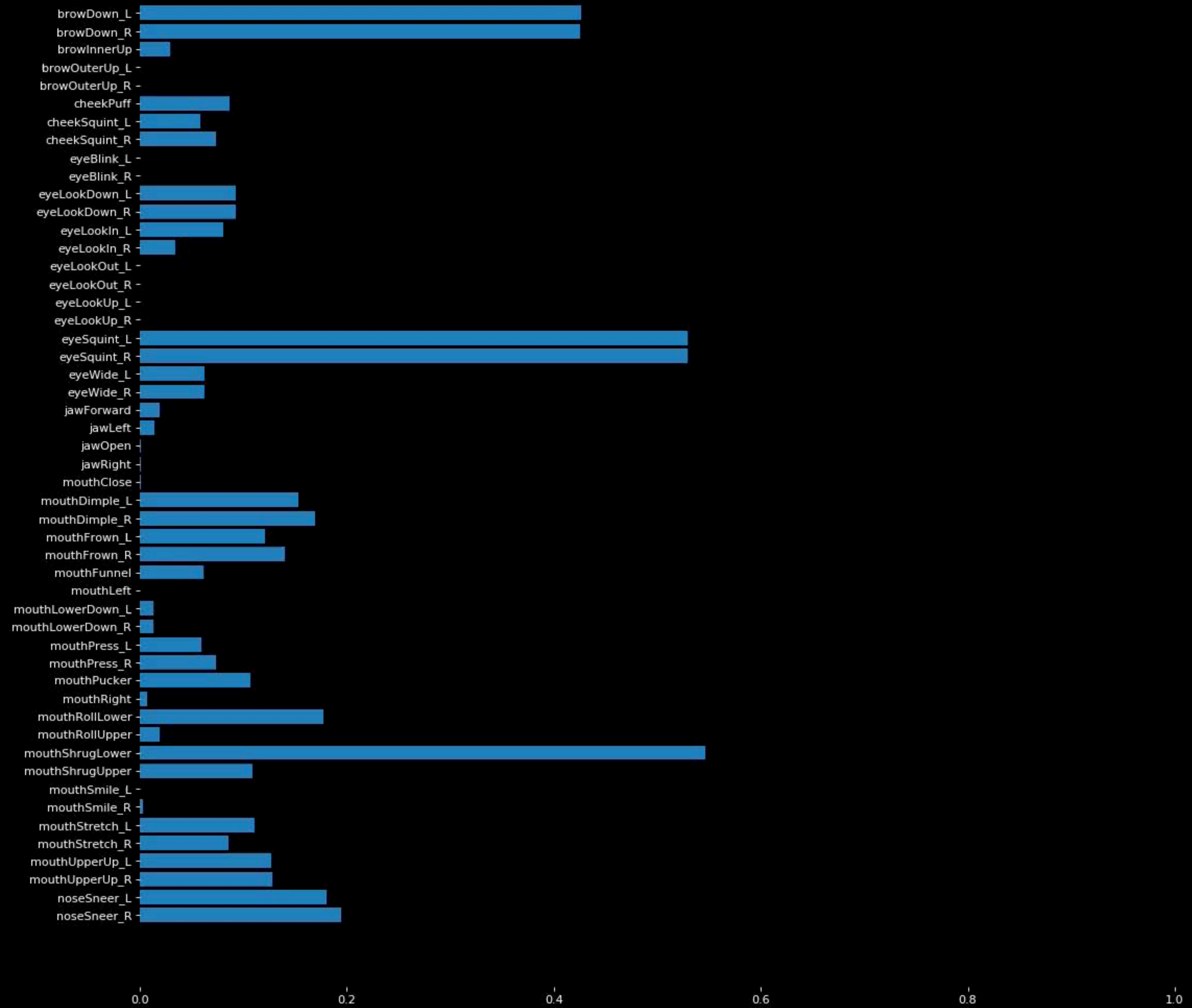
Expressions tracked in real time

Recognizes 50+ specific features

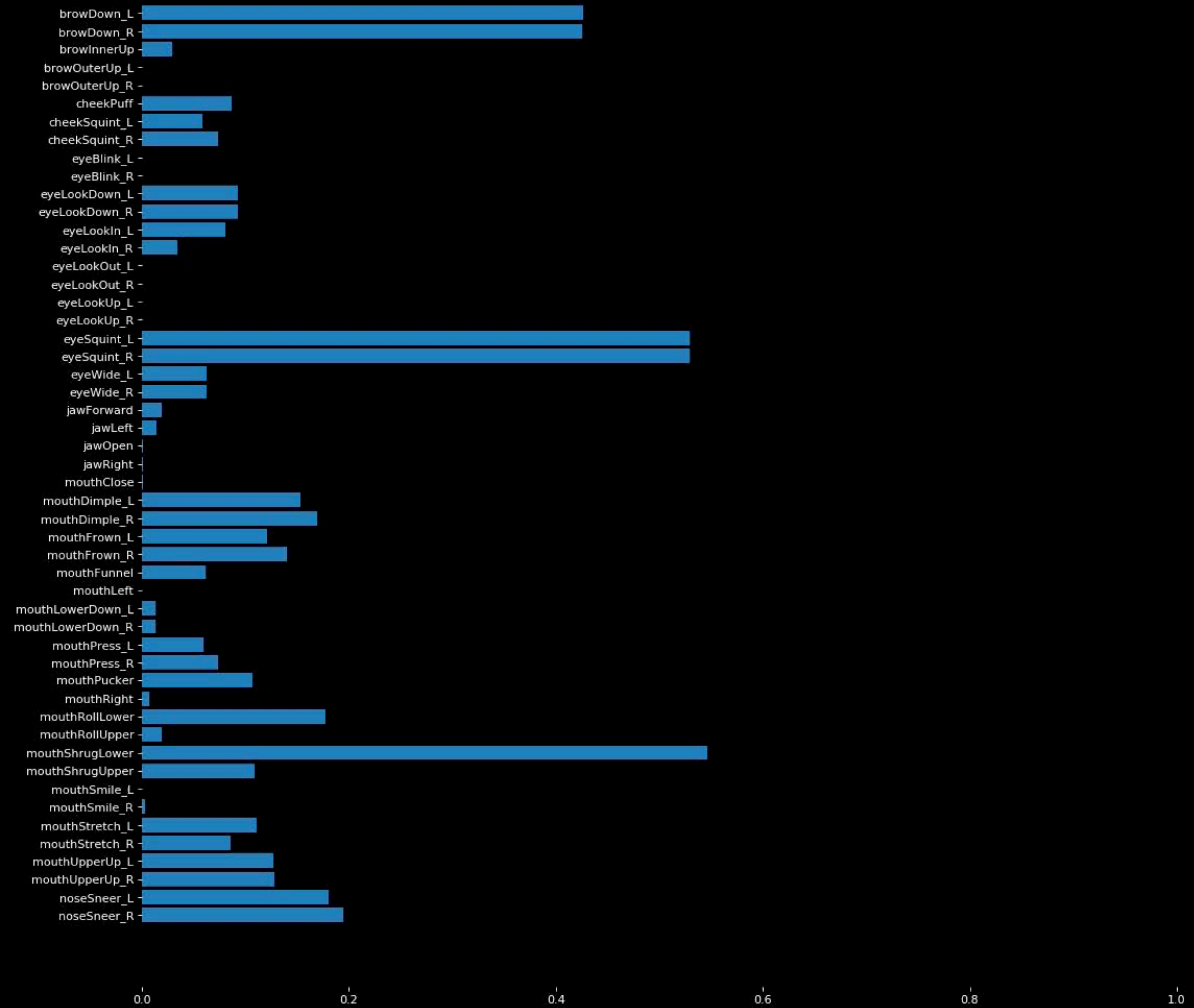
Animation of virtual character



Animoji

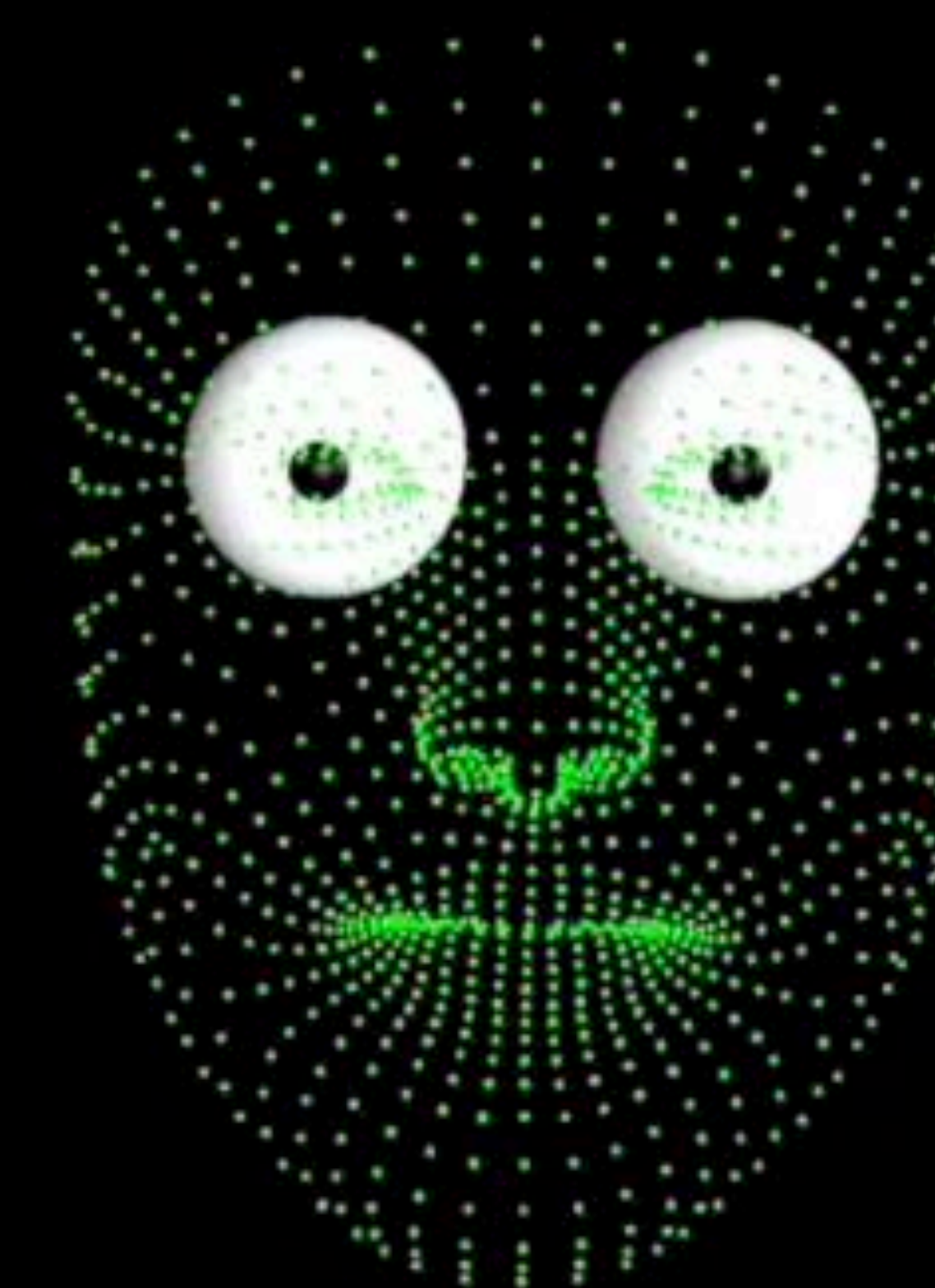


Animoji



Face Tracking Enhancements

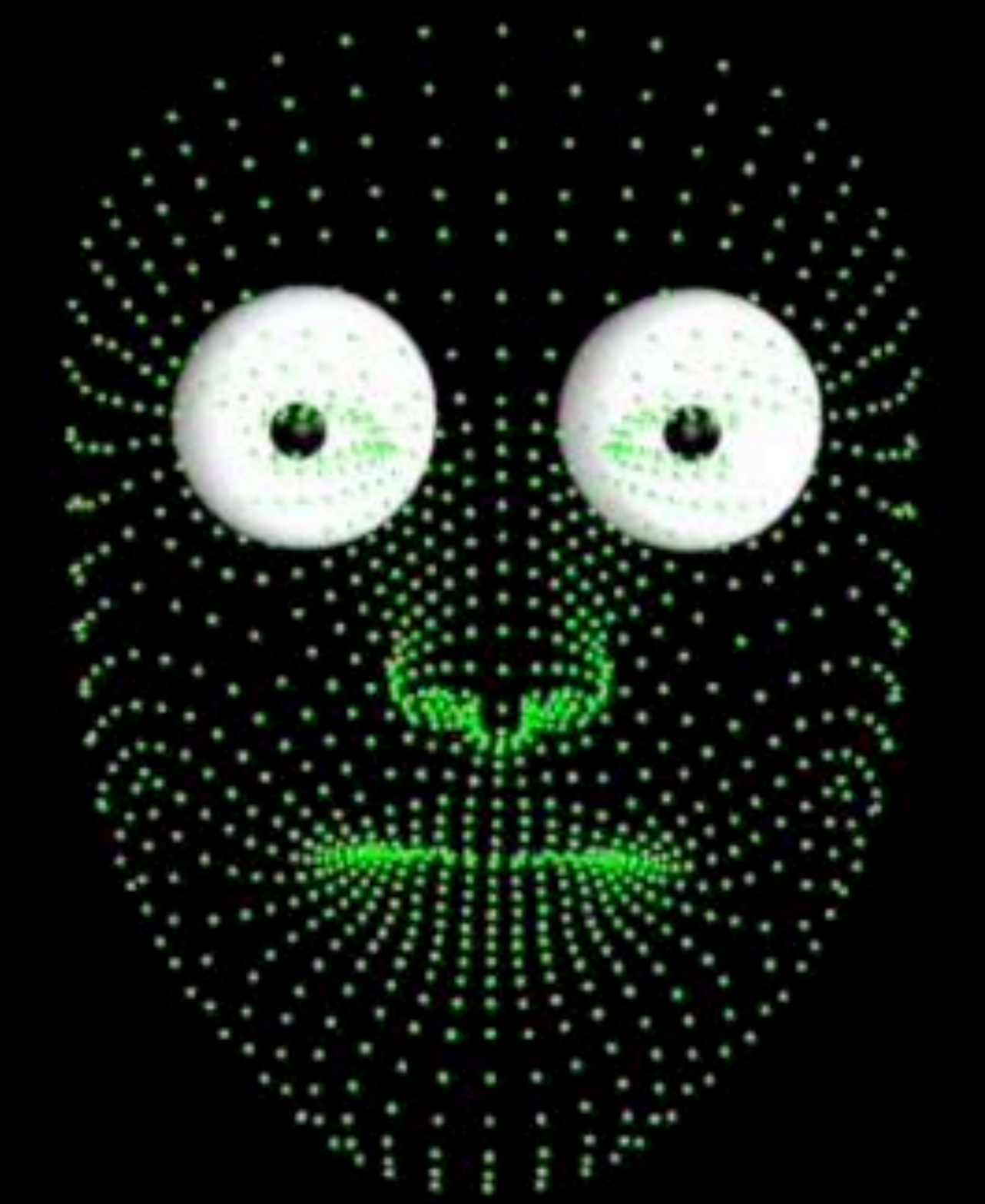
NEW



Face Tracking Enhancements

NEW

Gaze tracking

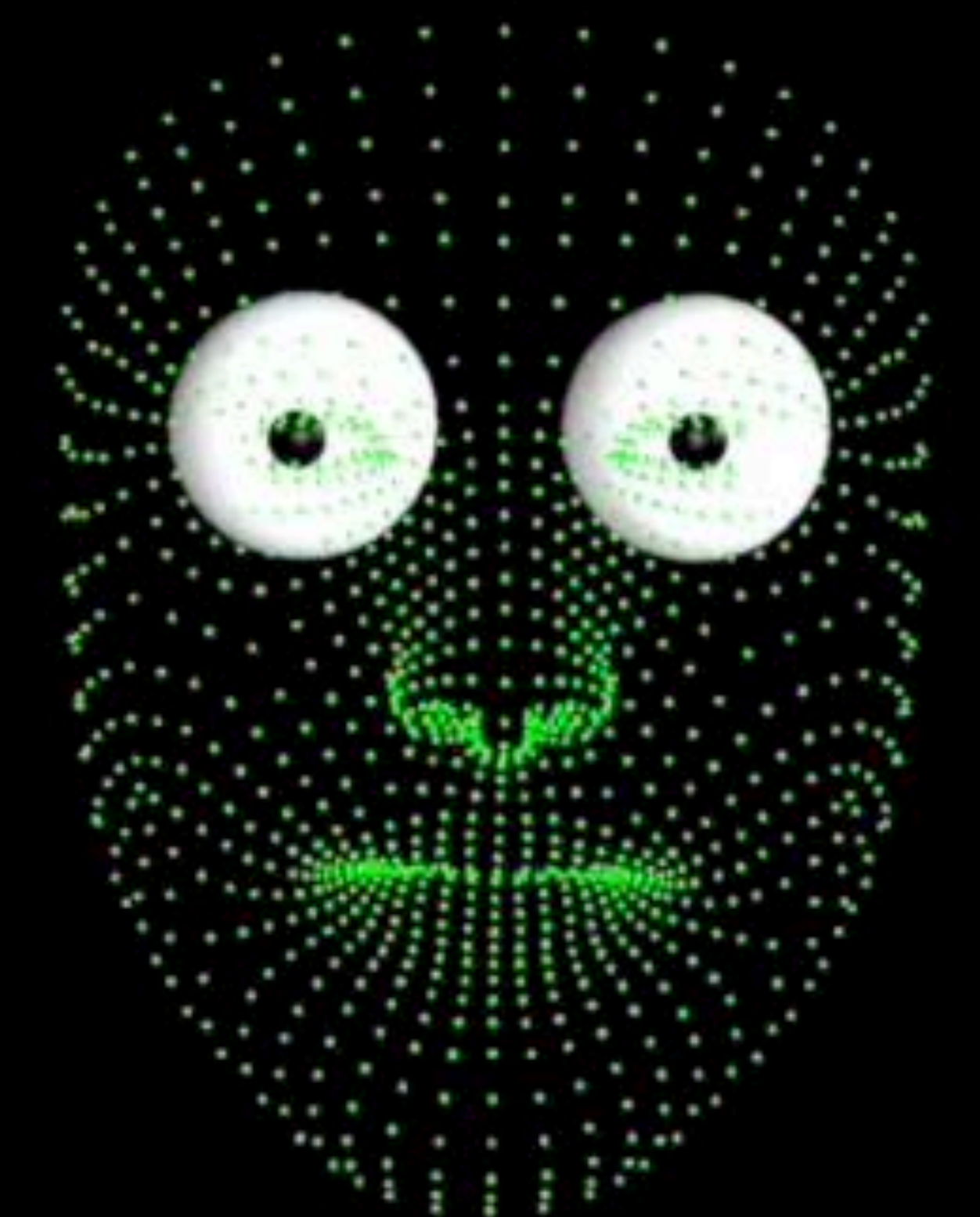


Face Tracking Enhancements

NEW

Gaze tracking

```
open class ARFaceAnchor : ARTrackable {  
  
    open var leftEyeTransform: simd_float4x4 { get }  
  
    open var rightEyeTransform: simd_float4x4 { get }  
  
    open var lookAtPoint: simd_float3 { get }  
  
}
```



Face Tracking Enhancements



Tongue support



Face Tracking Enhancements



Tongue support



Face Tracking Enhancements

NEW

Tongue support

```
extension ARFaceAnchor.BlendShapeLocation {  
  
    public static let tongueOut:  
        ARFaceAnchor.BlendShapeLocation  
  
}
```



Summary

ARKit 2—Summary



ARKit 2—Summary

Saving and loading maps



ARKit 2—Summary

Saving and loading maps

World tracking enhancements



ARKit 2—Summary

Saving and loading maps

World tracking enhancements

Environment texturing



ARKit 2—Summary

Saving and loading maps

World tracking enhancements

Environment texturing

Image tracking



ARKit 2—Summary

Saving and loading maps

World tracking enhancements

Environment texturing

Image tracking

Object detection



ARKit 2—Summary

Saving and loading maps

World tracking enhancements

Environment texturing

Image tracking

Object detection

Gaze and tongue



Building Blocks of ARKit

AROrientationTrackingConfiguration

ARWorldTrackingConfiguration

ARFaceTrackingConfiguration

ARImageTrackingConfiguration

NEW

ARObjectScanningConfiguration

NEW

ARSession

Building Blocks of ARKit

AROrientationTrackingConfiguration

ARFrame

ARWorldTrackingConfiguration

ARCamera

ARFaceTrackingConfiguration

ARLightEstimate

ARImageTrackingConfiguration

NEW

ARReferenceImage

ARObjectScanningConfiguration

NEW

ARReferenceObject

NEW

ARSession

ARWorldMap

NEW

Building Blocks of ARKit

AROrientationTrackingConfiguration

ARFrame

ARAnchor

ARWorldTrackingConfiguration

ARCamera

ARPlaneAnchor

ARFaceTrackingConfiguration

ARLightEstimate

ARFaceAnchor

ARImageTrackingConfiguration

NEW

ARReferenceImage

ARImageAnchor

ARObjectScanningConfiguration

NEW

ARReferenceObject

NEW

ARObjectAnchor

NEW

ARSession

ARWorldMap

NEW

AREnvironmentProbeAnchor

NEW

Building Blocks of ARKit

AROrientationTrackingConfiguration

ARFrame

ARAnchor

ARWorldTrackingConfiguration

ARCamera

ARPlaneAnchor

ARFaceTrackingConfiguration

ARLightEstimate

ARFaceAnchor

ARImageTrackingConfiguration

NEW

ARReferenceImage

ARImageAnchor

ARObjectScanningConfiguration

NEW

ARReferenceObject

NEW

ARObjectAnchor

NEW

ARSession

ARWorldMap

NEW

AREnvironmentProbeAnchor

NEW

More Information

<https://developer.apple.com/wwdc18/602>

ARKit Lab

Technology Lab 5

Tuesday 12:00PM

Integrating Apps and Content with AR Quick Look

Hall 2

Tuesday 4:00PM

AR Get Together

Technology Lab 10

Wednesday 6:15PM

 **WWDC18**