iOS Application Development

Lecture 10: ARKit

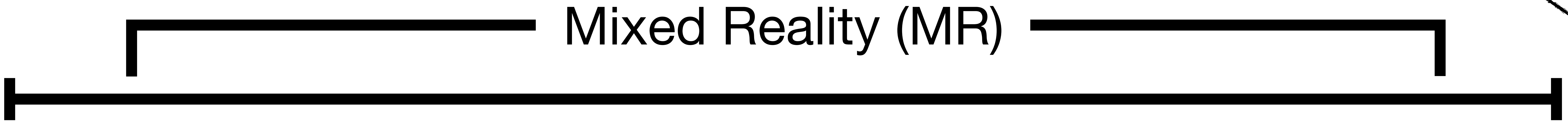
Simon Völker & Philipp Wacker
Media Computing Group
RWTH Aachen University

hci.rwth-aachen.de/ios

[Milgram & Kishino, 1994]

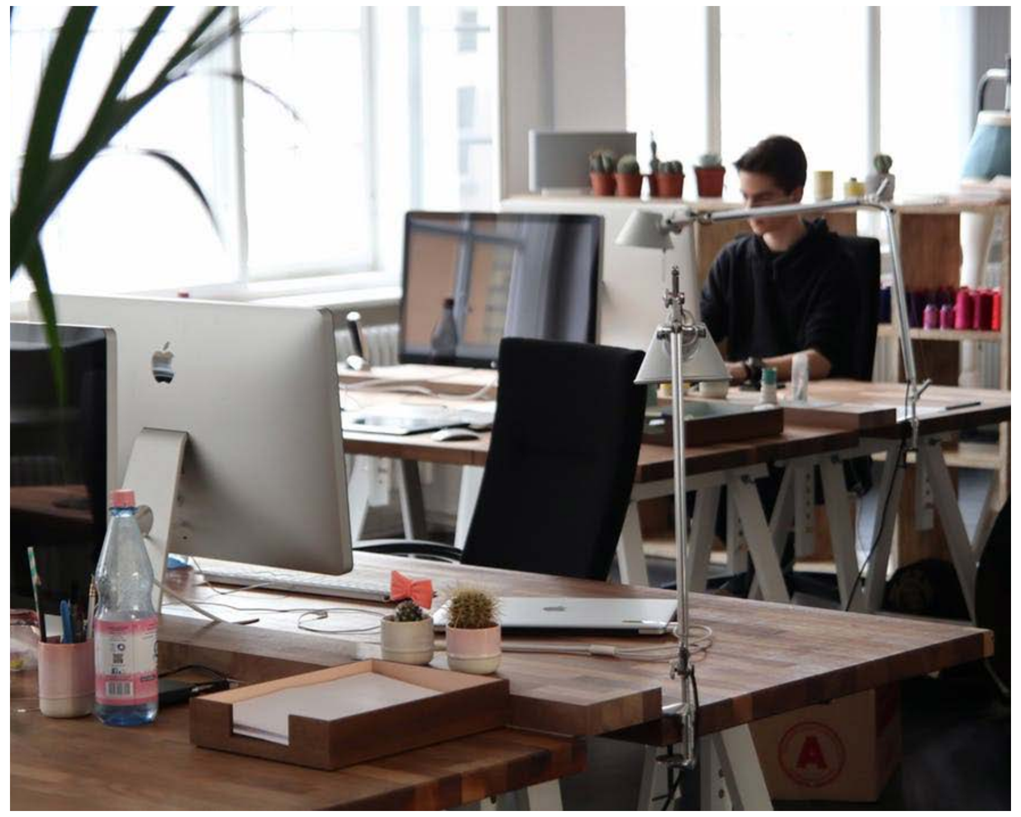
Reality-Virtuality Continuum

AR != VR



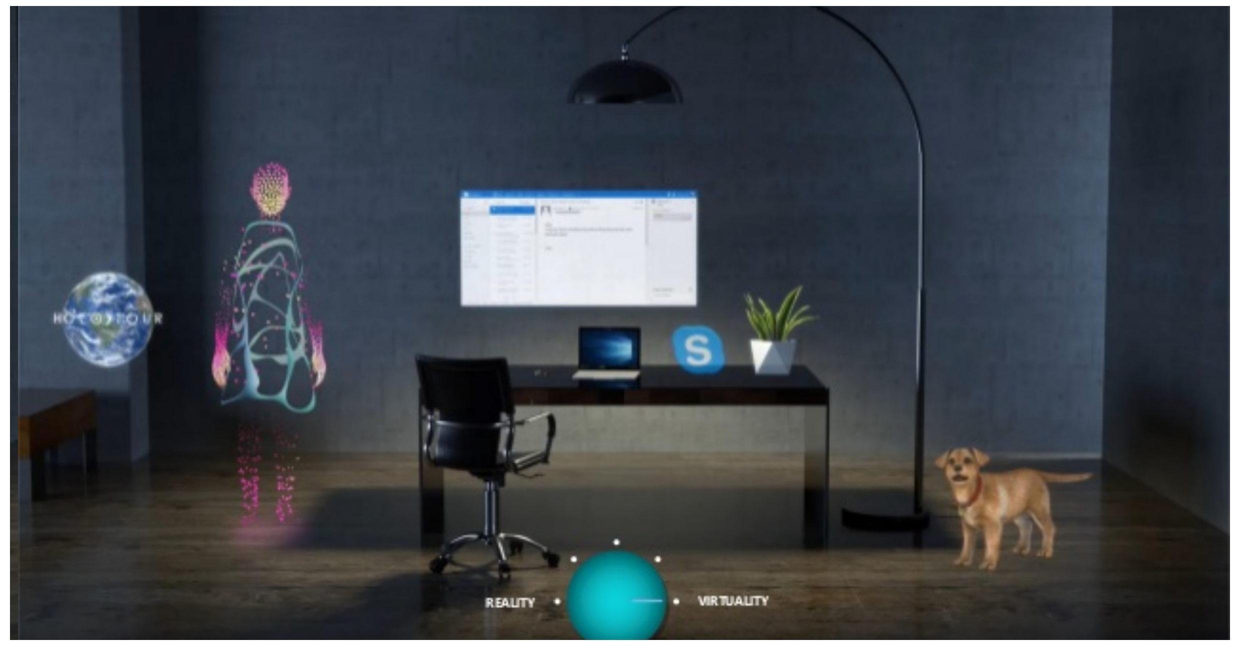
Mixed Reality (MR)

Real Environment



Augmented Reality (AR)

Augmented Virtuality (AV)



Virtual Environment

- In AV and VE/VR the surrounding environment is virtual, in AR the surrounding environment is real

How to show it?

Display technologies

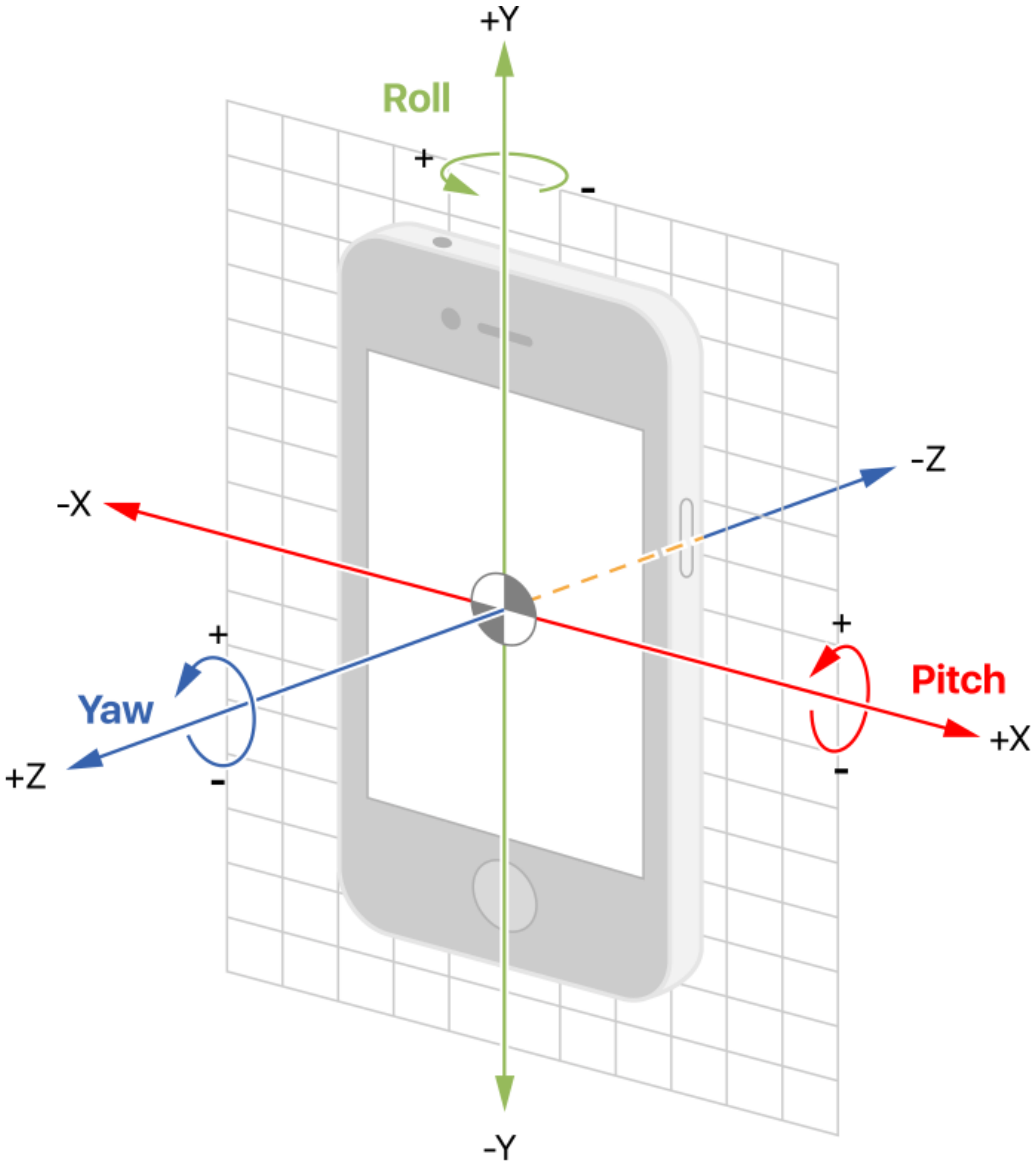
- Show virtual objects overlaying the real world in 3D space
- Head mounted
- Spatial
- Handheld

Where to show it?

Tracking (and registration) technologies

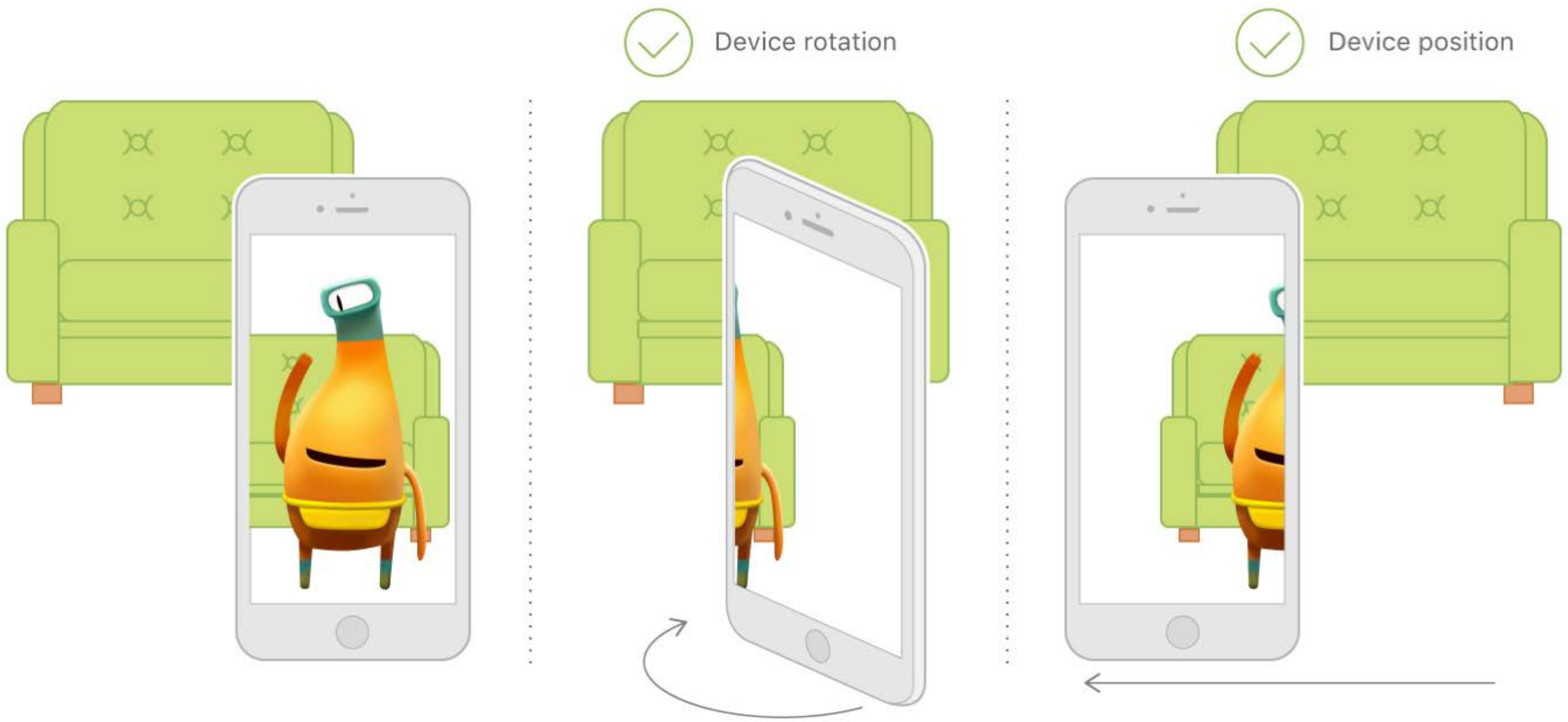
- To register virtual objects in 3D space and track user input
- Track the
  - (a) scene
  - (b) the user's 6DOF viewpoint (head and/or eyes)
  - (c) the user's hands/body for input
  - (d) input devices

Tracking



World Tracking

- Back-facing camera



Face Tracking

- Front-facing camera

ARConfiguration & ARSession

- `ARConfiguration` defines which camera and tracking algorithms are being used
  - Configurations for: – world, body, orientation, image, and face tracking
– object scanning
- `ARSession` manages the camera and motion processing

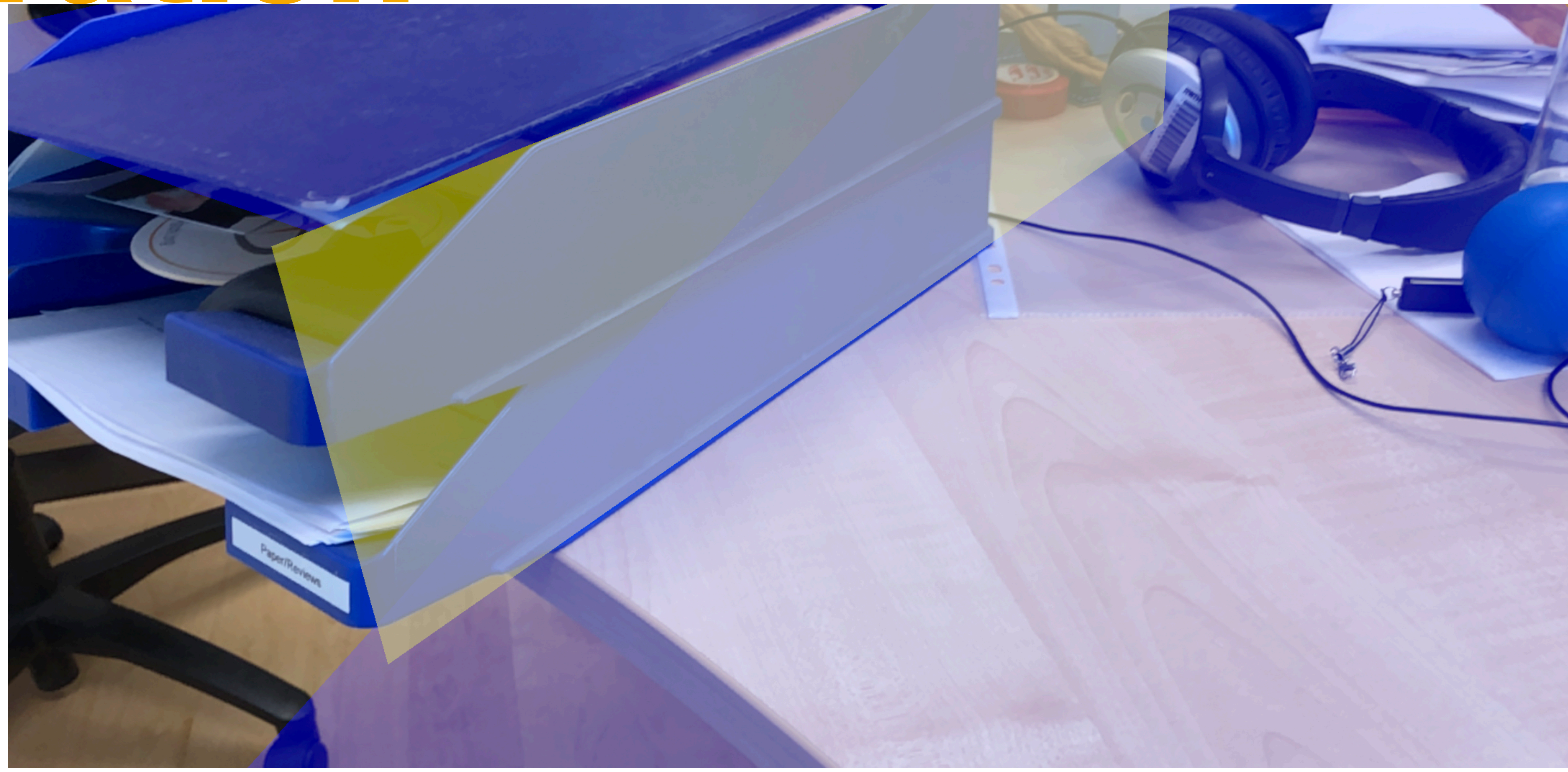
```
// Create a session configuration
let configuration = ARWorldTrackingConfiguration()

// Run the view's session
sceneView.session.run(configuration)
```

```
// Pause the view's session
 sceneView.session.pause()
```

ARWorldTrackingConfiguration

- Define what to look for in the scene
  - Plane detection
  - Image detection
  - Object detection
- Create a world map
  - Persistence
  - Multiple viewers on the same scene



ARSceneView

- Automatically aligns SceneKit's coordinate system with the world coordinate system & moves the "virtual" camera

- Show statistics

```
// Show statistics such as fps and timing information
sceneView.showsStatistics = true
```

- Debug options

```
sceneView.debugOptions = [ARSCNDebugOptions.showWorldOrigin,
                          ARSCNDebugOptions.showFeaturePoints]
```

- `ARSCNViewDelegate`
  - Notifications as features, such as planes, are detected

Feature Points

Finding flat surfaces

- Define what planes to look for

```
configuration.planeDetection = [.horizontal, .vertical]
```

- Implement `ARSCNViewDelegate` methods to respond to found planes

```
func renderer(_ renderer: SCNSceneRenderer, didAdd node: SCNNode, for anchor: ARAnchor) {}

func renderer(_ renderer: SCNSceneRenderer, didUpdate node: SCNNode, for anchor: ARAnchor) {}

func renderer(_ renderer: SCNSceneRenderer, didRemove node: SCNNode, for anchor: ARAnchor) {}
```

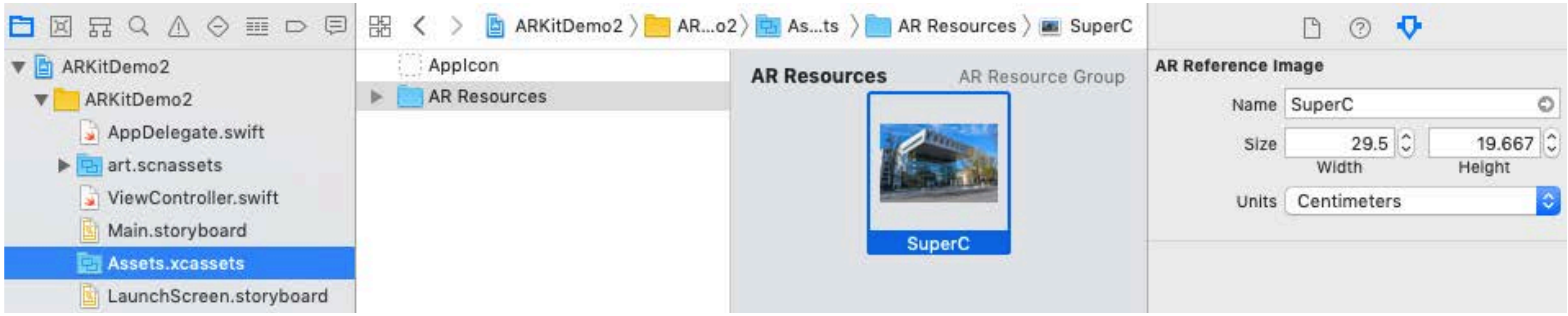
```
guard let planeAnchor = anchor as? ARPlaneAnchor else {return}
switch planeAnchor.alignment {
 case .horizontal: //...
 case .vertical:   //...
}
```

Anchors

- Matching anchors for different feature tracking
  - `ARPlaneAnchor`, `ARObjectAnchor`, `ARImageAnchor`, `ARFaceAnchor`
- Feature specific properties
  - PlaneAnchor: `alignment`, `center`, `extend`, `geometry`, …
  - ImageAnchor: `referenceImage`

Image Recognition with ARKit

- Define what images to look for:



AR Reference Image

Name: SuperC

Size: 29.5 (Width) 19.667 (Height)

Units: Centimeters

- Use `referenceImage` property of an `ARImageAnchor` to access the specific images `name` and `physicalSize`

```
let referenceImages =  ARReferenceImage.referenceImages(
                       inGroupNamed: "AR Resources", bundle: nil)!
configuration.detectionImages = referenceImages

//to enable continuous tracking
configuration.maximumNumberOfTrackedImages = 1
```

Interaction with AR

- Cast a ray into the scene and find intersections with the real world

```
//get the point from which to cast the ray
let touchLocation = sender.location(in: sceneView)
//perform the hit test
let hitTestResult = sceneView.hitTest(touchLocation, types: [.existingPlaneUsingExtent])
```

- Array of `ARHitTestResult`:
  - Type & Anchor
  - Distance & transforms

```
.featurePoint
.estimatedHorizontalPlane
.estimatedVerticalPlane
.existingPlane
.existingPlaneUsingExtent
.existingPlaneUsingGeometry
```

ARKit Demo