MANOMOTION

SDK PRO v_1.4.9 AR Foundation BETA

Documentation

Table of contents:

Introduction

This documentation is a complement to the SDK PRO Technical Documentation. It explains how ManoMotion SDK Pro can be used together with the Unity AR Foundation framework. This framework is a wrapper for ARCore XR plugin (Android) and ARKit XR Plugin (iOS).

Prerequisites

- General knowledge about the Unity 3D engine
- General knowledge about AR Foundations, ARCore or ARKit.
- Being able to deploy applications for Android or iOS using Unity.
- A device of preference (iOS or Android) for deployment (preferably a high end device) with AR support.

Glossary

Term	**Description**
ARCore	SDK from Google for AR applications
ARKit	SDK from Apple for AR applications
SLAM	Simultaneous Localization and Mapping, maps the environment with feature points for AR understanding.
Plane Detection	Creates planes based on the feature points from the SLAM tracking that can map to example the ground.
device	The android or iOS device.

AR Foundations

AR Foundation is a cross-platform framework for Unity that can be used together with ARCore and ARKit. AR uses SLAM to track the real world through the device camera to make it possible to place virtual objects in the “real environment”.

Deploy ManoMotion SDK PRO AR Foundations

Before deploying the ManoMotionSDKProFeatures scene to the device, AR foundation needs to be added to the Unity project. AR foundation can be added using the Unity package manager (Window → Package Manager). The packages needed to be installed are “AR Foundation” together with “ARCore XR Plugin” (for Android) or/and “ARkit XR Plugin” (for iOS). After this the bundleID (Project Settings) and license key (ManoMotionManager) needs to be put in. Also ARCore does not support Vulkan graphics API, this can be removed from the project settings → Graphics API.

Input Manager AR Foundations

The main difference between the SDK PRO AR and SDK PRO is the way the SDK receives the images. In SDK PRO ManoMotion takes control over the device camera and sends the images to the SDK. When using SDK PRO AR, AR Foundation will take control over the camera and SDK PRO uses frame cloning from the AR Camera Background to get and send the images to the SDK. From the InputManagerARFoundations script it's possible to set the MaxCustomResolution variable that's used to set the scale of the image cloned, if a smaller image is used the detection quality might reduce but it will gain some speed. We recommend using a value between 500 and 800.

Add On (Sessions)

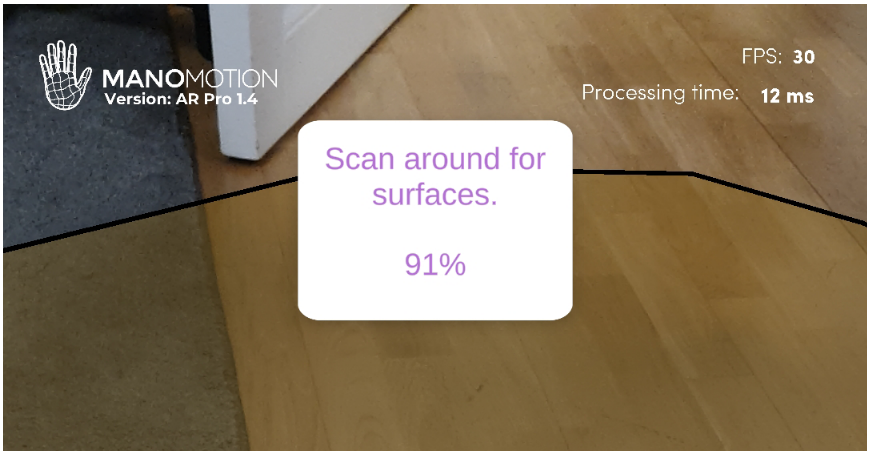
When using SDK PRO AR it needs to use the ARFoundations AddOn from the session settings. This is so the SDK knows how to handle the images it receives.

ManoUtils

For the SDK to calculate the depth for the skeleton joints together with finger and wrist points, the method CalculateNewPositionWithDepth. This method takes the Vector3 of the joint position, it also takes a float value, the DepthEstimation from the tracking information and applies this value to the joints. CalculateNewPositionWithDepth makes use of the AR Camera in the Unity project so make sure that the AR Camera is tagged with the MainCamera.

ManoMotionSDKProFeatures Scene

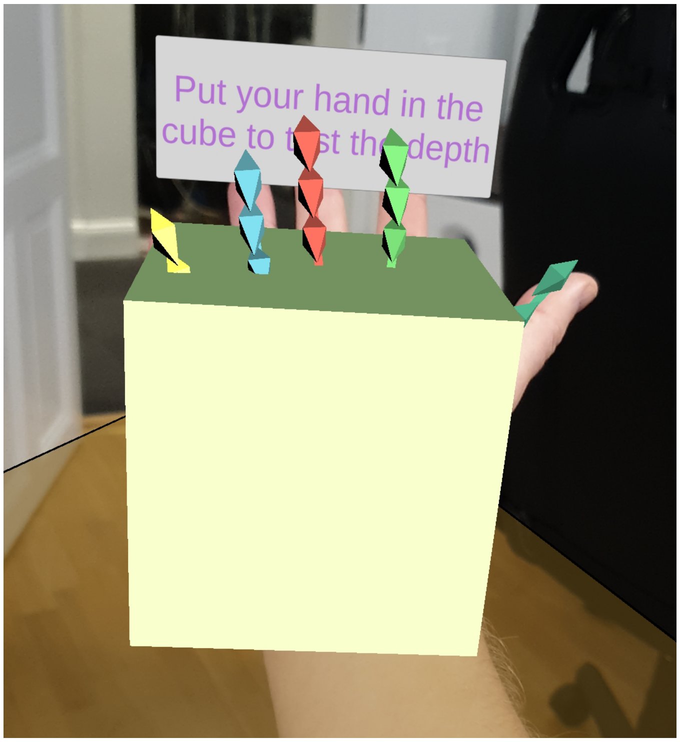
The features scene that comes with the package, is a scene that will guide you and also showcase the different features that SDK PRO AR comes with.
After camera and storage permissions are granted and the ManoMotion policy accepted the scan instructions appear. To scan for surfaces move the camera around facing the ground so AR foundation can find feature points. This is needed for the AR tracking to work. When reaching 100% the manomotion skeleton will show when the hand is placed in front of the camera.



Scan around for surfaces.

91%

In the scene you will see a cube, shown in the image below. This cube can be used to see how the depth works with the skeleton hand. The skeleton will go into the cube if the hand is moved there. The depth value from the SDK is used to place the hand in different depths. When the whole hand is moving closer or thunder away from the camera all the joints will be affected but all the joints also have their individual depth value in relation to the wrist joint.



Interaction Example Scene

There is also one scene in the project that demonstrates how it can be possible to interact with AR objects. In the Interaction Example there are 4 cubes that can be interacted with by putting the hand into it and if hold (closed pinch) is performed the cube will rotate and if the click trigger gesture is performed small cubes will spawn on top of the big cube.

Limitations

Image Tracking

When using ManoMotion SDK PRO together with image tracking the hand might cover the image that is tracked and therefore it might lose the tracking.

Grab detection

The detection when performing the Grab gesture is unstable which can make it hard to interact with AR objects using this gesture.

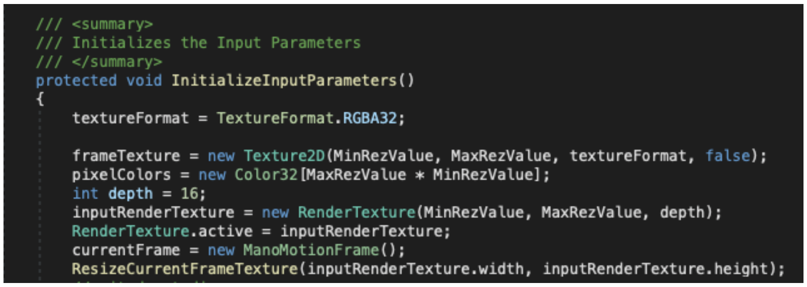
Troubleshooting

ARFoundation 4.x iOS, black screen

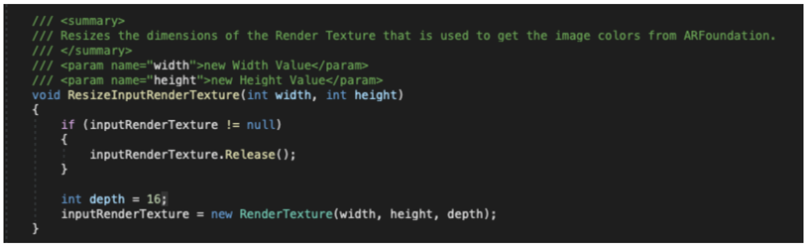
If you build an application using ARFoundation 4.x packages along with ManoMotion's SDK, you may experience an issue where the screen is completely black with some yellow figures (which correspond to a non-existent plane detection)

This is our current fix for this issue:

- In project settings → XR Plug-in Manager, check ARKit
- Project settings → Check Requires ARKit support
- Update ARFoundation and ARkit Face tracking and ARKit to version 4
- Import Universal Renderer Pipeline → create a pipeline Asset, add ARBackground renderer feature, and set the renderer pipeline asset in Project settings - Graphics. Also materials need to be upgraded for URP (Edit- render pipeline- universal render pipeline - upgrade project materials...)
- Add some depth to the render texture used to get the AR Background:



```
/// <summary>
/// Initializes the Input Parameters
/// </summary>
protected void InitializeInputParameters()
{
    textureFormat = TextureFormat.RGBA32;

    frameTexture = new Texture2D(MinRezValue, MaxRezValue, textureFormat, false);
    pixelColors = new Color32[MaxRezValue * MinRezValue];
    int depth = 16;
    inputRenderTexture = new RenderTexture(MinRezValue, MaxRezValue, depth);
    RenderTexture.active = inputRenderTexture;
    currentFrame = new ManoMotionFrame();
    ResizeCurrentFrameTexture(inputRenderTexture.width, inputRenderTexture.height);
```



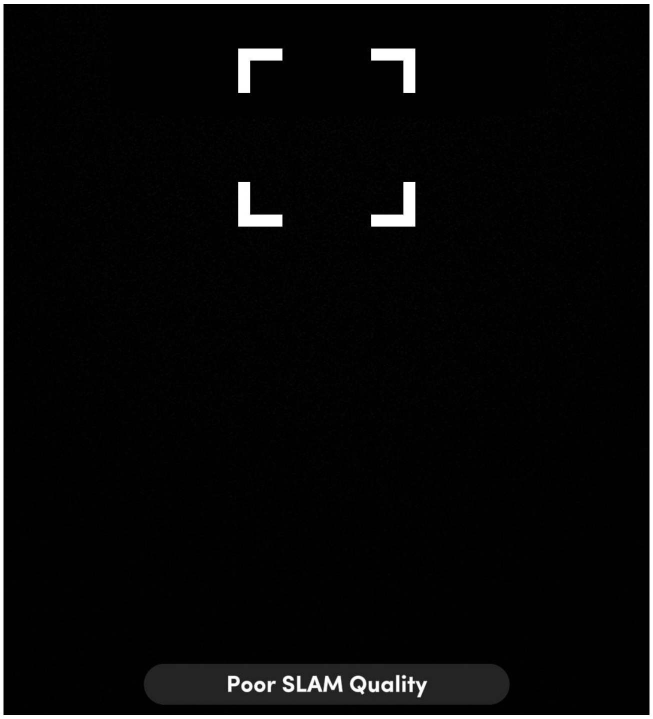
```
/// <summary>
/// Resizes the dimensions of the Render Texture that is used to get the image colors from ARFoundation.
/// </summary>
/// <param name="width">new Width Value</param>
/// <param name="height">new Height Value</param>
void ResizeInputRenderTexture(int width, int height)
{
    if (inputRenderTexture != null)
    {
        inputRenderTexture.Release();
    }

    int depth = 16;
    inputRenderTexture = new RenderTexture(width, height, depth);
}
```

Missing references

If unity gives errors about missing references after importing the Unity package, this is because it needs to have the AR foundation Unity package installed. See Deploy ManoMotion SDK PRO AR Foundations.

AR related issues

If the SLAM Quality for some reason (example to little light) is not good enough to track any feature points this information will be displayed on screen. Also if the device does not support AR that information will be shown like this.



Poor SLAM Quality