



ARCore Integration Guide

Overview

This document demonstrates how to recognize visual content with DMSDK and render its location in space using Google ARCore. The samples provided are based on Sceneform for simplicity, but the same principles can be applied to ARCore without Sceneform. This document will be updated as Google adds features to ARCore.

Configure the ARCore session

Configure the AR camera with optimal settings for reading Digimarc Barcode and traditional 1D barcodes:

```
// Create Session
Session session = new Session(this);

// This will always return 3 CameraConfig -> VGA, something between 480p and
// 1080p, 1920x1080
List<CameraConfig> cameraConfigs = session.getSupportedCameraConfigs();

// Pick the largest one
CameraConfig cameraConfig = cameraConfigs.stream().max(Comparator.comparing(o
-> o.getImageSize().getWidth())).get();

session.setCameraConfig(cameraConfig);

// The camera subsystem uses fixed focus by default - configure it to use
// autofocus instead.
Config config = new Config(session);
config.setFocusMode(Config.FocusMode.AUTO);
session.configure(config);
```

Add a VideoCaptureReader

The AR camera returns Image objects with format of ImageFormat.YUV_420_888 (3 planes, including luminance and subsampled Cb and Cr). Configure the VideoCaptureReader to process this pixel format:

```
videoCaptureReader = new VideoCaptureReader(
    BaseReader.All_Image_Readers,
    null,
    CaptureFormat.YUV420P) // YUV420 planar image format
{
    @Override
    public void onRead(List<ReadResult> list)
    {
        // Place Node at each ReadResult location
        for (ReadResult result : list)
        {
            performHitTest(result);
        }
    }

    @Override
    public void onError(ReaderError readerError)
    {
        // Handle error
    }
};
```

Subscribe to SceneView updates

Add a Scene.OnUpdateListener to the SceneView. The onUpdate() callback is invoked once per frame before the scene is updated.

Within the onUpdate() callback, retrieve the latest Camera image and convert it into ImageData for consumption by DMSDK. The following code snippet demonstrates how to obtain the latest camera image, convert it into ImageData, and process it with the VideoCaptureReader:

```
private void onUpdate(FrameTime frameTime) {
    // Get the latest frame
    Frame frame = arSceneView.getArFrame();

    // Get camera image that corresponds to frame
    Image image = frame.acquireCameraImage();

    // Convert to ImageData
    ImageData imageData = convertImage(image);
    image.close();

    try
    {
        // Pass ImageData to DMSDK
        videoCaptureReader.processImageFrame(imageData);
    } catch (ReaderException e)
    {
        // Handle error...
    }
    // ...
}

private ImageData convertImage(Image image) {
    Image.Plane[] planes = image.getPlanes();

    // Copy planes into internal buffer, so that we can relinquish
    // camera buffer back to the system
    for ( int i = 0; i < 3; i++ )
    {
        if ( planeBuffers[i] == null || planeBuffers[i].capacity() <
            planes[i].getBuffer().capacity() )
        {
            planeBuffers[i] = ByteBuffer.allocateDirect(
                planes[i].getBuffer().capacity() );
        }
        planeBuffers[i].clear();
        planeBuffers[i].put( planes[i].getBuffer() );
    }

    // Create ImageData
    return new ImageData<>( planeBuffers,
                           image.getWidth(),
                           image.getHeight(),
                           image.getWidth(),
                           1,
                           HelperCaptureFormat.YUV420P,
                           false );
}
```

Attach nodes to view

When a mark (Digimarc Barcode, 1D barcode or QR code) is detected, we can retrieve its coordinates from the `ReadResult.Metadata` class. The coordinates provided by DMSDK are in the camera's coordinate space. To display the correct position on the device's screen, we have to translate from the camera to view coordinate space. Once translated, we can test the center point of the bounding box for an intersection in the scene geometry:

```
private void performHitTest(ReadResult result) {
    // Get location coordinates
    Object obj =
    result.getMetadata().getValue(DataDictionary.BarcodeLocation);

    // Perform hit test on latest frame
    Frame frame = arSceneView.getArFrame();

    // Convert Point collection into float array, so that they can
    // be passed into transform convenience function
    List<Point> points = (List<Point>) obj;
    float inputVertices[] = new float[8];
    convertPointsToFloats(points, inputVertices);

    // Transform image coordinates to view coordinates
    float outputVertices[] = new float[8];
    frame.transformCoordinates2d(Coordinates2d.IMAGE_PIXELS, inputVertices,
    Coordinates2d.VIEW, outputVertices);

    // Test center point of coordinates for intersection on plane
    for (HitResult hit : frame.hitTest(midPoint[0], midPoint[1]))
    {
        Trackable trackable = hit.getTrackable();

        if ((trackable instanceof Plane &&
        ((Plane) trackable).isPoseInPolygon(hit.getHitPose()))
        {

            // Create Anchor at hit location
            Anchor anchor = hit.createAnchor();
            AnchorNode anchorNode = new AnchorNode( anchor );
            anchorNode.setParent( arSceneView.getScene() );

            // Create and add Node
            Node node = new Node();
            anchorNode.addChild( node);

        }
    }
}
```