What did i change ?

* Change 1 picture in slide 1
* Add 2 more slide 3,5
* Change RealityKit to ARKit in slide 5
* Delete slide 7 ( too long now)
* Change the order of slide 6

Introducing + people occlusion

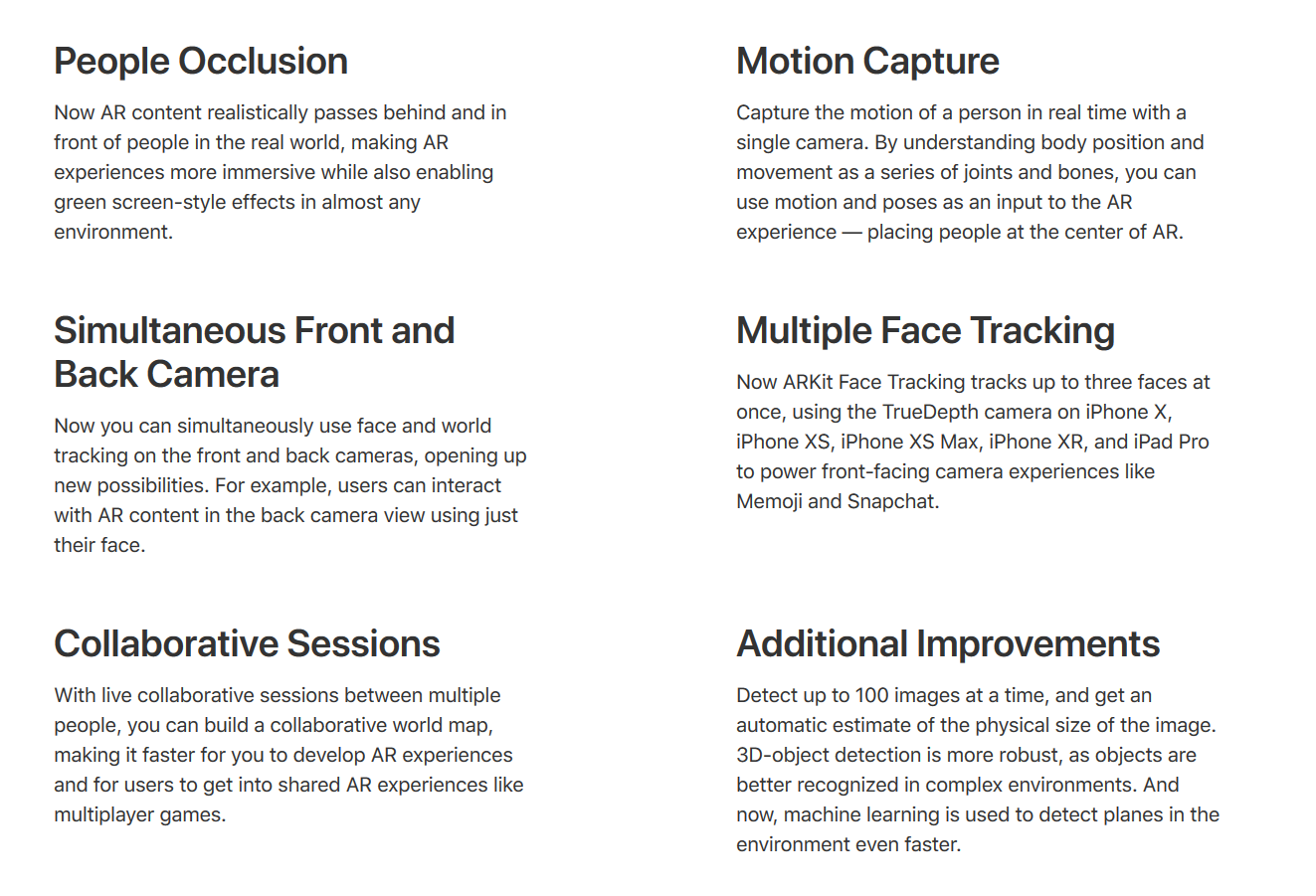
1. Slide

Hello every1 and wellcome to our presentation. My name is Trung, here are Andi and Luke. Today we will talk to you about Bringing People into AR

1. Slide

As all of you may already know what AR means, I will just go straight and show you what arkit can do.

By default the virtual contents cover everything in the camera feed. Now with the help of ARKit we can make people occlusive to the virtual content.



* Face tracking:…. And a little bit more information, the ARKit 3 now can track up to 3 faces at once

In this presentation we will focus on things: people occlusion and motion/body capture.

1. Slide

By default, virtual content covers anything in the camera feed. For example, when a person passes in front of a virtual object, the object is drawn on top of the person, which can break the illusion of the AR experience.

1. Slide

+

1. Slide

+

1. Slide

So how can we solve this depth-ordering problem.

And in order to understand how we gonna solve this, I gonna decompose this image.

In the image we can see that we have different things in different depth plane mixing real and rendered. And here we have each person in their own depth plane with the rendered object in between. For rendered content, the graphic “spylight???” already know exactly where It is simply by using a depth buffer.

And in order for us to do the same thing for the real content we need to understand where the people are in the scene.

In order to do this, we add 2 new buffers. The first buffer is segmentation buffer, it will show you where the people are in the scene. The second buffer is depth buffer, it will show you where that person is in depth.

now the amazing thing about this features is that the way we generating this buffers is by using machine learning in order to generate this buffers using only the camera image.

We would also like this buffer to be at the same resolution of the camera image. And in oder to do that, we need an additional processing called matting.

And now with the matted image we are able to correctly extract the people from the scene and together with the estimated depth data we can now put them in the correct depth plane. Finally letting us solve the depth odering problem.

1. Slide

That was a lot of technology, and all them are now packed into ARKit.

People occlusion is supported on Apple A12 and later devices. Before attempting to enable people occlusion, verify that the user’s device supports it.

If the user’s device supports people occlusion, enable it by adding the [personSegmentationWithDepth](https://developer.apple.com/documentation/arkit/arconfiguration/framesemantics/3194576-personsegmentationwithdepth) option to your configuration’s frame semantics.

Then, rerun the session to effect the configuration change.

1. Slide

the personSegmentationWithDepth option specifies that a person occludes a virtual object only when the person is closer to the camera than the virtual object.

Alternatively, the [personSegmentation](https://developer.apple.com/documentation/arkit/arconfiguration/framesemantics/3089125-personsegmentation) frame semantic gives you the option of always occluding virtual content with any people that ARKit perceives in the camera feed irrespective of depth. This technique is useful, for example, in green-screen scenarios.

To temporarily disable people occlusion, remove that option from your app’s [frameSemantics](https://developer.apple.com/documentation/arkit/arconfiguration/3089121-framesemantics).

Then, rerun your session to effect the configuration change.