# Fundamental concepts

Before diving into ARCore, it's helpful to understand a few fundamental concepts. Together, these concepts illustrate how ARCore enables experiences that can make virtual content appear to rest on real surfaces or be attached to real world locations.

Motion tracking

As your phone moves through the world, ARCore uses a process called simultaneous localization and mapping (https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping), or SLAM, to understand where the phone is relative to the world around it. ARCore detects visually distinct features in the captured camera image called **feature points** and uses these points to compute its change in location. The visual information is combined with inertial measurements from the device's IMU to estimate the **pose** (position and orientation) of the camera relative to the world over time.

By aligning the pose of the virtual camera that renders your 3D content with the pose of the device's camera provided by ARCore, developers are able to render virtual content from the correct perspective. The rendered virtual image can be overlaid on top of the image obtained from the device's camera, making it appear as if the virtual content is part of the real world.
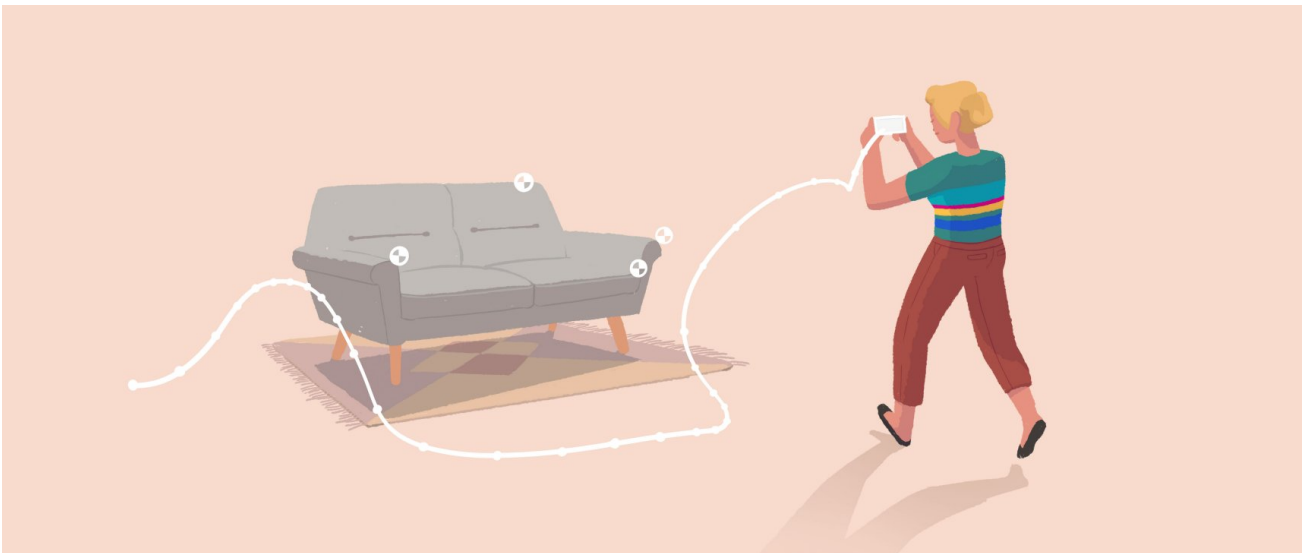

Environmental understanding

ARCore is constantly improving its understanding of the real world environment by detecting feature points and planes.

ARCore looks for clusters of feature points that appear to lie on common horizontal or vertical surfaces, like tables or walls, and makes these surfaces available to your app as geometric **planes**. ARCore can also determine the boundary of each geometric plane and make that information available to your app. You can use this information to place virtual objects resting on flat surfaces.

Because ARCore uses feature points to detect planes, flat surfaces without texture, such as a white wall, may not be detected properly.
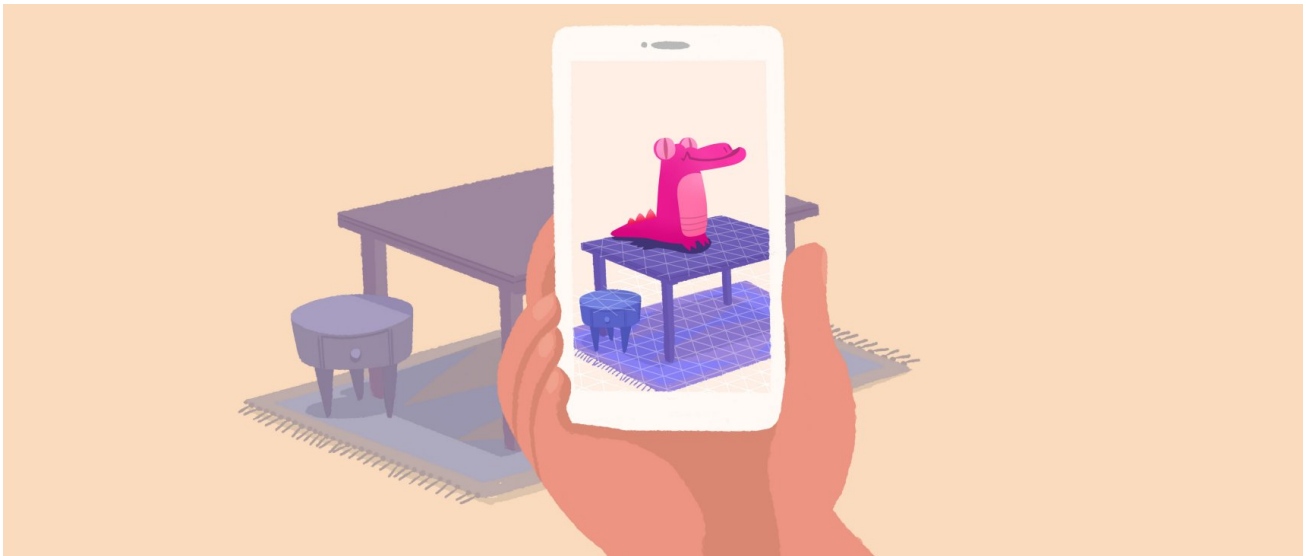

Depth understanding

ARCore can create depth maps, images that contain data about the distance between surfaces from a given point, using the main RGB camera from a <u>supported device</u> (/ar/discover/supported-devices). You can use the information provided by a depth map to enable immersive and realistic user experiences, such as making virtual objects accurately collide with observed surfaces, or making them appear in front of or behind real world objects.

Light estimation

ARCore can detect information about the lighting of its environment and provide you with the average intensity and color correction of a given camera image. This information lets

you light your virtual objects under the same conditions as the environment around them, increasing the sense of realism.
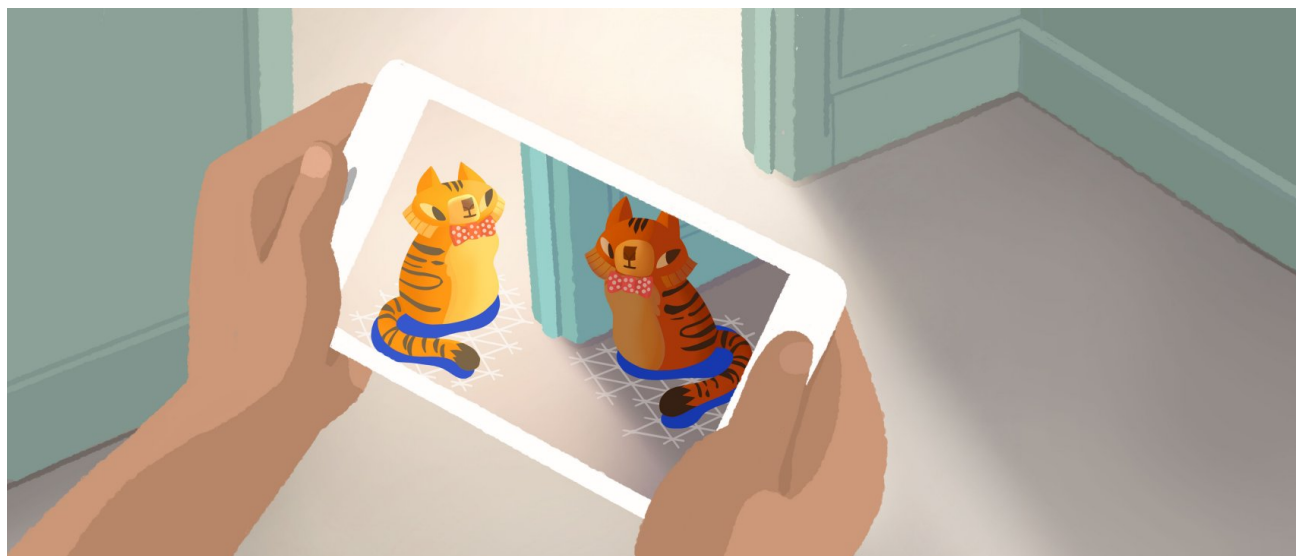
User interaction

ARCore uses hit testing to take an (x,y) coordinate corresponding to the phone's screen (provided by a tap or whatever other interaction you want your app to support) and projects a ray into the camera's view of the world, returning any geometric planes or feature points that the ray intersects, along with the pose of that intersection in world space. This allows users to select or otherwise interact with objects in the environment.

Oriented points

Oriented points lets you place virtual objects on angled surfaces. When you perform a hit test that returns a feature point, ARCore will look at nearby feature points and use those to attempt to estimate the angle of the surface at the given feature point. ARCore will then return a pose that takes that angle into account.

Because ARCore uses clusters of feature points to detect the surface's angle, surfaces without texture, such as a white wall, may not be detected properly.

Anchors and trackables

Poses can change as ARCore improves its understanding of its own position and its environment. When you want to place a virtual object, you need to define an **anchor** to ensure that ARCore tracks the object's position over time. Often times you create an anchor based on the pose returned by a hit test, as described in <u>user interaction</u> (#user_interaction).

The fact that poses can change means that ARCore may update the position of environmental objects like geometric planes and feature points over time. Planes and points are a special type of object called a **trackable**. Like the name suggests, these are objects that ARCore will track over time. You can anchor virtual objects to specific trackables to ensure that the relationship between your virtual object and the trackable remains stable even as the device moves around. This means that if you place a virtual Android figurine on your desk, if ARCore later adjusts the pose of the geometric plane associated with the desk, the Android figurine will still appear to stay on top of the table.

Note: To reduce CPU costs, reuse anchors when possible and detach anchors that you no longer need.

For more information, see <u>Working with anchors</u> (/ar/develop/anchors)

Augmented Images

Augmented Images is a feature that allows you to build AR apps that can respond to specific 2D images such as product packaging or movie posters. Users can trigger AR experiences when they point their phone's camera at specific images - for instance, they could point their phone's camera at a movie poster and have a character pop out and enact a scene.

ARCore also tracks moving images such as, for example, a billboard on the side of a moving bus.

Images can be compiled offline to create an image database, or individual images can be added in real time from the device. Once registered, ARCore will detect these images, the images' boundaries, and return a corresponding pose.

Sharing

ARCore Cloud Anchor API lets you create collaborative or multiplayer apps for Android and iOS devices.

With Cloud Anchors, one device sends an anchor and nearby feature points to the cloud for hosting. These anchors can be shared with other users on either Android or iOS devices in the same environment. This enables apps to render the same 3D objects attached to these anchors, letting users have the same AR experience simultaneously.

Learn more

Start putting these concepts into practice by building AR experiences on the platform of your choice.

- Quickstart: Android (/ar/develop/java/quickstart)

- Quickstart: Android NDK (/ar/develop/c/quickstart)

- Quickstart: iOS (/ar/develop/ios/cloud-anchors-quickstart-ios)

- Quickstart: Unreal (/ar/develop/unreal/quickstart)

Last updated 2022-02-01 UTC.