

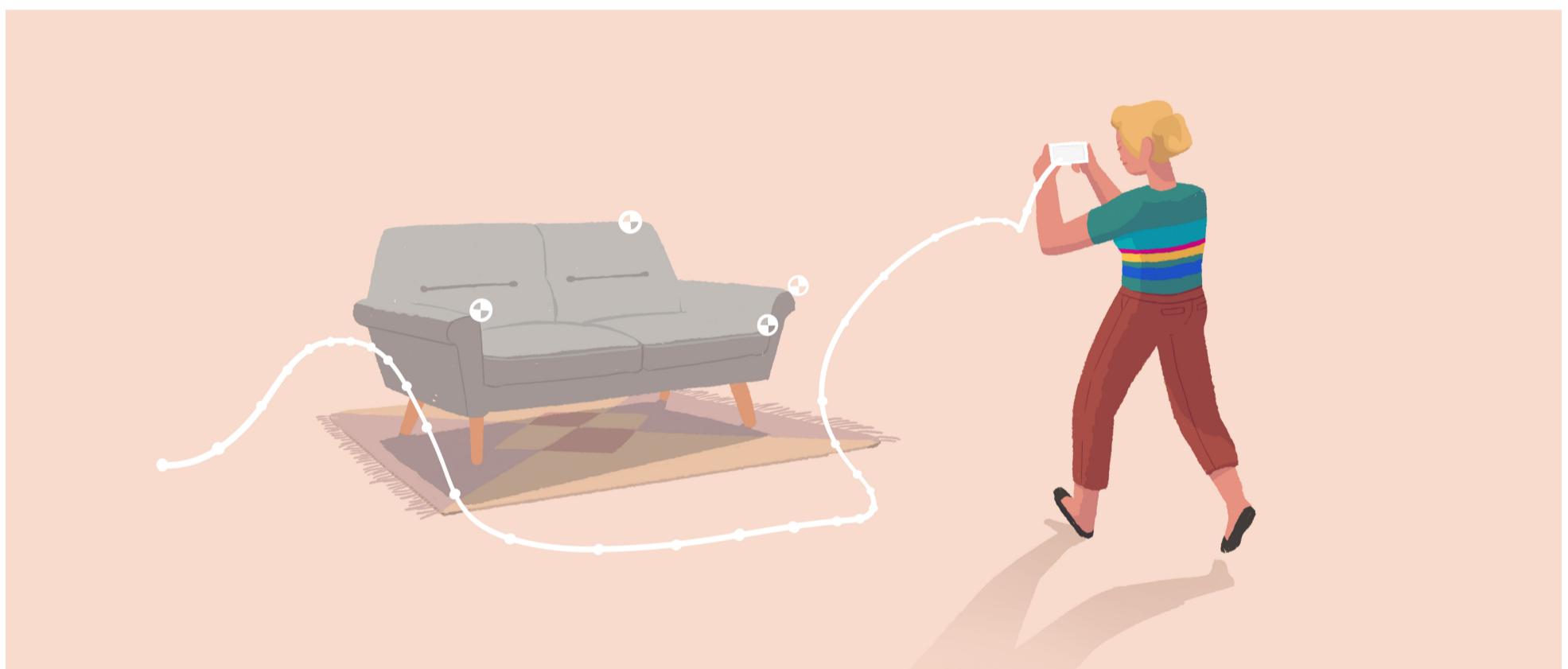
# Fundamental Concepts

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 concurrent odometry and mapping, or COM, 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 overlayed 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.

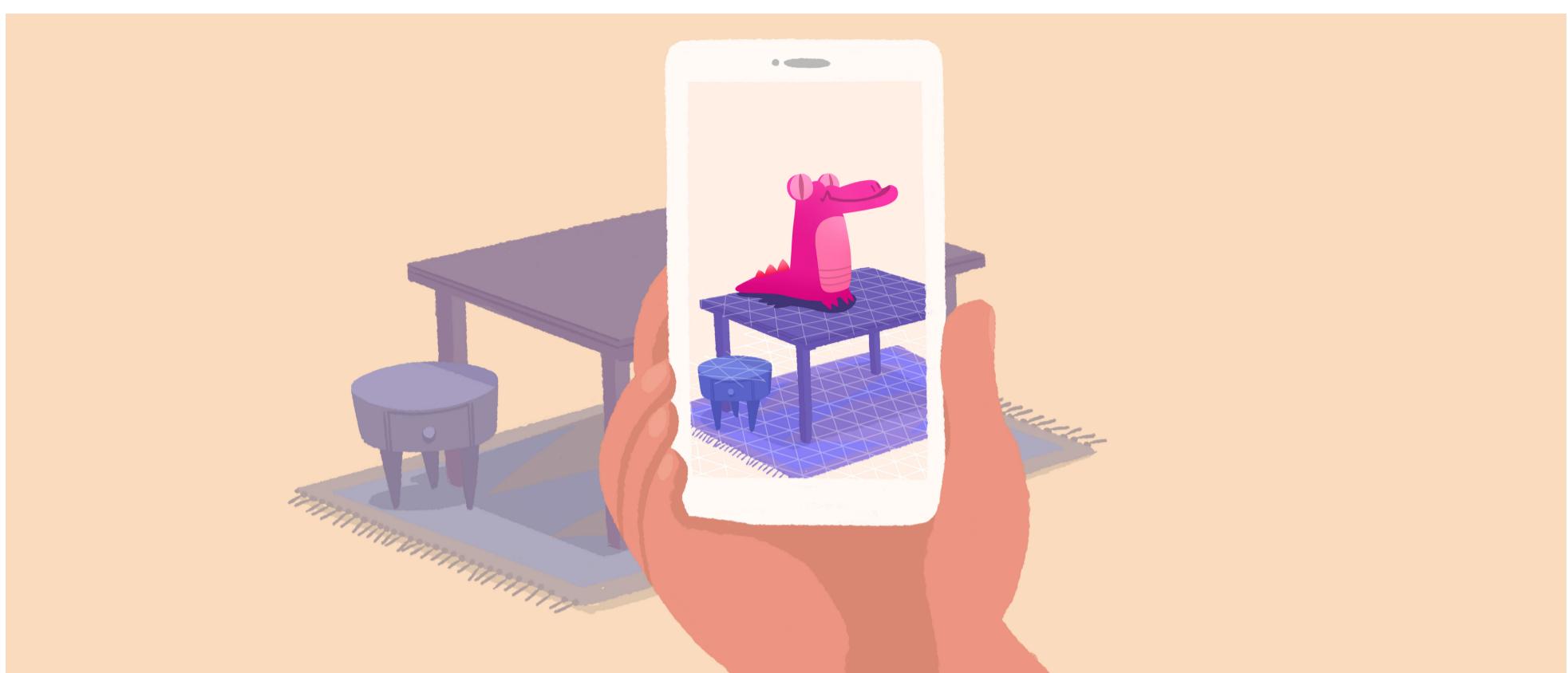


## Environmental understanding

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

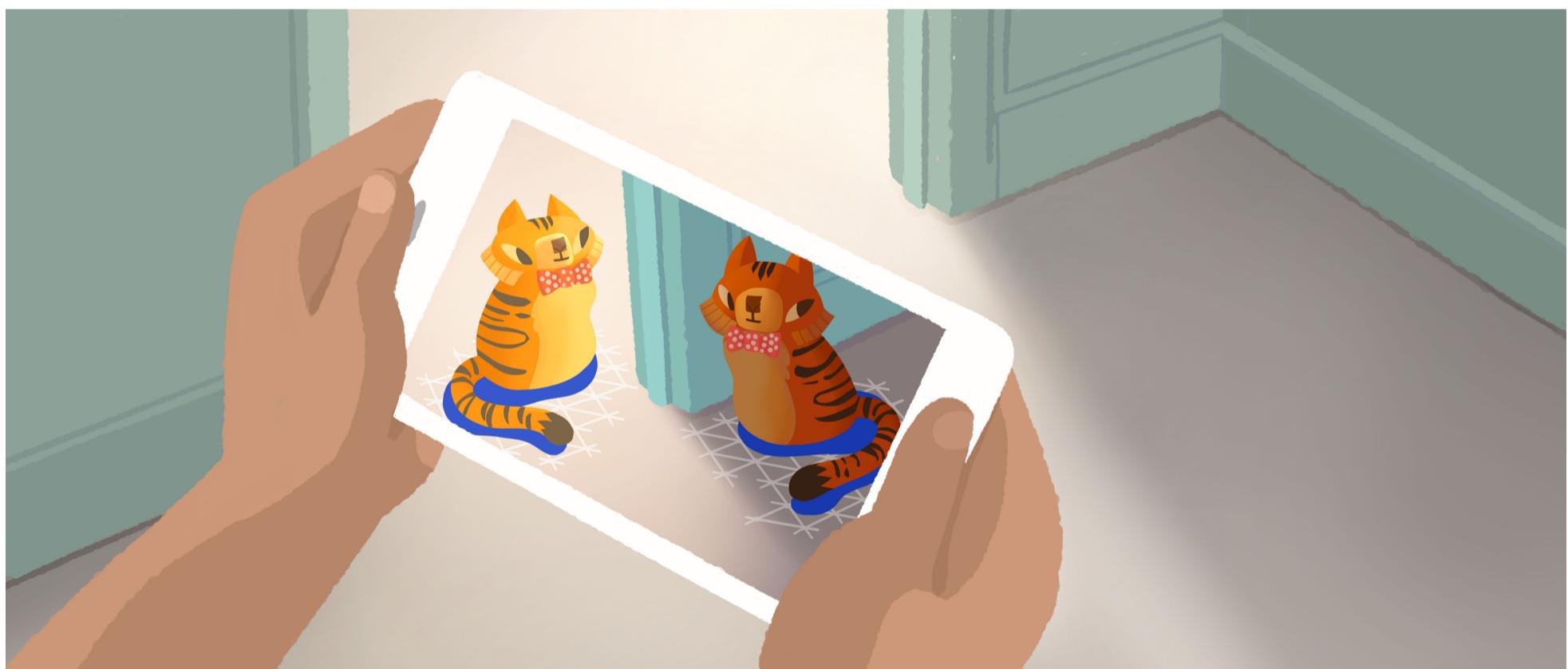
ARCore looks for clusters of feature points that appear to lie on common horizontal surfaces, like tables and desks, and makes these surfaces available to your app as **planes**. ARCore can also determine each plane's boundary and make that information available to your app. You can use this information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

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



## Light estimation

ARCore can detect information about the lighting of its environment and provide you with the average intensity 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 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.

## Anchoring objects

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 [user interaction](#) (#user\_interaction). This allows your virtual content to remain stable relative to the real world environment even as the device moves around.

## Learn more

---

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

- [Getting started with Android Studio](https://developers.google.com/ar/develop/java/getting-started) (<https://developers.google.com/ar/develop/java/getting-started>)
  - [Getting started with Unity](https://developers.google.com/ar/develop/unity/getting-started) (<https://developers.google.com/ar/develop/unity/getting-started>)
  - [Getting started with Unreal](https://developers.google.com/ar/develop/unreal/getting-started) (<https://developers.google.com/ar/develop/unreal/getting-started>)
  - [Getting started with Web](https://developers.google.com/ar/develop/web/getting-started) (<https://developers.google.com/ar/develop/web/getting-started>)
- 

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated August 28, 2017.*