

# 1. Motion tracking

ARCore usa un processo chiamato *Simultaneous localization and mapping (SLAM)* per determinare lo stato del dispositivo che si trova all'interno di un ambiente sconosciuto. Questo stato è descritto dalla sua posa (posizione e orientazione) che viene stimata attraverso prestazioni di odometria eccezionali e rilevazione di punti caratteristici. Con odometria si intende l'uso di dati ricavati da sensori di movimento che permettono di valutare il cambiamento della posizione nel tempo. Nel caso degli smartphone viene utilizzato il sensore IMU che rileva misure inerziali come la velocità, accelerazione e posizione. La rilevazione di punti caratteristici è l'individuazione di immagini con caratteristiche differenti che consentono al dispositivo di calcolare la sua posizione relativa. Questi punti di riferimento insieme alle misurazioni ricavate dai sensori permettono di avere una buona stima della posa e di ricavare la rappresentazione di una mappa dell'ambiente circostante. Tuttavia, il movimento sequenziale stimato dallo SLAM include un certo margine di errore che si accumula nel tempo causando una notevole deviazione dai valori reali. Una soluzione che può essere adottata per risolvere questo problema consiste nel considerare come punto di riferimento un luogo visitato in precedenza di cui si sono memorizzate le sue caratteristiche. Grazie alle informazioni di questo luogo è possibile minimizzare l'errore nella stima della posa.

I contenuti virtuali possono essere renderizzati nella giusta prospettiva allineando la posa della telecamera virtuale con quella calcolata da ARCore. Il contenuto virtuale sembra reale perché è sovrapposto all'immagine ottenuta dalla fotocamera del dispositivo.



## 2. Depth understanding

ARCore Depth API permette agli sviluppatori di generare mappe di profondità attraverso l'uso di algoritmi di profondità del movimento. Una mappa di profondità offre una visualizzazione in 3D del mondo reale, ogni pixel è associato alla distanza dalla scena e attraverso l'uso di colori differenti è possibile riconoscere quali aree dello spazio sono più vicine al dispositivo. Quando viene avviata una nuova sessione ARCore il display dello smartphone è nero, ma non appena si effettua un piccolo movimento la profondità viene rilevata. La stima della profondità è ricavata attraverso il movimento dello smartphone. Quando si progettano delle applicazioni che si concentrano sulla profondità bisogna considerare che la profondità viene calcolata meglio quando la scena rimane la stessa con piccoli spostamenti. Il dispositivo ha bisogno di muoversi un pò per generare la profondità. Un altro fattore da prendere in considerazione è quando l'utente compie lunghi spostamenti; in questo caso la stima della profondità arriva fino a 8 metri ma la migliore accuratezza si ha tra 0 e 5 metri.

Le principale funzionalità offerte da depth API sono tre:

- **Copertura dei contenuti:** permette di posizionare accuratamente dei contenuti virtuali di fronte o dietro degli oggetti reali.
- **Immersione:** permette di decorare una scena con oggetti virtuali che interagiscono tra di loro.
- **Interazione:** i contenuti virtuali sono in grado di interagire con il mondo reale attraverso cambiamenti fisici e collisioni.

### 2.1 Sessione ARCore con depth API

Prima di iniziare una nuova sessione ARCore è necessario controllare se il dispositivo supporta depth API. A volte questa opzione può essere disattivata oppure non supportata nonostante il dispositivo supporti ARCore. Dopo aver definito la sessione con le opportune configurazioni è possibile controllare se il dispositivo e la fotocamera supportano una determinata modalità di profondità invocando il metodo `isDepthModeSupported(Config.DepthMode mode)` sull'istanza della sessione. Se la modalità è supportata viene configurata la sessione e sarà possibile sfruttare depth API.

Esempio

```
val config = session.config

//Check whether the user's device supports the Depth API.
val isDepthSupported = session.isDepthModeSupported(Config.DepthMode.AUTOMATIC)
if (isDepthSupported) {
    config.depthMode = Config.DepthMode.AUTOMATIC
}
session.configure(config)
```