

NEARBY INTERACTION ZUR POSITIONSBESTIMMUNG

VON LUCA, LARS, ERIC, MAGNUS UND AHMED



PROJEKT

- Interdisziplinäres Projekt an der FH Aachen als Modul am FB 5
- Nutzung von UWB fähigen Chips zur Abstandsbestimmung zum iPhone
- Positionsbestimmung durch Trilateration
- Nutzung der Daten für 3D Audio-Rendering
- Anwendung z. B. im Museum für interaktive Ausstellungen

AGENDA

1. Nearby Interaction
2. Core Bluetooth
3. UWB Chip
4. Positionsbestimmung
5. 3D Audio
6. Live-Demo

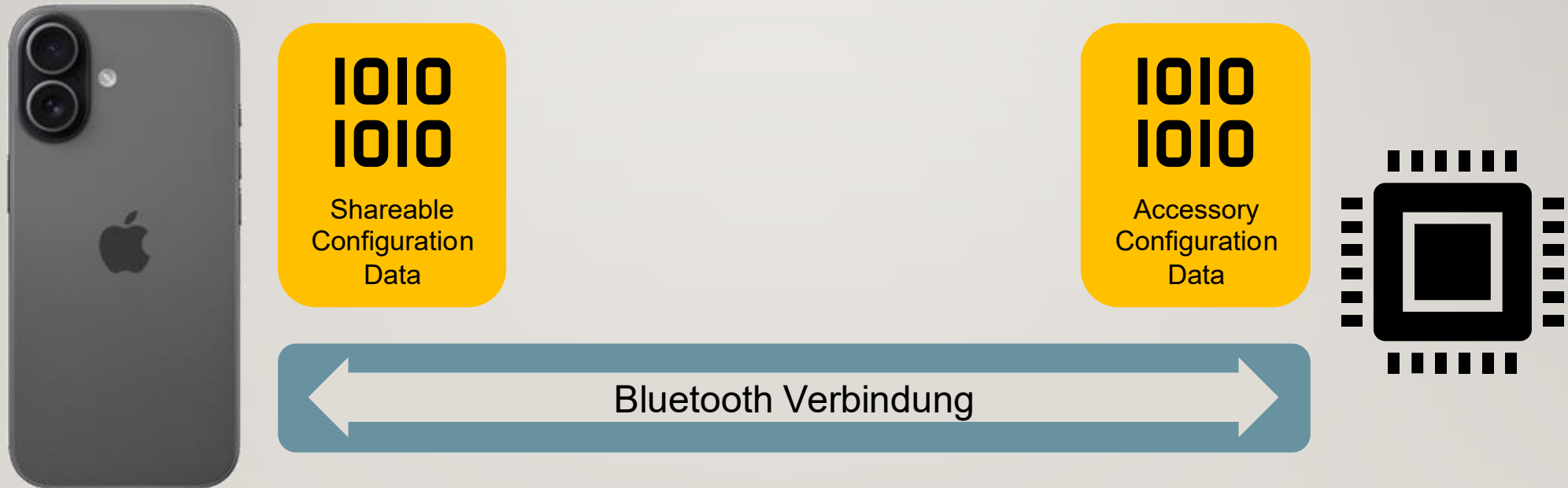
NEARBY INTERACTION



NEARBY INTERACTION

- Framework zur Nutzung und Verarbeitung der Empfangenen UWB-Daten
- Verwendung zur Air Tag Positionsbestimmung oder zur automatischen Fahrzeugöffnung per iPhone App
- Nutzbar auf iPhones mit mind. U1 Chip (ab iPhone 11)
- Liefert Abstands sowie Orientierungsdaten abhängig vom iPhone

NEARBY INTERACTION



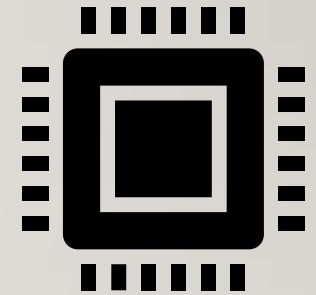
NEARBY INTERACTION



NISession

- `run(_:NIConfiguration)`
- `pause()`
- `invalidate()`

Bluetooth Verbindung



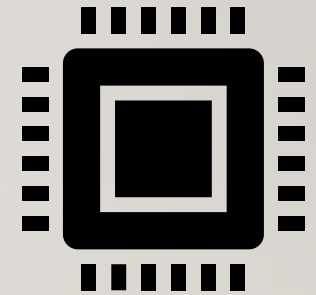
NEARBY INTERACTION



NISSessionDelegate

NISSession

Bluetooth Verbindung



NISESSION DELEGATE

- session(_:NISession, didGenerateShareableConfigurationData: Data, for: NINearbyObject)
- session(_: NISession, didUpdate: [NINearbyObject])
- session(_: NISession, didRemove:[NINearbyObject], reason:NINearbyObject.RemovalReason)
- sessionWasSuspended(_: NISession)
- sessionSuspensionEnded(_: NISession)
- session(_: NISession, didInvalidateWith: Error)

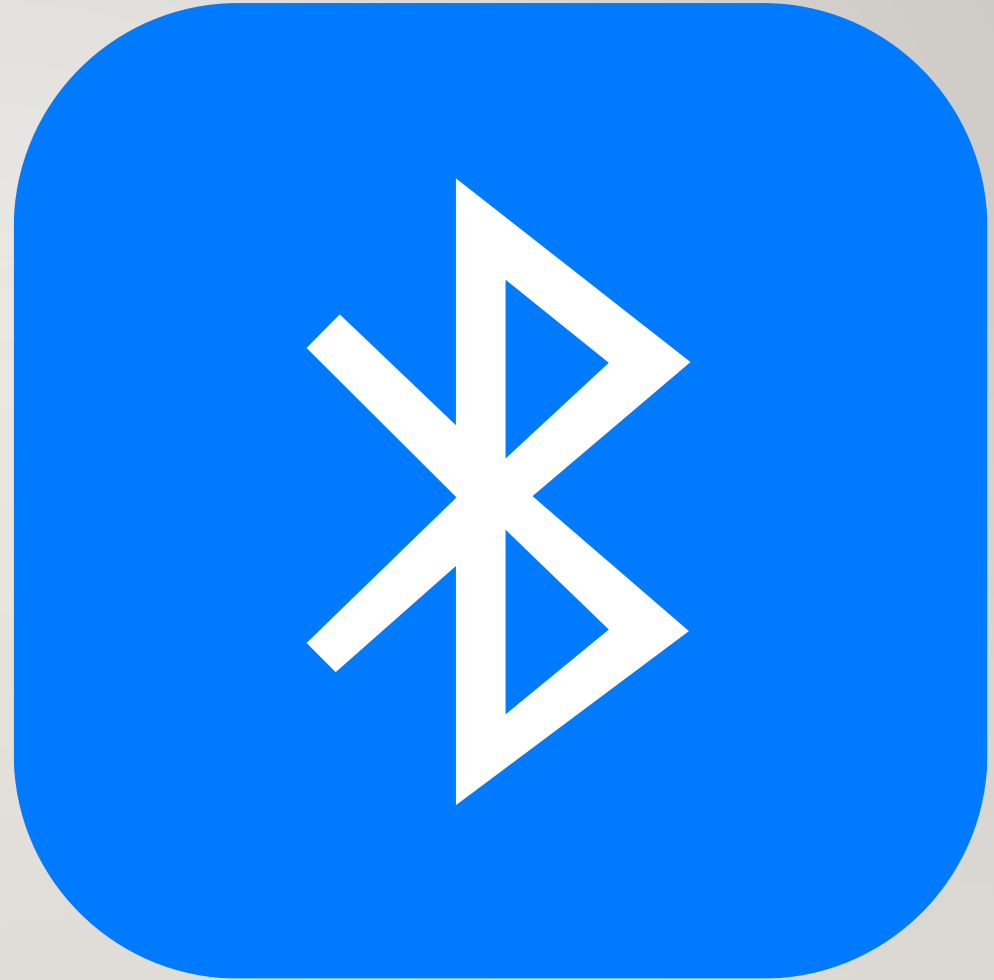
NINearbyObject

- discoveryToken: NIDiscoveryToken {get}
- distance: Float? {get}
- direction: simd3<float>? {get}
- horizotalAngle: Float? {get}
- verticalDirectionEstimate: verticaleDirectionEstimate {get}

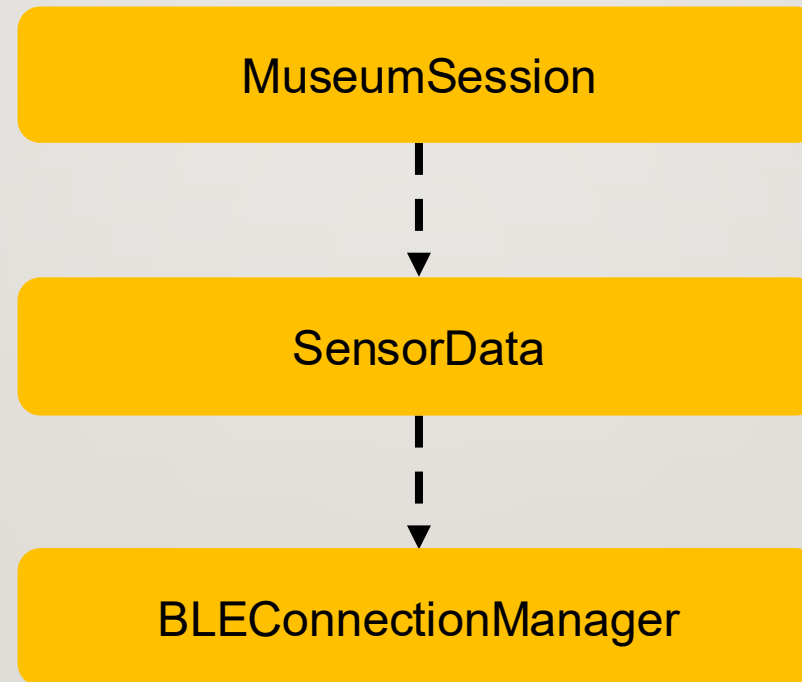
NISSESSION DELEGATE

- session(_:NISession, didGenerateShareableConfigurationData: Data, for: NINearbyObject)
- session(_: NISession, didUpdate: [NINearbyObject])
- session(_: NISession, didRemove:[NINearbyObject], reason:NINearbyObject.RemovalReason)
- sessionWasSuspended(_: NISession)
- sessionSuspensionEnded(_: NISession)
- session(_: NISession, didInvalidateWith: Error)

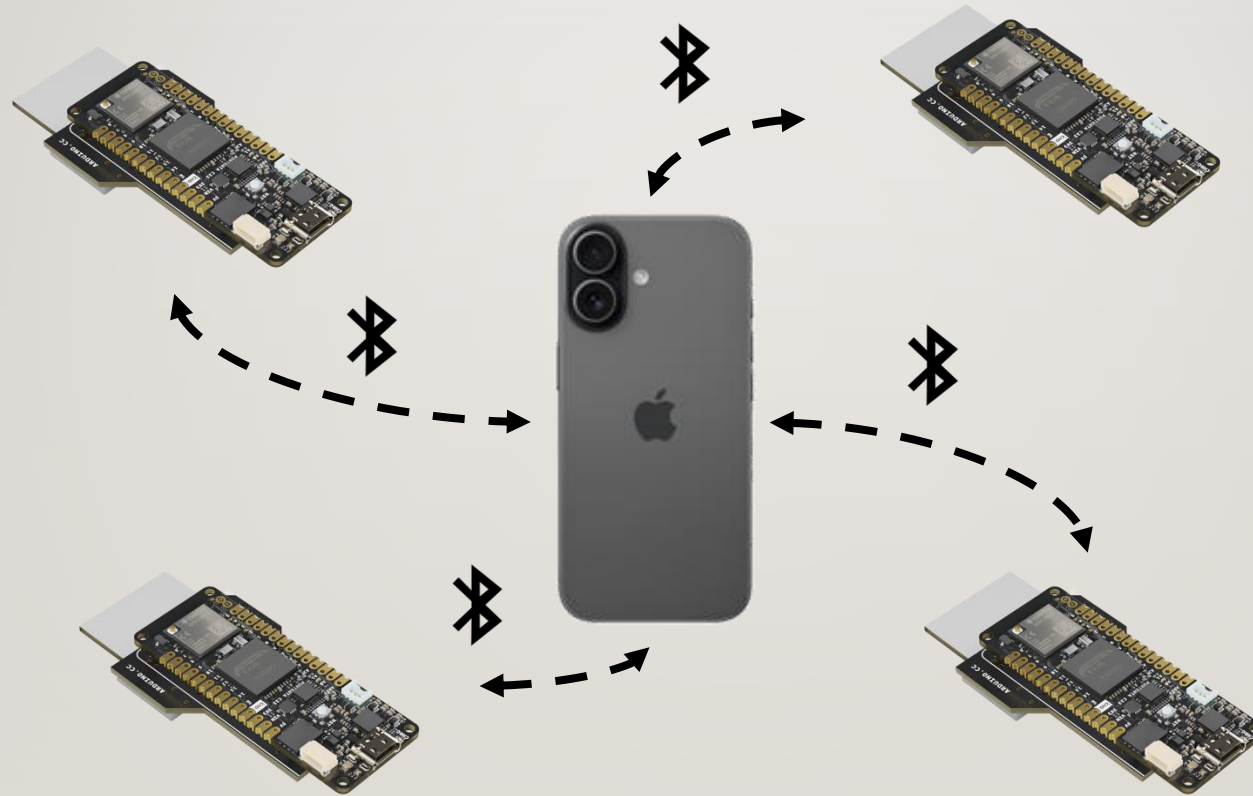
CORE BLUETOOTH



CORE BLUETOOTH



CORE BLUETOOTH



CORE BLUETOOTH



Scan



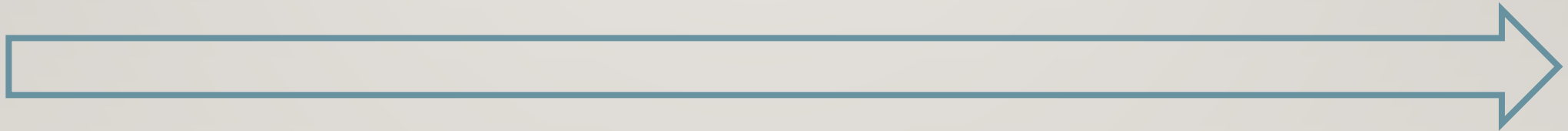
Connect



Handshake
(Session Tag)



UWB-Start



UWB CHIP



UWB CHIP

- Portenta C33 und Portenta UWB Shield
- Portenta UWB Shield Library
- Arduino BLE Library
- Nearby Interaction Accessory Protocol Specification
- Session management und Session Life cycle



SESSION MANAGEMENT

Library Code:

```
bool NearbySessionManager::addSession(NearbySession &sess)
{
    NearbySession *newSess = new NearbySession();
    newSess->sessionID(sess.sessionID());
    newSess->sessionType(sess.sessionType());
    newSess->bleDevice(sess.bleDevice());

    if (numSessions >= maxSessions)
        return false;

    sessions[numSessions++] = newSess; //&sess;
    return true;
}
```

```
protected:
    static const int maxSessions = 5;
    int numSessions;
    UWBSession* sessions[maxSessions];
    UWBSession emptySession;
};
```

SESSION MANAGEMENT

- In der API ist die libUWBShieldApi.a
- Diese Dateien enthalten schon kompilierten Maschinencode, der direkt in unser Programm eingebunden wird
- UWBHAL ist die hardwarenahe Schicht, die UWB-Sessions auf Chip-Ebene verwaltet

SESSION MANAGEMENT

Wir stoppen die UWB-Session und deinitialisieren sie explizit in der UWBHAL

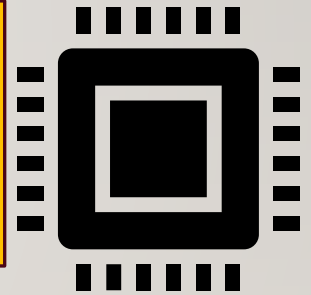
```
void clientDisconnected(BLEDevice dev) {  
  
    UWBSession sess = UWBNearbySessionManager.find(dev);  
  
    uint32_t handle = sess.sessionHandle();  
  
    sess.stop();  
    UWBHAL.sessionDeinit(handle);  
}
```

SESSION LIFE CYCLE



0x0A

0x01
Config Data

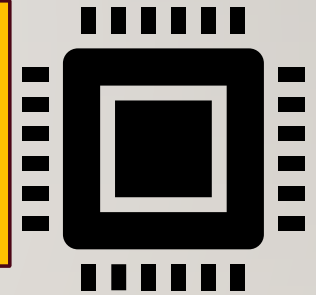


SESSION LIFE CYCLE



0x0B
Config Data Handy

0x02
Ranging Daten
austausch startet

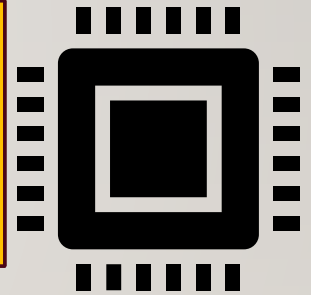


SESSION LIFE CYCLE



0x0C

0x03

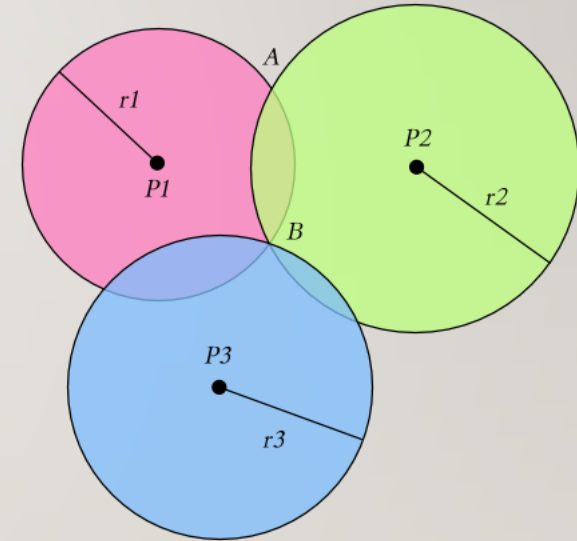


POSITIONS- BESTIMMUNG



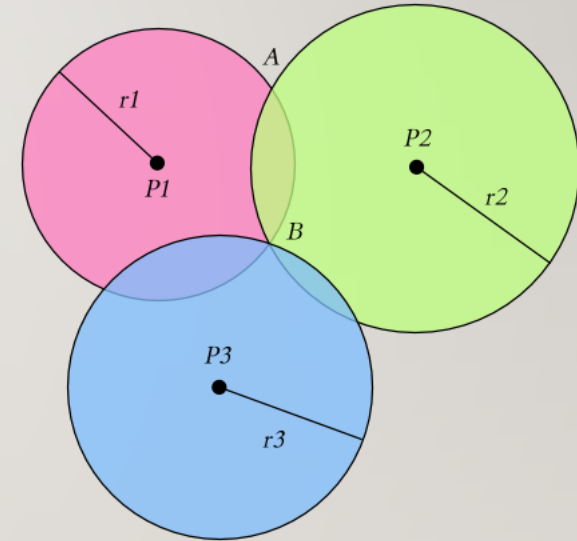
TRILATERATION

- Punkte der Sensoren sind bekannt
- Abstand wird gemessen
- Eindeutige Bestimmung ab 3 Messungen
- $r = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2 + (p_3 - x_3)^2}$



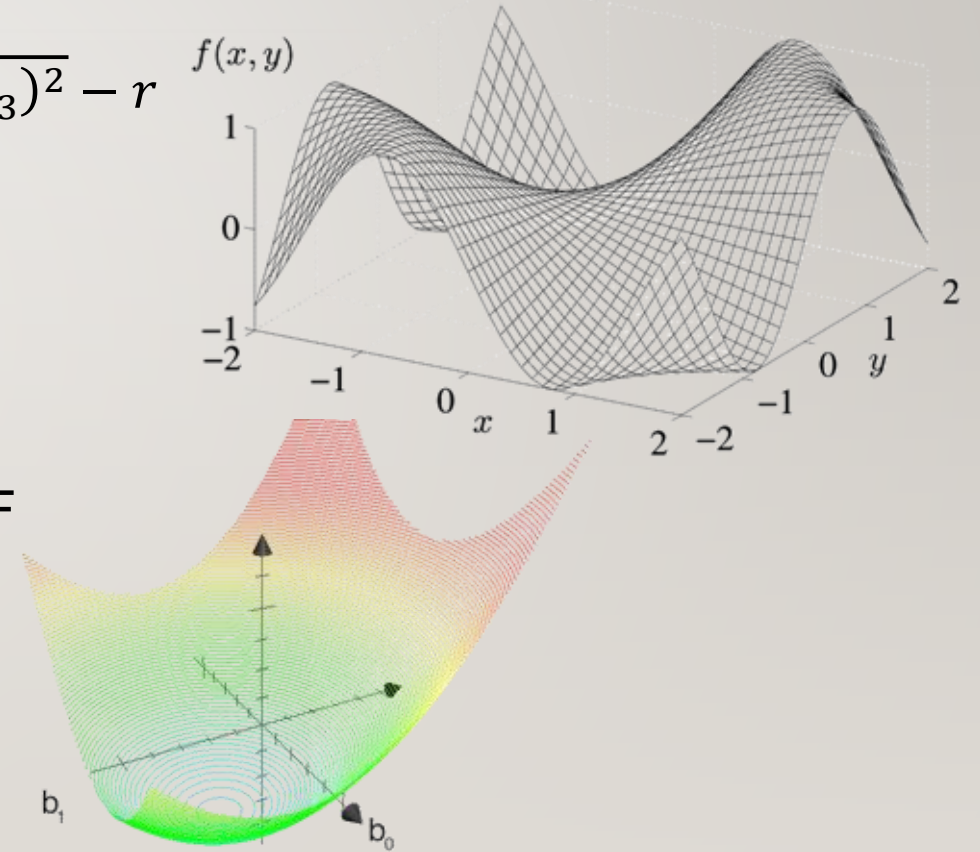
TRILATERATION

- Punkte der Sensoren sind bekannt
- Abstand wird gemessen
- Eindeutige Bestimmung ab 3 Messungen
- Messungen allerdings nicht exakt
- $$e(x_1, x_2, x_3) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2 + (p_3 - x_3)^2} - r$$



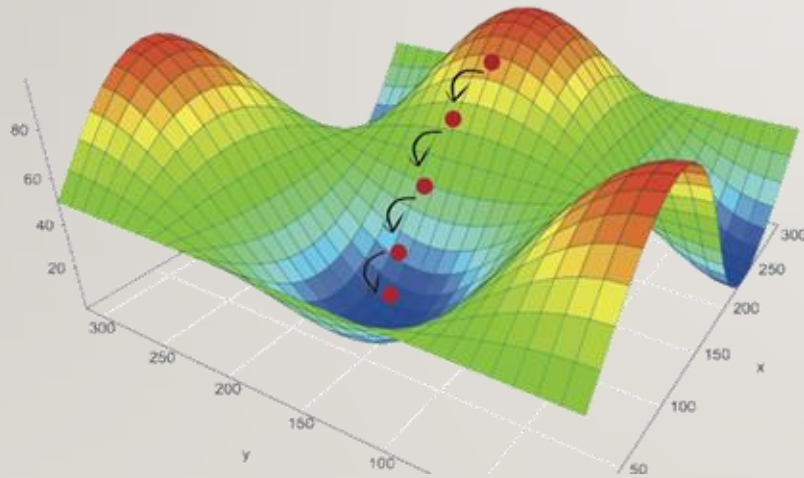
TRILATERATION

- $e(x_1, x_2, x_3) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2 + (p_3 - x_3)^2} - r$
- Summiere alle Fehlerquadrate zusammen
- $F(x_1, x_2, x_3) = \sum_i e_i(x_1, x_2, x_3)^2$
- Nehme an das Messfehler relativ klein sind
- Position des iPhones ist dann am Minimum von F

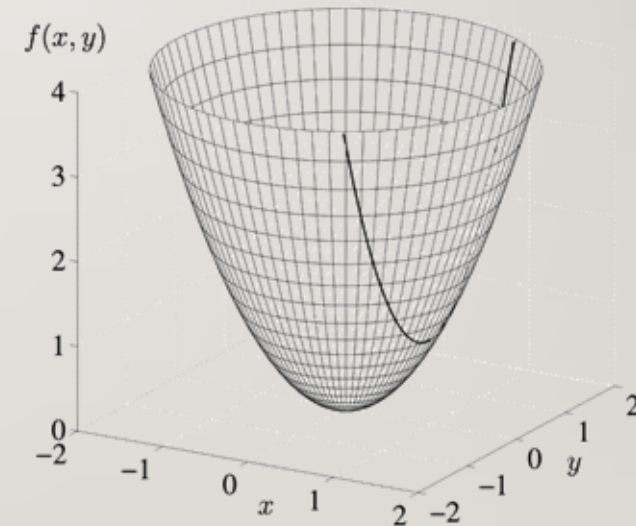


MINIMUM FINDEN - LEVENBERG-MARQUARDT- VERFAHREN

GRADIENTENVERFAHREN



GAUSS-NEWTON-VERFAHREN

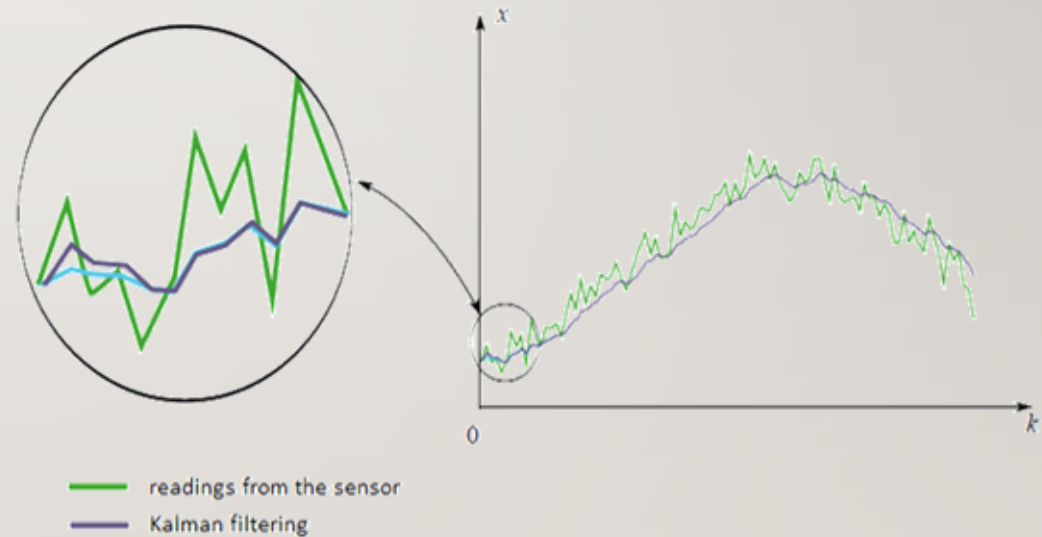


GEWICHTUNG VON MESSUNGEN

- Messungen mit großem Abstand -> ungenauer
- Messungen die länger her sind -> User hat sich evtl. weiter bewegt
- Jede Messung bekommt ein Gewicht, dass vom Abstand und dem Messzeitpunkt abhängt
- Somit stören potentiell falsche Messungen die Berechnung weniger

REDUZIERUNG VON RAUSCHEN – KALMAN FILTER

- Positionsdaten rauschen
- Tatsächliche Position + Ungewissheit
- Vorhersage für nächste Position
- Kombination mit Messung der nächsten Position
- Dadurch Dämpfung des Rauschens

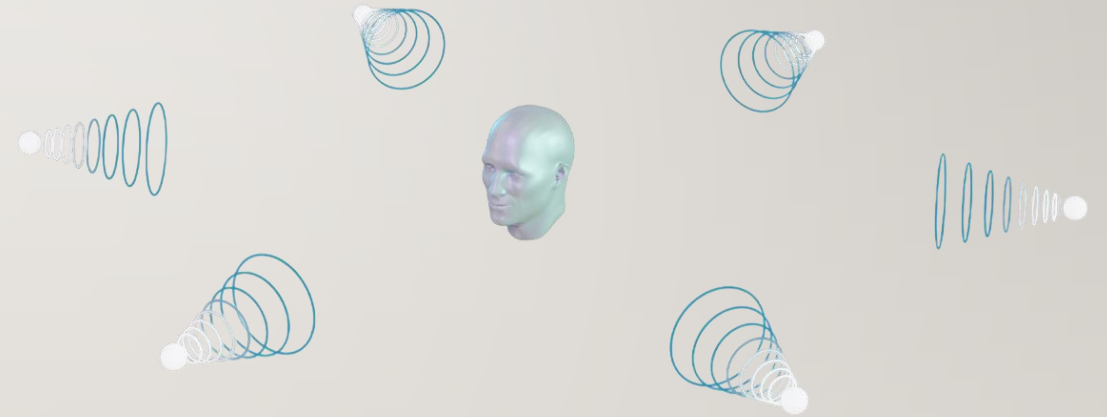


SPATIAL AUDIO



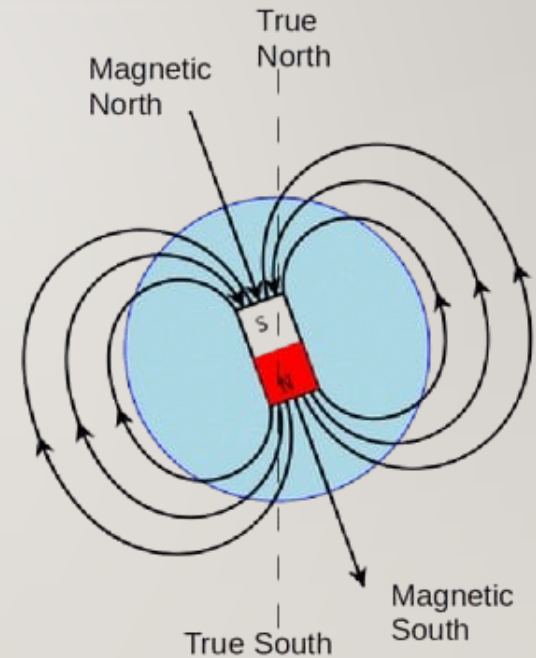
SPATIAL AUDIO

- Kopf/Handy Ausrichtung
- Virtueller Raum
- Audio Quellen



CORE LOCATION

- Interner Location Manager
 - .startUpdatingHeading
- locationManager(_ : CLLocationManager, didUpdateHeading: CLHeading)
 - CLHeading -> true und magnetic Heading



AVAUDIO

- Virtueller Raum -> AVAudioEnvironmentNode

AVAUDIO

- **AVAudioEnvironmentNode**
- Handy Position
- Handy Ausrichtung
- Rendering Modus
- Parameter wie Raum Klang

AVAUDIO

- Virtueller Raum -> AVAudioEnvironmentNode
- Audio Quellen -> AVAudioPlayerNode

AVAUDIO

- **AVAudioPlayerNode**
- Position der Audio Quelle
- Audio Files laden und abspielen
- Analoge Lautstärkeregelung
- Automatische Lautstärkeregelung

AVAUDIO

- Virtueller Raum -> AVAudioEnvironmentNode
- Audio Quellen -> AVAudioPlayerNode
- Audio Engine -> AVAudioEngine

AVAUDIO

- **AVAudioEngine**
- Environment hinzufügen
- Audio Quellen hinzufügen
- Audio starten und stoppen

AVAUDIO

- Virtueller Raum -> AVAudioEnvironmentNode
- Audio Quellen -> AVAudioPlayerNode
- Audio Engine -> AVAudioEngine
- Realität -> Raum Ausrichtung

LIVE DEMO

