

# Mobile Augmented Reality

Semantische Bauwerksmodelle als  
Datengrundlage einer Smartphone-basierten  
Augmented Reality Anwendung

Christoph Blut

RWTH Aachen University  
Geodätisches Institut  
Lehrstuhl für Bauinformatik & Geoinformationssysteme

# Inhalt

---

- Augmented Reality
- Mobile Augmented Reality System
- Aufbau des mobilen AR-Systems
- Demo

# Augmented Reality (AR)



Quelle: Wikitude    Quelle: Nokia    Quelle: Mercedes    Quelle: BMW

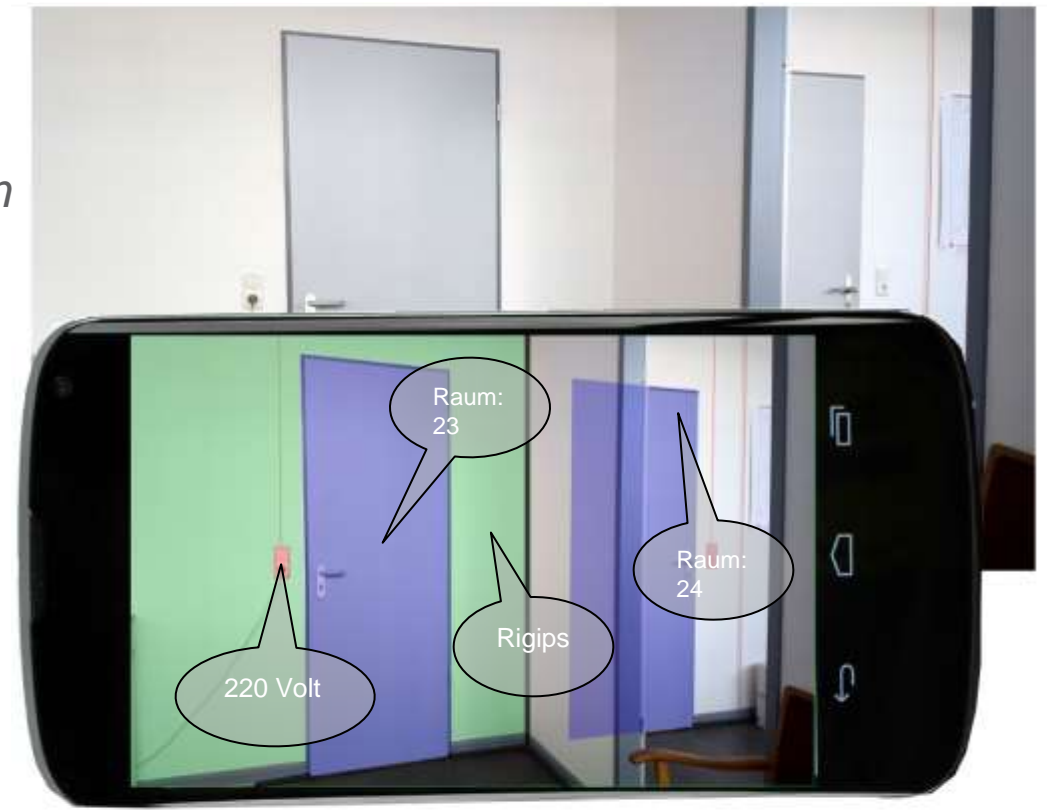
# Mobile AR-System - Motivation

- Semantische digitale Bauwerksmodelle
  - *Visualisierung geplanter / historischer Bauwerke*
  - *Abruf von Semantik / Topologie*



# Mobile AR-System - Motivation

- Semantische digitale Bauwerksmodelle (Innenraum)
  - *Visualisierung verdeckter Objekte (X-Ray)*
  - *Abruf von Objektinformationen*



# Mobile AR-System - Anforderungen

- Flexibel
- Unabhängig
- Low-Cost



# Mobile AR-System - Hardware

## ➤ Nexus 5

- Android 5.1
- 4,95"-Full-HD-IPS Display (445ppi)
- 8 MP Kamera mit optischer Bildstabilisierung
- 32 GB interner Speicher
- 2 GB RAM
- Sensoren
  - *GPS*
  - *Gyroskop*
  - *Beschleunigungsmesser*
  - *Kompass*
  - *Barometer*
  - *Halleffekt*

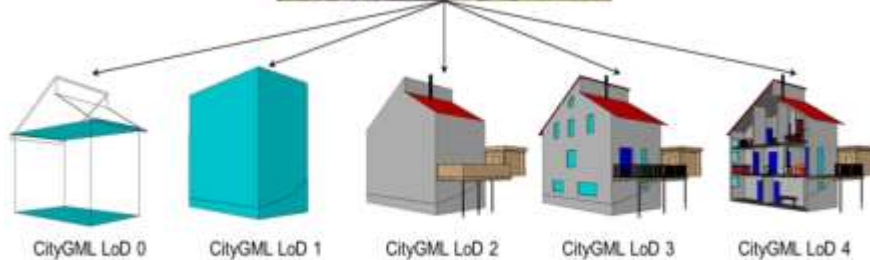


Quelle: Google

# Mobile AR-System - Datenbasis

## ➤ City Geography Markup Language (CityGML)

- Mehrskalige Visualisierung durch Level of Detail (LoD) System
- Semantik
- Topologie



Source: Karlsruhe Institute of Technology



# CityGML

```

<bldg:WallSurface gml:id=„WallSurface_1“>
...
<bldg:lod4MultiSurface>
  <gml:MultiSurface>
    <gml:surfaceMember>
      <gml:Polygon>
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList count="5" srsDimension="3">
              6.176 3.714 -1.202
            ...
          </gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    ...
  
```



# Mobile AR-System - Komponenten

- Komponenten des AR-Systems
  - Datenverarbeitung
    - *Datenimport*
    - *Datenspeicherung*
    - *Datenabfrage*
  - Datenvisualisierung
    - *Datenaufbereitung*
    - *Rendering*
  - Tracking
    - *Tracking im realen 3D-Raum*
    - *Positionierung und Ausrichtung im virtuellen 3D-Raum*

# Mobile AR-System - Komponenten

- Komponenten des AR-Systems
  - Datenverarbeitung
    - *Datenimport*
    - *Datenspeicherung*
    - *Datenabfrage*
  - Datenvisualisierung
    - *Datenaufbereitung*
    - *Rendering*
  - Tracking
    - *Tracking im realen 3D-Raum*
    - *Positionierung und Ausrichtung im virtuellen 3D-Raum*

# Datenverarbeitung - CityGML

## ➤ Datenimport

### ➤ CityGML-Parser (*Android Pull-XML-Parser*)

- *Sequentielle Verarbeitung der CityGML-Daten vs. DOM*

## ➤ Datenspeicherung

### ➤ SpatiaLite (*Erweiterung von SQLite; Serverlose, transaktionale SQL-Datenbankengine*)

- *Unterstützung von Geometrietypen wie Punkte, Linien und Polygonen und komplexeren Typen wie multiplen Polygonen, im 2D, als auch im 3D-Raum*

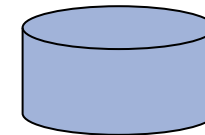
## ➤ Datenabfrage

### ➤ Verwendung von SQL für positionsabhängige Datenabfragen

- *Abruf von Geometrien*
- *Abruf von ergänzende Objektinformationen*

CityGML-Daten

*Android Pull-XML-Parser*



SpatiaLite-DB

SQL

Virtuelle Welt

# Mobile AR-System - Komponenten

## ➤ Komponenten des AR-Systems

### ➤ Datenverarbeitung

- *Datenimport*
- *Datenspeicherung*
- *Datenabfrage*

### ➤ Datenvisualisierung

- *Datenaufbereitung*
- *Rendering*

### ➤ Tracking

- *Tracking im realen 3D-Raum*
- *Positionierung und Ausrichtung im virtuellen 3D-Raum*

## ➤ CityGML

- Polygone sind über Punkte definiert
  - *Graphics Processing Unit (GPU) kann nur Dreiecke zeichnen*
  - *CityGML Polygone müssen daher trianguliert werden*

```

<gml:surfaceMember>
  <gml:Polygon gml:id="PolyID58718_509_420914_99840">
    <gml:exterior>
      <gml:LinearRing gml:id="PolyID58718_509_420914_99840_0">
        <gml:pos>458877 5438358.492403877 3.91317591116654</gml:pos>
        <gml:pos>458877.12 5438358.492403877 3.91317591116654</gml:pos>
        <gml:pos>458877.12 5438358.498097349 3.95642212862617</gml:pos>
        <gml:pos>458877 5438358.498097349 3.95642212862617</gml:pos>
        <gml:pos>458877 5438358.492403877 3.91317591116654</gml:pos>
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
</gml:surfaceMember>

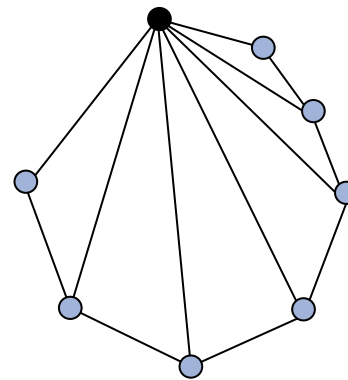
```

# Datenvisualisierung – Triangulierung

## ➤ Ablauf (konvexer Fall)

- Geordnete Punkte der Reihe nach verbinden
- Beliebigen Punkt wählen und mit allen anderen Punkten verbinden, außer den direkten Nachbarn

Polygon: *PolyID58718\_509\_420914\_99840*



konvex

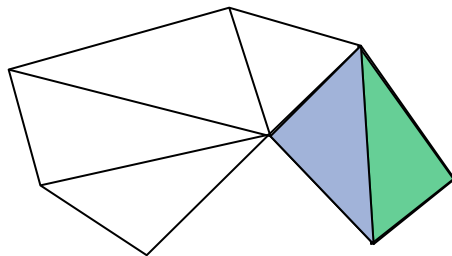
# Datenvisualisierung - Triangulierung

## ➤ Ear Clipping Algorithmus

### ➤ Pseudo-Code

```

while (points.length > 3)
  earTip = findEarTip()
  cutEarTip(earTip)
  classifyPoint(prevNeighbour(earTip))
  classifyPoint(nextNeighbour(earTip))
  
```



konkav

Speicher



`findEarTip`: Finde den ersten konvexen Punkt, für den gilt: Kein anderer Punkt liegt im Dreieck, das sich aus dem Punkt und seinen Nachbarn zusammensetzt. Ignoriere dabei alle konvexen Punkte, da diese nicht im Dreieck liegen können.

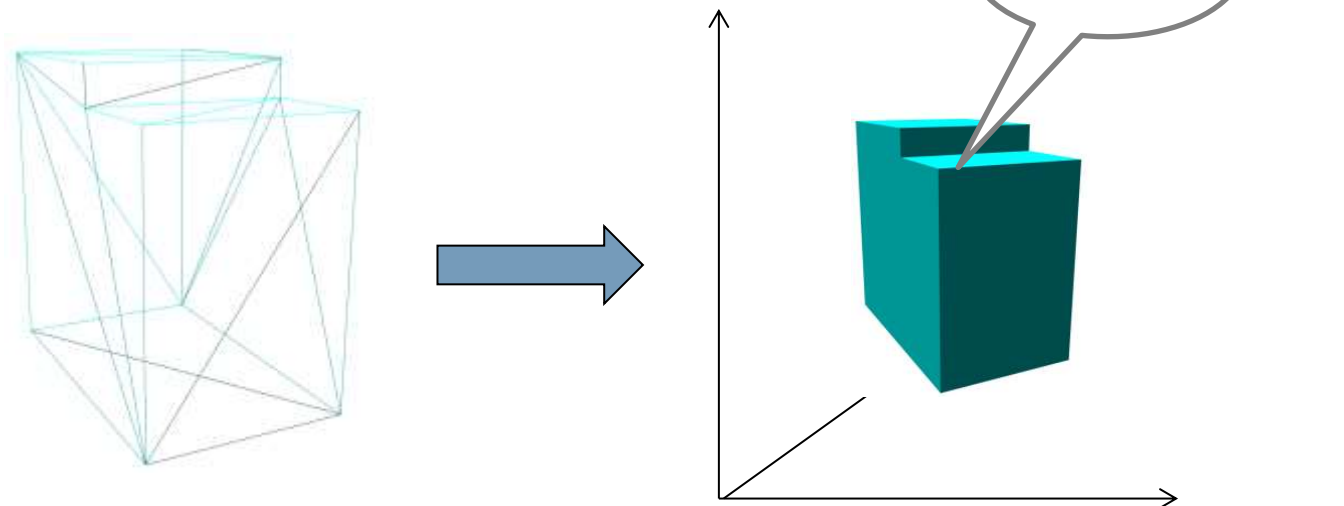
`cutEarTip(point)`: Entferne das Ohr vom Polygon und speichere das Dreieck, das sich aus dem Punkt und seinen Nachbarn bildet.

`classifyPoint(point)`: Prüfe ob der Punkt konvex, konkav oder kollinear ist.  
Gleicher Test, wie oben beschrieben.

# Datenvisualisierung – Rendering

## ➤ Visualisierung im virtuellen 3D-Raum

- Open Graphics Library for Embedded Systems (*OpenGL ES*)
  - Zusammenfassung aller Objekt-Polygone
  - Platzieren der triangulierten Polygone im virtuellen Koordinatensystem
- Verknüpfung der Objekte mit den ergänzenden Informationen





# Datenvisualisierung – Rendering

## ➤ Testdaten (CityGML)

### ➤ LoD2-Modell (*StädteRegion Aachen*)

- 8 Städte und 2 Gemeinden
- 707,1 km<sup>2</sup>
- 249277 Gebäude und 110494 Gebäudeteile
- 3058391 Polygone



Quelle: TUBS

# Datenvisualisierung – Rendering

Ausschnitt LoD2-Modell (*StädteRegion Aachen*) mit Standardfarbe



Quelle der Modelldaten: StädteRegion Aachen

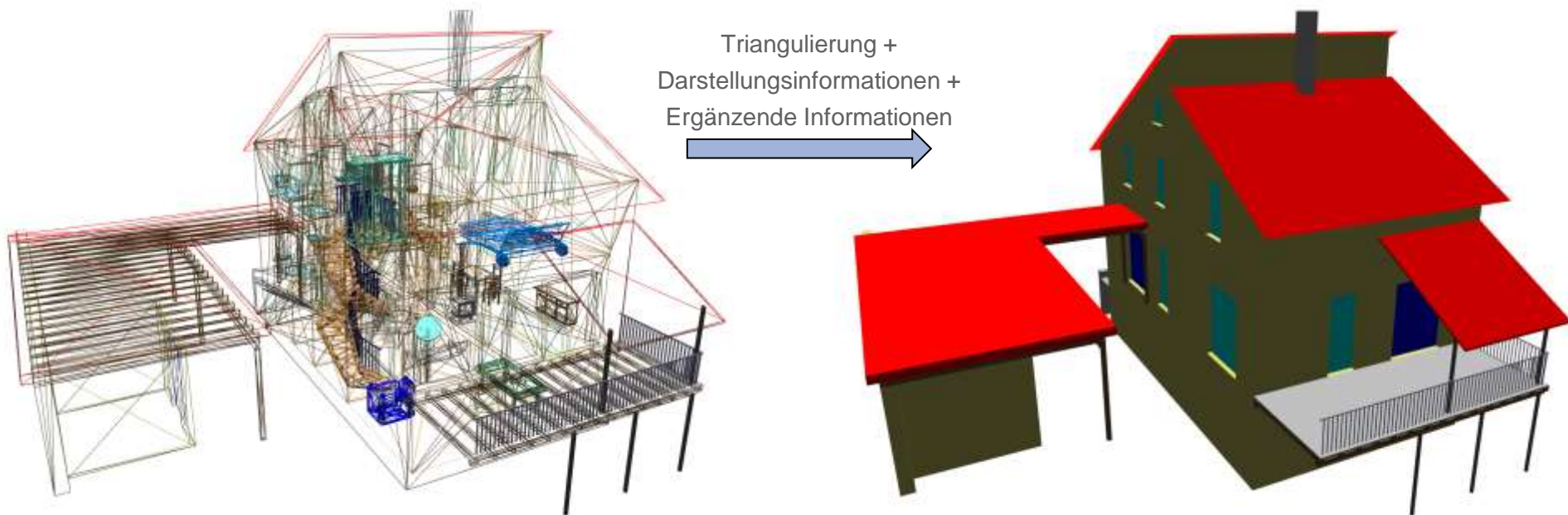
## ➤ Darstellungsinformationen

### ➤ Extensible 3D (X3D)-Material

- *Liefert Darstellungsinformationen für Polygone*
- *Quelle der Informationen: CityGML-Datei*

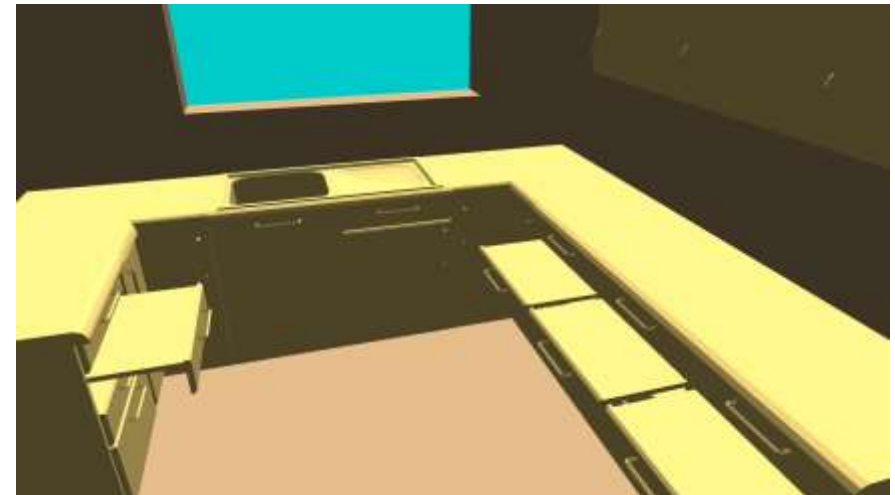
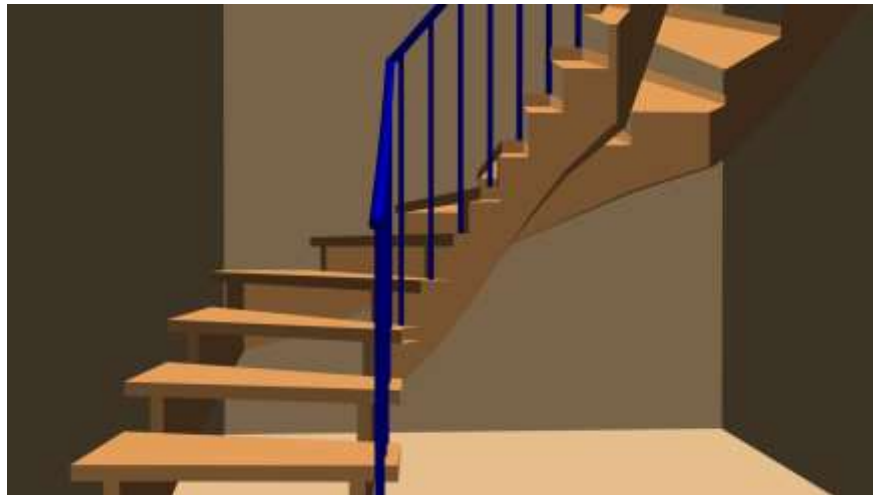
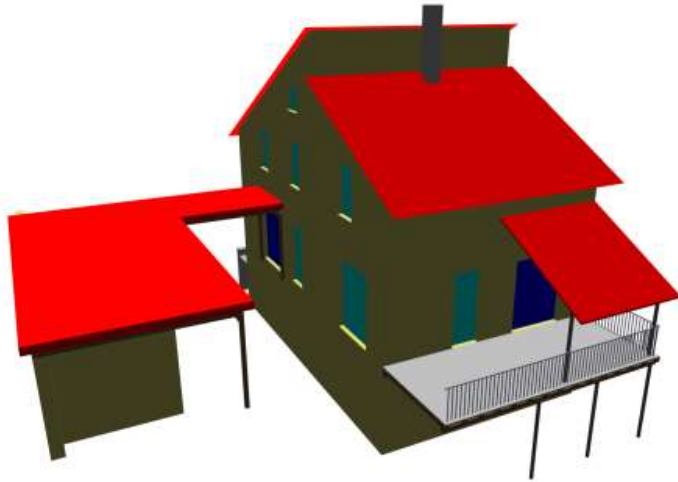
```
<app:appearanceMember>
  <app:Appearance>
    <app:surfaceDataMember>
      <app:X3DMaterial>
        <app:diffuseColor>0.95703125 0.75 0.44921875</app:diffuseColor>
        <app:emissiveColor>0 0 0</app:emissiveColor>
        <app:specularColor>1 1 1</app:specularColor>
        <app:transparency>1.0</app:transparency>
        <app:target>#PolyID58718_509_420914_99840</app:target>
        <app:target>#PolyID58803_371_698036_77126</app:target>
      </app:X3DMaterial>
    </app:surfaceDataMember>
  </app:Appearance>
</app:appearanceMember>
```

# Datenvisualisierung – Rendering



Quelle der Modelldaten: Karlsruher Institut für Technologie

# Datenvisualisierung – Rendering



Quelle der Modelldaten: Karlsruher Institut für Technologie

# Mobile AR-System - Komponenten

## ➤ Komponenten des AR-Systems

### ➤ Datenverarbeitung

- *Datenimport*
- *Datenspeicherung*
- *Datenabfrage*

### ➤ Datenvisualisierung

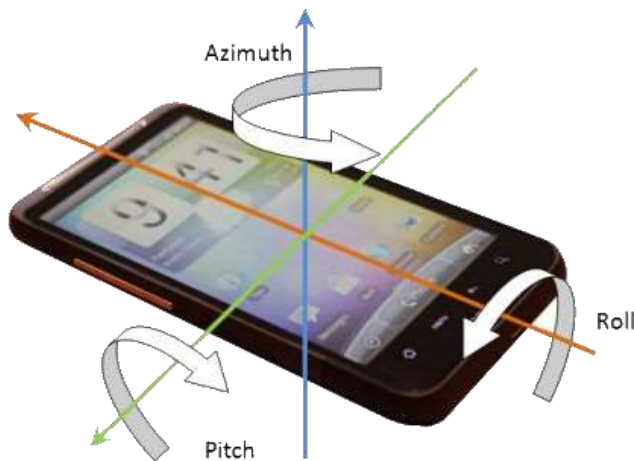
- *Datenaufbereitung*
- *Rendering*

### ➤ Tracking

- *Tracking im realen 3D-Raum*
- *Positionierung und Ausrichtung im virtuellen 3D-Raum*

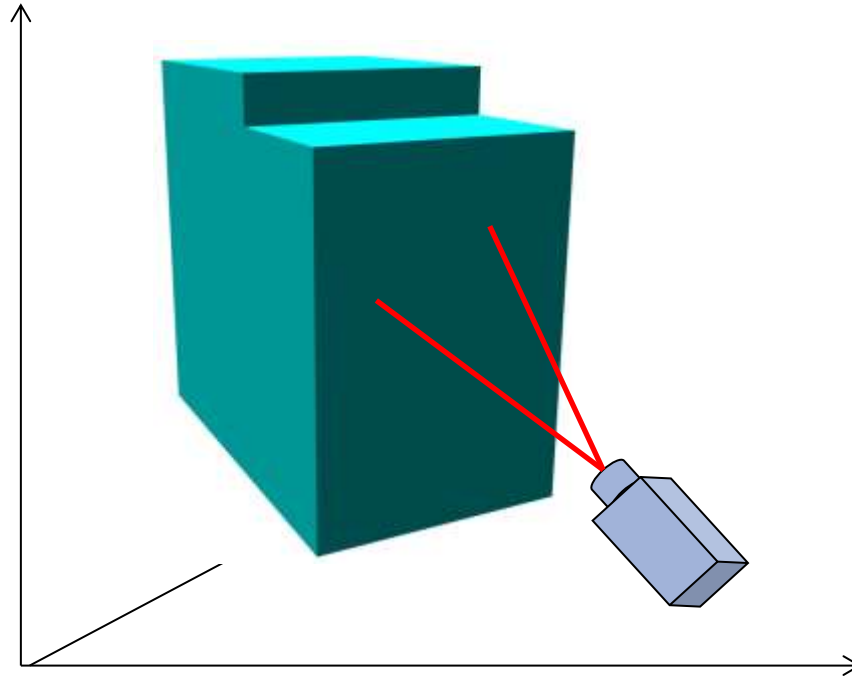
# Tracking

- Ermittlung der Realweltkoordinaten/  
Orientierung des Smartphones
- Positionierung per GPS
- Orientierung über Smartphone interne Sensoren



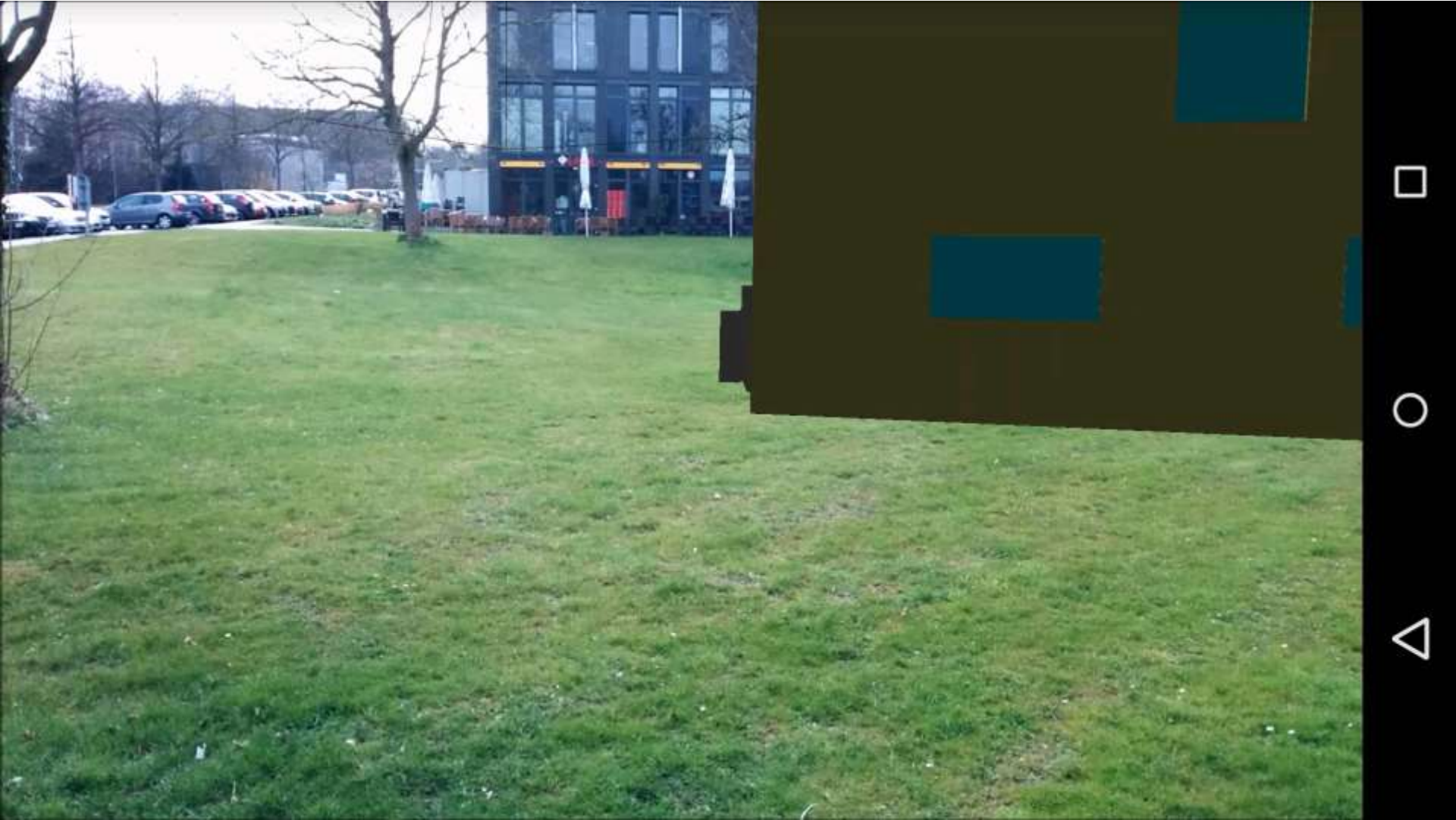
# Tracking

- Positionierung und Orientierung der Kamera in der virtuellen 3D-Welt





# AR-System - Demo





Danke für die Aufmerksamkeit

