

Context Based Computing

Prof. Dr. Grischa Schmiedl
DI Kerstin Blumenstein
Fachhochschule St. Pölten



Versuch einer Definition

Context Based Computing bedeutet:

- dem richtigen User
- zur richtigen Zeit
- aktuell relevante (und richtige) Information
- In einem an die Situation angepassten Format zu geben.

26. Juni 2008



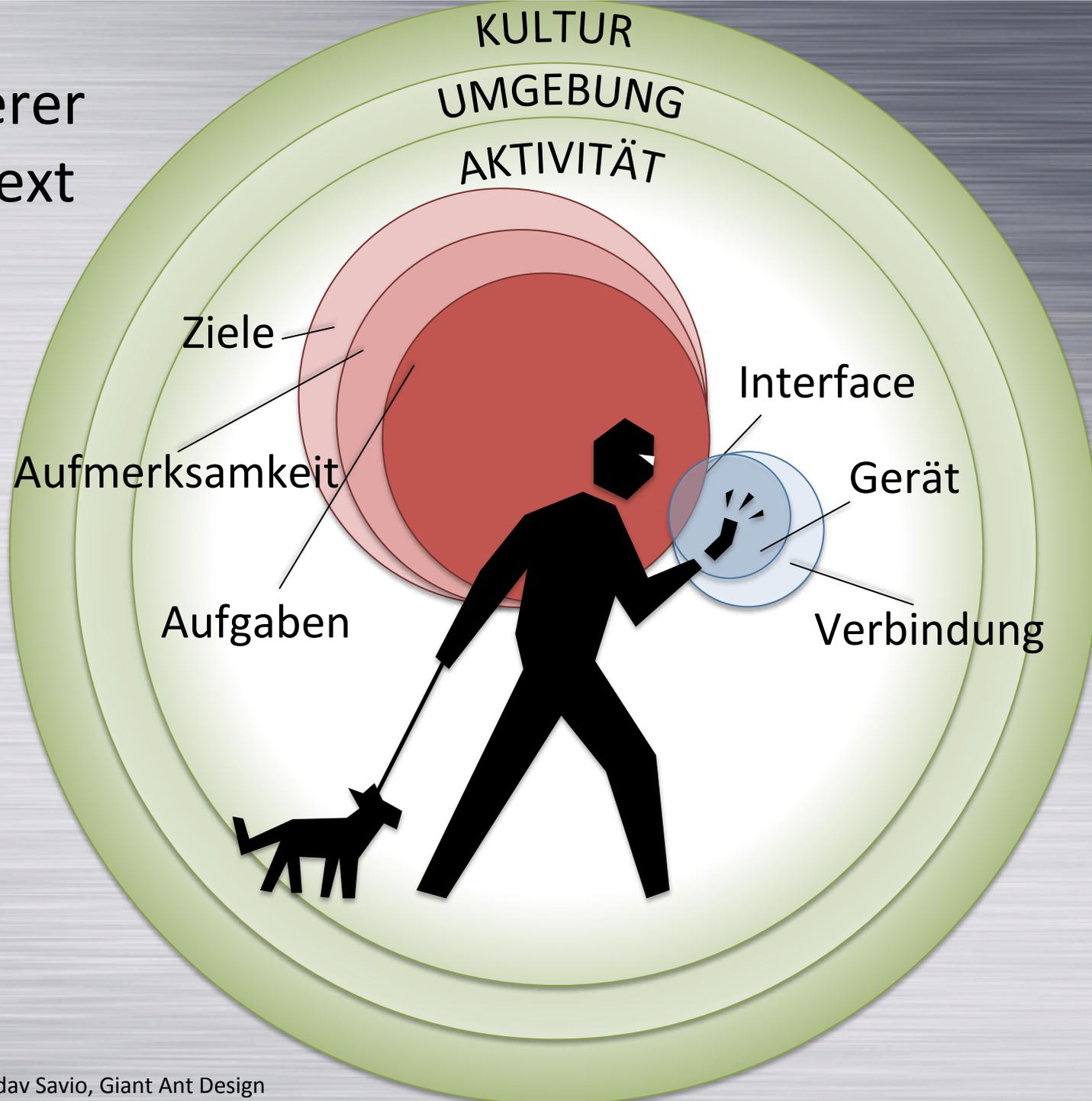


Da bin ich





Äußerer
Kontext



DWX

@developer_week #dwx13

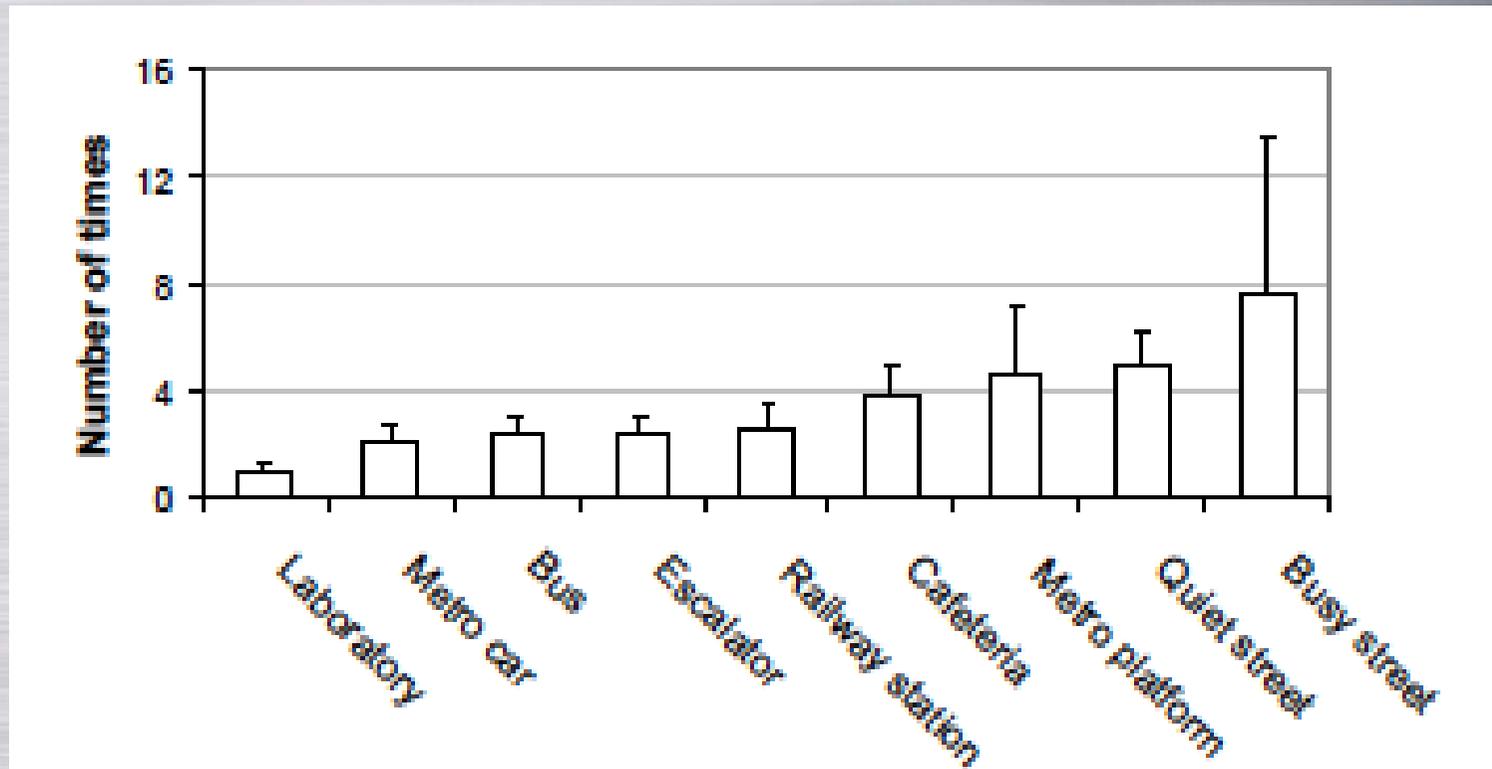
Heute: 2 Spezialthemen

1. Welchen Einfluss hat der „äußere“ Kontext (ob wir es wollen oder nicht) auf die User Experience?
 - Primäraktivität des Users (Ablenkung)
 - Einfluss der Umgebung (Environment)

2. Sensoren zur Ermittlung des äußeren Kontexts
 - Von den Werten zum Wissen

Aktivität und Umgebung

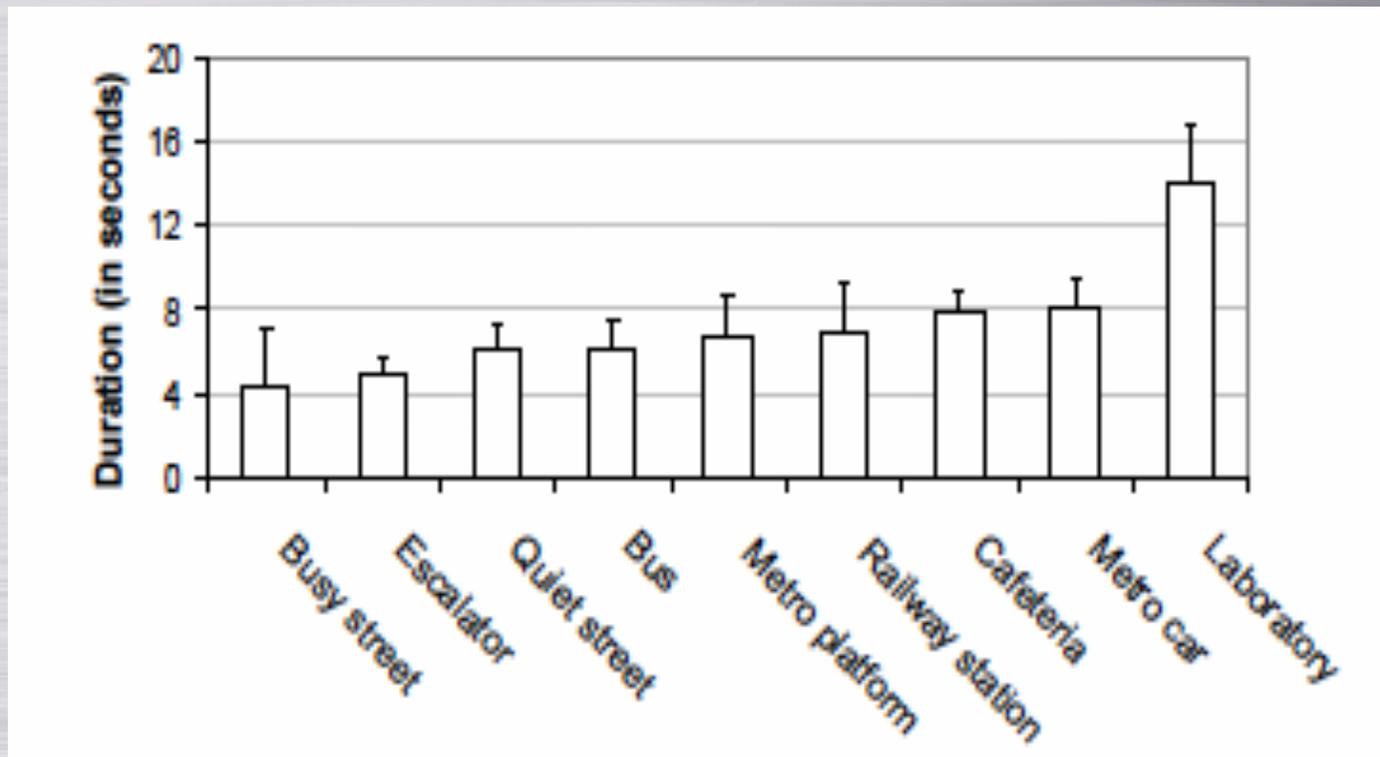
Aufmerksamkeitswechsel zur Primäraktivität



Source: Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J

Aufmerksamkeitsdauer

Ununterbrochene Aufmerksamkeit für das Handy



Source: Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J

Das Aufmerksamkeitsproblem

- Wenn Apps unterwegs benutzt werden, dann wird die Durchführung eines Use-Cases in mehrere “Attention-Units” zerteilt.
- Die Anzahl dieser “Attention-Units” und die durchschnittliche Zeit einer einzelnen Unit hängt von der Aktivität des Users und der Umgebung ab.
- Dauert eine “Attention-Unit” länger als die verfügbare Aufmerksamkeitsdauer, so wird die Applikation unbenutzbar.

Wie kann man das testen?

- Wie kann man eine realistische primäre Useraktivität „simulieren“?
- Wie kann man den User während dieser Aktivität beobachten und damit den Einfluss des Kontexts messen?
- Wie kann man den Kontext der Umgebung in einem mobilen Szenario reproduzieren?
- Welche Daten sollen gesammelt werden und wie interpretiert man diese?

Versuche, mobile Usability zu testen...



Equipment used by Google Source: Schusteritsch et al. (2007)

Weitere Versuche...



Equipment used by Oulasvirta et al. (2005)

Und noch ein paar...



Equipment used by Reichl et al. (2007)

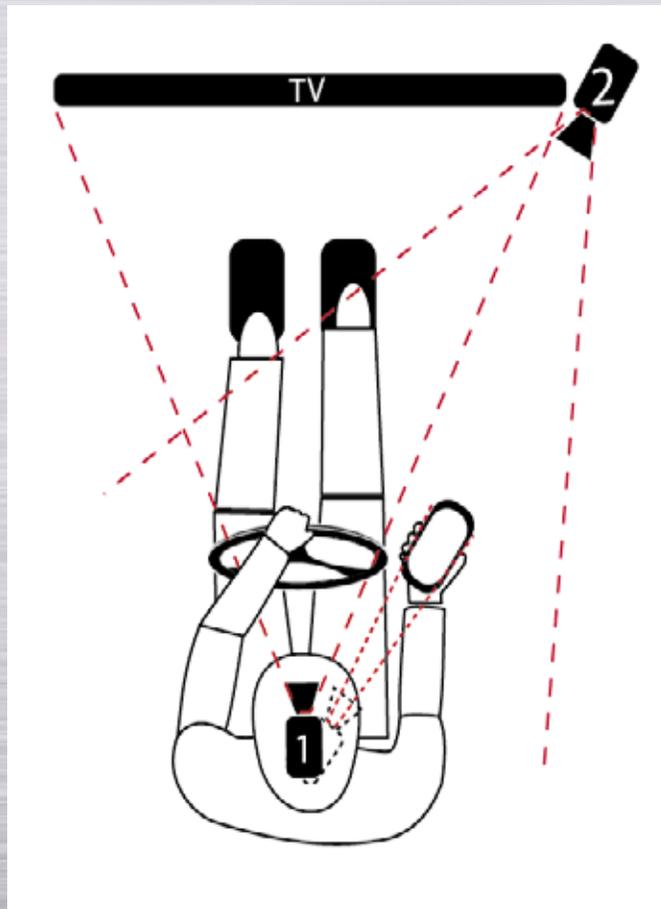
Unser Ansatz

- Wir simulieren ein pseudo-realistisches mobiles Szenario im Labor, welches
 - einfach zu beobachten ist
 - problemlos reproduzierbar ist
 - in dem wir die Umgebungsparameter beeinflussen können.

Unsere Test Umgebung

Szenario:

- Primärtask Autofahren (in einem Simulator!)
- Sekundärtask: die App benutzen



Please don't try that in your car!

- Eine Autosimulation SIMULIERT einen aufwändigen Task.
- Die meisten Leute können schon Autofahren – kaum Training der Probanden nötig.
- Im Autosimulator fahren und gleichzeitig ein Handy nutzen ist
 - legal
 - ungefährlich

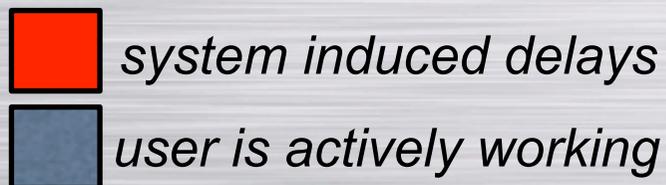
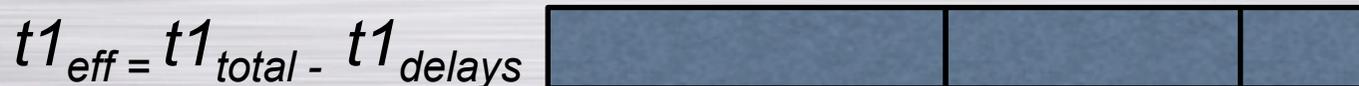


Vorteile der Autosimulation

- Geringer Platzbedarf
- Preiswert
- Einfach zu beobachten und zu filmen.
- Umgebungsparameter (Licht...) einstellbar und reproduzierbar.
- Gute Simulation “verteilter Aufmerksamkeit mit anspruchsvollem Primärtask.
- Komplexität des Primärtasks einstellbar (verschiedene Autos, Routen, “Wetterbedingungen”)

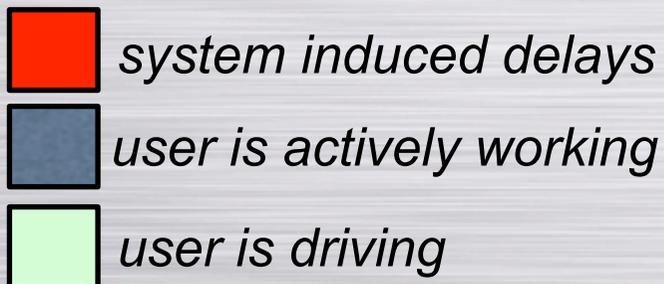
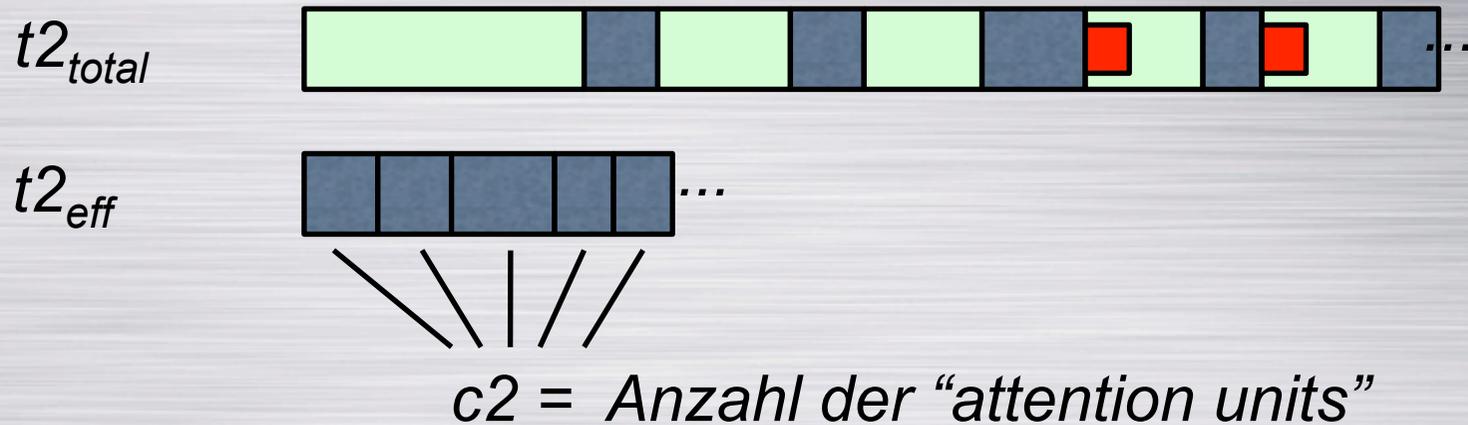
Testdurchführung 1/2

- Vorbereitung: User sollten den Simulator, den Task und die zu testende Applikation kennen.
- Reference step: Szenario ohne Störung (ohne Simulator)



Testdurchführung 2/2

- Simulator step: Szenario wird während des Fahrens im Simulator durchgeführt.



Key figures

- Durchschnittliche Dauer einer “attention unit”:

$$t2_{avg} = t2_{eff} / c2$$

- *Entschleunigungsfaktor:*

$$df_{12} = t2_{eff} / t1_{eff}$$

Erwartung: $df_{12} > 1$

Test run

App	$t1_{\text{eff}}$	$t2_{\text{eff}}$	c2	$t2_{\text{avg}}$	df_{12}
G.Maps	17	29	30	0.97	1.71
Navigon	18	26	25	1.04	1.44
TomTom	20	25.5	24	1.06	1.28
Pizzeria	24	41.3	54	0.76	1.72

Data was originally gathered in 24 test runs with 3 test participants.
Additional tests were done later to validate our findings.

Interpretation der Daten

- Weshalb reduziert sich die durchschnittliche Dauer einer Attention Unit in dem komplexen Szenario “Pizzabestellung”?
- Es gibt 2 Klassen von Attention-Units:
 - Attention-Units mit Dauer ≥ 1.5 Sekunden
 - Viele kurze Attention-Units von 0.1 bis 0.4 Sekunden
- Interpretation der Klassen:
 - Produktive Interaktionen
 - State Inspections

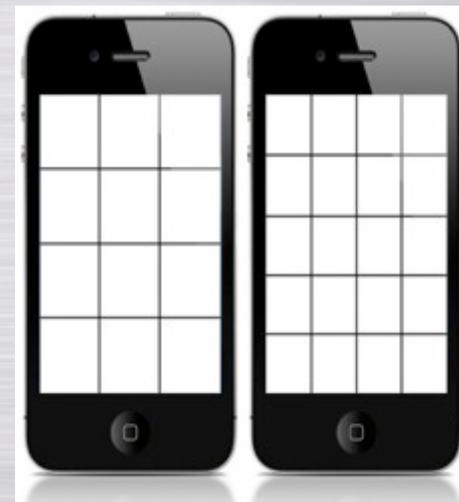
State inspections

Types:

- Check ob das System wieder verfügbar ist
- Check ob Eingabe korrekt war.

Wie kann man State Inspections vermeiden?

- Wartezeitfreies System ;-)
- Hörbares oder taktiles Feedback
- Respektiere das minimale „grid for interaction“



Limitationen der Testmethode

- Die Zahlen zeigen Dir, dass etwas falsch läuft, aber nicht genau was.
- Eine Autosimulation ist vielleicht zu komplex.
 - Apps, die im Simulator unbedienbar werden, sind nicht notwendigerweise schlecht.
 - Ein weniger komplexer Task wäre für viele echten Szenarien realistischer
- Keine echtes Eye-Tracking
- Timing der Simulator-Step ist aufwändig
 - Use our MultiTimer-App Github



Thema 2: Sensoren

Sensoren – Ermittlung des äußeren Kontexts

- Kamera / Mikrofon (Ton, Bild, Helligkeit, Gestenerkennung)
- Bewegung
 - Accelerometer – Beschleunigung (3 Achsen)
 - Gyroskopischer Sensor (3 Achsen)
- Magnetfeldsensor (2-3 Achsen)
- Annäherungssensor (Infrarot)
- Position (GPS, Glonass)
- Barometer (Luftdruck -> Höhe)
- Thermometer
- Funkbasierte Sensorik
 - WI-FI
 - Bluetooth (Bluetooth LE)
 - NFC

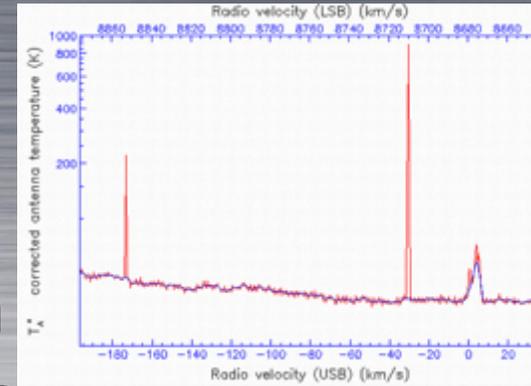
Mess- und Berechnungsverfahren

- Glättung der skalaren Messung eines Sensors
- Angulationsverfahren
- Laterationsverfahren
- Fingerprint-Verfahren
- Kontinuierliche Beobachtung

Sensorglättung

Problem: Viele Sensoren haben einen starken Jitter und einzelne Störsignale

- Lösung 1: Ausbrecher eliminieren
 - Wenn eine Abweichung von einem Messwert zum nächsten unrealistisch hoch ist, wird der Wert verworfen.
- Lösung 2: Werte glätten
 - Einfache Variante: Durchschnitt einer Range: $y_k^* = \frac{1}{2m+1} \sum_{j=-m}^m y_{k+j}$
- Lösung 3: rekursiv arbeitende Filter zur Berechnung von Sensoren mit stetiger, differenzierbarer Werteänderung
 - z.B.: Kalman-Filter



DWX

@developer_week #dwx13

Angulationsverfahren

Voraussetzungen:

- Gerichteter Sensor
- Eigene Lage in Bezug auf den Sensor bekannt

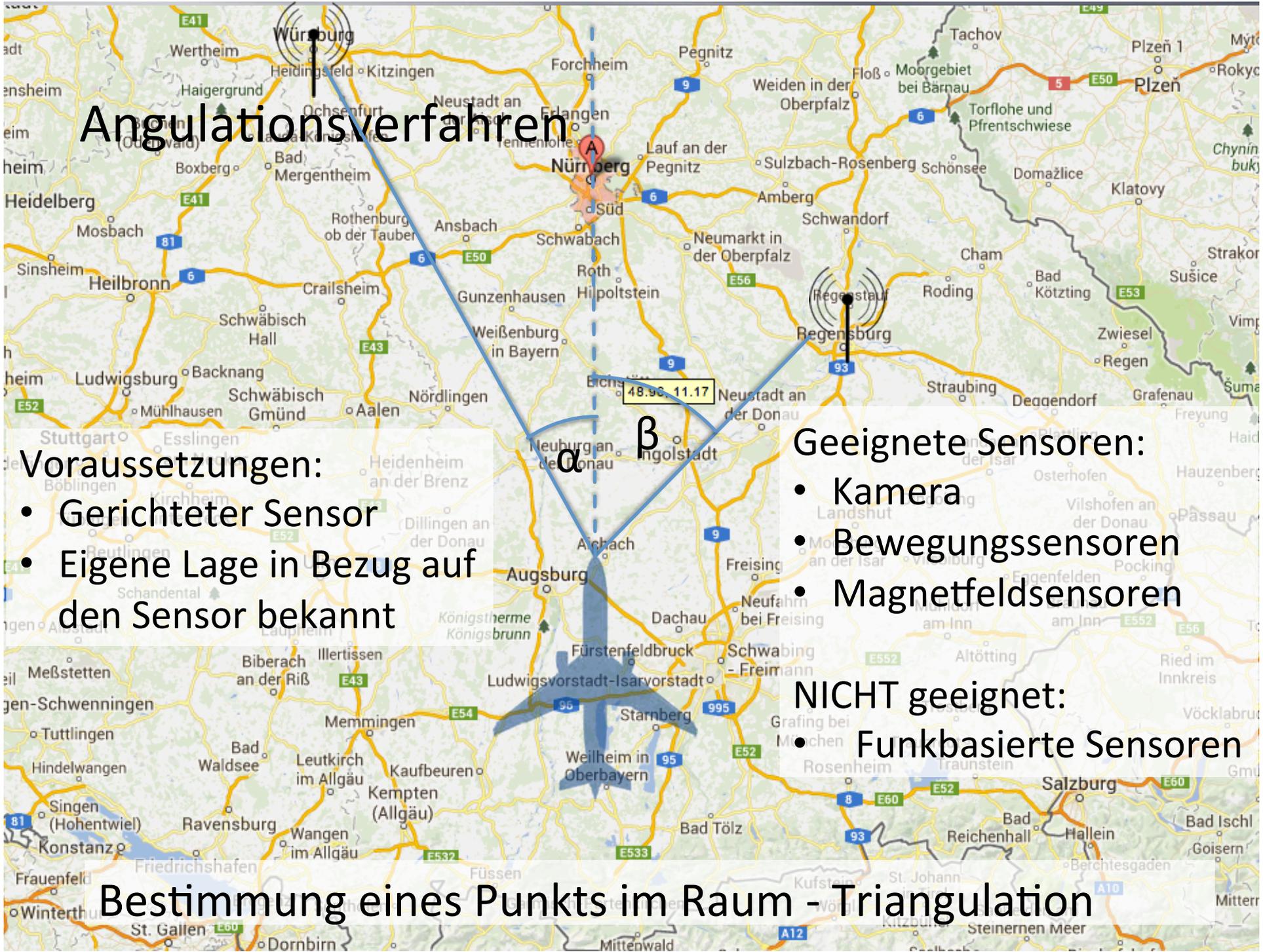
Geeignete Sensoren:

- Kamera
- Bewegungssensoren
- Magnetfeldsensoren

NICHT geeignet:

- Funkbasierte Sensoren

Bestimmung eines Punkts im Raum - Triangulation



Beispiel: Peak.ar

Methode: Angulation

Sensoren:

- Eigene Position: GPS
- Eigene Lage im Raum: Accelerometer

Externe Daten:

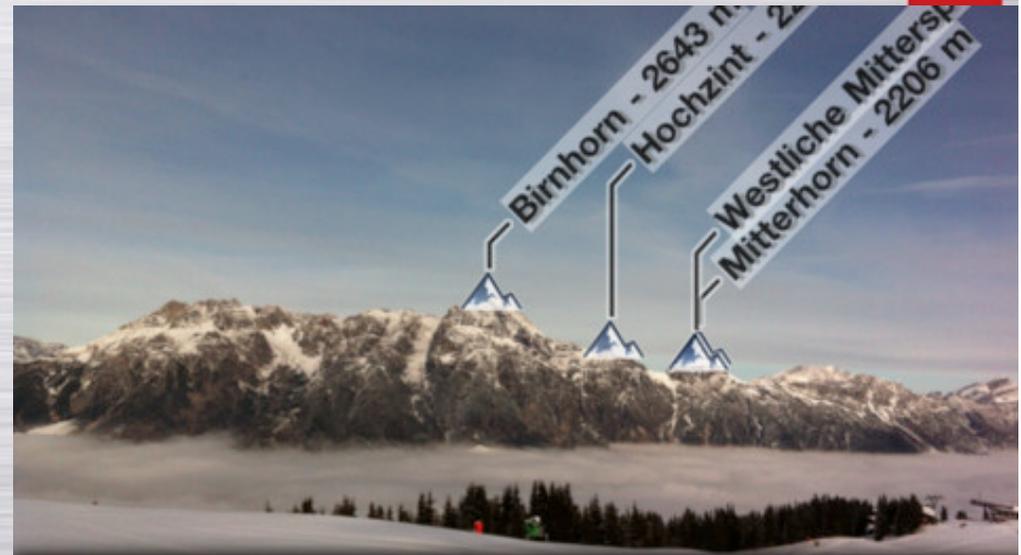
- Positionen der umliegenden Berggipfel & Höhenprofil

Voraussetzung:

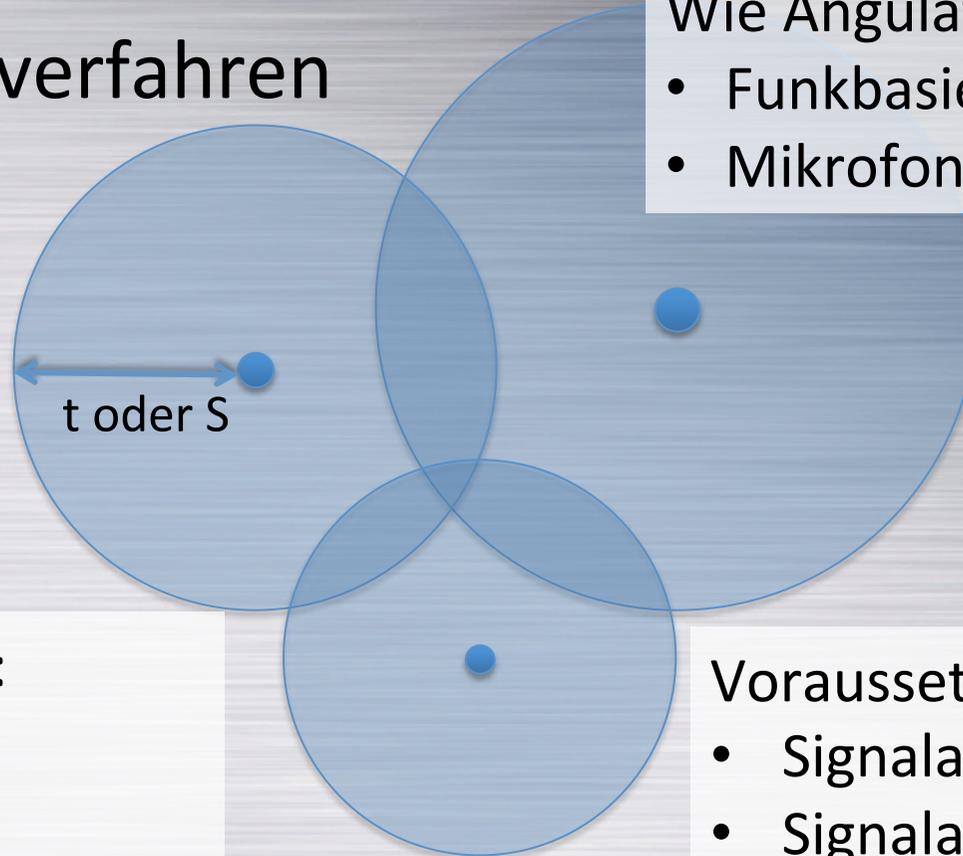
- Öffnungswinkel der Kamera bekannt / konstant

DWX

@developer_week #dwx13



Laterationverfahren



Geeignete Sensoren:

Wie Angulation +

- Funkbasierte Sensoren
- Mikrofon

Benötigte Sender:

- 2D-Ortung: 3
- 3D-Ortung: 4
- Ohne Sender/Receiver
Zeitsynchronisation: 5

Voraussetzung:

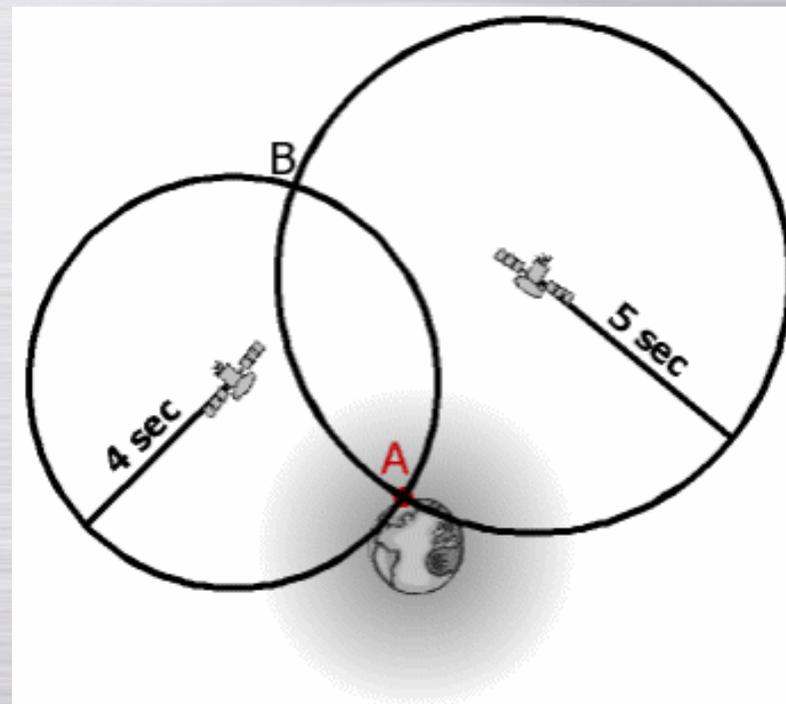
- Signalausbreitung oder
- Signalabschwächung
bekannt (ideal: linear)
und **umgebungsunabhängig**

t... Laufzeit des Signals = Sendezeit – Empfangszeit

S... Gemessene Signalstärke

Beispiel: GPS

- Methode: Lateration ohne Sender/Receiver Zeitsynchronisation
- Reduktion der benötigten Satelliten durch Ausschluss von Schnittpunkten im Weltraum.
- Verbesserung durch Differential GPS (DGPS)
- Sensor: Built-In GPS



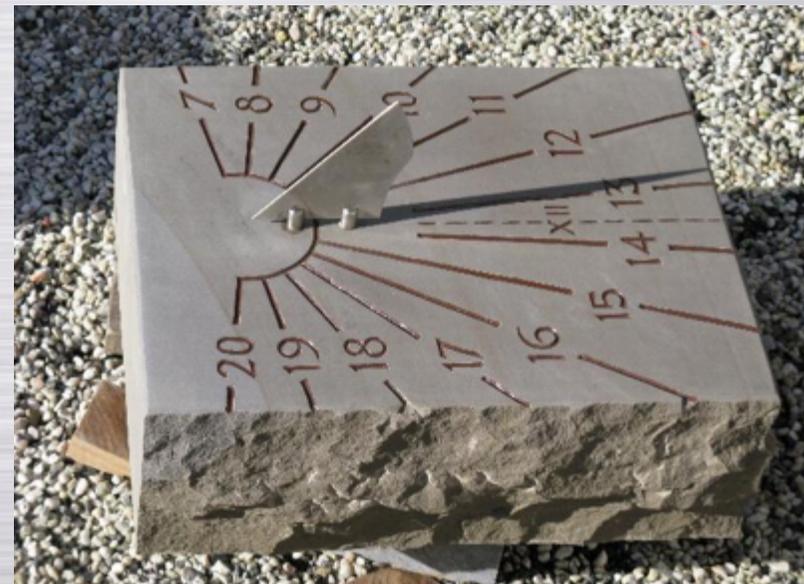
Fingerprintverfahren

- Vergleichen einen (oder mehrere) aktuelle Messwerte (ev. mehrerer Sensoren) mit einer Anzahl früher gemessener Vergleichswerte.
- Wird die aktuelle Messung in den Vergleichswerten gefunden, so kann auf (z.B. die Zeit) geschlossen werden.

Geeignete Sensoren:

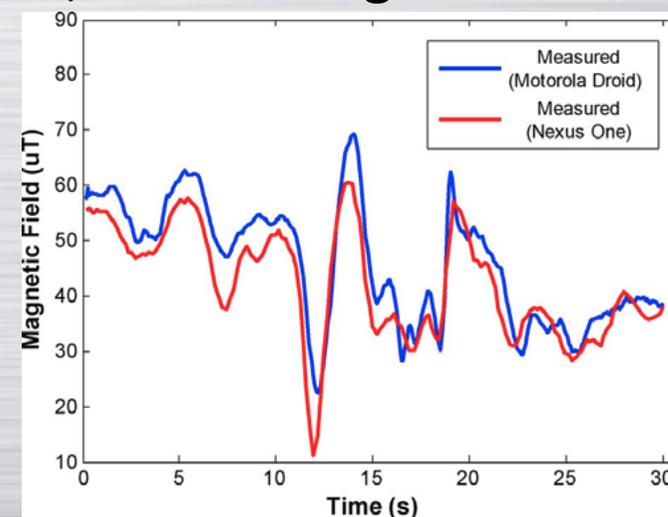
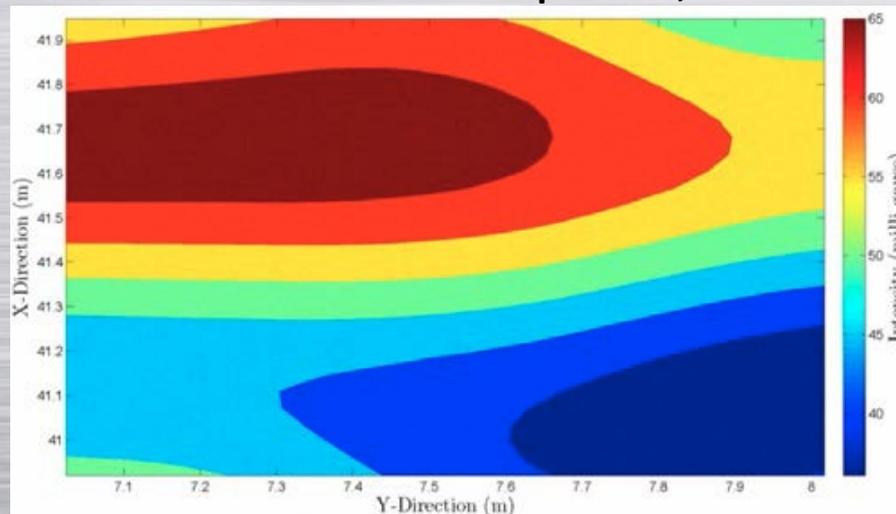
Alle, deren Messwerte charakteristisch für

- einen bestimmten Ort
 - eine bestimmte Zeit, oder
 - einen Zustand
- sind, der ermittelt werden soll.



Beispiel: Indoor-Location per Magnetfeld

- Methode: Fingerprinting
- Sensor: Magnetfeldsensor (ev. + Bewegungssensor)
- Metallträger, Leistungen, usw. verursachen charakteristische Störungen im magnetischen Feld
 - Vorteil: Diese Bausubstanz-basierten Störungen sind sehr konstant.
 - Problem: Computer, Metalltüren, Einrichtung verursacht

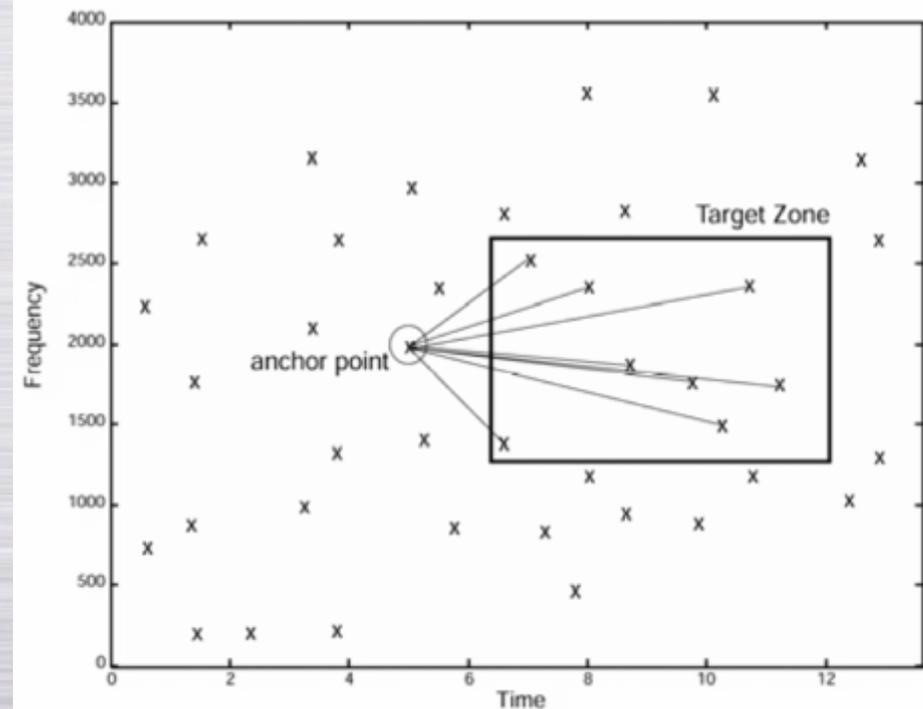


Beispiel: Shazam - Musikerkennung

Methode: Fingerprinting

Sensor: Mikrofon

- Aufnahme nur eines kurzen Teils
- Suche nach Peaks im Spektrogramm
- Suche nach Musikstücken in der DB, die das gleiche Peakmuster haben.
- Ausgehend von einem besonders ausgeprägten „Anchor Punkts“ werden die anderen Punkte in relativer, zeitlicher Abhängigkeit gespeichert.



Kontinuierliche Beobachtung

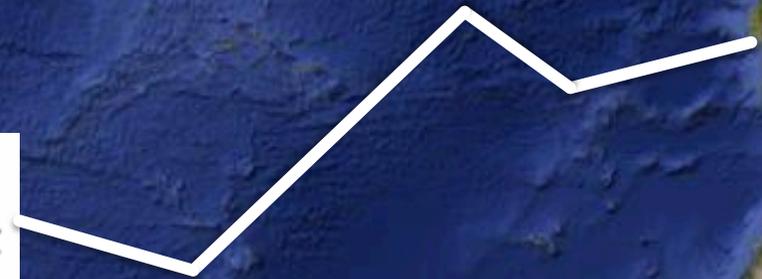
z.B. Koppelnavigation (Dead Reckoning)

Bekannter Startort: Lissabon

Kontinuierliche Messung von:

- Richtung (Steuerkurs, Abdrift)
- Geschwindigkeit

45.8, -32.2



Beispiel: Inhouse Routing ohne Funkverfahren

- Methode: Kontinuierliche Beobachtung: Koppelnavigation
- Ausgangspunkt:
 - QR-Code an bestimmter Position
- Pfaderkennung:
 - Schritterkennung / Bewegungserkennung durch Accelerometer
 - Richtung durch Kompass (Magnetometer)
 - Matching realistischer Pfade durch Pfad/Grundriss-Matching
- Verbesserung möglich durch
 - Kurze Teilstrecken (Step by step - Navigation)
 - Genaue Anweisung an User / optische Darstellung der Steps

Empfehlungen

- Vorhandene kontextbasierte Information nicht ignorieren.
- Realistische User-Stories erstellen
 - Welcher Kontext für welche Story?
 - User-Story im realen Kontext testen
- Sensordaten intelligent interpretieren
- Public Data nutzen
- Social Data nutzen
- Offen legen, was man tut + Privacy ernst nehmen

Danke



DWX

@developer_week #dwx13

FH-Prof. DI Dr. Grischa Schmiedl
grischa.schmiedl@fhstp.ac.at

DI Kerstin Blumenstein
kerstin.blumenstein@fhstp.ac.at

Forschungsgruppe Mobile <http://mfg.fhstp.ac.at>
Institut für Creative\Media/Technology
Fachhochschule St. Pölten