

Large-Scale Indoor Tracking

Im Projekt „Large-Scale Indoor Tracking in transitorischen Räumen“ schliessen sich drei Forschungsinstitute von drei verschiedenen Teilschulen der FHNW mit externen Künstlern zusammen, um gemeinsam neue Trackingverfahren für ein interaktives Kunstwerk zu entwickeln. Der daraus entstandene interaktive Soundwalk „LautLots“ wurde im September 2013 im Badischen Bahnhof in Basel uraufgeführt. Die technischen Zielsetzungen, Hintergründe und Ergebnisse dieses Soundwalks werden in diesem Beitrag beleuchtet.

Matthias Krebs, Thomas Resch, Christoph Stamm | christoph.stamm@fhnw.ch

Techniken des Verfolgens von sich bewegenden Objekten, so genannte Trackingverfahren spielen in verschiedensten Zusammenhängen in der FHNW-Forschung eine wichtige Rolle: Nutzt die Hochschule für Musik Infrarot-Tracking für die Performance elektronischer Musik, dienen im Institut für Design- und Kunstforschung (HGK) räumliches Tracking als Untersuchungswerkzeug in Stadt- und Siedlungsentwicklungsprozessen, als digitales Entwurfswerkzeug oder Ausgangsmaterial für künstlerische Projekte; das Institut für Mobile und Verteilte Systeme (HT) beschäftigt sich mit der Lokalisierung von mobilen Objekten/Personen und vor allem mit den darauf aufbauenden, kontextsensitiven Diensten in mobilen Geräten.

Im Projekt „Large-Scale Indoor Tracking in transitorischen Räumen“ schliessen sich die Forschungsinstitute der zuvor genannten Hochschulen mit externen Künstlern zusammen, um gemeinsam neue Trackingverfahren für ein interaktives Kunstwerk zu entwickeln. Inbegriff eines transitorischen Raums ist der Badische Bahnhof in Basel: als Raum auf einer Grenze (zwischen der Schweiz und Deutschland) und als Gebäude,

das man aufsucht, um es schnell zu verlassen. An einem solchen Ort können Transformationen von Kulturen und Lebenswelten wie durch ein Brennglas beobachtet werden. Im Projekt gilt es erstens diesen geschichtsträchtigen Grenzraum in Kunst und Technik neu zu erforschen und eine Vision seiner Zukunft zu gestalten und zweitens anhand des gewählten Fallbeispiels neue Mittel in Technik und Kunst zu entwickeln. Das Hauptergebnis dieses Projekts ist der Soundwalk „LautLots“: Ein akustischer Guide durch den Badischen Bahnhof, innerhalb dessen das Publikum in unmittelbare Interaktion mit Geschichte und Gegenwart des Badischen Bahnhofs tritt und diesen Grenzort neu erkunden kann. Der Soundwalk findet in der Zeit zwischen dem 8. und 13. September 2013 statt.

Soundwalk

In vorliegendem Bericht beschränken wir uns hauptsächlich auf die technischen Aspekte des Gesamtprojekts und geben Einblicke in die verwendeten Technologien und angewandten Techniken. Dennoch ist es wohl hilfreich, kurz darzustellen, was das künstlerische Ergebnis dieses

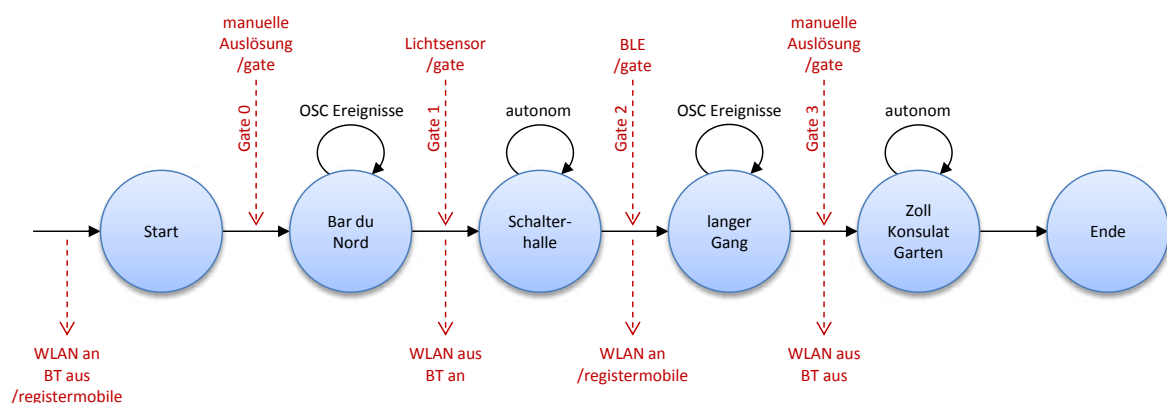


Abbildung 1: Zustandsmaschine zur Steuerung des Soundwalks „LautLots“ durch verschiedene Räumlichkeiten des Bad. Bahnhofs.



Abbildung 2: Head Mounted Device: Auf dem Bügel zwischen den Ohrmuscheln sind ein Smartphone und sechs Infrarot-LEDs angebracht. Der Kopfhörer ist ans Smartphone angeschlossen.

Projekts ist. Für weitere Einblicke sei auch auf den begleitenden Blog verwiesen [Blog].

Künstlerisches Ergebnis des Projekts ist der Soundwalk „LautLots“, der durch sehr unterschiedliche Räumlichkeiten des Bahnhofs führt (Abb. 1). Ausgerüstet mit *Head Mounted Devices* begeben sich die Teilnehmenden auf eine For-

schungsexpedition durch den Badischen Bahnhof, auf der sie Raum-, Orientierungs- und Navigationsexperimente durchführen. Das nur für die Teilnehmenden wahrnehmbare akustische Erlebnis hängt ab von deren Position im Raum sowie von deren Bewegungen und Verhalten zueinander. Der Soundwalk bringt Erkenntnisse über die vielgestaltige Topographie von Innenräumen, die Fernreisenden verschlossen bleiben. Er ist Hörspiel, interaktiver Audioguide und literarische Zimmerreise zugleich.

Head Mounted Device

Wer am Soundwalk teilnehmen will, muss den zur Verfügung gestellten Kopfhörer aufsetzen und schon kann es losgehen (Abb. 2). Nun, Kopfhörer ist ein wenig untertrieben, denn auf dem Bügel zwischen den Ohrmuscheln ist ganz schön viel Technik angebracht. Als erstes sticht das Smartphone ins Auge, welches unterschiedlichste Sensoren, die Tonwiedergabe und natürlich die Kommunikationsvorrichtung in sich vereint. Da der Kopfhörer ans Smartphone angeschlossen ist, werden offensichtlich die Stimmen und Klangwelten, welche in den Ohrmuscheln erschallen, auf dem Smartphone abgespielt oder sogar darauf produziert. Im Gegensatz zu einer Abspielliste von Musiktiteln, die in festgelegter oder zufälliger Reihenfolge gespeicherte Musikstücke abspielt, ist hier jedoch ein unsichtbarer „Regisseur“ am Werk, der die Abspielfolge auf virtuose Art lenkt. Mehr als das, der „Regisseur“ tritt in gleichzei-

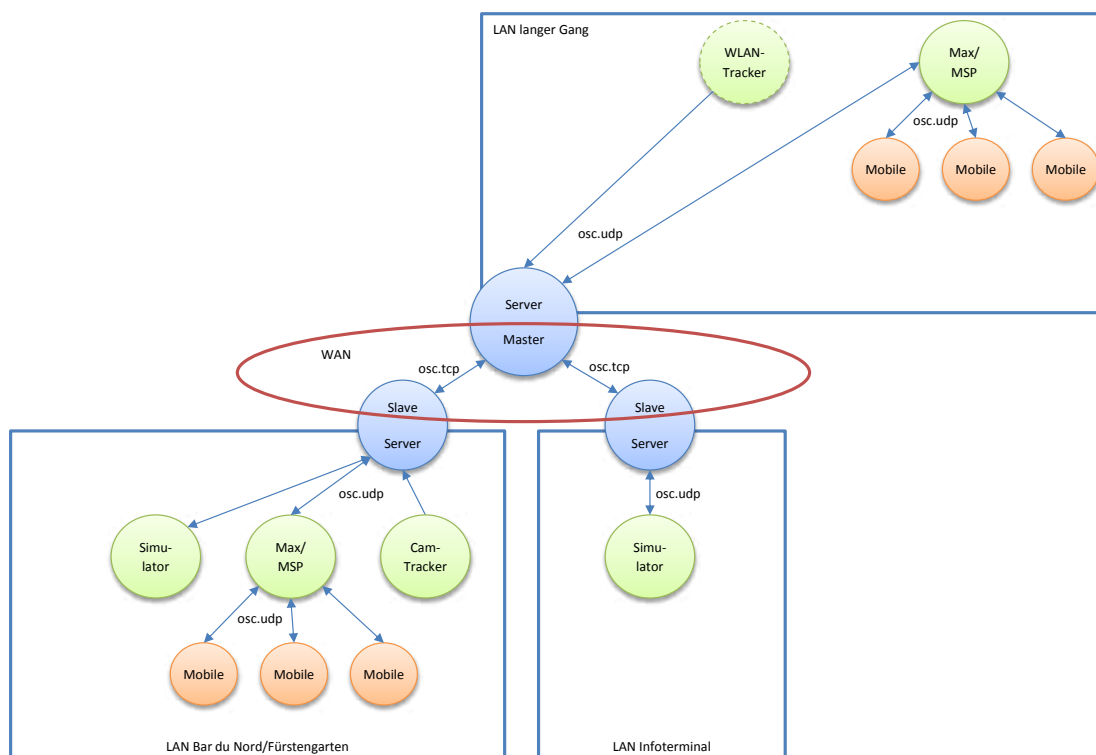


Abbildung 3: Kommunikationsstruktur. Im Zentrum befinden sich drei über das Internet miteinander verbundene Server, welche jeweils Teil eines separaten WLANs sind. Für die Kommunikation zwischen mobilen und stationären Clients und den Servern wird das leichtgewichtige und weitverbreitete OSC Protokoll verwendet.

tige und individuelle Dialoge mit den Trägern der *Head Mounted Devices* (HMD). Über eine drahtlose Netzwerkverbindung sendet der „Regisseur“ individuelle Kommandos an die Smartphones und löst Ton- und Klangwiedergabe aus. Mit Gesten wie Kopfnicken oder Kopfschütteln oder mit Bewegungen im Raum können dann die Träger der HMDs auf die Regieanweisungen reagieren und dem „Regisseur“ ein Feedback geben.

Während die Gesten auf dem Smartphone direkt mithilfe der eingebauten Sensoren und spezieller Software ermittelt und über ein Drahtlosnetzwerk an den „Regisseur“ gesendet werden, braucht es für die Bestimmung der Raumpositionen eine Lokalisierungstechnik analog zu GPS, die im Gegensatz dazu aber innerhalb von Gebäuden funktioniert.

Kommunikationskonzept

In drei Räumlichkeiten des Soundwalks sind unabhängige lokale Drahtlosnetzwerke (WLANs) in Betrieb, welche über Internetverbindungen miteinander verbunden sind (Abb. 3). In den WLANs befinden sich stationäre Server, die für die Kommunikation innerhalb des WLANs und für den Datenaustausch zwischen den verschiedenen WLANs verantwortlich sind. Infolge der geringen Anzahl Server kommt eine einfache Master/Slave-Architektur zum Einsatz, bei der sich die Slaves beim Master registrieren und aller Datenverkehr zwischen den Servern über den Master abgewickelt wird.

Der ganze Meldungsaustausch zwischen den Systemen basiert auf dem Open Sound Control (OSC) Protokoll, welches leichtgewichtig, auf verschiedensten Plattformen verfügbar und einfach handzuhaben ist. Ursprünglich ist OSC vor allem für den Austausch von Sound-Daten verwendet worden. Dabei spielen kleine Latenzzeiten eine wichtige Rolle und der Verlust reduziert eines einzelnen Datenpaketes reduziert zwar die Soundqualität, zerstört die Datenübermittlung aber nicht als Ganzes. Daher kommt bei OSC vorwiegend UDP als Trägerprotokoll zum Einsatz. Für spezielle Einsätze kann jedoch auch TCP verwendet werden, wobei nicht alle OSC-Implementierungen TCP gleich gut unterstützen. Die von uns eingesetzte OSC C-Bibliothek hatte beispielsweise noch Fehler im TCP-Teil und musste zuerst korrigiert werden.

Innerhalb der WLANs verwenden wir UDP als Trägerprotokoll für OSC. Da in einem WLAN häufig mal ein Datenpaket verloren geht, kann es schnell passieren, dass eine OSC-Meldung nicht ankommt. Für wichtige Meldungen wie zum Beispiel die Registrierung bei einem Server müssen daher spezielle Vorkehrungen getroffen werden, damit solche Meldungen mit hoher Wahrscheinlichkeit ankommen. Zwischen den Servern kommt TCP als Trägerprotokoll zum Einsatz. So wird sichergestellt, dass die Meldungen zwischen den einzelnen WLANs wirklich ausgetauscht werden und allfällige bestehende Internetverbindungen bzw. Router können unverändert weiterverwendet, da die Slaves zum Master eine TCP-Verbindung aufbauen und nur der Master-Server im Internet sichtbar und ansprechbar sein muss.

In den WLANs befinden sich verschiedene mobile und stationäre Clients, welche über OSC mit dem lokalen Server unidirektional oder bidirektional kommunizieren können. Die mobilen Clients dürfen sich zwischen den WLANs bewegen, müssen sich aber beim Eintritt in ein WLAN beim entsprechenden Server bzw. bei einem stationären Client anmelden. Dadurch können die anderen Server informiert werden, dass sie keine weiteren Meldungen an die entsprechenden mobilen Clients mehr senden sollen. Die stationären Clients übernehmen pro WLAN individuelle Aufgaben, wie beispielsweise die Lokalisierung der mobilen Clients (xTracker) oder die Regie des Soundwalks (Max/MSP). Für die stationären Clients dürfen eigenständige Computer eingesetzt werden oder sie dürfen auch als separate Prozesse auf dem gleichen Rechner wie der Server ablaufen. Auf die einzelnen Aufgaben der stationären Clients wird in weiteren Abschnitten des Artikels noch genauer eingegangen.

Kameratracking

Der Soundwalk beginnt in der Bar du Nord, dem ursprünglichen Buffet dritter Klasse des Badischen Bahnhofs. Der aufmerksamen Beobachterin fällt auf, dass an der Decke mehrere Kameras installiert worden sind, welche üblicherweise nicht vorhanden sind. Doch diese Kameras sind unauffällig und zudem beinahe blind, denn sie „sehen“ die normalen Barbesucher nicht. Es findet also keine klassische Videoüberwachung statt, bei der Videoaufnahmen auf Vorrat gespeichert oder direkt auf spezielle Vorkommnisse hin untersucht

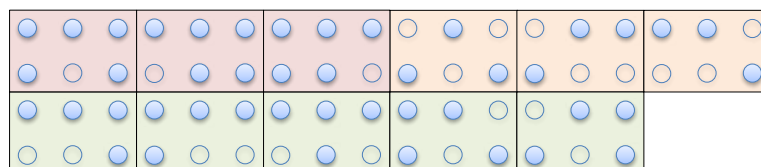


Abbildung 4: Rotationsasymmetrische Identifikationsmuster mit je 3 bis 5 Punkten angeordnet in einer regulären 2x3 Matrix. Die drei Farben grenzen die Muster mit gleicher Anzahl Punkte ab. In „LautLots“ sind die unteren fünf Muster verwendet worden.

werden. Die Kameras sind nur empfindlich für infrarotes und für uns Menschen unsichtbares Licht. Die Infrarotstrahlung wird von sechs weitwinkligen Infrarot-LEDs erzeugt, welche um das Smartphone herum auf dem HMD platziert worden sind. Die sechs rasterförmig angeordneten LEDs genügen, um mindestens elf verschiedene LED-Muster zu erzeugen, welche paarweise rotationsasymmetrisch sind und zudem eine eindeutige Ausrichtung haben (Abb. 4). Dadurch können mehrere Muster gleichzeitig verfolgt, analysiert und eindeutig einer Person zugeordnet und anhand der Musterausrichtung auch die Blickrichtung der Person ermittelt werden.

Bei den eingesetzten Kameras handelt es sich um handelsübliche, weitwinklige Überwachungskameras mit einer Auflösung von 1240x1024 Bildpunkten, welche auch in der Nacht im nahen Infrarotbereich betrieben werden können. Durch zusätzlich eingebaute Infrarotfilter werden störende Bildinhalte aus dem sichtbaren Spektrum herausgefiltert, so dass die von den HMDs ausgestrahlten Infrarotsignale ohne grosse Bildbearbeitung (Binarisierung genügt) analysiert und weiterverarbeitet werden können. Die Grösse des Raumes erfordert den Einsatz von mehreren Kameras, welche nebeneinander und mit vertikaler Ausrichtung unter der Decke angeordnet sind, so dass es zu schmalen Überlappungen an den Rändern kommt. Die Kameras verfügen über einen PoE-Anschluss, der sowohl für die Datenkommunikation als auch die Speisung verwendet wird. Die Videoströme werden über Ethernet zu einem GBit-Switch geführt und von dort zu einem Rechner geleitet, der problemlos fünf Videoströme mit 15 Bildern pro Sekunde parallel zu analysieren vermag. Dass die Prozessoren nur 15 Bilder pro Sekunde verarbeiten, liegt primär am Flaschenhals der Netzwerkverbindung zwischen Switch und Rechner. Die ermittelten Positionen und Ausrichtungen werden wie bei allen eingesetzten Clients mittels des OSC-Protokolls an einen Server gesendet, welcher die empfangenen Informationen an alle registrierten Datenkonsumenten weiterleitet. Dadurch lassen sich die Bewegungen der getrackten Personen in verschiedenen Räumen des Badischen Bahnhofs gleichzeitig verwenden bzw. visualisieren.

Der Einsatz von weitwinkligen, preiswerten Kameras führt oft dazu, dass die Bildeigenschaften am Bildrand nicht sonderlich gut sind, dass es zu stärkeren Verzerrungen kommt. Diese Verzerrungen sind vorgängig experimentell ermittelt worden und werden noch vor der Bildanalyse herausgerechnet. Dadurch wird zwar das Kamerasichtfeld etwas kleiner, aber die aufgenommenen Muster sind bis zum Rand hin gut sichtbar. Diese gute Sichtbarkeit hängt jedoch nicht nur von der Entzerrung des Kamerabilds, sondern in grösserer Masse auch von der Weitwinkligkeit (90°

der eingesetzten IR-LEDs ab. Nachdem die einzelnen Kamerabilder parallel analysiert worden sind, werden die Pixelkoordinaten der erkannten Muster mittels einer perspektivischen Transformation in Raumkoordinaten umgerechnet. Für diese Transformation ist vorgängig eine (interaktive) Ermittlung der Transformationsmatrix pro Kamera notwendig. Dabei verwenden wir vier im Bild sichtbare und bekannte Raumpunkte, welche im Kamerabild angewählt werden und daraus lässt sich die Transformationsmatrix ermitteln. Je genauer diese Transformationsmatrizen sind, desto geringer sind die Abweichungen zwischen den Raumkoordinaten, wenn zwei Kameras das gleiche bekannte Muster innerhalb eines Überlappungsbereichs detektieren.

Die eigentliche Musteridentifizierung ist ein mehrstufiger Prozess, welcher mit der Identifikation der LEDs im Bild beginnt. Aus den detektierten und vermeintlichen LEDs werden durch räumliche Nachbarschaft Clusters gebildet. Von Clustern mit relevanter Anzahl von Einzelpunkten werden dann charakteristische Merkmale bestimmt, um eine möglichst eindeutige Zuordnung zwischen entdeckten und erwarteten Clustern zur erreichen. Die Güte der einzelnen Zuordnungen nennen wir hier Relevanz. Diese Relevanz wird zusammen mit dem Schwerpunkt des Clusters und der räumlichen Ausrichtung des Musters an einen Controller übermittelt, welcher von allen Kameras die erkannten Muster zusammenträgt. Sollte das gleiche Muster von verschiedenen Kameras an ganz verschiedenen Raumpositionen entdeckt werden, so kann die Relevanz herangezogen werden, um die relevanteste Position zu ermitteln. Verschwindet ein Muster nur für kurze Zeit, das heisst, wenn ein Muster in wenigen Bildern hintereinander nicht mehr detektiert wird, so wird vorerst davon ausgegangen, dass sich das Muster in der zuletzt ermittelten Geschwindigkeit in Richtung seiner Ausrichtung bewegt. Sobald das Muster dann wieder sichtbar ist, wird die geschätzte Position durch die ermittelte ersetzt, andernfalls wird für das Muster keine Position mehr an den Controller übermittelt.

Als charakteristisches Merkmal verwenden wir die Sequenz der Innenwinkel der konvexen Hülle der Einzelpunkte eines Clusters. Da jedes von uns gewählte Muster einen eindeutigen grössten Winkel besitzt, lassen sich zwei Folgen in linearer Zeit miteinander vergleichen und der quadratische Fehler berechnen. Darüber hinaus dient der grösste Winkel auch zur Bestimmung der Ausrichtung, indem der Vektor vom Schwerpunkt des Clusters zum grössten Winkel verwendet wird.

Gates

Beim Verlassen der Bar du Nord in Richtung der Schalterhalle des Bahnhofs passieren wir ein mit einer speziellen LED-Leiste beleuchtetes Tor.

Situation	Lichtstärke [Lux]
Bar mit wenig Beleuchtung	50
Künstliche Beleuchtung, z.B. Schalterhalle	200 - 300
LED-Leiste in 30 cm Entfernung	3000
Tageslicht im Schatten	10'000
Tageslicht an der Sonne	85'000

Tabelle 1: Verschiedene Lichtsituationen mit entsprechenden Lichtstärken gemessen mit einem Samsung Galaxy S3

Diese Torbeleuchtung wird ähnlich einer Lichtschranke wahrgenommen und löst eine Reihe von Aktivitäten aus, unter anderem eine Abmeldung beim „Regisseur“ und die Deaktivierung der Drahtlosnetzwerkverbindung auf dem Smartphone. Das Smartphone verliert somit den Kontakt zum „Regisseur“ und befindet sich nun in einem autonomen Zustand. Nach wie vor hören wir die Stimme im Kopfhörer, welche uns durch die Schalterhalle zu einer unscheinbaren und verschlossenen Tür lotst.

Für die technische Umsetzung verwenden wir den Lichtsensor auf der Vorderseite des Smartphones, welcher normalerweise die Display-Helligkeit regelt. Unter Android wird ein Lichtsensor unter der Kennung *Sensor.TYPE_LIGHT* angesteuert. Der Sensor liefert einen ganzzahligen Wert zurück, welcher der Einheit Lux entspricht. Im Falle unseres Samsung Galaxy S3 liegt die Empfindlichkeit im Bereich zwischen 0 und 85'000 Lux. Sie ist stark richtungsabhängig; die besten Ergebnisse erzielt man, wenn sich die Lichtquelle fast senkrecht zum Smartphone-Display befindet. Die Tabelle 1 führt verschiedene Lichtsituationen und die gemessenen Lichtstärken auf.

Anhand dieser Beispiele zeigt sich, dass solche Messungen der Lichtstärke nicht mit dem subjektiven Empfinden übereinstimmen. Beispielsweise wird eine LED-Leiste bei direktem Blickkontakt bereits als gleissend hell empfunden, während ein Aufenthalt im Freien an einer Stelle ohne direkte Sonneneinstrahlung als deutlich weniger hell empfunden wird.

Um den Lichtsensor als eine Art Lichtschranke nutzen zu können, muss ein Lichtstärkenverlauf erzielt werden, welcher sich ausreichend von der Umgebung unterscheidet. Mit den von uns getesteten LED-Lichtquellen haben wir im besten Fall ca. 3500 Lux erreicht. Da in der recht düsteren Bar du Nord im Durchschnitt nur 50 bis 100 Lux gemessen werden, ist der Unterschied zu der am Türrahmen angebrachten LED-Lichtquelle ausreichend gross.

Beim Durchschreiten der Lichtschranke in normaler Gehgeschwindigkeit zeigt sich eine Lichtspitze, welche innerhalb einer Sekunde ihr Maximum erreicht und danach in derselben Zeit abfällt (Abb. 5). Nimmt man einen unteren Schwellwert von 1000 Lux und einen oberen Schwellwert von 3000 Lux, so ergibt sich bei einer Abtastrate von ca. 5 Hz ein Anstieg von

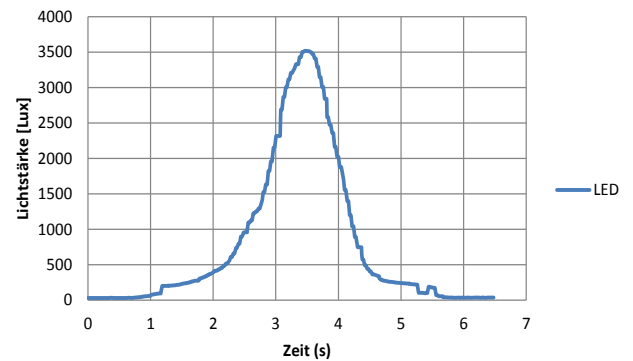


Abbildung 5: Messung der Lichtstärke beim Durchschreiten der Bar du Nord, an welcher eine LED-Lichtquelle angebracht worden ist.

ca. 400 Lux pro Messwert. Mit den Parametern für Schwellwerte und Flankensteilheit kann das Durchschreiten der LED-Leiste zuverlässig detektiert werden. Einziger Nachteil dieser Methode ist, dass im Bereich eines Fensters einfallendes Tageslicht, welches problemlos 3000 Lux oder mehr erreichen kann, eine Fehldetektion auslösen kann. Aus diesem Grund kann eine solche einfache Detektionstechnik nur in Räumen eingesetzt werden, welche gleichmässig und mit künstlichem Licht beleuchtet sind.

Beim Übergang von der Schalterhalle in den langen Gang wird ebenfalls ein Gate benötigt. An dieser Stelle muss die Drahtlosnetzwerkverbindung wieder aktiviert werden, damit sich das Smartphone erneut beim „Regisseur“ anmelden kann. Aus baulichen Gründen kommt hier keine Lichtschranke, sondern eine Art Funkbarke zum Einsatz. Ein Sender vom Typ *Stick'n'Find*, welcher den Bluetooth-Low-Energy-Standard verwendet, ist so konfiguriert, dass er im Abstand von 4 Sekunden ein Broadcast-Signal sendet, welches von einem Smartphone mit passender Bluetooth-Schnittstelle empfangen werden kann. Das Smartphone empfängt die Kennung des Senders in Form einer MAC-Adresse und die Stärke des Signals als RSSI-Wert (dBm). Befindet sich das Smartphone in einem Abstand von zwei bis drei Metern zum Sender, so wird eine Signalstärke oberhalb des konfigurierten Schwellwerts von -90 dBm erreicht. Dadurch wird die Drahtlosnetzwerkverbindung reaktiviert, was eine Registrierung im Netzwerk des langen Gangs ermöglicht.

BLE-Tracking

Der zweite von uns getrackte Raum des Soundwalks ist ein 120 Meter langer Gang parallel zu den Geleisen, welcher dem Publikum normalerweise nicht offen steht. Neben zwei WLAN-Zugriffspunkten kann der aufmerksame Beobachter mehrere blau leuchtende LEDs an den Seitenwänden des Gangs entdecken. Jede dieser blauen LEDs gehört zu einem *Bluetooth Low Energy Tag* (BLE-Tag). Diese batteriebetriebenen Tags senden in regelmässigen Abständen ein Advertising-Signal ab, welches von einem handelsüblichen Smartphone mit Bluetooth 4.0 Unterstützung

empfangen werden kann. Das Smartphone empfängt die eindeutigen MAC-Adressen der Tags, welche sich in der Nähe befinden, sofern die elektromagnetischen Signale nicht durch Wände oder andere Abschirmungen zu stark gedämpft sind.

Seit Ende 2009 ist der Industriestandard Bluetooth um die Ergänzung Low Energy (BLE) erweitert worden. BLE verwendet dieselben Frequenzen im 2.4-GHz-Band wie das herkömmliche Bluetooth-Protokoll, ist aber dazu nicht kompatibel. Entsprechende Geräte müssen diesen Standard also explizit unterstützen. Momentan sind vor allem BLE-Chips von Texas Instruments und Nordic Semiconductors erhältlich.

BLE in der Version 4.0 definiert für verschiedene Anwendungsgebiete unterschiedliche Profile. Für unseren Anwendungsfall der Lokalisierung bietet sich vor allem das Proximity-Profil an, mit welchem abgefragt werden kann, welche BLE-Tags sich in welcher Distanz zum Smartphone befinden. Dieses Profil wird beispielsweise in der zuvor erwähnten *Stick'n'Find*-Lösung verwendet.

Die Firma *Stick'n'Find* bietet kleine runde BLE-Tags an, mit deren Hilfe Gegenstände lokalisiert werden können [SnF]. Die Tags werden mit einer Knopfzelle betrieben und enthalten den BLE-Chip 51822 von Nordic Semiconductor. Dieser unterstützt das Proximity-Profil, verwendet aber eine proprietäre Firmware. *Stick'n'Find* bietet ein SDK für iOS an, nicht aber für Android. Die Spezifikationen sind ebenfalls nicht offengelegt, weshalb wir die Standardkonfiguration der BLE-Tags getestet haben. Sind die Tags aktiviert, senden sie alle 4 Sekunden ein Broadcast-Signal, sofern sie nicht mit einem Gerät verbunden sind. Dieses Intervall

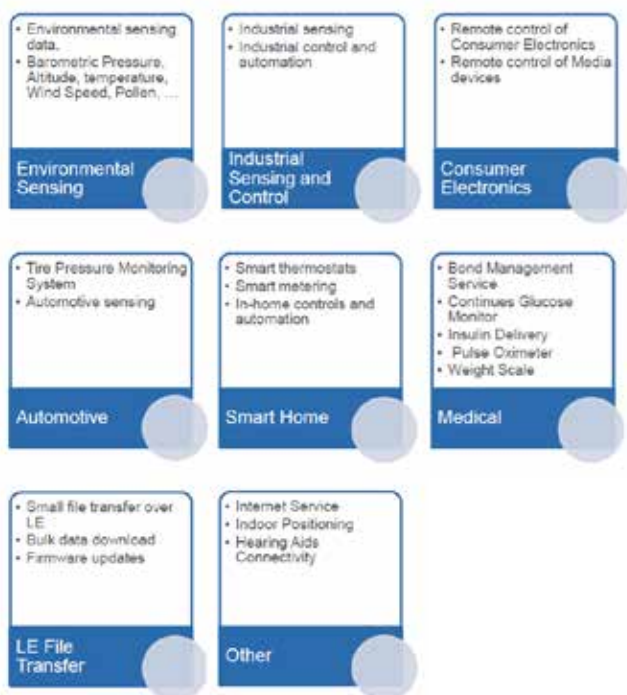


Abbildung 6: Zukünftige Einsatzgebiete von Bluetooth Low Energie gemäss der Bluetooth Special Interest Group [BSIG].

kann mit Hilfe des iOS-SDK umprogrammiert werden, wobei eine dauerhafte Umstellung des Intervalls bei unseren Tests nicht funktionierte.

Als Alternative zu *Stick'n'Find* haben wir BLE-Module der Firma Wavetek getestet, welche den gleichen BLE-Chip beinhalten [Wav]. Die Tags sind für uns so vorprogrammiert worden, dass sie bei minimaler Sendeleistung arbeiten und dafür zehn Mal pro Sekunde ein Advertising-Signal senden. Der Preis dafür ist eine relativ kurze Batterielaufzeit von weniger als einer Woche. Ein Advertising-Signal überträgt hauptsächlich die MAC-Adresse des BLE-Geräts. Das empfangende Smartphone ermittelt zusätzlich noch die Signalstärke in Form eines RSSI-Werts (dBm). Diese beiden Parameter können dann für eine Lokalisierung verwendet werden.

Die Unterstützung Seitens der Smartphones ist jedoch eher mangelhaft. Aktuelle Android-Geräte besitzen in der Regel zwar ein Bluetooth-Modul, welches BLE unterstützt, BLE selbst wird jedoch in Android erst ab Version 4.3 offiziell unterstützt. Die meisten Geräte laufen jedoch noch mit Android 4.1 oder 4.2. Manche Hersteller, beispielsweise Samsung, bieten deshalb eigene SDKs an, um BLE-Geräte ansteuern zu können. Diese SDKs sind leider in der Regel hersteller- oder gar modellspezifisch.

Lokalisierung

In der Bluetooth Special Interest Group sind bereits Profile für Bluetooth 4.1 definiert worden (Abb. 6). Eines dieser Profile soll in Zukunft auch Indoor-Lokalisierung ermöglichen. Verschiedene Hersteller, unter anderem auch Microsoft (Ex-Nokia) sind bereits an der Entwicklung entsprechender Lösungsansätze [CPK13, WYZ13, ZLS13].

Für eine Lokalisierung basierend auf BLE-Tags bieten sich vor allem zwei Techniken an: Fingerprinting und Triangulierung.

Beim Fingerprinting-Ansatz müssen die Positionen der Sender (BLE-Tags oder WLAN-Zugriffspunkte) nicht bekannt sein. Stattdessen werden in einem Vorbereitungsschritt an verschiedenen bekannten Testpunkten des Raumes Signalmessungen durchgeführt und alle vorhandenen Signale pro Position zu einem positionsspezifischen also charakteristischen „Fingerabdruck“ vereint und zusammen mit den Positionskoordinaten in einer zentralen Datenbank abgespeichert. Sobald diese Datenbank aufgebaut ist, kann sie für die eigentliche Lokalisierung herbeigezogen werden. Zur Bestimmung der Position misst das Smartphone im Raum alle Signale und erstellt daraus wieder einen „Fingerabdruck“. Diesen sendet es an einen Server, welcher Zugriff auf die zentrale Datenbank hat, und der Server durchsucht die Datenbank nach möglichst ähnlichen „Fingerabdrücken“. Aus den gespeicherten Raumpositionen der

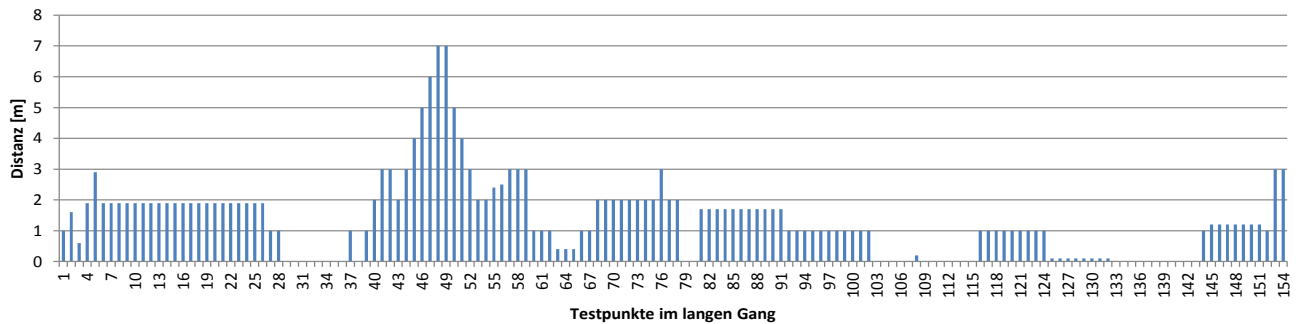


Abbildung 7: Lokalisierungsfehler beim Triangulierungsverfahren mit 18 BLE-Tags gemessen bei mehr als 150 Testpunkten in einem 120 Meter langen Gang.

ähnlichsten „Fingerabdrücke“ lässt sich dann die Raumposition des Smartphones abschätzen.

Dieser Ansatz funktioniert für statische Lokalisierung (die Bestimmung von Personen oder Objekten, welche sich kaum bewegen) recht gut. In einer ruhigen Umgebung stabilisieren sich die empfangenen Signale nach wenigen Sekunden und es kann eine Lokalisierungsgenauigkeit von 2.5 bis 5 Metern erzielt werden. Der Vorteil dieser Methode besteht darin, dass alle möglichen Signalquellen für die Bestimmung der „Fingerabdrücke“ verwendet werden können, egal ob die Senderpositionen bekannt sind oder nicht. Ein gewichtiger Nachteil ist jedoch der vorgängige Aufbau der Datenbank und die spätere Aktualisierung dieser, auch dann, wenn die Aktualisierung automatisch von den Benutzern der Datenbank erledigt werden kann.

Beim zweiten Ansatz, der Triangulierung, müssen die Positionen der Signalquellen (BLE-Tags) und mit Vorteil auch die Signalstärken der „sichtbaren“ Tags bekannt sein. Mit Sichtbarkeit meinen wir hier, dass das Smartphone ein Tag genau dann „sieht“, wenn es von ihm ein Advertising-Signal empfängt. Da das Advertising-Signal die eindeutige MAC-Adresse des Tags beinhaltet, kann mithilfe eines Mappings die Position des Tags eruiert werden. Für den Fall, dass mindestens drei Tags sichtbar sind, können die bekannten Positionen der Tags verwendet werden, um die Position des Smartphones abzuschätzen. Zur eigentlichen Triangulierung kommt es dann, wenn aus den empfangenen Signalstärken von drei verschiedenen Sendern auf die Distanzen zu den Sendern geschlossen wird und diese Distanzen zusammen mit den Positionen der Sender verwendet werden, um die Position des Empfängers genau zu berechnen.

Dieser Ansatz funktioniert auch für eine dynamische Lokalisierung recht gut, vorausgesetzt, dass die Advertising-Signale beim Empfänger in kurzen Abständen (ca. 0.1 s) eintreffen. Erste Tests zeigen, dass sich damit durchschnittliche Lokalisierungsgenauigkeiten von 1.5 bis 3 Metern erzielen lassen (Abb. 7). Der Nachteil dieses Ansatzes ist, dass die Positionen der Signalquellen

bekannt sein müssen (oder die Signalquellen müssen nicht nur ihre eindeutige MAC-Adresse, sondern auch gleich noch ihre Position übermitteln) und dass ein relatives enges Netz an Tags vorhanden sein sollte, wenn eine hohe Lokalisierungsgenauigkeit angestrebt wird. Da sich BLE-Tags aber sehr einfach anbringen lassen und der Stromverbrauch nicht sehr gross ist, kann der Aufbau einer entsprechenden Tag-Infrastruktur durchaus für grosse Räume in Erwägung gezogen werden. Zukünftige Forschungsprojekte könnten hier aufzeigen, mit welchem finanziellen Aufwand welche Genauigkeit erzielt werden kann.

Android App

Die Steuerung unseres *Head Mounted Devices* (HMD) wird von einem Smartphone übernommen. Die Anforderungen an Hard- und Software sind dabei im Vergleich zu gewöhnlichen Smartphone-Apps aussergewöhnlich hoch: Es müssen in Echtzeit verschiedenste Sensoren ausgewertet, drahtlose Verbindungen wie WLAN und Bluetooth genutzt und auch Audio-Processing durchgeführt werden. Dazu kommt, dass die App nicht durch Stromsparmechanismen des Betriebssystems beeinträchtigt werden darf.

Bei der Wahl der Smartphone Plattform haben wir uns für Android entschieden, da Android den App-Entwicklern die grösstmögliche Flexibilität im Umgang mit dem Betriebssystem bietet. Geschlossenerer Plattformen wie Apple iOS unterliegen starken Einschränkungen durch den Hersteller, welche nicht oder nur mit grossem Aufwand umgangen werden können.

Eine erste Herausforderung ist die Entwicklung einer App, deren Funktionen laufend im Hintergrund operieren und auch dann noch funktionieren, wenn beispielsweise das User Interface der App durch Druck auf den Home-Button in den Hintergrund versetzt oder das Display ausgeschaltet wird. Im Normalfall würde dadurch die App einfach durch das Android-System pausiert.

Apps, welche sich im Hintergrund befinden, können durch die Speicherverwaltung von Android jederzeit aus dem Speicher entfernt werden. Wir lösen dieses Problem durch Verwendung ei-

Kopfnicken	Schwellwert
Ableitung der linearen Beschleunigung	0.3 m/s ³
Gyroskop	1.1 rad/s
Maximale Zeit zwischen linearer Beschleunigung und Gyroskop	45 ms
Maximale Zeit zwischen Teilgesten	500 ms
Minimale Anzahl Teilgesten	2 - 3

Tabelle 2: Empirisch ermittelte Schwellwerte zur Detektion von Kopfnicken mithilfe eines Gyroskops und Beschleunigungssensors.

nes Hintergrund-Services, welcher zusammen mit der App gestartet wird und praktisch die gesamte Funktionalität der App enthält. Android erlaubt es, Services zu definieren, welche weiter im Hintergrund operieren können, auch wenn andere Apps im Vordergrund sind. Auch die Umgehung von Stromsparmechanismen ist möglich. Android bietet dazu sogenannte *Wake Locks*. Durch Akquirierung eines solchen Wake Locks kann ein Service weiter ausgeführt werden, auch wenn beispielsweise das Display abgeschaltet wird. Ohne ein Wake Lock greifen in diesem Fall die Stromsparmechanismen des Systems, und die CPU wird in den Standby-Modus versetzt.

Ein letztes Problem des Android-Systems im Zusammenhang mit der Speicherverwaltung kann nur mittels eines Workaround umgangen werden: Das explizite Beenden einer App ist nicht vorgesehen. Wenn das User Interface einer App, egal auf welche Weise, geschlossen wird, wird die

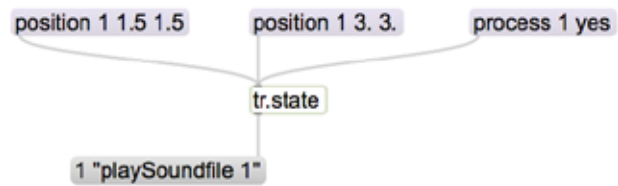


Abbildung 9: Das Backend Objekt „tr.state“ für Max/MSP mit eingehenden und ausgehenden Meldungen für Client 1.

App nicht wirklich beendet. Android entscheidet selbst, wann der Speicher freigeräumt wird. Entwickler können darauf keinen Einfluss nehmen. Weil unser Hintergrund-Service auch dann weiterlaufen muss, wenn das User Interface geschlossen wird, ist eine Beendigung unserer App nicht einfach. Ohne eine vorgängige Terminierung wäre jedoch ein Neustart unmöglich. Wir umgehen dieses Problem, indem wir im User Interface eine spezielle „Exit“-Funktion anbieten, die vor dem Schliessen den Service beendet. Android kann anschliessend den Speicher bei Bedarf freigeben. Beim nächsten Aufrufen der App wird der Service einfach neu gestartet, falls er sich noch im Speicher befindet oder neu instanziiert.

Für das Audio-Processing verwenden wir *Pure Data* [PD]. PD ist eine spezielle Programmiersprache zur Verarbeitung und Erzeugung von Audio-Daten, kann aber auch zur Auswertung von Sensoren und Steuerung von Abläufen genutzt

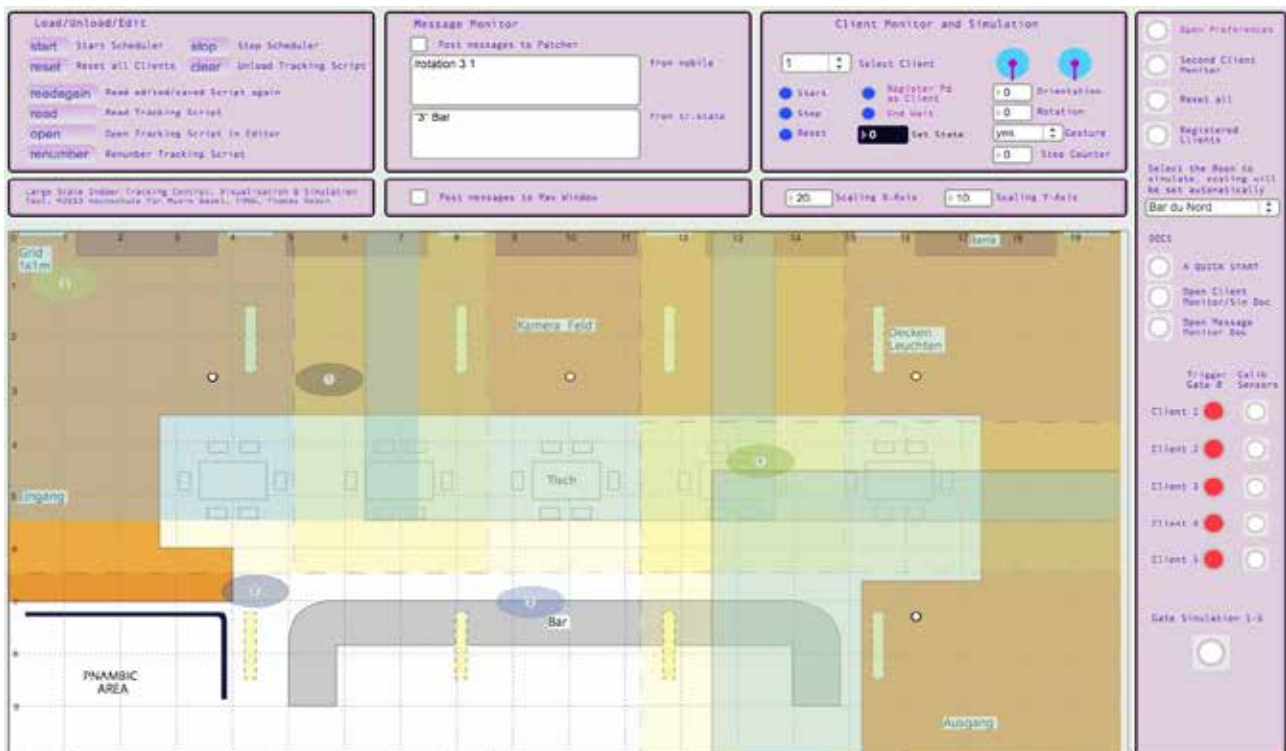


Abbildung 8: Grafische Benutzeroberfläche des „Regisseurs“ in Max/MSP programmiert. Sie ermöglicht die Visualisierung von Position, Orientierung und Gesten als auch die Simulation dieser Ereignisse.


```

1: newClient 1;
2: newClient 2;
3: nextState 1 STATE 0 position 1. 1. 1. 1. message sender "playSoundfile 1";
4: nextState 2 STATE 1 !position 1. 1. 1. 1. message sender "playSoundfile 2";
5: nextState 3 STATE 2 yes message sender "playSoundfile 3";
6: nextState 1000 maxTime 60. message sender "The End";

```

Listing 1: Einfaches Skript zur Programmierung einer Zustandsmaschine

werden. PD kann mit Hilfe der Open-Source-Bibliothek *libpd* genutzt werden. Es handelt sich um eine C-Bibliothek, welche auch unter Android lauffähig ist. In unserer App wird *libpd* für die Ausgabe von vorgefertigten Audiodateien sowie für die Steuerung von Teilen des Audiowalks eingesetzt. Dabei werden auf dem Smartphone entsprechende PD-Skripte, sogenannte Patches, ausgeführt. Diese verarbeiten Ereignisse von Sensoren des Smartphones oder Anweisungen, die vom „Regisseur“ über das Netzwerk erhalten werden.

Die Android App generiert auch selber Meldungen, die an den „Regisseur“ gesendet und von ihm ausgewertet werden müssen. Dazu gehören u.a. die Gestenerkennung für Kopfnicken und Kopfschütteln und die Schritterkennung, die immer dann einspringt, wenn sich das Smartphone im ungetrackten Bereich befindet.

Das Detektieren von Kopfnicken und Kopfschütteln erfolgt mit Hilfe des eingebauten Gyroskops und Beschleunigungssensors. Zunächst wird die Ableitung der linearen Beschleunigung gebildet. Sofern auf der für Kopfnicken/Kopfschütteln relevanten Achse ein bestimmter Schwellwert sowohl beim Gyroskop als auch bei der Ableitung der linearen Beschleunigung überschritten wird und beide Überschreitungen innerhalb eines bestimmten Zeitbereiches stattfinden, wird dies als Teilbewegung der Geste gezählt. Folgen mehrere solcher Teilbewegungen innerhalb einer bestimmten Zeit aufeinander, wird dies als vollständiges Kopfnicken/Kopfschütteln interpretiert. Die Schwellwerte in Tabelle 2 sind im Laufe des Projekts empirisch ermittelte Werte.

Die Schritterkennung basiert auf Messungen der Beschleunigung entlang der z-Achse. Wird ein bestimmter Schwellwert überschritten, wird dies als Schritt gewertet und die Detektion für die nächsten 250 ms blockiert. Dabei muss sauber darauf geachtet werden, dass andere Bewegungen wie beispielsweise Kopfnicken nicht als Schritt gezählt werden.

Max/MSP

Unsere Reise durch den Bahnhof wird am Ende des langen Gangs fortgesetzt. Dabei passieren wir noch weitere Tore und weitere Räume und kommen schliesslich wieder in der Bar du Nord an.

Auf einer grossen Leinwand sehen wir die aktuellen Positionen aller Teilnehmerinnen und deren zurückgelegte Wege durch den Bahnhof. Der „Regisseur“ – eine in Max/MSP programmierte Software – weiss also zu jedem Zeitpunkt, wo wir uns befinden. Er kann uns mit positions- und orientierungsabhängigen Informationen versorgen und unseren weiteren Weg beeinflussen. Die in Abbildung 8 dargestellte grafische Benutzungsoberfläche des „Regisseurs“ erlaubt zudem sowohl die Visualisierung von Position, Orientierung und Gesten als auch die Simulation dieser Ereignisse.

Um den Künstlern möglichst schnell eine Software zur Verfügung stellen zu können, die es ihnen erlaubt, den Ablauf des Soundwalks selbstständig zu planen, ohne auf die ständige Hilfe der Programmierer angewiesen zu sein, wird als Basis zur Entwicklung des „Regisseurs“ die visuelle Programmierumgebung Max/MSP benutzt, welche aus dem bereits erwähnten *Pure Data* entstanden ist. Max/MSP erlaubt sehr schnelles Prototyping, insbesondere die GUI-Entwicklung ist mittels Drag-and-Drop sehr effizient. Kontrollstrukturen werden allerdings ab einer gewissen Komplexität sehr schlecht lesbar und schwierig zu warten, daher ist das Backend als Objekt für Max/MSP in C programmiert. Das Backend besteht im Wesentlichen aus einer programmierbaren Zustandsmaschine, die in Echtzeit auf die eingehenden Events reagiert, den Zustand der mobilen Clients wechselt und Meldungen an die Smartphones verschickt (Abb. 9). Ein exemplarisches Skript zur Programmierung der Zustandsmaschine ist in Listing 1 abgebildet und nachfolgend beschrieben.

Im Skript werden zunächst die mobilen Clients mit den Namen „1“ und „2“ generiert. Beide starten automatisch in Zustand 0. In Zeile 3 wird ein Zustandsübergang von 0 nach 1 beschrieben, welcher durch Einnehmen der angegebenen Raumposition ausgelöst wird. Mit dem Zustandsübergang wird auch die Aktion „playSoundfile 1“ an das entsprechende Smartphone des Clients gesendet. Das Schlüsselwort sender drückt dabei aus, dass die Nachricht an den gleichen Client gesendet wird, welcher auch den Zustandsübergang vollzieht. In Zeile 4 wird ein Zustandsübergang von 1 nach 2 beschrieben, welcher durch Verlassen der ange-

geben Raumposition ausgelöst wird. Der in Zeile 5 beschriebene Zustandsübergang von 2 nach 3 wird durch den Erhalt der Nachricht *yes* ausgelöst. Schliesslich in Zeile 6 wird unabhängig vom aktuellen Zustand in Zustand 1000 gewechselt, falls sich der mobile Client länger als 60 Sekunden im gleichen Zustand befindet.

Zusätzlich zu *position*, *yes* und *no*, *maxTime* und *STATE* existieren zahlreiche weitere Bedingungen für Zustandsänderungen, u.a. *orientation* und *rotation* oder auch der Name eines Clients, was individuelle Abläufe für unterschiedliche Clients erlaubt. Bis zu 20 dieser Bedingungen können logisch miteinander AND-verknüpft werden. Ebenso können bis zu 20 unterschiedliche Meldungen pro Zustandsübergang verschickt werden.

Ergebnisse und Ausblick

Das Kunstprojekt „LautLots“ fand sowohl bei den Teilnehmerinnen und Teilnehmern als auch den Medien (Radio SRF2, Basellandschaftliche Zeitung, Programm-Zeitung) erfreulich grossen Anklang. Die Technik dahinter interessierte die hauptsächlich kunstaffinen Teilnehmerinnen und Kulturberichtstatter nur bedingt, obwohl ein Grossteil der gesamten Projektzeit in die Technik und die technische Umsetzung der künstlerischen Inszenierung floss.

Das gesamte Projekt war von Anfang an eine enge Verwebung von Kunst und Technik und es war frühzeitig klar, dass der pädagogische Anspruch der Hochschule für Technik, die verwendete Technik offen darzulegen und so das Interesse daran zu wecken, dem künstlerischen Anspruch der Hochschule für Musik widersprach, die Magie der akustisch generierten virtuellen Welt auf-

rechtzuerhalten. Hier prallten gänzlich verschiedene Kulturen aufeinander, welche sich auch bei der generellen Herangehensweise und Umsetzung immer wieder offenkundig zeigten.

Aus technischer Sicht sind etliche Probleme mehr oder weniger elegant gelöst worden, andere konnten nur teilweise gelöst oder angepackt werden. Exemplarisch zu erwähnen ist die Auseinandersetzung mit Bluetooth Low Energie, welche nicht von Anfang an geplant war, sondern erst im späteren Projektverlauf als Alternative zu WLAN-basierter Lokalisierung Aufmerksamkeit erhielt. Da erste Tests mit BLE vielversprechend verliefen, besteht das Bedürfnis in diesem Bereich weitere Untersuchungen machen zu können und in einer Machbarkeitsstudie das Potential für Indoor-Lokalisierung auf BLE-Basis aufzuzeigen.

Referenzen

- [Blog] LautLots. Ein akustischer Guide durch den Badischen Bahnhof. <http://www.lautlots.ch>
- [BSIG] Bluetooth Special Interest Group. <http://www.bluetooth.org>
- [CPK13] Liang Chen, Ling Pei, Heidi Kuusniemi, Yuwei Chen, Tuomo Kröger, Ruizhi Chen: Bayesian Fusion for Indoor Positioning Using Bluetooth Fingerprints. *Wireless Personal Communications (WPC)* 70(4):1735-1745 (2013)
- [PD] Pure Data. <http://puredata.info/>
- [SnF] Stick-N-Find. <https://www.sticknfind.com/>
- [Wav] WaveTek. <http://www.wavetek.com.hk/>
- [WYZ13] Yapeng Wang, Xu Yang, Yutian Zhao, Yue Liu, Laurie G. Cuthbert: Bluetooth positioning using RSSI and triangulation methods. *CCNC 2013*:837-842
- [ZLS13] Li Zhang, Xiao Liu, Jie Song, Cathal Gurrin, Zhiliang Zhu: A Comprehensive Study of Bluetooth Fingerprinting-Based Algorithms for Localization. *AINA Workshops 2013*:300-305