

Ein wissensbasiertes Konzept für die 3D-Klanggenerierung in virtuellen Welten

Julia Fröhlich, B.Sc., Prof. Dr. Ipke Wachsmuth
Arbeitsgruppe Wissensbasierte Systeme – Technische Fakultät
Universität Bielefeld
Universitätsstraße 25, 33615 Bielefeld
Tel. 0521/106-2921, Fax. 0521/106-2962
jfroehli@techfak.uni-bielefeld.de

Zusammenfassung

Virtuelle Realität gewinnt immer größere Bedeutung in verschiedenen Bereichen, insbesondere in der Produktentwicklung, aber auch in der industriellen und wissenschaftlichen Forschung. Ein Großteil dieser Arbeiten konzentriert sich jedoch hauptsächlich auf die grafische Repräsentation der virtuellen Welt. Weitere Ausgabemodalitäten wie Sound, Geruch und taktiles Feedback werden eher wenig berücksichtigt. Um aber dem Anwender einen möglichst hohen Grad an Realismus zu vermitteln, sind auch diese Modalitäten von Bedeutung.

Dieser Beitrag stellt ein wissensbasiertes Konzept zur intuitiven Entwicklung von akustisch unterlegten virtuellen Welten vor. Unser Ansatz ist die semantische Anreicherung der virtuellen Welt, um Klänge zu integrieren. Dabei entstehen sogenannte intelligente Objekte, welche mit Wissen über ihre weiteren Eigenschaften versehen werden, zum Beispiel, in welcher Art und Weise ein Klang an einer bestimmten Position abgespielt wird. Auf diese Weise können den Objekten Informationen zu ihren akustischen Eigenschaften in intuitiver Weise zugewiesen werden.

Schlüsselwörter

Virtuelle Realität, Methoden und Werkzeuge zur effizienten Entwicklung von Anwendungen, 3D-Klanggenerierung

Ersch. in: Gausemeier, J., Grafe, M., & Meyer auf der Heide, F. (eds.) 10. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, pp. 133–143. Paderborn: HNI Verlagsschriftenreihe Band 295 (2011)

1 Einleitung

Im Bereich der virtuellen Produktentwicklung wird in der Regel viel Wert auf eine überzeugende grafische Repräsentation gelegt. Die akustischen Eigenschaften der technischen Produkte werden häufig nicht berücksichtigt. Dies liegt sicher auch an dem erhöhten Entwicklungsaufwand, der mit einer zusätzlichen Soundausgabe verbunden ist.

Während sich innerhalb eines effizienten Produktentwicklungsprozesses virtuelle Repräsentationen bereits stark etabliert haben, beschränken sich diese jedoch meistens auf eine grafische Darstellung der beteiligten Objekte. Grafische virtuelle Objekte erlauben beispielsweise eine kostengünstige Erstellung von Prototypen, welche in frühen Phasen der Entwicklung Fehler in der Konzeption und Konstruktion verringern bzw. vermeiden können [SGG07]. Bei vielen technischen Produkten, die heutzutage schon virtuell getestet werden, ist das akustische Verhalten von essentieller Bedeutung und sollte somit ebenfalls berücksichtigt werden. Geht es zum Beispiel um die Entwicklung lärmentwickelnder Maschinen, könnte mittels einer Simulation der Klangeigenschaften die Belastung für die bedienenden Personen erprobt werden. Wichtige Warnsignale beim Training mit sicherheitskritischen Apparaturen sind ebenso von großer Bedeutung.

Ein Teil der Forschung befasst sich damit wie die akustische Ausgabe möglichst korrekt klingen kann. Eine realistische Klangsynthese kann beispielsweise mittels einer Wellenfeldsynthese realisiert werden, ist allerdings sehr rechenaufwendig [HNG09]. Bei der Erstellung eines Klangsystems sollten aber zumindest richtungs- und zustandabhängige Eigenschaften berücksichtigt werden, so wie daraus resultierend ein geeignetes Ausgabemedium eingesetzt werden [Vor08]. Ein anderer Teil der Forschung konzentriert sich darauf, wie man den enormen Entwicklungsaufwand solch akustisch angereicherter virtueller Welten senken und trotzdem ein gutes Klangergebnis schaffen kann.

Ein Ansatz, weiteres Wissen in virtuelle Welten zu speichern, sind Intelligente Virtuelle Umgebungen. Dabei werden Objekte mit semantischen Informationen angereichert. Im Allgemeinen stellen semantisch angereicherte Objekte eine gute und effiziente Möglichkeit zur Integration weiterer Eigenschaften in virtuellen Welten dar [LA00]. Bisher wurde dieser Ansatz hauptsächlich genutzt, um den Objekten Wissen über ihre grafischen Eigenschaften mitzugeben. Beispielsweise wurden in dem Projekt Virtuelle Werkstatt Bauteile mit Wissen angereichert, um intelligente Verbindungsstellen oder parametrische Veränderungen zu ermöglichen [LBW05]. Die Operationen mit diesen Bauteilen wurden bereits mit charakteristischen Geräuschen unterlegt, die aber zunächst von Hand in den Code eingepflegt wurden.

Der folgende Beitrag stellt ein Framework vor, welches es ermöglichen soll, mit sehr geringem Entwicklungsaufwand auch nicht-visuelle Informationen in virtuellen Umgebungen zu verankern. Die Realisierung des Frameworks konzentriert sich im ersten Schritt auf die Generierung von virtuellen Soundquellen. Die Grundlagen dafür sind wissensbasiert direkt in Objekten der Welt verankert. Die so generierten intelligenten Objekte „wissen“ eigenständig wie sie klingen, und bringen ihre akustische Ausgabe in

die Umgebung ein. Ein Fernziel ist es, dass die Klangobjekte darüber hinaus mit ihrer Umgebung interagieren. Ähnlich wie ein Spotlight zur ambienten Beleuchtung beiträgt, könnten lokale Schallereignisse (Geräusche, Klänge, etc.) den Raumklang beeinflussen.

2 Virtuelle Soundquellen

Für die Simulation von räumlichen Klängen muss auf zwei wesentliche Eigenschaften eingegangen werden. Zum einen muss die Richtung des Klanges bestimmt werden und zum anderen ist die Entfernung zum Benutzer wichtig, um ein realistisches Klangerlebnis zu schaffen. Auf diese Weise können die geeigneten Lautsprecher ausgewählt und ein angemessener Lautstärkepegel bestimmt werden.

2.1 Das Konzept

Um virtuelle Soundquellen zu erzeugen, müssen bestimmte Informationen ermittelt werden. Dazu gehören nach dem oben Gesagten

- das zugehörige Objekt
- ein passendes Soundfile
- Lautstärke
- Wiedergabemodus
- Position

Um diese Arbeitsschritte zu vereinfachen, wird in dem Framework auf eine Realisierung mittels intelligenter virtueller Objekte gesetzt, die die nötigen Informationen bereits gespeichert haben. Angelehnt an das InstantReality-Framework wird eine VR-Soundkomponente entwickelt, die weitere Informationen zu Objekten ausliest, speichert und mit Hilfe eines 3D-Soundservers wiedergeben kann (siehe Abbildung 1).

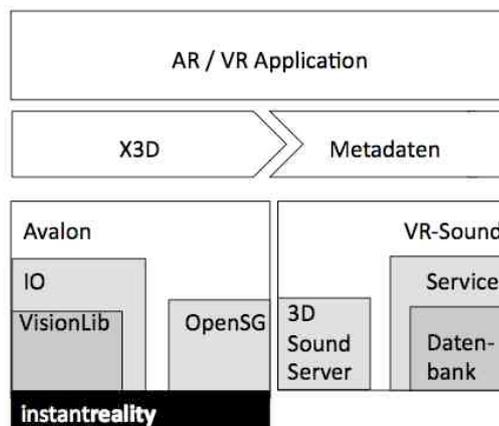


Abbildung 1: InstantReality mit zusätzlich integrierter Soundkomponente.

Das InstantReality Framework wurde vom Fraunhofer IGD und ZGDV entwickelt, um ein einfaches und konsistentes Werkzeug für AR- und VR-Entwickler bereit zu stellen [FBB09]. Aufbauend auf dem X3D-Standard können die Knoten des Szenengraphen mit so genannten Metadaten versehen werden. Auf diese Weise können den Objekten Eigenschaften verschiedener Ausgabemodalitäten in intuitiver Weise zugewiesen werden. Diese finden während der Interpretation des eigentlichen Quelltextes keine Beachtung. Deswegen eignen sie sich sehr gut um semantische Informationen zu hinterlegen.

Abbildung 2 zeigt exemplarisch die semantische Anreicherung einer virtuellen Welt. Als Veranschaulichung wurde eine Biosphäre gewählt, die in einem studentischen Projekt entwickelte wurde. Diese eignet sich gut, da viele Objekte, die Klänge erzeugen, vorhanden sind. Es wurden den Objekten Wasserfall und Feuer Informationen zur Klanggenerierung innerhalb der Metadaten zugewiesen. Diese haben die Struktur `<MetadataString name='NAME' value='WERT' containerField='value'>`. Die Felder `name` und `value` können dabei mit beliebigem Inhalt gefüllt werden. In diesem Beispielfall steht `name` für die Ausgabemodalität und `value` beschreibt den Klang den dieses Objekt wieder geben soll.

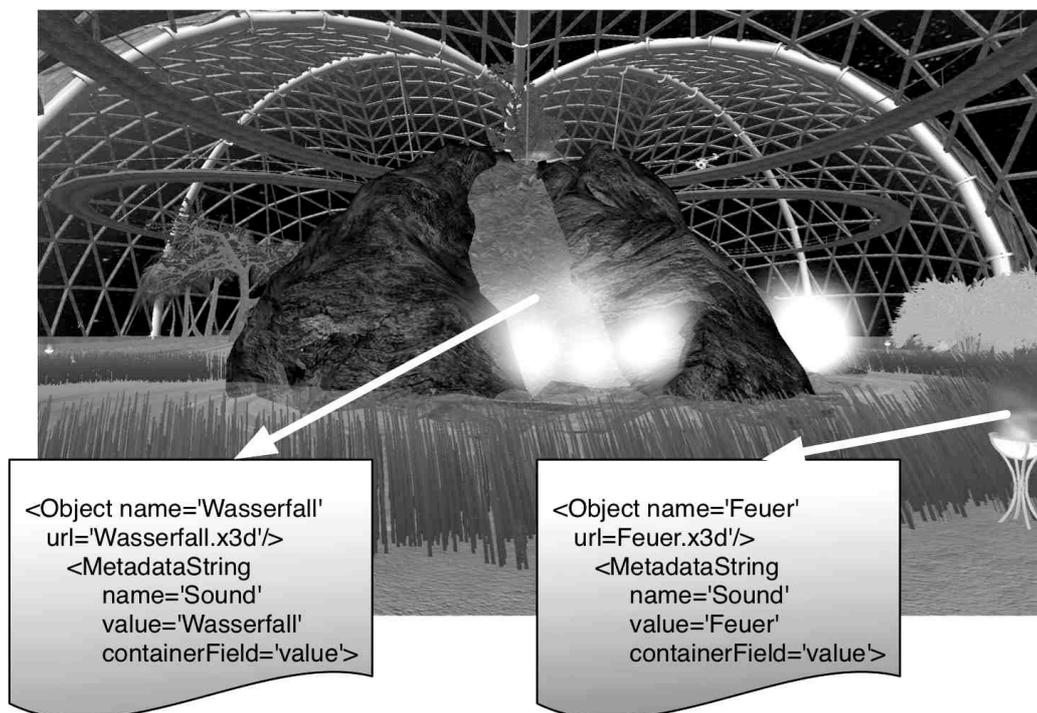


Abbildung 2: Semantisch angereicherte Welt am Beispiel Wasserfall und Feuer.

Zusätzlich gibt es die Möglichkeit zu einem Objekt mehrere verschiedene Metadaten anzugeben, welche dann zu einem <MetadataSet> zusammengefasst werden. Dies bietet einen Ausgangspunkt für das vorgestellte Framework, da so sehr leicht beispielsweise auch haptische oder klangliche Informationen hinterlegt und auf ein entsprechendes Ausgabemedium übertragen werden können.

2.2 Das technische Setup

Unser Arbeitsraum für die Anwendung des wissensbasierten Klangsystems ist eine dreiseitige CAVE (Front, linke Seite, Boden). Die Seiten werden jeweils von zwei Beamern beleuchtet, um so (mittels Zirkular-Polarisationsfiltern) einen orientierungsunabhängigen Stereoeffekt zu erzeugen. Jedes der insgesamt 6 Bilder wird von einem Renderclient erzeugt, zusätzlich gibt es einen Server für die interaktive Anwendung. Die Rechner sind untereinander über ein Infiniband-Hochgeschwindigkeitsnetzwerk mit einem Datendurchsatz von bis zu 20Gbit/s verbunden.

Für die Interaktion ist ein Trackingsystem (A.R.T.) mit 9 Kameras im Einsatz. Mit Hilfe von Infrarot-Markern kann so zuverlässig die Position des Benutzers verfolgt werden. Um den akustischen Eindruck im virtuellen Raum zu erzeugen, sind 8 Lautsprecher an den Ecken des Interaktionsraums angebracht. Zusätzlich sorgen 2 Subwoofer unter der Bodenprojektion für Tiefen. Abbildung 3 verdeutlicht schematisch den Aufbau der Cave mit Tracking- und Soundsystem.

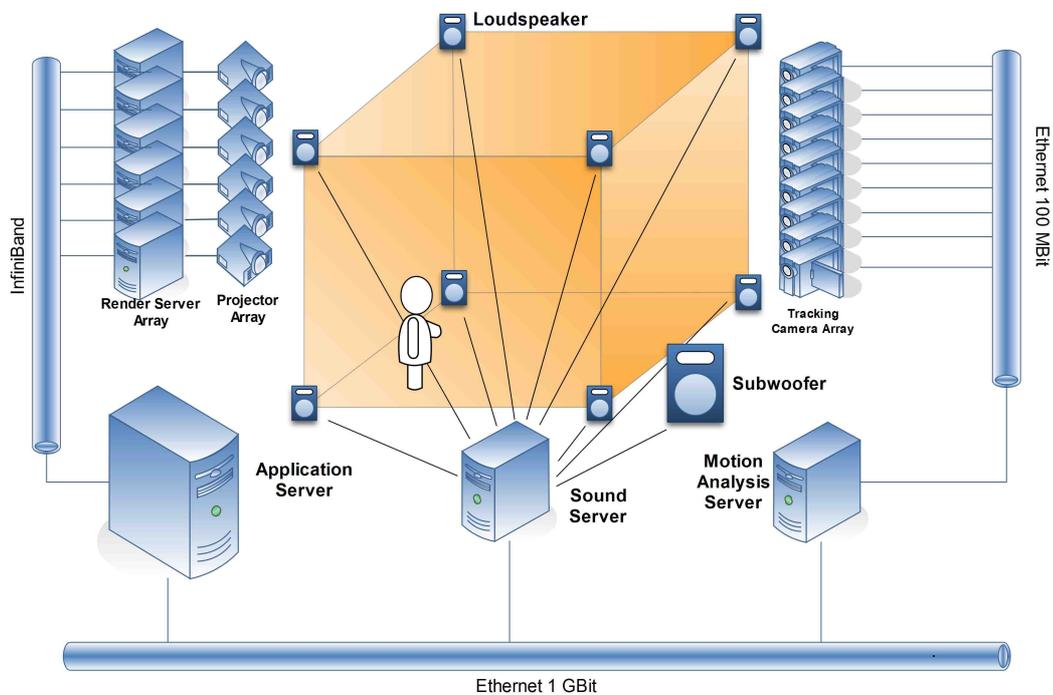


Abbildung 3: Skizze des technischen Setups.

3 Implementierung

Die praktische Umsetzung des wissensbasierten Klangsystems erfolgt in Java. Abbildung 4 zeigt den schematischen Aufbau der Vorverarbeitungskomponente. Diese unterteilt sich in drei Verarbeitungsschritte, welche im Folgenden genauer erklärt werden:

1. X3D-Datei einlesen
2. Metadaten mit einer Datenbank abgleichen
3. Informationen maschinenlesbar einfügen

Als Grundlage müssen X3D-Dateien vorhanden sein, in der die Objekte in Metadaten Informationen zu ihren Klangeigenschaften enthalten (das eigentliche Klangwissen). Dieses Wissen wird nicht separat in einer Wissensbasis vorgehalten, sondern ist direkt am Objekt verankert.

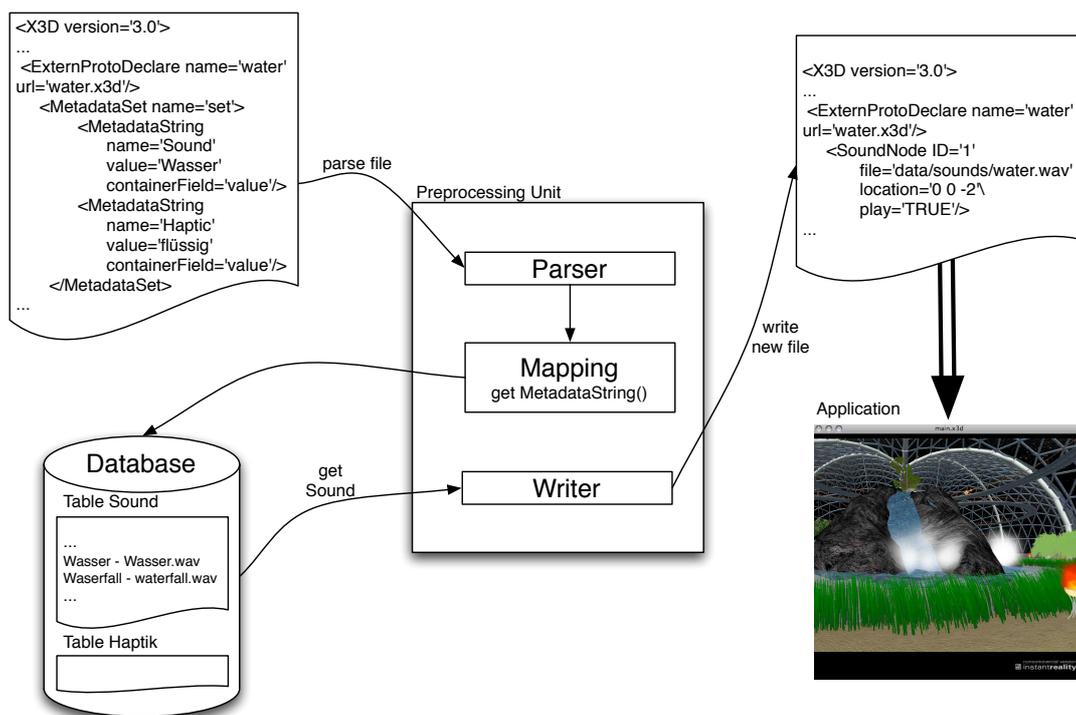


Abbildung 4: Schematischer Aufbau des wissensbasierten Klangsystems.

1. Einlesen und Suchen der semantischen Strukturen

In einem ersten Schritt wird die Datei mit einem XML-Parser (vom Typ JDOM) eingelesen und mit Hilfe der Abfragesprache XPath nach bestimmten Strukturen durchsucht. Die Metadaten können herausgefiltert und für eine weitere Verarbeitung gespeichert

werden. Zu jedem Metadaten-Set wird automatisch das zugehörige Objekt gespeichert, um den zu generierenden Soundknoten dem richtigen Objekt zuzuweisen. Der Soundknoten wird im Szenengraphen unter das zugehörige Objekt gehängt, um auch bei bewegten Objekten eine realistische Soundausgabe zu ermöglichen.

Diese Metadaten werden von dem Parser ausgelesen und anhand der Werte im Feld **name** gruppiert. Der Name **Sound** drückt also aus, dass es sich um eine Klangeigenschaft handeln muss. Das Feld **value** beinhaltet den konkreten Wert, welcher mit einer internen Sound-Datenbank abgeglichen wird.

2. Abgleich mit der Datenbank

In einem zweiten Schritt werden die gespeicherten Werte mit der Datenbank abgeglichen. Der Eintrag im Feld **name** beschreibt die Ausgabemodalität und bildet auf eine Tabelle ab. Innerhalb dieser Tabelle wird mittels des Wertes aus dem Feld **value** nach einem passenden Datenbankeintrag gesucht. Wenn ein passender Eintrag gefunden wird, kann so das korrespondierende Soundfile an die nächste Verarbeitungskomponente (JDOM-Writer) weiter gegeben werden. Sollte kein passender Eintrag existieren, wird dem Objekt kein Soundfile zugewiesen und auf der Konsole eine entsprechende Fehlermeldung ausgegeben.

Im Rahmen dieses Projekts wird eine Datenbank aufgebaut, die ein möglichst großes Spektrum an Sounds abdecken soll. Zu den weiteren Anforderungen gehört es, dass sie auch über eine hinreichend gute semantische Annotation verfügen muss um dem Entwickler die Möglichkeit zu geben feine Abstufungen innerhalb seiner Anwendung auch akustisch zu verdeutlichen. Die Soundfiles stammen hauptsächlich aus frei verfügbaren Online-Quellen [SB-ol]. Die semantische Annotation der Datenbank erfolgt per Hand. Um dem persönlichen Wortschatz eines Entwicklers entgegenzukommen, soll der Parser zusätzlich durch ein Wortnetz unterstützt werden. So wird es ermöglicht auch Synonyme zuzuordnen, und die aufwendige Annotation der einzelnen Sounds kann eingegrenzt werden.

3. Anlegen der neuen Knoten

In einem letzten Schritt werden konkrete Soundknoten in dem X3D File angelegt. Hierzu wird der korrespondierende JDOM-Writer benutzt, welcher an bestimmten Stellen im eingelesenen Dokument neue Knoten einfügt. Diese bekommen eine eindeutige ID, den Sound aus der Datenbank, eine Position und einen Wiedergabemodus zugewiesen. Anschließend kann mittels eines Startscripts automatisch die neu generierte Datei gestartet werden.

Durch diese drei Schritte ist es zunächst möglich, virtuelle Welten mit konstanten Klängen anzureichern. Mit dem bisher beschriebenen Mechanismus ist es beispielsweise möglich, einen Wasserfall dauerhaft zum Klingen zu bringen, und parallel dazu eine Vielzahl weiterer konstanter Sounds wieder zu geben. Da die Welt aber nicht nur aus

Objekten besteht, die dauerhaft einen gleichbleibenden Klang haben, müssen noch spezielle Objekteigenschaften beachtet werden, um ein angemesseneres Klangerlebnis zu schaffen.

3.1 Objekte mit nicht konstantem Sound

Viele Objekte haben einen nicht immer gleichbleibenden Sound. Während zwar der Wasserfall immer rauscht, geben andere Objekte Geräusche ab. Um auch diese Fälle abzudecken, wurde der Datenbank ein weiteres Feld hinzugefügt. Dabei werden derzeit zwei Fälle unterschieden:

Objekte mit unregelmäßigen Klangabständen: Zur Illustration soll hier ein Vogel betrachtet werden, der in unregelmäßigen Abständen Töne von sich gibt. Bei entsprechenden Soundfiles ist in der Datenbank ein weiteres Feld mit einem **random**-Eintrag versehen. Diesem Eintrag kann ein Wert zwischen 0 und 100 zugewiesen werden. 0 bedeutet dabei kein Sound und 100 wäre ein konstanter Sound. So kann mit verschiedenen Werten die Wiederholfrequenz gesteuert werden.

Jeder dafür in Betracht kommende Sound ist mit einem Default-Wert belegt. Wird in dem **value**-Feld der Metadaten nach dem Namen ein ‚random x‘ angegeben, wird der default-Wert überschrieben. Wenn der entsprechende Soundknoten (von dem JDOM-Writer) angelegt wird, werden diese Werte in den Wiedergabemodus eingetragen. Dementsprechend kann eine Funktion aufgerufen werden, um den Sound in zufälligen Zeitintervallen abzuspielen.

Objekte mit regelmäßigen Klangabständen: Andere Fälle erfordern eine Wiedergabe in immer gleichen Zeitabständen. So ist es zum Beispiel möglich eine Kirchturmuhre jede Viertelstunde schlagen zu lassen. Dafür ist in der Datenbank ein **regular** eingetragen. Auch hier ist es möglich die vorab generierten Defaults zu überschreiben. In diesen Fällen wird dann eine Funktion aufgerufen, die abhängig von der Systemzeit in den gegebenen Intervallen das Abspielen der Soundfiles veranlasst.

3.2 Objekte mit größerer Ausdehnung

Ein weiterer Aspekt, der beachtet werden muss, ist dass es Objekte gibt, die eine räumliche Ausdehnung haben. Es somit nicht praktikabel, nur an einer Position das Soundfile einzuhängen. Sonst hätte der Anwender beispielsweise das Gefühl sich von einem Fluss zu entfernen, obwohl er nur an ihm entlang läuft.

Als Lösungsansatz wurde hier ein Verfahren gewählt, das immer den kürzesten Weg von dem Benutzer zu dem Objekt berechnet. Es wird eine Art Schnur durch das Objekt gespannt, an dem die Soundquelle mit dem Benutzer entlang wandert. So wird bei einem virtuellen Spaziergang entlang eines Flusses die Immersion nicht durch ein falsches Klangerlebnis gestört.

3.3 Soundausgabe

Die Soundausgabe erfolgt über ein Client-Server-Modell, das über das User Datagram Protocol (UDP) kommuniziert. Dabei handelt es sich um ein minimales, verbindungsloses Netzwerkprotokoll. Es wird nicht auf die Sicherung der Datenübertragung geachtet, um Verzögerungen bei der Übertragung zu verhindern. Als Voraussetzung dafür, wurde für den technischen Aufbau in der CAVE ein Interface entwickelt, welches auf der Audiobibliothek FMOD, der Firma ‚Firelight Technologies‘ aufbaut. Dieses Interface wird für die Wiedergabe genutzt.

Der Server ist in Java implementiert und übernimmt die Aufgabe, das Interface für die FMOD Audiobibliothek zu benutzen. Der Client ist ebenfalls in Java implementiert, wird aber aus der InstantReality Anwendung aufgerufen. Somit sind Teile des Clients in X3D realisiert worden. Abbildung 5 veranschaulicht die Kommunikation zwischen den einzelnen Komponenten. Es gibt N Soundinstanzen, die ihre Daten an den Communicator senden. Dieser benutzt den JSoundClient, um die Daten an den Server zu übermitteln. Dieser benutzt den JSoundServer, um die Daten an den FMOD zu übermitteln.

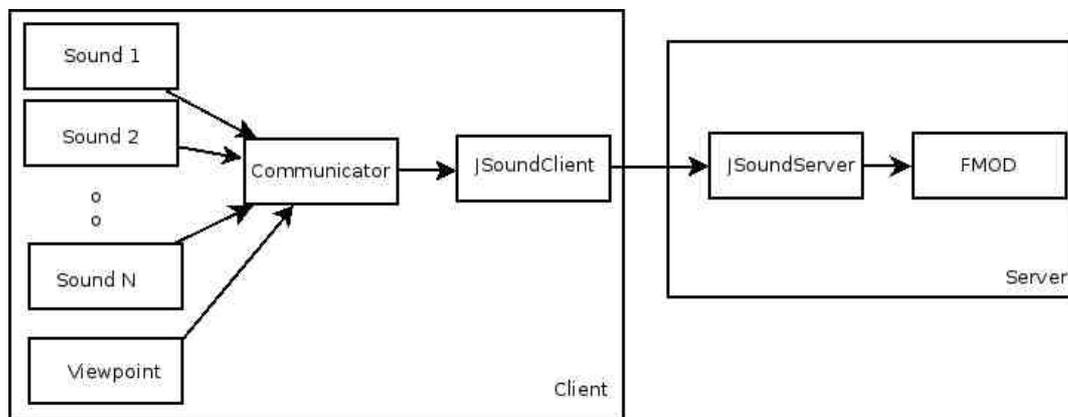


Abbildung 5: Veranschaulichung der Soundwiedergabe.

Für die Simulation der Richtung wird das ‚Vector based amplitude panning‘ verwendet. Es werden die 3 zum Objekt nächstliegenden Lautsprecher ermittelt. Auf diesen wird das Soundfile abgespielt. Um die Entfernung darzustellen, wird die Lautstärke reguliert. Dabei gilt, dass eine Verdopplung der Entfernung bedeutet, dass die Lautstärke um 6 dB abnimmt [Beg94].

4 Resümee und Ausblick

Erläutert wurde ein wissensbasiertes Konzept zur 3D-Klanggenerierung in virtuellen Welten. Die Komponenten ermöglichen es, dem Anwender eine realistische akustische Ausgabe zur Verfügung zu stellen. Somit wird es ermöglicht, mit sehr geringem Entwicklungsaufwand eine weitere Ausgabemodalität hinzuzufügen.

Ein wichtiger Faktor für den Realismus liegt allerdings in der Arbeit des Entwicklers. Durch die semantische Annotation gibt es keine vorgeschriebenen Werte. So ist die Angabe ‚Wasser‘ sehr allgemein und könnte unter Umständen zu einem nicht ganz realistischen Eindruck führen. Würden aber Werte wie kleiner Bach, Wasserfall, großer Wasserfall, See, Meer, Bucht etc übergeben, können passende Soundfiles aus der Datenbank gesucht werden. Während der zukünftigen Arbeiten wird dabei auch Wert darauf gelegt, die Sounddatenbank so vollständig wie möglich anzulegen.

Weiterführende Arbeiten werden sich vor allem mit der Generierung von Sounds während der laufenden Anwendung befassen. Ein wichtiger Aspekt von Klängen in der realen Welt besteht aus dem Zusammentreffen von mehreren Objekten. Solche event-basierten Klänge werden in dem aktuellen Framework noch nicht berücksichtigt. Es sollen auch zur Laufzeit Klänge erzeugt und gestoppt werden können, um weitere Facetten des akustischen Feedbacks zu ermöglichen. So kann die Interaktivität einer Virtuellen Welt auch im akustischen Bereich weiter unterstützt werden.

In zukünftigen Arbeiten soll auch darauf eingegangen werden wie sich verschiedene Soundquellen gegenseitig beeinflussen. Dabei soll ein Raumklang entstehen, der die Beeinflussung vorhandener Sounds miteinander verrechnet, um somit eine noch realistischere Ausgabe zu ermöglichen.

Literatur

- [Beg94] BEGAULT, D.R.: 3-D Sound for Virtual Reality And Multimedia. Academic Press, 1994
- [FBB09] FELLNER, D.W.; BEHR, J.; BOCKHOLT, U.: instantreality – A Framework for Industrial Augmented and Virtual Reality Applications. Virtual Reality & Augmented reality in Industry, Shanghai, 2009
- [HNG09] HUNSUNG, S.; NETHER, J.; GRAMSTAT, S.: Auralisierung simulierter akustischer Eigenschaften eines Hinterachsgetriebes unter Nutzung von Virtual Reality. 8. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung 2009, Heinz Nixdorf Institut, Universität Paderborn
- [LA00] LUCK, M.; AYLETT, R.: Applying artificial intelligence to virtual reality: Intelligent virtual environments. Applied Artificial Intelligence, 14(1), 3-32, 2000
- [LBW05] LATOSCHIK, M.E.; BIERMANN, P.; WACHSMUTH, I.: Knowledge in the Loop; Semantics Representation für Multimodal Simulative Environments. International Symposium on Smart Graphics, Berlin: Springer (LNCS 3638), 2005

- [SGG07] SHEN, Q.; GRAFE, M.; GAUSEMEIER, J.: Systemkomposition mechatronischer Prototypen in virtuellen Umgebungen. 6. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung 2007, Heinz Nixdorf Institut, Universität Paderborn
- [SB-ol] SoundBible.com: Free SoundClips, Sound Bites, and Sound Effects, unter: <http://www.SoundBible.com>; zuletzt besucht: 02. Februar 2011
- [Vor08] VORLÄNDER, M.: Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality. Springer Verlag, 2008

Autoren

Julia Fröhlich, B.Sc. studierte Kognitive Informatik und Bioinformatik an der Universität Bielefeld. Sie ist Mitarbeiterin der Arbeitsgruppe Wissensbasierte Systeme von Prof. Dr. Ipke Wachsmuth an der Universität Bielefeld und arbeitet an ihrer Dissertation im Bereich der wissensbasierten Klanggenerierung.

Prof. Dr. Ipke Wachsmuth studierte Mathematik und Informatik an der TU Hannover, wo er 1980 promovierte. Nach Tätigkeiten an der Universität Osnabrück, der Northern Illinois University und bei IBM Deutschland habilitierte er sich 1989 an der Universität Osnabrück; im gleichen Jahr wurde er auf eine Professur für Wissensbasierte Systeme an der Universität Bielefeld berufen. Er war dort Gründungsmitglied der Technischen Fakultät (1990), Mitinitiator zweier Sonderforschungsbereiche und von 2002 bis 2009 geschäftsführender Direktor des Zentrums für interdisziplinäre Forschung (ZiF). Er ist derzeitiger Sprecher des Sfb 673 „Alignment in Communication“.