

[VR]Instrument – Das virtuelle Piano

Michael Haller, Markus-Juan Bohensky, Stephan Enzinger, Markus Gerhard,
Harald Milchrahm

Fachhochschule Hagenberg, Medientechnik und –design, Hagenberg, AUSTRIA,
[michael.haller | markus.bohensky | stephan.enzinger |
markus.gerhard | harald.milchrahm] @ fhs-hagenberg.ac.at

Zusammenfassung: *[VR]Instrument ist eine VR-Applikation, bei der man virtuell Piano spielen kann. Der Benutzer kann mittels Datenhandschuh die Finger bewegen und einen über MIDI-Interface angeschlossenen Synthesizer ansteuern. Ziel dieses Prototypen war es vor allem, festzustellen, ob ein Datenhandschuh geeignet sein könnte, um Bewegungsabläufe für Musikinstrumente zu trainieren bzw. zu verbessern.*

1. Einleitung

Das Projekt **[VR]Instrument** entstand im Rahmen der Lehrveranstaltung „Anwendungsbezogene Seminare und Projekte“ an der Fachhochschule Hagenberg, Medientechnik und –design [7]. Die Idee, die wir dabei hatten war, dass eine Person mittels der Datenhandschuhe ein virtuelles Musikinstrument spielen sollte. Vor allem der Umgang mit den Datenhandschuhen bzw. die Machbarkeit einer solchen Applikation mittels der Datenhandschuhe sollte dabei besonders im Mittelpunkt stehen. Ursprünglich wurde nicht festgelegt, welches Instrument implementiert werden sollte. Die Palette reichte von einer virtuellen Knopfgel bis hin zu einem Schlagzeug. Da mittels der Datenhandschuhe jedoch alle Finger einer Person „getrackt“ werden, war es naheliegend, eine Applikation zu entwickeln, bei der eine hohe Fingerfertigkeit gefragt war. Schließlich entschieden wir uns für eine etwas einfachere Variante der Knopfgel und implementierten ein virtuelles Piano, bei welchem der Benutzer mittels der Datenhandschuhe über ein MIDI-Interface ein Synthesizer ansteuern können sollte.

Wenn man sich die Palette der VR-Applikationen ansieht, bei denen ein Datenhandschuh im Einsatz kommt, dann sind das vor allem Programme, bei denen das Fingerspitzengefühl trainiert wird oder wo man

Routinearbeiten durchführt, welche einen hohen Präzisionsgrad verlangen. Die Anwendungen reichen dabei vom Erlernen der Gebärdensprache bis hin zur Steuerung eines Roboters mittels Dataglove [3, 4].

In der Applikation [VR]Instrument wollten wir nun testen, ob es überhaupt möglich ist, ein virtuelles Klavier mittels der Datenhandschuhe zu bedienen. Weiters wollten wir feststellen, ob die Reaktionszeit bzw. die Geschwindigkeit der Datenhandschuhe unseren Anforderungen entspricht. In weiterer Folge sollte der Prototyp erweitert werden. Dabei sollte er vor allem für Schulungszwecke eingesetzt werden, bei denen man bestimmte Handstellungen bzw. Fingerabläufe beim Erlernen eines Musikinstrumentes trainieren könnte und bei falschen Fingerstellungen einen Hinweis bekommt, dass man das Klavierstück zwar akustisch richtig, aber von der Fingerstellung her falsch gespielt hat.

2. Der Aufbau

Das System besteht aus den folgenden Komponenten: für das Tracking, also die Ermittlung der Koordinaten des Datenhandschuhs im Raum, ist der Polhemus Fastrak verantwortlich [1]. Die Fingerbewegungen werden durch den CyberGlove der Firma Immersion Corporation ermittelt [8]. Leider unterstützte das von uns benutzte Modell des Datenhandschuhs kein Force Feedback, das wir in späterer Folge noch vermissen sollten. Weiters dient für die akustische Ausgabe ein Yamaha Synthesizer, der mittels MIDI-Schnittstelle zum PC verbunden wurde. Das Programm läuft auf einem PC unter Microsoft Windows 2000. Zur graphischen Ausgabe diente eine GeForce II GTS.

3. Das Szenario

Hat der Benutzer einmal die Applikation gestartet und den Datenhandschuh angezogen, so kann er mühelos „klavierspielen“. Dies wird durch die modellierte Szene, die aus einem Klavier besteht, ermöglicht, da sich der Benutzer in einer natürlichen Umgebung wiederfindet.



Abbildung 1: Das [VR]Instrument-Hardware-Setup.

Um dem Benutzer ein entsprechendes Feedback geben zu können, ist es nun notwendig, nicht nur eine akustische Ausgabe zu erzwingen, sondern auch eine visuelle Ausgabe zu erzeugen (siehe Abschnitt 4.1).

3.1 Szenenaufbau

Die gesamte Szene bestehend aus einem Klavier und den Tasten wurde in 3D Studio Max modelliert, in OBJ-Files exportiert und anschließend mit einem Loader in OpenGL integriert. Das Modell für die Interaktionshand hingegen stammt von der VHT-Bibliothek.

3.2 Interaktion

Der Datenhandschuh hat den großen Vorteil, dass jede kleine Fingerbewegung registriert und erkannt wird. Auf diese Weise kann der Benutzer eine sehr intuitive Handbewegung machen und diese wird auf sehr natürliche Weise im Programm sichtbar gemacht. Kommt es zu einer Kollision, wird ermittelt, welche Taste gedrückt worden ist. Anschließend wird ein MIDI-Befehl, welcher die Note dieser Taste repräsentiert, an den Synthesizer geschickt, der seinerseits die Note akustisch hörbar macht. Weiters wird auch die Taste rotiert, sodass der Benutzer auch ein visuelles Feedback von der gedrückten Taste erhält.



Abbildung 2: Mittels Datenhandschuh kann der Benutzer mit dem virtuellen Piano spielen und über MIDI den Synthesizer ansteuern.

4. Implementierung

4.1 Die Integration des Datenhandschuhs

Immersion Corporation liefert mit den Datenhandschuhen eine umfangreiche Bibliothek für C++. Diese API (Virtual Hand Toolkit, kurz VHT) ist Teil der Virtual Hand Suite 2000 und ist relativ einfach zu bedienen (vgl. [5]). Neben einem haptischen und einem neutralen Szenengraphen (mehr dazu in [5]) bietet die VHT-Bibliothek eine gute Unterstützung für die Interaktion und Simulation der Datenhandschuhe, sowie eine integrierte Kollisionserkennung mit Objekten und eine einfache Integration in die OpenGL-Umgebung. Auch die zwei bekanntesten magnetischen Trackingsysteme (Polhemus [1] und Ascension [6]) werden softwaremäßig von der VHT-Bibliothek unterstützt, und es war deshalb relativ einfach, unser Fastrak-Trackingsystem zu integrieren.

4.2 Die graphische Umsetzung

Um dem Benutzer die Bedienung des Programmes [VR]Instrument zu erleichtern, haben wir das Piano in der Szene auch visuell integriert. Als Graphik API diente OpenGL. OpenGL deshalb, weil wir unabhängig von der VHT-Bibliothek sein wollten, und deshalb nicht die graphische Szenengraphen-API von VHT nutzten. Sämtliche Objekte wurden mit 3D Studio Max modelliert und in OBJ-Modelle umgewandelt, die schließlich in OpenGL geladen werden konnten.

4.3 MIDI – Die Schnittstelle zwischen Anwendung und Ausgabe

Als Schnittstelle zwischen Anwendung und Sound-Ausgabe haben wir MIDI (Musical Instrument Digital Interface) verwendet. Für die Ausgabe über MIDI diente die MultiMedia-System-API von Microsoft, bei der das Aufrufen von MIDI-Events und das Abspielen von Klängen ganz einfach ermöglicht wird. Eine genauere und detaillierte Beschreibung der API bzw. der MIDI-Schnittstelle kann der Leser in [2] finden.

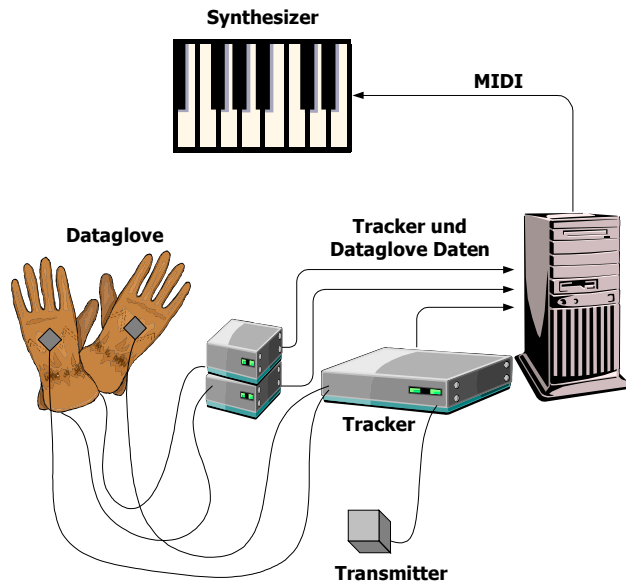


Abbildung 3: Vom Datenhandschuh und vom Tracker kommen die Daten, die dann über den PC und über MIDI an den Synthesizer weitergeleitet werden.

4.4 Von der Eingabe zur Ausgabe

Wenn der Benutzer nun mit einem Finger eine Klaviertaste anstößt, kommt es zu einer Kollision zwischen dem Finger der virtuellen Hand und dem entsprechenden Knoten im haptischen Szenengraphen, welcher genau diese Taste repräsentiert. Wir haben in unserem Prototypen nicht den neutralen Szenengraphen genutzt, der eine Verbindung zwischen dem haptischen und dem graphischen Szenengraphen herstellen würde, sondern wir haben uns selber um diese Verbindung gekümmert. Somit haben wir zwar den haptischen Szenengraphen auf der einen Seite genutzt und auch alle bedienbaren Objekte in ihm gespeichert, jedoch die graphische Visualisierung selber in die Hand genommen und die Verknüpfung von Haptik und Graphik selber gesteuert. Die graphische Ausgabe erfolgte bei uns gänzlich in OpenGL.

5. Ergebnisse und Tests mit Personen

Nach erfolgter Entwicklung des Systems, gingen wir daran, es nun zu testen. Zu diesem Zwecke luden wir diverse Testpersonen ein. Teilweise waren diese Personen musikalisch – zum Teil spielten sie sogar Klavier. Teilweise waren aber unter den Testpersonen auch welche dabei, die noch nie zuvor ein Instrument in der Hand gehabt hatten. Schnell konnten wir erkennen, dass Personen, die noch nie auf einem Klavier gespielt hatten, relativ schnell einfache Melodien spielen konnten. Geübte Spieler jedoch, die gleich einige Akkorde zu spielen versuchten, stießen schnell an ihre Grenzen, da sie das haptische Gefühl eines echten Klaviers vermissen.

5.1 Ergebnisse für den Programmierer

Dank des Szenengraphen-Konzepts der VHT-API ist es dem Programmierer leicht möglich, eine komplexe Szene mit integrierter Kollisionserkennung aufzubauen. Man erzielt dadurch relativ rasch brauchbare Ergebnisse. Um so mehr muss man sich um die Installation, Kalibrierung der einzelnen Komponenten und die manchmal auftretenden Konflikte zwischen Hard- und Software kümmern.

5.2 Endergebnisse für den Endbenutzer

Für den Benutzer bietet unser Prototyp einen relativ einfachen und schnellen Einstieg in die virtuelle Welt. Aufgrund der Szene und der einfachen Bedienung ist es auch Personen, für die der Computer eher fremd ist, ein Einfaches, mit [VR]Instrument ein Klavier zu spielen.

Dank der äußerst kurzen Reaktionszeit, der präzisen Erkennung der Fingerstellung des Handschuhs und der besonders gut funktionierenden Kollisionserkennung lässt sich [VR]Instrument leicht und präzise bedienen. Für den professionellen Einsatz ist [VR]Instrument allerdings nicht gedacht, da sich geübte Spieler durch den Datenhandschuh in ihrer Bewegung etwas beeinträchtigt fühlten. Außerdem vermissen einige der Testpersonen, die bereits Klavier spielen konnten, den Widerstand der Taste beziehungsweise das haptische Gefühl, wenn man die Klaviertaste wieder ausließ. Durch die Weiterentwicklung des Systems mittels Force Feedback Technologie wäre dieses Problem jedoch zu lösen.

6. Resümee und Ausblick

Zusammenfassend lässt sich sagen, dass der Einsatz des Datenhandschuhs für das Spielen eines Klaviers nur dann geeignet bzw. sinnvoll ist, wenn man noch keinen Kontakt mit dem Klavier hatte und die Fingerstellungen üben

möchte. Obgleich der Datenhandschuh relativ gut handhabbar ist, waren es gerade die „Profi-Klavierspieler“, die sich über dem Abstand der Finger beschwerten, weil durch den Handschuh der Zwischenraum der Finger etwas größer war als sonst. Nichtsdestotrotz waren vor allem Klavierspieler von der Performance überrascht. Im weiteren Verlauf planen wir nun, das Programm für das Erlernen von gewissen Fingerstellungen auszubauen bzw. das Programm auch für das Erlernen von Gitarrengriffen zu erweitern.

7. Referenzen

[1] Polhemus, *Fastrak User's Manual, Revision*, November 1993, Technischer Report, OPM3609-002C, <http://www.polhemus.com> (Stand: Juli 2001)

[2] J. Glatt, *What is MIDI?*, 2001, <http://www.borg.com/~jglatt/tech/miditech.htm> (Stand: Juli 2001)

[3] Chris Culbert, Telepresence Interface, DART (Dexterous Anthropomorphic Robotic Testbed), http://vesuvius.jsc.nasa.gov/er_er/html/dart/index.html (Stand: Juli 2001)

[4] Sigrid Brunner, Multimedia und 3D für Gehörlose, <http://www.fhs-hagenberg.ac.at/mtd/projekte/FFF/3dSign/english.html> (Stand: Juli 2001)

[5] Ron Carmel, Chris Ullrich, Jason Silver, *VirtualHand v2.5 – User's Guide*, Technischer Report, Immersion Corporation, 2001

[6] Ascension, <http://www.ascension-tech.com>, (Stand: Juli 2001)

[7] [VR]Instrument-Homepage, <http://www.fhs-hagenberg.ac.at/staff/haller/projects/ss2001/vrinstrument> (Stand: Juli 2001)

[8] Immersion Corporation, *Immersion 3D's award-winning instrumented glove*, 2001, <http://www.immersion.com/products/3d/interaction/cyberglove.shtml> (Stand: Juli 2001)