

Aufbau eines VR-Systems zur multimodalen Interaktion mit komplexen physikalischen Modellen

Guido Böttcher, Rasmus Buchmann, Maximilian Klein, Franz-Erich Wolter

Institut für Mensch-Maschine-Kommunikation

Welfengarten 1

30167 Hannover

Tel.: +49 (0)511 / 762 - 2910

Fax: +49 (0)511 / 762 - 2911

E-Mail: {boettcher,buchmann,mklein,few}@welfenlab.de

Zusammenfassung: Wir stellen ein neuartiges, semi-immersives VR-System vor, welches neben visuellem auch haptisches Feedback in einem großen Arbeitsbereich ermöglicht. Die auf den Verwendungszweck angepasste haptische Hardware gekoppelt mit einem 3D-Projektionssystem soll eine natürlichere Interaktion mit der VR-Umgebung bieten. Zusätzlich soll mit einer hybriden Rechnerstruktur die Simulation komplexer mechanischer Modelle in haptischer Echtzeit erreicht werden. Bestehend aus einem auf Cell-Prozessoren basierenden HPC-Cluster und einem GPU-Subsystem, besitzt unser VR-System die nötige Rechenleistung. Wir geben zunächst einen Überblick über den Aufbau und die Strategien zur Ausnutzung der verwendeten Hardware. Die hierfür entwickelten Softwarekomponenten wurden in ein freies modulares Simulationsframework eingebettet. Ein erster Vergleich der Leistungsdaten unserer Textilsimulation, die auf den Cell Prozessor portiert wurde, zeigt das mögliche Potential des hybriden Systems auf.

Stichworte: VR-System, Haptik, verteilte Systeme

1 Einleitung

VR-Systeme finden heutzutage vielfältige Anwendungen in verschiedenen Wissensgebieten. Wenn z.B. technische Konstruktionen geplant, Simulationsdaten ausgewertet oder Gefahrenzonen untersucht werden sollen, dann kann mithilfe eines VR-Systems dem Benutzer eine verbesserte Vorstellungen der Situation vermittelt werden. Bei der Entwicklung eines solchen VR-Systems wird häufig der Fokus auf eine visuelle Immersion gelegt. Anhand von stereoskopischen Darstellungen durch 3D-Projektionssysteme mit vorzugsweise fotorealistischem Anspruch werden Personen in die virtuelle Szenerie versetzt und erhalten somit einem räumlichen Eindruck der simulierten Umgebung. Dieser Schwerpunkt ist jedoch für manche Anwendungszwecke unzureichend, da der Benutzer meist nur beschränkte Interaktionen innerhalb der simulierten Realität durchführen kann. Dies liegt vornehmlich daran, dass der Eingriff in die Umgebung nur eine unvollständige sensorische Rückkopplung bietet. Obwohl der Tastsinn das fundamentale Instrument zur Kontrolle unserer Interaktionen mit der Um-

welt ist, werden diesem keine oder nur unzureichende Informationen zurückgeliefert. So wird durch ein gebräuchliches Interaktionsgerät, wie z.B. einen räumlich erfassten Stab oder auch einen Datenhandschuh, zwar die Manipulationsmöglichkeit des Benutzers räumlich korrekt dargestellt, jedoch werden deren physikalische Auswirkungen in Form von Deformationskräften oder Trägheitswiderständen nicht vermittelt. Dies führt dazu, dass die motorischen Aktionen des Benutzers nicht der intendierten Manipulation entsprechen und somit nur durch langwieriges Training zu den gewünschten Ergebnissen führen. Im Folgenden wird daher ein neues, intuitives semi-immersives VR-System vorgestellt, welches neben dem visuellen auch ein haptisches Feedback in einem großen Arbeitsbereich ermöglicht. Eine hybride Systemarchitektur bestehend aus einem Computercluster und einer GPU-Recheneinheit soll darüber hinaus ermöglichen, komplexe physikalische Simulationen in haptischer Echtzeit darzustellen. Als erstes Beispiel wurde die physikalische Simulation einer Textilie auf den Rechnercluster portiert und für die zugrundeliegende Architektur optimiert.

2 Bisherige Arbeiten

In [JHS05] wurde ein ähnlich immersives System vorgestellt. Die stereoskopische Projektion erfolgte in diesem Fall auf eine gekrümmte Leinwand. Dadurch bietet das System im Vergleich zu unserer Lösung eine höhere Immersion, verwendete jedoch eine einfache physikalische Simulation. Zur multimodalen Eingabe dienten ein kabelbasiertes Force Feedback Gerät, sowie eine druckgesteuerte Plattform, die Position und Schwerpunkt des Benutzers misst. Darüber hinaus werden in [GASM08] alternative haptische Geräte für Anwendungen mit großem Arbeitsbereich vorgestellt, wie z.B. durch längere Hebel in der kinematischen Kette oder ein Aufbau mit zusätzlichen, redundanten Freiheitsgraden.

Altomonte et al. [AZBF08] haben eine GPU-basierte Methode zur medizinischen Simulation vorgestellt. Dafür wurde ein Feder-Massen-Modell mit Volumenkonstanz implementiert. Sie konnten zeigen, dass ihre GPU-basierte Simulation, trotz der hohen Komplexität des Modells, sehr kurze Latenzzeiten von 0.68ms auf einer GeForce8 erreichen konnte. Dadurch war es möglich die Kraftberechnung für haptische Anwendungen auf die Grafikkarte auszulagern.

In [ACF⁺06] wurde eine, auf der GPU arbeitende Implementation erstellt, die zusätzlich zu der Laufzeitanalyse auch den Overhead durch die *Simulation Open Framework Architecture* (SOFA) [ACF⁺07] untersucht. Dabei entstand ein CUDA spezifischer Overhead, der durch den flexiblen Aufbau von SOFA verursacht wurde.

Für die haptische Kontaktsituation wurde in [CA09] das zugehörige Optimierungsproblem, das auf die Lösung eines linearen Gleichungssystem hinausläuft, mit dem Gauss-Seidel-Algorithmus auf aktuellen GPUs gelöst. Dabei wurde eine spezielle Partitionierung verwendet, die die Synchronisationskosten verringert und die Parallelisierbarkeit des Algorithmus, bei stark besetzten Gleichungssystemen, verbessert. Besonders für größere Kontaktprobleme ist die GPU-basierte Berechnung einer CPU-Implementation klar überlegen. Youquan et al. haben eine GPU-basierte Lösung zur haptischen Operationssimulation auf der Basis von CUDA vorgestellt [YS08], die die Echtzeitfähigkeit dieses Verfahrens demonstriert.

3 VR-System Hardware

Im Folgenden stellen wir den Hardware-Aufbau des VR-Systems vor und erklären die Funktion der dabei verwendeten Hardwarekomponenten. Wie in Abb.1 dargestellt, ist das System strukturell in vier Komponententypen unterteilt. Die Kernkomponente des Aufbaus stellt die Berechnungseinheit dar. Sie ist zentral für die Ein- und Ausgabeverarbeitung verantwortlich und koordiniert den Datenaustausch mit den anderen Komponenten. Aufgrund der hohen Anforderung an das System, numerische Berechnungen zur Simulation komplexer physikalischer Vorgänge durchzuführen und diese auch haptisch interaktiv darzustellen, ist die Einheit in drei Elemente unterteilt, die verschiedene Teilaufgaben wahrnehmen. Das zentrale Element bildet eine Hochleistungsworkstation, an die zwei Komponenten zur Rechenbeschleunigung gekoppelt sind, die die Workstation bei der Bewältigung von numerischen Berech-

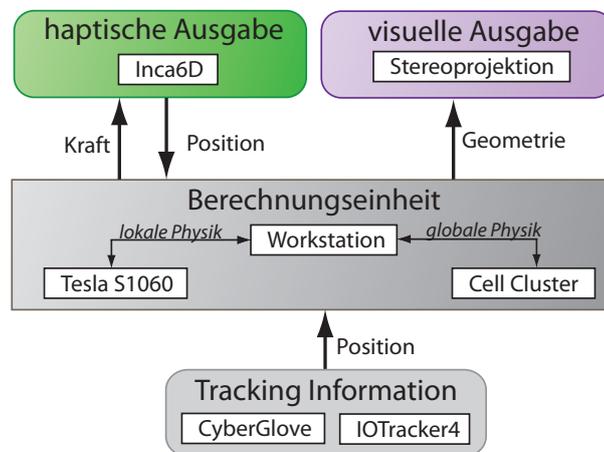


Abbildung 1: Übersicht der Komponenten des VR-Systems

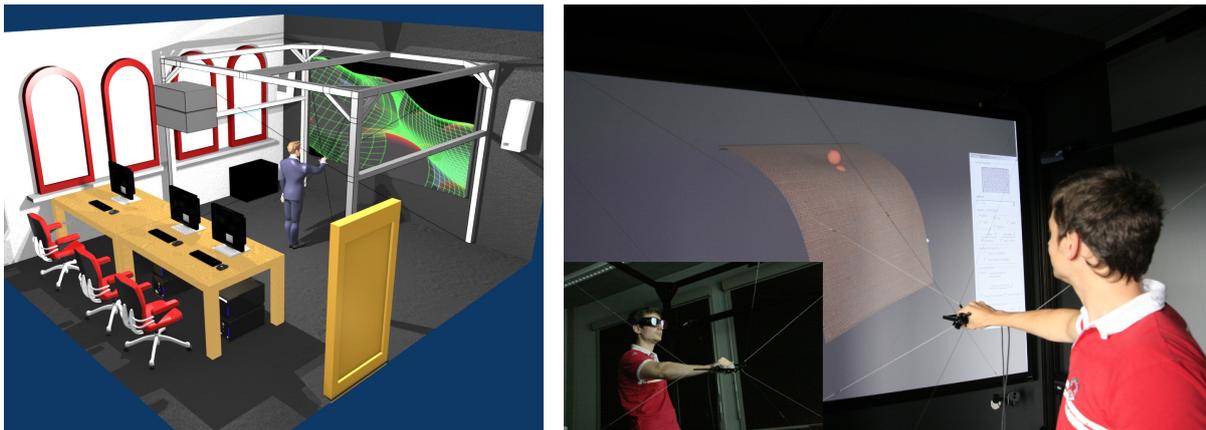
nungen unterstützen. Für die Verteilung der Rechenleistung ergeben sich unterschiedliche Anforderungen unter Berücksichtigung der Ausgabegeräte. Während zur visuellen Darstellung einer Szene Aktualisierungsraten von bis zu 25 Bildern pro Sekunde ausreichen, um einen kontinuierlichen Eindruck zu erzeugen, benötigt das haptische Feedback für ein plausibles Kraftempfinden üblicherweise Raten von bis zu 1 kHz. Damit das System den hohen Anforderungen der Kraftrückkopplung gerecht werden kann, ist es notwendig, dass die an der Berechnung beteiligten Komponenten eine möglichst geringe Latenz und hohe Bandbreite in der Datenkommunikation aufweisen.

Im Zusammenspiel von haptischer Interaktion mit visueller Darstellung unterscheiden wir in unserem Ansatz zwischen lokaler und globaler Physik. Dabei wird im Kontaktbereich zwischen Nutzer und Objekt eine lokale Simulation mit kleinen Zeitschritten durchgeführt, die die auftretenden Kräfte an den Benutzer durch das haptische Gerät vermittelt und die zugeführte Energie im Modell speichert. In der lokalen Simulation sind die Modelldaten hierbei ein lokal verfeinerter Ausschnitt des gesamten Objektes. Nebenläufig dazu wird auf globaler Simulationsebene das gesamte Objekt mit den aktuellen externen Kräften simuliert.

Auf den Zeitschritten der globalen Simulation werden innerhalb der Synchronisationsphase die Zustände beider Simulationen abgeglichen. Zusätzliche Randbedingungen werden an das lokale Modell gestellt, damit Abweichungen der Zustände beider Modelle nicht zu groß werden. In [BAGW08] haben wir diesen Ansatz der Synchronisation bereits zur Ausnutzung von Mehrkernsystemen verwendet.

3.1 Visuelle Hardware

Das VR-System nutzt für die stereoskopische Darstellung ein Projektionssystem in Full HD (1920x1080), basierend auf einer Interferenzfiltertechnik [JF03], d.h. statt linearer oder zirkulärer Polarisationsfilter werden die drei Grundfarben für das linke und rechte Auge im Farbspektrum leicht verschoben. Die Brillengläser filtern dann die überlagerten Bilder für das jeweilige Auge im entsprechenden Spektrum des Lichts. Vorteile dieser Technik sind die totale Unabhängigkeit von Blickwinkel und Kopfneigung sowie hervorragende Kanaltrennung. Zusätzlich dazu wird der Blickwinkel in der Simulation durch ein markerbasiertes Trackingsystem von der Position des Benutzers abhängig gemacht. Für das Tracking verwenden wir das IOTracker-System [PK07]. Um eine möglichst hohe Immersion zu gewährleisten,



(a) Überblick

(b) Bedienung des Systems

Abbildung 2: VR System

war es unser Ziel, die Schnittmenge zwischen dem Projektionsbereich unseres 3D Projektionssystems und dem Arbeitsbereich unseres haptischen Eingabegeräts zu maximieren.

Abb. 2(a) zeigt einen Überblick des Aufbaus.

3.2 Berechnungseinheit

Unter Berücksichtigung der Anforderungen wird für die haptische Echtzeitberechnung (*lokale Physik*) ein Tesla S1060-System von Nvidia zur Unterstützung verwendet. Es ist mit einem PCIe-16x-Interface an die Workstation gebunden und erfüllt die oben genannten Voraussetzungen. Intern besteht die Hardware aus 4 gekoppelten Grafikkarten mit je 4GB GDDR3

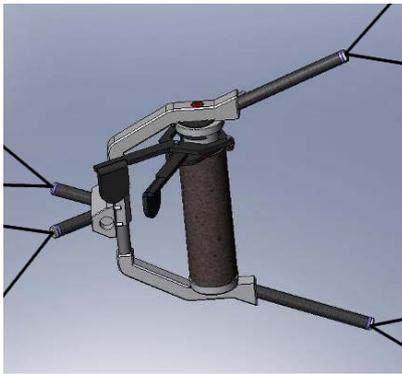
Speicher. Die auf den Grafikkarten verwendete GPU verfügt über eine massiv-parallele Architektur mit 240 Streamprozessoren, die theoretisch eine Leistung von 933GFlop/s in einfacher, beziehungsweise 78GFlop/s in doppelter Genauigkeit erreichen können.

Ergänzend wird für die visuelle Echtzeit (*globale Physik*) ein spezielles Clustersystem verwendet, welches mit dem angebundenen Hochgeschwindigkeitsnetz *InfiniBand* (DDR-4x Interface) nicht an die Leistungsdaten des GPU-Systems heran reicht, aber mit einer Transferate von $\sim 1,5\text{GB/s}$ und Latenzen von $\sim 1\mu\text{s}$ ausreichenden Spielraum für eine Skalierung der Simulation erlaubt. Insbesondere ist hier anzumerken, dass das Clustersystem im Gegensatz zu den Grafikkarten über vollständige Prozessoren verfügt. Hierdurch ergeben sich bei Rechnungen mit höherem Synchronisierungsaufwand Leistungsvorteile, die wir bei unserer bereits erwähnten globalen Physiksimulation ausschöpfen wollen.

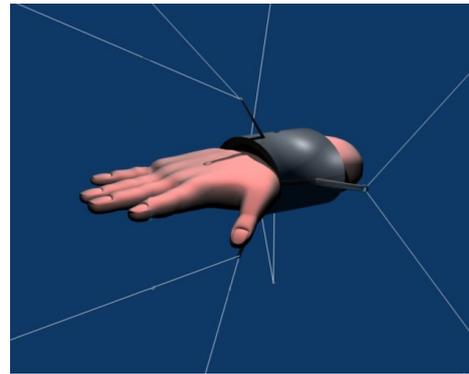
Hardware-seitig besteht das Clustersystem aus 12 Blades, die mit jeweils 2 Cell B.E.-Prozessoren bestückt sind. Den beiden Prozessoren stehen insgesamt 8 GB DDR2 Speicher zur Verfügung, die über eine NUMA-Architektur bereitgestellt werden. Intern besteht ein Prozessor aus einem PowerPC Processor Element (PPE) und acht Vektoreinheiten, den sogenannten Synergistic Processor Elements (SPEs). Die SPEs sind auf Fließkommaverarbeitung optimiert und jedes Element erreicht theoretisch bis zu 25,4 GFlop/s in einfacher Genauigkeit, respektive 12,8 GFlop/s bei doppelter Genauigkeit [IBM08]. Eine Blade kann folglich in Single-Precision eine Leistung von 406 GFlop/s erreichen, somit könnte der gesamte Cluster maximal 5,5TFlop/s erreichen. Die PPE und SPEs sind über einen vierfachen Ring-Bus verbunden. Dieser Element Interconnect Bus (EIB) erlaubt einen Datentransfer zwischen zwei Elementen mit einer maximalen Rate von 204GB/s [GEMN07]. Dadurch ist die Kommunikation zwischen den einzelnen Prozessorelementen effizient und optimal für Streamverarbeitung ausgelegt. Dennoch unterliegen die SPEs einigen Restriktionen, die spezielle Anpassungen des Codes erfordern. Beispielsweise müssen die SPEs als reine Vektorprozessoren mehrere Fließkommazahlen (128-Bit) pro Instruktion gleichzeitig verarbeiten. Ebenso verfügt eine Einheit lediglich über maximal 256kb lokalen Speicher für Programmcode und Datenhaltung. Damit das verfügbare Leistungspotential ausgenutzt werden kann, müssen diese und noch weitere Aspekte bei der Programmierung beachtet werden.

3.3 Haptische Hardware

Das INCA6D ist ein auf VR-Umgebungen speziell ausgelegtes haptisches Gerät, das sich durch einen großen Arbeitsbereich und sechs aktive Freiheitsgrade auszeichnet. Das kommerziell verfügbare System der Firma Haption basiert auf dem “scalable SPIDAR”, einem Entwurf von Sato et al. [BIS01]. Die Kraftübertragung auf den Benutzer erfolgt durch einen sogenannten *Endeffektor*. Dieser ist in ein flaschenzuggeführtes Kabelsystem eingespannt und kann gegen alternative Endeffektoren ausgetauscht werden. Die vom Endeffektor ausgehenden Kabel werden durch acht in den Ecken eines Stützrahmens befindliche Motoren gespannt. In unserem aktuellen Aufbau verwenden wir einen unseren Anforderungen entsprechend von der Firma Haption entworfenen Endeffektor (Abb. 3(a)). Er besitzt einen



(a) 7-DoF Endeffektor



(b) Manschetten Endeffektor

Abbildung 3: Wahl möglicher Endeffektoren

zusätzlichen Freiheitsgrad, der beispielsweise das Greifen von virtuellen Objekten ermöglicht. Der dem INCA zugrundeliegende Entwurf war auf die Bedienung mit zwei Händen ausgelegt, nutzte zwei mal vier Kabel und unterstützte damit nur drei Freiheitsgrade pro Hand. Das INCA6D wurde gegenüber dem SPIDAR System von Dominjon et al. [DPL07] unter anderem um Rücktriebsfedern erweitert. Diese halten die Übertragungskabel immer gespannt und vermeiden typische Probleme kabelbasierter Systeme wie Abwicklung bei deaktivierter Kraftübertragung oder Überlappungen der Kabel bei unsauberer Aufwicklung.

Einen anderen möglichen Aufbau zeigt Abb. 3(b). Unter Verwendung des Endeffektors als Manschette kann das System beispielsweise mit einem Datenhandschuh (*CyberGlove*) kombiniert werden. In diesem Fall würde das INCA zur Krauftausgabe auf das Handgelenk genutzt und ein vollständiges Kontaktmodell der Hand über den Datenhandschuh implementiert werden. Der Aufbau des INCAs ermöglicht ein schnelles, unkompliziertes Austauschen der Endeffektoren. Ein dem INCA zugehöriger PC übernimmt die Ansteuerung des Geräts, wie Messung der Kabellängen und Ausgabe von Motorspannungen. Durch ein Netzwerkinterface ist der Kontroll-PC von beliebigen Rechnern aus nutzbar. Einige technische Daten:

Maximale Krauftausgabe	37.5 N
Maximale anhaltende Krauftausgabe	12.5 N
Windungslänge der Übertragungskabel	2.5m
Auflösung der Position	0.2mm
Arbeitsbereich Translation	bis 2m
Arbeitsbereich Rotation	10° - 30°

Tabelle 1: Technische Daten

Ein auf dünnen Kabeln basierendes System ist ideal, um die Projektion so wenig wie möglich zu beeinträchtigen. Dies gewinnt eine besondere Bedeutung, wenn, wie in unserem Fall, aufgrund räumlicher Beschränkung keine Rückprojektion möglich ist. Im aktuellen Aufbau kann ein durchschnittlich großer Benutzer (1.80m) nahezu okklusionsfrei arbeiten (siehe

Abb.2(b)). Dadurch kann der große Arbeitsbereich optimal genutzt werden, um so direkt in dem (durch die Projektion) visualisierten Bereich zu arbeiten.

4 VR-System Software

Die Verbindung der unterschiedlichen Hardware innerhalb des VR-Systems stellt eine besondere Herausforderung dar. Zur Ausschöpfung des vollen Leistungspotentials muss gewährleistet sein, dass die Komponenten sehr effizient Daten austauschen können. Hierbei bildet die Workstation das zentrale Element, welches die Ansteuerung aller verfügbaren Ressourcen übernimmt. Die für das VR-System entwickelte Software bildet letztendlich die Datenquelle für die Komponenten. Folglich musste ein Framework aufgebaut werden, welches physikalische Simulationen mit visueller und haptischer Hardware verbindet. Mittlerweile gibt es verschiedene Arbeitsgruppen, die sich mit der Entwicklung von physikalischen Simulationsframeworks beschäftigen. Hierbei erwies sich das SOFA Framework [ACF⁺07] als eine flexible Plattform für unsere speziellen Anforderungen. Basierend auf dem SOFA Framework haben wir daher Module entwickelt, die für die Kommunikation mit der Hardware sorgen.

4.1 Ansteuerung des INCA6D

Die sogenannte VirtuouseAPI, die zur Ansteuerung des INCA verwendet wird, bietet einen einfachen Kontrollmodus um mit solchen Frameworks zu kommunizieren, den sogenannten Virtmech Modus. In diesem Modus wird das haptische Gerät (*Controller*) über ein Federmodell an ein virtuelles Objekt in der Simulation gekoppelt. Die Anwendung teilt dem Controller die aktuelle Position und Geschwindigkeit des virtuellen Objekts mit. Dieser berechnet dann die auf das Objekt in der Simulation wirkenden Kräfte und gibt die passenden Gegenkräfte auf den Endeffektor aus. Dabei werden optimale Integrationskonstanten bezüglich der Schrittgeschwindigkeit der Simulation gewählt. Kommt das virtuelle Objekt innerhalb der Simulation mit einem anderen Gegenstand in Kontakt, wird diese Interaktion indirekt für den Benutzer spürbar gemacht. Da die Position des Controllers innerhalb der Simulation keinen Beschränkungen unterliegt, das virtuelle Objekt aber andere Objekte der Simulation nicht durchdringen kann, führt das zu einer Spannung der Feder in Richtung der Überschneidung. Wir haben diese Methode für erste Testanwendungen eingesetzt. Dieses Modell limitiert jedoch die Einflussnahme auf die Haptik. Für komplexere Simulationen ist es daher sinnvoll, einen anderen Kontrollmodus des INCA zu verwenden, der dem Programmierer volle Kontrolle über die durch das Gerät ausgegebenen Kräfte verleiht. Lediglich die Position des Controllers wird ausgelesen, sämtliche Kräfte werden von der Simulation berechnet und direkt mit dem Gerät ausgegeben.

4.2 Physikalische Simulation

Die Basis der physikalischen Simulation bildet SOFA, das eine große Menge schon implementierter Verfahren und Algorithmen bietet. Diese können, dank des modularen Aufbaus

von SOFA, mit minimalem Aufwand verwendet und ausgetauscht werden. Um diese Flexibilität zu erreichen, kann jedes Objekt auf unterschiedliche Repräsentationen abgebildet werden (*Mapping*). Diese sogenannte Dekomposition eines physikalischen Modells (auch *Behaviour Model* genannt) führt zu der Darstellung des Behaviour Models als eine Menge an Freiheitsgraden (*DoF*). Die Freiheitsgrade beschreiben die physikalischen Eigenschaften, wie Position, Geschwindigkeit und Beschleunigung, die zur Lösung der allgemeinen Bewegungsgleichung herangezogen werden, wie in [ZC00] prinzipiell gezeigt. Ein Modell kann in einer sogenannten *Multi-Model-Representation* zudem verschiedene Repräsentationen besitzen. Diese werden über Mappings bestimmt. Ein Beispiel dafür ist das *Visual Model*, welches bei unseren Textil-Simulationen über ein *IdentityMapping* die Freiheitsgrade eins zu eins auf Punkte zur visuellen Darstellung mappt. An den Freiheitsgraden der Behaviour Models können Kräfte angreifen, somit haben sie einen Effekt auf alle Repräsentationen dieses Modells. Dafür besitzt das Behaviour Model vier generische Komponenten: Die angesprochenen *DoFs*, *ForceField*, *Solver* und *Mass*. Die inneren und äußeren Kräfte werden in den *ForceField* Komponenten berechnet. Die *Solver* Komponenten steuern den zeitlichen Ablauf, rufen in jedem Zeitschritt die Kraftberechnung auf und aktualisieren das Behaviour Model. Die *Mass* Komponenten stellen verschiedene Methoden zur Berechnung der Masse eines Objektes, z.B. durch Angabe der Dichte zur Verfügung.

Jede Szene wird dabei durch einen Szenengraph repräsentiert, der unter anderem diese vier Komponenten enthalten kann. In diesem Szenengraph ist jede Komponente an einem Knoten angebunden. Die Knoten selber bilden eine Baumstruktur, in der die Komponenten auf deren Nachbarkomponenten und deren Kinder Einfluss nehmen. Mit dieser Struktur kann der Szenengraph aus einer XML-Datei importiert oder in eine XML-Datei exportiert werden, was den einfachen Austausch der Komponenten, ohne Neukompilierung, ermöglicht. Neben den vier Komponenten des Behaviour Models kann der Szenengraph alle möglichen Konstrukte enthalten, wie die verschiedenen Repräsentationen der Objekte, Algorithmen zur Topologieberechnung oder auch selbst entwickelte Werkzeuge.

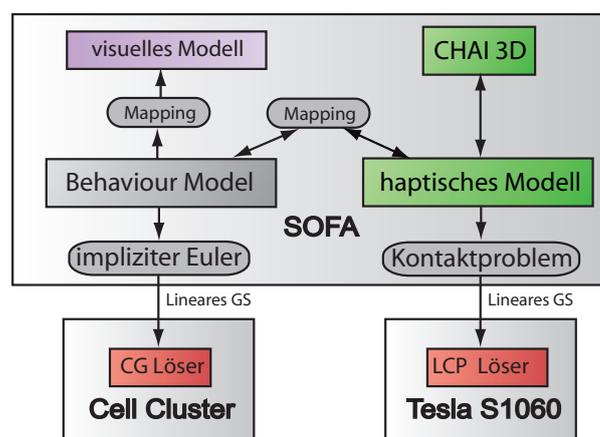


Abbildung 4: Haptisches Mapping

Für unser VR-System ist diese Architektur optimal. Die einzelnen Komponenten kön-

nen unabhängig voneinander entwickelt und problemlos zusammen verwendet werden. Des Weiteren hilft die Multi-Model-Representation bei der klaren Separation der haptischen und visuellen Berechnung. Durch die CHAI3D [CMBS] Bibliothek werden die haptischen Geräte für SOFA zugänglich gemacht. CHAI3D ist ein quelloffenes Haptik Framework, das bereits eine große Anzahl haptischer Geräte unterstützt und beliebig erweiterbar ist. Das unter Verwendung von CHAI3D erzeugte Device Objekt stellt die Position des haptischen Geräts zur Verfügung, nimmt Kräfte entgegen und wird dann über ein beliebig konfigurierbares Mapping mit der Simulation verbunden. Andere Aspekte der Repräsentation innerhalb der Simulation, wie Kollisionsmodell oder visuelle Darstellung, sind dadurch beliebig austauschbar. Durch die Szenengraphen Architektur können bereits bestehende Simulationen mit einer haptischen Komponente erweitert werden. Wie in Abb. 4 gezeigt, wird über ein Mapping ein eigenes haptisches Modell erstellt. Dieses wird auf einen lokalen Bereich begrenzt, um den nötigen Rechenaufwand zu verringern. Die Berechnung der Kräfte in der Kontaktsituation läuft auf ein quadratisches Optimierungsproblem hinaus, das sogenannte “linear complementary problem“ (LCP). Durch Überführung in ein lineares Gleichungssystem wird schließlich das Problem mit dem Gauß-Seidel-Algorithmus gelöst (vgl. [DDKA06]). Dieser Algorithmus soll dann, wie in [CA09] verwendet, als GPU-basierte Implementation auf der Tesla S1060 ausgeführt werden.

5 Erste Erfahrungen und Ergebnisse

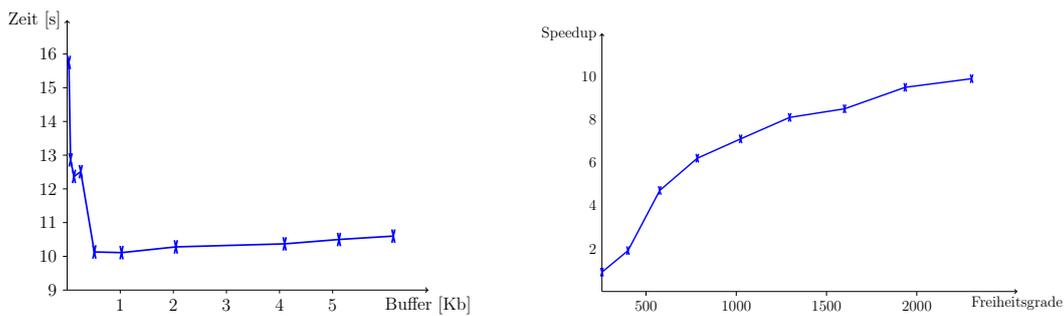
Wir haben zunächst mit der Einbettung unserer Textilsimulation, welche sich auf die Berechnung von [VMT05] stützt, in das SOFA Framework begonnen. Vereinfacht gesagt wird hierbei in der Simulation durch das implizite Eulerverfahren ein lineares Gleichungssystem aufgebaut, das über das präkonditionierte konjugierte Gradientenverfahren (PCG) gelöst wird. Dafür wurde die Kraftberechnung beibehalten und das Lösungsverfahren auf den Cluster ausgelagert. Aus unserem mechanischen Modell der Textilie ergab sich eine schwach besetzte Matrix, mit der die zu übertragenden Datenmengen gering gehalten werden konnten. Die Daten werden über die einzelnen Blades gleichmäßig auf die SPEs verteilt. Da die SPEs nicht über genügend lokalen Speicher verfügen, um alle Vektoren- und Matrixdaten zu speichern, mussten wir die Daten blockweise abrufen. Zur Minimierung von Wartezeiten bei der Berechnung wurde ein Double-Buffering-Ansatz implementiert. Dabei stellten wir fest, dass die Transferzeiten einen starken Einfluss auf die Effizienz des Verfahrens haben. Einer unserer ersten Optimierungsansätze war deswegen, die Transferzeiten bei festen Kommunikationsparametern zu minimieren. Bei Berechnungen mit fester Problemgröße n , die eine lineare Laufzeit aufweisen, konnten wir den folgenden Zusammenhang zwischen der Wartezeit t_{wait} und der Buffergröße b feststellen.

$$t_{wait} = \frac{n}{r} + l + \frac{nl}{b} - zn + zb \quad \text{für} \quad bz < \frac{b}{r} + l$$

Wobei n die Anzahl der Freiheitsgrade angibt (in unserem Fall die Dreieckspunkte im Textil), r ist die Transferrate, l ist die Latenzzeit, die beim Anfordern der Daten auftritt und

$zn \in \mathcal{O}(n)$ beschreibt die Zeit, die für die Berechnung benötigt wird. Diese Formel zeigt, dass bei Berechnungen, die eine lineare Laufzeit aufweisen, durch ein größeres n kein Buffering mit weniger Wartezeit möglich wäre. Jedoch gibt es ein Minimum bei $b = \sqrt{\frac{nl}{z}}$. Die Auswirkungen dieser Wartezeiten konnten wir in unserer portierten Textilsimulation für verschiedene Textilien auch sehr gut sichtbar machen. Dafür haben wir die Buffergrößen variiert und so den in Abb. 5(a) für $n = 400$ gezeigten Verlauf zwischen der Zeit, die für das wiederholte Lösen des linearen Gleichungssystems zur Anwendung des impliziten Eulerverfahrens in 300 Zeitschritten benötigt wurde, und der Größe des Buffers erhalten.

Neben dem Buffering konnten wir noch an einigen Stellen spezielle Optimierungen für die Cell-Architektur anwenden und erhielten dadurch ein Verfahren, das auf dem Cluster einen deutlichen Performancegewinn gegenüber der ursprünglichen Implementation brachte. Dazu haben wir den Speedup unserer Implementation der Textilsimulation über 200 Zeitschritte gemessen. Wir haben die sequentielle Variante, auf die sich der Speedup bezieht, auf einem aktuellen Highend-PC (Nehalem Xeon 3.2GHz) mit der Clustervariante auf 16 SPEs, unter gleichen Simulationsbedingungen, verglichen. Beim Vergleich in Abb. 5(b) wurden außerdem auch die Transferzeiten in der Clustervariante mit einbezogen.



(a) Zeit für das Lösen mit Double-Buffering

(b) Speedup bei optimaler Buffergröße

Abbildung 5: Erste Ergebnisse auf dem Cluster

Wir konnten bisher bei 16 SPEs nur einen Speedup von 10 erreichen. Dies ist hauptsächlich auf die Kommunikationsbibliothek zurückzuführen, die nur einen Bruchteil der verfügbaren Bandbreite von Infiniband nutzen kann. Wir erhoffen uns, durch die Umstellung von der architekturenspezifischen Bibliothek auf generische MPI-Kommunikation die volle Bandbreite auszunutzen zu können.

6 Fazit und Ausblick

In diesem Beitrag haben wir ein neuartiges VR-System mit multimodaler Interaktion vorgestellt, welches aktuelle Computer-Technologien in einem physikalischen Modell koppelt, um die Rechenleistung zu erhöhen. Dieser hybride Ansatz erlaubt eine deutliche Erhöhung der Freiheitsgrade sowohl im physikalischen Modell als auch im Kontaktproblem. Die ersten Ergebnisse unserer Forschung haben uns in diesem Ansatz bestätigt. Durch diese Möglichkeit

und dem für Greifvorgänge angepassten Endeffektor hoffen wir, den Realismus und die Intuitivität in der VR-Anwendung verbessern zu können. Anhand einiger Beispiele haben wir gezeigt, wie hierbei die Komponenten zusammenarbeiten und wie das System flexibel an die jeweiligen Anforderungen anpassbar ist. Eine besondere Schwierigkeit stellt in diesem Ansatz die Kopplung der heterogenen Rechenhardware dar, deren Zusammenspiel erst die komplexe Simulation in haptischer Echtzeit ermöglicht. Der Entwicklungs- und Optimierungsaufwand erhöht sich somit signifikant, da die Architekturen unterschiedliche Anforderungen stellen. Wir denken jedoch, dass sich dieser Aufwand mit der allgemeinen Verfügbarkeit der Open Computing Language [Mun08] auf dem Cell Prozessor und den GPUs wieder reduzieren wird, da hier die Beschreibung der Probleme losgelöst von der Hardwareebene stattfindet. Folglich verfügt das VR-System über das Potential, bestehende Systeme bezüglich der Interaktionsfähigkeit auf Hard- und Softwareebene signifikant zu verbessern. Hybride Algorithmen zur besseren Darstellung der Kontaktsituation und die Flexibilität bei der Wahl neuer Endeffektoren bieten interessante Ausgangspunkte für weitere Untersuchungen.

Literatur

- [ACF⁺06] ALLARD, J., S. COTIN, F. FAURE, P.J. BENSOUSSAN, F. POYER, C. DURIEZ, H. DELINGETTE und L. GRISONI: *Sofa-an open source framework for medical simulation*. *Studies in health technology and informatics*, 125:13, 2006.
- [ACF⁺07] ALLARD, JÉRÉMIE, STÉPHANE COTIN, FRANÇOIS FAURE, PIERRE-JEAN BENSOUSSAN, FRANÇOIS POYER, CHRISTIAN DURIEZ, HERVÉ DELINGETTE und LAURENT GRISONI: *SOFA an Open Source Framework for Medical Simulation*. In: *Medicine Meets Virtual Reality (MMVR'15)*, Long Beach, USA, February 2007.
- [AZBF08] ALTOMONTE, M., D. ZERBATO, D. BOTTURI und P. FIORINI: *Simulation of deformable environment with haptic feedback on GPU*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, Seiten 3959–3964, 2008.
- [BAGW08] BÖTTCHER, G., D. ALLERKAMP, D. GLÖCKNER und F.E. WOLTER: *Haptic two-finger contact with textiles*. *The Visual Computer*, 24(10):911–922, 2008.
- [BIS01] BUOGUILA, LAROUCSI, MASAHIRO ISHII und MAKOTO SATO: *Scaleable SPIDAR: a haptic interface for human-scale virtual environments*. In: *Haptic Human-Computer Interaction*, Band 2058 der Reihe *Lecture Notes in Computer Science*, Seiten 182–193. Springer Berlin / Heidelberg, 2001.
- [CA09] COURTECUISSE, H. und J. ALLARD: *Parallel Dense Gauss-Seidel Algorithm on Many-Core Processors*. 2009.

- [CMBS] CONTI, F., D. MORRIS, F. BARBAGLI und C. SEWELL: *CHAI 3D*. Online: <http://www.chai3d.org>.
- [DDKA06] DURIEZ, C., F. DUBOIS, A. KHEDDAR und C. ANDRIOT: *Realistic haptic rendering of interacting deformable objects in virtual environments*. IEEE Transactions on Visualization and Computer Graphics, 12(1):36–47, 2006.
- [DPL07] DOMINJON, LIONEL, JÉRÔME PERRET und ANATOLE LÉCUYER: *Novel devices and interaction techniques for human-scale haptics*. The Visual Computer, 23(4):257–266, April 2007.
- [GASM08] GOSSELIN, FLORIAN, CLAUDE ANDRIOT, JOAN SAVALL und JAVIER MARTÍN: *Large Workspace Haptic Devices for Human-Scale Interaction: A Survey*. In: *Haptics: Perception, Devices and Scenarios*, Band 5024 der Reihe *Lecture Notes in Computer Science*, Seiten 523–528. Springer Berlin / Heidelberg, 2008.
- [GEMN07] GSCHWIND, M., D. ERB, S. MANNING und M. NUTTER: *An open source environment for cell broadband engine system software*. Computer, 40(6):37–47, 2007.
- [IBM08] IBM: *PowerXCell8i processor product brief*. http://www-03.ibm.com/technology/resources/technology_cell_pdf_PowerXCell_PB_7May2008_pub.pdf, 2008.
- [JF03] JORKE, H. und M. FRITZ: *INFITEC-A new Stereoscopic Visualisation Tool by Wavelength Multiplex Imaging*. Electronic Displays (September), 2003.
- [JHS05] JEONG, S., N. HASHIMOTO und M. SATO: *Immersive Multi-Projector Display on Hybrid Screens with Human-Scale Haptic Interface*. IEICE TRANSACTIONS on Information and Systems, Seiten 888–893, 2005.
- [Mun08] MUNSHI, A.: *The OpenCL Specification*. Documentation, december, 2008.
- [PK07] PINTARIC, T. und H. KAUFMANN: *Affordable infrared-optical pose-tracking for virtual and augmented reality*. In: *Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop, IEEE VR*, Seiten 44–51, 2007.
- [VMT05] VOLINO, P. und N. MAGNENAT-THALMANN: *Accurate garment prototyping and simulation*. Computer-Aided Design & Applications, 2(1-4), 2005.
- [YS08] YOUQUAN, LIU und DE SUVRANU: *CUDA-based Real Time Surgery Simulation*. Medicine Meets Virtual Reality 16: Parallel, Combinatorial, Convergent: NextMed by Design, Seite 260, 2008.
- [ZC00] ZHUANG, Y. und J. CANNY: *Haptic interaction with global deformations*. In: *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00*, Band 3, 2000.