

Haptisches Feedback eines Roboters durch virtuelle 3D-Modelle

Carl Gathmann
Universität zu Lübeck

Lübeck, Germany
carl.gathmann@student.uni-luebeck.de

Marten Buchmann
Universität zu Lübeck

Lübeck, Germany
marten.buchmann@student.uni-luebeck.de

Abstract—!!!NOCH GPT!!! In dieser Studie wird ein Ansatz zur Generierung von haptischem Feedback in Mensch-Roboter-Interaktionen vorgestellt. Mittels maßgeschneiderter Software erzeugen wir eine sensorische Wahrnehmung, die durch frei gestaltbare, virtuelle 3D-Modelle gesteuert wird. Der Anwender erfährt eine Art abweisende Kraft, die durch die räumlichen Grenzen des virtuellen Modells definiert ist, wodurch eine physische Interaktion mit dem immateriellen Modell simuliert wird. Die vorgestellte Technologie findet breite Anwendungsbereiche, von medizinischen Simulationen und Exploration in gefährlichen Zonen bis hin zur Erhöhung der Spielerfahrung in virtuellen Umgebungen. In Verbindung mit Virtual-Reality-Ausrüstung eröffnet unsere Methode neue Wege zur Verbesserung der Benutzerimmersion durch die Vermittlung eines realistischeren Gefühls für die Form und Textur virtueller Objekte. Die in dieser Arbeit vorgestellten Prinzipien und Implementierungen können als Grundlage für weiterführende Forschungen und Entwicklungen auf diesem aufstrebenden Gebiet dienen.

Index Terms—component, formatting, style, styling, insert

I. EINLEITUNG

!!! GPT =_i

Die Mensch-Roboter-Interaktion (MRI) hat in den letzten Jahren zunehmend an Bedeutung gewonnen und sich als ein dynamisches Forschungs- und Anwendungsfeld etabliert. Im Zentrum dieser Interaktion steht die Verbesserung der Benutzererfahrung durch die Erweiterung der sinnlichen Wahrnehmung des Menschen.

In dieser Studie präsentieren wir einen Ansatz, der es dem Benutzer ermöglicht, virtuelle 3D-Modelle zu "ertasten", indem er über ein Roboterinterface mit ihnen interagiert. Dies wird durch die Erzeugung einer abweisenden Kraft erreicht, die auf den Grenzen der virtuellen Modelle basiert. Dieses haptische Feedback simuliert das physische Berühren eines realen Objekts, obwohl kein tatsächlicher physischer Kontakt mit dem virtuellen Modell besteht.

Dieser Ansatz bietet zahlreiche Anwendungsmöglichkeiten, darunter die Simulation von Operationen für Ausbildungszwecke, die Erkundung von Objekten in gefährlichen oder unzugänglichen Umgebungen und die Erhöhung der Immersion in virtuellen Spielen. Darüber hinaus kann die Kombination unserer Technologie mit Virtual-Reality-Brillen zu einem verbesserten Gefühl von Präsenz und Realismus in virtuellen Umgebungen führen.

Das vorliegende Paper beleuchtet die zugrundeliegenden Prinzipien, technische Details und potenzielle Anwendungen

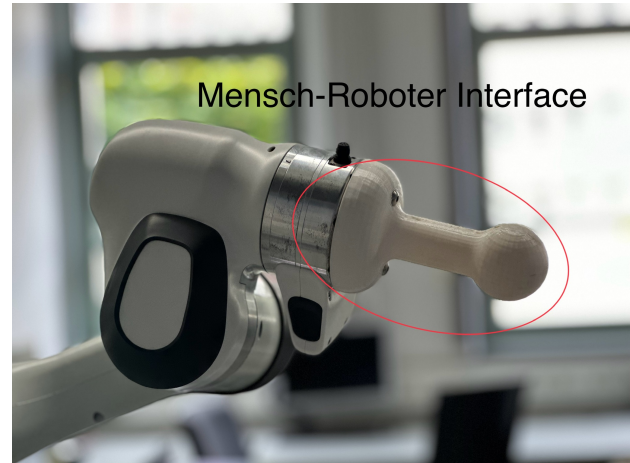


Fig. 1. Mensch-Roboter-Schnittstelle

dieses Ansatzes. Es leistet einen wertvollen Beitrag zur Erforschung und Weiterentwicklung von Technologien für haptisches Feedback in der Mensch-Roboter-Interaktion.

= GPT !!!

II. METHODS

Die Erzeugung des haptischen Feedbacks beruht auf der Berechnung einer abweisenden Kraft, die so konzipiert ist, dass dem Benutzer ein realistisches Gefühl für die Form des virtuellen 3D-Modells vermittelt wird. Die Modelle können als Datei im STL-Standard übergeben werden. Der STL-Standard speichert 3D-Modelle als Dreiecksmesh, das aus Vertices (Eckpunkten) und den Normalen der Dreiecke besteht. Es ist keine besondere CAD Software nötig um 3D-Modelle im STL Format zu erstellen. Außerdem ist das Format weit verbreitet und wird in der Industrie zur Speicherung und Übertragung von 3D-Modellen eingesetzt. Das sorgt für hohe Flexibilität und einfache Handhabung. !!!Quelle zu STL Standard!!!

Für die physische Implementierung der Mensch-Roboter-Interaktion wurde der Panda-Roboter von Franka Emika ausgewählt. Der Panda zeichnet sich durch seine 7-achsige Struktur aus, die hohe Präzision und Flexibilität bietet. Eine benutzerfreundliche, 3D-gedruckte Schnittstelle (1) wurde am Endeffektor des Roboters installiert, um die Führung durch menschliche Benutzer zu erleichtern.

Bei der Entwicklung der Software für dieses System stand die Leistungsfähigkeit im Vordergrund, um eine nahtlose Benutzererfahrung zu gewährleisten. Dazu war es entscheidend, dass die Anwendung in Echtzeit ausgeführt werden kann, wobei eine maximale Zeitverzögerung von weniger als 1 Millisekunde akzeptiert wurde. Die Programmiersprache C++ wurde aufgrund ihrer hohen Performance ausgewählt, um diesem Anspruch gerecht zu werden. Die Hardware-Konfiguration bestand aus !!!HARDWARE!!!. Die Verarbeitung und Nutzung der 3D-Modelle wurden durch die Umwandlung der STL-Daten in Eigen-Matrizen optimiert. Für diesen Prozess wurde durch einen STL-Parser benutzt. Dadurch ist eine einfache und effiziente Handhabung der 3D-Daten innerhalb des Programms möglich. Insgesamt zeichnet sich unsere Methodik durch die Kombination von Standard-technologien und innovativen Ansätzen aus, um eine effiziente und benutzerfreundliche Lösung für haptisches Feedback in der Mensch-Roboter-Interaktion zu schaffen.

Während der Recherche des Projekts wurden verschiedene Ansätze zur Erzeugung von haptischem Feedback in der Mensch-Roboter-Interaktion untersucht, welche in den folgenden Abschnitten vorgestellt werden.

Als Vorbereitung auf die Verwendung der Ansätze muss zuerst ein zweidimensionales Problem aus dem dreidimensionalen Problem gemacht werden. Erreicht wird dies durch die Erstellung eines neuen Koordinatensystems für jedes Dreieck, welches sich ergibt durch:

- 1) Eine Kante des Dreiecks als x-Achse
- 2) Das Kreuzprodukt der entstandenen Achse mit einem weiteren Vektor als z-Achse
- 3) Das Kreuzprodukt der beiden entstandenen Achsen als y-Achse

Die Koordinaten des Punktes werden dann in diesem Koordinatensystem angegeben. Auf diese Art repräsentieren die x- und y-Koordinaten die Position und die z-Koordinate die Entfernung des Punktes von der Ebene des Dreiecks.

A. Edge Funktion

Die Edge-Funktion ist eine häufig verwendete Funktion im Bereich der Computer-Grafik. Sie wird verwendet, um herauszufinden, wo sich ein Punkt relativ zu einem Dreieck befindet. Definiert ist sie wie folgt:

$$E_i = (x_{i+1} - x_i)(y - y_i) - (y_{i+1} - y_i)(x - x_i) \quad (1)$$

x, y sind die Koordinaten des Punktes, x_i, y_i sind die Koordinaten des i-ten Eckpunktes des Dreiecks. Im Fall eines Dreiecks mit den Eckpunkten A, B und C und einem Punkt P ist der Punkt innerhalb des Dreiecks, wenn gilt:

$$E_{AB} \geq 0 \wedge E_{BC} \geq 0 \wedge E_{CA} \geq 0 \quad (2)$$

!!!BILD!!!

Mit dem Wissen, ob ein Punkt innerhalb des Dreiecks liegt und der Höhe des Punktes, die über die z-Koordinate des transformierten Punktes gegeben ist, kann ein Kraftvektor berechnet werden. Dieser Kraftvektor zeigt in Richtung der

Normalen des Dreiecks und ist proportional zur Höhe des Punktes. Die Kraft kann dann wie folgt berechnet werden:

$$F = \frac{1}{d} \cdot n \quad (3)$$

d ist die Entfernung des Punktes von der Ebene des Dreiecks und n ist die Normale des Dreiecks. Die großen Vorteile dieser Methode sind die Einfachheit und die Geschwindigkeit. Die Berechnung der Edge-Funktion ist sehr einfach und schnell, da sie nur 2 Multiplikationen und 4 Subtraktionen benötigt. Ein Problem, was sich daraus ergibt, ist die Unstetigkeit entlang der Kanten benachbarter Dreiecke. Dies kann zu unintuitivem und sprunghaftem Verhalten führen, wenn der Benutzer sich entlang der Kante bewegt. Aus diesem Grund haben wir die Methode verworfen und uns an einer weiteren Methode versucht.

B. Baryzentrische Koordinaten

Eine weitere Methode, die wir untersucht haben, ist die Verwendung von baryzentrischen Koordinaten. Diese Methode ist sehr ähnlich zur Edge-Funktion, da sie auch verwendet wird, um herauszufinden, wo sich ein Punkt relativ zu einem Dreieck befindet. Definiert ist sie wie folgt:

$$\lambda_1 = \frac{(y_B - y_C)(x_P - x_C) + (x_C - x_B)(y_P - y_C)}{(y_B - y_C)(x_A - x_C) + (x_C - x_B)(y_A - y_C)}$$

$$\lambda_2 = \frac{(y_C - y_A)(x_P - x_C) + (x_A - x_C)(y_P - y_C)}{(y_B - y_C)(x_A - x_C) + (x_C - x_B)(y_A - y_C)} \quad (4)$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2$$

A, B, C sind hierbei die Eckpunkte des Dreiecks und P die in das Koordinatensystem des Dreiecks transformierte Roboter Position. λ_1, λ_2 und λ_3 sind die baryzentrischen Koordinaten des Punktes P . Diese sagen aus wie weit der Punkt von den jeweiligen Eckpunkten entfernt ist. Wenn alle drei Koordinaten positiv sind, liegt der Punkt innerhalb des Dreiecks. Nun lässt sich ein Schwellwert festlegen, ab dem eine Interpolation zwischen den Kanten/Eckpunkten des Dreiecks stattfindet. Dazu ist es notwendig, für jede Kante zu wissen, welche Dreiecke sie enthält und welche Normalen diese Dreiecke haben. Das gleiche gilt für die Eckpunkte. Anschließend kann die Kraft wie folgt berechnet werden:

$$F_{ab} = \begin{cases} \lambda_1 \frac{F_{AB1} + F_{AB2} + \dots + F_{ABn}}{n} & 0 < \lambda_1 < \text{Schwellwert,} \\ 0 & \text{sonst.} \end{cases}$$

$$F_{bc} = \begin{cases} \lambda_2 \frac{F_{AB1} + F_{AB2} + \dots + F_{ABn}}{n} & 0 < \lambda_2 < \text{Schwellwert,} \\ 0 & \text{sonst.} \end{cases}$$

$$F_{ca} = \begin{cases} \lambda_3 \frac{F_{AB1} + F_{AB2} + \dots + F_{ABn}}{n} & 0 < \lambda_3 < \text{Schwellwert,} \\ 0 & \text{sonst.} \end{cases}$$

F_1, F_2, \dots, F_n sind die Normalen der Dreiecke, die die jeweilige Kante enthalten. F_{ab} ist die Kraft, die auf den Benutzer wirkt, wenn er sich auf der Kante zwischen den Punkten A und B befindet. F_{bc} und F_{ca} sind analog dazu.

Für die Eckpunkte gilt der gleiche Ansatz nur sind hier zwei Bayzentrische Koordinaten gleichzeitig zwischen 0 und Schwellwert. Entsprechend wird die Kraft wie folgt berechnet:

$$F_a = \begin{cases} \frac{(\lambda_1 + \lambda_2)}{2} \frac{F_{A1} + F_{A2} + \dots + F_{An}}{n} & 0 < \lambda_1, \lambda_2 < \text{Schwellwert} \\ 0 & \text{sonst.} \end{cases}$$

$$F_b = \begin{cases} \frac{(\lambda_2 + \lambda_3)}{2} \frac{F_{A1} + F_{A2} + \dots + F_{An}}{n} & 0 < \lambda_2, \lambda_3 < \text{Schwellwert} \\ 0 & \text{sonst.} \end{cases}$$

$$F_c = \begin{cases} \frac{(\lambda_3 + \lambda_1)}{2} \frac{F_{A1} + F_{A2} + \dots + F_{An}}{n} & 0 < \lambda_3, \lambda_1 < \text{Schwellwert} \\ 0 & \text{sonst.} \end{cases}$$

Der Vorteil gegenüber der Edge-Funktion ist, dass so die Unstetigkeit entlang der Kanten vermieden wird, die Berechnung der baryzentrischen Koordinaten ist jedoch etwas komplexer und langsamer als die Berechnung der Edge-Funktion. Daher wurde auch diese Variante letztendlich nicht verwendet.

C. Point-Triangle-Distance-Funktion

III. RESULTS

IV. DISCUSSION

V. CONCLUSION