DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Pepper VR – Teleopperation eines humanuiden Roboter auf Basis der Analyse menschlicher Bewegung

Studienarbeit

für die Prüfung zum

Bachelor of Science

des Studienganges Informatik / Angewandte Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Matthias Schuhmacher und Marlene Rieder

Abgabedatum 20. Mai 2024

Bearbeitungszeitraum	300 Stunden
Matrikelnummer	4128647 und 8261867
Kurs	tinf21b3 und tinf21b5
Gutachter der Studienakademie	Prof. Dr. Marcus Strand

Erklärung

Ich versichere hiermit, dass ich meine Studienarbeitmit dem Thema: »Pepper VR – Teleopperation eines humanuiden Roboter auf Basis der Analyse menschlicher Bewegung« selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort Datum

Unterschrift

Zusammenfassung

Das Ziel der vorliegenden Studienarbeit ist es, eine Verbindung zwischen einer Virtual-Reality Brille und dem humanoiden Roboter Pepper herzustellen. Das Kamerabild des Roboters soll auf der Brille angezeigt werden, ebenfalls soll es möglich sein, den Roboter mit Hilfe der Controller der Brille zu steuern.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Kapitel 2

Grundlagen

2.1 Pepper

Pepper ist ein humanoider Roboter, der entwickelt wurde, um die Gefühle und Gesten von Menschen zu analysieren und basierend auf diesen, darauf zu reagieren. Das Projekt entstand durch eine Zusammenarbeit des französischen Unternehmens Aldebaran Robotics **SAS!** (**SAS!**) und des japanischen Telekommunikations- und Medienkonzerns SoftBank Mobile Corp. Ziel diese Projektes war es, einen humanuiden "Roboter-Gefährten" oder einen "persönlichen Roboter-Freund" zu schaffen, der zunächst im Gewerbesektor in Verkaufsräumen, an Empfangstischen oder in Bildungs- und Gesundheitseinrichtungen eingesetzt werden sollte. Die Produktion wurde jedoch aufgrund geringer Nachfrage bis auf Weiteres pausiert.

Das Konzept von Pepper distanziert sich von herkömmlichen Industrierobotern und reinen Spielzeugrobotern, indem er als informativer und kommunikativer Begleiter konzipiert wurde. Sein Aussehen, das im etwa an die Größe eines Kindes angelehnt ist, sowie ein freundliches Gesicht und eine kindliche Stimme sind im ästhetischen Konzept von "kawaii" (japanisch für "niedlich" oder auch "liebenswert") gehalten.

Pepper wurde im Rahmen einer Präsentation am 5. Juni 2014 als der "erste persönliche Roboter der Welt mit Emotionen" vorgestellt. Die Vermarktung begann damit, dass SoftBank Pepper-Geräte in ihren Verkaufsräumen einsetzte, um Kunden zu unterhalten und zu informieren. Die Roboter sollte dabei den Umgang mit Kunden erlernen, um zukünftige Anwendungsmöglichkeiten zu erforschen. Verkauft wurde offiziell ab dem 3. Juli 2015 zu einem Preis von 198.000 Yen pro Einheit, zuzüglich monatlicher Gebühren für Zusatzleistungen. Im Laufe der Zeit wurde Pepper auch für den Einsatz in weiteren Unternehmen und Einrichtungen verfügbar gemacht.

Pepper wird mit einer Grundausstattung an Anwendungen geliefert, jedoch sind für spezifische Anwendungen, individuell entwickelte Softwarelösungen erforderlich wie auch zum Beispiel in diesem. SoftBank ermöglichte unabhängigen Entwicklern durch die Veröffentlichung der Schnittstellen den Zugang zu einem Interface für Applikationsprogramme, um zusätzliche Anwendungen für Pepper zu erstellen. Das NAOqi-Framework welches für diesen Nutzen bereitgestellt wurde, beinhaltet eine **API!** (**API!**), eine **SDK!** (**SDK!**) und

weitere Tools, welche in den Sprachen Python und C++ uneingeschränkten Zugriff auf die Komponenten, Sensoren und Aktoren des Roboters bieten, dazu später Ausführlicheres in Abschnitt 2.2. Mit Hilfe diese Interfaces haben verschiedene Unternehmen integrierte Lösungen entwickelt, die Pepper beispielsweise bei der Kundenberatung unterstützen können.

Das Design von Pepper ist dem Menschen ähnlich und umfasst einen Kopf mit integrierten Mikrofonen und Kameras sowie einen Torso mit weiteren Sensoren für Stabilität und Sicherheit. Der Roboter verfügt über verschiedene Mechaniken, die es ihm ermöglichen, sich flüssig zu bewegen und mit Personen zu interagieren. Durch die Verwendung von Kameras und bereitgestellter Software ist Pepper in der Lage, Emotionen bei seinen Gesprächspartnern zu erkennen und darauf zu reagieren, obwohl er selbst keine Mimik besitzt. Sicherheitsvorkehrungen wie Abstandssensoren und Stabilisatoren gewährleisten einen sicheren Einsatz von Pepper in verschiedenen Umgebungen. Diese können jedoch bedingt durch den Entwickler deaktiviert werden, um den Roboter in komplexeren oder laborähnlichen Umgebungen zu betreiben.

2.2 NAOqi

2.2.1 Definition

NAOqi ist die Bezeichnung für die Hauptsoftware, die auf dem Pepper Roboter ausgeführt wird und ihn intern steuert. Das NAOqi Framework ist das Programmiergerüst, welches zur Programmierung von NAO und Pepper Robotern verwendet wird. Es implementiert alle allgemeinen Anforderungen der Robotik, einschließlich: Parallelität, Ressourcen-Management, Synchronisation und Ereignisse. Dieses Framework ermöglicht eine homogene Kommunikation zwischen verschiedenen Modulen wie etwa die Bewegung, Audio oder Video sowie eine homogene Programmierung und einen homogenen Informationsaustausch. Das Framework ist:

- plattformübergreifend, d.h. es ist möglich, damit auf Windows, Linux oder Mac zu entwickeln. Genaueres dazu im Abschnitt 2.2.1.
- sprachübergreifend, mit einer identischen API für C++ und Python. Weitere Details dazu sind in Abschnitt 2.2.1 aufgeführt.
- bereit für Introspektion, was bedeutet, dass das Framework weiß, welche Funktionen in den verschiedenen Modulen verfügbar sind und wo. Für Details diesbezüglich siehe Abschnitt 2.2.1.

Sprachübergreifend

Software kann in C++ und Python entwickelt werden. Eine Übersicht über die Sprachen selbst in den Abschnitten Unterabschnitt 2.4.1 und Unterabschnitt 2.4.2. In allen Fällen sind die Programmiermethoden genau die gleichen, alle vorhandenen **API!**s können unabhängig von den unterstützten Sprachen aufgerufen werden:

- Wird ein neues C++-Modul erstellt, können die C++-**API!**-Funktionen von überall aus aufgerufen werden,
- Sind sie richtig definiert, können auch die **API!**-Funktionen eines Python-Moduls von überall aus aufgerufen werden.

In der Regel werden die Verhaltensweisen in Python und Ihre Dienste in C++ entwickelt.

Introspektion

Die Introspektion ist die Grundlage der Roboter-**API!**, der Fähigkeiten, der Überwachung und der Maßnahmen bei überwachten Funktionen. Der Roboter selbst kennt alle verfügbaren **API!**-Funktionen. Wird eine Bibliothek entladen, werden die entsprechenden **API!**-Funktionen automatisch ebenfalls entfernt. Eine in einem Modul definierte Funktion kann der **API!** mit einem `BIND_METHOD` hinzugefügt werden.

Wird eine Funktion gebunden, werder automatisch folgende Funktionen ausgeführt:

- Funktionsaufruf in C++ und Python, wie in Abschnitt 2.2.1 beschrieben
- Erkennen der Funktion, wenn sie gerade ausgeführt wird
- Funktion lokal oder aus der Ferne, z.B. von einem Computer oder einem anderen Roboter, ausführen weiter im Detail beschrieben in Abschnitt 2.2.1
- Generierung und Aufruf von `wait`, `stop`, `isRunning` in Funktionen

Die **API!** wird im Webbrowser angezeigt wenn auf das Gerät per **URL!** (**URL!**) oder **IP!** (**IP!**)-Addresse auf dem Port 9559 zugegriffen wird. In dieser Übersicht, zeigt der Roboter seine Modulliste, Methodenliste, Methodenparameter, Beschreibungen und Beispiele an. Der Browser zeigt auch parallele Methoden an, die überwacht, zum Warten veranlasst und gestoppt werden können.

Die Introspektion und derer Implementation im NAOqi-Framework, ist also ein mächtiges Werkzeug, welches es ermöglicht, die Roboter-**API!** zu verstehen und zu verwenden aber auch zu überwachenund zu steuern.

Verteilter Baum und Kommunikation

Eine Echtzeitanwendung kann aus einer einzelnen ausführbaren Datei oder einem Baum von mehreren Systemen wie etwa Robotern, Prozessen oder Modulen bestehen. Unabhängig davon sind die Aufrufmethoden immer dieselben. Eine ausführbare Datei kann durch eine Verbindung mit einem anderen Roboter mit **IP!**-Adresse und Port verbunden werden, sodass alle **API!**-Methoden von anderen ausführbaren Dateien sind auf die gleiche Weise verfügbar sind, genau wie bei einer lokalen Methode. NAOqi trifft dabei selbst die Wahl zwischen schnellem Direktaufruf **LPC!** (**LPC!**) und Fernaufruf **RPC!** (**RPC!**).

Unterstützte Betriebssysteme

2.2.2 NAOqi Vorgehensweise

Die NAOqi Software, welche auf dem Roboter läuft, ist ein Broker. Wenn dieser startet, lädt er eine Voreinstellungsdatei in den Speicher, in der festgelegt ist, welche Bibliotheken in dieser Konfiguration geladen werden sollen. Jede Bibliothek enthält ein oder mehrere Module, die den Broker benutzen, um ihre Methoden bereitzustellen.

Der Broker selbst bietet Nachschlagdienste an, so dass jedes Modul im Baum oder im Netzwerk jede Methode finden kann, die an dem Broker bekannt gegeben wurde.

Das Laden von Modulen bildet dann einen Baum von Methoden, die mit Modulen verbunden sind, und von Modulen, die mit dem Broker verbunden sind.

2.2.3 Broker

2.3 Robot Operating System

2.4 Programming Languages

2.4.1 C++

2.4.2 Python

2.5 Virtual Reality

2.6 Entwicklung für Virtual Reality

2.7 TCP

Kapitel 3

Technologieauswahl

3.1 VR-Brillen

3.2 Entwicklung für VR-Brillen

Kapitel 4

Umsetzung

4.1 Pepper

4.2 MetaQuest3

4.3 Verbindung

Kapitel 5

Fazit

Kapitel 6

Fortsetzung des Projekts