Научная статья

Original article

УДК 681.5.017

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ МАНИПУЛЯЦИОННЫМ РОБОТОМ С ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

DEVELOPMENT OF SOFTWARE AND HARDWARE COMPLEX FOR REMOTE CONTROL OF MANIPULATION ROBOT WITH VIRTUAL REALITY TECHNOLOGY

Жэнь Гуаншэн, магистрант МГТУ им. Н. Э. Баумана, Москва, Российская Федерация

Zhen Guangsheng, MSTU student N. E. Bauman, Moscow, Russian Federation

Аннотация

Виртуальная реальность — это технология, которая объединяет компьютерное оборудование и программное обеспечение для всестороннего применения различных технологий, таких как компьютерная графика, взаимодействие человека с компьютером, мультимедиа и сетевое общение, чтобы предоставить пользователям виртуальный трехмерный мир. Технология управления на основе виртуальной реальности имеет широкий спектр перспектив промышленного применения. Благодаря постоянному развитию коммуникационных технологий была создана технология 5G, а скорость связи на уровне миллисекунд может вывести технологию виртуального дистанционного управления на более высокий уровень. В будущем требования к контролю будут

все выше и выше, а рабочая среда еще хуже.Использование технологий виртуальной реальности и сверхдальнего управления может освободить персонал от ограничений рабочей среды. Управление роботом-манипулятором на основе технологии виртуальной реальности может предоставить пользователям захватывающий опыт, моделируя реальную среду с помощью системы виртуальной реальности, сочетая человеческую субъективную инициативу с точной работой робота-манипулятора.Роботы работают.

В данной работе разработана схема программно-аппаратного комплекса для дистанционного управления манипулятором с использованием технологии виртуальной реальности. Основные функции комплекса реализованы на имитационном программном обеспечении, надежность схемы доказана экспериментально.

Это эталонное решение для дистанционного управления промышленными роботами-манипуляторами с помощью технологии виртуальной реальности в сложных и опасных условиях.

Annotation

Virtual reality is a technology that integrates computer hardware and software to comprehensively apply various technologies such as computer graphics, human-computer interaction, multimedia, and network communication to provide users with a virtual three-dimensional world. Virtual reality-based control technology has a wide range of industrial applications. With the continuous development of communication technology, 5G technology has been created, and the millisecond communication speed can bring virtual remote control technology to a higher level. In the future, control requirements will become higher and higher, and the work environment will become even worse. The use of virtual reality technology and ultra-long-range control can free personnel from the limitations of the work environment. The control of the robot arm based on virtual reality technology can provide users with an immersive experience by simulating the real environment through the virtual reality system, combining human subjective initiative with the precise operation of the robot arm. The robots work.

In this work, a scheme of a software and hardware complex for remote control of a manipulator using virtual reality technology has been developed. The main functions of the complex are implemented on simulation software, the reliability of the circuit has been proven experimentally.、It is the reference solution for remote control of industrial robotic arms using virtual reality technology in complex and hazardous environments.

Ключевые слова: Виртуальная реальность; Манипулятор; Взаимодействие человека с компьютером.

Keywords: Virtual reality; Manipulator; Human-computer interaction.

Введение

Благодаря постоянному развитию робототехники роботы играют важную роль в развитии промышленного производства и ликвидации последствий стихийных бедствий. Однако в практических приложениях у робота также будут проблемы, такие как недостаточная гибкость терминала управления, ограниченная скорость передачи данных и узкое рабочее поле зрения. Особенно в случае обрушения здания и утечки токсичного газа, вызванного землетрясениями, взрывами химических заводов и т. д., это сильно угрожает безопасности спасателей. Как быстро и эффективно провести аварийно-спасательные работы с учетом обеспечения безопасности спасателей – одна из забот современного общества. Для решения этой проблемы в данной статье разработана система дистанционного управления манипулятором на основе виртуальной реальности, робот, оснащенный этой системой, заменит спасателя при входе на различные опасные и сложные объекты для обнаружения и поиска застрявших людей. Кроме того, по сравнению с традиционной системой дистанционного управления роботом, более заметным преимуществом этой системы является внедрение технологии виртуальной реальности, которая увеличивает ощущение присутствия оператора, упрощает операцию и повышает точность операции, а манипулятор маленький и гибкий, устанавливается на различных мобильных роботах.

Исследовательская работа по этой теме в основном делится на три аспекта. Одним из них является разработка решения на основе контроллера виртуальной реальности для управления манипулятором. Второй — установка на манипуляторе бинокулярной камеры для имитации человеческого глаза, а изображения, собранные удаленно, сжимаются с помощью технологии WebP и визуализируются в VR-шлеме в режиме реального времени. В-третьих, как обеспечить надежную и бесперебойную связь между Unity и ROS, а также возможность управлять удаленным манипулятором в режиме реального времени с малой задержкой.

Наконец, в результате общего теста было подтверждено, что система может быть более удобной для оператора, чтобы управлять манипулятором для удаленного, своевременного и эффективного выполнения задач, когда состояние сети хорошее.

Проектирование аппаратной архитектуры системы управления роботом

Ссылаясь на существующее решение для управления роботом-манипулятором VR, я в основном спроектировал свою систему управления из пяти частей. Вход — это оборудование VR, используемое оператором, особенно часть контроллера. Оператор отправляет команды управления роботом в виртуальный мир на компьютере через VR-контроллер. При этом оператор также может видеть через шлем виртуальный мир, созданный средой разработки в компьютере. После получения компьютером команды управления, выдаваемой гироскопом в контроллере ВР и базовой станции, он далее передает команду управления на устройство управления манипулятором нижнего уровня. Завершение расчета в аппаратуре нижнего уровня управления и управления движением исполнительного механизма. В то же время на захватной части манипулятора можно установить независимую камеру для передачи изображений в режиме реального времени, чтобы предоставить оператору информацию об окружающей среде. Камера может напрямую

передавать изображение на компьютер через WIFI и отображать и отображать его в виртуальном мире.

После определения различных используемых аппаратных устройств я спроектировал аппаратную архитектуру системы, как показано ниже.

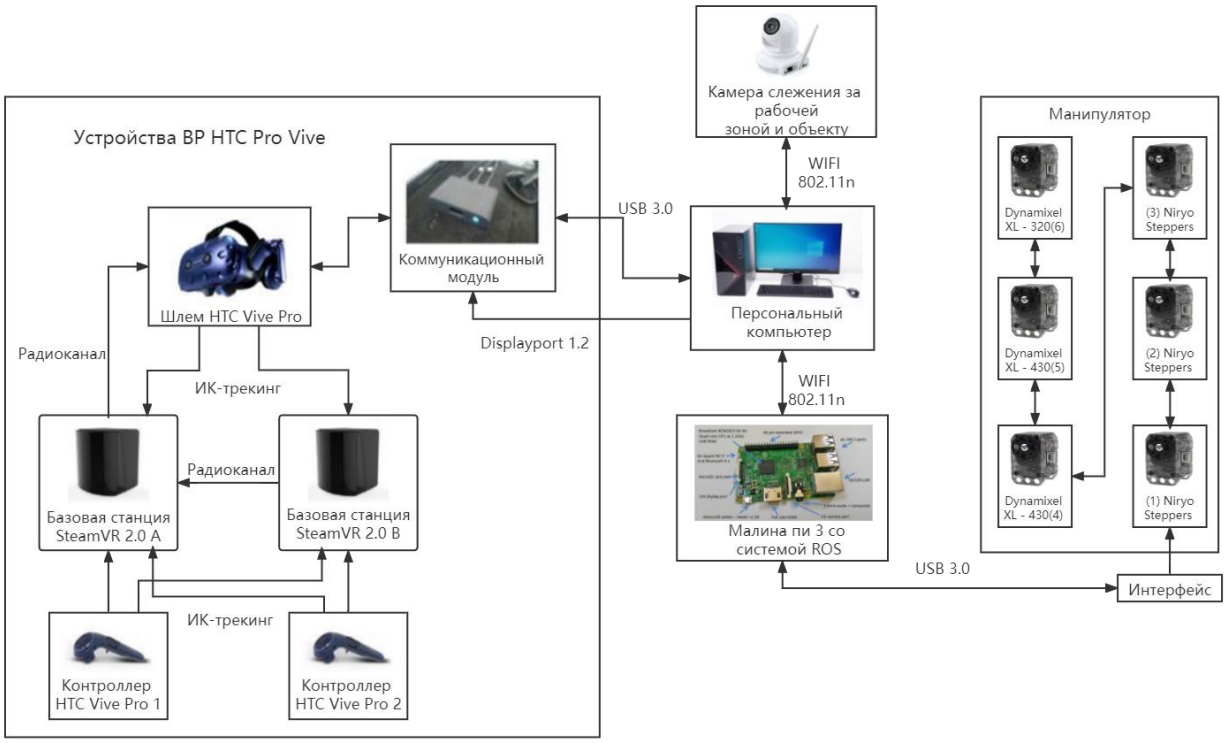

Рисунок 1 - Архитектура аппаратной

В части VR-устройства HTC Vive Pro оператор нажимает кнопки на контроллере, чтобы начать отправку команд управления. Данные об угле рыскания, угле тангажа и угле крена контроллера собираются через гироскоп в контроллере и передаются на базовую станцию. Базовая станция также оснащена технологией отслеживания пространственного позиционирования Lighthouse2.0. Технология определяет изменение пространственного положения манипулятора с помощью инфракрасных датчиков, установленных на базовой станции и контроллере. Наконец, базовая станция отправляет шестимерную информацию о положении и положении, включая изменение пространственного положения контроллера и угол рыскания, угол наклона и угол крена, на шлем с помощью технологии беспроводной передачи, а затем на компьютер через Коммуникационный модуль.

Верхний компьютер системы — это персональный компьютер, а нижний — Raspberry Pi 3, используемый Niryo One. Есть три функции, которые реализует персональный компьютер. Один из них заключается в установке Unity3 в качестве среды разработки технологии виртуальной реальности и передаче разработанного экрана виртуального мира на шлем, который носит оператор. Во-вторых, собирать живые изображения, передаваемые камерами, и отображать их в виртуальном мире в режиме реального времени. Третий — принять управляющую команду, выданную оператором, через контроллер и передать ее на следующий уровень Raspberry Pi. Raspberry Pi 3 устанавливает систему ROS, после получения управляющей команды от компьютера вычисляет ее и соответствующим образом управляет движением манипулятора.

Установление связи между роботом и компьютером

Связь между роботом и компьютером разделена на две части: передача видеопотока камеры и передача управляющих инструкций оператора.

Инструкции оператора по управлению манипулятором собираются VR-контроллером и отправляются в Unity, при этом управление манипулятором осуществляется через Raspberry Pi, оснащенный системой ROS. Для обеспечения удаленного управления необходимо установить беспроводную связь между ROS и Unity. После того, как команды управления оператора собраны посредством отслеживания позиционирования контроллера, их необходимо отправить в ROS, и соответствующая схема движения манипулятора рассчитывается с использованием решателя обратной кинематики ROS, а результаты возвращаются обратно в Unity3D.

Инструмент связи между ROS и Unity3D, после анализа и сравнения, мы решили использовать ROS# с открытым исходным кодом. Связь между двумя платформами может быть показана на рисунке 2.

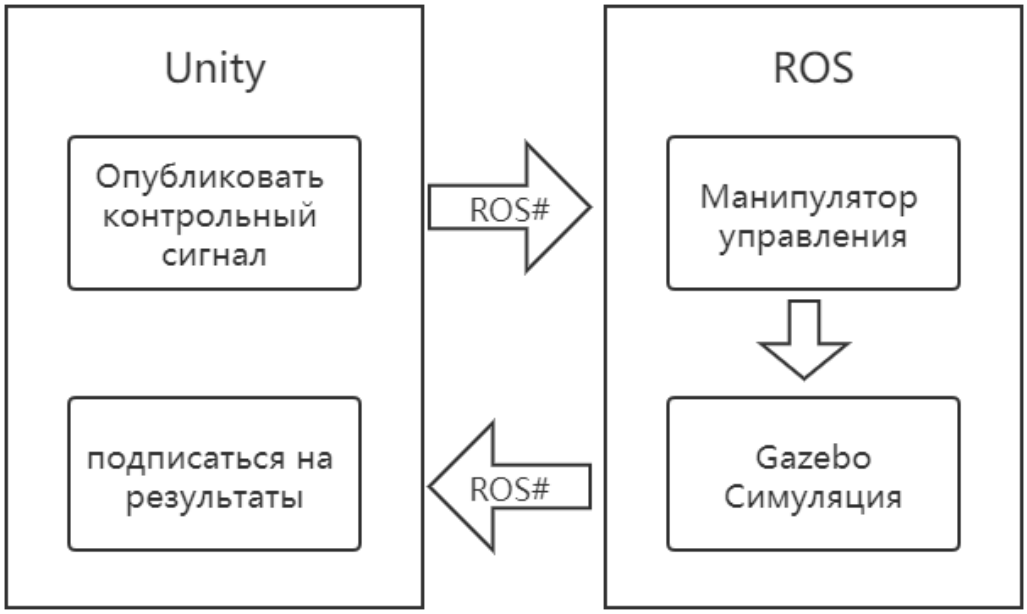

Рисунок 2 - Связи между ROS и Unity3D

ROS — это библиотека инструментов с открытым исходным кодом, помогающая программировать приложения для робототехники. Процесс соединения программ ROS называется узлом. Узлы обмениваются данными посредством потоковой передачи данных, каналов или тематических сетей в локальной сети TCP, называемой ROS. Узел создает объект издателя для подписки на тему через объект темы или подписчика в опубликованной сети данных.

Сначала установлю пакет ROS, затем найдите домашнюю страницу ROS# на github, загрузите и установите библиотеку инструментов ROS# в среде ROS в соответствии с руководством. Затем используйте связь ROS# на узле ROS, чтобы получить информацию о шестимерной позе от Unity, а затем используйте решатель обратной кинематики, который поставляется с ROS, для расчета схемы движения манипулятора.

После этого импортирую инструмент TCP-соединения ROS# в диспетчере ресурсов Unity3D, найдите IP-адрес сервера в ROS и введите его в Unity, чтобы установить соединение, как показано на следующем рисунке.

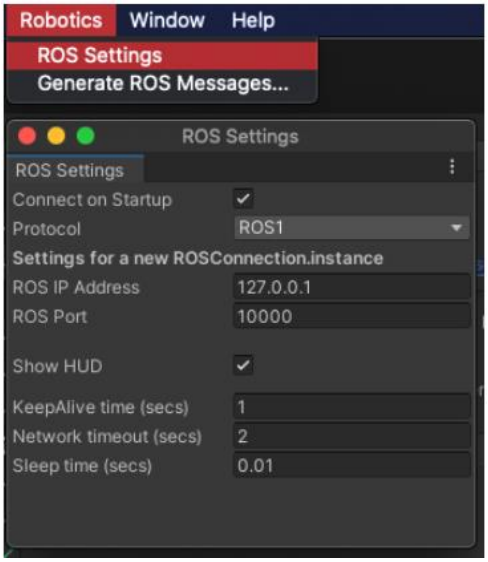

Рисунок 3 - TCP-коннектор

Передача данных видеопотока осуществляется путем установки сокет-сервера.Эта система завершает проектирование дисплея бинокулярного стереозрения в виртуальной среде разработки Unity3D Engine. После получения видеопотока бинокулярной камеры со стороны робота он обрабатывается виртуальной камерой в Unity, и, наконец, в Unity отображается стереоскопическое изображение конца робота, что повышает удобство работы оператора.

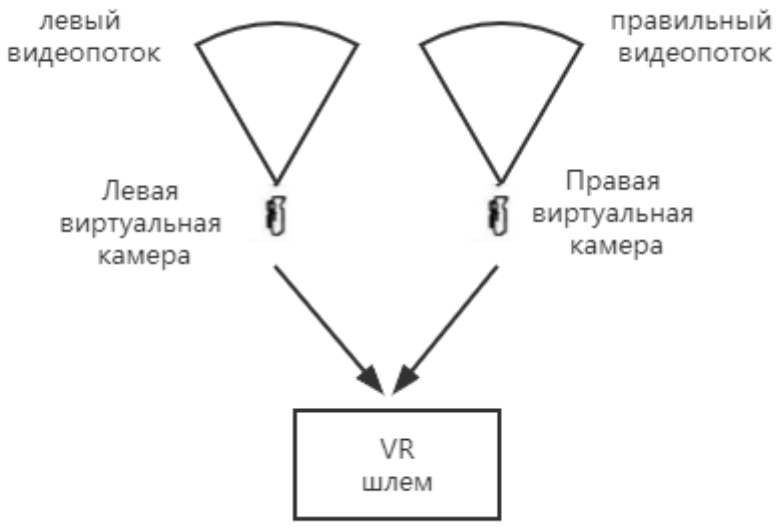

Рисунок 4 - Стерео видеопоток, отображаемый в Unity

Получение изображения существует на основе передачи изображения и является первым шагом в общем процессе. Потому что в устройстве получения

изображений используется бинокулярная стереокамера Logitech, а платформа разработки — Unity. В Unity для вызова камеры используется класс WebCamTexture. Часть кода показана ниже. Во-первых, я определяю класс WebCamTexture, использую переменную devices в классе, чтобы получить список доступных камер, и получаю имя устройства из devices . Затем я передаю имя устройства, устанавливаю длину, ширину, частоту кадров и другие параметры изображения, создаю экземпляр класса WebCamTexture и, наконец, запускаю его функцию WebCamTexture.play(), чтобы получить изображение, захваченное соответствующим устройством. Весь процесс будет зациклен в сопрограмма ImageGet, чтобы изображение можно было непрерывно собирать.

```
WebCamDevice[] devices = WebCamTexture.devices;
DeviceName = devices[0].name;
_webCameraTexture = new WebCamTexture(DeviceName,400 , 300, 30);
_webCameraTexture.Play();
```

Рисунок 5 - Код получения изображения

Чтобы видеть захваченные изображения в реальном времени на стороне Unity, необходимо создать панель рисования в исходном интерфейсе пользовательского интерфейса для отображения изображений в реальном времени. Panel — это GameObject в Unity, используемый для отображения изображения, снятого камерой.. Изображения, собранные с помощью класса WebCamTexture, не могут отображаться непосредственно на Panel, необходимо пройтись по пикселям каждого собранного изображения, чтобы получить информацию RGB, сохранить данные в формате PNG в области буфера, а затем добавить их в Texture. 2D, чтобы он появился на Panel.

Формат изображения, полученный в процессе получения изображения, — PNG, а объем данных относительно велик. Если он передается напрямую через сетевое окружение, часто возникает серьезная задержка. Поэтому данные изображения должны быть сжаты и преобразуется в небольшой объем данных в формате данных. Эта система выбирает для кодирования WebP, кодирование WebP делится на кодирование с потерями и без потерь, оно может сжимать изображения в соответствии с требованиями и обеспечивать качество

изображения при уменьшении объема данных. Поскольку исходный код WbP написан на C++, его необходимо инкапсулировать в библиотеку динамической компоновки (DLL), которую может использовать C#. А WebP поддерживает только двухпиксельные форматы Format3.2 bppArgb и Format3.2 bppRgb, вам необходимо используйте метод DrawImage для перерисовки. Изображение можно вызвать для кодирования изображения, вызвав функцию mazen.WebP.SimpleEncoder..Encode(). Третий параметр этой функции используется для настройки качества изображения. 100 — качество без потерь. код показан на следующем рисунке.

```
Stream stm = new memoryStream(buffer);
Image img = Image.FromStream(stm);
Bitmap bmp = new Bitmap(img.Width, img.Height, Imaging.PixelFormat.Format32bppArgb);
Graphics g = Graphics.FromImage(bmp);
g.DrawImage(img, 0, 0, img.Width, img.Height);
Imazen.WebP.SimpleEncoder.Encode(bmp, stm, 100);
buffer = Stm.getBuffer();
```

Рисунок 6 - Код для кодирования видеопотока

Сторона робота действует как клиент для стороны компьютера. Для передачи изображений на сервер необходимо общаться с сервером через Socket. Процесс выглядит следующим образом: сначала установите сокет, подключитесь к соответствующему IP-адресу сервера и установите номер порта, а затем создайте класс NetworkStream для указанного сокета. Класс NetworkStream может поддерживать синхронный или асинхронный доступ к сетевым данным, это можно рассматривать как построение соответствующего канала данных между источником данных и получателем, в этом случае чтение и запись данных могут быть направлены на этот канал. Для потока NetworkStream операция записи в основном происходит из передачи данных из буфера памяти в сеть, операция чтения в основном из сети в буфер памяти приемника для передачи данных. После создания экземпляра класса NetworkStream через этот класс отправляется запрос данных изображения, чтобы сервер был готов к приему данных. После того, как сервер отправляет обратно соответствующее подтверждающее сообщение, он начинает передавать данные изображения.

Процесс установления связи показан на следующем рисунке.

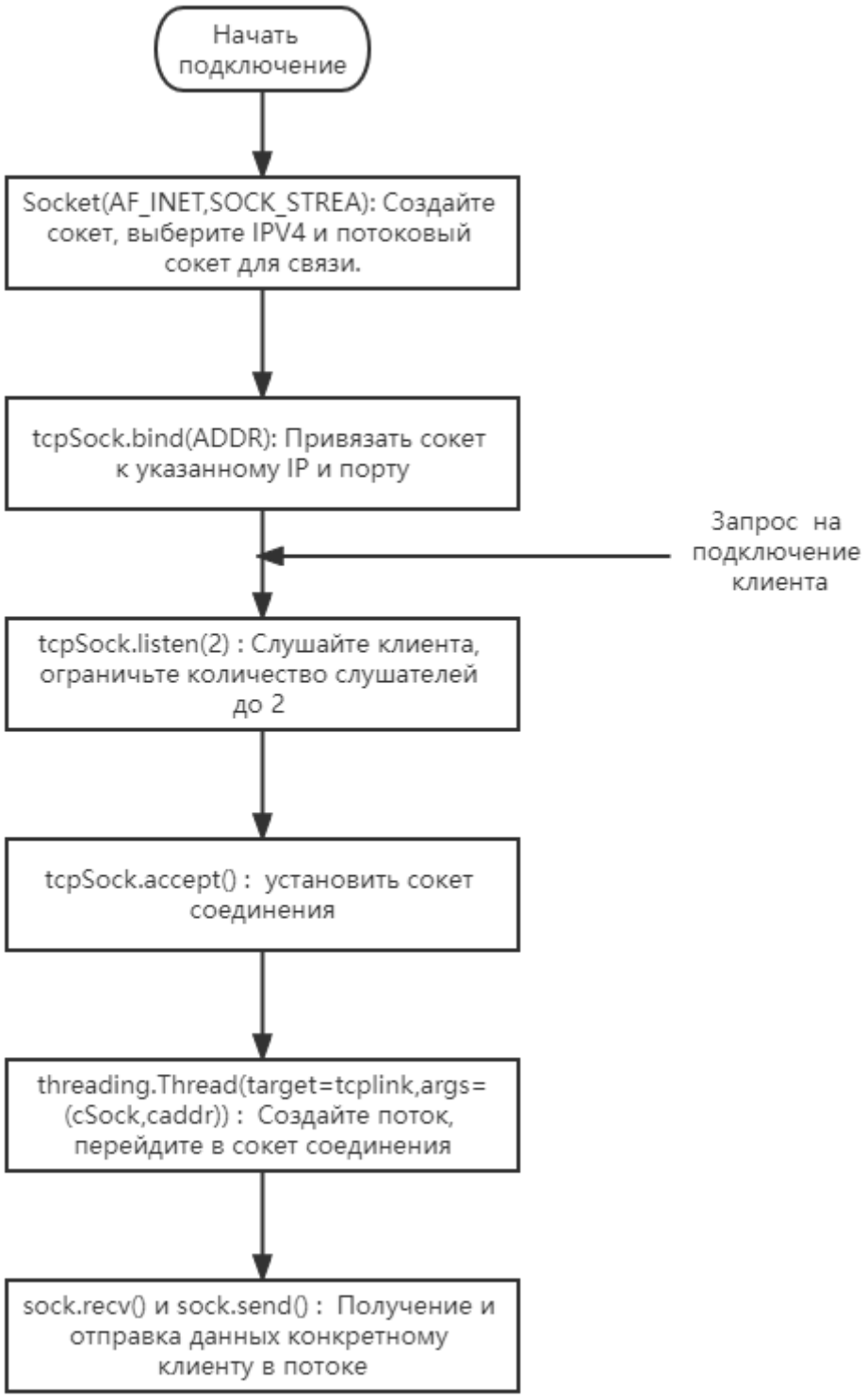

Рисунок 7 - Процесс потокового видео

Когда VR шлем необходимо отобразить видеоэкран в режиме реального времени, она инициирует видеозапрос к серверу, сначала создает сокет и использует IP-адрес сегмента сервера и указанный порт для подачи заявки на

подключение. Создайте поток NetworkStream и отправьте сообщение «готово» на сервер, чтобы сообщить серверу, что он готов получать данные изображения. Сервер получает IP-адрес VR шлем и рассматривает его как адрес назначения для передачи изображения для реализации задачи пересылки.

Наконец, экран стереозрения, синтезированный бинокулярным видеопотоком, успешно реализован на платформе Unity.



Рисунок 8 - Отображение стереоскопического вида в Unity

Конфигурация контроллера VR

Для получения команд контроллера VR необходимо настроить действия контроллера VR. Эта система разработана на движке Unity с использованием комплекта разработки SteamVR. Формат вывода команды виртуального контроллера задан как тип шестимерного массива.При срабатывании триггерной

кнопки контроллера VR информация о координатах контроллера VR может быть получен.

В последней версии SteamVR предыдущий интерфейс ISteamGameController устарел, а для редактирования контроллера используется визуальный интерфейс SteamVR Input. После успешного подключения контроллера HTC VIVE PRO, SteamVR Input может автоматически определить модель контроллера и открыть интерфейс для редактирования настроек кнопок этого типа контроллера, как показано на рисунке ниже. В этом интерфейсе вы можете выбрать общий дизайн или частичный дизайн контроллера в соответствии с потребностями дизайна системы, и соответственно спроектировать кнопку запуска, кнопку меню, сенсорную панель и т. д.



Рисунок 9 - Редактировать интерфейс SteamVR Input

Конкретный рабочий процесс выглядит следующим образом: нажмите кнопку триггера, чтобы используйте функцию ActivateActionSet() создать новый набор действий Pose и выберите тип виртуального дескриптора Pose в качестве данных действия типа Vector.3 в типе действия, чтобы вернуть шестимерный массив. Затем используйте функцию ActivateActionSetLayer(), чтобы перенастроить и указать набор действий, используемый контроллером. Затем

используйте функцию InputMotionData_t() для сбора и ввода изменений позы контроллера в виде кватернионов. Затем используйте k_EInputActionOrigin_Gyro_Move, k_EInputActionOrigin_Gyro_Pitch, k_EInputActionOrigin_Gyro_Yaw, k_EInputActionOrigin_Gyro_Roll четыре функции интерфейса для чтения данных в гироскопе контроллера, записи их в набор данных действий и, наконец, отправки данных в ROS через ROS#.

Поскольку в Unity3D нет функции кинематического решения, для фактического управления роботом также требуется ROS. Набор данных действий, упакованный в Unity, передается в ROS через ROS# и обрабатывается в соответствии с методом, описанным в предыдущем содержимом, могут быть получены фактические результаты кинематических расчетов, а результаты моделирования могут быть отправлены обратно в Unity.

Конкретный процесс показан на следующем рисунке.

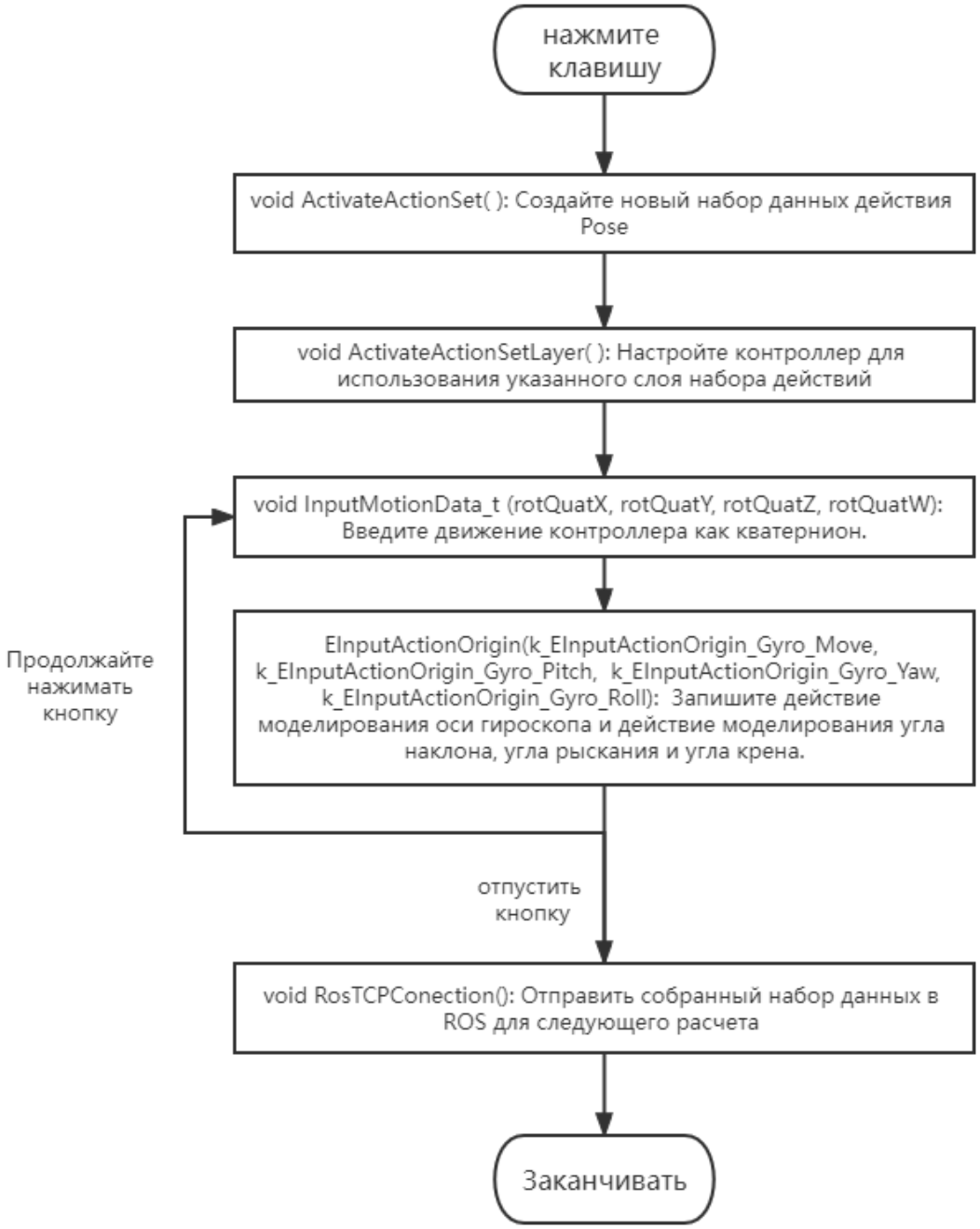

Рисунок 10 - Процесс реконфигурации контроллера

Таким образом, реконфигурация контроллера виртуальной реальности в Unity завершена для управления движением манипулятора.

Эксперименты для проверки функциональности

Содержание этой главы основано на дистанционном управлении виртуальным манипулятором с помощью ранее исследованного и разработанного

контроллера виртуальной реальности, и проверяется надежность функции дистанционного управления в этой системе. В тесте использовались виртуальный манипулятор и оборудование виртуальной реальности.После подключения ручки виртуальной реальности можно было наблюдать предварительно настроенную модель руки в виртуальном пространстве, и модель могла реагировать на движения руки оператора и изменения позы в режиме реального времени. Оператор держит контроллер виртуальной реальности и использует контроллер виртуальной реальности для управления виртуальным манипулятором, чтобы имитировать дистанционно управляемый манипулятор в реальной среде. Оператор использует контроллер VR для управления четырьмя движениями манипулятора вверх, вниз, влево, вправо и вращением запястья соответственно, чтобы проверить и сравнить эффект и задержку выполнения этих движений. Процесс тестирования показан на рисунке ниже.

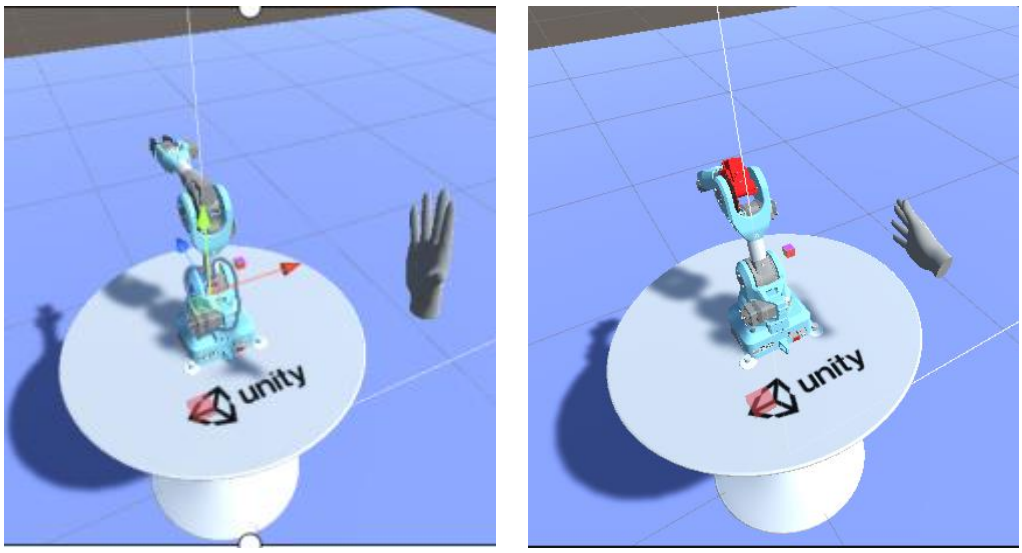

Рисунок 11(а) - Контроллер управляет манипулятором вниз

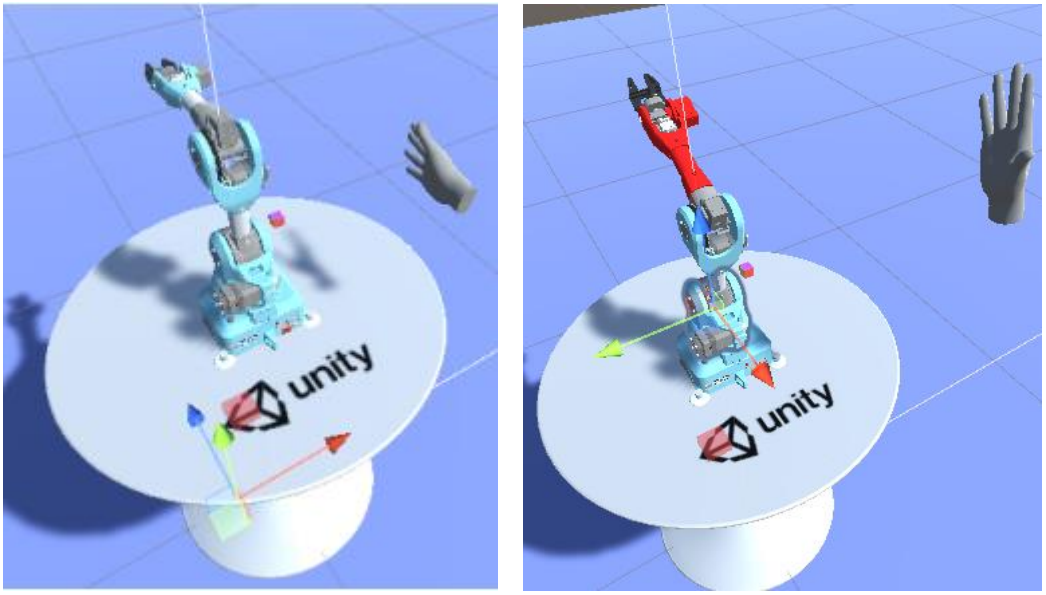

Рисунок 11(b) - Контроллер управляет манипулятором вверх

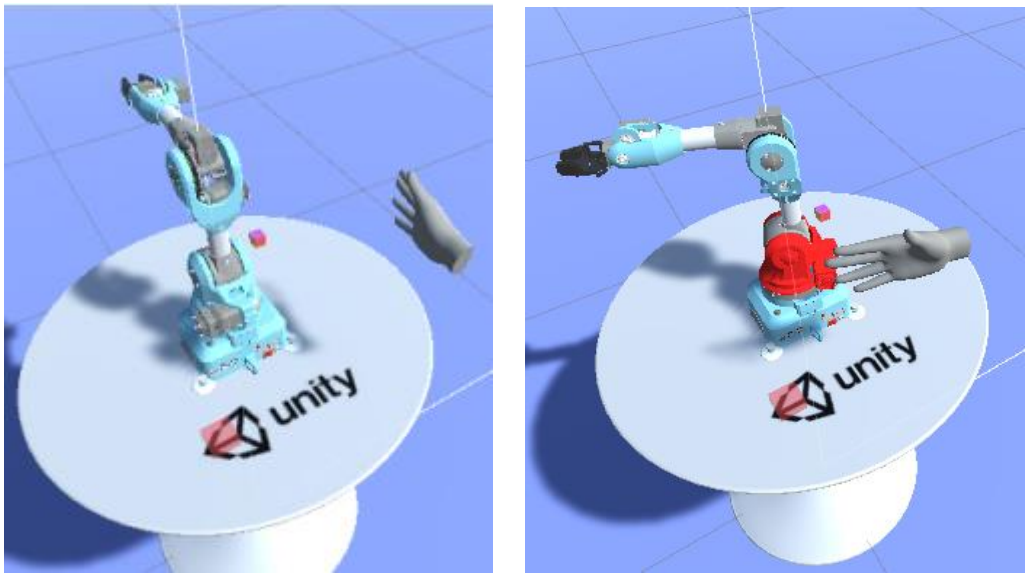

Рисунок 11(c) - Контроллер управляет манипулятором слева

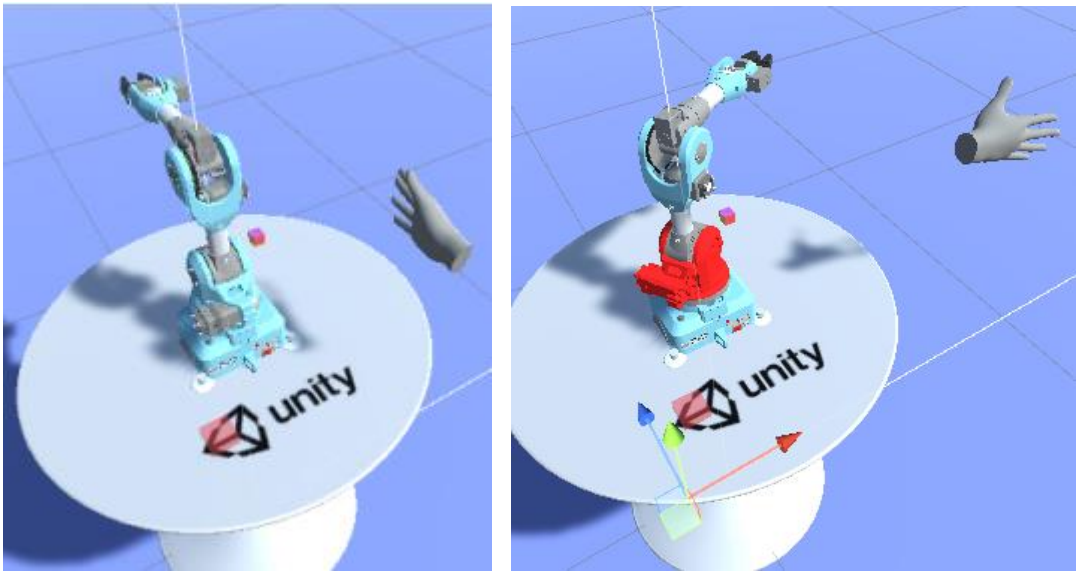

Рисунок 11(d) - Контроллер управляет манипулятором справа

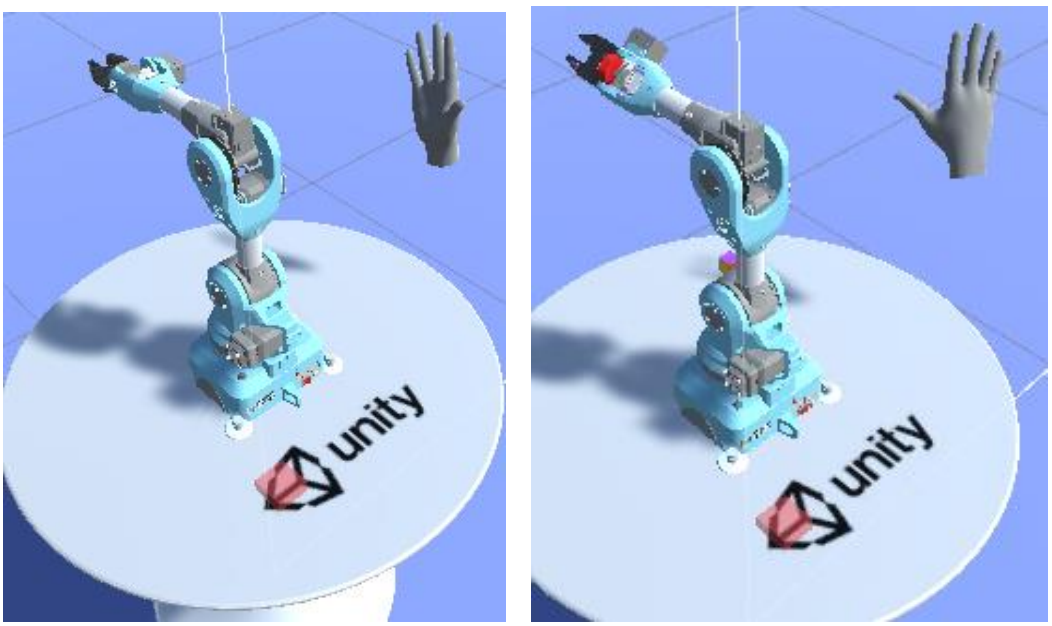

Рисунок 11(e) - Контроллер управляет вращением механического запястья.

В процессе проверки функции дистанционного управления выполняется несколько тестов данных о движении в каждом направлении, количество случайных тестов в каждом направлении, тестер случайным образом контролирует направление движения манипулятора и подсчитывает количество ответов манипулятора. манипулятора и количество задержанных ответов. Результаты приведены в таблице. Тесты с временем реакции на движение более 1 секунды классифицировались как отложенные реакции во время тестирования, но ни в одном из тестов время реакции не превышало 2 секунд.

Таблица 1 - Базовый функциональный тест контроллера VR для управления манипулятором

Направление движения манипулятора	Количество тестирования	Количество успешных тестов		
		успешный ответ	неправильный ответ	Нет ответа
вниз	20	20	0	0
вверх	19	18	1	0
слева	18	18	0	0
справа	22	21	1	0
вращение	23	21	2	0

Заключение

В этой статье в основном изучается система дистанционного управления роботом в качестве носителя, сочетается популярная в последние годы технология виртуальной реальности с технологией дистанционного управления роботом и реализуется изображение удаленного мониторинга, отображаемое после рендеринга шлема VR. Виртуальная реальность с управлением контроллера позволяет достичь цели простого и удобного дистанционного управления движением в реальном времени.

Я объединил эти две технологии, чтобы спроектировать эту систему дистанционного управления манипулятором, основанную на технологии виртуальной реальности, и экспериментально доказал, что система может эффективно помогать оператору выполнять различные задачи.

Список литературы

1. [1] Цзоу Сянцзюнь. Эволюция и перспективы технологии виртуальной реальности.// Журнал системного моделирования - 2004 - №8.
2. [2] Кот. Т, Новак. С. Применение виртуальной реальности в телеуправлении военной мобильной робототехнической системой TAROS.// Международный журнал перспективных робототехнических систем. -2018.
3. [3] Ding Guofu. Investigation of Excavator Remote Operation Simulation Based on Natural Reality.// Journal of the Southwestern Jiaotong Society. - 2002 - No. 2.
4. [4] Э. Баклути, М. Джаллули, Л. Амури и Н. Бен Амор. Дистанционное управление мобильным роботом через трехмерную среду виртуальной реальности.// Международная конференция по индивидуальному и коллективному поведению в робототехнике (ICBR) - 2013 - с. 9-14.
5. [5] Чжан Синьи. Построение тестовой среды моделирования поля боя на основе технологий Unity3D и VR. // Сидянский университет - 2020 - с.12-17.
6. [6] Джен И, Таха. Z, & Vui, L. Система программирования и моделирования роботов на основе виртуальной реальности для промышленного робота. // Международный журнал промышленной инженерии: теория, приложения и практика. - 2008 - 15(3) - 314-322.
7. [7] Ван Минюй. Виртуальная система вождения на базе Unity3D и технологии VR // Сидянский университет - 2020 - с.4-6.
8. [8] Ван Хан ,Чжан Ихонг , Ван Шаоди. Unity AR / VR Development: практический тренинг для мастеров // Опубликовано Machinery Industry Press - 2021.
9. [9] Тан Ин, Тан Чанжуй, Сян Илинь.Проектирование и реализация обучающего экзамена по вождению в виртуальной реальности на основе

Unity3D и HTC VIVE [J] // Высокие технологии Китая, 2020(07):49-50.DOI:10.13535/j .цнки .10-1507/н.2020.07.015.

10. [10] Чжан Ци.Прикладное исследование метода жестового взаимодействия на базе HTC Vive [D] // Пекинский технологический университет, 2019. DOI: 10.26935/d.cnki.gbjgu.2019.000054.
11. [11] Ши Шаовей, Ши Шаоминь, Интеллектуальная система мониторинга рабочего состояния робота на основе виртуальной реальности [J],// Автоматизация и приборостроение, 2021(12):172-175+180. DOI:10.14016/j.cnki.1001-9227.2021 .12.172 .

References

1. [1] Zou Xianjun. Evolution and technologies of the future of reality.// Journal of system modeling - 2004 - No. 8.
2. [2] Kot. T, Nov'ak. P. Application of virtual reality in teleoperation of the military mobile robotic system TAROS.// International Journal of Advanced Robotic Systems. -2018.
3. [3] Ding Guofu. Investigation of Excavator Remote Operation Simulation Based on Natural Reality.// Journal of the Southwestern Jiaotong Society. - 2002 - No. 2.
4. [4] E. Baklouti, M. Jallouli, L. Amouri and N. Ben Amor. Remote control of mobile robot through 3D virtual reality environment.// International Conference on Individual and Collective Behaviors in Robotics (ICBR) - 2013 - c. 9-14.
5. [5] Zhang Xinyi. Building a battlefield simulation test environment based on Unity3D and VR technologies. // Sidian University - 2020 - c.12-17.
6. [6] Jen Y, Taha. Z, & Vui, L. VR-Based Robot Programming and Simulation System for an Industrial Robot.// International Journal Of Industrial Engineering: Theory, Applications And Practice. - 2008 - 15(3) - 314-322.
7. [7] Wang Mingyu. Virtual driving system based on Unity3D and VR technology // Sidian University - 2020 - p.4-6.

8. [8] Wang Han, Zhang Yihong, Wang Shaodi. Unity AR/VR Development: Hands-On Training for Craftsmen // Published by Machinery Industry Press - 2021.

9. [9] Tan Ying, Tan Changrui, Xiang Yilin. Designing and Implementing a Virtual Reality Driving Training Test Based on Unity3D and HTC VIVE [J]. // China High Tech, 2020(07):49-50.DOI:10.13535/j .tsnki .10-1507/n.2020.07.015.

10. [10] Zhang Qi. Applied Research on HTC Vive Gesture Interaction Method [D], Beijing University of Technology, 2019.

11. [11] Shi Shaowei, Shi Shaomin, Virtual Reality-Based Intelligent Robot Operating Condition Monitoring System [J]//Automation and Instrumentation, 2021(12):172-175+180.DOI:10.14016/j.cnki.1001-9227.2021.12.172 .

© Жэнь Гуаншэн, 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.

Для цитирования: Жэнь Гуаншэн ИМПОРТОЗАМЕЩЕНИЕ В КОНТЕКСТЕ ГАРМОНИЗАЦИИ АГРОПРОДОВОЛЬСТВЕННОЙ СФЕРЫ РОССИИ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.