

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
« _____ » _____ 2021 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Анимация 3D-модели человека по технологии инерциального захвата движения.

Исполнитель
студент группы 957-ом _____ К.Ю. Решетов
(подпись, дата)

Руководитель
профессор, док. техн. наук _____ И.Е. Еремин
(подпись, дата)

Руководитель научного
содержания программы
магистратуры
профессор, док. техн. наук _____ И.Е. Еремин
(подпись, дата)

Нормоконтроль
Инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Рецензент
Генеральный директор
ООО «Тематика» _____ Р.А. Кузьменко
(подпись, дата)

Благовещенск 2021

РЕФЕРАТ

Магистерская диссертация содержит 84 с., 52 рисунка, 42 источника.

АНИМАЦИЯ 3D-МОДЕЛИ ЧЕЛОВЕКА, MOTION CAPTURE, ЗАХВАТ ДВИЖЕНИЯ, ТЕХНОЛОГИЯ ИНЕРЦИАЛЬНОГО ЗАХВАТА ДВИЖЕНИЯ

Целью данной выпускной квалификационной работы является создание из общедоступных компонентов и обладающего относительно низкой стоимостью костюма инерциального захвата движения для компьютерной анимации 3D-модели человека. Предложенная методика может ускорить процесс создания анимации или разработки компьютерной игры, т. к. при ее использовании отпадет необходимость анимировать персонажи покадрово и вручную

В данной работе описывается технология создания инерциального костюма захвата движения, построенного на программно-аппаратной платформе Arduino, датчиках абсолютной ориентации BNO055 и I2C-мультиплексор TCA9548A, и разработки программного обеспечения для передачи структурированной информации в 3D редактор Blender для анимации трехмерной модели человека.

Задачи данной выпускной квалификационной работы:

- изучение существующих программно-технических средств, используемых в технологии захвата движения;
- техническое решение инерциального костюма;
- разработка программного обеспечения для инерциального костюма.

СОДЕРЖАНИЕ

Введение	5
1 Компьютерные технологии захвата движения	6
1.1 Классификация методов захвата движения	6
1.2 Инерциальная технология захвата движения	14
2 Алгоритмическое и программное обеспечение решения задачи	16
2.1 Разработка алгоритма анимации 3D-модели человека	16
2.1.1 Существующие алгоритмы инерциального захвата движения	16
2.1.2 Предлагаемый алгоритм инерциального захвата движения	24
2.2 Обзор технического обеспечения реализуемого алгоритма	26
2.2.1 Система захвата движения XSENS MVN Animate	26
2.2.2 Система захвата движения Motionshadow	27
2.2.3 Система захвата движения Rokoko Smartsuit Pro	29
2.3 Обзор программного обеспечения реализуемого алгоритма	30
2.3.1 Коммерческие программные продукты	30
2.3.2 Свободно распространяемые программные продукты	33
2.4 Программно-техническое решение задачи	41
2.4.1 Обоснование выбора электронных компонентов	41
2.4.2 Обоснование выбора программных сред разработки	48
3 Программно-техническая реализация предлагаемого решения	50
3.1 Основные этапы разработки инерциального костюма	50
3.1.1 Проектирование электротехнической схемы костюма	50
3.1.2 Разработка микрокода сбора данных	55
3.1.3 Разработка программы интерпретации данных	60
3.2 Примеры фактического тестирования инерциального костюма	66
3.2.1 Аппаратное и программное обеспечение для тестирования	66
3.2.2 Сценарий тестирования	67
3.2.3 Результаты тестирования	69
3.3 Анализ практической значимости результатов	72

Заключение	73
Библиографические ссылки	72
Библиографический список	80

ВВЕДЕНИЕ

Анимация 3D-модели человека применяется при разработке компьютерных игр, производстве фильмов и т.д., но если анимация персонажа осуществляется вручную, то данный процесс занимает продолжительное время, т.к. необходимо анимировать 3D-модель покадрово и вручную. Для ускорения процесса анимации 3D-человека в настоящее время активно применяются различные программно-технические средства, например, системы захвата движения.

Для ускорения анимации 3D-модели человека в настоящее время активно применяется технология захвата движения человека (Motion Capture) и оценка его кинематических и динамических параметров в реальном масштабе времени. Технология применяется во многих областях науки, медицине, спорте, искусстве и индустрии развлечений [1], но большая часть представленных на рынке систем производится за рубежом и обладают высокой стоимостью [2].

По данным сайта <https://arxiv.org/> количество зарубежных публикаций по теме Motion capture составляет 341, тогда как на сайте <https://cyberleninka.ru/> представлено только 10 отечественных публикаций и 84 автореферата. По этой причине научные исследования в данной области актуальны.

Предмет: компьютерная технология захвата движения.

Объект: анимация 3D-модели человека.

Целью работы является создание из общедоступных компонентов и обладающего относительно низкой стоимостью костюма инерциального захвата движения для компьютерной анимации 3D-модели человека.

Задачи работы:

- изучение существующих программно-технических средств, используемых в технологии захвата движения;
- техническое решение инерциального костюма;
- разработка программного обеспечения для инерциального костюма.

1 КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ ЗАХВАТА ДВИЖЕНИЯ

1.1 Классификация методов захвата движения

Для ускорения процесса анимации 3D-человека в настоящее время активно применяется технология захвата движения, т.к. если анимация персонажа осуществляется вручную и покадрово, то данный процесс занимает продолжительное время. Технологии захвата движения обычно включают синхронизированные камеры и специальные костюмы с маркерами или датчиками, которые носят исполнители. Маркеры устанавливаются на частях тела (или суставах) и отслеживаются, чтобы идентифицировать движение по их положению или углам. Записывается только движение актеров, а не их внешний вид, и это движение отображается на 3D-модель с помощью компьютера [3].

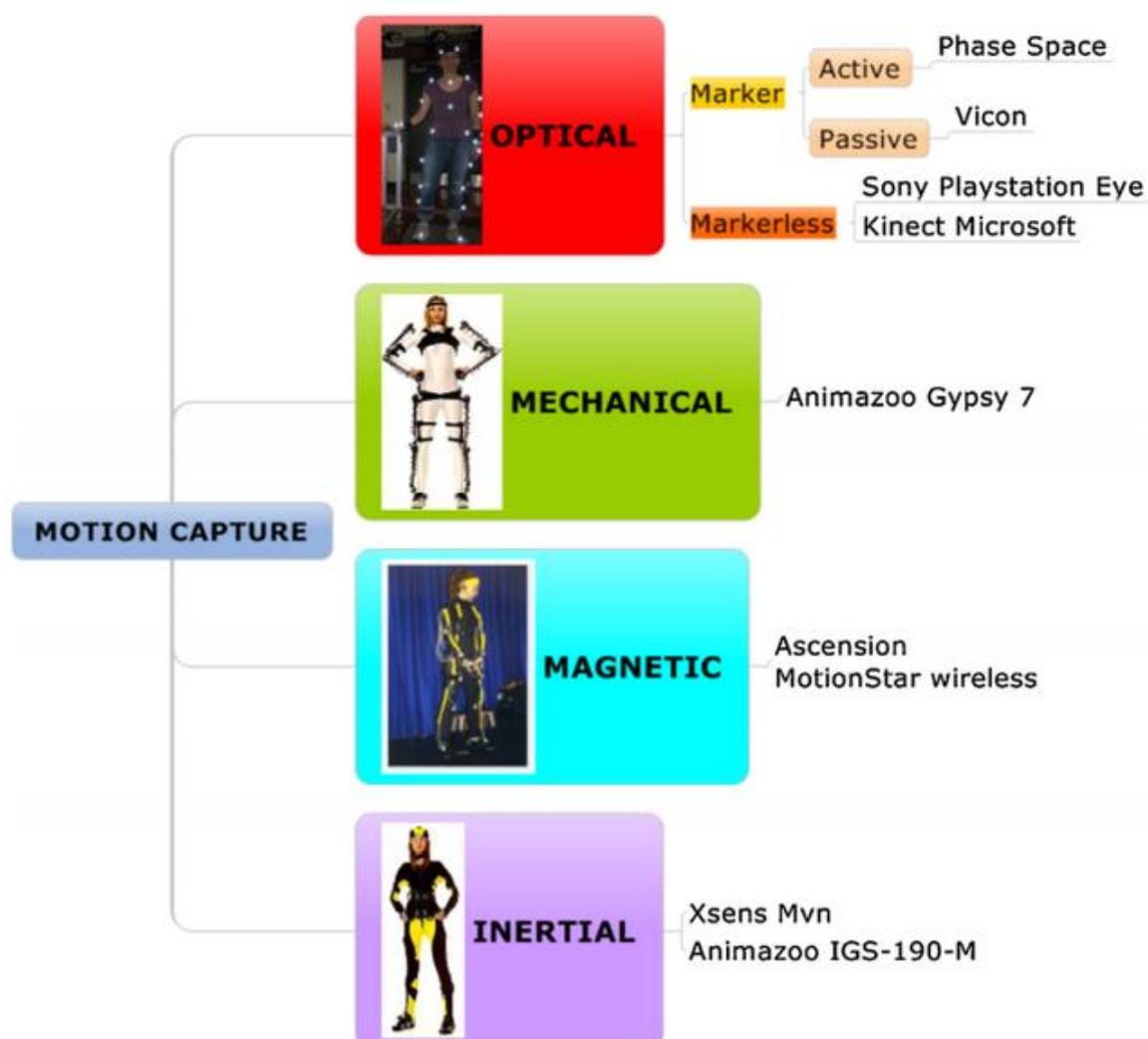


Рисунок 1 – Общая классификация систем захвата движения

Методы захвата движения могут быть классифицированы по их методам ввода, а именно оптическим, механическим, магнитным, акустическим и инерционным. Общая классификация систем захвата движения представлена на рисунке 1 [4].

Оптические системы. В оптических системах захвата движения сбор данных осуществляется с помощью специальных маркеров, прикрепленных к актеру; этот подход использует по крайней мере три синхронизированные камеры и надлежащее освещение для оценки положения исполнителя в 3D-пространстве. Эти системы создают данные с 3D-местоположениями для каждого маркера. Можно использовать два типа маркеров: пассивные и активные [5].

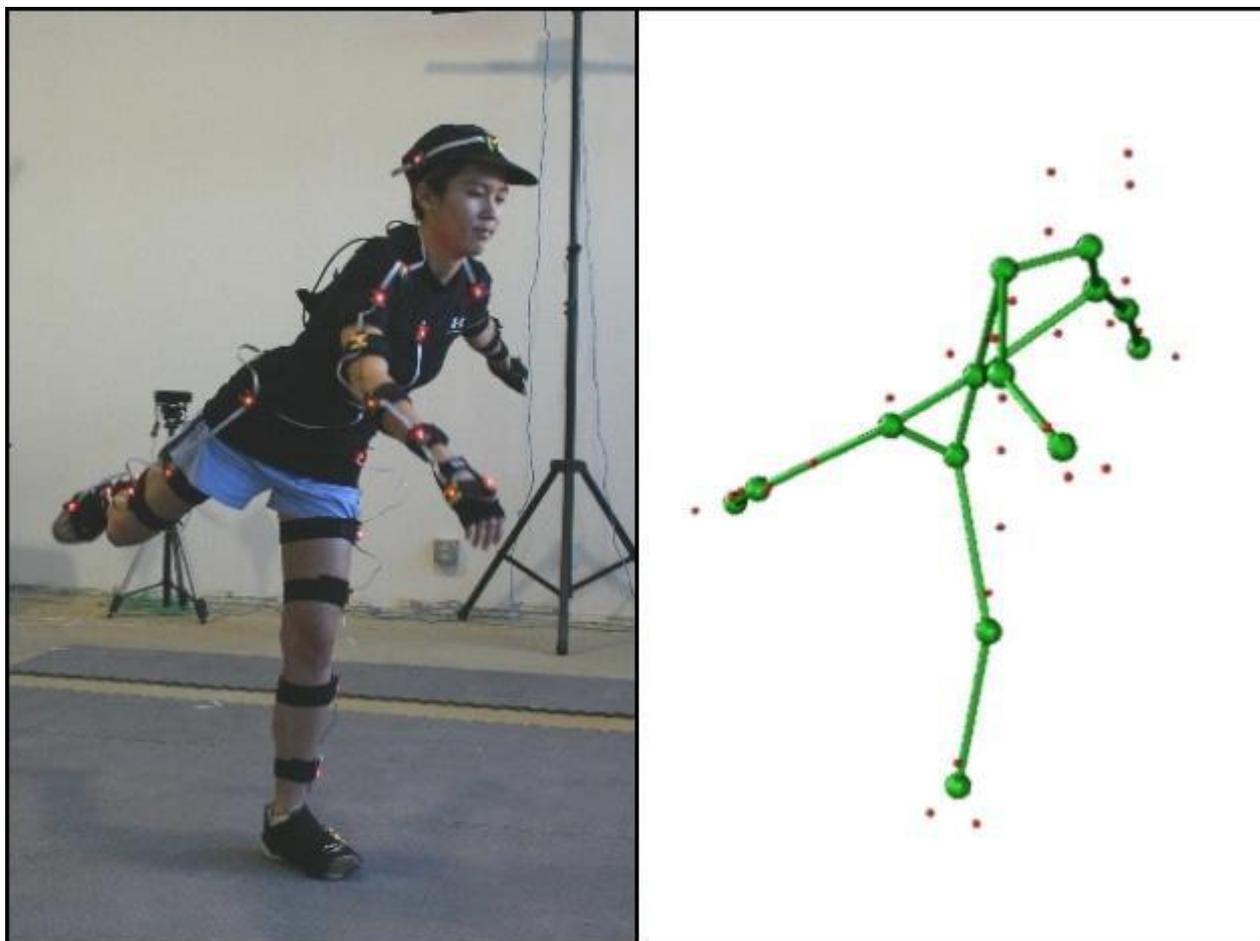


Рисунок 2 – Оптическая система захвата движения

На рисунке 2 [6] представлена оптическая система захвата движения. Актер в костюме оптического захвата движения с инфракрасными маркерами (слева). 3D-поза человека, выведенная из относительного положения каждого

маркера (справа).

Пассивные маркеры: Пассивная оптическая система использует маркеры, покрытые светоотражающим материалом, для отражения заднего света, генерируемого вблизи объектива камеры. Принцип действия аналогичен радару: камеры излучают излучение (обычно инфракрасное), которое отражается маркерами и возвращается в ту же камеру. Камеры чувствительны к узкому диапазону длин волн и воспринимают маркеры как яркие пятна. Такая система обычно состоит из 6-24 камер. [7]

Пассивные системы не требуют от пользователя ношения проводов или электронного оборудования. Маркеры обычно прикрепляются непосредственно к коже или к костюму из лайкры для всего тела, предназначенному для захвата движения.

Активные маркеры. Вместо того, чтобы отражать свет, генерируемый извне, сами маркеры работают на излучение собственного инфракрасного света. Активные оптические системы триангулируют позиции, очень быстро освещая один светодиод за раз или несколько светодиодов с помощью программного обеспечения, позволяющего идентифицировать их по их относительному положению. Некоторые системы могут модулировать амплитуду или ширину импульса, чтобы обеспечить идентификатор маркера. Этот уникальный идентификатор маркера обеспечивает гораздо более чистые данные, чем пассивные маркерные системы [8].

Безмаркерные оптические системы не требуют маркеров и дополнительного оборудования, кроме камер для съёмки. Специальные алгоритмы позволяют системе анализировать несколько потоков изображения и определять форму человеческого силуэта, движения частей тела (пример такой системы Kinect от Microsoft). Такие системы обладают низкой точностью, на их работу сильно влияет освещение, окружение, в котором находится актёр [9].

Механические системы. Механические методы используют экзоскелет – скелетно-механическую конструкцию, прикрепленную к телу исполнителя (рис. 3). Экзоскелеты представляют собой жесткие конструкции из металличе-

ских или пластиковых стержней, которые сочленяются в суставах тела, соединенных вместе с потенциометрами. Они непосредственно отслеживают углы суставов тела с помощью датчиков, прикрепленных к экзоскелету. Они предоставляют реальную информацию, свободную от окклюзии, однако костюм экзоскелета может быть тяжелым и громоздким, ограничивая свободу передвижения пользователя [10].

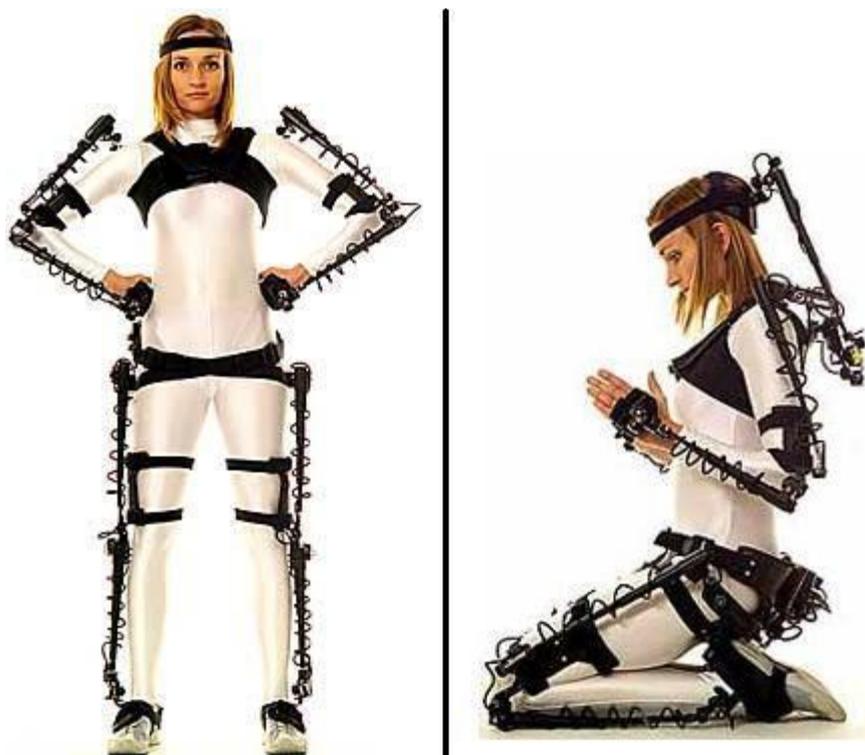


Рис. 3 – Механический экзоскелет для захвата движения.

Магнитные системы. Магнитные системы используют датчики, расположенные на корпусе, для измерения низкочастотного магнитного поля, генерируемого источником передатчика (рис. 4). Выходами этих датчиков являются 3D-положения и информация о вращении. Количество используемых датчиков обычно составляет от 6 до 11. Датчики и источник подключены к электронному блоку управления, который соотносит их местоположение в магнитном поле. Обратная кинематика затем используется для восстановления углы для различных суставов тела. Большим преимуществом этих систем является то, что полезные результаты в реальном времени могут быть получены только с 6 датчиков. Однако датчики чувствительны к магнитным и электрическим помехам от

металлических предметов в окружающей среде. Кроме того, проводка от датчиков имеет тенденцию ограничивать движения актера. В настоящее время разрабатываются новые беспроводных магнитных систем [11].

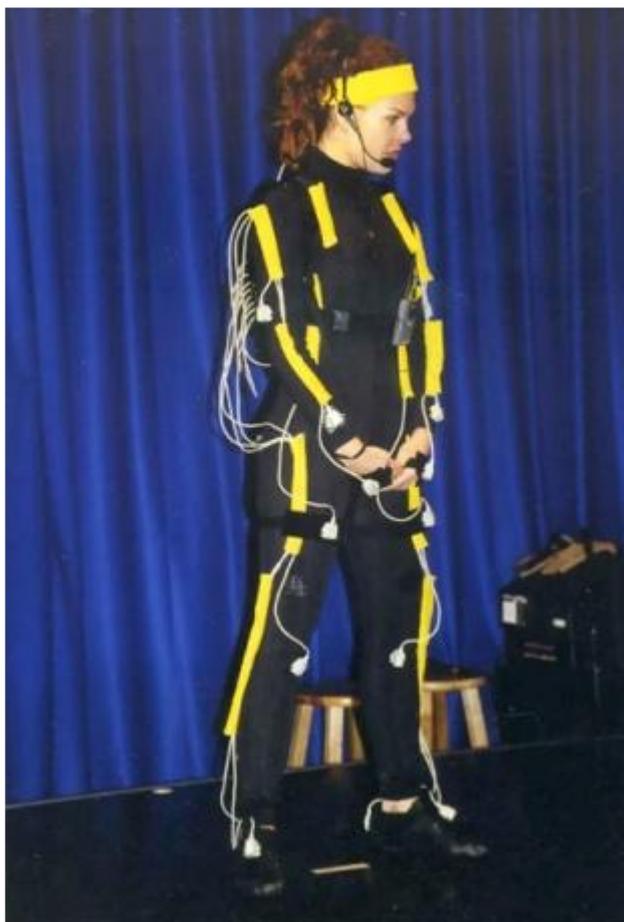


Рисунок 4 – Исполнитель в костюме для магнитного захвата движения

Акустические системы. Акустические системы используют три аудиоприемника и множество аудиопередатчиков на теле исполнителя. Аудиопередатчики посылают щелкающие звуки, а приемник измеряет время в пути от каждого передатчика. Вычисленное расстояние от трех приемников триангулируется, чтобы обеспечить точку в трехмерном пространстве. Эти данные о положении обычно подаются в обратную кинематическую систему, которая, в свою очередь, управляет анимированным скелетом. Преимуществом этого метода является относительное отсутствие окклюзий. К сожалению, на точность могут повлиять звуковые отражения или звуковые помехи. Система Power Glove основана на принципе акустического захвата движения [12].

Системы на основе компьютерного зрения. Использование методов компьютерного зрения для определения движения человеческого тела является одним из наиболее привлекательных и практичных решений, поскольку оно не требует дорогостоящего или инвазивного оборудования или маркеров (требуются только камеры) и может работать на открытом воздухе (на улицах, в офисах, парках).

Были предложены алгоритмы, которые фиксируют движение человека с частотой кадров, близкой к реальной; однако они в основном полагаются на системы с несколькими камерами в контролируемых условиях, которые ограничивают их применимость. Некоторые подходы к монокулярному зрению направлены на захват конкретных движений (ходьба, качание в гольф, прыжки и т.д.) С использованием некоторой модели обучения или отслеживания движения для определенных частей тела. Некоторые другие системы могут отслеживать свободное движение, но не работают в режиме реального времени.

В последнее время датчики 3D-изображений используются для захвата 3D-формы тела и устранения неоднозначных поз с помощью измерений глубины. Датчики времени полета и система активной триангуляции (например, Microsoft Kinect) являются такими специализированными датчиками. Отслеживание позы человека в режиме реального времени без маркеров с помощью монокулярного компьютерного зрения остается сложной, но актуальной проблемой [13].

Многие современные коммерческие системы используют одновременно несколько принципов захвата движения [14].

В зависимости от типа используемых датчиков системы делятся на два типа: активные и пассивные. В активной системе используются датчики, которые могут мгновенно передавать данные о текущей ситуации с целью определения их местоположения. Системы такого типа используют активные датчики, такие как перемещение данных, которые должны предоставляться непосредственно узлам костюма на основе обработки сигнала, полученного от внешнего источника, или определения угла между отдельными частями костюма. Глав-

ным преимуществом таких систем является удобство и точность обработки собранных данных. Однако такие системы подходят не для всех ситуаций. Активные датчики обычно имеют относительно большие размеры и поэтому способны препятствовать свободе передвижения, что позволяет получать необходимую информацию о маломасштабных перемещениях. А кроме того, использование системы захвата движения с использованием активных датчиков является активным развитием экономики и даже имеет много преимуществ перед системами с пассивными датчиками. Одним из главных преимуществ активных датчиков является относительная простота алгоритмов анализа полученных системой данных, что позволяет получить гораздо более точную информацию об объекте, чем можно получить с помощью пассивных датчиков.

Системы называются пассивными, если датчики в них не имеют прямого вида информации о местоположении и движении исследуемого объекта, но считывают некоторую основную информацию, например видеоизображение. Данные о движении объекта получаются путем анализа информации с помощью специального алгоритма. Например, при съемке актера с микрофоном было бы очень трудно считывать данные с помощью большого количества механических и инерциальных датчиков, прикрепленных к лицу человека. В такой ситуации используется изображение с видеокамер. Такие системы, как правило, самые простые с аппаратной стороны, но обработка данных, алгоритмы гораздо сложнее, особенно в тех случаях, когда требуется большая точность результатов [15].

Наиболее важные области, как признается в прибыли или убытке, пассивные системы захвата движения можно разделить на системы с одноразрядными системами распознавания, с распознаванием с использованием искусственного интеллекта. Набор наиболее важных областей, таких как выбор деталей и изображение объекта, которые также могут быть использованы для реконструкции фильма с определенной степенью точности. Кроме того, должно быть выполнено еще одно условие: должна быть возможность четко определить соответствующие поля.

В соответствии с разнообразием различных типов важных областей системы ее можно разделить на три группы:

Системы только с одним отличительным знаком. В этих системах и все важные участки здания обозначены знаками. В этом случае каждый из индикаторов или набор индикаторов представляет собой область, которая может быть однозначно идентифицирована (например, цвет или форма индикатора) и в которой нет необходимости использовать сложные алгоритмы искусственного интеллекта.

Системы этого общего типа. В таких системах соответствующие поля помечаются закладкой, а также в системе с одним экраном. Разница заключается в том, что показатели внешне отличаются друг от друга. Это позволяет значительно расширить возможности использования неограниченных токенов. Кроме того, эта система позволяет значительно упростить аппаратное обеспечение, значительно снизить требования к датчику и дисплею. Однако в такой системе существует очень сложная и дополнительная задача-задача компиляции экрана. В каждом наборе кадров система должна быть однозначно идентифицирована одним из индикаторов, иначе вы не сможете обнаружить движение объекта, даже если у него есть вся информация о местоположении каждого гена в течение заданного времени. [16]

Третий тип систем захвата движения является некоммерческим. Он вообще не использует экран. Основные области которых определяются анализом характеристик каждого из ряда последовательных кадров [17]. Дается идентификация важных областей путем соотнесения относительного положения некоторых неоднородных частей объекта с изображением. Такой анализ, очевидно, требует гораздо более сложных вычислений с использованием искусственного интеллекта, но есть и существенное преимущество: вся информация о структуре здания доступна, поскольку в ней нет элемента, подлежащего замене индикатором. Кроме того, каждая область объекта может стать важной в любое время, и поэтому вам не нужно отмечать ее маркером [18].

1.2 Инерциальная технология захвата движения

Инерциальные системы основаны на миниатюрных инерциальных датчиках (гироскопах и акселерометрах) с алгоритмами слияния датчиков для измерения частоты вращения. Данные о движении передаются по беспроводной сети на виртуальный скелет в программном обеспечении. Данные системы похожи на контроллеры Nintendo Wii, но более чувствительны и более точны. Инерциальные системы практичны, поскольку они портативны и не требуют больших площадей захвата или сложной калибровки. Недостатком является то, что позиционный дрейф, который может со временем усугубиться, если настройка неправильна из-за плохой калибровки [19].



Рисунок 5 – Костюм Xsens MVN

Костюм захвата движения Xsens MVN состоит из 17 модулей инерционного и магнитного датчиков (рис. 5). Данные передаются по беспроводному соединению на портативный компьютер, на котором выполняется и визуализируется обработка. Костюм используется для быстрого и удобного размещения датчиков и кабелей [20].

Система захвата движения MVN состоит из 17 датчиков MTx с двумя мастерами Xbus. MTx представляет собой инерциальный и магнитный измери-

тельный блок и включает в себя 3D-гироскопы, 3D акселерометры и 3D магнитометры (38 x 53 x 21) мм, 30 г. Сенсорные модули соединены цепочкой последовательностей с мастерами Xbus, что означает, что существует только одна кабель ведущие к каждой конечности. Xbus Master синхронизирует все выборки датчиков, обеспечивает питание датчиков и управляет беспроводной связью с ПК или ноутбуком. Для быстрого и удобного размещения датчики и кабели встроены в костюм из лайкры и мастера Xbus установлены на спине.

Общий вес системы (включая 8 батареек типа AA) составляет 1,9 кг. Сенсорные модули размещаются на ступнях, голених, верхней части ног, тазу, плечах, груди, голове, предплечьях, передних руках и кистях рук (см. Рис. 5).

Система захвата движения MVN полностью мобильна и состоит только из датчиков, надетых на тело. Система уникальна в своем подходе к оценке изменений ориентации и положения сегментов тела путем интеграции сигналов гироскопа и акселерометра, которые постоянно обновляются с использованием биомеханической модели человеческого тела. Это позволяет отслеживать динамическое движение. Облегчая ограничения модели, в частности, сегменты соединяются шарнирами, кинематика сегментов тела корректируется на дрейф и другие ошибки. Система работает в режиме реального времени с максимальной частотой обновления для всей кинематики 120 Гц. С помощью программного обеспечения MVN Studio пользователь может легко наблюдать, записывать и экспортировать движения в 3D. В этой статье описывается конструкция, принципы работы, форматы вывода и технические особенности MVN [21].

2. АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РЕШЕНИЯ ЗАДАЧИ

2.1 Разработка алгоритма анимации 3D-модели человека

2.1.1 Существующие алгоритмы инерциального захвата движения

Для обобщенной системы захвата движения для обработки данных процесс можно разбить на несколько общих шагов:

- инициализация системы;
- отслеживание наиболее важных областей;
- обнаружение положения и формы.

Начальная фаза инициализации системы включает в себя следующие задачи: установка и калибровка датчиков, установка системы и данные о первоначальном положении исследуемого объекта и датчика, инициализация начальной установки объекта и, при необходимости, создание таблицы прав и ограничений, с помощью которой система анализирует информацию, полученную на заключительном этапе работы. Граф свобод и ограничений при использовании в системе определяет степень свободы, и границы изменений, эти степени свободы для каждой части скелета модели.

Начальная поза формируется следующими способами. Либо заданная заранее поза модели, когда захват движения начинается только после того, как поза модели будет соответствовать предустановленной начальной позе. Либо принимается любая начальная поза, но система отслеживает несколько первых кадров для калибровки, чтобы определить, что данные корректны. [22]

Существуют большое количество алгоритмов применяемых для отслеживания важных областей. Все они различаются методом решения следующих задач: пошаговой дифференциации областей, минимизации собранной информации и отслеживания изменений. Что касается этих этапов, то они зависят от того как представлены данные, и они различаются почти во всех вариантах реализации. Если система применяет однозначно дифференцируемые маркеры или

даже активные датчики, то на этом этап отслеживания важных областей завершает свою работу. Однако возможна ситуация, когда применяются однотипные маркеры, либо корреляционное получение данных. В таких случаях в этап трекинга добавляется дополнительная подзадача. Эта подзадача заключается в сопоставлении важных областей друг другу в полученных в разное время кадрах информации.

Если объект не имеет способности к свободной трансформации, то всё что остается сделать это сопоставить положения важных областей модели объекта, и будут получены форма и положение. Примером такого объекта может быть, к примеру, стол. Положение каждой его части относительно всех остальных постоянно. Но человек является более сложным объектом, который в своем локальном пространстве может изменять форму. Для таких объектов этап распознавание формы происходит сложнее [23].

Алгоритм XSENS MVN. Чтобы осуществить захват движения человеческого тела в полевых условиях без необходимости использования внешних излучателей или камер, доступно несколько систем. В механических трекерах используются жесткие или гибкие гониометры, которые носит пользователь. Эти устройства измерения угла предоставляют данные об угле сустава кинематическим алгоритмам, которые используются для определения положения тела. Присоединение рычажных механизмов к телу, а также установка гониометров представляют несколько проблем. Мягкие ткани тела позволяют изменять положение связей относительно тела при движении. Даже без этих изменений выравнивание гониометра с суставами тела затруднено, особенно для суставов с несколькими степенями свободы [24].

Использование инерциальных датчиков стало обычной практикой в анализе движений. Для точной и свободной от дрейфа оценки ориентации сообщалось о нескольких методах комбинирования сигналов от трехмерных гироскопов, акселерометров и магнитометров. Акселерометры используются для определения направления местной вертикали путем измерения ускорения свободного падения. Магнитные датчики обеспечивают стабильность в горизонтальной

плоскости, определяя направление магнитного поля Земли как компас. Данные от этих дополнительных датчиков можно использовать для устранения дрейфа путем непрерывной коррекции ориентации, полученной путем интеграции данных датчика скорости [25].

Используя рассчитанные ориентации отдельных сегментов тела, измерения и знания о длинах сегментов, вращения между сегментами могут быть оценены, а положение сегментов может быть получено при строгих предположениях связанной кинематической цепи. Этот метод предполагает шарнирное твердое тело, в котором суставы имеют только вращательные степени свободы. Однако человеческое тело и его суставы не могут быть смоделированы как чистая кинематическая цепь с четко определенными соединениями, такими как шарнирные соединения и шарнирные соединения. Каждый сустав человека допускает некоторую расслабленность во всех направлениях (как положения, так и ориентации), кроме его основного направления движения. Более того, чтобы иметь возможность точно отслеживать сложные человеческие суставы и нежесткие части тела, такие как спина и плечо, требуется более трех степеней свободы, определяемых измерением ориентации. Более того, что важно, при захвате движения только с ориентацией невозможно проанализировать просвет обеих ног, который возникает во время бега или прыжков. Используя этот подход, также невозможно точно определить смещение тела относительно системы координат, не прикрепленной к телу [26].

Чтобы обеспечить полное отслеживание сегментов тела с шестью степенями свободы с подключенными модулями инерционных датчиков, ориентацию и положение каждого сегмента тела можно оценить, соответственно, путем интегрирования данных гироскопа и двойного интегрирования данных акселерометра во времени. Однако из-за присущего дрейфа интегрирования эти нескорректированные оценки являются точными только в течение нескольких секунд. Комбинируя инерционные оценки с другими носимыми на теле вспомогательными системами, такими как акустический или магнитный трекер, можно предотвратить несвязанный интеграционный дрейф [27].

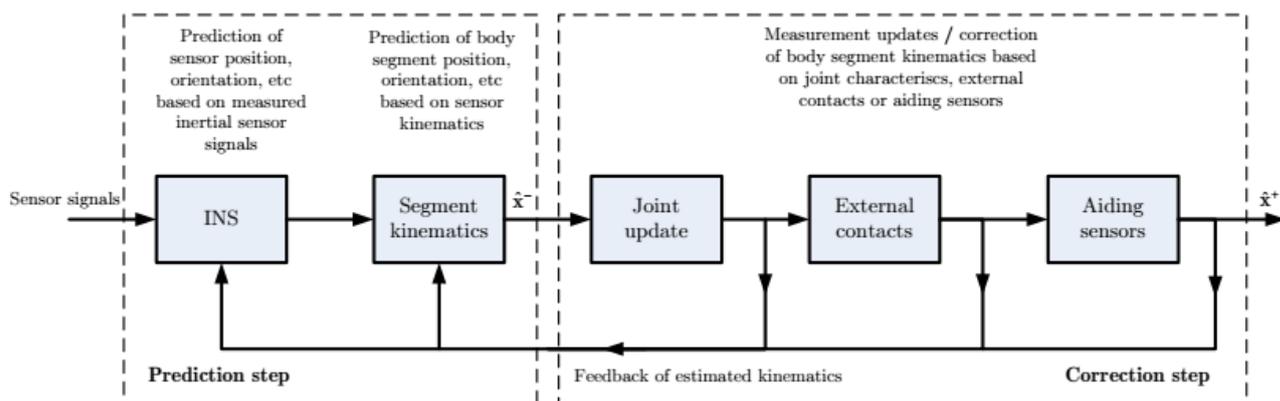


Рисунок 6 – Схема слияния датчиков Xsens MVN

Схема слияния датчиков представлена на рисунке 6. На этапе прогнозирования кинематика датчика рассчитывается с использованием алгоритмов инерциальной навигации на основе измеренных ускорений и угловых скоростей. Используя биомеханическую модель, кинематика сенсора преобразуется в кинематику сегмента. На этапе коррекции к сегментам применяются совместные обновления с последующим обнаружением контактов точек с внешним миром. По желанию, в схему объединения датчиков могут быть включены другие вспомогательные датчики. После всех этапов коррекции оцененная кинематика возвращается на соответствующий этап прогнозирования.

Данный алгоритм подразумевает следующие этапы работы системы: установка системы, калибровка, инерциальный захват движения, кинематика сегментов, обновление положения суставов, обнаружение контактов, позиционная помощь [28].

При прикреплении датчиков к телу начальная поза между датчиками и сегментами тела неизвестна. Кроме того, оценка расстояний между сегментами тела затруднена численным интегрированием ускорения из-за неизвестного начального положения. Поэтому процедура калибровки должна быть выполняться, в котором определяется ориентация датчика на корпус и размеры корпуса. Поскольку сигналы датчиков и биомеханическая модель могут быть описаны, они могут быть включены в схему слияния датчиков с шагом прогнозирования и коррекции (см. Рис.6). На этапе прогнозирования все сигналы датчиков обра-

батываются с помощью так называемой инерциальной навигационной системы алгоритмы. За этим следует прогноз кинематики сегмента с использованием известного датчика выравнивания тела и биомеханической модели тела [29].

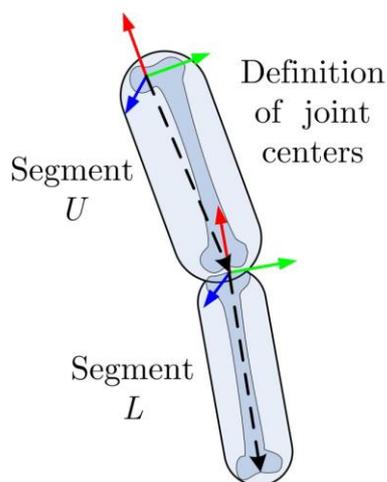


Рисунок 7 – Определение осей сегментов и определение длин сегментов.

Со временем интеграция данных инерциальных датчиков приводит к ошибкам дрейфа из-за наличия шума датчика, смещения сигнала датчика или ошибок ориентации датчика (рис. 7). Для корректировки оценочных величин, таких как ориентация, скорость и положение, схема слияния датчиков постоянно обновляет оценки.

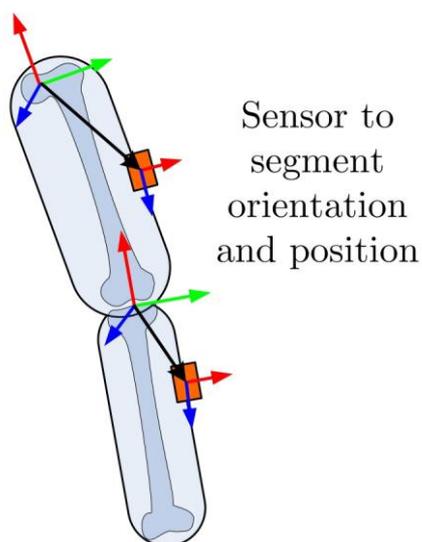


Рисунок 8 – Датчик для выравнивания сегментов

Этап коррекции включает в себя обновление на основе биомеханических

характеристик человеческого тела, в частности суставов, обнаружение точек контакта тела с внешним миром, которые ограничивают глобальное положение и скорость, и, необязательно, другие вспомогательные датчики (рис. 8).

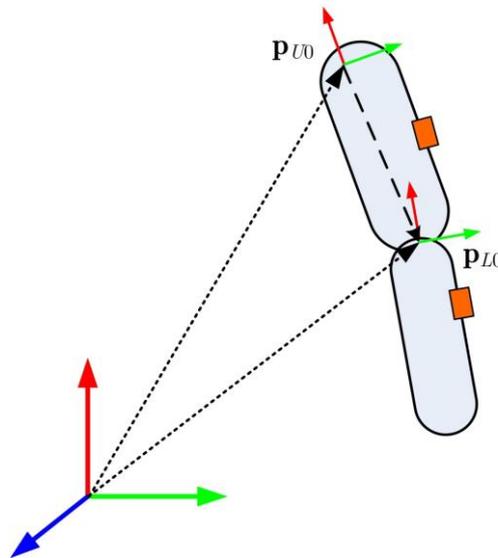


Рисунок 9 – Соотношение двух соединительных сегментов

Оцененная кинематика возвращается в алгоритмы инерциальной навигационной системы и сегментный кинематический шаг, который будет использоваться в следующем временном интервале (рис. 9).

Инерциальный захват движения осуществляется следующим образом. Скоростные гироскопы измеряют угловую скорость и, если они интегрированы с течением времени, обеспечивают изменение угла (или ориентации) относительно первоначально известного угла, где кватернион, описывающий вращение от датчика в глобальный кадр в момент времени, что является кватернионным представлением угловой скорости. Линейные акселерометры измеряют вектор ускорения a и гравитационное ускорение в координатах датчика. Сигналы датчика могут быть выражены в глобальной системе отсчета, если известна ориентация датчика. После удаления гравитационной составляющей ускорение может быть интегрировано один раз в скорость и два раза в положение, все в глобальном кадре [30].

Захват движения с помощью датчиков IMU. В исследовании описан недорогой и простой в настройке для многих приложений костюм захвата инерционного движения. Костюм основан на датчиках IMU и включает в себя программное обеспечение промышленного класса, которое позволяет получать, визуализировать и анализировать данные о движении человека [31].

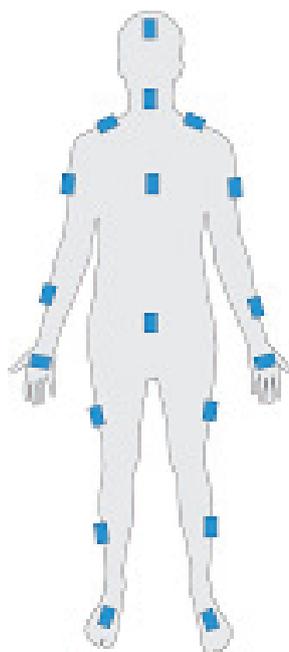


Рисунок 10 – Костюм с 18 датчиками

Система захвата инерциального движения основана на сенсорной сети тела, где датчики IMU подключены к каждому сегменту, который должен отслеживаться. Датчики вшиваются в костюм с помощью растягивающейся проводящей нити (рис. 10). Связь определяется протоколом связи. Отображение ориентации ИДУ на конкретные сегменты скелета позволяет фиксировать движение объекта. Зная все ориентации сегментов с течением времени, можно отслеживать общую позу. Схема конвейера обработки данных представлена на рисунке 11. В конвейере обработки данных все ориентации (обозначаемые как Q) представлены единичными кватернионами. [32]

Отслеживание движения всего скелета с помощью инерциальной сенсорной системы возможно благодаря двум основным алгоритмам: калибровке и определению ориентации одного сегмента. Чтобы найти относительное враще-

ние между датчиком и сегментом QS, датчик должен быть подвергнут этапу статической калибровки. Калибровка основана на общеизвестной T-позе [33].

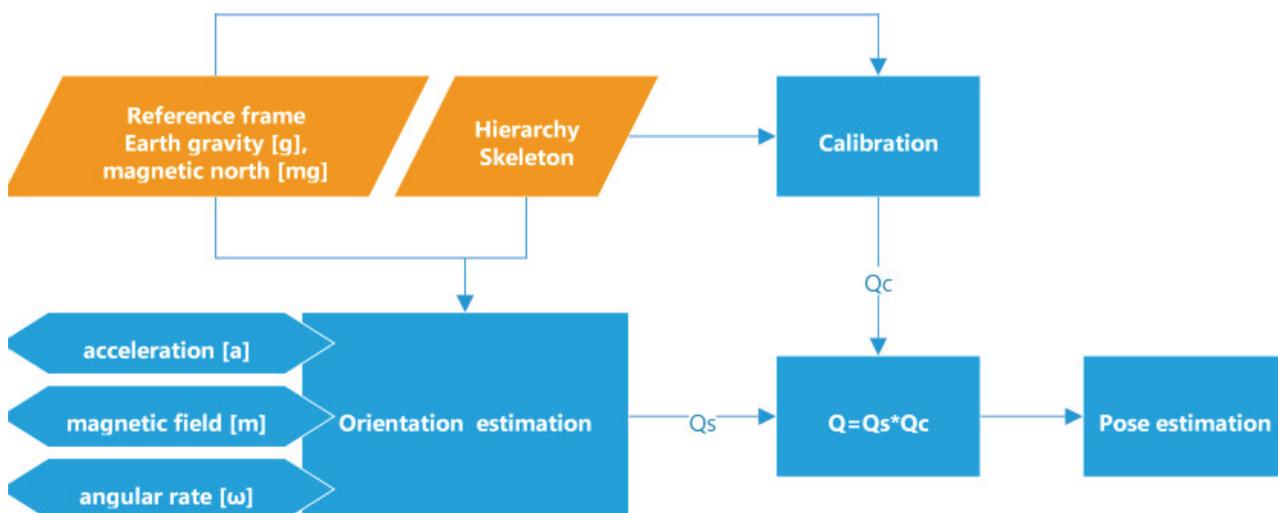


Рисунок 11 – Конвейер обработки данных.

Для оценки ориентации QS датчика авторы реализовали и протестировали набор алгоритмов, описанных в литературе. Наилучшие результаты были получены для расширенного фильтра Калмана, адаптированного для преодоления эффекта внешнего линейного ускорения [34]. Алгоритм был дополнительно расширен функциональностью, использующей кинематические зависимости для кинематических цепей (для верхних и нижних конечностей) [35].

Для оценки костюма в качестве эталона использовалась форма данных оптической системы захвата движения (Vicon). Для обеспечения информативного сравнения потоков данных ориентации с различными системами отсчета и измеренных в соответствии с отдельными таймерами с одинаковой частотой, данные должны быть нормализованы. Авторы исследования использовали алгоритм, основанный на PCA, для временного согласования движения, представленного в виде временного ряда угловой ориентации, полученного с помощью различных систем [36].

Захват движения с помощью Microsoft Kinect. В этих обстоятельствах мы используем набор из двух кинектов Microsoft (MS) для Windows, которые захватывают сцену достаточно подробно для последующей обработки. Мы мо-

жем просто перевезти две небольшие коробки, и калибровка займет несколько минут. Весь алгоритм процесса захвата движения показан на рисунке 12. Как показано в исследовании, представленном в публикации, использование только одного MS Kinect имеет низкую точность и не позволяет фиксировать движения, видимые сбоку, когда используется только передний MS Kinect [37].

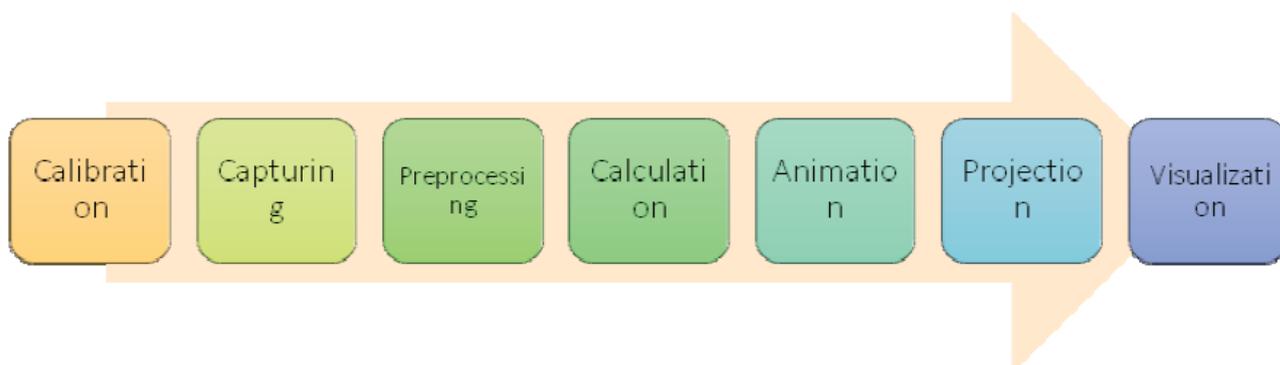


Рисунок 12 – Процесс захвата движения с помощью Microsoft Kinect

Процесс захвата движения с помощью Microsoft Kinect начинается с калибровки системы. После чего происходит захват движения, затем осуществляется обработка полученных данных с последующим вычислением координат. Далее осуществляется анимация модели и проецирование координат. Заключительным этапом является отображение конечного результата.

2.1.2 Предлагаемый алгоритм инерциального захвата движения

Основной алгоритм костюма инерциального захвата движения представлен на рисунке 13.

Обработка данных на аппаратной платформе Arduino осуществляется следующим образом. Происходит обработка потока сырых данных с датчиков абсолютной ориентации BNO055. Если данные с определенного датчика не изменяются, то координаты датчика остаются неизменными до момента изменения данных, и посылается неизменный кватернион. Если данные с определенного датчика изменяются, то вычисляется новые координаты и пересылается обновленный кватернион.

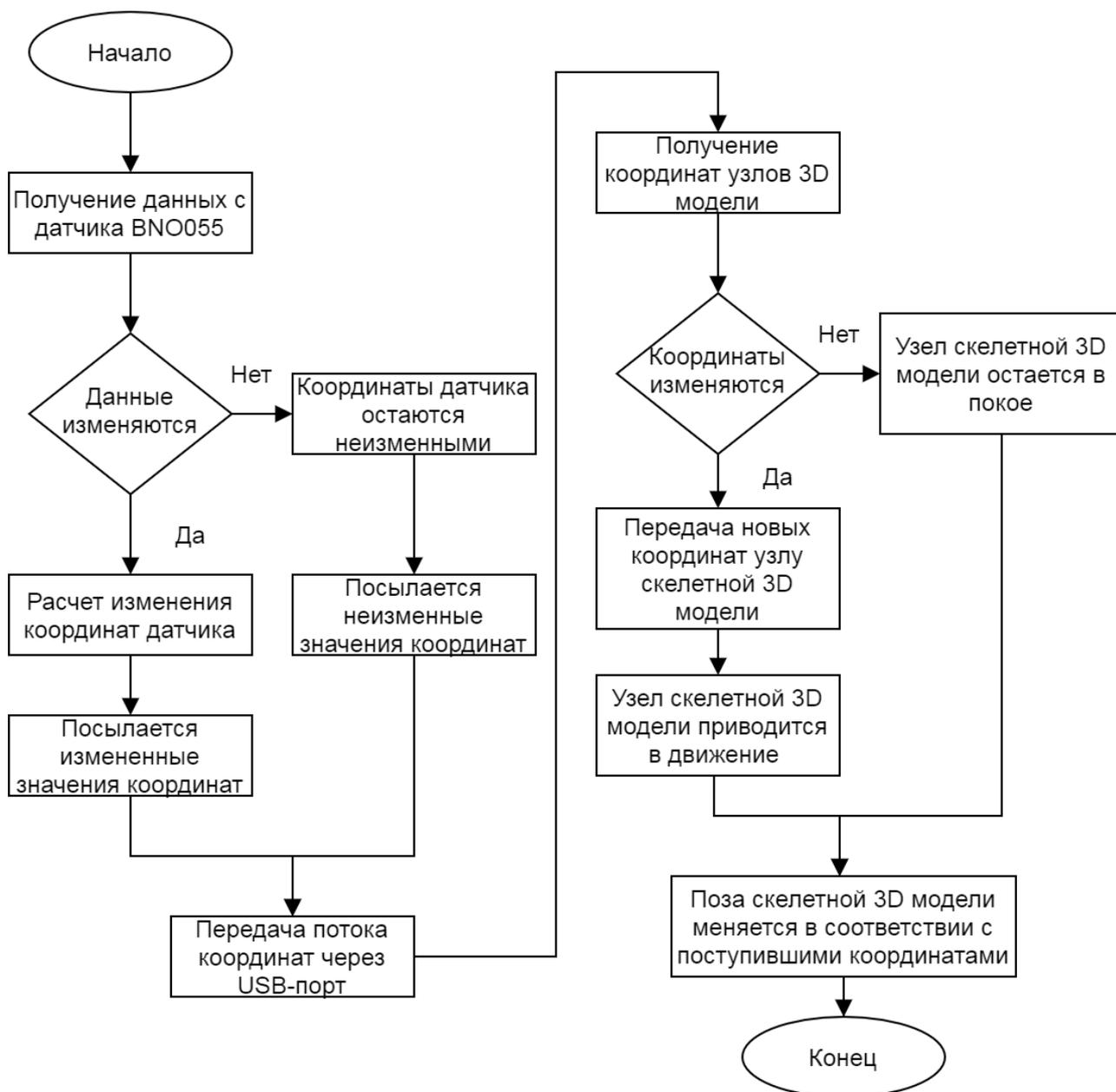


Рисунок 13 – Основной алгоритм костюма инерциального захвата движения

Обработка потока данных для 3D-редактора Blender осуществляется следующим образом. Если поступает поток измененных данных в виде кватернионов, то выполняется расчет изменения координат узлов на скелетной 3D модели человека. Если данные поступают неизменными, то пересчет координат не осуществляется.

Обработка данных в Blender Game Engine осуществляется следующим образом. Если поступают измененные координаты, то осуществляется передача измененных координат точкам на скелетной 3D-модели человека, благодаря

чему узел модели приводится в движение. Если поступают неизменные координаты, то узел модели остается в покое.

2.2 Обзор технического обеспечения реализуемого алгоритма

2.2.1 Система захвата движения XSENS MVN Animate

XSENS MVN Animate – это «инерционная система захвата движений (inertial motion capture), оптимизированная под создание анимации для кино, компьютерных игр, анимационных фильмов и живых выступлений в режиме реального времени с использованием компьютерной графики. Система состоит из двух компонентов: программного обеспечения MVN Animate и аппаратной части - костюмов Xsens Link и Xsens Awinda.» [38].

Xsens Link представляет собой лайкровый костюм, на котором зафиксированы датчики захвата движения (рис. 14 а). Xsens Awinda представляет собой комплект из легких беспроводных датчиков, которые крепятся на теле актера специальными ремешками-липучками (рис. 14 б) [39].

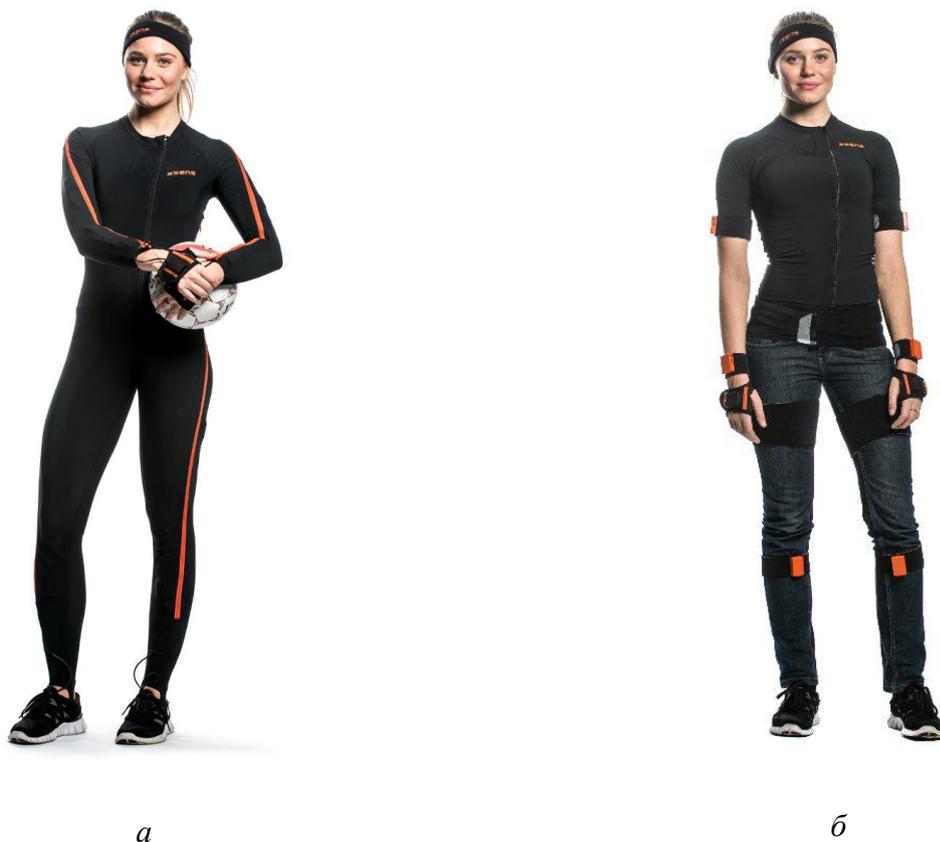


Рисунок 14– Костюмы Xsens Link (а) и Xsens Awinda (б)

Технические характеристики костюма Xsens Link [40]:

- вариант: лайкровый костюм;
- трекеры: 17 проводных + модуль передачи данных;
- качество данных: сразу в продакшн;
- скорость установки: быстрая;
- время отклика: 20 мс;
- способ связи: одна точка доступа WiFi на несколько костюмов;
- батарея: одна на костюм;
- время работы батареи: 10 часов;
- дальность действия: 50 метров в помещении, 150 метров на улице;
- буферизация: 10 минут;
- частота обновления (внутренняя): 1000 Гц;
- качество записи: 240 кадров/сек.

Технические характеристики костюма Xsens Awinda [41]:

- вариант: ремешки-липучки;
- трекеры: 17 проводных;
- качество данных: сразу в продакшн;
- скорость установки: очень быстрая;
- время отклика: 30 мс;
- способ связи: одна радио станция на каждый костюм;
- батарея: своя в каждом трекере;
- время работы батареи: 6 часов;
- дальность действия: 20 метров в помещении, 50 метров на улице;
- буферизация: 30 секунд;
- частота обновления (внутренняя): 1000 Гц;
- качество записи: 60 кадров/сек.

Стоимость: по запросу коммерческого предложения.

2.2.2 Система захвата движения Motionshadow

Motionshadow – инерциальная система захвата движения состоящая из

комплекта проводных датчиков, которые крепятся на теле актера эластичными ремешками, и управляющего устройства, которое крепится на пояс человека (рис. 15) [42].



Рисунок 15 – Motionshadow

Технические характеристики системы [43]:

датчики (каждый датчик представляет собой прецизионный инерциальный измерительный блок (IMU), состоящий из акселерометра с высоким разрешением, малошумящего акселерометра, гироскопа и магнитометра):

- 17 прецизионных инерциальных измерительных блоков;
- 3-осевой акселерометр, гироскоп и магнитометр;
- 2 пары стелек для измерения давления;
- частота дискретизации: выходная скорость передачи данных 100, 200 или 400 Гц, внутренняя частота обновления 1000 Гц;

- ориентация: без дрейфа, трехмерное вращение;
- статическая точность 0,5°;
- 2° динамическая точность;
- размер: миниатюрные датчики (33 x 18 x 6) мм;
- масса: 1,1 кг пригодный для носки, дорожный кейс 6,5 кг;
- аккумулятор: 8 часов, емкость 10 А/ч, 2 упаковки и двойное зарядное устройство;
- беспроводная связь: Wi-Fi 802.11n, внутренний диапазон 32 м, наружный диапазон 95 м;
- контроллер: 32 Гб памяти, светодиод состояния;
- задержка: 20 мс беспроводной (Wi-Fi), 5 мс проводное (USB).

Стоимость: \$4000 [44].

2.2.3 Система захвата движения Rokoko Smartsuit Pro

Rokoko Smartsuit Pro – инерциальная система захвата движения, представляющая из себя костюм из эластичной нейлоновой ткани с регулируемыми стяжными ремнями и интегрированными каналами для защиты всех электронных компонентов (рис. 16) [45].



Рисунок 16 – Rokoko Smartsuit Pro

Технические характеристики системы:

- датчики: 19 шт. 9-DoF IMU;
- концентратор: сбор всех данных отслеживания с датчиков, их объединения и потоковая передача данных на компьютер в реальном времени;
- батарея и питание: внешний портативный аккумулятор USB;
- беспроводная связь: 2,4 ГГц и 5 ГГц - 802.11ac Wi-Fi;
- кадров в секунду: 100;
- задержка: около 20 мс на стандартном Wi-Fi-роутере;
- дальность действия: до 100 м.

Стоимость комплекта (костюм и программное обеспечение): начиная от 2495\$ [47].

2.3 Обзор программного обеспечения реализуемого алгоритма

2.3.1 Коммерческие программные продукты

XSENS MVN Animate PRO – коммерческое программное обеспечение для осуществления захвата движений с помощью костюмов Xsens Link и Xsens Awinda, оптимизированное для использования в играх, фильмах, анимации и развлечениях (рис. 17) [48].

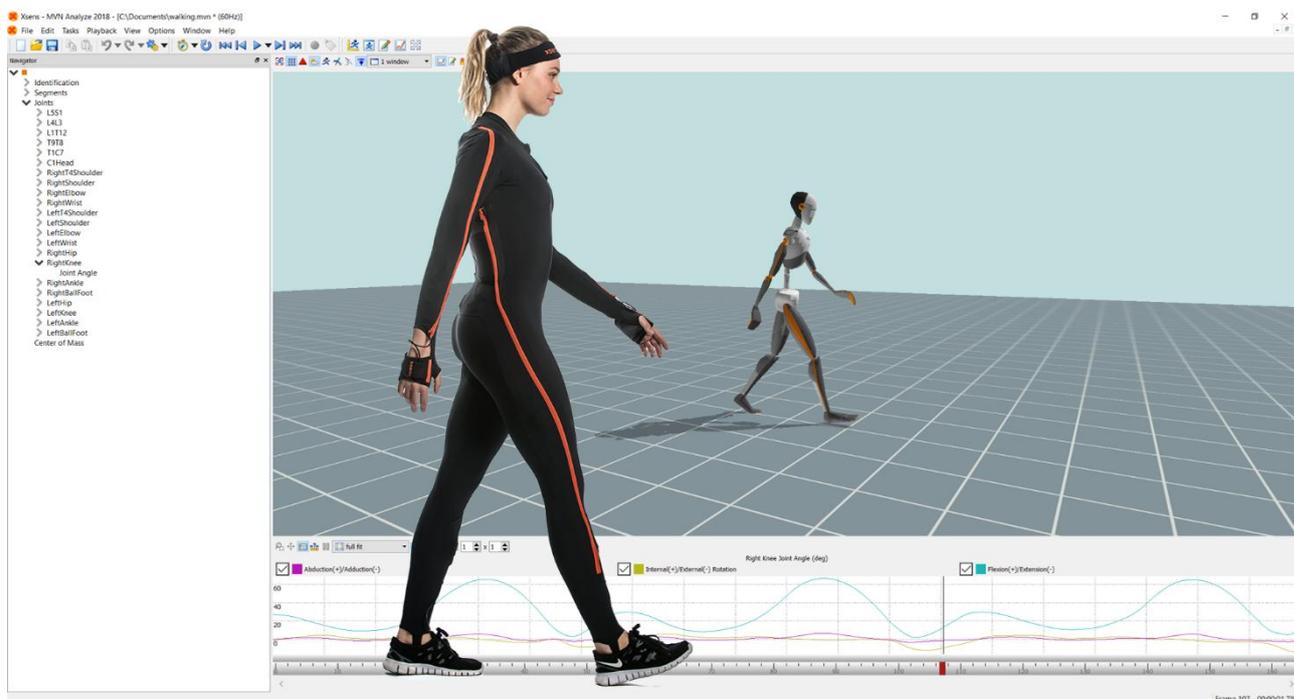


Рисунок 17 – Пример использования XSENS MVN Animate PRO

Основные функции [49]:

- работа одновременно с несколькими актерами;
- различные сценарии использования;
- магнитный иммунитет;
- поддержка внешней камеры;
- Animate Engine Reprocessing;
- Reprocessing HD;
- процесс калибровки: не более 5 минут;
- визуализация в режиме реального времени – просмотр движений в реальном времени, также есть возможность подключения референс-камеры;
- автоматическая чистка полученной анимации – функция постобработки полученных данных в оффлайн режиме, что позволяет добиться максимального качества финальной анимации;
- экспорт анимации в форматы .fbx, .bvh и .mp4 (превью), а также батч-экспорт;
- передача данных в режиме реального времени в следующие рабочие среды – Autodesk Maya, 3dsMax, Autodesk MotionBuilder, Unity и Unreal Engine 4, Blender или в любой другой программный пакет с помощью функции Network Streaming;
- поддержка внешнего таймкода для синхронизации с остальным оборудованием, используемом при создании анимации.

Стоимость программного обеспечения XSENS MVN Animate PRO: по запросу коммерческого предложения [50].

Motionshadow – коммерческое программное обеспечение для захвата движения с помощью системы Motionshadow (рис. 18) [51].

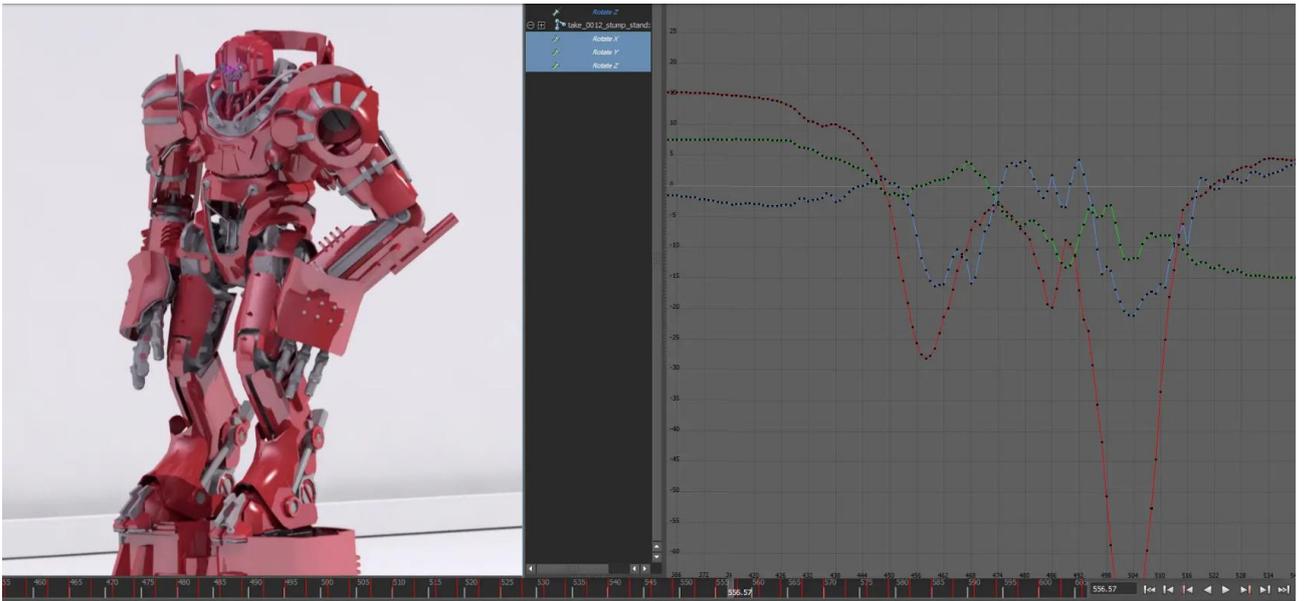


Рисунок 18 – система Motionshadow

Основные функции [52]:

- запись данных на жесткий диск или потоковая передача;
- прямая передача в Maya, MotionBuilder, Unity, и Unreal Engine (например, плагин Shadow Mocap Plugin для Unreal Engine);
- экспорт данных в следующие форматы: Autodesk FBX (*.fbx), Comma Separated Values (*.csv), Biovision Hierarchy (*.bvh), Coordinate 3D (*.c3d);
- импорт данных из файла *.csv в Excel;
- поддерживаемые операционные системы: Windows, macOS, Linux, iOS, Android.
- связь: поддержка устройств с Wi-Fi 802.11n и USB Standard-A;
- требование современного браузера с поддержкой HTML5.

Стоимость программного обеспечения Motionshadow: 4000\$ [53].

Rokoko Studio – программное обеспечение для костюма инерциального захвата движения Rokoko Smartsuit Pro (рис. 19). Программное обеспечение поставляется в комплекте с костюмом [54].

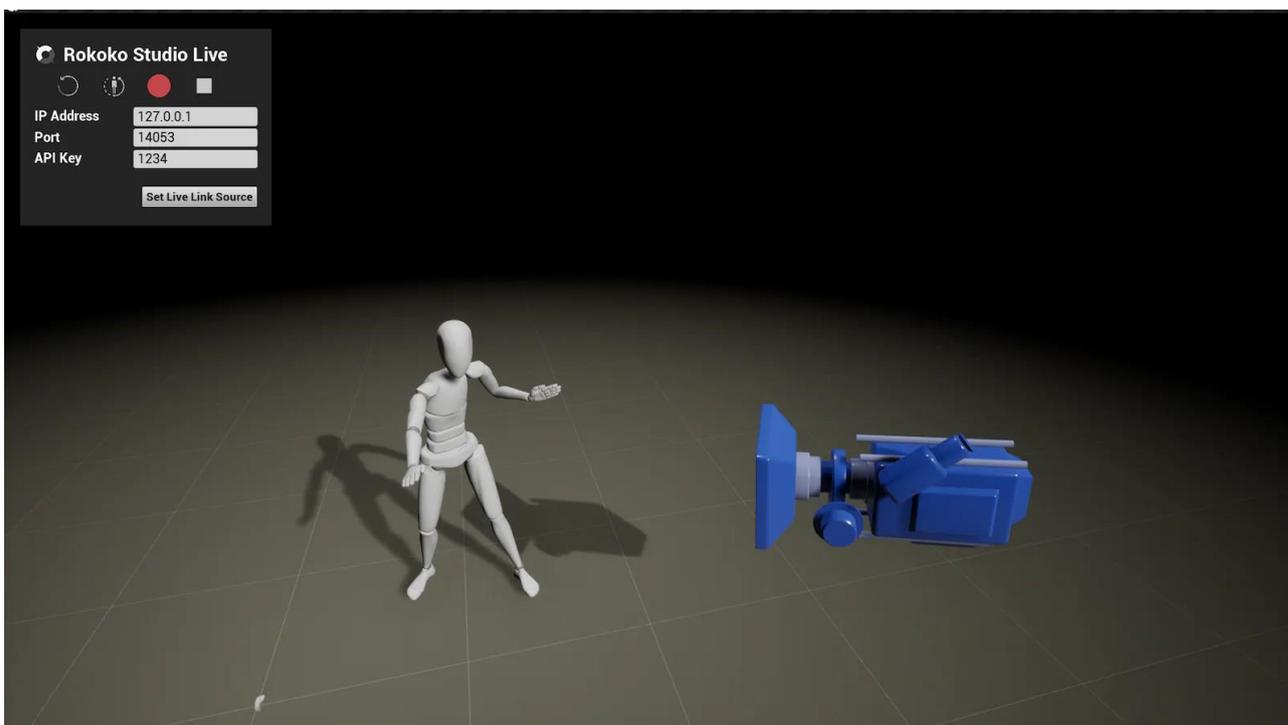


Рисунок 19 – Rokoko Studio

Доступные плагины для прямой передачи данных в следующее программное обеспечение: Unreal, Unity, Blender, Maya, MoBu, Houdini, Cinema4D, iClone.

Стоимость комплекта, состоящего из костюма, программного обеспечения и плагинов: от 2495\$ [55].

2.3.2 Свободно распространяемое программные продукты

Bewegungsfelder – общедоступный бесплатный проект программного обеспечения для мобильной и настраиваемой инерционная система захвата движения для скелетной анимации, состоящий из приложения C # / WPF для захвата и записи скелетной анимации и автономных сенсорных модулей IMU MPU6050 (акселерометр и гироскоп) на основе ESP8266 Wi-Fi SoC (рис. 20) [56].

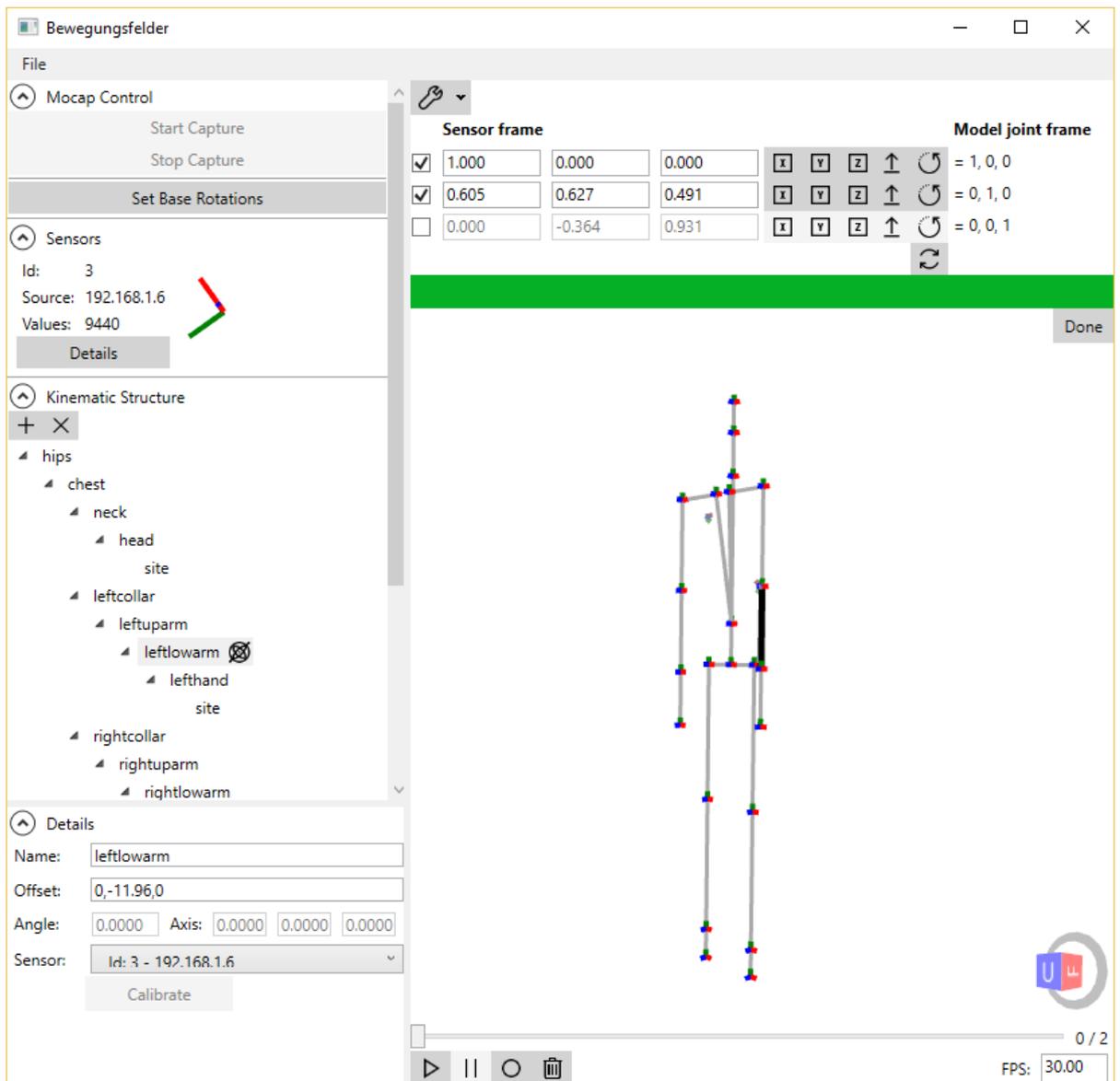


Рисунок 20 — Bewegungsfelde

Основные функции [57]:

- определение гибкого каркаса;
- живая 3D-визуализация датчиков и позы скелета;
- запись/воспроизведение анимации;
- BVH экспорт и импорт;
- UDP-сервер принимает входящие значения датчиков.

Основные недостатки: использование датчиков не абсолютной ориентации и костюм с датчиками и контроллерами необходимо разрабатывать само-

стоятельно из общедоступных компонентов; отсутствует экспорт результатов в распространенные форматы файлов захвата движения.

Motiosuit – общедоступное и бесплатное программное обеспечение для инерциальной системы захвата движения, на основе датчиков абсолютной ориентации BNO055 в каждом соединении с одним или двумя датчиками на шину I2C и мультиплексор I2C для переключения между шинами, подключенный через Bluetooth к модулю моста Bluetooth-USB, который обеспечивает связь между костюмом и компьютером. Данные передаются в 3D редактор Blender, в котором движения обрабатываются в модуле Blender Game Engine (рис. 21) [58].

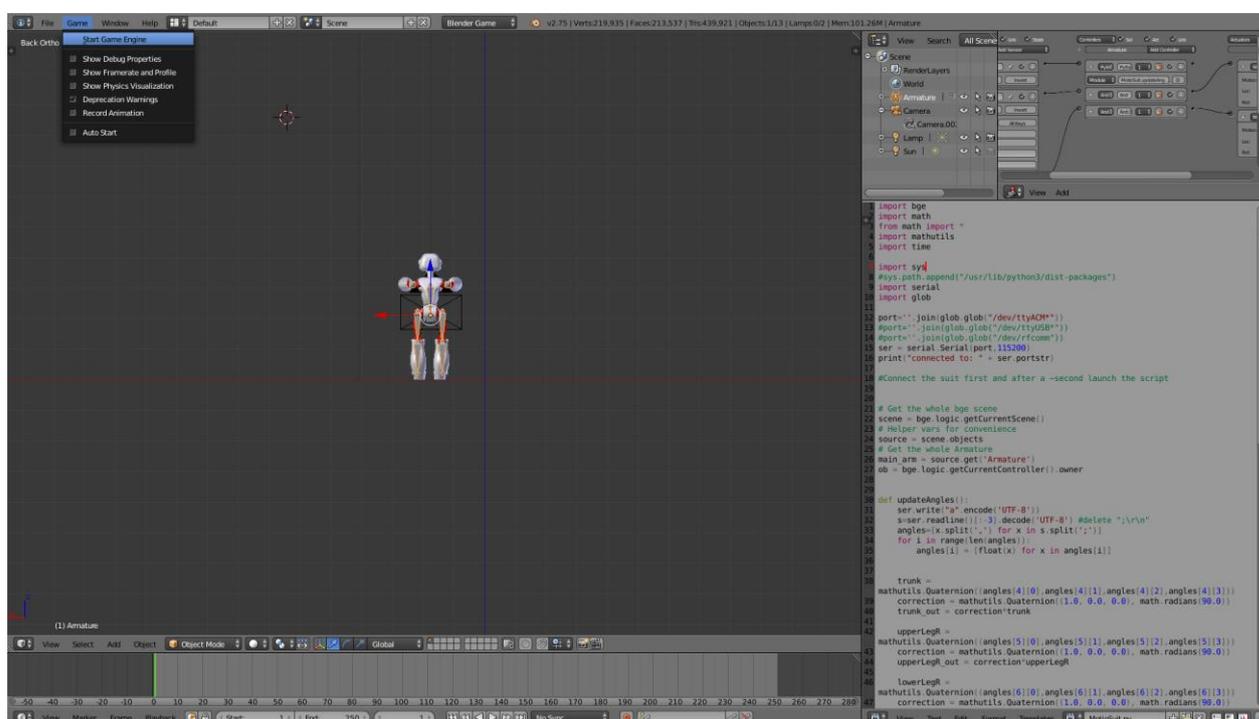


Рисунок 21 – Модуль Motiosuit для Blender Game Engine

Основные недостатки: программное обеспечение требует доработки; костюм с датчиками и контроллерами необходимо разрабатывать самостоятельно из общедоступных компонентов.

Blender – это бесплатный 3D-редактор с открытым исходным кодом (что не маловажно, т.к. коммерческие 3D-редакторы обладают высокой стоимостью). Blender поддерживает весь объем процесса 3D-моделирования (анимацию, моделирование, рендеринг, компонование, захват движения, редактиро-

вание видео и создание игр) [59]. Пользовательский интерфейс Blender представлен на рисунке 22.

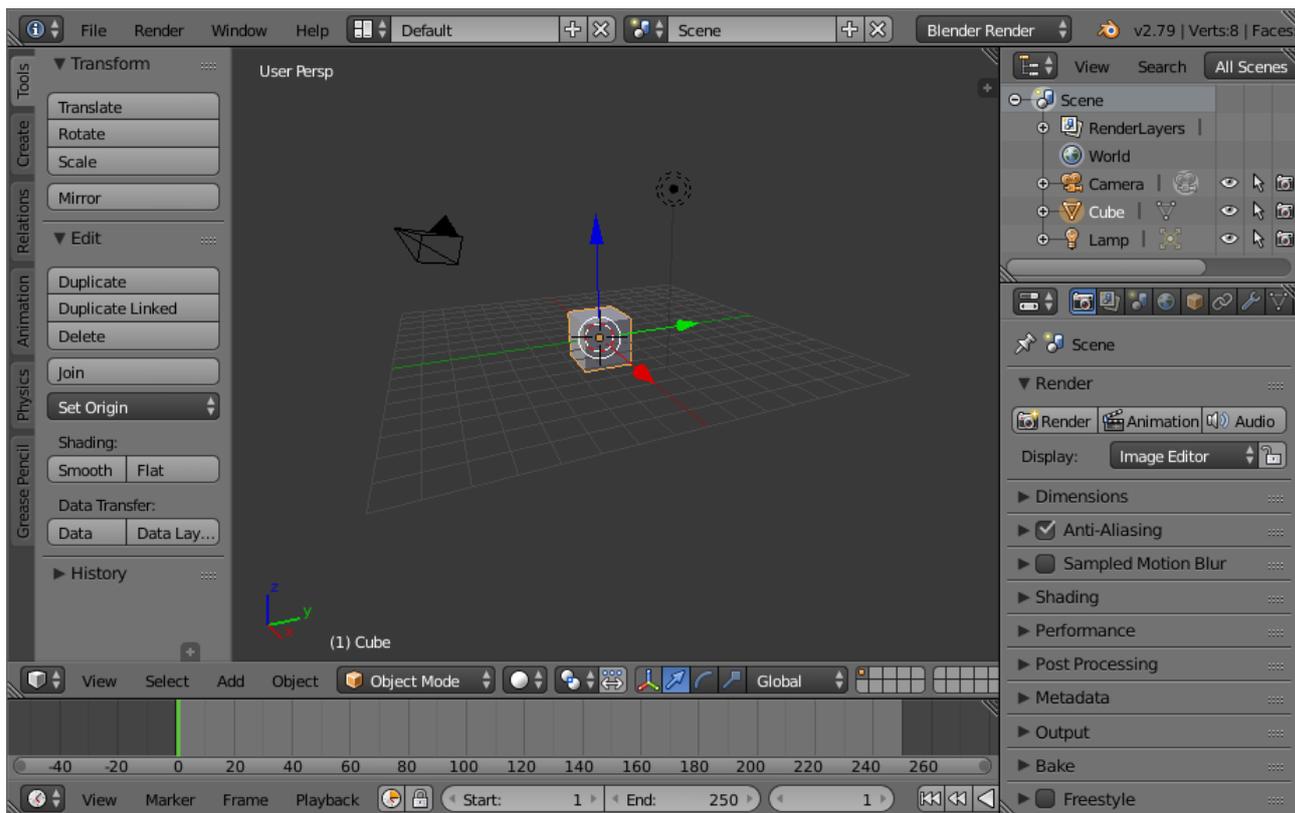


Рисунок 22 – Пользовательский интерфейс Blender

Продвинутые пользователи используют API Blender для сценариев Python для настройки приложения и написания специализированных инструментов; часто они включены в будущие версии Blender. Blender хорошо подходит для частных лиц и небольших студий, которые извлекают выгоду из его унифицированного конвейера и гибкого процесса разработки. Blender является кросс-платформенным и работает одинаково эффективно на компьютерах под управлением операционных систем Linux, Windows, и Macintosh [60].

Blender Game Engine – это компонент Blender, бесплатного и открытого 3D-пакета, используемого для создания интерактивного контента в реальном времени. Игровой движок является независимым компонентом и включает в себя поддержку таких функций, как скрипты на языке Python и OpenAL 3D sound [61].

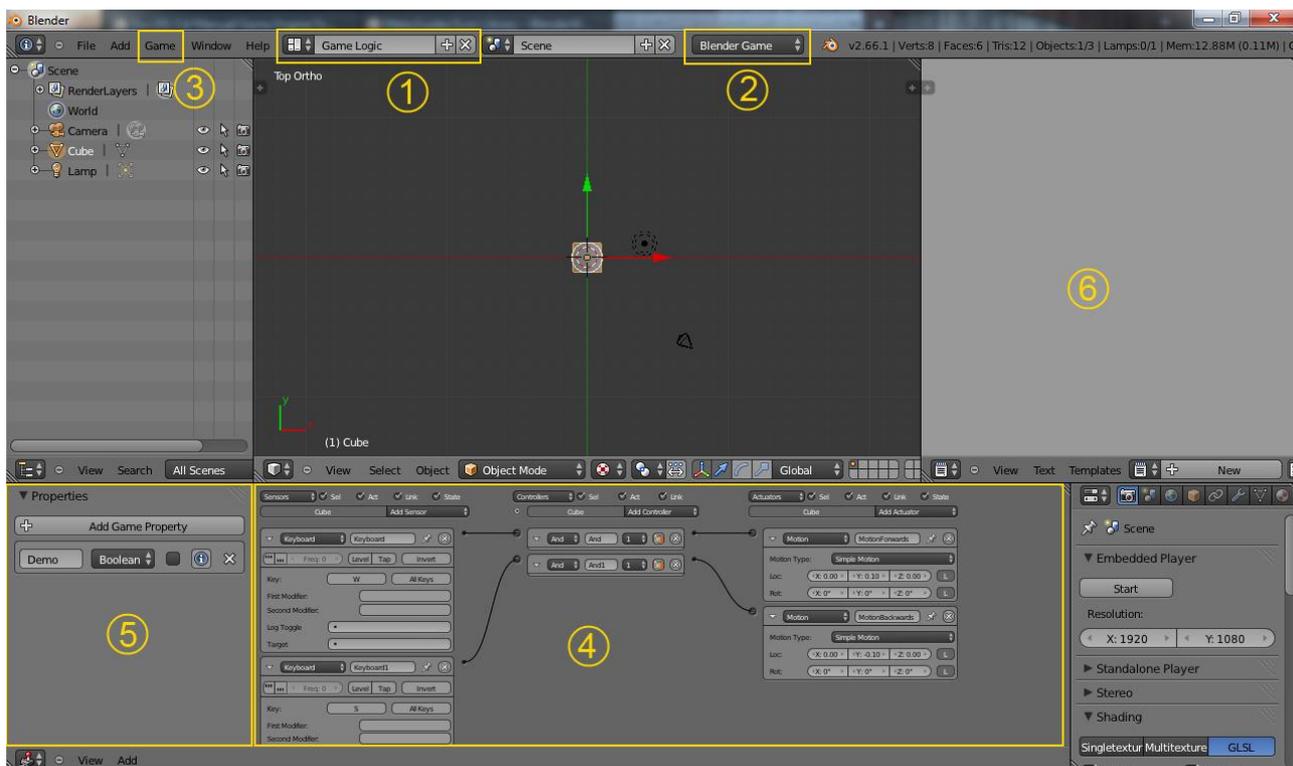


Рисунок 23 – Интерфейс Blender Game Engine

Интерфейс Blender Game Engine включает в себя следующие основные элементы, пронумерованные на рисунке 23 [62].

1) Game Logic – выбирается из списка макетов экрана для различных приложений и включает в себя множество стандартных панелей Blender (информация, 3D-вид, свойства), а также новую панель редактора логики, связанную исключительно с Blender Game Engine.

2) Blender Game – выбирается из меню движка рендеринга. Это указывает, что все выходные данные будут выводиться рендерером игрового движка Blender в реальном времени.

3) Game – данное меню предоставляет различные варианты условий для запуска игрового движка. Меню доступно только в том случае, если движок рендеринга настроен на игру Blender. Start Game – запуск игры в игровом движке (P или Shift-P, когда курсор мыши находится над редактором 3D-вида). Show Debug Properties – показывать свойства, отмеченные для отладки во время запуска игры. Show frame rate and profile – отображение частоты кадров и ин-

формации о профилировании во время запуска игры. Show Physics visualization – отображение визуализации физических границ и взаимодействий. Deprecation warnings – вывод предупреждений при использовании устаревших функций в API Python. Record animation – запись анимации в F-кривых. Auto Start – автоматический запуск игры во время загрузки.

4) Logic Editor panel – область, где логика, свойства и состояния настраиваются для управления поведением объектов в игре. (Панель Logic Editor также можно отобразить, выбрав Logic Editor в меню Display Editor, нажав Shift-F2 или Ctrl-Right).

5) Properties – панель свойств экрана выбирается из главного информационного меню, но несколько разделов панели свойств изменяются при изменении движка рендеринга с Blender Render на Blender Game.

Blender Game Engine использует систему графических «логических кирпичей» (logic bricks) – комбинацию «датчиков» (sensors), «контроллеров» (controllers) и «исполнительных механизмов» (actuators) для управления движением и отображением объектов. Игровой движок также может быть расширен с помощью набора привязок Python [63].

MakeHuman. Разработка трехмерных моделей человека, требующая объемных знаний в области анатомии человеческого тела, а также мастерства в сфере 3D-моделирования, является достаточно трудоемкой задачей. Для того, чтобы сделать процесс моделирования более простым существует достаточно много разных пакетов прикладных программ, имеющих различные функции и предназначение для решения вопросов определенного круга задач.

Программное обеспечение MakeHuman [64], было создано специально для разработки фотореалистичной модели человека без его проектирования вручную. Данный пакет программ применяется в процессе создания моделей специальные шаблоны, именно им, с помощью большого арсенала разнообразных характеристик и параметров, впоследствии придается необходимая геометрическая форма. Потенциал MakeHuman реально увеличить используя плагины и

установку косвенных заготовок (одежды, поз и других моделей человека). Важным преимуществом этого программного продукта является то, что он бесплатный (пользование программой предлагается по лицензии GNU AGPL).

На рисунке 24 представлено основное окно программы. Интерфейс разделен на несколько частей [65].

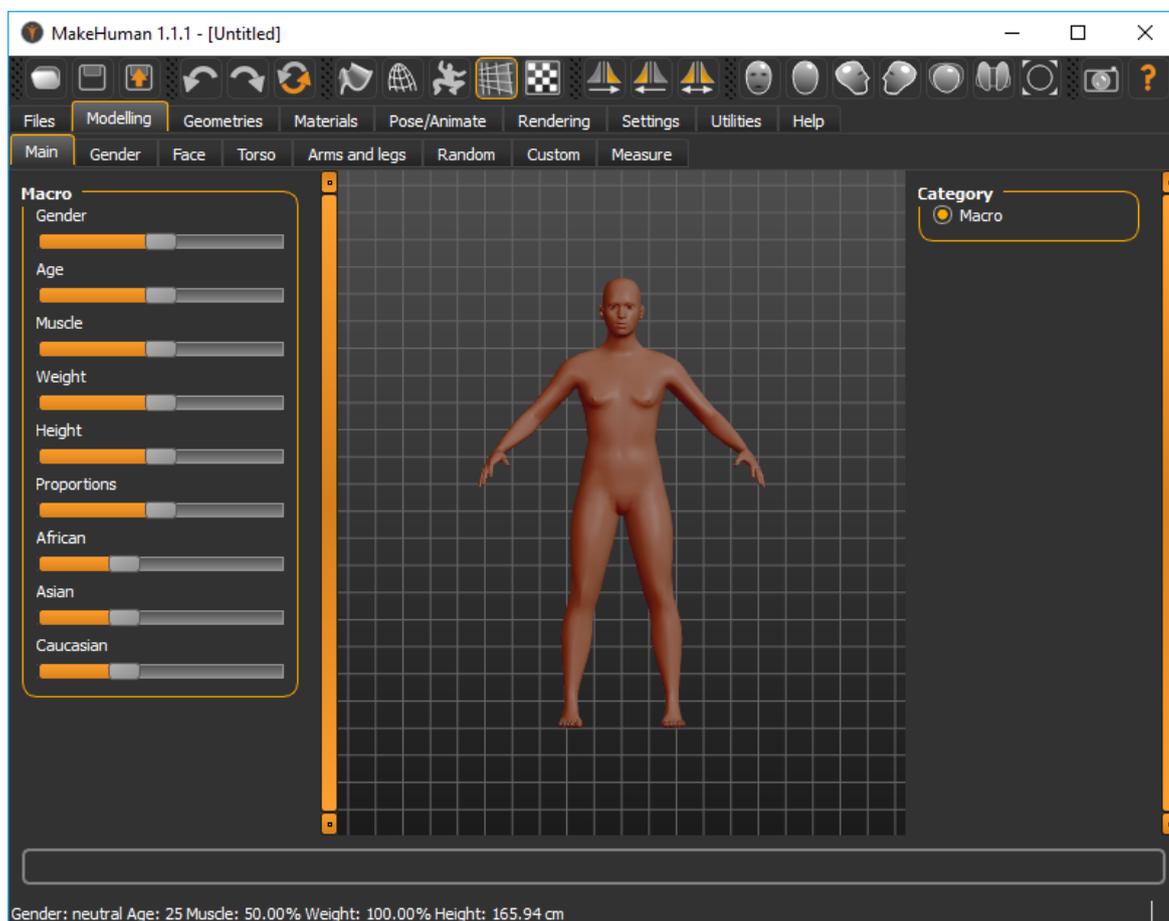


Рисунок 24 – Основное окно программы MakeHuman

В верхней части окна находятся вкладки, отвечающие за переключение параметров, в нижней части находится экран предпросмотра модели. В левой части окна находятся панели, отвечающие за настройки характеристик.

Первая вкладка «Modeling» имеет следующее содержание:

- вкладка «Main» дает возможность настроить основные характеристики модели (рост, вес, возраст и т. д.) В дополнение к этому данная вкладка включает в себя параметры расовых черт: что дает возможность настроить значение

таким образом, что модель может обрести черты различных рас (монголоидной, европейской или негроидной);

- вкладка «Gender» отвечает за первичные половые признаки;
- на вкладке «Face» формируются черты лица модели человека;
- на вкладке «Torso» задаются характеристики туловища модели;
- на вкладке «Arms and legs» задаются характеристики рук и ног модели.

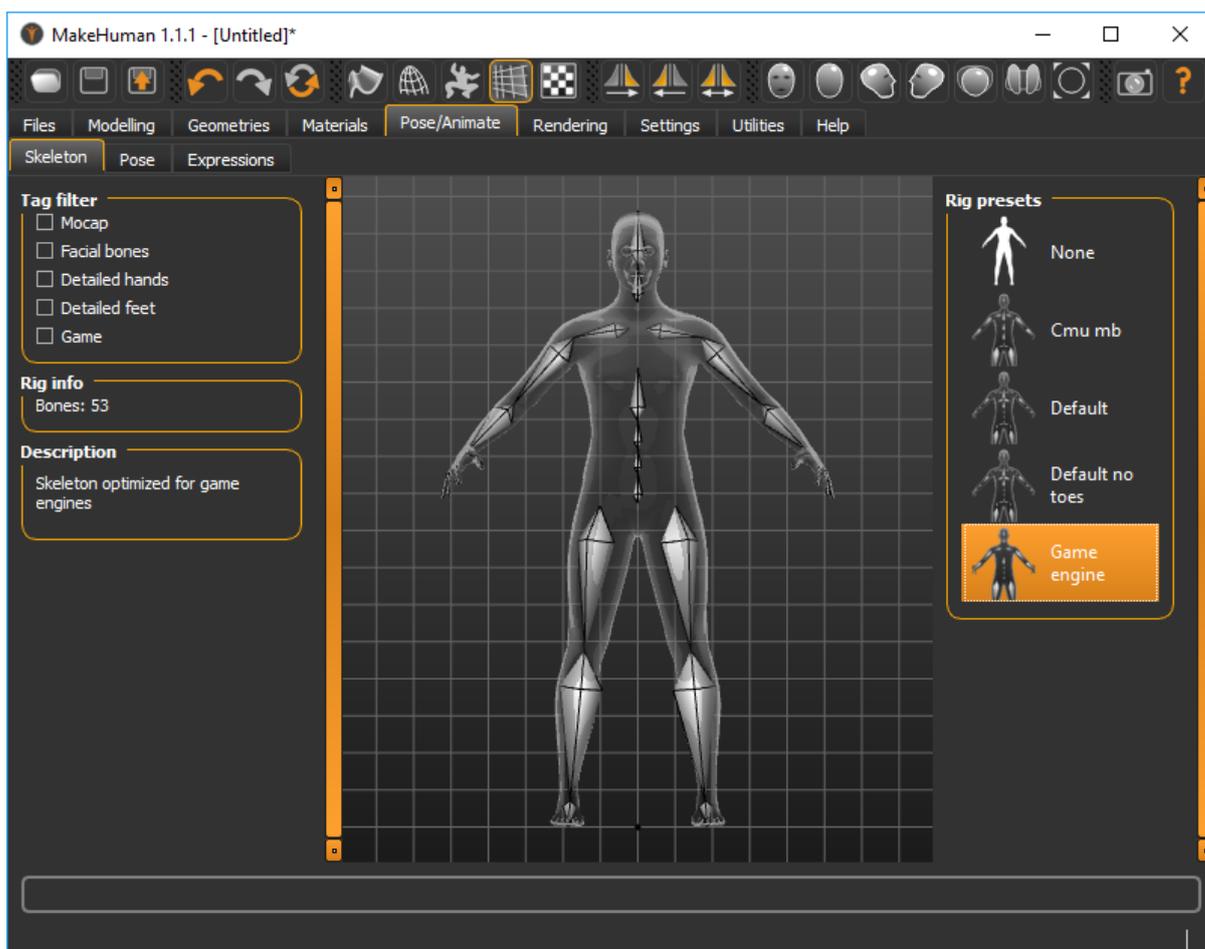


Рисунок 25 – Окно «Pose/Animate»

На вкладке «Pose/Animate» возможно создание скелетной модели «Game engine» для экспорта в программы для создания анимации или компьютерных игр, что применялось в данной работе (рис. 25). Итоговую модель возможно экспортировать в файл 3D-объекта, поддерживаемый различным программным обеспечением для 3D-моделирования и создания игр [66].

2.4 Программно-техническое решение задачи

Программно-техническое обеспечение решения задачи представляет собой совокупность технического и программного обеспечения.

2.4.1 Обоснование выбора электронных компонентов

Предлагается использование аппаратно-программной платформы Arduino, т.к. для обработки данных с датчиков абсолютной ориентации BNO055 (рис. 26) уже существует общедоступная бесплатная библиотека BNO055.h для Arduino. Для обработки данных с I2C-мультиплексоров TCA9548A уже существует библиотека I2CMux.h. Также использование аппаратно-программной платформы Arduino является экономически более целесообразно, по сравнению с другими платформами.

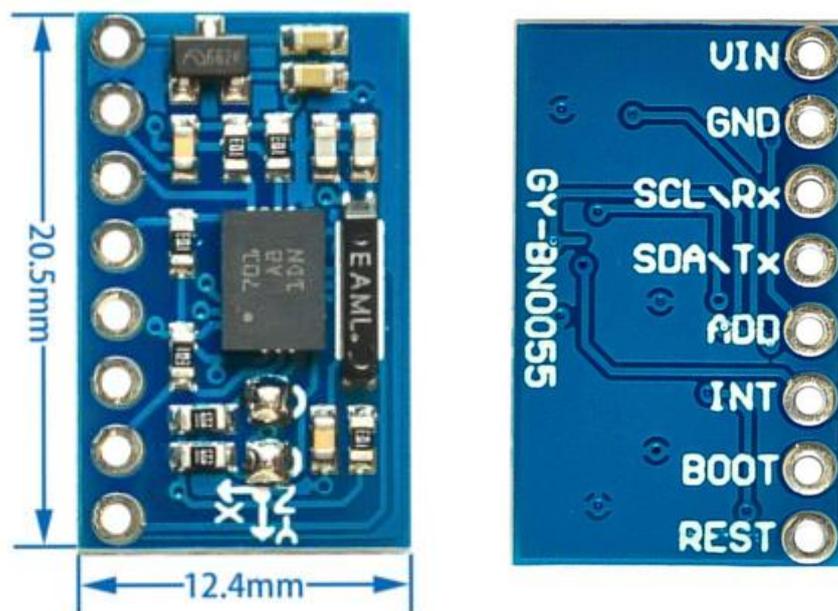


Рисунок 26 – Внешний вид датчика BNO055

Выбор датчика абсолютной ориентации BNO055 обусловлен значительными преимуществами, по сравнению с более дешевыми датчиками-акселерометрами, такими как MPU-6050, в которых все вычисления ложатся на микроконтроллер Arduino, что значительно увеличивает время задержки передачи данных в секунду, что крайне негативно сказывается на качестве анима-

рования персонажей. Тогда как в BNO055 есть свой модуль вычислений – это 32-битное ядро ARM Cortex M0, что позволяет производить большинство расчетов непосредственно внутри BNO-055 и пользоваться уже готовыми результатами [67].

Ниже приведена внутренняя архитектура описываемого датчика и его взаимодействие с внешним процессором (рис. 27).

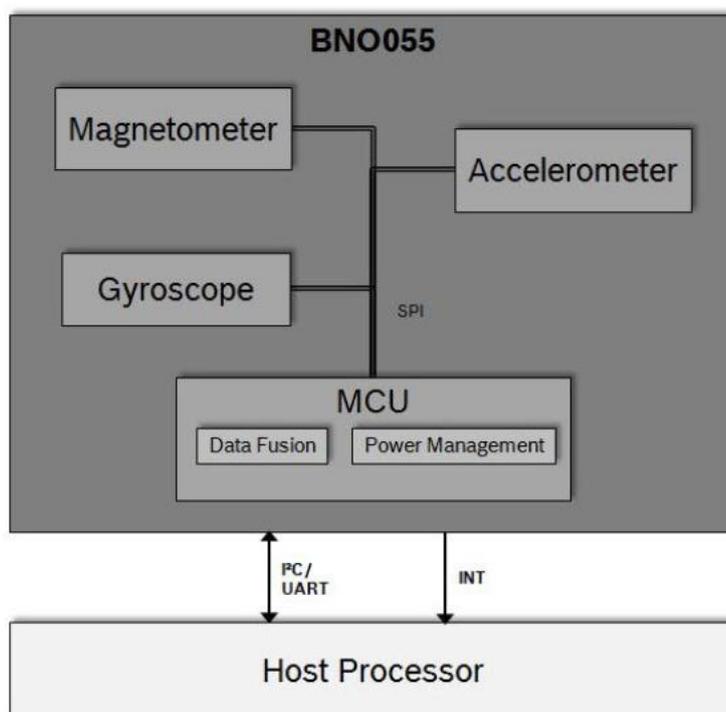


Рисунок 27 – Внутренняя архитектура датчика

Ядро получает данные: акселерометра, гироскопа, магнитометра, производит расчеты и сохраняет результаты. Результаты в дальнейшем могут быть запрошены внешними устройствами. Прибор может выдавать информацию: по кватернионам, линейному ускорению, вектору гравитации, направлению движения устройства [68].

Согласно документации BNO-055 может поддерживать три интерфейса обмена с внешними устройствами: I2C, HID-I2C, UART (Serial). Какой интерфейс будет применяться, задается логическими значениями на паре контактов микросхемы 5 (PS1) и 6 (PS0), если оба контакта подтянуты к нулю, то с датчи-

ком можно осуществлять обмен данными по I2C протоколу. Ниже приведена стандартная схема подключения устройства при использовании его по протоколу I2C (рис. 28) [69].

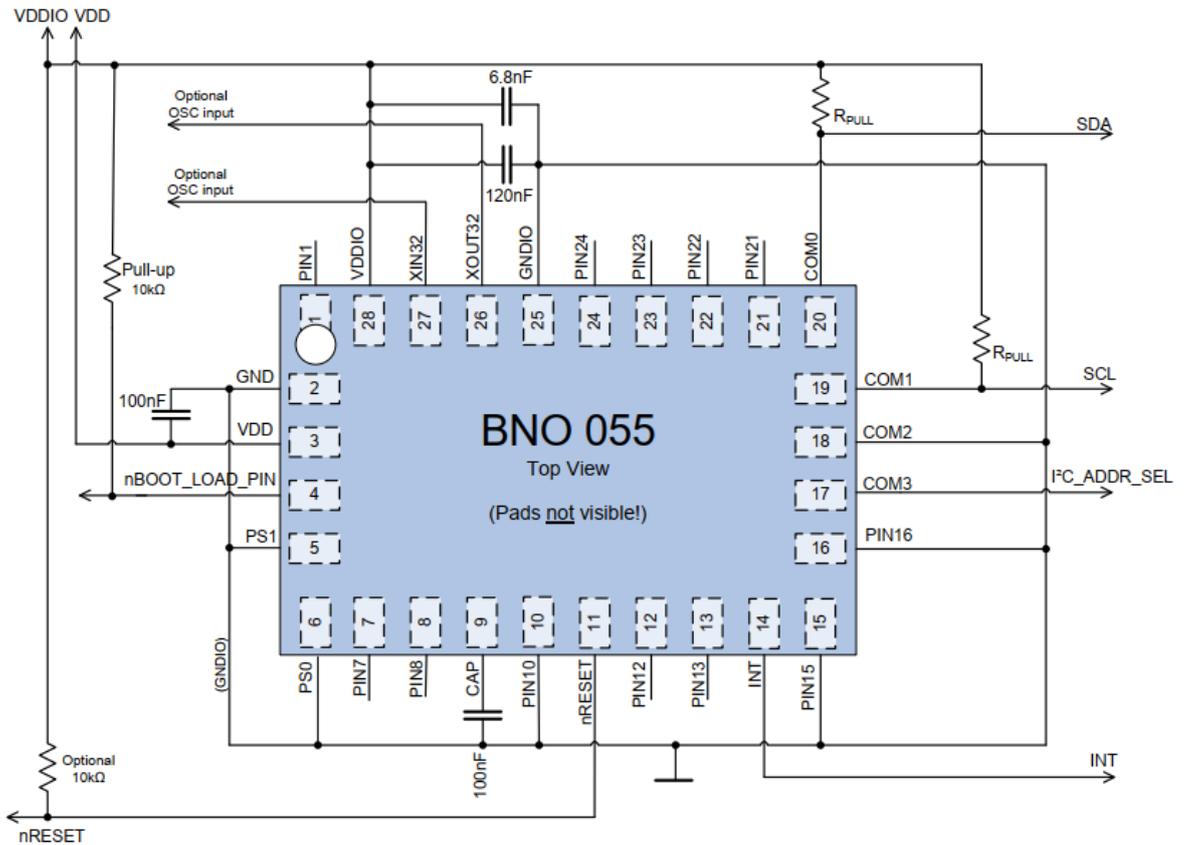


Рисунок 28 – Схема подключения датчика BNO-055 по интерфейсу I2C

Приведем основные характеристики микросхемы Bosch Sensortec BNO-055 [70]:

- рабочие диапазоны акселерометра: (± 2 , ± 4 , ± 8 , ± 16) g;
- настраиваемая полоса ФНЧ акселерометра: 1 кГц – менее 8 Гц;
- рабочие диапазоны гироскопа: от ± 125 °/сек до ± 2000 °/сек;
- настраиваемая полоса ФНЧ гироскопа: (523 - 12) Гц;
- рабочий диапазон магнитометра: ± 1300 мкТл по осям x, y, ± 2500 мкТл по оси z;
- разрешение магнитометра: ~ 0.3 мкТл;
- напряжение питания микросхемы: (2.4 - 3.6) В;

- корпус: 28-выводный LGA размером (3.8 × 5.2 × 1.13) мм;
- диапазон рабочих температур: от (−40 до 85) °С.

I2C-мультиплексор TCA9548A – это устройство, имеющее 8 двунаправленных переключателя, которые могут управляться по шине I2C (рис. 29) [71].



Рисунок 29 – Внешний вид мультиплексора TCA9548A

Устройство обладает следующими характеристиками [72]:

- 1-8 двунаправленных преобразующих коммутаторов;
- совместимость с шиной I2C и SMBus
- вход сброса, активируемый низким уровнем сигнала;
- адрес, устанавливаемый с помощью трех аппаратных адресных выводов, для подключения до восьми приборов;
- выбор канала через шину I2C;
- включение при всех отключенных коммутирующих каналах;
- низкое сопротивление включения коммутаторов;
- преобразование уровней напряжения между 2,5 В, 3,3 В, и 5 В шинами;
- отсутствие сбоев при включении питания;
- поддержка горячего подключения;
- низкий ток потребления в дежурном режиме;
- диапазон рабочего напряжения питания от 1,65 В до 5,5 В;

- входы имеют допустимый уровень напряжения 5 В;
- быстрая шина I2C, частота 400 кГц;
- тиристорное защелкивание превышает 100 мА в соответствии со стандартом JESD 78, Class II;
- защита от электростатического разряда JESD 22 превышает:
 - 2000 В модель тела человека (A114-A);
 - 200 В модель машины (A115-A);
 - 1000 В модель зарядного устройства (C101).

Аппаратно-программная платформа Arduino Mega 2560 это плата микроконтроллера на базе ATmega2560. Она имеет 54 цифровых входа/выхода (из которых 15 могут использоваться в качестве ШИМ-выходов), 16 аналоговых входов, 4 UART (аппаратные последовательные порты), кварцевый генератор 16 МГц, USB-соединение, разъем питания, заголовок ICSP и кнопку сброса [73].



Рисунок 30 – Внешний вид Arduino Mega 2560

Внешний вид платформы представлен на рисунке 30. Arduino Mega 2560 содержит все необходимое для поддержки микроконтроллера; для работы достаточно подключения к компьютеру с помощью кабеля USB или подключение с помощью адаптера переменного тока в постоянный или аккумулятора. Плата Mega 2560 совместима с большинством экранов, предназначенных для Uno и бывших плат Duemilanove или Diecimila [74].

Технические характеристики аппаратно-программной платформы Arduino Mega 2560 [75]:

- микроконтроллер: ATmega2560;
- рабочее напряжение: 5 В;
- входное напряжение (рекомендуемое): (7-12) В;
- входное напряжение (предельное): (6-20) В;
- цифровые Входы/Выходы: 54 (14 из которых могут работать также как выходы ШИМ);
- аналоговые входы: 16;
- постоянный ток через вход/выход: 40 мА;
- постоянный ток для вывода 3.3 В, 50 мА;
- флеш-память: 256 Кбайт (из которых 8 Кбайт используются для загрузчика);
- ОЗУ: 8 Кбайт;
- энергонезависимая память: 4 Кбайт;
- тактовая частота: 16 МГц;
- LED_BUILTIN: 13;
- длина: 101.52 мм;
- ширина: 53.3 мм;
- вес: 37 г.

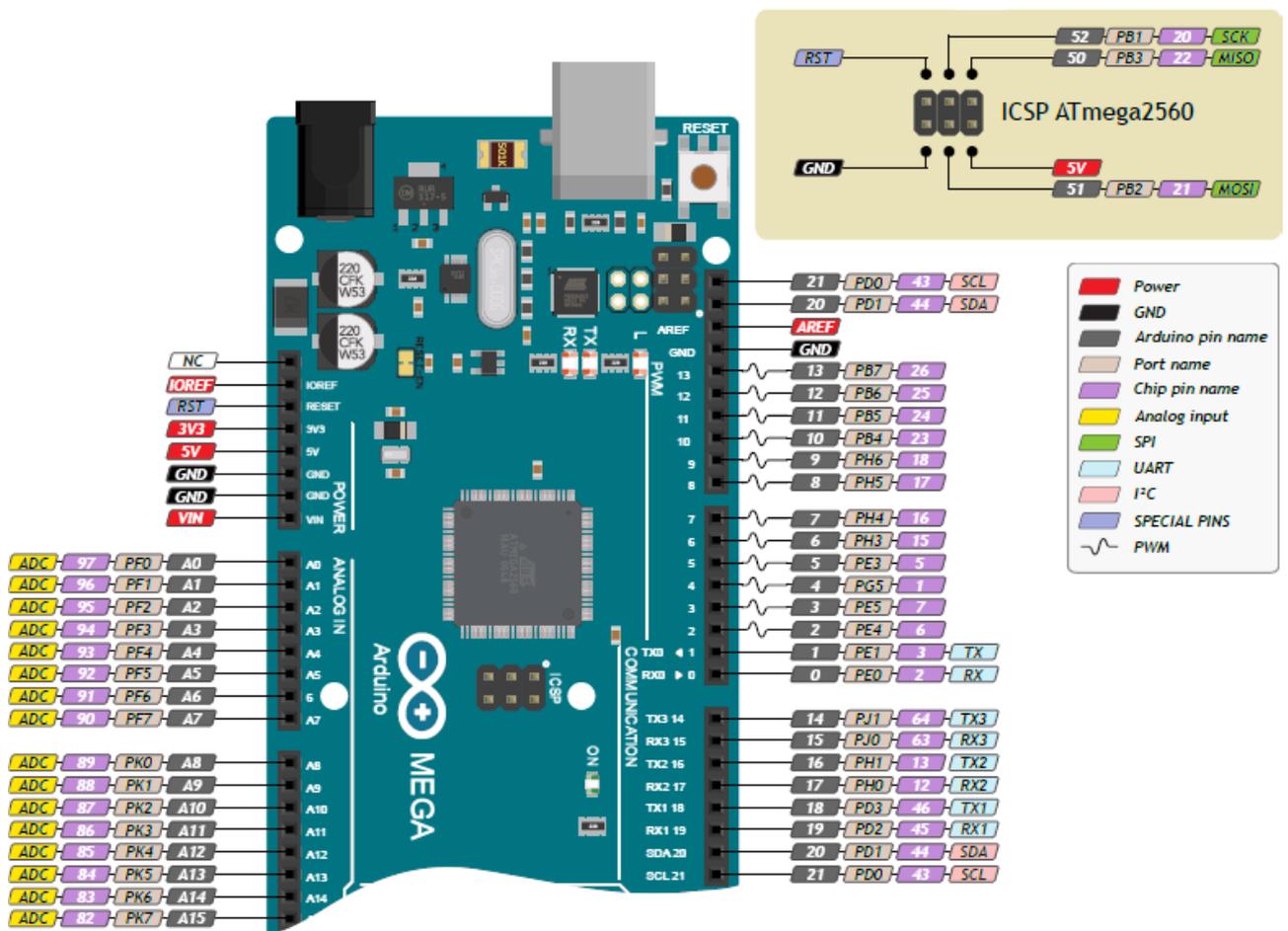


Рисунок 31 – Расположение контактов Arduino MEGA2560

Расположение контактов ввода-вывода аппаратно-программной платформы Arduino MEGA2560 представлено на рисунке 31 [76].

Питание осуществляется с внешнего источника, либо по USB-шине с управляющего компьютера, либо с адаптера переменного тока в постоянный через разъем питания VIN 12 В. Коммутация платформы с внешними устройствами по шине I2C осуществляется через контакты SDA20 и SCL21. Контроль работы и диагностика ошибок осуществляется визуально через светодиодные индикаторы на платке платформы.

Питание внешних устройств (в нашем случае мультиплексора и датчиков абсолютной ориентации) осуществляется через разъем 3V3, напряжение на выводе 3.3 В генерируемое микросхемой FTDI на платформе. Максимальное потребление тока 50 мА. Также внешние устройства подключаются к выводам

заземления GND.

2.4.2 Обоснование выбора программных сред разработки

Arduino Software (IDE) – среда для разработки микрокода для Arduino и установки микрокода в ПЗУ аппаратной платформы Arduino Mega. Контроллером костюма инерциального захвата движения является аппаратная платформа Arduino Mega 2560 ввиду доступности и разнообразного функционала. Платформа Arduino Mega 2560 программируется с помощью свободного программного обеспечения Arduino Software IDE (Integrated Development Environment). По этой причине разработку программного обеспечения для контроллера костюма необходимо будет вести средствами Arduino Software (IDE) [77].

Датчики абсолютной ориентации BNO055 (разработки фирмы Bosch) подключены к аппаратной платформе Arduino Mega 2560 по интерфейсу I2C через I2C-мультиплексор на микрочипе TCA9548A. Arduino Software (IDE) содержит текстовый редактор для написания кода, поле для отображения сообщений, текстовую консоль, панель меню с кнопками, вызывающими определенные функции и меню. При подключении аппаратной платформы Arduino к Arduino Software (IDE) возможен обмен данными, а также загрузка программ в аппаратную платформу Arduino. Язык программирования для Arduino Software (IDE): выбор языка программирования обусловлен тем, что для программирования устройств Arduino на Arduino Software (IDE) используется специальная версия языка C++, на котором пишутся программы, которые называются «скетчи». Другие языки программирования Arduino Software (IDE) не поддерживаются. [78]

Для Arduino IDE существуют общедоступная бесплатная библиотека BNO055.h для взаимодействия между датчиками абсолютной ориентации BNO055 и аппаратно-программной платформой Arduino. Библиотека BNO055.h представляет собой набор классов для управления и обработки данных, поступающих с датчиков абсолютной ориентации BNO055. Чтение данных с одного из датчиков или блоков микросхемы BNO055: readEul() – эйлеров; readQuat() –

кватернионов; readAccel() – датчика ускорения; readGyro() – гироскопа; readMag() – магнитного датчика; readLinAcc() – датчика linAcc; readGrav() – датчика гравитации; readAbs-Acc() – вращение вектора из кватернионов; readTemp() – датчика температуры.

Blender и Blender Game Engine. Для анимации скелетной трехмерной модели человека используется бесплатное программное обеспечение Blender для Windows или Blender для Linux. В состав 3D-редактора Blender входит модуль Blender Game Engine, с помощью которого осуществляется применение захвата движения скелетной трехмерной модели человека.

Программное обеспечение Blender поддерживает программирование скриптов только на языке Python. Предусматривается использование библиотеки mathutils для обработки кватернионов. Среда разработки скрипта на языке Python: IDLE Python 3.7, 3.8.

3 ПРОГРАММНО-ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОГО РЕШЕНИЯ

3.1 Основные этапы разработки инерциального костюма

3.1.1 Проектирование электротехнической схемы костюма

После подбора и обоснования электронных компонентов для отработки технологии была произведена сборка макета на макетной плате из следующих компонентов.

Аппаратно-программная платформа Arduino Mega, к которой по шине I2C к разъемам D20/SDA и 21/SCL подключен мультиплексор TCA9548A к разъемам SDA и SCL. Питание мультиплексора TCA9548A осуществляется через разъем 3V3, напряжение на выводе которого составляет 3.3 В и генерируется микросхемой FTDI платформы Arduino, и заземление через разъем GND. К мультиплексору питание подключено к разъему VIN, заземление подключено к разъему GND.

К мультиплексору с помощью макетных проводников подключены 3 датчика абсолютной ориентации BNO055 к разъемам шины I2C: первый датчик к разъемам SD0 и SC0, второй датчик – SD1 и SC1, третий датчик – SD2 и SC2 соответственно. Питание датчиков осуществляется с разъемов и GND.

Питание и управление Arduino Mega осуществляется по шине USB с управляющего компьютера с установленным программным обеспечением Arduino IDE для загрузки микрокода и мониторинга выполнения программы микрокода.

После отработки технологии на макете с тремя датчиками был собран костюм инерциальной системы захвата движения, состоящий из сети 8 датчиков абсолютной ориентации BNO055, подключенных через I2C-мультиплексор TCA9548A к аппаратно-программной платформе Arduino Mega, где датчики абсолютной ориентации прикреплены к каждому сегменту, положение которого

должно быть отслежено. Схема расположения датчиков на теле человека представлена на рисунке 32 [79].

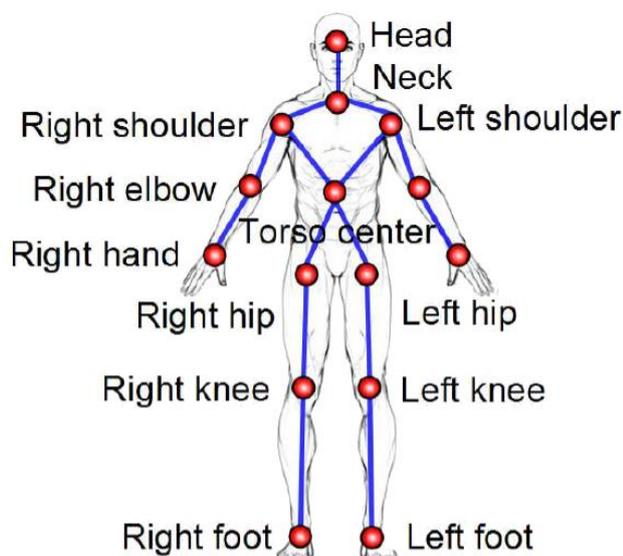


Рисунок 32 – Схема расположения датчиков на теле человека

Датчики располагаются в следующих точках: голова, шея, правое плечо, правый локоть, правое запястье, левое плечо, левый локоть, левое запястье, правое бедро, правое колено, правая ступня, левое бедро, левое колено, левая нога. Схема подключения датчиков к управляющему блоку представлена на рисунке 33.

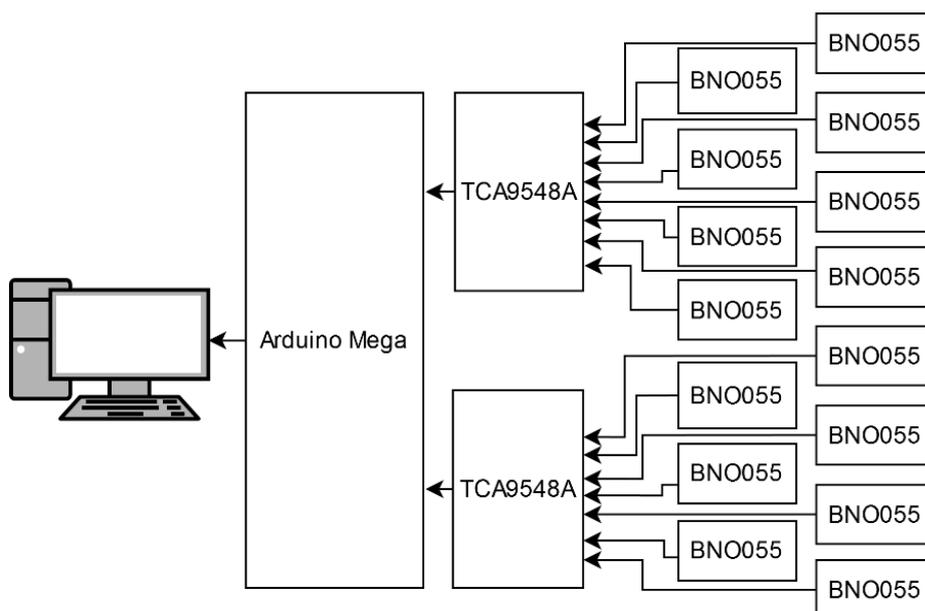


Рисунок 33– Схема подключения датчиков к управляющему блоку

С датчиков данные об изменении их положения в пространстве поступают в виде кватернионов.[80] Датчики закреплены на эластичном прототипе костюме и подключены с помощью 2-парной витой пары УТР-2, коммуникация определяется протоколом связи. Внешний вид прототипа костюма с подключенными датчиками представлен на рисунке 34.



Рисунок 34 – Прототип костюма

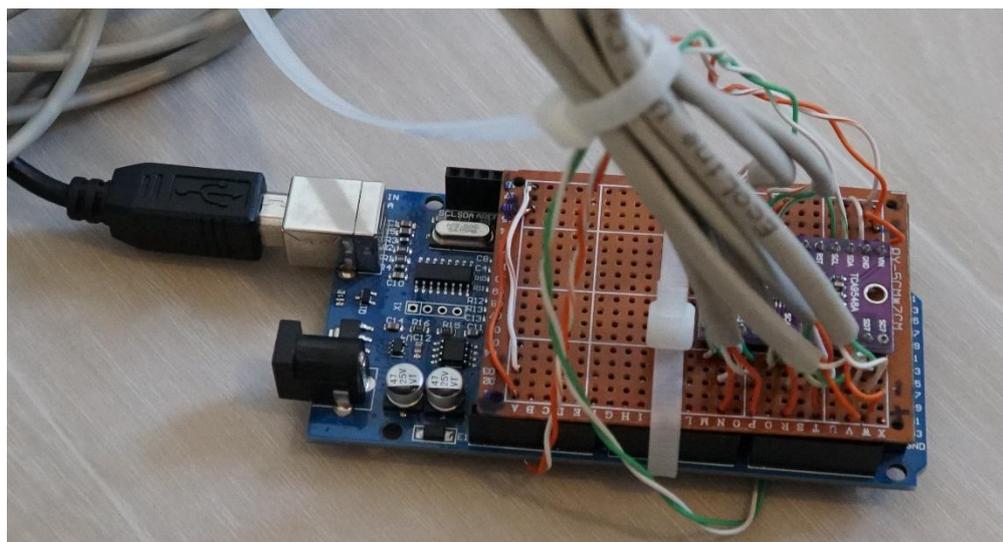


Рисунок 35 – Прототип контроллера

Прототип контроллера представлен на рисунке 35. На монтажной плате размещен мультиплексор, к контактам которого с помощью 2-парной витой пары UTP-2 подключены 8 датчиков BNO055. Мультиплексор в свою очередь подключен к контактам аппаратно-программной платформы Arduino Mega.

Отображение ориентации на определенные сегменты скелета позволяет фиксировать движение объекта. Зная все изменения сегментов во времени, можно проследить общую позу человека [81].

Каждые 8 датчиков подключены к одному мультиплексору TCA9548A, который мультиплексирует данные и передает по шине I2C в аппаратно-программную платформу Arduino Mega. Принципиальная схема прототипа контроллера костюма представлена на рисунке 36.

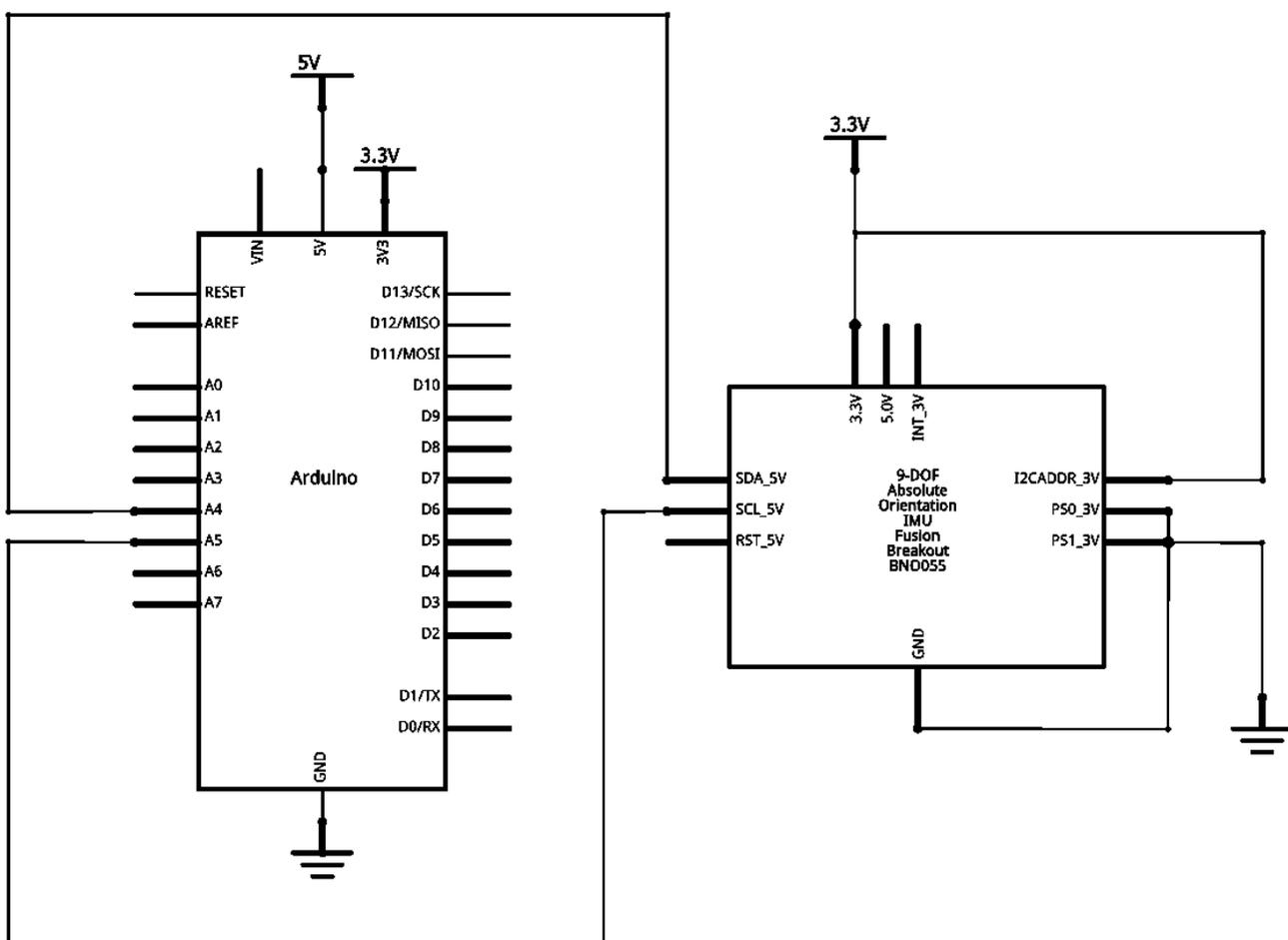


Рисунок 36 – Принципиальная схема контроллера костюма

Операционная среда включает в себя программное и аппаратное обеспечение.

Программное обеспечение:

1) Операционная система Microsoft Windows 7, 8, 10 или Linux 32/64 bits.
2) Программное обеспечение для анимации скелетной 3D-модели человека: Blender для Windows или Linux.

3) Среда разработки скрипта на языке Python: IDLE Python 3.7, 3.8 для Windows или Linux.

4) Среда для разработки микрокода для Arduino: Arduino IDE для Windows или Linux.

Аппаратное обеспечение:

1) Персональный компьютер. Системные требования для работы Blender [82]:

Минимальные:

- 64-bit двух-ядерный центральный процессор 2 ГГц с SSE2;
- 4 ГБ оперативной памяти;
- 1280×768 дисплей;
- мышь, трекпад или графический планшет+стилус;
- видеокарта с 1 ГБ RAM, OpenGL 3.3;
- менее 10 лет.

Рекомендованные:

- 64-bit четырех-ядерный центральный процессор;
- 16 ГБ оперативной памяти;
- Full HD дисплей;
- трехкнопочная мышь или графический планшет+стилус;
- видеокарта 4 ГБ RAM.

2) Аппаратная платформа Arduino Mega для отладки и тестирования.

Краткие характеристики платформы Arduino Mega представлены в подразделе 2.4.1.

3) Сенсоры для захвата движений человека: датчики абсолютной ориентации BNO055 Bosch Sensortec. Краткие характеристики датчиков представлены подразделе 2.4.1.

3.1.2 Разработка микрокода сбора данных

Микрокод сбора данных: модуль на языке C++ для аппаратной платформы Arduino Mega, к которой подключены датчики абсолютной ориентации BNO055 через I2C-мультиплексор TCA9548A.

Модуль осуществляет обработку на аппаратной платформе Arduino Mega 2560 сырых данных с датчиков абсолютной ориентации BNO055, подключенных по интерфейсу I2C через I2C-мультиплексор на микрочипе TCA9548A. На аппаратной платформе Arduino Mega 2560 вычисляется изменения координат датчиков абсолютной ориентации BNO055 в трехмерном пространстве в режиме реального времени, закрепленных на теле человека в определенных точках, соответствующих узлам на скелетной 3D модели человека в 3D-редакторе Blender. Обработанные данные передаются с контроллера на Arduino в формате кватернионов в режиме реального времени. Передача обработанных данных с контроллера осуществляется через COM-порт на компьютер в 3D-редактор Blender.

Входные и выходные данные микрокода сбора данных определены следующим образом. Данные, поступающие с датчиков абсолютной ориентации BNO055 и обрабатываемые контроллером, представляют собой изменения координат датчиков в трехмерном пространстве в режиме реального времени.

Выходные данные имеют следующий вид:

0.51, 0.53, -0.57, 0.36; 0.99, 0.10, 0.01, 0.06;

0.51, 0.52, -0.58, 0.36; 0.99, 0.10, 0.00, 0.07;

0.51, 0.51, -0.59, 0.37; 0.99, 0.09, -0.01, 0.08;

0.51, 0.49, -0.60, 0.37; 0.99, 0.09, -0.01, 0.08;

Представленные данные можно прокомментировать следующим образом: микроконтроллер считывает ориентацию каждого датчика в кватернионах и от-

правляет их в виде строки. В каждой строке представлено два кватерниона, состоящих из четырех чисел через запятую. Каждый кватернион соответствует изменению положения определенного узла на скелетной модели в 3D-редакторе Blender. [83]

Декомпозиция модуля микрокода сбора данных на аппаратно-программной платформе Arduino представлена на рисунке 37.

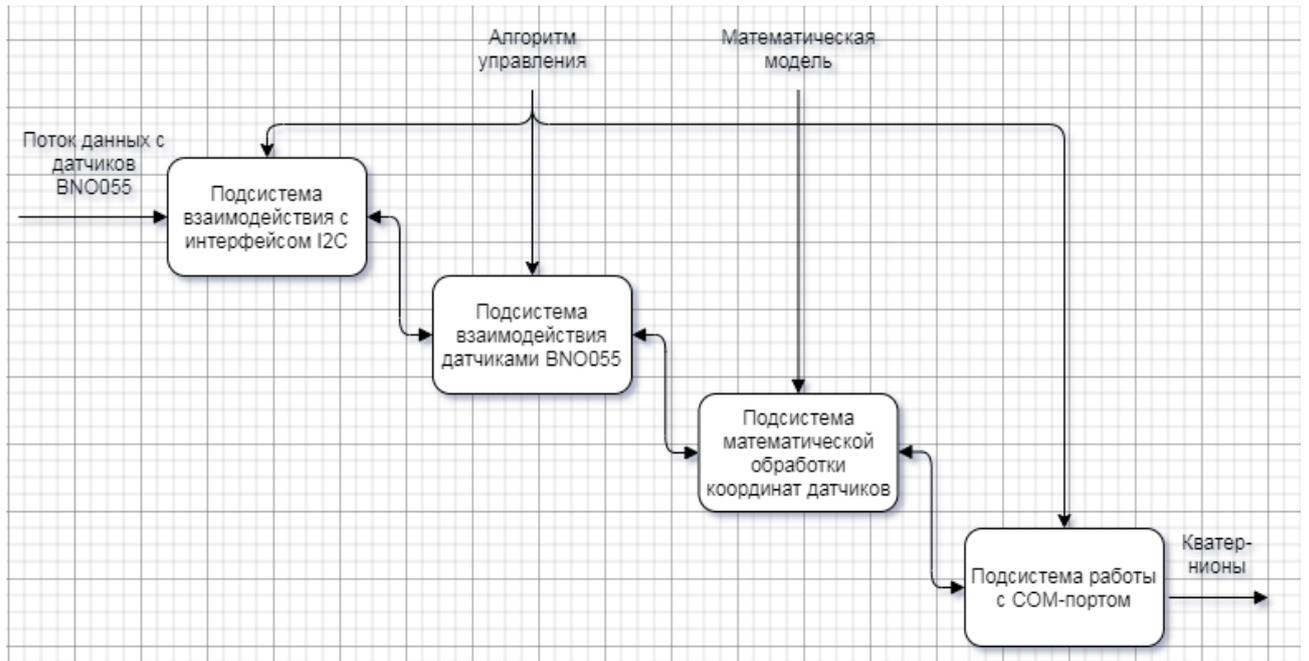


Рисунок 37 – Декомпозиция модуля микрокода сбора данных

Программа интерпретации данных состоит из двух категорий классов. Классы библиотеки BNO055 для датчиков абсолютной ориентации: классы библиотеки BNO055 представляют собой набор классов для управления и обработки данных, поступающих с датчиков абсолютной ориентации BNO055. Чтение данных с одного из датчиков или блоков микросхемы BNO055: readEul() – эйлеров; readQuat() – кватернионов; readAccel() – датчика ускорения; readGyro() – гироскопа; readMag() – магнитного датчика; readLinAcc() – датчика linAcc; readGrav() – датчика гравитации; readAbsAcc() – вращение вектора из кватернионов; readTemp() – датчика температуры.

UML-диаграмма классов микрокода сбора данных представлена на рисунке 38.

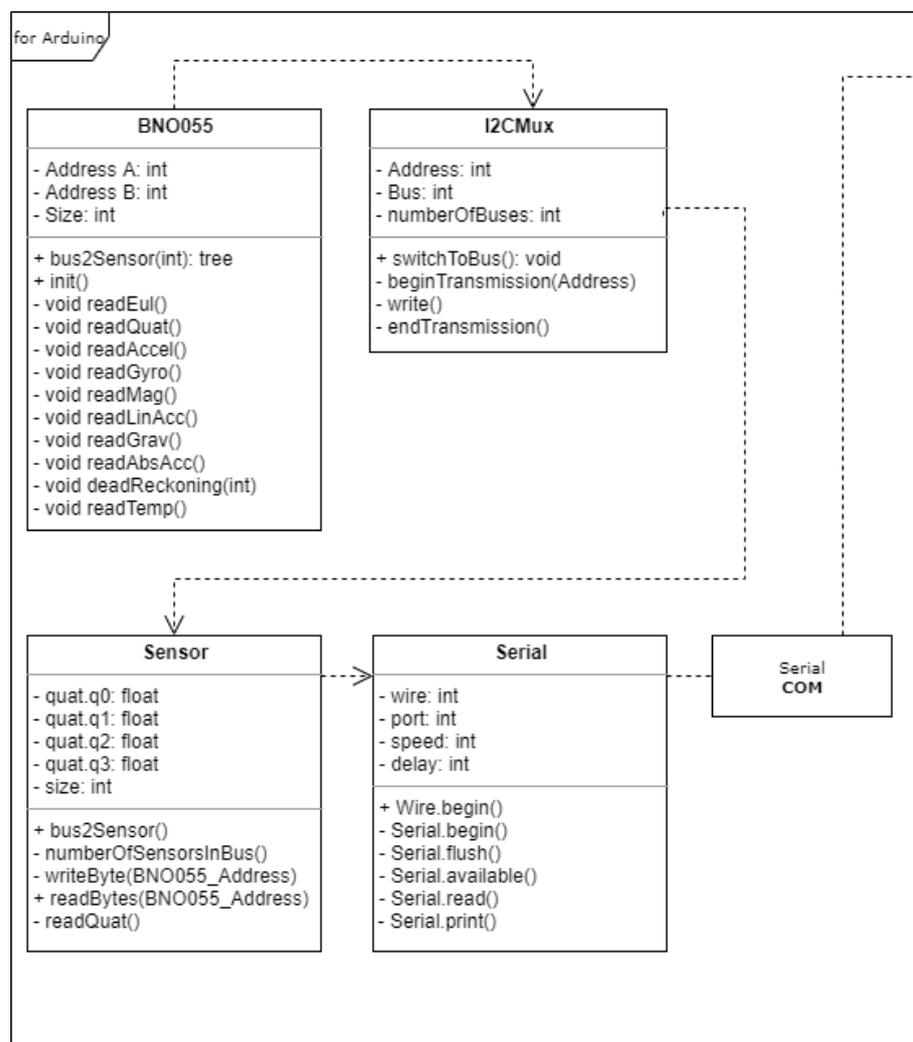


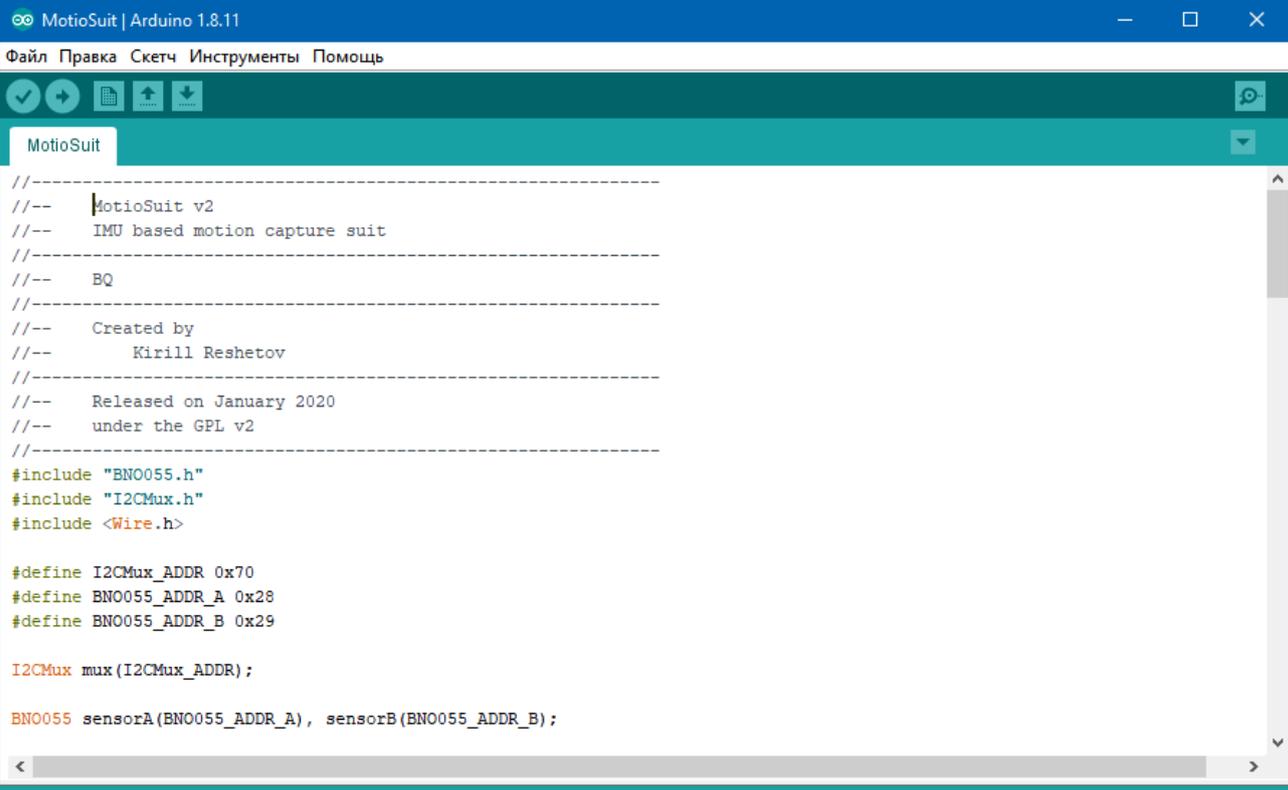
Рисунок 38 – UML-диаграмма классов микрокода сбора данных

Классы модуля для аппаратно-программной платформы Arduino. Класс BNO055 для работы с данными с датчиков BNO055 с использованием библиотеки BNO055: Address A и Address B – адресное пространство шины для группы датчиков в байтах; bus2Sensor() – подключение шины к датчикам.

Класс I2CMux для работы с микросхемой мультиплексора: Address – адрес шины в байтах; Bus – шина; numberOfBuses – количество шин; switchToBus() – триггер переключения на шину; beginTransmission() – начать передачу; write() – осуществить запись в память; endTransmission() – завершить передачу.

Класс `Sensor` для работы с сущностями сенсоров, для которых используются обработанные данные с датчиков BNO055: `quat.q0`, `quat.q1`, `quat.q2`, `quat.q3` – четыре кватерниона положения датчика BNO055 в пространстве; `bus2Sensor()` – передача данных от шины к сенсору; `numberOfSensorsInBus()` – количество сенсоров на шине; `writeByte(BNO055_Address)` – записать в память адрес датчика; `readBytes(BNO055_Address)` – прочитать адрес датчика из памяти; `readQuat()` – прочитать кватернион положения датчика в пространстве.

Класс `Serial` для передачи результирующих данных через последовательный порт с платформы Arduino на персональный компьютер: `wire`: обращение к библиотеке `wire`; `port`: COM-порт; `speed`: скорость в бод; `delay`: задержка сигнала.



```
//-----  
//-- MoCapSuit v2  
//-- IMU based motion capture suit  
//-----  
//-- BQ  
//-----  
//-- Created by  
//-- Kirill Reshetov  
//-----  
//-- Released on January 2020  
//-- under the GPL v2  
//-----  
#include "BNO055.h"  
#include "I2CMux.h"  
#include <Wire.h>  
  
#define I2CMux_ADDR 0x70  
#define BNO055_ADDR_A 0x28  
#define BNO055_ADDR_B 0x29  
  
I2CMux mux(I2CMux_ADDR);  
  
BNO055 sensorA(BNO055_ADDR_A), sensorB(BNO055_ADDR_B);
```

Рисунок 39– Скetch MoCapSuit.ino в интерфейсе Arduino IDE

Модуль для аппаратно-программной платформы Arduino представляет собой библиотеки `BNO055.h` и `I2CMux.h`, а также файл скетча `MoCapSuit.ino`. Библиотека `BNO055.h` для датчиков абсолютной ориентации BNO055 и биб-

библиотека I2CMux.h для мультиплексора устанавливаются в Arduino IDE согласно руководству пользователя. Файл MoCapSuit.ino импортируется в энергонезависимую память Arduino через интерфейс Arduino IDE, пункт меню Скетч/Загрузка. Скетч MoCapSuit.ino в интерфейсе Arduino IDE представлен на рисунке 39.

Пользователь изменяет положение датчиков абсолютной ориентации BNO055, закрепленных на его теле. Данные передаются с каждого датчика на микросхему мультиплексора, после чего мультиплексируемые данные передаются в микропроцессор аппаратно-программной платформы Arduino, где происходит обработка потока данных, преобразование потока данных в кватернионы, проверка на ошибки.

Процесс захвата движения осуществляется следующим образом. При подаче питания на платформу Arduino происходит инициализация оборудования и программного обеспечения, загруженного в память. Далее происходит последовательный опрос датчиков, подключенных к шине SD0 микропроцессора Arduino. В случае, если опрос завершается не удачей, то отображается LED-индикация ошибки. В случае, если опрос датчиков завершается успешно, то с помощью микросхемы мультиплексора осуществляется мультиплексирование данных с каждого датчика и прием потока данных в память микропроцессора. Далее осуществляется преобразование данных о координатах с датчиков в кватернионы. Затем осуществляется передача обработанного потока данных через COM-порт в компьютер.

В случае возникновения ошибок осуществляется индикация с помощью светодиодов, размещенных на плате Arduino. Обработанные данные поступают через последовательный порт с Arduino в персональный компьютер в Blender Game Engine 3D редактора Blender.

Управление переходами состояний аппаратно-программной платформой Arduino осуществляется следующим образом. Вначале осуществляется проверка каждого датчика, подключенного к шине SD0, если какой-либо датчик не

вернул ответ, то происходит остановка и LED-индикация ошибки, если Все датчики последовательно возвращают ответ, то происходит переход к следующему блоку. При получении данных с датчиков происходит мультиплексирование потока данных со всех датчиков. Далее происходит переход к преобразованию данных о координатах датчиков в кватернионы. После чего происходит переход к передаче данных в COM-порт, в ходе которого проводится проверка доступности COM-порта: если COM-порт не доступен, то происходит остановка и LED-индикация ошибки, если COM-порт доступен, то происходит переход к успешному завершению этапа.

3.1.3 Разработка программы интерпретации данных

Программа интерпретации данных состоит из двух модулей:

1) модуль на языке Python для обработки потока данных, получаемых с датчиков, для 3D-редактора Blender.

2) Модуль для Blender Game Engine, в котором получаемые структурированные данные будут применяться для анимации 3D-модели человека в режиме реального времени.

Проектирование взаимодействия модулей представлено на диаграмме взаимодействия (рис. 40).

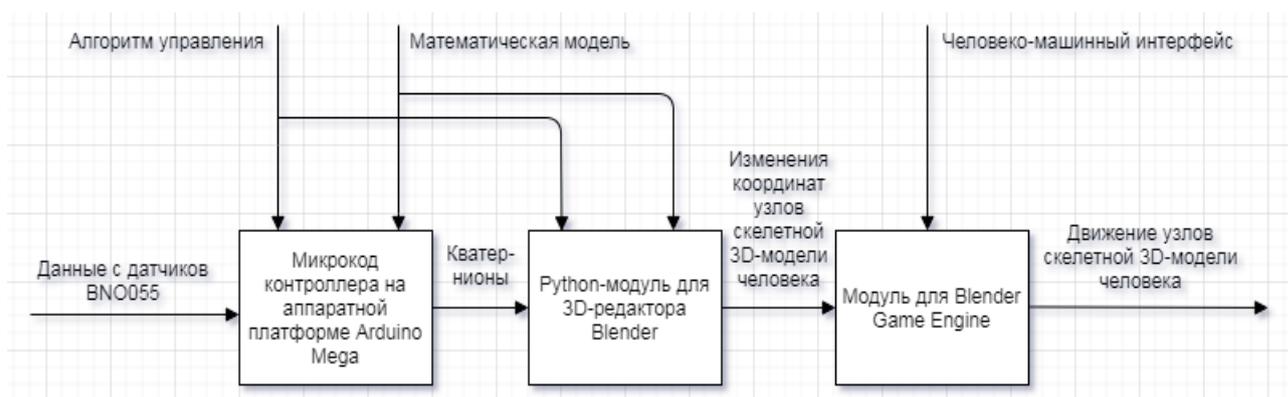


Рисунок 40 – Диаграмма взаимодействия модулей программного обеспечения для костюма инерциального захвата движения

Программа интерпретации данных содержит следующие классы. Класс `Serial` для получения данных с последовательного порта персонального компьютера: `name` – имя COM-порта в компьютере; `port`: COM-порт; `speed`: скорость в бод; `delay`: задержка сигнала. Класс `Angles` для изменения углов узлов на модели в соответствии с поступившими кватернионами. Класс `mathutils` для обращения к библиотеке `mathutils`. Класс `bge.scene` – загрузка текущей сцены из Blender. Класс `bge.source` – загрузка текущих объектов из Blender. Класс `Armature` для обращения к скелету 3D модели, координаты узлов которого необходимо изменить. Класс `3DObject` для работы с 3D моделью, для передачи измененных координат узлов на модели, благодаря чему осуществляется передача движений человека к 3D модели. UML-диаграмма классов программы интерпретации данных представлена на рисунке 45.

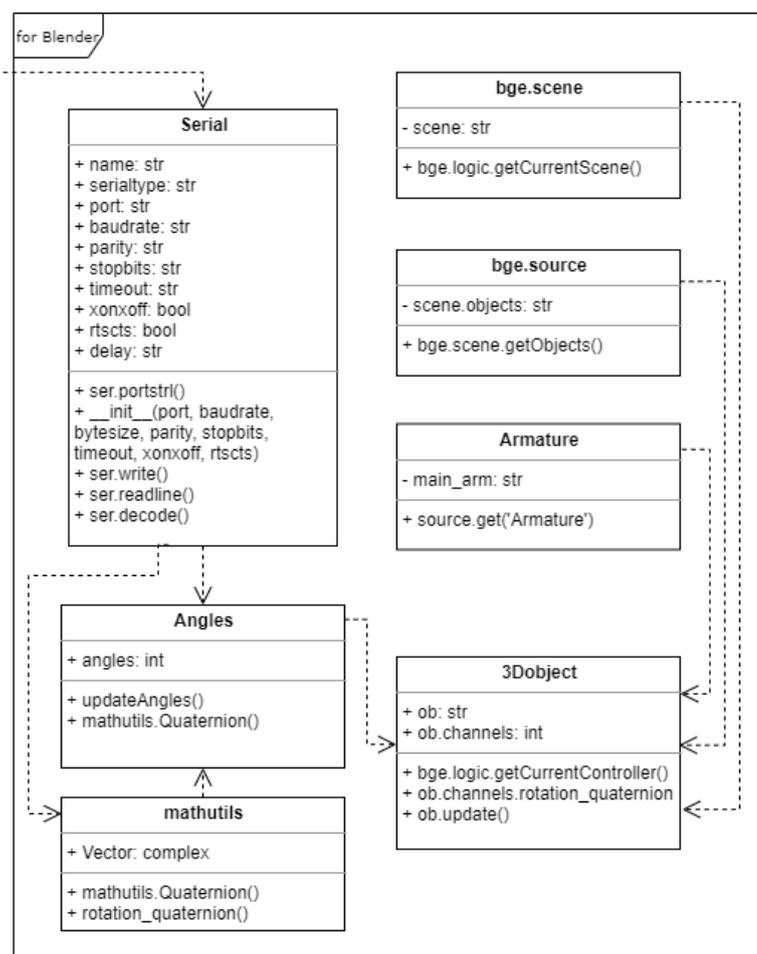
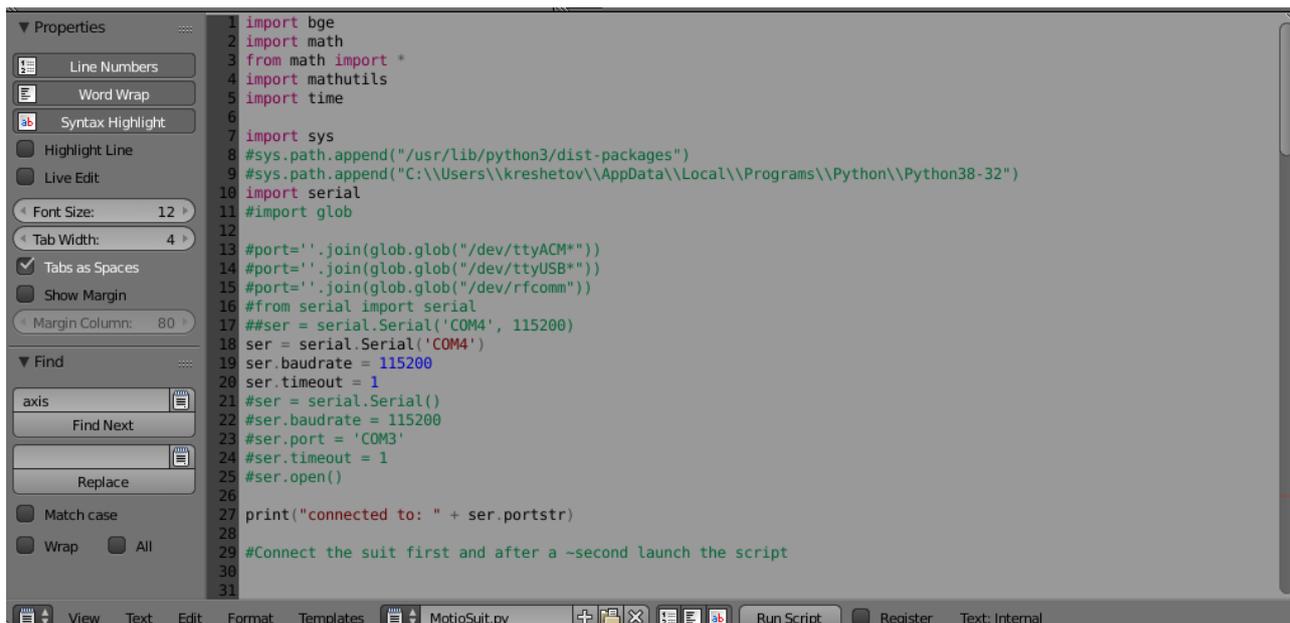


Рисунок 45 – UML-диаграмма классов программы интерпретации данных

Модуль на языке Python для обработки потока данных. В 3D-редакторе Blender осуществляется обработка потока данных скриптом на языке Python с использованием математической библиотеки `mathutils` для обработки кватернионов. Результатом работы модуля является изменение координат узлов на скелетной 3D модели человека, благодаря чему осуществляется передача движения от человека, на котором закреплены датчики скелетной 3D-модели в 3D-редакторе Blender.



```
1 import bge
2 import math
3 from math import *
4 import mathutils
5 import time
6
7 import sys
8 #sys.path.append("/usr/lib/python3/dist-packages")
9 #sys.path.append("C:\\Users\\kreshetov\\AppData\\Local\\Programs\\Python\\Python38-32")
10 import serial
11 #import glob
12
13 #port='.join(glob.glob("/dev/ttyACM*"))
14 #port='.join(glob.glob("/dev/ttyUSB*"))
15 #port='.join(glob.glob("/dev/rfcomm*"))
16 #from serial import serial
17 #ser = serial.Serial('COM4', 115200)
18 ser = serial.Serial('COM4')
19 ser.baudrate = 115200
20 ser.timeout = 1
21 #ser = serial.Serial()
22 #ser.baudrate = 115200
23 #ser.port = 'COM3'
24 #ser.timeout = 1
25 #ser.open()
26
27 print("connected to: " + ser.portstr)
28
29 #Connect the suit first and after a ~second launch the script
30
31
```

Рисунок 46 – Скрипт на языке Python в интерфейсе Blender

На рисунке 46 представлено отображение скрипта на языке Python в интерфейсе Blender. Скрипт выполняется и отлаживается только в редакторе Blender, т.к. используется встроенная в Blender библиотека `bge`.

Входными данными модуля на языке Python является структурированный поток данных в виде кватернионов, поступающих с контроллера устройства в режиме реального времени. Выходными данными является изменение координат узлов на скелетной 3D модели человека, благодаря чему осуществляется передача движения от человека, на котором закреплены датчики скелетной 3D-модели в 3D-редакторе Blender.

Декомпозиция модуля обработки потока данных для 3D-редактора Blender представлена на рисунке 47.

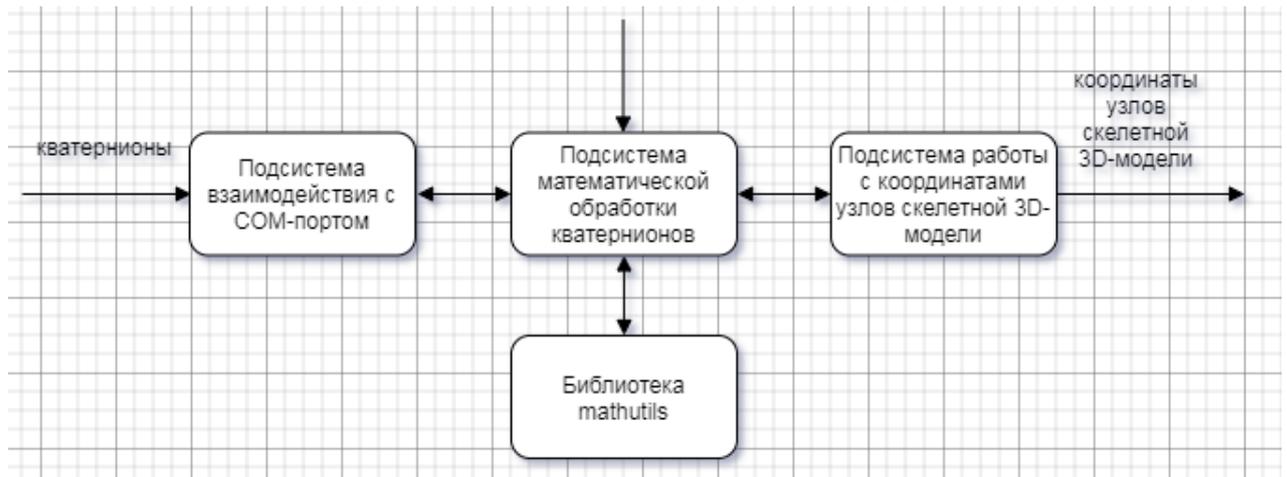


Рисунок 47 – Декомпозиция модуля обработки потока данных для 3D-редактора Blender

Модуль для Blender Game Engine, в котором получаемые структурированные данные будут применяться для анимации 3D-модели человека в режиме реального времени (рис. 48).

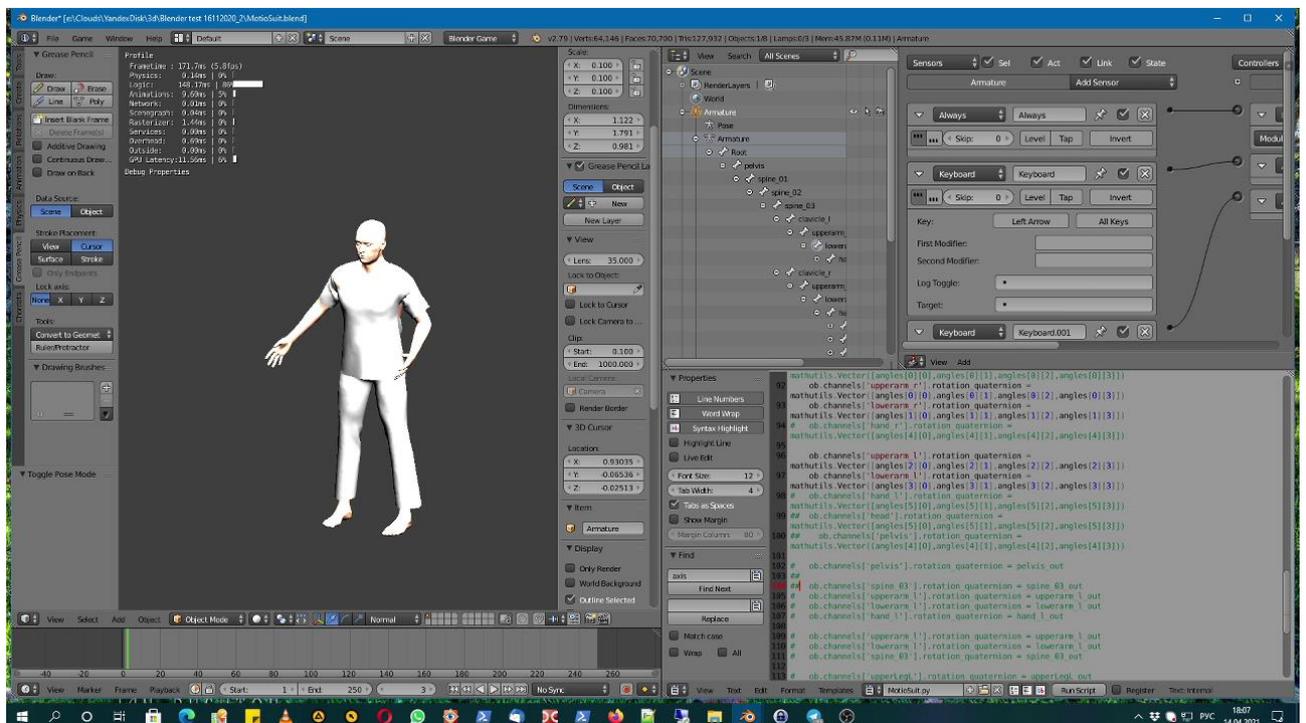


Рисунок 48 – Модуль для Blender Game Engine

В модуле для в Blender Game Engine происходит обработка потока данных из кватернионов и передача измененных углов скелету 3D модели человека с учетом окружения (сцены и объектов сцены) и скелета модели. В случае возникающих ошибок на мониторе компьютера осуществляется отображение ошибок модуля для Blender Game Engine. В случае отсутствия ошибок происходит отображение изменения положения узлов 3D модели человека.

Входные данные модуля для Blender Game Engine: каждый кватернион соответствует изменению положения определенного узла на скелетной модели в 3D-редакторе Blender. Выходные данные: воздействие на узлы на скелетной 3D модели человека, благодаря чему осуществляется передача движения скелетной 3D-модели в 3D-редакторе Blender.

Декомпозиция модуля для Blender Game Engine представлена на рисунке 49.

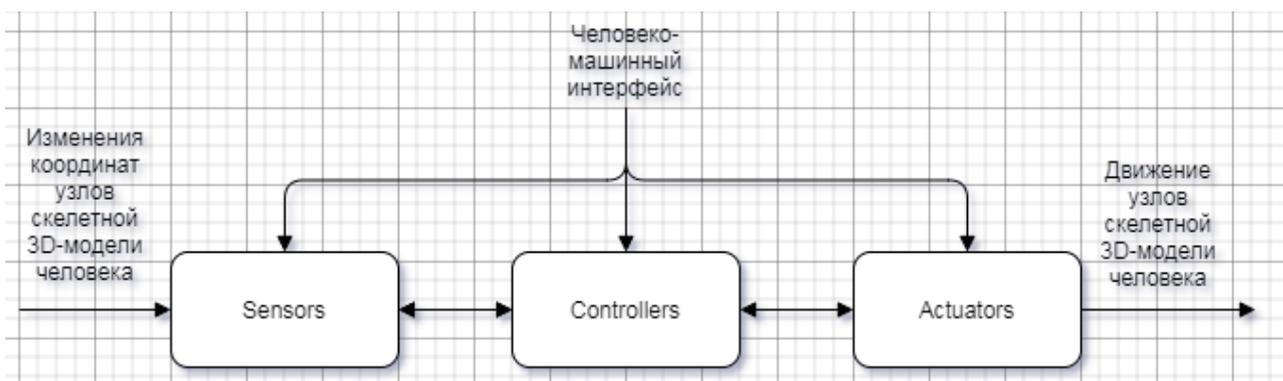


Рисунок 49 – Декомпозиция модуля для Blender Game Engine

Для обработки движений скелетной 3D-модели человека в 3D-редакторе Blender необходимо в Blender Game Engine (компонент 3D-редактора Blender) создать элементы «логические блоки» (logic bricks). Пример «логических блоков» в Blender Game Engine представлен на рисунке 50.

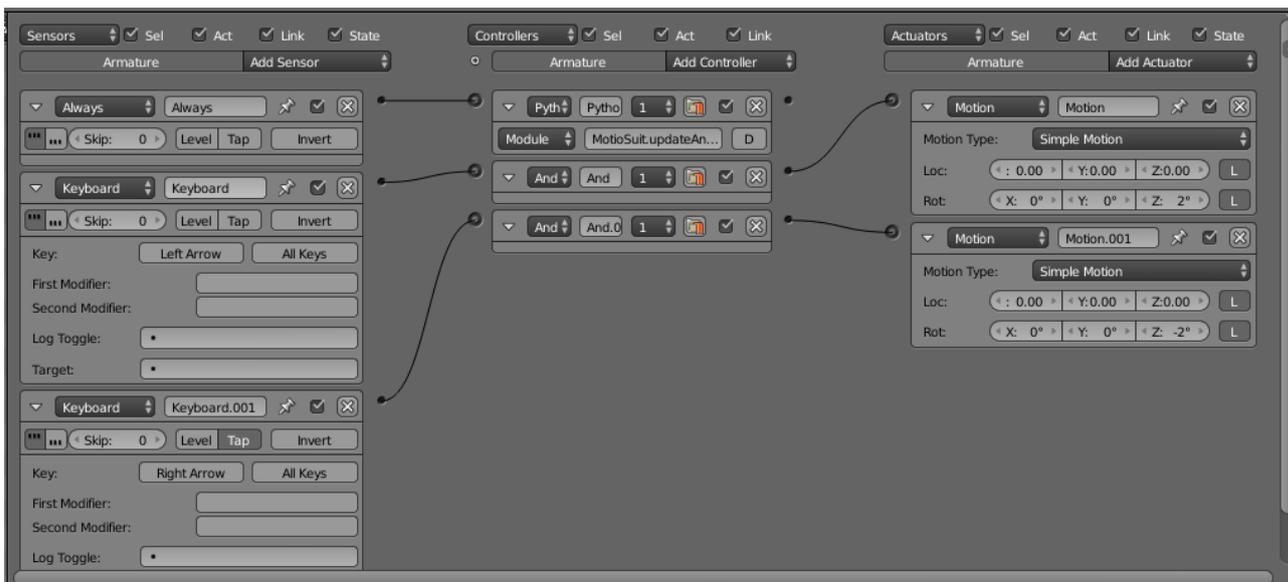


Рисунок 50 – «Логические блоки» в Blender Game Engine

«Логические блоки» состоят из «датчиков» (sensors), «контроллеров» (controllers) и «исполнительных механизмов» (actuators) для взаимодействия с 3D-моделью человека и осуществлением захвата движений человека и передачи изменения координат точкам на скелетной 3D-модели человека, полученных из модуля обработки потока данных для 3D-редактора Blender.

На компьютере в 3D редакторе Blender в модуле Blender Game Engine происходит обработка потока данных из кватернионов и передача измененных углов скелету 3D модели человека с учетом окружения (сцены и объектов сцены) и скелета модели. В случае возникающих ошибок на мониторе компьютера осуществляется отображение ошибок модуля для Blender Game Engine. В случае отсутствия ошибок происходит отображение изменения положения узлов 3D модели человека.

Управление переходами модуля для Blender Game Engine осуществляется следующим образом. Вначале проводится проверка потока данных с COM-порта: если поступающие данные не корректны, то происходит остановка и индикация ошибки, если данные корректны, то происходит переход к успешному следующему блоку передачи потока данных в Blender Game Engine. Вначале

происходит обработка координат датчиков и переход к изменению углов костей скелета, на которое также воздействует сцена и объекты сцены.

3.2 Примеры фактического тестирования инерциального костюма

3.2.1 Аппаратное и программное обеспечение для тестирования

Программное обеспечение для тестирования:

- операционная система Microsoft Windows 7, 8, 10 или Linux 32/64 бит;
- программное обеспечение для анимации скелетной 3D-модели человека:

Blender для Windows или Linux;

- среда для установки модуля в аппаратно-программную платформу

Arduino: Arduino IDE для Windows или Linux;

- модуль захвата движений Mocap.ino для аппаратно-программной платформы Arduino на языке программирования C++;

- библиотека BNO055.h для контроллера костюма на Arduino;

- модуль захвата движений Mocap.py для Blender Game Engine на языке программирования Python;

Аппаратное обеспечение для тестирования:

- контроллер костюма на основе аппаратно-программной платформы Arduino Mega, представленный в разделе 3.1.1;

- персональный компьютер. Далее представлены системные требования к персональному компьютеру для работы Blender.

Минимальные системные требования:

- 64-bit двух-ядерный центральный процессор 2 ГГц с SSE2;
- 4 ГБ оперативной памяти;
- 1280×768 дисплей;
- мышь, трекпад или графический планшет+стилус;
- видеокарта с 1 ГБ RAM, OpenGL 3.3;
- менее 10 лет.

Рекомендованные системные требования:

- 64-bit четырех-ядерный центральный процессор;

- 16 ГБ оперативной памяти;
- Full HD дисплей;
- трехкнопочная мышь или графический планшет+стилус;
- видеокарта 4 ГБ RAM.

3.2.2 Сценарий тестирования

Для подготовки к тестированию необходимо установить программное обеспечение. Модуль контроллера костюма устанавливается с помощью программатора Arduino IDE в энергонезависимую память аппаратно-программной платформы Arduino. Для работы с Arduino IDE необходимо использовать официальную документацию Arduino IDE, представленную по ссылке: <https://www.arduino.cc/en/Guide/Environment>. Также необходимо установить библиотеки для Arduino BNO055.h и I2CMux.h из комплекта поставки, согласно официальной документации Arduino: <https://www.arduino.cc/en/Guide/Libraries>.

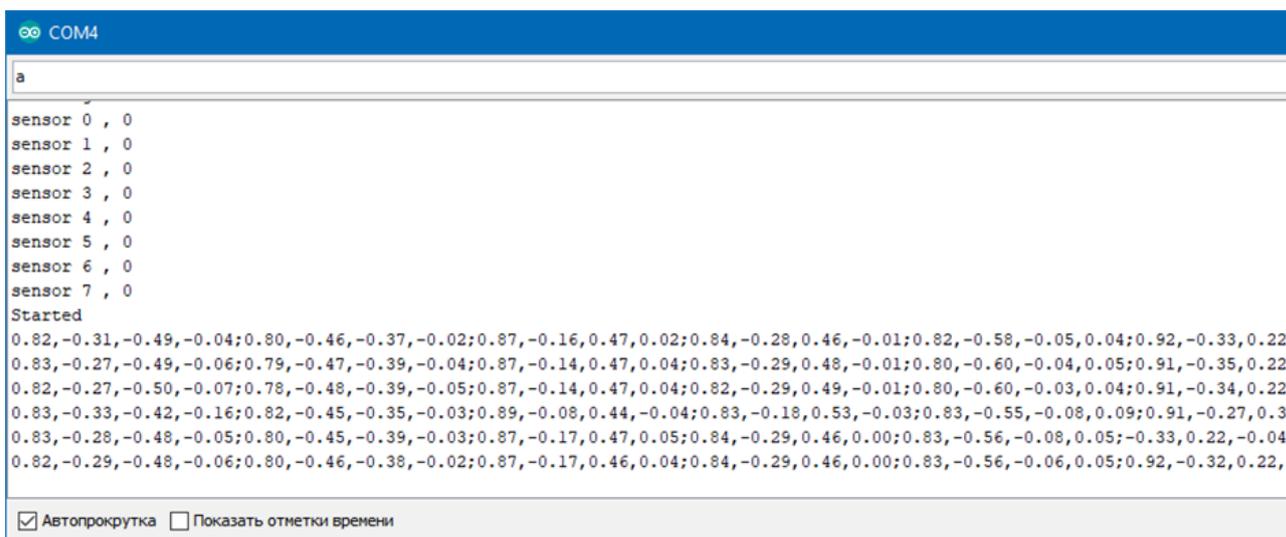
После установки библиотек необходимо загрузить модуль MoCapSuit.ino в память аппаратно-программной платформы Arduino Mega через интерфейс Arduino IDE: пункт меню Скetch/Загрузка. Установку модуля необходимо выполнять в соответствии с официальной документацией Arduino IDE: <https://www.arduino.cc/en/Guide/Environment#sketch>.

Для применения захвата движения к 3D-модели человека необходимо в 3D-редактор Blender импортировать файл MoCapSuit.blend, содержащий подготовленную 3D-модель человека, логические элементы для Blender Game Engine и скрипт захвата движения MoCapSuit.py на языке Python для Blender Game Engine. Для установки библиотек для Blender необходимо скопировать папки с библиотеками pyserial и mathutils из пакета поставки в папку с Python-библиотеками Blender по пути: C:\Program Files\Blender\blender-2.79-windows32\2.79\python\lib\site-packages\. После выполнения подготовительных процедур можно приступить к тестированию.

1) Диагностика датчиков и контроллера осуществляется двумя способами: с помощью светодиодной индикации на плате Arduino Mega и монитора портов

Arduino IDE. Светодиодная индикация ошибок на плате Arduino Mega: если последовательный опрос датчиков завершается не удачей, то на плате загорается красный светодиод.

Диагностика с помощью монитора портов Arduino IDE необходима для локализации неработающего датчика. Монитор портов отображает список всех рабочих датчиков. В случае, если один из датчиков сбоят или с ним потеряна связь, опрос датчиков остановится на нем, и таким образом будет локализована сбой (рис. 51).



```
COM4
a
sensor 0 , 0
sensor 1 , 0
sensor 2 , 0
sensor 3 , 0
sensor 4 , 0
sensor 5 , 0
sensor 6 , 0
sensor 7 , 0
Started
0.82,-0.31,-0.49,-0.04;0.80,-0.46,-0.37,-0.02;0.87,-0.16,0.47,0.02;0.84,-0.28,0.46,-0.01;0.82,-0.58,-0.05,0.04;0.92,-0.33,0.22,
0.83,-0.27,-0.49,-0.06;0.79,-0.47,-0.39,-0.04;0.87,-0.14,0.47,0.04;0.83,-0.29,0.48,-0.01;0.80,-0.60,-0.04,0.05;0.91,-0.35,0.22,
0.82,-0.27,-0.50,-0.07;0.78,-0.48,-0.39,-0.05;0.87,-0.14,0.47,0.04;0.82,-0.29,0.49,-0.01;0.80,-0.60,-0.03,0.04;0.91,-0.34,0.22,
0.83,-0.33,-0.42,-0.16;0.82,-0.45,-0.35,-0.03;0.89,-0.08,0.44,-0.04;0.83,-0.18,0.53,-0.03;0.83,-0.55,-0.08,0.09;0.91,-0.27,0.31,
0.83,-0.28,-0.48,-0.05;0.80,-0.45,-0.39,-0.03;0.87,-0.17,0.47,0.05;0.84,-0.29,0.46,0.00;0.83,-0.56,-0.08,0.05;-0.33,0.22,-0.04,
0.82,-0.29,-0.48,-0.06;0.80,-0.46,-0.38,-0.02;0.87,-0.17,0.46,0.04;0.84,-0.29,0.46,0.00;0.83,-0.56,-0.06,0.05;0.92,-0.32,0.22,-
```

Рисунок 51 – Выполнение программы в мониторе портов Arduino IDE

В поле ввода монитора портов необходимо ввести латинский символ «а», после чего будет произведен опрос датчиков и вывод на экран данных с датчиков за один такт (рис. 51).

2) Тестирование программного обеспечения осуществляется следующим образом. Необходимо надеть костюм, подключить USB шнур к контроллеру и компьютеру. Принять начальную позу, для этого встать прямо, вытянуть руки вниз, вперед и развести их в стороны (рис. 52).

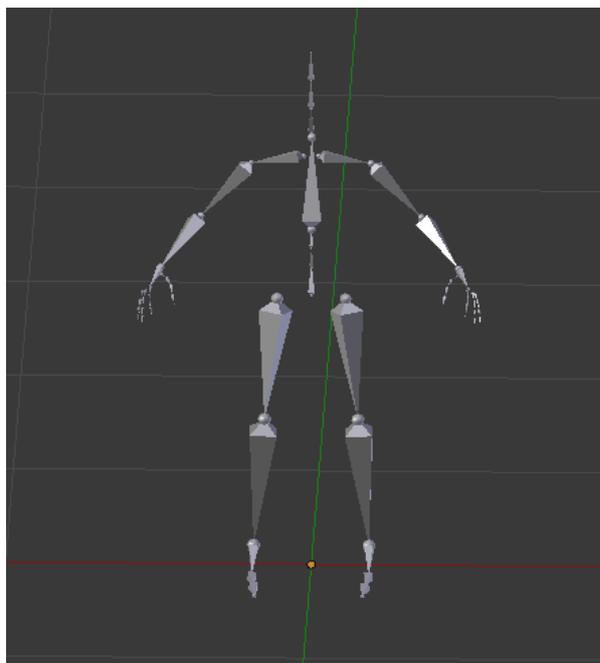


Рисунок 52 – Начальная поза

На компьютере в 3D редакторе Blender открыть файл модуля MotionSuit.blend, затем запустить Blender Game Engine и начать захват движений. В модуле для Blender Game Engine происходит обработка потока данных из кватернионов и передача измененных углов скелету 3D модели человека с учетом окружения (сцены и объектов сцены) и скелета модели. При изменении положения тела человека, на котором надет костюм захвата движения, на экране компьютера в 3D редакторе Blender должны отображаться изменения положения скелетной трехмерной модели человека в соответствии с изменениями положения тела человека.

3.2.3 Результаты тестирования

Было успешно выполнено тестирование предварительной версии программного обеспечения на прототипе костюма. Процесс тестирования программного обеспечения показал достоверность результатов работы программно-технического обеспечения. При изменении положения верхней конечности человека, на котором надет костюм захвата движения, на экране компьютера в 3D редакторе Blender отображаются изменения положения верхней конечности

скелетной трехмерной модели человека в соответствии с изменениями положения верхней конечности тела человека в костюме.

Также в 3D редакторе Blender визуально определяется высокая задержка сигнала и низкая скорость обработки потока данных (7 кадров в секунду), что позволяет сделать вывод о необходимости использования более высокопроизводительного компьютера для работы с данным программным обеспечением.

В результате тестирования программного обеспечения на прототипе костюма можно сделать следующие выводы:

1) захват движений человека, который одет в костюм, осуществляется успешно, трехмерная модель человека в 3D-редакторе Blender двигается в соответствие с движениями человека;

2) движения человека, который одет в костюм, передавались скелетной трехмерной модели человека, но с искажениями. Причиной данных искажений послужило неточное расположение датчиков на костюме: датчики должны быть расположены в соответствии с осями XYZ на 3D модели. По этой причине потребовалась калибровка расположения датчиков на костюме;

3) было выяснено, что датчики не корректно откалиброваны, поэтому в дальнейшем необходимо проводить калибровку каждого датчика;

4) в дальнейшем возможна доработка программного обеспечения для обработки данных с 16 датчиками абсолютной ориентации;

5) разработанное программное обеспечение для анимации 3D-модели человека ускоряет процесс создания анимации или разработки компьютерной игры, т. к. при ее использовании отпадет необходимость анимировать персонажи покадрово и вручную.

3.3 Анализ практической значимости результатов

Практическая значимость результатов работы оценивается по следующим критериям:

- для отработки технологии создан прототип костюма инерциального захвата движения, основанный на сети из 8 датчиков абсолютной ориентации

BNO055 (разработки фирмы Bosch), подключенных к контроллеру, который состоит из аппаратной платформы Arduino Mega и I2C-мультиплексор TCA9548A, витая пара и датчики были прикреплены к эластичному хлопчатобумажному костюму с помощью тканевых «липучек»;

- было разработано программное обеспечение для костюма инерциального захвата движения, которое состоит из двух модулей: 1) микрокод сбора данных на аппаратно-программной платформе Arduino; 2) программа интерпретации данных в 3D-редакторе Blender;

- костюм создан из общедоступных недорогих компонентов, на которые было потрачено 10 тыс. руб;

- прототип инерциального костюма осуществляет захват движения верхней части тела человека;

- возможна доработка костюма из 16 датчиков для захвата движения с нижней части тела;

- прототип костюма можно использовать для анимации персонажей в редакторе Blender для создания компьютерных игр или анимации;

- предложенная методика может ускорить процесс создания анимации или разработки компьютерной игры, т. к. при ее использовании отпадет необходимость анимировать персонажи вручную покадрово.

ЗАКЛЮЧЕНИЕ

Автором было предложено экономичное решение вопроса анимации 3D-модели человека с помощью костюма инерциального захвата движения и программного обеспечения для костюма. Предложенная методика может ускорить процесс создания анимации или разработки компьютерной игры, т. к. при ее использовании отпадет необходимость анимировать персонажи покaдрово и вручную.

Был проведен анализ предметной области существующих алгоритмов захвата движения, были изучены существующие программно-технические средств, используемых в технологии захвата движения. На основе изученного материала было найдено техническое решение инерциального костюма. Также для инерциального костюма было реализовано программное обеспечение, состоящее из двух модулей – микрокод сбора данных на аппаратно-программной платформе Arduino и программа интерпретации данных в 3D-редакторе Blender.

Была проведена оценка достоверности результатов – захват движений человека, который одет в костюм, осуществляется успешно, трехмерная модель человека в 3D-редакторе Blender двигается в соответствие с движениями человека.

Практическая значимость результатов:

- для отработки технологии создан прототип костюма из 8 датчиков для захвата движения верхней части тела человека;
- возможна доработка костюма из 15 датчиков для захвата движения с нижней части тела;
- прототип костюма можно использовать для анимации персонажей в редакторе Blender для создания компьютерных игр или видео;
- костюм создан из общедоступных недорогих компонентов, на которые было потрачено 10 тыс. руб.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Решетов К. Ю. Технология инерциального захвата движения для анимации 3D моделей исторических персонажей // Молодежь XXI века: шаг в будущее: мат. XXI регион. науч.-практ. конф. – Благовещенск: ДальГАУ, 2020. – Т.4. – С. 134-135. С. 134.
- 2 Решетов К. Ю. Анимация 3D-модели человека по технологии инерциального захвата движения // Научное сообщество студентов XXI столетия. ТЕХНИЧЕСКИЕ НАУКИ: сб. ст. по мат. ХСІХ между-нар. студ. науч.-практ. конф., 2021. – № 3(98). – С. 50-53. С. 51.
- 3 Jáuregui, D. 3D motion capture by computer vision and virtual rendering [Электронный ресурс]. – Режим доступа : https://www.researchgate.net/publication/255990108_3D_motion_capture_by_computer_vision_and_virtual_rende-ring – 10.04.2021.
- 4 Regazzoni, D, Vecchi, G, Rizzi, C, RGB cams vs RGB-D sensors: Low cost motion capture technologies performances and limitations // Journal of Manufacturing Systems, Elsevier, 2014 – Volume 33, Issue 4. – С. 719-728. С. 720.
- 5 Гайнияров И.М., Обабков И.Н., Хлебников Н.А. Метод захвата движений как средство естественного интерфейса [Электронный ресурс] // Конференция «GraphiCon 2017» 24–28 сентября 2017, Пермь. Режим доступа : <https://www.graphicon.ru/html/2017/papers/pp193-196.pdf> – 10.04.2021.
- 6 Jáuregui, D. 3D motion capture by computer vision and virtual rendering [Электронный ресурс]. – Режим доступа : https://www.researchgate.net/publication/255990108_3D_motion_capture_by_computer_vision_and_virtual_rende-ring – 10.04.2021.
- 7 Yeonkyung L., Hoon Y. Low-cost 3D motion capture system using passive optical markers and monocular vision // Optik - International Journal for Light and Electron Optics, 2017. – Volume 130. – С. 1397-1407. С. 1398.
- 8 Белоус А.Н., Рыбкин С.В. Алгоритмы функционирования и классифика-

ция систем захвата движения [Электронный ресурс] // Международный студенческий научный вестник. – 2019. – № 2.; Режим доступа : <http://eduherald.ru/ru/article/view?id=19588> – 10.04.2021.

9 Yeonkyung L., Hoon Y. Low-cost 3D motion capture system using passive optical markers and monocular vision // *Optik - International Journal for Light and Electron Optics*, 2017. – Volume 130. – С. 1397-1407. С. 1399.

10 Jáuregui, D. 3D motion capture by computer vision and virtual rendering [Электронный ресурс]. – Режим доступа : https://www.researchgate.net/publication/255990108_3D_motion_capture_by_computer_vision_and_virtual_rende-ring – 10.04.2021.

11 Gregor, M., Horejsi, P., Šimon, M. Case study: Motion capture for ergonomics. // *Innovation Vision 2020: from Regional Development Sustainability to Global Economic Growth*. – 2020. – С. 468-476. С. 469.

12 Jáuregui, D. 3D motion capture by computer vision and virtual rendering [Электронный ресурс]. – Режим доступа : https://www.researchgate.net/publication/255990108_3D_motion_capture_by_computer_vision_and_virtual_rende-ring – 10.04.2021.

13 Там же.

14 Восков Л. С., Вабищевич А. Н. Проблема беспроводного захвата движения и система для ее решения на основе программно-аппаратной платформы беспроводной сенсорной сети // *Качество. Инновации. Образование*. 2013. № 1. С. 84-91. С.87.

15 Белов Ю.С., Рыбкин С.В. Математические модели геометрических преобразований изображений в задаче анализа видеопоследовательностей // *Электронный журнал: Наука, техника и образование*. 2015. № 3 (3). С. 15-24. С. 21.

16 Там же, С. 22.

17 Нестеров А. Ю., Бурмистров А. В., Белов Ю. С. Метод определения положения объекта в задачах распознавания образов // *Электронный журнал: наука, техника и образование*. 2016. № 2 (6). С. 82-89. С 83.

- 18 Казичкина П.С., Белов Ю.С. Разработка системы классификации движения человека на основе сети Джордана // Электронный журнал: наука, техника и образование. 2017. № СВ1 (11). С. 125-132. С. 129.
- 19 Chen P., Kuang Y., Li J. Human Motion Capture Algorithm Based on Inertial Sensors // Journal of Sensors, 2016, Volume 2016, С. 1-15. С. 5.
- 20 Roetenberg D., Luinge H., Slycke P. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors // XSENS TECHNOLOGIES, 2013 VERSION APRIL 3 С. 1-9. С. 1.
- 21 Там же, С. 2.
- 22 Белоус А.Н., Рыбкин С.В. Алгоритмы функционирования и классификация систем захвата движения [Электронный ресурс] // Международный студенческий научный вестник. – 2019. – № 2.; Режим доступа : <http://eduherald.ru/ru/article/view?id=19588> – 10.04.2021.
- 23 Там же.
- 24 Roetenberg D., Luinge H., Slycke P. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors // XSENS TECHNOLOGIES, 2013 VERSION APRIL 3 С. 1-9. С. 1.
- 25 Там же, С. 1.
- 26 Zhu, R., Zhou, Z. «A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package» // IEEE Transactions on Neural Systems and Rehabilitation Engineering – 2004 – vol. 12, №2. – С. 295-302. С. 296.
- 27 Там же, С. 297.
- 28 Roetenberg D., Luinge H., Slycke P. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors // XSENS TECHNOLOGIES, 2013 VERSION APRIL 3 С. 1-9. С. 2.
- 29 Там же, С. 2.
- 30 Там же, С. 3.
- 31 Szczęsna, A., Pruszowski, P., Skurowski, P., Lach, E., Słupik, J., Pęszor, D., Paszkuta, M., Polanski, A., Wojciechowski, K., Janiak, M., Lebek, K. Inertial Motion

- Capture Costume // *Procedia Technology*. – 2017. – №27. – С. 139-140. С. 139.
- 32 Там же, С. 139.
- 33 Там же, С. 140.
- 34 Pruszowski P, Szczesna A, Polański A, Słupik J, Wojciechowski K. Adaptation Mechanism of Feedback in Quaternion Kalman Filtering for Orientation Estimation *Artificial Intelligence and Soft Computing* // Springer International Publishing. – 2015. – С.739-748. С.740.
- 35 Szczesna A, Pruszowski P, Słupik J, Pęszor D, Polański A. Evaluation of Improvement in Orientation Estimation Through the Use of the Linear Acceleration Estimation in the Body Model Man // *Machine Interactions 4*, Springer International Publishing. – 2016. – С.377-387. С.380.
- 36 Там же, С. 381.
- 37 Gregor, M., Horejsi, P., Šimon, M. Case study: Motion capture for ergonomics. // *Innovation Vision 2020: from Regional Development Sustainability to Global Economic Growth*. – 2020. – С. 468-476. С. 469.
- 38 XSENS MVN Animate [Электронный ресурс] – Режим доступа : <https://mocap-tech.ru/animate/> – 26.05.2021.
- 39 Xsens Link [Электронный ресурс] – Режим доступа : <https://mocap-tech.ru/animate/#hardware> – 26.05.2021.
- 40 XSENS MVN Animate [Электронный ресурс] – Режим доступа : <https://mocap-tech.ru/animate/> – 26.05.2021.
- 41 Там же.
- 42 Motionshadow [Электронный ресурс] Режим доступа : <https://www.motionshadow.com/> – 26.05.2021.
- 43 Там же.
- 44 Там же.
- 45 Rokoko Smartsuit Pro [Электронный ресурс] – Режим доступа : <https://www.rokoko.com/products/smartsuit-pro/tech-specs> – 26.05.2021.
- 46 Там же.

- 47 Rokoko Smartsuit Pro [Электронный ресурс] – Режим доступа : <https://www.rokoko.com/products/smartsuit-pro/tech-specs> – 26.05.2021.
- 48 XSENS MVN Animate PRO [Электронный ресурс] – Режим доступа : <https://www.xsens.com/products/mvn-animate> – 26.05.2021.
- 49 Там же.
- 50 Там же.
- 51 Motionshadow [Электронный ресурс] Режим доступа : <https://www.motionshadow.com/> – 26.05.2021.
- 52 Там же.
- 53 Там же.
- 54 Rokoko Smartsuit Pro [Электронный ресурс] – Режим доступа : <https://www.rokoko.com/products/smartsuit-pro/tech-specs> – 26.05.2021.
- 55 Там же.
- 56 Bewegungsfelder [Электронный ресурс] – Режим доступа : <https://github.com/herzig/bewegungsfelder> – 26.05.2021.
- 57 Там же.
- 58 Motiosuit [Электронный ресурс] Режим доступа : <https://github.com/alvaroferran/motiosuit> – 26.05.2021.
- 59 Blender [Электронный ресурс]. – Режим доступа : <https://www.blender.org/> – 26.05.2021.
- 60 Там же.
- 61 Blender Game Engine. Game Logic Screen Layout [Электронный ресурс]. – Режим доступа : https://docs.blender.org/manual/ru/2.79/game_engine/screen_layout.html – 26.05.2021.
- 62 Там же.
- 63 Там же.
- 64 MakeHuman Community [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://www.makehumancommunity.org/> – 10.03.2021.
- 65 Там же.

- 66 MakeHuman Community [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://www.makehumancommunity.org/> – 10.03.2021.
- 67 Data sheet BNO055 [Электронный ресурс] Bosch Sensortec – Режим доступа : https://www.bosch-sensortec.com/media/boschsensortec/downloads/smart_sensors_1/bno055/bst-bno055-ds000.pdf – 10.04.2021.
- 68 Там же.
- 69 Там же.
- 70 Там же.
- 71 Data sheet TCA9548A [Электронный ресурс] Texas Instruments – Режим доступа : <https://www.ti.com/lit/ds/symlink/tca9548a.pdf> – 10.04.2021.
- 72 Там же.
- 73 Arduino MEGA2560 [Электронный ресурс]. – Режим доступа : <https://store.arduino.cc/usa/mega-2560-r3> – 26.05.2021.
- 74 Там же.
- 75 Arduino MEGA2560 Guide [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Guide/ArduinoMega2560> – 26.05.2021.
- 76 Там же.
- 77 Arduino Environment [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Guide/Environment> – 26.05.2021.
- 78 Там же.
- 79 Han, F., Reily, B., Hoff, W., Zhang, H. Space-time representation of people based on 3D skeletal data: A review // Computer Vision and Image Understanding. – 2017. – №158. – С. 85-105. С. 91.
- 80 Pruszowski P, Szczesna A, Polański A, Słupik J, Wojciechowski K. Adaptation Mechanism of Feedback in Quaternion Kalman Filtering for Orientation Estimation Artificial Intelligence and Soft Computing // Springer International Publishing. – 2015. – С.739-748. С.740.
- 81 Szczęsna, A., Pruszowski, P., Skurowski, P., Lach, E., Słupik, J., Pęszor, D., Paszkuta, M., Polanski, A., Wojciechowski, K., Janiak, M., Lebek, K. Inertial Motion

Capture Costume // Procedia Technology. – 2017. – №27. – С. 139-140. С. 139.

82 Blender. System Requirements [Электронный ресурс]. – Режим доступа : <https://www.blender.org/download/requirements/> – 26.05.2021.

83 Решетов К. Ю. Технология инерциального захвата движения для анимации 3D моделей исторических персонажей // Молодежь XXI века: шаг в будущее: мат. XXI регион. науч.-практ. конф. – Благовещенск: ДальГАУ, 2020. – Т.4. – С. 134-135. С. 135.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Авербух В.Л., Авербух Н.В., Стародубцев И.С., Тоболин Д.Ю. Использование жестовых интерфейсов при взаимодействии с объектами // Научная перспектива. № 10 (56) / 2014. Стр. 57-66.
- 2 Алфимцев А. Н., Демин Н. А. Захват и отслеживание удаленных объектов в видеопотоке //Инженерный журнал: наука и инновации. – 2013. – №. 11. – С. 23.
- 3 Белов Ю.С., Рыбкин С.В. Математические модели геометрических преобразований изображений в задаче анализа видеопоследовательностей // Электронный журнал: Наука, техника и образование. 2015. № 3 (3). С. 15-24.
- 4 Белоус А.Н., Рыбкин С.В. Алгоритмы функционирования и классификация систем захвата движения [Электронный ресурс] // Международный студенческий научный вестник. – 2019. – № 2.; Режим доступа : <http://eduherald.ru/ru/article/view?id=19588> – 17.12.2019.
- 5 Гайниязов И.М., Обабков И.Н., Хлебников Н.А. Метод захвата движений как средство естественного интерфейса [Электронный ресурс] // Конференция «GraphiCon 2017» 24–28 сентября 2017, Пермь, Россия. Режим доступа : <https://www.graphicon.ru/html/2017/papers/pp193-196.pdf>
- 6 Датчик абсолютной ориентации BNO-055 [Электронный ресурс] Режим доступа : <https://www.zizibot.ru/directory/sensor/bno055/> – 17.03.2021.
- 7 Иванов Д. С., Овчинников М. Ю. Использование одноосного гироскопа для определения ориентации макета в лабораторных условиях //Препринты Института прикладной математики им. МВ Келдыша РАН. – 2008. – №. 0. – С. 11-32.
- 8 Крючков Б.И., Усов В.М., Михайлюк М.В. Применение методов захвата движений в космической робототехнике при инженерно-психологическом проектировании человеко-машинного взаимодействия // Пилотируемые полеты в космос. N 4 (21), 2016. Стр. 57-78.

- 9 Решетов К. Ю. Анимация 3D-модели человека по технологии инерциального захвата движения [Электронный ресурс] // Научное сообщество студентов XXI столетия. ТЕХНИЧЕСКИЕ НАУКИ: сб. ст. по мат. ХСIX междунар. студ. науч.-практ. конф. № 3(98). – С. 50-53. Режим доступа : [https://sibac.info/archive/technic/3\(98\).pdf](https://sibac.info/archive/technic/3(98).pdf) – 24.04.2021.
- 10 Решетов К. Ю. Программное обеспечение костюма инерциального захвата движения // День науки: мат. ХХIX науч. конф. Амурского гос. ун-та – Благовещенск: АмГУ, 2020. – С. 39-40.
- 11 Решетов К. Ю. Технология инерциального захвата движения для анимации 3D моделей исторических персонажей // Молодежь XXI века: шаг в будущее: мат. ХХI регион. науч.-практ. конф. – Благовещенск: ДальГАУ, 2020. – Т.4. – С. 134-135.
- 12 Стародубцев И.С. Захват движения на основе иерархического гибкого скелета // Научно-технический вестник Поволжья. Направления: физико-математические науки, химические науки, технические науки. N 1. 2014. Стр. 159-162.
- 13 Arduino Environment [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Guide/Environment> – 26.05.2021.
- 14 Arduino MEGA2560 [Электронный ресурс]. – Режим доступа : <https://store.arduino.cc/usa/mega-2560-r3> – 26.05.2021.
- 15 Arduino MEGA2560 Guide [Электронный ресурс]. – Режим доступа : <https://www.arduino.cc/en/Guide/ArduinoMega2560> – 26.05.2021.
- 16 Bewegungsfelder [Электронный ресурс] – Режим доступа : <https://github.com/herzig/bewegungsfelder> – 26.05.2021.
- 17 Blender [Электронный ресурс]. – Режим доступа : <https://www.blender.org/> – 26.05.2021.
- 18 Blender. System Requirements [Электронный ресурс]. – Режим доступа : <https://www.blender.org/download/requirements/> – 26.05.2021.
- 19 Blender Game Engine. Game Logic Screen Layout [Электронный ресурс]. –

- Режим доступа : https://docs.blender.org/manual/ru/2.79/game_engine/screen_layout.html – 26.05.2021.
- 20 Chen P., Kuang Y., Li J. Human Motion Capture Algorithm Based on Inertial Sensors // Journal of Sensors, 2016, Volume 2016, С. 1-15.
- 21 Data sheet BNO055 [Электронный ресурс] Bosch Sensortec – Режим доступа : https://www.bosch-sensortec.com/media/boschsensortec/downloads/smart_sensors_1/bno055/bst-bno055-ds000.pdf – 10.04.2021.
- 22 Data sheet TCA9548A [Электронный ресурс] Texas Instruments – Режим доступа : <https://www.ti.com/lit/ds/symlink/tca9548a.pdf> – 10.04.2021.
- 23 Gregor, M., Horejsi, P., Šimon, M. (2015). Case study: Motion capture for ergonomics. // Innovation Vision 2020: from Regional Development Sustainability to Global Economic Growth. – 2020. – С. 468-476.
- 24 Ezekwe C. D., Geiger W., Ohms T. A 3-axis open-loop gyroscope with demodulation phase error correction //Solid-State Circuits Conference-(ISSCC), 2015 IEEE International. – IEEE, 2015. – С. 1-3.
- 25 Jáuregui, D. 3D motion capture by computer vision and virtual rendering [Электронный ресурс]. – 2011. – Режим доступа : https://www.researchgate.net/publication/255990108_3D_motion_capture_by_computer_vision_and_virtual_rendering – 17.12.2020.
- 26 Han, F., Reily, B., Hoff, W., Zhang, H. Space-time representation of people based on 3D skeletal data: A review // Computer Vision and Image Understanding. – 2017. – №158. – С. 85-105.
- 27 MakeHuman Community [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://www.makehumancommunity.org/> – 10.03.2021.
- 28 Motionshadow [Электронный ресурс] Режим доступа : <https://www.motionshadow.com/> – 26.05.2021.
- 29 Motiosuit [Электронный ресурс] Режим доступа : <https://github.com/alvaroferran/motiosuit> – 26.05.2021.
- 30 Pruszowski P, Szczesna A, Polański A, Słupik J, Wojciechowski K. Adaptation

Mechanism of Feedback in Quaternion Kalman Filtering for Orientation Estimation Artificial Intelligence and Soft Computing // Springer International Publishing. – 2015. – С.739-748.

31 Regazzoni, D, Vecchi, G, Rizzi, C, RGB cams vs RGB-D sensors: Low cost motion capture technologies performances and limitations // Journal of Manufacturing Systems, Elsevier, 2014 – Volume 33, Issue 4. – С. 719-728.

32 Roetenberg, D., Luinge, H., Slycke, P. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors. // Xsens Motion Technol. BV Tech. Rep. 3. XSENS TECHNOLOGIES - VERSION APRIL 3, 2013. С. 1-9.

33 Rokoko Smartsuit Pro [Электронный ресурс] – Режим доступа : <https://www.rokoko.com/products/smartsuit-pro/tech-specs> – 26.05.2021.

34 Szczęsna, A., Pruszowski, P., Skurowski, P., Lach, E., Słupik, J., Pęszor, D., Paszkuta, M., Polanski, A., Wojciechowski, K., Janiak, M., Lebek, K. Inertial Motion Capture Costume // Procedia Technology. – 2017. – №27. – С. 139-140.

35 Szczęsna A, Pruszowski P, Słupik J, Pęszor D, Polański A. Evaluation of Improvement in Orientation Estimation Through the Use of the Linear Acceleration Estimation in the Body Model Man // Machine Interactions 4, Springer International Publishing. – 2016. – С.377-387.

36 Wang J, Liu Z, Wu Y, Yuan J. Learning Actionlet Ensemble for 3D Human Action Recognition // IEEE Trans Pattern Anal Mach Intell. 2014 May;36(5): С. 914-927.

37 Xsens Link [Электронный ресурс] – Режим доступа : <https://mocap-tech.ru/animate/#hardware> – 26.05.2021.

38 XSENS MVN Animate [Электронный ресурс] – Режим доступа : <https://mocap-tech.ru/animate/> – 26.05.2021.

39 XSENS MVN Animate PRO [Электронный ресурс] – Режим доступа : <https://www.xsens.com/products/mvn-animate> – 26.05.2021.

40 Yabukami S., Kikuchi H., Yamaguchi M., Arai K.I., Takahashi K., Itagaki A., Wako N. Motion capture system of magnetic markers using three-axial magnetic

field sensor // IEEE Transactions on Magnetics, 2000, Volume: 36, Issue: 5, C.3646 – 3648

41 Yeonkyung L., Hoon Y. Low-cost 3D motion capture system using passive optical markers and monocular vision // Optik - International Journal for Light and Electron Optics, 2017. – Volume 130. – C. 1397-1407.

42 Zhu, R., Zhou, Z. «A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package» // IEEE Transactions on Neural Systems and Rehabilitation Engineering – 2004 – vol. 12, №2. – C. 295-302.