СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ

3D – 3-Dimensional

URL – Uniform Resource Locator

ПО – Программное Обеспечение

CCD – Cyclic Coordinate Descent

FABRIK – Forward and Backward Reach Inverse Kinematics

DLS – Damped least squares

SVD – Singular value decomposition

SVD-DLS – Pseudo-inverse damped least squares

SDLS – Selectively damped least squares

iTaSC – Task Specification using Constraints

FK – Forward Kinematics

IK – Inversive Kinematics

Motion Capture – Захват движений

Root – Корневой элемент

Ragdoll – Кукла

ВВЕДЕНИЕ

На момент исследования, тему процедурной генерации анимаций активно изучают, ведь с текущими вычислительными мощностями появляется возможность освободить аниматоров от части работы и автоматизировать этот процесс алгоритмически. К тому же в процессе разработки темы была выявлена проблемная область с переносом анимаций с видео. Этой области было решено выделить больше всего внимания при проведении исследования.

Во время активной автоматизации всех возможных операций с помощью алгоритмов и нейронных сетей, ниша анимаций становится одним из самых очевидных мест в видеоигровой индустрии, где требовалась бы данная автоматизация. Каждая отдельная анимация отнимает немало времени, особенно когда дело касается боевых систем, в которых от аниматора помимо глубоких познаний в физике ударов и человеческого тела в целом требуется также знание специфики конкретной боевой системы. В данный момент существует немало сервисов и площадок, на которых можно взять готовые анимации, а затем перенести их на интересующего персонажа, с помощью различных инструментов, в том числе включенных в состав игровых движков. К сожалению, наборы анимаций там до сих пор очень сильно ограничены и зачастую не получается найти интересующую анимацию.

При наличие всевозможных общедоступных анимаций и инструментов для их создания, все еще невозможно быстро создать и имплементировать определенную анимацию в игровой проект или фильм. Все дело в том, что каждый раз решается проблема создания определенных качественных движений с нуля. Большие студии не применяют готовую продукцию, так как время на переработку скачанных анимаций может занять больше времени чем создание своих анимаций с нуля. Для этого, например, применяются

специальные костюмы и системы, которые записывают движения человека. Затем, уже записанные анимации перерабатываются, очищаются от шумов и применяются в проекте. Стоит отметить, что дороговизна подобных систем не позволяет компаниям поменьше применять подобные технологии в своих играх, за счет чего иногда качество страдает.

В данный момент времени активно разрабатываются процедурные анимационные подходы, основанные на нейронных сетях, которые позволяют считывать движения человека прямо с обычного видео. При этом видео может быть записано на камеру обычного телефон. Но до сих пор существуют узкие места в которых нейронные сети по какой-либо причине не могут повторить движения человека. В первую очередь это происходит из-за недостаточности данных на видео. Например, когда человека перекрывает другой человек или предмет. В данных случаях нейронные сети с трудом могут восстановить ход действий человеческого скелета в невидимой области.

Главной проблемой является то, что среди существующих методов считывания движений человека с видео записи нет достаточных для восстановления качественно и в полном объеме неполноценных данных, записанных в неблагоприятных условиях или с перекрытиями объекта слежения.

Целью работы является создание и обучения нейронной сети на основе существующих подходов к восстановлению неполноценных видео данных, которая бы решала проблему предсказания движения человека за перекрывающими физическими объектами или другими людьми для повышения качества выходных анимаций.

Задачи работы:

- поиск и оценка видео данных по интересующему боевому стилю;
- описание методов и подходов;
- проектирование алгоритма генерации анимаций;
- тестирование на примере боевого стиля.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ методов анимации

На текущий момент выделяют следующие типы 3D анимаций, использующихся в производстве кино и игровой индустрии [13][14]:

- Трансформация или вертексная анимация – анимация объектов трехмерной сетки модели (вершин, ребер, граней) с использованием простейших инструментов перемещения, вращения и масштабирования;
- Скелетная анимация – анимация, основанная на анимационном скелете, который представляет из себя иерархическую связь из множества объектов (костей) [19], положение каждого из которых зависит от своего иерархического предка;
- Захват движения (motion capture) – анимация, основанная на захвате движений реального объекта при снятии показателей с датчиков и специализированного оборудования;
- Ротоскопирование – покадровый перенос движений с видео или серии изображений реального объекта;
- Симуляция движения – анимирование происходит автоматически за счет физических правил и функций некого тела [18];
- Процедурная анимация – анимация объектов сцены, выполняющаяся на основе каких-либо процедур или функций. Часто может применяться как для моделирования системы частиц, таких как вода, дым и т.д., но также данный подход применим к скелетной анимации.

Наиболее популярным и общепринятым подходом к анимации человека является скелетная анимация. Скелет в данном случае являет собой упрощенный скелет человека с основными сочленениями. При этом в редакторах можно гибко настраивать все параметры, определяя таким

образом степени свободы каждого отдельного сустава, а также набор вершин, которые будут подвергаться изменениям во время сгиба. Отдельно стоит упомянуть про веса вершин, это понятие определяет силу воздействия поворота сустава на конкретную вершину. В самом начале веса всех вершин для определённого сустава равны 0, что означает, что они не будут подвергаться изменениям при его повороте. Далее аниматор может выставить для каждой интересующей вершины числа от 0 до 1. Единица в данном случае будет означать реакцию в полной мере на поворот сустава.

Скелетная анимация в свою очередь также подразделяется на несколько типов:

- Классическая покадровая, когда аниматор выставляет все кости персонажа в нужное положение ориентируясь на видео или другой источник для каждого кадра анимации [24]. Именно из-за этого на каждую отдельную анимацию требуется огромное количество времени и сил разработчика;
- Процедурная анимация в реальном времени [4]. Данный тип отличается гибкостью и возможностью в реальном времени контролировать положение костей в зависимости от окружающей среды и обстоятельств. Минусом является то, что для данного способа необходимо написать алгоритм.

Процедурную анимацию также принято разделять на генерируемою (заранее подготовленную) и процедурную анимацию в реальном времени [4]. В обоих случая анимации создаются программным кодом и набором функций [11]. Ключевое различие заключается в том, что в первом варианте алгоритм генерирует конечный файл анимации, которую впоследствии можно загрузить в любой редактор. Обычно сгенерировать анимацию можно изменяя некие входные параметры. Что касается второго варианта, анимации там создаются в процессе работы. Результат анимации зависит от положения анимируемого скелета, расстояния до определенной цели и других параметров, заложенных в конкретный алгоритм.

Что касается процедурной анимации в целом, то сначала стоит поговорить о процедурной анимации «марионетки» [12]. Этот тип анимации основан на инверсной кинематике. Другими словами, правильно ограничивая движения человеческого тела, можно создавать реалистичные модели поведения при падениях и так далее. Один из самых простых типов анимации — это создание процедурных анимаций с использованием физики тряпичных кукол или “ragdoll physics”, как и говорилось ранее.

Идея заключается в том, чтобы управлять телом гуманоида, всеми его суставами и всеми связями с суставами, чтобы воссоздать степени свободы реального человека. Просто используя физику твердого тела и ограничения на суставы, можно смоделировать падение конечностей и всех суставов. Это не только экономит деньги на «анимации смерти», но и позволяет создавать персонажей, которые реалистично падают и взаимодействуют с окружающей средой. Такую задачу практически невозможно решить с помощью простого стандартного набора анимации, каким бы точным он ни был.

Метод процедурной анимации, основанный на симуляции твердого тела, является дополнением метода ragdoll-анимации и в отличии от него включает в себя возможность контролировать движения. В основе все также лежит ragdoll-физика, однако ключевые звенья «скелета» персонажа управляются кодом, а промежуточные звенья все так же подвержены физике

Следующей составляющей процедурных анимаций является инверсивная кинематика [5]. Инверсная кинематика зародилась в области робототехники как задача перемещения кинематического манипулятора с определенные степени свободы для определенной цели, позже нашедшие свое применение в компьютерной графике. Положение конечного эффектора можно описать как функцию степени свободы следующим образом.

Для заданных n соединений полная конфигурация мультитела задается скалярами $\theta 1$, ..., θn, называемыми соединительными углами. Определенные k точек звеньев идентифицируются как конечные эффекторы, а их положения

обозначаются c1, ..., ск. Каждое положение конечного эффектора si является функцией углов соединения.

Инверсная кинематика используется, когда необходимо создать анимацию, на которую накладываются не только ограничения, связанные с гравитацией и физическими свойствами самих суставов [26], но и когда в это вмешиваются другие внешние факторы. Например, может быть неровный пол или ступеньки, на которых модель персонажа должна стоять ровно, но сгибая ноги. Другими словами, с помощью алгоритма мы можем создать дополнительные ограничения, обуславливающие траектории, по которым скелетные элементы, чаще всего конечности, могут двигаться под воздействием окружающих обстоятельств.

Также IK может быть использована для решения многих задач, не связанных с движениями ног. Конечно, наиболее распространенными задачами такого типа являются естественные движения гуманоидных персонажей по определенным неровностям, например ямам [9], но этот подход прекрасно применим и в условиях, когда мы не можем точно знать, в каком положении окажутся руки нашего виртуального скалолаза или же когда в игре используются стулья разной высоты и на каждый из них персонаж должен садиться красиво. Вместо того чтобы использовать возможности аниматоров, разработчики просто указывают цель, которую должна достичь рука или нога. Все остальное делает обратная кинематика.

Кроме методов физической анимации существуют следующие методы, активно развивающиеся на сегодняшний день:

- генерация анимации по ключевым кадрам;
- процедурная анимация с использованием нейросетей.

Стоит подробнее рассказать об анимации с помощью нейронных сетей, ведь нельзя недооценивать значимость и возможности данного метода.

В новейших исследованиях используются несколько инновационных подходов к формированию нейронной сети [2]. Во-первых, в отличие от классического физического подхода, когда под каждую отдельную задачу

нейронная сеть обучается с нуля, здесь используется комплексный поход. Данный способ позволяет дообучать модель, чтобы “научить” ее использовать конкретные необходимые приемы и навыки. Это возможно благодаря схеме обучения.

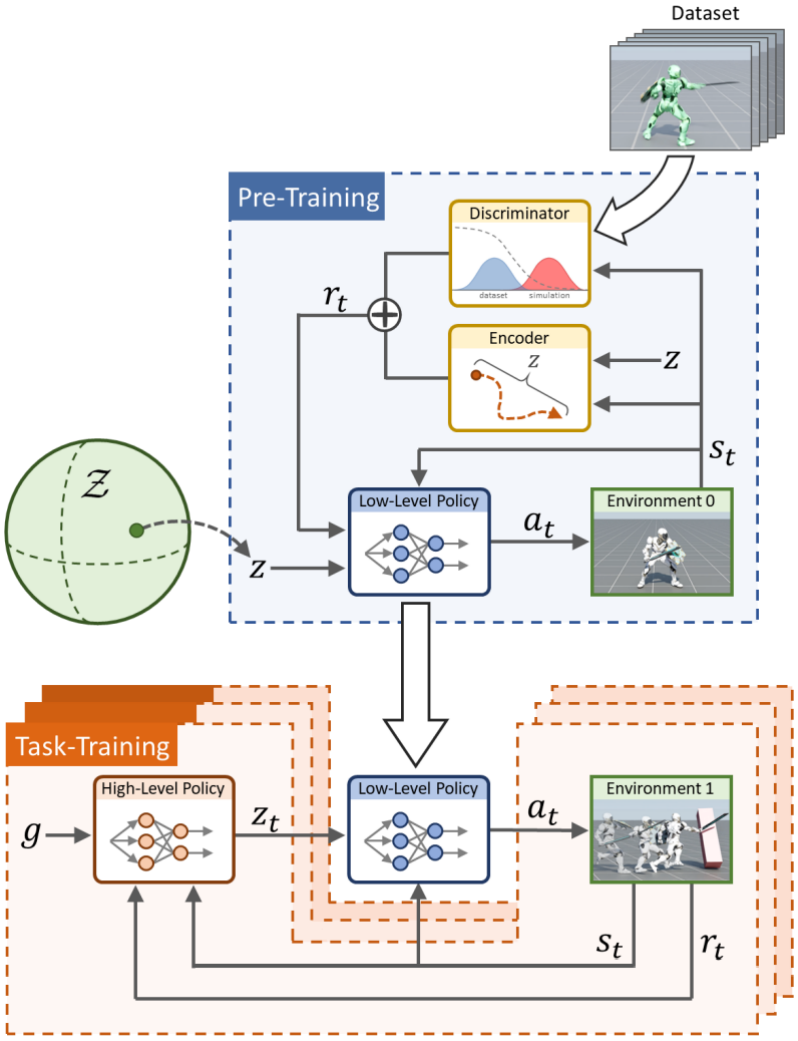

Рисунок 1 – Обучение нейронной сети

Как видно на рисунке, на вход модели при обучении поступает сразу информация об уже изученных навыках (z), а также массив видеоданных. Анализируя полученные данные и добавляя их у к уже изученным, получается набор низкоуровневых контроллеров, каждый их, которых отвечает за простую задачу или навык, например ходьба вперед. При этом обучение также производится необычным путем. Симуляции начинаются из случайных временных точек навыка, таких образом способствуя лучшей

генерации реакций и достижения цели. Этот подход сильно отличается от классического, когда симуляции всегда начинаются из стартовой точки навыка. В процессе обучения используется система вознаграждений, благодаря которой избирательно определяются самые подходящие исходы.

Таким образом, модели обучаются с использованием комбинации цели обучения с использованием имитации и некой цели. Состязательное имитационное обучение позволяет моделям имитировать реалистичные движения на основе предоставленных пользователем данных, в то время как методы обучения с подкреплением без присмотра позволяют модели изучать представления навыков, которые становятся более управляемыми.

Далее происходит обучение высокоуровневых контроллеров, которые необходимы для применения низкоуровневых контроллеров в нужные моменты времени. Высокоуровневые контроллеры решают, когда именно и в какой ситуации применять конкретный низкоуровневый контроллер. Например, высокоуровневый контроллер может управлять передвижением персонажа, он в нужные моменты вызывает котроллеры, которые отвечают за более простые действия как движение вперед, назад или вправо и влево.

Модели не обязательно должны точно соответствовать какому-либо конкретному клипу движения в наборе данных. Вместо этого данной научной работы состоит в том, чтобы выявить разнообразный и разносторонний набор навыков, который демонстрирует общие характеристики данных о движении. Затем на этапе дообучения низкоуровневый контроллер повторно используется для выполнения новых задач путем обучения высокоуровневого контроллера для конкретной задачи или навыка. Высокоуровневые контроллеры обучены под конкретную задачу, поэтому их нужно переобучать каждый раз при изменении цели, но делать это можно, без использования каких-либо внешних данных, достаточно набора низкоуровневых контроллеров. Также благодаря тому, что низкоуровневые контроллеры обучаются с помощью набора реальных данных о движениях, это ведет к реалистичности движений в любой момент времени в целом.

1.2 Алгоритмы обратной кинематики

Помимо обратной или инверсивной кинематики, существует также и прямая кинематика [10], она выполняет противоположную логическую задачу. С помощью нее можно определить положение конечной точки цепи после манипуляции со всеми костями.

Прямая кинематика (FK) как обратная чаще всего применяются к некоторому скелету. Скелет в трехмерном моделировании – это объект, состоящий из костей, отрезков в пространстве, которые олицетворяют человеческие кости. Каждый такой отрезок отвечает за определённую группу полигонов модели, к которой привязывается. На концах кости находятся точки – суставы или сочленения. Они в свою очередь, как и следует из названия выполняют поворотную функцию, именно относительно них кость может поворачиваться в пространстве.

Кости могут быть опционально соединены между собой, для обратной и прямой кинематики, это обязательное условие. Соединяясь вместе через суставы, кости образуют цепи, по котором мы уже можем определять позиции соседних костей. Таким образом цепи сами по себе имеют отношения родитель-ребенок, внутри которых в редакторе пользователь может определять тип воздействия на соответственного соседа в зависимости от типа движения. Например, самым распространённым типом связи является наследование, во время которого, на примере цепи руки, при повороте плеча, за ним также соответственно следует и предплечье, не ломая структуру конечности.

Именно это и являет собой суть прямой кинематики, все наследники выставляются в соответствии с их родителями, беря за основу сустав, связующих их с предком. То есть на потомка будет влиять как поворот, так и любое перемещение родителя. Это также будет работать и гораздо глубже, например при сдвиге таза, будет также сдвигаться и кисть, так как в целом скелет является связанной единицей.

Главное отличие прямой кинематики от инверсивной именно фактор наследования, описанный выше. Для инверсивной кинематики будет правильнее сказать, что перемещение или поворот компонентов наследников влечет за собой соответствующие изменения компонентов предков. Например, в ситуации, когда мы хотим расположить ступню модели дальше фактической длины ноги, соответствующим действием мы так же повлияем на тазобедренный сустав, который будет двигаться за полностью вытянутой ногой.

С точки зрения теории, в инверсивной кинематике дочерний компонент, влияющий на родительские, называется эффектором. Если про него однозначно можно сказать, что он завершает цепочку из других эффекторов, он называется конечным эффектором. Если в цепочке из нескольких эффектором был перемещен или повернут не конечный эффектор, это событие будет просчитано как случай прямой кинематики. Другими словами, будет произведено воздействие на дочерние элементы по логичным правилам.

Частным случаем инверсивной кинематики является робо-рука, или если мы касаемся трехмерных моделей, то, к примеру, это рука человека. Практической задачей является подведение руки таким образом, чтобы модель или робот рукой схватили стакан со стола. В данной задаче мы знаем конечную точку положения условной кисти руки, ну или ее кости. Алгоритм в данном случае должен уметь просчитать положения остальных костей руки относительно кисти, также желательно, чтобы проводилось сглаживание, то есть не было случаев, при которых одно кость вытянута в направление стакана, а другие две компенсируют расстояние, согнувшись под острым углом относительно друг друга. При этом тело человека не должно наклоняться или ка-либо участвовать в процессе если длины руки достаточно [20].

Для анимации ударных движений больше всего подходит именно IK анимация, так как в данном случае известна конечная точка движения

последнего звена руки – точка нанесения удара. 23 На сегодняшний день существует множество алгоритмов обратной кинематики и их модификации:

1. Алгоритмы Якобиана. Эти методы предлагают линейное приближение к задаче IK, итеративно вычисляя оценочное значение для изменения конфигурации полной цепочки таким образом, чтобы это приближало конечный эффектор к целевому положению на каждом шаге. Это приближение первого порядка, означающее, что каждый сустав считается независимым от остальных: когда сустав изменяется, все его дочерние сегменты рассматриваются как единая кость. К таким методам относятся:
    1.1. Jacobian transpose – транспонирование Якоби;
    1.2. Jacobian pseudo-inverse – псевдообратный затухающий метод Якоби;
    1.3. Damped least squares (DLS) – – метод наименьших квадратов с демпфированием;
    1.4. Singular value decomposition (SVD) – разложение по сингулярным значениям;
    1.5. Pseudo-inverse damped least squares (SVD-DLS) – псевдообратный затухающий метод наименьших квадратов;
    1.6. Selectively damped least squares (SDLS) – затухающий метод выборочных квадратов;
    1.7. Gauss-Seidel – метод Гаусса-Зейделя;
    1.8. Task Specification using Constraints (iTaSC).
2. Методы Ньютона. В отличие от решения первого порядка в методах Якобиана, это решение основано на решении разложение Тейлора второго порядка по $f(x + \sigma)$.
3. Эвристические алгоритмы. Следующие решения реализуют простые способы решения проблемы IK без необходимости сложных уравнения и вычисления. Они имеют низкие вычислительные затраты, поэтому обычно нахождение конечной

позы очень быстро. Они действительно хорошо справляются с простыми задачами, но могут привести к неестественным движениям и жестикуляции при более сложных артикулированных фигурах. К таким алгоритмам относятся:

3.1. FABRIK (Forward and Backward Reach Inverse Kinematics) – метод прямого и обратного следования [15];

3.2. CCD (Cyclic Coordinate Descent) – циклический координатный спуск.

Алгоритмы Якобиана [21] позволяют получить более сглаженные позы, однако, большинство этих подходов страдают от высокой вычислительной стоимости и времени вычислений. Ниже представлены сравнения времени работы самых часто применяемых алгоритмов.

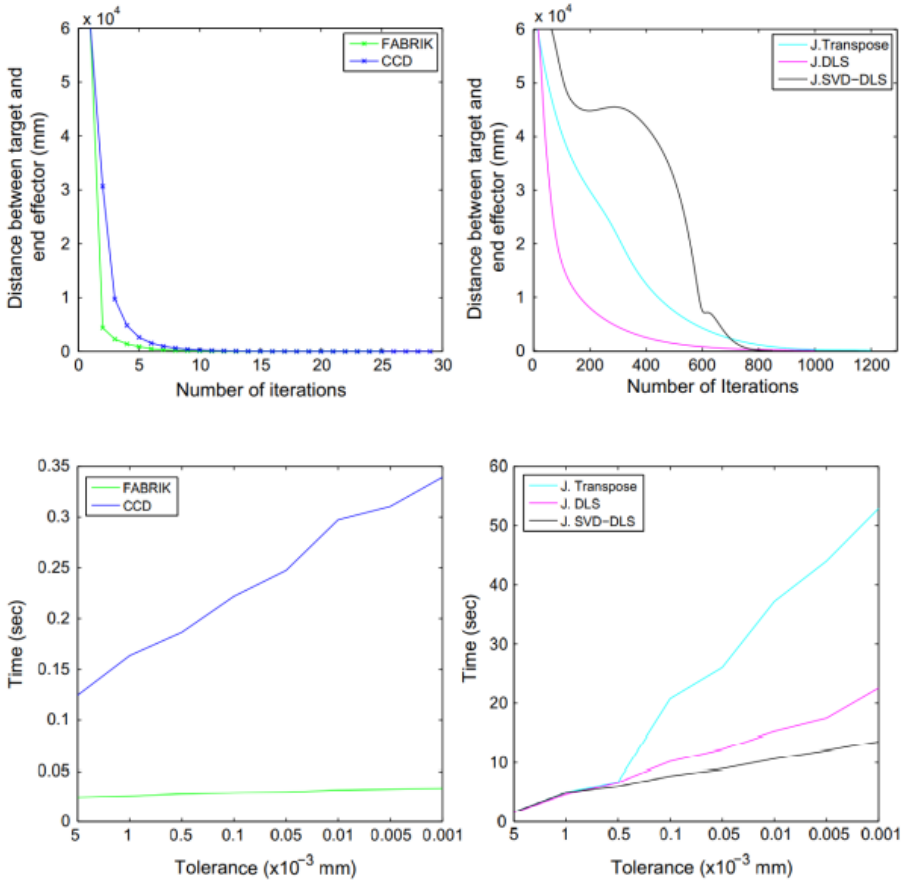

Рисунок 2 – Сравнение времени работы алгоритмов обратной кинематики

Исходя из полученной информации было принято решение, использовать один из Эвристических алгоритмов, так как одной из главных задач процедурной анимации является скорость ее работы. Для этого необходимо рассмотреть каждый из алгоритмов более подробно.

Алгоритм FABRIK находит положение каждого соединения, не используя при этом матрицы [14]. Таким образом, он сходится за несколько итераций, но что самое главное имеет низкие вычислительные затраты [23]. Данный алгоритм легко модифицируется и дорабатывается под необходимость благодаря своей простоте. Также данный алгоритм отличается реалистичностью поз получаемых конечностей.

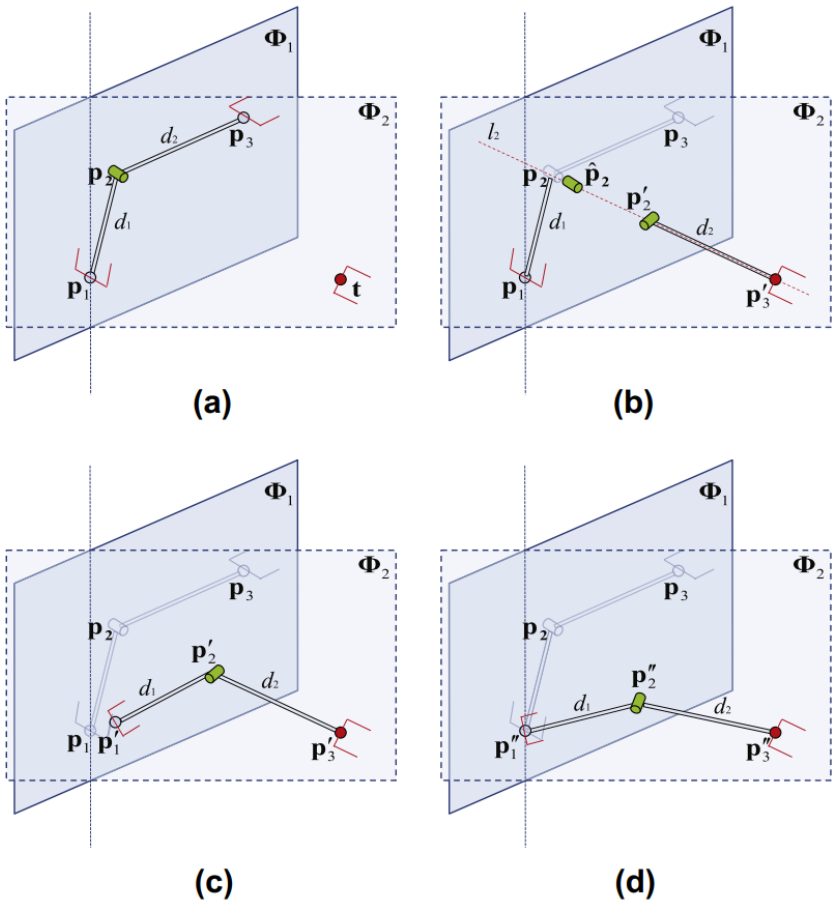

Рисунок 3 – Пример полного цикла работы алгоритма FABRIK

Одним из самых популярных методов IK является CCD или алгоритм циклического координатного спуска. Алгоритм CCD использует

эвристический подход для нахождения решение путем итеративного вращения звеньев так, чтобы конечный эффектор приближался к цели. На каждой итерации выполняется последовательность поворотов звеньев i, начиная с конечного эффектора к корню (root элементу), стараясь свести к минимуму угол θi между вектором от звена к рабочему органу и вектором к цели.

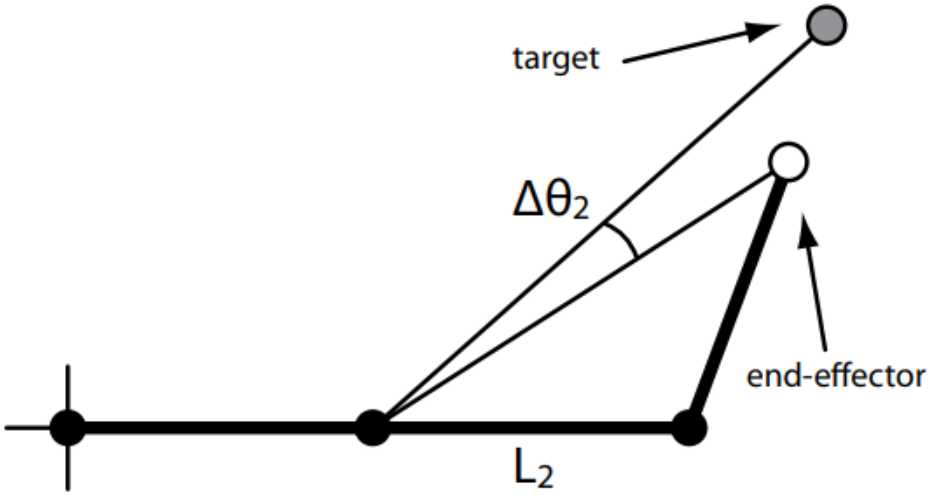

Рисунок 4 – Пример итерации работы алгоритма CCD

В силу различия решаемых задач обратной кинематики, описанные выше алгоритмы могут быть применены к созданию разных типов движения. Предполагается, что, комбинируя различные алгоритмы обратной кинематики в разные моменты удара и для различных звеньев анимационного скелета, можно добиться реалистичных движений.

Выбор алгоритма будет напрямую зависеть от получаемого после визуальной проверки получаемых анимаций для каждого из подходов [6][7][25]. Таким образом, с каждой итерацией такого подхода в итоге можно будет добиться реалистичных движений во все моменты боя. Отдельной сложностью будет реализация захватов, так как во время них очень важна плавность позы, чтобы физические модели не наслаивались друг на друга.

1.3 Биомеханика тела во время боя

Говоря о биомеханике тела, сильно влияющей на положение человеческого тела во время ударов, в первую очередь стоит упомянуть о силах, действующих на тело человека в целом. Данные силы оказывают воздействие, которое сложно переоценить, ведь от них зависит устойчивость и то, с какой силой будет нанесен удар. К этим силам относятся:

- сила реакции опоры;
- сила притяжения;
- внешние силы;
- сила внутреннего напряжения.

Данные силы в купе формируют понимание о том, в каком именно состоянии в данный момент находится тело человека. Например, из указанных показателей можно однозначно понять, стоит человек на месте или же двигается, защищается или наносит удар.

Согласно теории биомеханики главными типами состояния тела является динамическое и статическое. Данные состояния противоположны друг другу, а вместе олицетворяют равенство. Итак, если речь идет о статическом состоянии, то сразу можно точно сказать, что тело человека в данный момент направлено на сохранение энергии. Данному состоянию гораздо больше характерны прямые и ломанные положения конечностей и системы в целом. В науке данное состояние описывают фигурой треугольник или пирамидой. Это объясняется тем, что человек в первую очередь хочет сохранить целостность равновесия анатомической системы за счет добавления дополнительных точек опоры или уменьшения высоты тела за счет сгибов. Ситуация с динамическим состоянием обратная. Для этого типа гораздо больше характерны плавные кривые и неустойчивые состояния. Обычно динамику сравнивают с шаром или в отдельных случаях с пирамидой, стоящей на своей вершине. Такое состояние противоположно устойчивому и в данном случае речь не идет о сохранении энергии.

В боевых искусствах существует строгое разделение на типы единоборств, исходя из физиологического строения тела. Айкидо предполагается для людей восточного типа сложения, так как в нём главную роль играют возможность занижения центра тяжести и высокая вертикальная устойчивость при исполнении техник высокой динамики. Бокатор же объединяет динамическую и статическую модель. Дело в том, что во время защиты боец должен крепко стоять на ногах, чтобы его не уронили на землю, а в моменты атак центр тяжести может кардинально перемещаться, ведь речь идет о захватах, во время которых боец держится за оппонента руками или ногами.

Правила устойчивости:

1. Чем больше опора и ниже центр тяжести, тем более устойчива пирамида.
    1.1 Чем меньше опора и выше центр тяжести, тем менее устойчива пирамида.
2. При малой опоре и низком центре пирамида наименее подвижна.
    2.1 При малой опоре и высоком центре пирамида наиболее подвижна.
3. Наибольшая устойчивость в пирамиде достигается при положении, в котором централь, проходящая через центр тяжести, совпадает с вертикалью человека.
4. Малая устойчивость предполагает высокую подвижность, и, наоборот, при высокой устойчивости хорошая подвижность достигается с трудом.
5. Правило 3-й точки опоры – если проекция центра тяжести смещается в сторону одной из несуществующих опор, человек теряет равновесие, а пирамида – устойчивость, принуждаясь к подвижности или разрушению структуры (падению).

Именно на этих правилах построено наибольшее количество техник опрокидывания и захватов.

Среди множества физиологических функций человеческого организма, двигательная функция единственная, обеспечивающая активное воздействие человека на внешнюю среду, преодоление ее сопротивления, адаптацию к ее условиям. С точки зрения механики, человек представляет собой систему подвижно связанных звеньев с определенными размерами, массой, моментами инерции и снабженную мышечными двигателями. Анатомическими структурами, образующими эти звенья и соединения, являются кости, сухожилия, мышцы, суставы костей, а также внутренние органы и так далее. Эта концепция легко ложится на логику процедурной анимации.

Человеческие суставы можно представить как шарниры двух типов: цилиндрические и шаровые. Цилиндрический шарнир — это соединение двух звеньев, которое позволяет им вращаться вокруг общей оси. Примерами такого шарнира в человеческом теле может служить локтевой или коленный суставы. В цилиндрическом шарнире с одним закреплённым звеном свободное звено может двигаться только одним образом: поворачиваться вокруг оси шарнира, оставаясь при этом в одной плоскости. Его незакреплённый конец двигается при этом только по одной линии — дуге окружности с центром на оси шарнира. У шарового шарового шарнира звенья вращаются вокруг общей точки. Можно считать одно звено неподвижной опорой, тогда второе звено будет вращаться вокруг некоторой точки этого шарнира. Примерами таких шарниров в человеческом скелете являются плечевой и тазобедренный суставы. Свободное звено может качаться во все стороны. К тому же оно может поворачиваться вокруг собственной продольной оси, оставаясь при этом на месте. Незакреплённый конец свободного звена двигается при этом уже не по линии, а по участку сферы с центром в шарнире. Также отдельным типом шарнира в человеческом организме является кисть, она имеет только две степени свободы.

Отдельной, но немаловажной частью физики тела и логики его движения являются связки. Логика нанесения удара заключается в следующем:

1. Боец проверяет, что до цели достаточное расстояние для нанесения удара, в ином случае делается шаг или несколько.
2. Производится замах, а после наносится удар.
    2.1 Минимальное расстояние – достаточно вытянутой руки, корпус при этом прямой.
    2.2 Среднее расстояние – необходимо довернуть грудную клетку (поворот всех позвонков).
    2.3 Расстояние чуть выше среднего – тянется связка ноги, ступня встает на носок и образуется выпад.
    2.4 Максимальное расстояние – делается выпад не опорной ногой вперед.

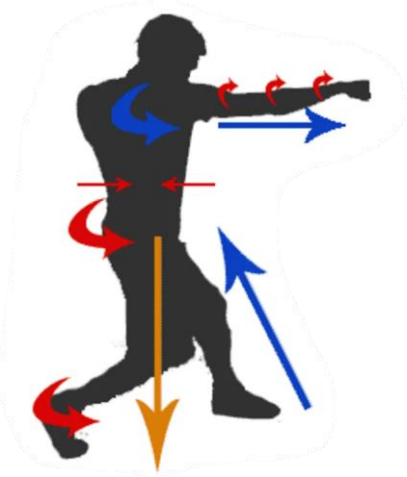

Рисунок 5 – Пример позы бойца при ударе рукой на средние расстояния

На этапе проектирования стоит разделить удары на данные подгруппы, чтобы гибко влиять на генерацию анимаций в зависимости от расстояния до врага. То есть на маленьких расстояниях задействовать только суставы и кости руки, на средних задействовать поворот груди и позвонков, далее начинать задействовать та и ноги.

1.4 Обзор боевого стиля бокатор

Стоит сказать несколько слов об истории искусства, ведь это напрямую отразилось на самих боевых техниках и приемах. Древнее азиатское боевое искусство, зародившееся в империи Кхмеров около 1700 лет назад [3]. Сам термин “бокатор” в переводе означает “бьющий льва”. Технически бокатор очень схож с тайским боксом. Некоторые ученые утверждают, что муай тай был основан на бокатор. Как и многие другие боевые искусства, бокатор основан на подражании различным животным. Однако, что его действительно отличает от остальных, так это жестокость и огромное разнообразие ударов.

По мнению историков, территория Кхмерской империи когда-то охватывала горные хребты, реки, озера и моря. Люди жили там вместе со всеми видами животных, такими как тигры, львы и многие другие хищники. Чтобы выжить, кхмеры стремились приобрести знания, которые позволили бы им приспособиться к природе и защитить себя от всевозможных диких животных.

Особенностями бокатора также являются удары. Бойцы используют локти, колени в согнутом состоянии. Помимо этого, также используются твердые части тела: кулаки, ступни. При этом, на практике в основном используются удары ногами в бедра или выше. Как противодействие таким ударам можно либо ответить аналогичным ударом или использовать один из типов захватов. Захватов существует огромное множество и все они нацелены на то, чтобы оказаться в мертвой зоне противника, закрепиться там, а после нанести один сокрушительный удар по голове или шее с целью убить. После успешного захвата бой продолжается на земле, но длится чаще всего не долго, как в реальном бою, так и в профессиональном спорте.

Бокатор подразделяется на стили различных животных (коней, орлов и журавлей) в зависимости от уровня владения искусством. Например, бойцы с белой Крамой (аналогично поясам в карате) могут использовать только

удары локтями и коленями. Со следующей Крамой открывается возможность использовать стиль коня и так далее.

Также стиль подразделяют на

1. Atmani Yuth (борьба без оружия)
    1.1. Kun Khmer (គុនខ្មែរ) – боксирование
    1.2. Baok Chambab (បោកចំបាប់) – борьба
2. Ani Yuth (с использованием оружия)
    2.1. Kbach Kun Dambong veign (ក្បាច់គុណដំបងវែង) – бой с длинной палкой
    2.2. L’Bokkatao (ល្បក្តោ) – бой с использованием Kun Khel

Особой сложностью реализации данного вида спорта на практике является огромная подвижность бойцов во время схватки и их способность буквально обвиваться вокруг своей жертвы, когда она откроется. Таким образом необходимо учитывать множество различных параметров, таких как рост соперников, положение ног, поворот корпуса, и расстояние между ними.

Рисунок 6 – Примеры захватов боевого стиля бокатор

Особенностями бокатора [2] также являются удары. Бойцы используют локти, колени в согнутом состоянии. Помимо этого, также используются твердые части тела: кулаки, ступни. При этом, на практике в основном используются удары ногами в бедра или выше. Как противодействие таким ударам можно либо ответить аналогичным ударом или использовать один из типов захватов. Захватов существует огромное множество и все они нацелены на то, чтобы оказаться в мертвой зоне противника, закрепиться там, а после нанести один сокрушительный удар по голове или шее с целью убить. После успешного захвата бой продолжается на земле, но длится чаще всего не долго, как в реальном бою, так и в профессиональном спорте.

Стоит сказать, что бокатор в том числе был выбран для восстановления и тестирования алгоритма именно из-за обилия захватов и удержаний (рисунок 7). Дело в том, что именно на таких типах движений текущие алгоритмы дают сбой и не могут точно определить положение конечностей, костей и всего тела в целом при перекрытии большей части объекта слежения. Таким образом, именно на таких видео можно будет в полной мере сравнить работу существующих алгоритмов и предлагаемого мной.

Рисунок 7 – Примеры захватов боевого стиля бокатор

Особой сложностью реализации данного вида спорта на практике является огромная подвижность бойцов во время схватки и их способность буквально обвиваться вокруг своей жертвы, когда она откроется. Таким образом необходимо учитывать множество различных параметров, таких как рост соперников, положение ног, поворот корпуса, и расстояние между ними, что невозможно если мы говорим о текущих методах считывания движений.

К сожалению, в данный момент не существует полноценных научных исследований или книг, посвященных данному виду искусств. Именно поэтому в том числе данный стиль является интересным примером для восстановления, ведь сейчас движения данного стиля можно восстановить простым образом только из оставшихся видео на эту тему.

Таким образом мы можем на практике проверить ситуацию, когда у нас нет возможности найти видео в условиях определенного освещения или фона и при наличии лишь одного человека в кадре, или снять бойца в специальном костюме. Ведь большинство текущих алгоритмов и реализаций предполагают именно наличие одного человека, снятого под определенным ракурсом с ровным светом и в идеальных условиях на зеленом фоне.

В данном случае необходим алгоритм, который, во-первых, сможет воспроизводить движения сразу нескольких человек в кадре. Во-вторых, к сожалению, мы не сможем создать определенные условия для освещения и нам надо будет подстраиваться под условия темных пятен и теней.

Единственным возможным вариантом станет прогнозирование положения каждого отдельного участника поединка за счет данных о других спортсменах, бывших в похожих положениях. Другими словами, нужна выборка размеченных данных спортсменов, чтобы иметь возможность на каждом кадре входного видео анализировать реальность прогнозируемой позы бойца.

2 МЕТОДЫ И ПОДХОДЫ К РЕАЛИЗАЦИИ АЛГОРИТМА

2.1 End-to-end Recovery of Human Shape and Pose

Данный подход [3] представляет комплексную систему для восстановления полной трехмерной сетки человеческого тела из одного RGB изображения. Создатели используют генеративную модель человеческого тела SMPL [4], которая параметризует сетку по углам 3D суставов, а также низкоразмерному пространству линейных форм. Как показано на рисунке 3, оценка 3D сетки открыла двери для широкого спектра задач восстановления позы человека в определенный момент времени.

На выходе данного алгоритма мы получаем положение и наклоны всех костей выбранного скелета. Эти данные могут быть использованы аниматорами, для корректировки, каких-либо других манипуляций или ретаргетинга на другие скелеты. Результат всегда является целостным – мы всегда получаем полное 3D тело даже в случаях окклюзии, усечения или заслонения объекта чем-либо.

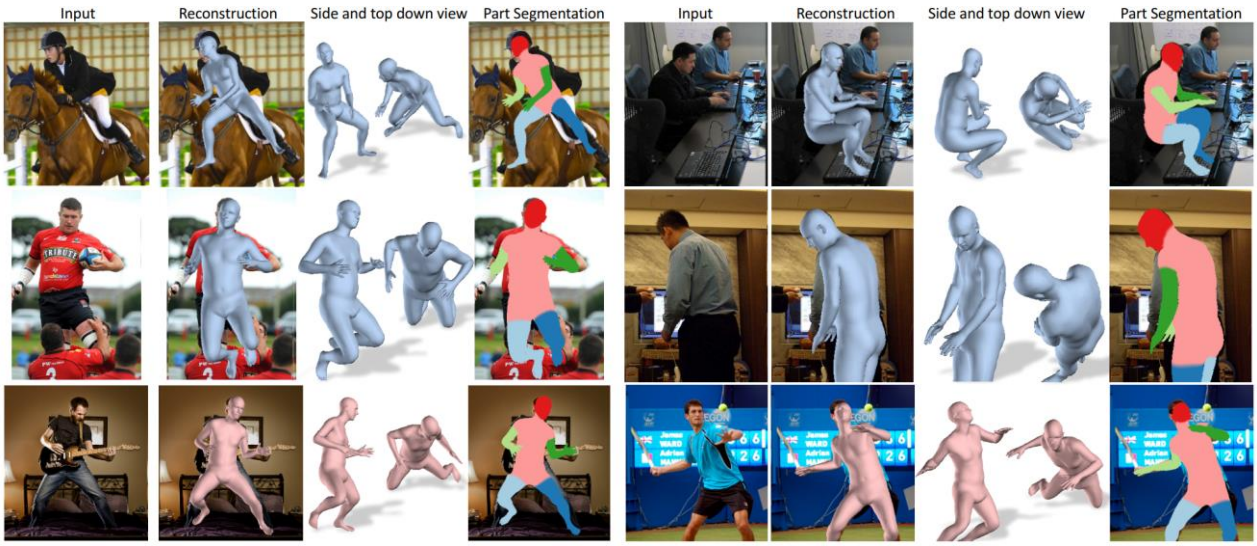

Рисунок 8 – Примеры восстановления положения человека по одному кадру системой HMR

Существует большое количество работ по 3D-анализу человека по одному изображению. Большинство подходов, однако, сосредоточены на восстановлении трехмерного расположения суставов. В данном подходе утверждается, что суставы сами по себе не дают полной картины. Суставы являются разреженными, в то время как человеческое тело определяется поверхностью в трехмерном пространстве.

Кроме того, расположение суставов само по себе не ограничивает полный DoF в каждом суставе. Это означает, что нетривиальная задача состоит в том, чтобы оценить полную позу тела на основе только трехмерного расположения суставов. В от описанных подходов, здесь выводятся относительные матрицы трехмерного вращения для каждого сустава в кинематическом дереве, получая информацию о 3D ориентации головы и конечностей. Предсказание вращений также гарантирует, что конечности симметричны и имеют правильную длины.

Существующие методы восстановления 3D сетки человека сегодня основаны на многоэтапном подходе [5][6]. Сначала они оценивают местоположения 2D суставов и на их основе оценивают параметры расположения 3D модели. Такой поэтапный подход обычно не является оптимальным, и в данном исследовании предлагается сквозное решение для обучения отображения от пикселей изображения непосредственно к параметрам модели.

Существующие наборы данных с точными 3D-аннотациями являются ограниченными, сделаны в ограниченных условиях. Модели, обученные на этих модели, обученные на этих наборах данных, плохо обобщаются на богатство изображений в реальном мире. Другая проблема заключается в неоднозначности, присущей многозначному отображению 2D в 3D. Когда несколько трехмерных конфигураций тела олицетворяют одни и те же 2D-проекции [7]. Кроме того, оценка камеры в явном виде вносит дополнительную неоднозначность масштаба между размером человека и расстоянием до камеры.

Данный алгоритм предлагает новый подход к реконструкции сетки, который решает обе эти проблемы. Ключевой идеей является то, что существуют крупномасштабные двухмерные аннотации ключевых точек изображений в естественных условиях и отдельная крупномасштабная база данных 3D-сетки людей с различными позами и формами.

Алгоритм использует преимущества этих не сопоставленных двухмерных аннотаций ключевых точек и 3D-сканирования в условно-генеративной состязательной манере. Идея заключается в том, что, получив изображение, сеть должна определить параметры 3D сетки и камеры таким образом, чтобы 3D ключевые точки совпадали с аннотированными 2D ключевым точкам после проецирования. Чтобы справиться с неоднозначностью, эти параметры передаются в сеть дискриминатора (рисунок 9), задача которой состоит в том, чтобы определить, соответствуют ли 3D параметры телам реальных людей или нет.

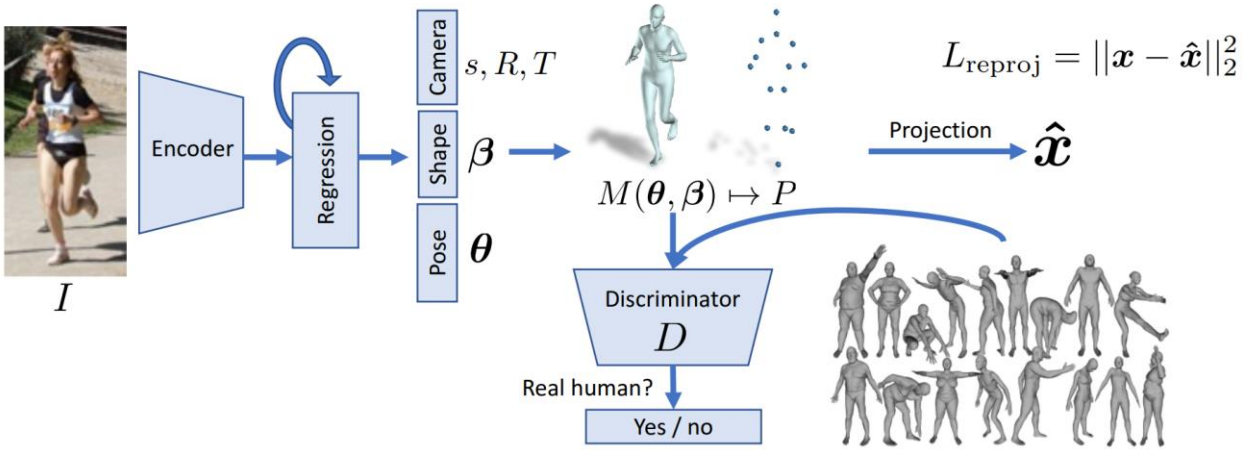

Рисунок 9 – Алгоритм работы системы HMR

Таким образом, сеть поощряется выводить параметры на человеческое многообразие, а дискриминатор действует в качестве надзирателя.

2.2 Learning 3D Human Dynamics from Video

В данной работе представляется вычислительный алгоритм, который может аналитическим образом получать модель трехмерной динамики человека из кадра видео. Получив временную последовательность изображений, сначала извлекаются особенности каждого изображения, а затем обучается простой одномерный временной анализатор, который изучает представление трехмерной динамики человека по временному контексту особенностей изображения. Этот механизм улавливает 3D динамику человека, предсказывая по текущей 3D позе и форме человека изменения позы в ближайших прошлых и будущих кадрах.

Полученные знания о трехмерной динамике переносятся на статические изображения путем обучения специальной нейронной сети, которая может предсказывать контекст из одной характеристики изображения. Данная сеть обучается самоконтролируемым образом. используя фактический выход временного кодера (рисунок 10).

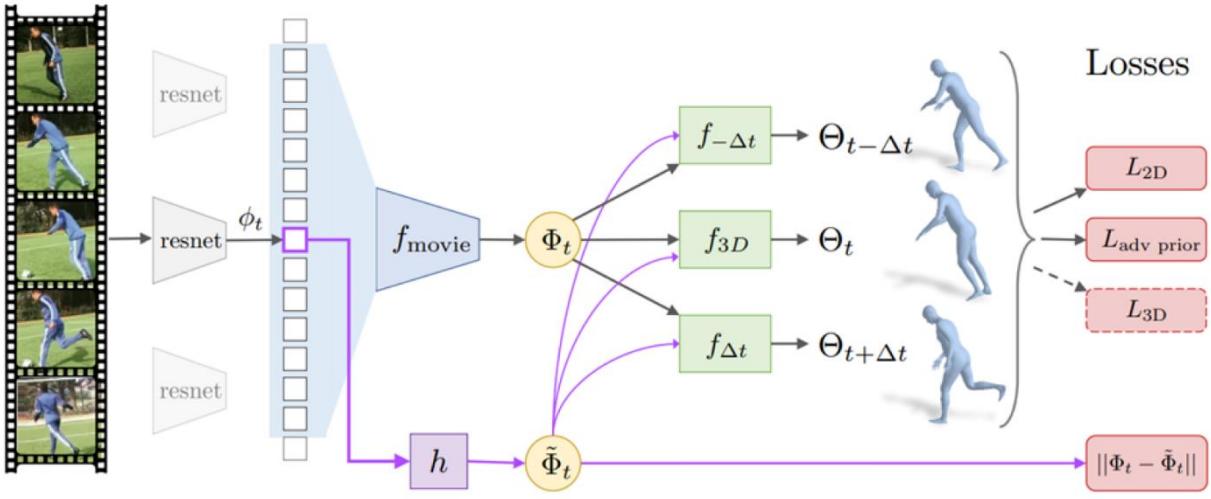

Рисунок 10 – Алгоритм работы системы HMMR

Учитывая временную последовательность изображений, сначала извлекаются особенности каждого изображения φt. Затем обучается временной кодер fmovie, который изучает представление трехмерной

динамики человека Φt во временном окне с центром в кадре t, показанном синей областью. На основе Φt предсказывается трехмерная поза и форма человека Θt, а также изменение позы в ближайших кадрах ±Δt.

Основной потерей является ошибка двумерной репроекции, а для того, чтобы убедиться, что восстановленные позы достоверны, используется состязательное предшествование. 3D-потери включаются в расчеты, когда доступны 3D-аннотации.

После этого обучается указанная выше нейронная сеть h, которая берет одно изображение φt и учится анализировать состояние необходимого представления Φ˜t.

2.3 Decoupling Human and Camera Motion from Videos in the Wild

В данном исследовании представлен подход, который моделирует движение камеры для восстановления трехмерного движения человека в мире по видеоданным. Система может работать с несколькими людьми и восстанавливает их движение в одном и том же кадре мировой координаты, что позволяет улавливать их пространственные отношения. Восстановление базовых движений людей и их пространственных отношений является ключевым шагом на пути к пониманию положению и позе людей по видео.

Существующие методы, которые восстанавливают глобальные траектории, требуют либо дополнительных датчиков, например, нескольких камер или датчиков глубины [8][9], либо плотная 3D реконструкция окружающей среды [10][11], и то и другое и то, и другое реально только при активном или контролируемом захвате в определенных условиях. Данный метод получает глобальные траектории из видео в естественных условиях, без ограничений на установку захвата, движения камеры или предварительных знаний об окружающей среде. Способность делать это с динамических камер стала особенно актуальна с появлением

крупномасштабных эгоцентрических видеоматериалов наборов данных [12][13][14].

Для реконструкции движения, получив входное видео в формате RGB, сначала оценивается относительное движение камеры между кадрами из статического движения пикселей сцены с помощью системы SLAM [15]. В то же время, также оцениваются личности и положения тел всех обнаруженных людей с помощью системы 3D-слежения за людьми [16] (рисунок 11).

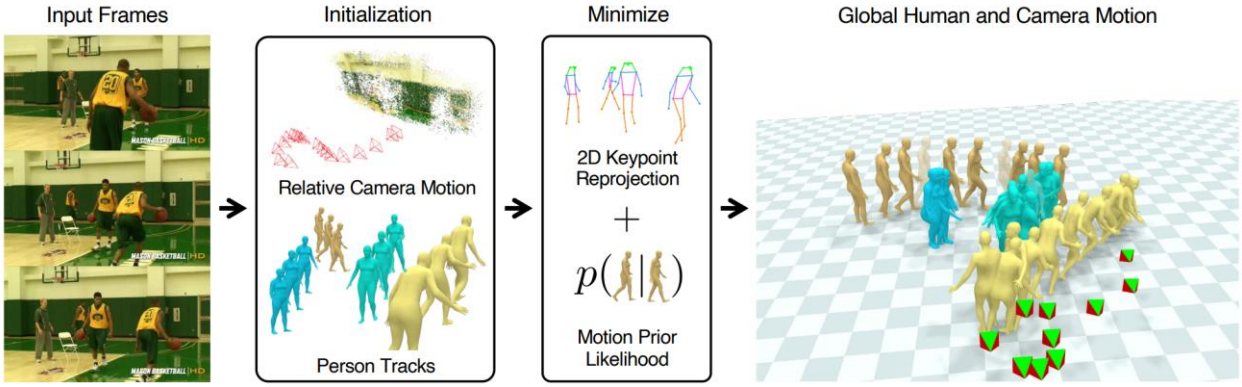

Рисунок 11 – Алгоритм работы системы SLAHMR

Все это необходимо для создания траекторий людей и камеры в мире данного изображения. Затем эти глобальные траектории оптимизируются на нескольких этапах, чтобы они соответствовали как с 2D наблюдениями в видео, так и с полученными предположениями о том, как человек перемещается в мире [17].

В отличие от существующих работ [8][18], здесь траектории движения человека и камеры в мире оптимизируются, не требуя точной 3D реконструкции статичной сцены. Благодаря этому данный метод работает на видео, снятых в естественных условиях, что является сложной задачей для предшествующих методов, требующих хорошей 3D геометрии, поскольку эти видеозаписи редко содержат точки обзора камеры с достаточными базовыми линиями для надежной реконструкции сцены.

Суть алгоритма заключается в объединении двух основных идей, позволяющих провести эту оптимизацию. Во-первых, даже когда параллакс сцены недостаточен для точной реконструкции сцены, он все равно позволяет получить разумные оценки движения камеры до произвольного масштабного фактора. Во-вторых, человеческие тела могут реалистично перемещаться в мире в небольшом диапазоне.

Все вышеуказанные знания используются для параметризации траектории камеры, чтобы она соответствовала как параллаксу сцены, так и двумерному воспроизведению реалистичных траекторий движения человека в мире. В частности, в данном исследовании оптимизируется масштаб смещения камеры, с помощью относительной оценки камеры, чтобы она также перемещалась в соответствии с движением человека. Более того, когда на видео присутствует несколько человек, как это часто бывает в обычных видео, движения всех людей дополнительно ограничивает масштаб камеры, что позволяет данному методу работать на сложных видео с множеством люд.

2.4 Модель алгоритма восстановления движений по видео

К сожалению, все указанные выше алгоритмы не выполняют в полной мере свою задачу, особенно когда речь идет о видео про боевые искусства или другие виды спорта. Это происходит из-за того, что в подобных видео зачастую встречаются перекрытия частей тела спортсмена другими игроками или же предметами. В указанных случаях мы получаем задачу реконструкции с неполными данными о движении и позах. Таким образом, мы должны попробовать воссоздать невидимые для камеры движения на основе каких-либо других данных, например, в идеале, того же вида спорта, но, к сожалению, это далеко не всегда возможно. Это также невозможно, когда решается тема данной научной работы по восстановлению движений исключительно из существующих роликов по боевому стилю Бокатор.

В данном случае мы просто не можем собрать достаточно достоверных данных, ведь мы не сможем найти видео с идеальным наклоном камеры и фоном. Все доступные видео как раз обладают недостатками, которые не умеют обходить существующие алгоритмы восстановления движений: темные неосвещенные помещения, быстрые движения, перекрытия.

Таким образом, для решения задачи исследования необходимо создать алгоритм, восстанавливающий движения человека по видео, а также учитывающий пропущенные кадры и улучшающий движения ситуациях перекрытия. Предлагаемая модель расположена на рисунке 12.

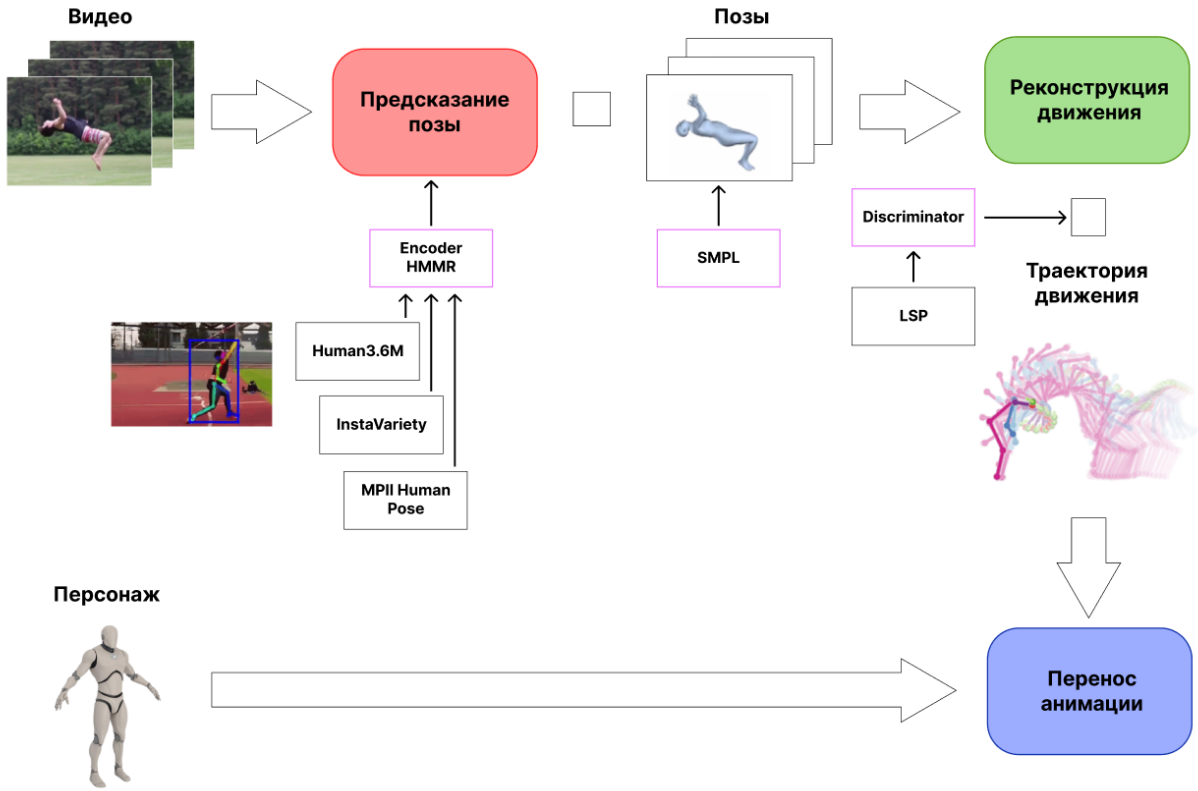

Рисунок 12 – Модель алгоритма генерации анимаций на основе видео

На вход мы получаем видео определенного размера, далее алгоритм начинает работать с каждым кадром отдельно. После разделения мы начинаем итерироваться по массиву кадров и для каждого из них пытаемся предсказать позу человека на них. При этом во время предсказания мы

можем использовать один из возможных существующих алгоритмов определения, например (HMR, HMMR, GLAMR, PHALP и др.). При этом каждый из алгоритмов обладает своими особенностями, скоростью работы, точностью и т.п. Главным условием является то, что на выходе данного алгоритма должен получаться массив сочленений и их углов наклона для каждого кадра.

В данной работе я буду использовать HMMR и SLAHMR и сравнивать качество их работы на разных данных и видео. В первую очередь это объясняется актуальностью данных алгоритмов. Каждый из алгоритмов предполагает обучение встроенной нейросети на определенной размеченной выборке данных. Под размеченной выборкой подразумевается набор видеоданных, каждый кадр которого специально обработан таким образом, что всем людям, попадающим в кадр дополнительно дорисованы упрощенные скелеты. По таким данным нейросеть может научиться определять по растровому изображению людей и их позы, а также делать предположения о будущей и прошлой позах человека в кадровом пространстве.

В решаемой проблеме сильно помогло бы достаточное количество размеченных данных о боевых искусствах. Таким образом, получилось бы обучить нейросеть более корректному расположению всех конечностей и туловища относительно другого спортсмена. Но, к сожалению, в данный момент из открытых источников мы имеем только большие наборы размеченных видео, снятых в обычных условиях. Именно поэтому в некоторых ситуациях нейросеть некорректно обрабатывает сложные движения, несвойственные людям в обычной жизни.

Именно таким образом на этапе Предсказания мы получаем позу из конкретного кадра. После того как мы уже получили массив углов поворотов сочленений мы непереводим абстрактный скелет в физическое воплощение человеческого тела. Это можно сделать с помощью библиотеки SMPL, которая на основе, массива сочленений может построить физическое тело.

При этом данное воплощение будет полностью корректно с точки зрения биофизики.

После этого наступает этап, когда необходимо реконструировать все движение целиком, для этого понадобится собрать все позы из кадров воедино. Обычно в подобных реализациях позы между кадрами линейно интерполируются, за счет чего достигается достаточный уровень плавности. Также применяются различные функции потерь, но это несильно влияет на конечный результат, ведь все данные уже собраны и получены, а небольшие отклонения не сильно повлияют на общую картину.

В этом месте возникает проблема, мы уже получили все необходимые данные, просчитали положения тел и при этом сделали это достаточно точно, для каждого кадра. В данной ситуации практически невозможно как-либо коренным образом изменить ситуацию и ход всего движения в целом. Также здесь же возникает и проблема неполноты данных. Если на видео отсутствует фрагмент с определенной частью тела, или она перекрывается чем-то, мы теряем большое количество данных, которые в процессе интерполяции превращаются в быстрое перемещение или же вообще искажает позу в целом.

Именно поэтому я считаю, что на этом этапе необходимо пойти немного иным путем, относительно остальных реализаций. На этом этапе мы добавим некого надзирателя, который будет заранее обучен на размеченной выборке LSP, содержащей некоторое количество данных о спортивных движениях в различных сферах. Надзиратель будет опираться на изученные данные и корректировать положение тела или конечностей некоторого количества соседних кадров согласно своему мнению.

Данный подход позволит, во-первых, восстанавливать потерянные в процессе предсказания элементы движений или положений, а также скорректирует все движения, которые не могли произойти с человеческим телом в реальности, но получились на основе неточности предсказанных данных.

Непосредственно после реализации планируется применить полученный алгоритм к реальным видео данным по боевому стилю Бокатор и сравнить полученные результаты с альтернативными алгоритмами восстановления движений. После этого необходимо провести точечное сравнение по скорости работы, производительности, а также точности результатов и количеству ошибок.

Эти данные можно впоследствии использовать для улучшения работоспособности алгоритма. Также возможно в процессе сбора статистики обнаружатся тонкие места, которые излишне замедляют или ухудшают работу. Их можно будет подсветить и заменить составляющие на другие, возможно из более новых научных исследований на данную тему. Данный этап будет более подробно описан в разделе возможных улучшений алгоритма в конце научного исследования.

Что касается переноса полученных на этапе обработки данных о движениях, есть несколько способов перенести их в анимации, которые уже позже можно будет загрузить в редактор и исправлять или дорабатывать. Во-первых, можно написать собственный интерпретатор файлов pkl, которые получаются на выходе данной модели алгоритма, основанной в том числе на алгоритме SMPL. Данный способ реализуем не сложно, но есть шанс получить некачественный инструмент, который впоследствии повлияет на подсчеты и подведение итогов. Во-вторых, можно взять один из существующих инструментов, представляющий возможность для переноса файла из формата выходного файла pkl в fbx, например [22]. Этот алгоритм представляет из себя интерпретатор массива из N поз стандартной модели SMPL в массив точек fbx анимации, которую уже впоследствии можно загрузить в Unreal Engine.

3 РЕАЛИЗАЦИЯ АЛГОРИТМА

3.1 Описание алгоритма

Для реализации на текущем этапе разработки в качестве метода для улучшения был выбран алгоритм HMMR. По нескольким причинам.

Во-первых, это самый новый алгоритм, который без доработок показывает лучшие результаты в качестве выходных анимаций. Несмотря на то, что все алгоритмы из списка (HMR, HMMR, GLAMR, PHALP) имеют различную структуру и не совсем корректно сравнивать их напрямую, качество все же является главным фактором в воде данной работы. Остальные алгоритмы планировалось интегрировать на последующих этапах разработки, чтобы проверить какого качества можно добиться на их основе.

Во-вторых, алгоритм HMMR использует новейшую систему 4D Humans, которая гораздо лучше справляется с распознанием положения человеческого чела в пространстве по каждому отдельному кадру.

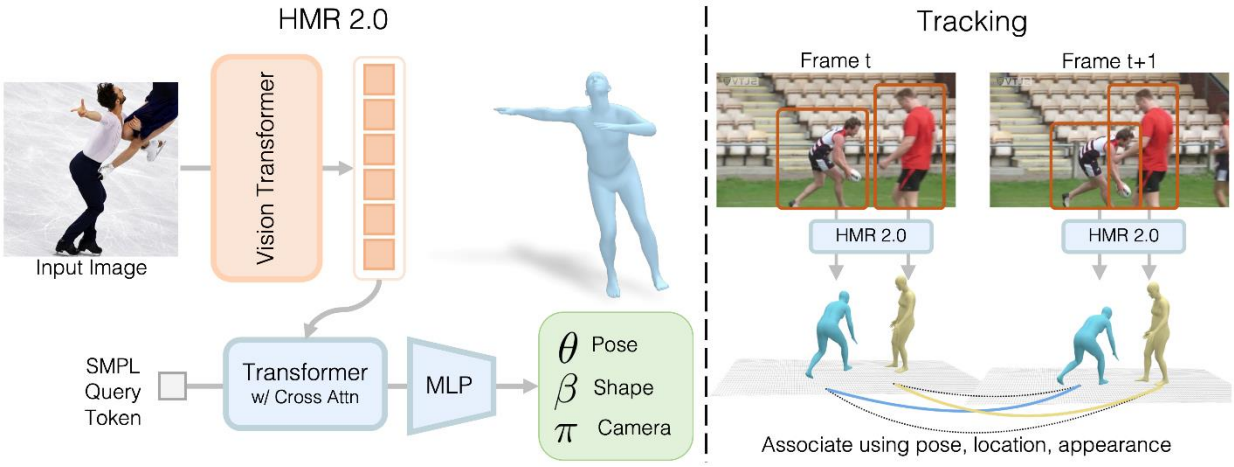

Рисунок 13 – Модель алгоритма HMR 2.0

Также отдельно стоит отметить, что алгоритм HMMR показал лучшие результаты среди соперников по метрикам W-MPJPE, W-MPJPE, а также PA-

MPJPE, которые отвечают за точность получаемой на выходе позы относительно начальной позы на кадре.

Method	W-MPJPE↓	WA-MPJPE↓	Acc Err↓	PA-MPJPE↓
PHALP$^+$ [46]	387.8	204.9	307.6	**72.16**
VIBE [22]	500.4	259.5	524.2	100.5
VIBE-opt [22]	453.2	246.0	481.1	100.4
GLAMR [65]	416.1	239.0	173.5	114.3
SLAHMR	**141.1**	**101.2**	**25.78**	79.13

Рисунок 14 – Сравнение алгоритмов переноса движений с видео. HMMR в данном случае озаглавлен как SLAHMR

Как можно увидеть по данным о статистике получившихся метрик, HMMR выигрывает в несколько раз, относительно остальных алгоритмов по параметрам евклидова расстояния от изначальной позы до получившейся на выходе физической модели.

Главным отличием данной системы является то, что разработчики используют 3D позы для дополнительного сглаживания движения человеческого тела в пространстве на каждом отдельном кадре. Таким образом в итоге получается картинка гораздо более похожая на движения реального человека. Стоит также отдельно отметить, что сглаживание осуществляется за счет линейной математичкой функции и не предоставляет возможности для улучшения данной составляющей. В свою очередь в данной работе я буду заменять данную функцию на специальный галлюцинатор, который будет отдельно обучен на специфических данных о позах.

Эти данные буду получены из датасета LSP, который основан на спортивных записях, где профессионалы выполняют те или иные специфические упражнения или трюки. Ограничив движения физической модели по данному дискриминатору, можно будет добиться правдоподобности движений человеческого тела на выходе алгоритма.

Также алгоритм HMMR написан под использование сразу с видео удобного формата, тогда как для остальных алгоритмов его придется предобрабатывать, чтобы подать на вход.

Когда выбор по поводу алгоритма сделан, далее необходимо выбрать датасеты на которых будет необходимо обучить HMMR, а также галлюцинатор, который будет реализован в рамках данного отчета. На предыдущих этапах было решено выбирать из следующих датасетов:

- Posetrack;
- Egobody;
- 3DPW;
- Custom video;
- Human3.6M.

Из предложенных датасетов более всего под решение конкретной спортивной задачи лучше всего подойдет датасет 3DPW, который представляет из себя размеченные спортивные данные с записей спортсменов. Остальные датасеты в меньшей мере затрагивают спортивные данные. Также выбор данного датасета подтверждает исследование [21], в котором проводилось сравнение самых популярных на данный момент датасетов на новейших алгоритмах.

Плюсом к этому, как и было сказано ранее будет использоваться датасет LSP и LSP-Extended. Данные датасеты являются основой для данной работы и сыграют главную роль в проверках качества, ведь именно такого решения, как стало понятно в процессе предыдущих исследований, не было.

3.2 Архитектура алгоритма

Далее будет представлена архитектура (рисунок 15) разрабатываемого алгоритма для общего понимания структуры работы системы переноса анимаций.

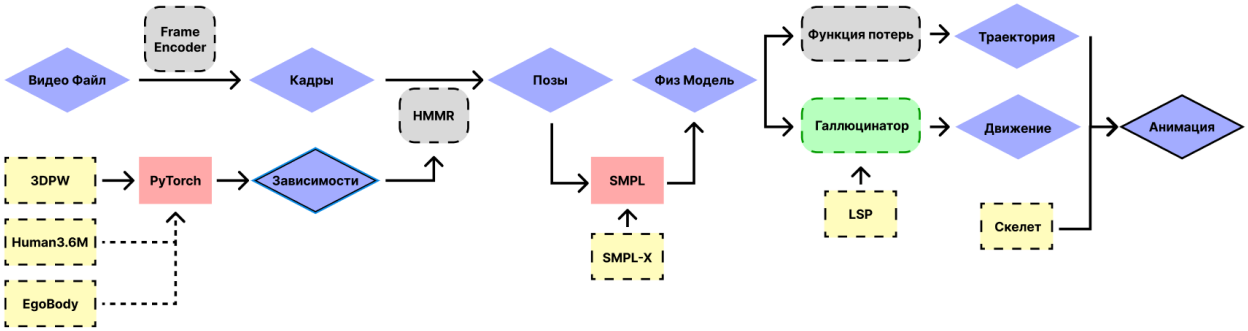

Рисунок 15 – Архитектура разрабатываемого алгоритма

Необходимо чуть подробнее рассказать про специфику работы системы. В самом начала алгоритм получает на вход видео, затем происходит разбивка полученного видео на кадры. По окончании данного этапа мы уже получаем массив картинок. Далее с помощью вышеописанного алгоритма HMMR, в данном случае обученного с помощью PyTorch на датасете 3DPW, мы обрабатываем каждый кадр и алгоритм выдает предсказание в виде скелета человека на каждый конкретный кадр, а также наследующие и предыдущие.

Следующим этапом становится перенос получившейся позы на реальную физическую модель тела человека, чтобы спроектировать реальное положение суставов. Для этого в данной работе используется алгоритм SMPL в группе со специальным датасетом SMPL-X, который показывает лучшие результаты в контексте физики поведения человеческого тела. На выходе этого этапа мы получаем преобразованный скелет, очищенный от основного анимационного шума, который уже можно использовать в качестве дальнейшего материала.

Основным отличием алгоритма из данной работы как раз является изменение на данном этапе. Оригинальный алгоритм использует функцию потерь, которая итерируется по кадрам получившейся физического тела и тем самым как бы создает связи между данными кадрами. Это помогает

сгладить движение, но не избавляет от странного дерганья если дело доходит до сложных, с точки зрения интерпретации, видео.

В данном исследовании будет использована технология галлюцинатора, итерирующаяся нейронная сеть, опирающаяся на будущее и прошлое. В случае с видео алгоритм будет опираться на предыдущие и последующие кадры. Таким образом, обученная на датасете LSP нейронная сеть, сможет отлавливать нереалистичное поведение тела, основываясь на крайних данных из датасета. С каждым кадром движение будет все более гладким и правильным, а в самом конце итерации мы получим скорректированный набор поз.

Архитектуру галлюцинатора можно увидеть на рисунке 16. Он представляет из себя нейронную сеть, обученную на специфическом датасете LSP, о котором говорилось ранее. Особенностью данной нейронной сети является то, что она заточена под предугадывание и корректировку поз в зависимости от результата обучения. В данном случае мы используем спортивные данные, поэтому галлюцинатор сможет на основе этих данных попробовать предугадать в каком положении было тело на предыдущем кадре и на последующем. Для большей точности также будет использоваться больший диапазон чем два кадра.

Таким образом из физической модели мы получаем набор кадров и прогоняем их через обученный галлюцинатор. Для каждого кадра он прогнозирует наиболее возможные позы до и после. На основе этих параметров он корректирует положение суставов для будущих кадров, тем самым формируя более правильную картину происходящего на кадре.

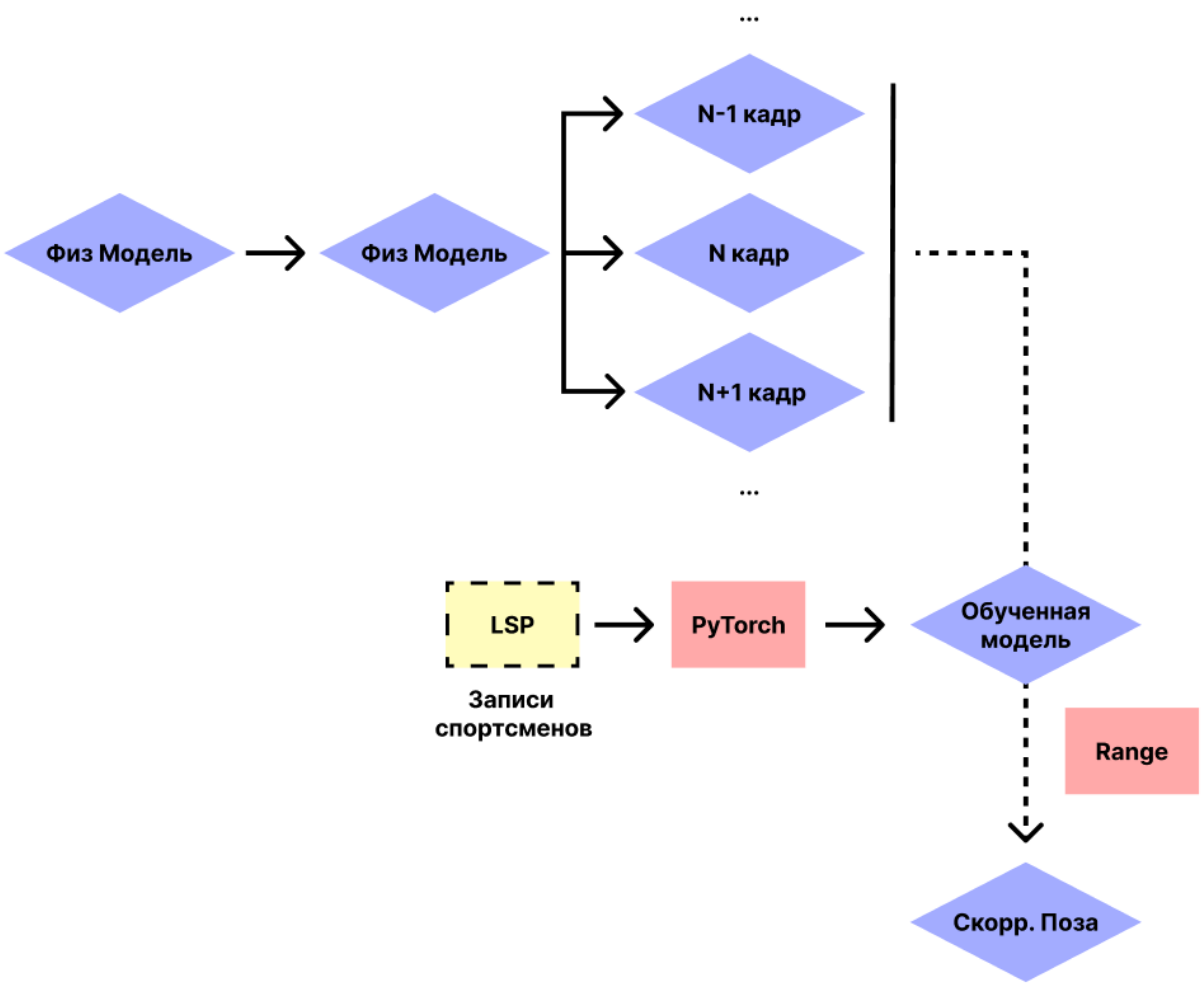

Рисунок 16 – Архитектура галлюцинатора

Далее, переходя снова в главный алгоритм, он также использует функцию потерь из оригинальной работы, но уже для того, чтобы просто получить траекторию движения, с этой задачей она справится лучше, чем галлюцинатор. На выходе мы получим скорректированные движения и траекторию, наложив их друг на друга можно будет составить точную картину происходящего.

Именно таким образом получается выходная анимация. Остается только наложить данную анимацию на какой-либо скелет. В данный момент такая функция не реализована, но возможно будет сделана в последующем. Сейчас же можно взять выходной файл алгоритма и самостоятельно

перенести анимацию с помощью сторонних программ, например Unreal Engine.

В данном исследовании не было необходимости реализовывать и подключать интерпретатор выходного файла алгоритма, так как внутри реализации библиотеки SMPL уже есть реализации визуализации через генерацию mp4 файла. Статистические же данные стало возможным получить с помощью библиотеки для Python “dhmlpe_utils”.

Были предприняты попытки по переносу данных из pkl файла, который получается на выходе алгоритма SMPL и HMMR. Но сложная специфичная структура и сериализованные данные не дали этого сделать просто, а предложенные готовые решения оказались нерабочими, в применении к конкретной задаче.

Таким образом, было принято решение оставить текущую научную работу на этапе реализации и доработки алгоритма не захватывая часть переноса анимации из выходного файла в какой-либо читабельный формат.

Тем не менее стоит отдельно рассказать про структуру выходного файла для проведения дальнейших исследований.

Файл с расширением pkl является сериализованным с помощью библиотеки pickle словарем содержащим следующие данные о модели SMPL:

1. TIME – номер кадра (int);
2. CAMERA – позиция камеры (List[[tx ty tz]]);
3. POSE – полная поза скелета соответствующая модели SMPL для каждого человека (List[np.array(229,)]);
4. SCALE – масштаб для каждой модели тела (List[float]);
5. SMPL – внутренние параметры библиотеки (List[Dict_SMPL]);
6. 2D_JOINTS – 45 двумерных соединений для каждого тела (List[np.array(90,)]);
7. 3D_JOINTS – 45 трехмерных соединений для каждого тела (List[np.array(45,3)]).

4 СРАВНЕНИЕ РЕЗУЛЬТАТОВ

4.1 Результаты на примерах

Ниже представлены результаты полученного алгоритма в сравнении с открытым ресурсом Deep Motion, который предоставляет подобный функционал. Deep Motion работает на улучшенной версии алгоритма HMMR, поэтому будет корректно сравнивать его результаты с получившемся в данной работе алгоритмом.

Представлены только несколько кадров из-за невозможности загрузить видео в текстовый документ. В представленных изображениях можно увидеть оригинал, результат DeepMotion, а также результат написанного алгоритма.

Первое видео (рисунок 17) достаточно хорошего качества и в большей степени экспозиция не мешает различать позиции бойцов за исключением времени, когда они накладываются друг на друга. В данном случае DeepMotion показала неплохой результат, но в сравнении с написанным алгоритмом выглядит все равно гораздо хуже.

Второе видео (рисунок 18) гораздо сложнее в интерпретации. В данном случае DeepMotion практически не справился с задачей, в то время как написанный алгоритм справился хорошо, если судить по визуальной составляющей.

Рисунок 17 – Результат на первом видео

Рисунок 18 – Результат на втором видео

4.2 Выбор метрик для сравнения

Для оценки полученных результатов на предыдущих этапах было решено использовать следующие метрики.

- **PCP** (**Percentage of Correct Parts**) конечность считается обнаруженной (корректной), если расстояние между двумя предсказанными расположениями суставов и истинным расположением суставов конечности составляет менее половины длины конечности;

- **PDJ** (**Percentage of Detected Joints**) обнаруженный сустав считается правильным, если расстояние между предсказанным и истинным суставом находится в пределах определенной доли диаметра туловища;

- **MPJPE** (**Mean Per Joint Position Error**) среднее расстояние между предсказанными суставами человеческого скелета и реальными суставами в данном наборе данных.

Также в качестве дополнительного показателя было решено взять скорость работы выборки алгоритмов на различных отрезках количества кадров загружаемых видео. Другими словами, необходимо построить график зависимости времени работы алгоритма от количества секунд видео или же от количества загружаемых кадров видео.

Данный параметр можно сравнить с показателями других алгоритмов, чтобы понять, насколько сильно отличаются временные показатели обработки видео.

Полученные данные могут в последствие использоваться для оптимизации решения с целью ускорения работы аниматоров, обрабатывающих видео с помощью алгоритма.

4.3 Оценка результатов

За основу данной статистики по другим алгоритмам были взяты соответствующие исследования, а в свою очередь из них уже были взяты статистические данные по соответствующим метрикам. После этого была сформирована следующая таблица.

Исследование проводилось именно на видео из датасета 3DPW, так данный датасет присутствует в каждой работе на данную тематику и по нему легче сравнивать получившиеся данные.

Для проверки алгоритм был полностью подготовлен, а затем в него было загружено несколько видео из датасета 3DPW, но без разметки. Таким образом алгоритм должен был самостоятельно определить анимации. Затем полученный файл анимации стало возможно покадрово сравнить с размеченным вариантом из датасета. На основе этих данных и была собрана статистика для метрик.

Таблица 1. Сравнение результатов

	PCP	**PDJ**	**MPJPE**
HMR	2.8	8.9	130.0
HMMR	3.3	10.9	87.6
GLAMR	3.2	**12.3**	**61.3**
Полученные результаты	**3.5**	10.7	70.0

Интерпретировать результаты можно следующим образом. Полученный эффект алгоритма гораздо лучше подобных решений. Это однозначно можно пронаблюдать по метрике PCP, которая описывает корректность постановки конечностей. Результат лучше всех своих коллег на достаточное количество. С остальными параметрами все не так однозначно. Можно заметить, что они проигрывают метрикам PDJ и MPJPE алгоритма

GLAMR. Это связано с тем, что данный алгоритм нацелен на наиболее корректное отображение каждого конкретного кадра, тем самым он сильно выигрывает при сборе вышеуказанных метрик, так как они как раз собираются покадрово. Тем не менее алгоритм GLAMR не дает плавной анимации видео как раз по вышеуказанным причинам, чаще всего анимация получается очень рваной и с большим количеством артефактов. Следовательно, главным фактором становится то, что результаты полученного алгоритма находятся на втором месте, ведь это значит, что он не только превосходит коллег по визуальной составляющей, но еще и лучше отображает физические свойства конечностей и суставов.

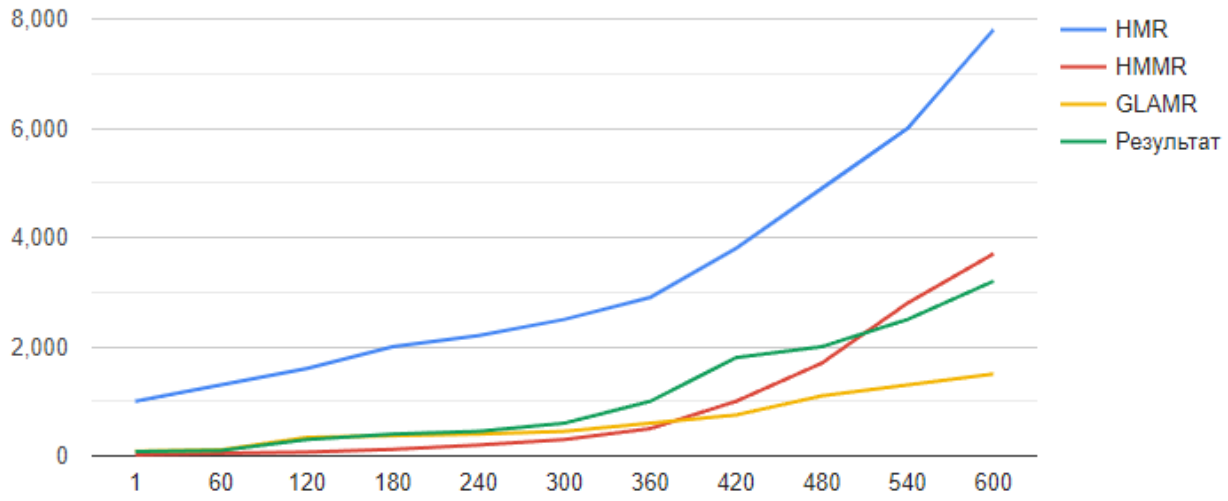

Рисунок 19 – График зависимости времени работы алгоритмов от количества кадров входного видео

Следующим шагом был проведен анализ времени работы алгоритмов. Данные замеров можно увидеть на рисунке 19. По ним можно однозначно сделать следующий вывод: полученный алгоритм работает на ровне с остальными подобными алгоритмами, но так как время генерации исчисляется в десятках минут, первостепенно стоит запланировать возможную оптимизацию решения.

ЗАКЛЮЧЕНИЕ

В результате выполненной работы была выделена проблемная область в сфере переноса анимаций с видео. На основе полученной проблемы был проведен обзор существующих решений, после чего наличие проблемы было подтверждено. Исходя из существующих решений были описаны основные подходы к решению задачи восстановления движений человека по неполноценным видео данным с перекрытиями.

На основе полученных данных была выдвинута гипотеза о возможном решении проблемы, затем по полученной гипотезе была успешно сформирована методология и модель разрабатываемого алгоритма, а также принцип его работы и узкие места реализации.

После выбора и обоснования основных инструментов реализации алгоритма было разработано решение, улучшающее текущие и представляющее функционал по переносу анимаций из видео. (https://github.com/Sashiyamo/VKR_3D_Reconstruction)

Далее был проведен сравнительный анализ получившегося результата в сравнении с главными конкурентными алгоритмами подобного типа. После чего были созданы сводная таблица результатов и график на основе параметров и метрик, выбранных на предыдущих этапах исследования. Все полученные результаты были интерпретированы, подтвердив гипотезу о том, что полученный алгоритм решает обнаруженную проблему, а также лучше выполняет свою основную качественную задачу.

Существует несколько направлений для улучшения полученного результата. Далее они будут рассмотрены по порядку.

Во-первых, самым главным узким горлышком в полученном алгоритме стала скорость работы. Однозначно можно сказать, что по времени работы алгоритм не сильно уступает похожим решениям, но все же речь идет о десятках минут для обработки видео около 10 секунд. Это является большой

проблемой и ее нужно решать в первую очередь. Основные вычисления происходят на стороне алгоритма SMPL, который определяет позу по кадру, именно из-за него расчет получается таким долгим. Первым делом необходимо глубже исследовать алгоритм SMPL и оптимизировать его работу, чтобы уменьшить общее время расчетов.

Во-вторых, хоть и были получены отличные результаты по качеству, все же можно также попробовать улучшить этот момент за счет обучения алгоритма HMMR на другом датасете или же рассмотреть вариант дообучения на нескольких датасетах сразу.

В-третьих, можно воспользоваться комбинированной структурой алгоритма и попробовать протестировать его на различных доступных параметрах. Например, можно менять количество кадров рассматриваемых галлюцинатором для получения еще более плавной картинки. Стоит также заметить, что это может повлиять на скорость работы в худшую сторону, но улучшит результат.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Starke, Sebastian, et al. "Neural animation layering for synthesizing martial arts movements." ACM Transactions on Graphics (TOG) 40.4 (2021): 1-16.

[2] Bokator (Cambodia) [Электронный ресурс]. – URL: http://www.traditionalsports.org/traditional-sports/asia/bokator-cambodia.html (Дата обращения 18.11.2022).

[3] Kanazawa, Angjoo, et al. "End-to-end recovery of human shape and pose." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[4] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia), 34(6):248:1– 248:16, Oct. 2015.

[5] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In European Conference on Computer Vision, ECCV, Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016.

[6] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR, July 2017.

[7] C. Taylor. Reconstruction of articulated objects from point correspondences in single uncalibrated image. Computer Vision and Image Understanding, CVIU, 80(10):349–363, 2000.

[8] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (HPS): 3D human pose estimation and self-localization in large scenes from body-mounted sensors. In CVPR, 2021.

[9] Nitin Saini, Chun-hao P. Huang, Michael J. Black, and Aamir Ahmad. Smartmocap: Joint estimation of human and camera motion using uncalibrated rgb cameras, 2022.

[10] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In International Conference on Computer Vision, pages 2282–2292, Oct. 2019.

[11] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4D human body capture from egocentric video via 3D scene grounding. In 3DV, 2021.

[12] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision (IJCV), 130:33–55, 2022.

[13] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In CVPR, 2022.

[14] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Marc Pollefeys, Federica Bogo, and Siyu Tang. EgoBody: Human body shape, motion and social interactions from head-mounted devices. In ECCV, 2021.

[15] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. NeurIPS, 2021.

[16] Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, and Jitendra Malik. Tracking people by predicting 3D appearance, location and pose. In CVPR, 2022.

[17] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. HuMoR: 3D human motion model for robust pose estimation. In ICCV, 2021.

[18] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4D human body capture from egocentric video via 3D scene grounding. In 3DV, 2021.

[19] Kanazawa, Angjoo, et al. "Learning 3d human dynamics from video." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[20] Ye, Vickie, et al. "Decoupling human and camera motion from videos in the wild." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[21] Goel, Shubham, et al. "Humans in 4d: Reconstructing and tracking humans with transformers." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

[22] SMPL to FBX [Электронный ресурс]. – URL: https://github.com/softcat477/SMPL-to-FBX (Дата обращения 23.04.2024).