Московский государственный университет
имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики
Кафедра исследования операций

Курсовая работа

«Метод реконструкции относительной трехмерной позы человека в полуконтролируемой среде»

студента группы 311

Фостенко Олега Андреевича

Научный руководитель:

к.ф.-м.н.

Г.А.Белянкин

Москва, 2025

Содержание

Введение

Определение ключевых точек тела человека является важным этапом для количественного анализа движений, особенно в медицинских исследованиях, связанных с нарушениями опорно-двигательного аппарата у пациентов.

В данной работе основное внимание уделяется применению методов компьютерного зрения для автоматического восстановления 2D- и 3D-координат ключевых суставных точек человека на основе видео. Для решения задачи используются state-of-the-art (SOTA) нейросетевые модели, обученные на больших датасетах с разметкой ключевых точек.

В отличие от подходов, основанных на использовании носимых устройств (сенсоров) [8; 10], предлагаемый метод не требует дополнительного оборудования. В работе рассматриваются два варианта съёмки: моно- и мультикамерный (с двумя камерами). При этом оба варианта реализованы без проведения калибровки камер и точного определения их взаимного расположения. Это делает подход потенциально применимым в ситуациях с ограниченным техническим оснащением.

Основной целью работы является создание метода восстановления ключевых точек тела человека по видеоданным, полученным в условиях ограниченного контроля, т.е. без предварительной калибровки камер и с минимальными требованиями к оборудованию.

1 Постановка задачи

Входные данные:

Видеоряд $V = \{I_1, I_2, \ldots, I_T\}$, где $I_t \in \mathbb{R}^{H \times W \times 3}$ – кадр в момент времени t.

Требуется реализовать пайплайн (т.е. метод) обработки видео, последовательно выполняющий следующие шаги:

1. Детекция людей в кадре

Для каждого кадра $I_t \in V$ найти ограничивающие рамки:

$$B_t = \big(b_t^1(I_t), \ldots, b_t^{N_t}(I_t)\big), \quad b_t^i(I_t) = \big(x_t^{i,1}, y_t^{i,1}, x_t^{i,2}, y_t^{i,2}, c_t^i\big) \tag{1}$$

где N_t – количество обнаруженных людей, $(x_t^{i,1}, y_t^{i,1})$ и $(x_t^{i,2}, y_t^{i,2})$ – координаты левого нижнего угла и правого верхнего угла рамки на изображении соответственно, а c_t^i – степень уверенности в обнаружении.

2. Оценка позы в 2D

Для каждого обнаружения b_t^i в кадре I_t получить ключевые точки:

$$S_t^i = \big(p_1(I_t, b_t^i), \ldots, p_K(I_t, b_t^i)\big), \quad p_j = (u_j, v_j, c_j) \in \mathbb{R}^3 \tag{2}$$

где K – число ключевых точек, описывающих 2D позу человека, u_j – горизонтальная координата, v_j – вертикальная координата, а c_j – уверенность в обнаружении соответствующей точки, $i = 1, \ldots, N_t$.

3. Трекинг между кадрами

Построить отображение между обнаруженными позами и реальными людьми с сохранением идентичности объектов во времени, т.е. отображение, сопоставляющее идентификатор $\mathrm{id}(i; B_1, \ldots, B_t)$ каждой обнаруженной позе таким образом, чтобы тот наилучшим образом соответствовал реальным людям в кадре:

$$\mathrm{id} : (i; B_1, \ldots, B_t) \mapsto k \in \mathbb{N}, \quad \text{где } \mathbb{N} \text{ задает множество траекторий} \tag{3}$$

Значение $\mathrm{id}(i; B_1, \ldots, B_t) = k$ обозначает, что обнаружение S_t^i (и, соответственно, рамка b_t^i) относятся к траектории (человеку) с номером $k \in \mathbb{N}$. Идентификатор может зависеть от рамок (включая степень уверенности), обнаруженных на всех предыдущих и текущем кадрах, и сопоставляется индексу i, $i \in \{1, \ldots, N_t\}$. Описанное отображение задано и является инъективным на каждом кадре.

Несмотря на то, что людей в кадре может быть более 1, пациент, движение которого нас интересует, лишь один. Требуется для начала выбрать пациента в кадре в автоматическом режиме, т.е. выбрать последовательность S_t^i поз с таким значением id, которое соответствует главному человеку в кадре, т.е. пациенту. Для этого вводится метрика качества траектории.

Перейдем к менее громоздким обозначениям.

Пусть $(\tau_1, \ldots, \tau_N)$ — вектор траекторий, соответствующих разным людям, где

$$\tau_k = \big((t_1^k, S_{t_1^k}^{i_1^k}, b_{t_1^k}^{i_1^k}), \ldots, (t_{T_k}^k, S_{t_{T_k}^k}^{i_{T_k}^k}, b_{t_{T_k}^k}^{i_{T_k}^k})\big), t_1^k < t_2^k < \cdots < t_{T_k}^k :$$

$$\forall k = 1, \ldots, N \implies \big((i,j) \in \{(i_1^k, t_1^k), \ldots, (i_{T_k}^k, t_{T_k}^k)\} \iff \mathrm{id}(i; B_1, \ldots, B_j) = k\big)$$

τ_k — последовательность 2D поз, связанных с человеком с идентификатором k. Здесь N есть множество всевозможных обнаруженных id. Так, траектория τ_k есть упорядоченный набор троек вида (кадр, 2D точки, рамка), где тройка содержится в данном наборе тогда и только тогда, когда соответствующей позе определен $\mathrm{id} = k$.

Пусть $Q : \tau \to \mathbb{R}$ – метрика качества траектории (здесь τ есть множество траекторий). Тогда под главным человеком в кадре (пациентом) подразумеваем траекторию с идентификатором:

$$k^* = \arg \max_{k=1,\ldots,N} Q(\tau_k)$$

Главная траектория τ_{k^*} далее используется для 3D реконструкции и анализа движений пациента.

4. 3D реконструкция позы

Требуется преобразовать 2D ключевые точки позы в 3D координаты (т.е. выполнить т.н. лифтинг из 2D в 3D). Пусть τ_{k^*} есть траектория пациента:

$$P = (P_1, \ldots, P_T) = \mathcal{R}(\tau_{k^*}) \tag{4}$$

$\mathcal{R}$ – оператор реконструкции 3D позы по 2D ключевым точкам. M — количество ключевых точек в 3D пространстве (в общем случае может отличаться от K).

5. Слияние

Построить метод, позволяющий использовать позы $P^{(1)}$ и $P^{(2)}$, предсказанные на этапе (4), для получения улучшенной итоговой оценки трехмерной позы P' в случае наличия нескольких (здесь – двух) камер.

Описанные отображения могут неявно зависеть от T – количества кадров в исходном видеоряде, от H, W – ширины и высоты видео соответственно – и, например, количества кадров в секунду, а также прочих метаданных. Тогда для входных данных с разными параметрами такого рода понимаются, вообще говоря, различные отображения.

Аналогично, стоит заметить, что в рамках практической реализации часть алгоритмов принимает в качестве входа нефиксированное число переменных, и тогда под, например, $\mathcal{R}(\tau_{k^*})$ понимаются разные отображения для разных траекторий.

2 Обзор существующих решений

Рассмотрим некоторые из современных методов оценки позы человека, применимых для анализа видеоданных. Укажем достоинства и недостатки, после чего поясним выбор конкретных моделей для данной работы.

2.1 2D и 3D оценка позы человека

Методы обработки видео, позволяющие извлечь 3D точки, можно разделить на две группы. Одни явно используют 2D точки, разделяя обработку видео на два этапа: извлечение двумерных и затем восстановление (лифтинг) трехмерных координат [19]. Другие же методы используют видеопоследовательность непосредственно для извлечения трехмерных координат [13; 16], используя 2D координаты только на этапе обучения, но не предсказания. Хотя бывают и модели, которые используют гибридный подход, принимая во внимание как визуальные признаки, так и двумерные координаты [29].

Есть методы, использующие различные конфигурации камер: предполагающие наличие сразу нескольких камер или же только одной.

Нейросетевые модели, предназначенные для задач восстановления как 2D, так и 3D координат, используют различную архитектуру. Модели, показывающие наилучшие результаты на бенчмарках, например, связанных с датасетами Human3.6M [12], MPI-INF-3DHP [5] и 3DPW [22], все имеют трансформерную архитектуру. Под точностью в данном абзаце подразумевается погрешность в предсказанных трехмерных координатах. Для оценки моделей, предсказывающих 2D точки (которые затем подаются на вход моделям, восстанавливающим 3D точки), используются иные бенчмарки. Однородность

же связана с тем, что нейросеть-трансформер способна учитывать контекст, длительные зависимости в последовательности данных, что необходимо при обработке видео, где множество предыдущих и последующих кадров тесно связаны с текущим. Иные методы также могут использовать нейросети и других архитектур. Популярное решение VideoPose3D [2] использует сверточные нейросети. По точности оно, впрочем, уступает более современным методам.

Бенчмарки позволяют количественно оценить предсказательную способность моделей.

Например, метод MotionBERT [19], используемый в данной работе – это модель, принимающая только двумерные координаты, но не визуальные признаки. Данная модель является SOTA на бенчмарке, связанном с датасетом Human3.6M, в котором видеокамера стационарна.

Подзадачей в рамках извлечения 3D точек из видео является обнаружение 2D точек. Существует множество методов, предназначенных для этого. Нейронные сети, лежащие в основе этих методов, также имеют различную архитектуру. Например, нейросеть-трансформер лежит в основе ViTPose [28] – точного, но требовательного к ресурсам метода извлечения 2D координат. Часто в решениях используются сверточные нейросети. В качестве примера: HRNet [7] и используемый в рамках этой работы RTMPose [23].

Одной из проблем выбора 2D модели является наличие готовых решений, осуществляющих детекцию точек в подходящем формате. Например, в одном из самых популярных форматов двумерной позы COCO [17] отсутствуют некоторые ключевые точки, которые может принять на вход модель, осуществляющая лифтинг 2D координат в 3D. Существует целое множество форматов 2D позы. Например: MPII [1], COCO-Wholebody [30], AIC [15], Halpe26 [4]. Одни модели лучше совместимы с наиболее популярным форматом 3D позы Human3.6M, а другие – хуже. Поэтому в рамках задачи извлечения 3D точек следует учитывать совместимость форматов 2D и 3D точек, т.к. отсутствие (неверное расположение) некоторых может значительно ухудшить результаты.

Почти все существующие решения предлагают формат COCO (упомянутые ViTPose, HRNet, RTMPose в т.ч.). Формат Halpe26 наилучшим образом совместим с Human3.6M, в котором предсказывают модели, осуществляющие лифтинг 2D в 3D точки, но решения, которые могут детектировать в данном формате, уже не столь распространены.

Современные методы позволяют в том числе обучиться на точках одного формата, а выводить другие, но в данном случае неизбежен процесс потери информации.

Задача трекинга также имеет множество решений (в т.ч. готовых), хорошо обобщаемых на произвольные видеопоследовательности. Например, OC-SORT [21], BoT-SORT [3], BoostTrack++ [26] – это решения, показывающие высокий результат на бенчмарке MOT17 [18]. В случае задачи извлечения 3D из видео с пациентом качество трекинга, если оно выше некоторого порога, оказывает лишь незначительное влияние на результат – хотя в этой работе использован SOTA-алгоритм, различия хорошо видны лишь при наличии окклюзий (ситуаций, когда один объект заслоняет собой другой), чего нет в данных, использованных в данной работе.

В итоге, стоит заметить, что есть множество решений поставленных задач 1-5 по отдельности, однако при решении нескольких из этих задач для in-the-wild видео (т.е. видео не из датасета, на котором обучалась модель) готовые алгоритмы часто комбинируют решения неэффективно. Например, для SOTA-алгоритма из этой работы [19] существует сразу несколько реализаций с различными недостатками. В их числе: отсутствие трекинга, что делает невозможным применение в условиях поставленной задачи, отсутствие ансамблирования, неэффективная нормализация данных (т.е. произведенная в неоптимальных местах пайплайна), детекция в несовместимом формате и использование устаревших моделей на этапе получения 2D точек. Последнее, как показала практика, значительно распространено в готовых реализациях алгоритмов лифтинга (в частности, [7; 9; 19]).

2.2 RTMPose и извлечение 2D ключевых точек

RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose [23] — это модель для оценки позы человека, которая может работать в реальном времени, разработанная в рамках фреймворка MMPose. Основана на так называемом top-down подходе (сначала производится детекция людей с помощью внешнего детектора (например, RTMDet), а затем – оценка позы для каждого человека отдельно) с использованием сверточной нейросети с архитектурой CSPNeXt и алгоритма SimCC (который позволяет получить точки с субпиксельной точностью: так, вместо дискретных значений, координаты точек задаются действительными числами).

Данная модель выбрана, т.к. позволяет достичь высокой точности при относительно небольшом потреблении ресурсов, что делает ее пригодной для поставленной задачи. Главное достоинство – возможность детекции в формате Halpe26, что обеспечивает максимальную совместимость результатов (2D точек) с форматом входных данных моделей для лифтинга в 3D. Для сравнения, метрика средней точности (AP) на бенчмарке COCO для RTMPose-l, использованного в данной работе, составляет 76.6%, в то время как популярный фреймворк для 2D детекции AlphaPose демонстрирует 72.6% [23] на своей лучшей модели (AlphaPose представляет набор моделей).

2.3 MotionBERT и восстановление 3D позы

Модель MotionBERT [19] — это SOTA (на бенчмарке Human3.6M) модель для восстановления трехмерной позы человека на основании 2D точек, полученных некоторым алгоритмом для извлечения 2D точек с одной камеры (т.е. модель выполняет лифтинг). В то время как 2D детекция и трекинг не столь сильно влияют на конечный результат (3D точки) в зависимости от поведения на бенчмарках, выбор модели для лифтинга, которая наилучшим образом показывает себя на соответствующих бенчмарках – это необходимость. Практика показала, что модели, лишь немногим уступающие MotionBERT [2; 9; 13], показывают визуально значительно худшие результаты.

MotionBERT построена на основе двухпотокового пространственно-временного трансформера (Dual-stream Spatio-temporal Transformer (DSTformer)), который обрабатывает пространственные и временные зависимости между суставами скелета с помощью двух параллельных потоков — пространственного и временного. Такая структура позволяет модели учитывать как геометрические связи между суставами, так и динамику движения во времени.

MotionBERT предлагает возможность использования окон различной ширины (например: 27, 81, 243), в которых учитывается контекст. При отсутствии ограничений по ресурсам разумнее выбрать большее окно, т.к. при большем окне улучшается способность модели учитывать взаимосвязи.

Данная модель также обучена работать с сильно зашумленными данными, что позволяет готовым реализациям использовать шумную 2D детекцию с удовлетворительным конечным результатом, но в рассматриваемом случае используется

точная 2D детекция, что уменьшает итоговую ошибку.

Формат входных данных для модели – Human3.6M. Этот формат отличается от Halpe26, используемого в 2D детекции, лишь присутствием средней точки на позвоночнике (при отображении остальных точек позы из Halpe26 в соответствующие точки Human3.6M). Результаты оценки двумерной позы готовыми моделями показали, что те из них, что изначально предлагают детекцию в желаемом формате (Human3.6M), демонстрируют значительно худшие результаты на in-the-wild видео, чем RTMPose, оценивающий 2D позу в Halpe26. Был сделан вывод о целесообразности использования формата без точки на позвоночнике, но с лучшей точностью детекции.

В сравнении с популярным методом VideoPose3D извлечения 3D точек из видео, MotionBERT показал среднюю ошибку 39.2 по MPJPE (среднее отклонение расположения суставов) на окне из 243 кадров при ошибке 46.8 у VideoPose3D на бенчмарке Human3.6M [19] с использованием протокола, подразумевающего 2D детекцию перед предсказанием 3D позы.

3 Реализация метода

Опишем реализацию выбранного подхода к восстановлению 3D позы человека из 2D ключевых точек, извлечённых с видео. Рассмотрим архитектуру пайплайна, включающего модели для 2D детекции и 3D лифтинга, формат используемых входных и выходных данных, а также методы обработки многокамерных данных для повышения точности восстановления трехмерных координат.

3.1 Архитектура пайплайна и используемые модели

Для извлечения 3D точек пациента из необработанного видео в рамках пайплайна (т.е. метода обработки видео) используется несколько последовательных этапов:

Для начала при помощи алгоритма RTMPose и встроенного детектора для каждого кадра I_t извлекаются рамки $B_t = \left(b_t^1(I_t), \ldots, b_t^{N_t}(I_t)\right)$ (задача 1) и двумерные позы $S_t^i = \left(p_1(I_t, b_t^i), \ldots, p_K(I_t, b_t^i)\right)$ (задача 2).

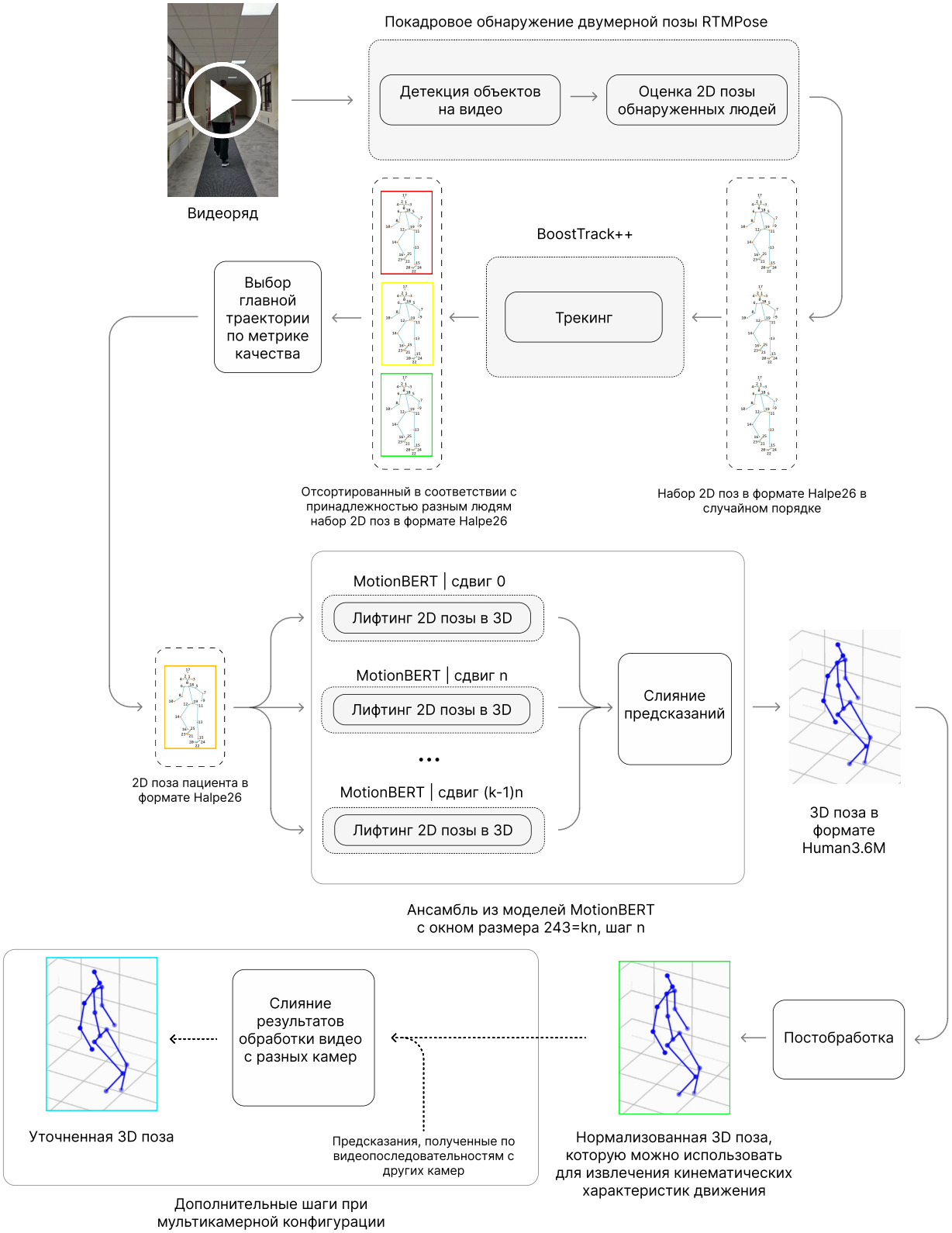

Рис. 1: Схема пайплайна

Далее при помощи алгоритма трекинга BoostTrack++ обнаруженные позы сопоставляются людям в кадре. Таким образом, каждому человеку соответствует траектория

$$\tau_k = \left((t_1^k, S_{t_1^k}^{i_1^k}, b_{t_1^k}^{i_1^k}), \ldots, (t_{T_k}^k, S_{t_{T_k}^k}^{i_{T_k}^k}, b_{t_{T_k}^k}^{i_{T_k}^k})\right), t_1^k < t_2^k < \cdots < t_{T_k}^k$$

По метрике Q, отображающей траектории τ_k в числа из $\mathbb{R}$, автоматически выбирается поза, соответствующая пациенту (задача 3).

На следующем этапе выбранной 2D позе сопоставляется 3D поза с помощью модели для лифтинга 2D координат в трехмерное пространство MotionBERT. Модель получает на вход траекторию τ_{k^*} и использует информацию на некотором окне (т.е. контекст) для восстановления трехмерной позы (задача 4):

$$P = (P_1, \ldots, P_T) = \mathcal{R}(\tau_{k^*})$$

Пайплайн допускает использование нескольких моделей, работающих с пересекающимися окнами, для увеличения точности и плавности получаемой позы. В конце также возможна постобработка для приведения масштаба обнаруженной 3D позы на всей последовательности кадров. На выходе получается 3D поза, из которой можно извлечь кинематические характеристики движения для дальнейшей работы с ними. При наличии нескольких камер, пайплайн также допускает слияние результатов, полученных с нескольких видео, в более точную позу (задача 5).

Схематично описанная последовательность шагов изображена на рис. 1.

3.2 Входные данные

Входные данные представляют собой набор необработанных видеозаписей (датасет), на которых пациент передвигается по коридору по направлению от камеры или к камере. Для части пациентов приведены видео, в которых оба процесса запечатлены одновременно на две камеры, расположенные по обеим сторонам коридора соответственно. В случае мультикамерной конфигурации отсутствует информация о внешних и внутренних параметрах конфигурации (взаимное расположение камер, фокусное расстояние), а также отсутствует синхронизация во времени.

3.3 Реализация обнаружения объектов и оценки 2D позы

Как детекция объектов на видео (и извлечение рамок), так и оценка двумерной позы произведены при помощи инструментов фреймворка MMPose [32].

В автоматическом режиме сначала при помощи встроенного детектора обнаруживаются объекты и извлекаются ограничивающие рамки, затем реализованный в рамках фреймворка алгоритм RTMPose извлекает 2D координаты. На выходе имеются как рамки B_t, включая степень уверенности, так и 2D позы S_t^i.

Реализация произведена на языке Python с использованием API MMPose.

3.4 Реализация трекинга и выбора главной траектории

Трекинг произведен при помощи средств языка Python и фреймворка boxmot [31], предоставляющего API к реализации алгоритма BoostTrack++. На этапе трекинга алгоритм сопоставляет набору рамок и 2D точек id, после чего соответствующие рамки и позы объединяются в траектории τ_k так, чтобы в рамках одной траектории были все позы, соответствующие одному конкретному идентификатору.

Опишем метрику Q, присваивающую траектории число из отрезка $[0, 1]$. $(\tau_1, \ldots, \tau_N)$ — вектор траекторий, полученный при помощи трекинга, где, как и ранее, траектория есть массив из номеров кадров, рамок и поз, соответствующих конкретному идентификатору:

$$\tau_k = \left((t_1^k, S_{t_1^k}^{i_1^k}, b_{t_1^k}^{i_1^k}), \ldots, (t_{T_k}^k, S_{t_{T_k}^k}^{i_{T_k}^k}, b_{t_{T_k}^k}^{i_{T_k}^k})\right), t_1^k < t_2^k < \cdots < t_{T_k}^k :$$

$$\forall k = 1, \ldots, N \implies \left((i, j) \in \{(i_1^k, t_1^k), \ldots, (i_{T_k}^k, t_{T_k}^k)\} \iff \mathrm{id}(i; B_1, \ldots, B_j) = k\right)$$

Приведем интуитивные соображения, в соответствии с которыми будет задана метрика для конкретной траектории τ_k:

1) У каждой рамки $b_t^i(I_t) = \left(x_t^{i,1}, y_t^{i,1}, x_t^{i,2}, y_t^{i,2}, c_t^i\right)$ в каждом кадре I_t есть степень уверенности в детекции c_t^i. Эта характеристика учитывает, в числе прочего, удаление человека от камеры, а также наличие окклюзий. Чем человек важнее в кадре, тем, ожидается, уверенность в его детекции на протяжении всего видео больше. Требуется учитывать уверенность на всех кадрах. Разумно задать ее как 0 в кадрах, где данный человек не был обнаружен. Пусть $\forall t \in \{1, \ldots, T\}$ если $t \in \{t_1^k, \ldots, t_{T_k}^k\}$, то это значит, что в данном кадре обнаружена некоторая поза S_t^i с идентификатором k, тогда обозначим

соответствующую уверенность c_t^i как $\tilde{c}_t^k$. Иначе $\tilde{c}_t^k = 0$. Т.е.:

$$\tilde{c}_t^k = \begin{cases} c_t^i, & \text{если в кадре } t \text{ есть некоторая поза } S_t^i \text{ с присвоенным ей идентификатором } k, \\ 0, & \text{иначе.} \end{cases}$$

Тогда средняя уверенность на всем видео равна:

$$\bar{c}^k = \frac{1}{T} \sum_{t=1}^{T} \tilde{c}_t^k$$

Уверенность $\tilde{c}_t^k$ на каждом кадре лежит на отрезке от 0 до 1 $\implies$ $\bar{c}^k \in [0, 1]$.

2) Ожидается, что главный в видео человек будет присутствовать в как можно большем количестве кадров.

Зададим функцию-индикатор присутствия человека в кадре a_t^k как:

$$a^k = \frac{T_k}{T}$$

Т.е. a^k равен доле кадров видео, где данный человек был обнаружен. $a^k \in [0, 1]$.

3) На важность человека в видео также непременно влияет размер рамки. Чем больше рамка, тем ближе человек к камере, и наоборот. Ожидается, что более важный в кадре человек будет находиться ближе к камере. На основании этого, для каждого кадра введем функцию $\tilde{b}_t^k$, зависящую от рамки, принимающую значения в $[0, 1]$. На кадрах, где человек не обнаружен, присвоим функции значение 0:

$$\tilde{b}_t^k = \begin{cases} \frac{(x_t^{i,2} - x_t^{i,1})^2 + (y_t^{i,2} - y_t^{i,1})^2}{H^2 + W^2}, & \text{если в кадре } t \text{ есть поза } S_t^i \text{ с соотв. идентификатором } k, \\ 0, & \text{иначе.} \end{cases}$$

H и W – высота и ширина видео соответственно (разрешение), а координаты взяты из соответствующего представления рамки: $b_t^i(I_t) = \left(x_t^{i,1}, y_t^{i,1}, x_t^{i,2}, y_t^{i,2}, c_t^i\right)$.

В качестве размера рамки взята диагональ. Диагональ рамки не может быть больше диагонали изображения. Отсюда, $\tilde{b}_t^k \in [0, 1]$.

Соответственно, введем среднее:

$$\bar{b}^k = \frac{1}{T} \sum_{t=1}^{T} \tilde{b}_t^k, \quad \bar{b}^k \in [0, 1]$$

4) Наконец, стоит заметить, что почти всегда главный в кадре человек находится ближе к центру кадра. Получается, разумно ввести меру того, насколько человек близко к центру.

За центр кадра возьмем $(\frac{W}{2}, \frac{H}{2})$. Как и ранее, на кадрах, где человек не попадает в кадр, зададим соответствующий функционал как минимально возможный (ноль). Наибольшее удаление от центра – углы изображения.

Центр рамки расположен в точке: $(\frac{x_t^{i,1}+x_t^{i,2}}{2}, \frac{y_t^{i,1}+y_t^{i,2}}{2})$.

Тогда метрика удаления от центра $\tilde{d}_t^k$ в кадре равна:

$$\tilde{d}_t^k = \begin{cases} 1 - \frac{\left(\frac{W}{2} - \frac{x_t^{i,1}+x_t^{i,2}}{2}\right)^2 + \left(\frac{H}{2} - \frac{y_t^{i,1}+y_t^{i,2}}{2}\right)^2}{\left(\frac{H}{2}\right)^2 + \left(\frac{W}{2}\right)^2}, & \text{если в кадре } t \text{ есть поза } S_t^i \text{ с идентификатором } k, \\ 0, & \text{иначе.} \end{cases}$$

$\tilde{d}_t^k \in [0, 1]$, причем значение 1 принимается при совпадении центра рамки с центром изображения, а 0 – на максимальном удалении от центра.

Соответственно, среднее:

$$\bar{d}^k = \frac{1}{T} \sum_{t=1}^{T} \tilde{d}_t^k, \quad \bar{d}^k \in [0, 1]$$

В итоге, имеем 4 значения $\bar{c}^k, a^k, \bar{b}^k, \bar{d}^k$, причем все из них лежат в $[0, 1]$.

Зададим итоговую метрику качества траектории Q как:

$$Q(\tau_k) = w_1 \bar{c}^k + w_2 a^k + w_3 \bar{b}^k + w_4 \bar{d}^k$$

В более общем случае можно было бы учесть не только рамки b_t^i, но и позы S_t^i. Например, так можно лучше учесть окклюзии. В рассматриваемом случае в этом нет необходимости.

В данной работе используется лишь небольшой датасет, потому веса можно выбрать равными:

$$w_1 = w_2 = w_3 = w_4 = \frac{1}{4}$$

При наличии большого датасета с разметкой на нем, веса $w_1, \ldots, w_4$ можно задать с учетом максимизации количества верно классифицированных объектов (т.е. так, чтобы на максимальном числе видео главный человек определялся правильно).

После этого остается лишь выбрать такой идентификатор k, что на нем достигается максимум $Q(\cdot)$, т.е.:

$$k^* = \arg \max_{k=1,\ldots,N} Q(\tau_k)$$

Заданная таким образом метрика с соответствующими весами позволяет определить главного человека (пациента) верно на всех видео, использованных в рамках данной работы (т.е. на 100% объектов).

Траектория пациента τ_{k^*} далее используется для 3D реконструкции.

3.5 Реализация лифтинга и ансамблирование

Для восстановления 3D позы применялся исходный код, опубликованный вместе с оригинальной статьей MotionBERT [19], с модификациями, внесенными в рамках данной работы. Был использован интерфейс модели, доступный через командную строку, а также средства языка Python. На этапе лифтинга алгоритм сопоставляет траектории τ_{k^*} набор трехмерных координат $P = (P_1, \ldots, P_T)_{j=1}^{M}$

На практике было получено, что MinMax-нормализация, выполняемая на окнах из кадров по отдельности, эффективнее, чем MinMax-нормализация, либо выполненная на всем видео, либо не выполняемая в принципе. Под MinMax-нормализацией здесь подразумевается приведение данных к значениям из $[0, 1]$ одинаково по всем осям (с сохранением пропорций), основываясь на максимальном и минимальном значениях на заданном диапазоне кадров. Соответственно, на каждом окне из кадров выполнена нормализация на, вообще говоря, разное значение константы нормализации.

Выбран размер окна 243, т.к. на бенчмарке MotionBERT показывает наилучшие результаты именно с этим размером окна.

Выбрано количество базовых моделей ансамбля 3 с шагом в 81 кадр. Т.е., возвращаясь к рис. 1, $243 = kn$, где k взят 3. Соответственно, $n = 81$. В экспериментах это позволило достичь плавного перехода при переключении между окнами. Пересечение окон между моделями, отличающимися по входным данным на шаг n относительно друг от друга, есть $(k-1)n = 162$ кадра. Взят равномерный шаг, поскольку при неравномерном шаге пересечение между окнами различно по ширине; отсюда потенциально неравномерное качество восстановления позы.

Иными словами, модель 1 восстанавливает позу на окне от кадра 1 до 243 (оба включительно), далее на окне от 244 до 486 (оба включительно) и т.д. со сдвигом 243; модель 2 – на кадрах от 82 до 324 (оба включительно) и т.д.; модель 3 – на кадрах от 163 до 405 (оба включительно) и т.д. Сохраним эти обозначения далее.

Таким образом, на каждом кадре есть 3 предсказания, а модели при предсказании имеют различный контекст, что позволяет добиться на практике получения положительного эффекта ансамблирования (улучшение качества результатов после слияния).

Практика показала, что предсказания модели на кадре, в котором отсутствует контекст с одной из сторон (например, на кадре 1 или кадре 243 для одной из моделей ансамбля), имеют значительную погрешность. Желательно, чтобы в любой момент времени веса, с которыми берутся предсказания моделей ансамбля, были таковы, чтобы больший вес был у той модели, которая имеет наибольший двусторонний контекст, ведь архитектура DSTFormer (на которой основан MotionBERT) имеет ту же особенность, что и BERT [6], где отсутствие контекста с одной из сторон существенно ухудшает результаты предсказания.

Т.е. следует выбрать веса $w_1(t), w_2(t), w_3(t)$, с которыми суммируются предсказания P_t^1, P_t^2, P_t^3 моделей 1, 2, 3 в рамках ансамбля соответственно на данном кадре I_t, зависящими от времени, причем следующим образом:

Пусть $\{\tilde{t}_1^i, \ldots, \tilde{t}_{243}^i\}$ – окно, на котором предсказывает модель с номером i в настоящий момент. Здесь момент времени $\tilde{t}_k^i$ соответствует кадру $I_{\tilde{t}_k^i}$. Тогда пускай текущий момент времени t равен $\tilde{t}_{k_1}^1, \tilde{t}_{k_2}^2, \tilde{t}_{k_3}^3$ соответственно. Считаем, что модель работает только с окнами длины 243, а $T \geq 243$. Под контекстом модели j на кадре t будем подразумевать:

$$context_t^j = \min\{\tilde{t}_{k_j}^j - \tilde{t}_1^j, \tilde{t}_{243}^j - \tilde{t}_{k_j}^j\} = \min\{k_j - 1, 243 - k_j\}$$

Отсюда, желаемое:

$$w_i(t) > w_j(t) \text{ при условии } context_t^i > context_t^j \tag{5}$$

Нетрудно заметить, что данное условие не выполнено при $w_1 = w_2 = w_3 = \frac{1}{3}$. Пример: при заданных весах на кадре 243:

$$context_{243}^1 = 0, \quad context_{243}^2 = 81, \quad context_{243}^3 = 80$$

$$\text{Однако, } w_1(243) = w_2(243) \text{ при том, что } context_{243}^1 < context_{243}^2$$

Отсюда необходимость зависимости от времени t.

Но этого условия недостаточно. На практике при переключении между окнами предсказания получаются значительно различными, что устраняет плавность итогового

результата. Требуется ее сохранить, задав веса на граничных кадрах (отличных от начального $t = 1$ и, возможно, конечного $t = T$) меньшими некоторого порога ϵ. На основе экспериментальных данных выбирается

$$\epsilon = 0.02, \text{ при условии } w_1 + w_2 + w_3 = 1 \tag{6}$$

Утверждение.

$$\text{Пусть } G(t) = \exp\left(-\frac{1}{2}\left(\frac{((t-1) \bmod 243) - 121}{40}\right)^2\right)$$

Здесь mod обозначает остаток от деления.

Тогда веса

$$w_1(t) = \frac{G(t)}{G(t) + G(t-81) + G(t-162)}$$
$$w_2(t) = \frac{G(t-81)}{G(t) + G(t-81) + G(t-162)}$$
$$w_3(t) = \frac{G(t-162)}{G(t) + G(t-81) + G(t-162)}$$

удовлетворяют условиям (5), (6) для кадров $t = 163, \ldots, (T - T \bmod 81) - 162$.

Доказательство. $\forall t\ G(t_1) = G(t_2),$ если $t_1, t_2 \geq 1$ и $|(t_1 - 1) \bmod 243 - 121| = |(t_2 - 1) \bmod 243 - 121|$. Также $G(t) = G(t + 243)$ при $t \geq 1$.

Отсюда:

$$w_1(243) = w_1(244) = w_1(486) = w_1(487) = \ldots$$

Численные значения функции: $G(81) \approx 0.5914, \quad G(162) \approx 0.6065, \quad G(243) \approx 0.0103$

$$w_1(243) = \frac{G(243)}{G(243) + G(162) + G(81)} \approx 0.0085 < \epsilon$$

Аналогично,

$$w_2(243 + 81) = w_2(244 + 81) = w_2(486 + 81) = \cdots = w_1(243) < \epsilon$$

$$w_3(162) = w_3(163) = w_3(243 + 162) = w_3(244 + 162) = \cdots = w_1(243) < \epsilon$$

Здесь значения t взяты так, что в любой момент времени доступны предсказания всех трех моделей.

Получили (6). Далее рассмотрим (5):

Заметим, что выражение $|(t-1) \bmod 243-121|$ для $t \in \{1,\ldots,243\}$ тем более, чем далее от t центр окна 1-243; $|(t-81-1) \bmod 243-121|$ для $t \in \{1+81,\ldots,243+81\}$ тем более, чем далее от t центр окна (1 + 81)-(243 + 81), и аналогично для $|(t-162-1) \bmod 243-121|$.

Вспомним, что $context_t^j$ равен минимальному расстоянию до края обрабатываемого окна от кадра t.

Найдем взаимосвязь между $context_t^1$ и $g(t) = |(t-1) \bmod 243 - 121|$ для модели 1 на окне из кадров с номерами 1-243:

$$context_t^1 = \min\{t-1, 243-t\}$$

Для $t \in \{1,\ldots,122\} \implies g(t) = 121-(t-1) = 121 - context_t^1$

Для $t \in \{122,\ldots,243\} \implies g(t) = (t-1) - 121 = -(243-t) + 121 = 121 - context_t^1$

Значит, чем больше $context_t^1$, тем меньше значение $(t-1) \bmod 243 - 121$, а значит, тем больше $G(t)$ на соответствующих t. Осталось показать, что при приближении к центру окна $G(t)$ возрастает не медленнее, чем сумма в знаменателе (и при удалении убывает не медленнее, чем сумма в знаменателе).

Вычисления производной $w_1(t)$ приведены в приложении А.1.

В центре окна ($t = 122$) она равна нулю, слева ($82 \le t < 122$) для любого t она больше нуля, а справа ($162 \ge t > 122$) меньше. Следовательно, эта функция достигает на центральном кадре максимума. Так, вес $w_1(t)$ тем меньше, чем дальше от центра находится текущий кадр (т.е. чем меньше контекст). Для $1 \le t \le 81$ производная строго больше 0, а для $163 \le t \le 243$ – строго меньше, причем для кадров $t = 81$, $t = 82$, $t = 162$, $t = 163$ верны соотношения:

$w_1(81) < w_1(82)$, $w_1(162) > w_1(163)$.

Функция $w_2(t)$ на окне $t \in \{1+81,\ldots,243+81\}$ равна $w_1(t-81)$. Для функции $w_3(t)$ верно аналогичное свойство.

Из полученного ранее:

$$G(t) = \exp\left(-\frac{1}{2}\left(\frac{121 - context_t^1}{40}\right)^2\right)$$

и $w_1(t)$ убывает/возрастает одновременно с $G(t)$ и имеет экстремум в той же точке, что и $G(t)$.

Так, верно, что чем больше контекст $context_t^i$, тем меньше для данной модели i расстояние от центра окна и тем больше вес $w_i(t)$. Т.к. для моделей i и j $121 - context_t^i$

равен $121 - context_t^j$ на равных расстояних от центра, а веса $w_i(t)$ и $w_j(t)$ одинаковы и увеличиваются/уменьшаются одновременно вместе с увеличением/уменьшением контекста, то, стало быть, при $context_{t_1}^i > context_{t_2}^j \iff w_i(t_1) > w_j(t_2)$

Это, в частности, доказывает свойство (5).

□

В рамках технической реализации, веса для t, не попавших в рассматриваемое окно ($t \notin \{163, \ldots, (T - T \bmod 81) - 162\}$), выбраны следующим образом: в $t \in \{1, \ldots, 81\}$ веса моделей 2, 3 приняты равными нулю; в $t \in \{82, \ldots, 162\}$ вес модели 3 принят нулевым, а вес модели 2 задан как $w_2(t) = \frac{G(t-81)}{G(t)+G(t-81)}$; в точках же $t \in \{T - T \bmod 81 - 162 + 1, \quad T\}$ веса заданы как в утверждении, однако модели используют лишь часть окна для предсказания, имея в каждый момент времени урезанный контекст. На практике выбранные таким образом веса позволяют достичь плавного перехода для всех t, не попавших в рассмотрение: между $t = 81, t = 82$, между $t = 162, t = 163$ и аналогичных t из $\{T - T \bmod 81 - 162, \quad T\}$.

На выходе данного этапа получается трехмерная поза:

$$P_t = w_1(t)P_t^1 + w_2(t)P_t^2 + w_3(t)P_t^3 \tag{7}$$

Здесь за P_t^i обозначено предсказание i-й модели ансамбля на кадре I_t.

MotionBERT позволяет получить позу относительно некоторой точки позы (например, одной из точек таза). Для восстановления абсолютных координат (т.е. с учетом движения человека по некоторой траектории в пространстве) могут использоваться иные модели. Для рассматриваемой задачи выполнено восстановление именно относительной позы.

3.6 Постобработка

MotionBERT предсказывает трехмерную позу с сохранением эффекта уменьшения/увеличения размера позы в двумерном случае (чем дальше от камеры пациент, тем он кажется меньше). Поэтому после предсказания трехмерной позы ансамблем моделей MotionBERT координаты позы нормализуются (ведь в реальности изменения размера позы нет – у человека постоянный размер). В случае одной камеры это может потребоваться для дальнейшего автоматического извлечения кинематических

характеристик движения. В случае нескольких камер, это необходимо перед применением метода слияния предсказаний, полученных с помощью видов с нескольких (двух) камер.

Модель трехмерной позы Human3.6M включает 17 точек позы. Обозначим за $d(t; i, j)$ евклидово расстояние между точками позы с номерами i и j в рамках предсказанных трехмерных координат P_t в момент времени t. Назовем «основными расстояниями» вектор:

$$d(t) = (d(t; i_1, j_1), \ldots, d(t; i_n, j_n))^T \tag{8}$$

где $\forall k$ (i_k, j_k) принадлежит некоторому множеству D пар точек (причем перечислены все элементы D, т.е. $|D| = n$). Детали формата Human3.6M, а также множество D указаны в приложении А.2.

Формально, под нормализацией понимается следующее: Пусть задан вектор весов

$$\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_n)^T$$

Полагаем:

$$\tilde{w}_i = \text{const}, \quad \sum_{i=1}^{n} \tilde{w}_i = 1.$$

Под значением усреднения в кадре I_t понимаем значение функции:

$$norm(t) = \tilde{w}^T d(t)$$

Под процессом нормализации (или нормализацией) понимаем переход к нормализованным координатам:

$$\tilde{P}_t = \frac{P_t}{norm(t)} \tag{9}$$

где $P_t \in \mathbb{R}^{17\times 3}$ состоит из 17 точек, задаваемых тремя числами; и под делением подразумеваем операцию умножения матрицы P_t на скаляр $\frac{1}{norm(t)}$. Заметим, что $\forall t$ $norm(t) > 0$ в силу природы данных: в рамках выполнения работы было замечено, что точки никогда не накладываются друг на друга. Если бы это предположение не было выполнено, то можно было бы принять $\tilde{P}_t = \frac{P_t}{norm(t)+\epsilon}$, однако в технической реализации нормализация выполнена именно согласно (9).

Как пример, позу можно нормализовать на сумму всех основных расстояний с одинаковыми весами, т.е. $\tilde{w}_i = \frac{1}{n}$.

Рассмотрим алгоритм нормализации, при котором минимизируется сумма дисперсий основных расстояний:

Пусть

$$no\bar{r}m = \frac{1}{T}\sum_{t=1}^{T} \tilde{w}^T d(t)$$

Положим, соответственно

$$\tilde{d}(t;i,j) = \frac{1}{\tilde{w}^T d(t)} \cdot no\bar{r}m \cdot d(t;i,j)$$

Умножение на $no\bar{r}m$ необходимо, т.к. в противном случае основные расстояния с бóльшим средним будут давать бóльшие значения усреднения, а отсюда, ожидается, большие веса будут у наибольших расстояний. Желаемым же является сравнение дисперсий при, в целом, одном и том же масштабе позы вне зависимости от выбора весов $\tilde{w}$.

T – общее число кадров. Тогда дисперсия расстояния между точками (i_k, j_k):

$$\mathrm{Var}_t(\tilde{d}(t;i_k,j_k)) = \frac{1}{T}\sum_{t=1}^{T}\left(\tilde{d}(t;i_k,j_k) - \frac{1}{T}\sum_{s=1}^{T}\tilde{d}(s;i_k,j_k)\right)^2 \tag{10}$$

Задача оптимизации формулируется следующим образом:

$$J(\tilde{w}) = \sum_{k=1}^{n}\mathrm{Var}_t\left(\tilde{d}(t;i_k,j_k)\right) \;\to\; \min_{\tilde{w}\in W}, \quad W = \{\tilde{w} : \sum_{i=1}^{n}\tilde{w}_i = 1, \tilde{w}_i \geq 0\} \tag{11}$$

Функция $J(\tilde{w})$ определена на $(n-1)$-мерном симплексе. Это компакт.

Далее, $g(\tilde{w}) = \tilde{d}(t;i,j) = \frac{1}{\tilde{w}^T d(t)} \cdot no\bar{r}m \cdot d(t;i,j)$ – это непрерывная по $\tilde{w}$ функция (в знаменателе стоит непрерывная по $\tilde{w}$ функция, которая по предположению не обращается в 0). $\tilde{d}(t;i_k,j_k) - \frac{1}{T}\sum_{s=1}^{T}\tilde{d}(s;i_k,j_k)$ также непрерывна как разность непрерывной функции и конечной суммы непрерывных. Отсюда, $\mathrm{Var}_t(\tilde{d}(t;i_k,j_k))$ также непрерывна, и, следовательно, оптимизируемый функционал $J(\tilde{w})$ непрерывен.

По теореме Вейерштрасса о непрерывной функции на компакте, $J(\tilde{w})$ достигает своих верхней и нижней грани. Значит, решение задачи оптимизации существует, однако, вообще говоря, может быть не единственным.

Заметим, что у $J(\tilde{w})$ есть непрерывные первая и вторая производные (т.е. градиент и матрица Гессе): $\tilde{d}(t;i_k,j_k)$ есть константа, деленная на линейную функцию, и имеет непрерывные вторые частные производные. Аналогичным свойством обладает $\mathrm{Var}_t(\tilde{d}(t;i_k,j_k))$, которая является суммой квадратов функций с непрерывными вторыми частными производными. Наконец, отсюда, $J(\tilde{w})$ обладает тем же свойством.

В этих условиях, применим алгоритм SLSQP [14] для численного решения этой задачи. Этот алгоритм способен учитывать ограничения, присутствующие в данной задаче, и создан для задач с нелинейным функционалом с приведенными выше свойствами.

В таблице 1 представлены результаты сравнения по коэффициенту детерминации (R^2) двух веществennозначных функций от t:
1) функции $\frac{\tilde{w}^T d(t)}{\frac{1}{T}\sum_{t=1}^{T} \tilde{w}^T d(t)}$ и 2) функции $\frac{\tilde{w}_0^T d(t)}{\frac{1}{T}\sum_{t=1}^{T} \tilde{w_0}^T d(t)}$
где $\tilde{w}_0^T = (\frac{1}{n}, \ldots, \frac{1}{n})$, а $\tilde{w}$ есть полученное алгоритмом решение поставленной задачи оптимизации. Значения приведены для трех различных видео из датасета.

Номер видео	Значение метрики R^2
1	0.9998573606931374
2	0.9999895523366403
3	0.9999718075015277

Таблица 1: Сравнение функций нормализации

Приведенные данные показывают значительную схожесть двух функций.

Соответственно, вопрос, брать ли нормализацию на сумму основных расстояний с одними и теми же весами или же брать веса в рамках поставленной задачи оптимизации, несущественен, что показывают экспериментальные данные.

В технической реализации выполнена нормализация с одинаковыми весами.

После выполнения нормализации получаем итоговый результат предсказания 3D позы с одной камеры. Далее из полученных данных можно извлечь кинематические характеристики движения.

3.7 Слияние предсказаний в мультикамерной конфигурации

При наличии видов с нескольких камер выполняется та же последовательность действий, что была описана ранее, для видео, снятых с каждой из камер.

Итак, имеем $\tilde{P}_{t_1}^{(1)}$ и $\tilde{P}_{t_2}^{(2)}$ – предсказания 3D позы из видео с камеры 1 и с камеры 2 для каждых t_1 и t_2. В поставленной задаче камеры не откалиброваны, т.е. неизвестны внешние и внутренние параметры конфигурации, необходимые для применения метода триангуляции восстановления трехмерной позы. Но в этих условиях все еще можно скомбинировать виды с нескольких камер.

Также в поставленной задаче камеры несинхронизированные, что означает, что видео 1 сдвинуто на некоторое неизвестное небольшое число кадров относительно видео 2. Под небольшим числом кадров подразумевается, что съемка начиналась и останавливалась вблизи момента времени, определенного некоторым внешним маркером (пример: пациент дошел до конца зала или пациент поднял руку), и при этом разница во времени между началом съемки видео 1 и началом съемки видео 2, а также концом съемки видео 1 и концом съемки видео 2 не превосходит T_{cap} кадров.

Для приведенного далее метода автоматической синхронизации видео во времени необходимо, чтобы одно видео было вложено в другое во времени. При наличии предположения о максимальной разнице в кадрах между началами и концами съемки с нескольких камер соответственно вложенность можно обеспечить путем вырезания первых и последних T_{cap} кадров из одного из видео (пусть это будет видео 1). Далее считаем, что видео 1 вложено во времени в видео 2. Пусть видео 2 начинается с $t = 1$ и кончается $t = T$, а видео 1 начинается с $t = \tilde{t} \geq 1$ и кончается $t = \tilde{T} \leq T$.

Чтобы синхронизировать видео, необходимо определить, насколько похожи во времени две последовательности из поз $(\tilde{P}_1^{(1)}, \ldots, \tilde{P}_{\tilde{T}-\tilde{t}+1}^{(1)})$ и $(\tilde{P}_1^{(2)}, \ldots, \tilde{P}_T^{(2)})$ при различных сдвигах одной последовательности относительно другой во времени. Предполагается, что максимальная похожесть при сдвиге t^* будет означать, что t^* есть истинная задержка во времени одного видео относительно другого, выраженная в кадрах.

Вводится метрика похожести двух видеопоследовательностей при различных сдвигах. Так, сначала $\tilde{t}$ (время начала видео 1) принимается равным единице, а $\tilde{T}$ – равным длине видео 1; после этого вычисляется метрика похожести видео 1 на видео 2, как если бы оба видео начинались одновременно. На втором шаге $\tilde{t} = 2$, а $\tilde{T}$ принимается равным длине видео 1 плюс единица; вычисляется метрика похожести видео 1 на видео 2, как если бы более короткое видео 1 начиналось позже видео 2 на один кадр. Описанная процедура далее повторяется для остальных сдвигов, при которых $\tilde{T} \leq T$ при вложенности видео 1 в видео 2 во времени. На выходе получается последовательность значений метрики для всевозможных сдвигов видео 1 относительно видео 2. Заметим, что снятие ограничения на вложенность потребовало бы введения метрики для кадров, на которых сравнение невозможно, что потенциально ухудшило бы результаты автоматической синхронизации.

Упомянутая метрика похожести основана на наложении одной позы на другую с

помощью т.н. прокрустова преобразования [24]. На каждом шаге одна из поз наилучшим возможным образом накладывается на другую позу, и оценивается качество этого наложения. В последующем слиянии предсказаний также используется схожий метод.

3.7.1 Прокрустово преобразование

Пусть даны две матрицы $A, B \in \mathbb{R}^{n \times d}$. Требуется найти ортогональную матрицу $R \in \mathbb{R}^{d \times d}$, число $s \in \mathbb{R}$ и вектор переноса $\mathbf{t}_{\text{trans}} \in \mathbb{R}^d$ такие, что на них достигается

$$\min_{s,R,\mathbf{t}_{\text{trans}}} \|B - sAR - \mathbf{1}\mathbf{t}_{\text{trans}}^T\|_F^2$$

где $\mathbf{1} \in \mathbb{R}^n$ — вектор из единиц, и R удовлетворяет условию ортогональности: $R^T R = I$. $\mathbf{1}$ и $\mathbf{t}_{\text{trans}}$ есть вектор-столбцы.

Применительно к поставленной изначально задаче, A и B есть соответственно трехмерные позы, предсказанные по видео 1 и видео 2. Одна поза сдвинута, повернута и, возможно, отличается по размеру относительно другой, поэтому для наложения поз требуется повернуть одну из них в пространстве с помощью преобразования R, сдвинуть на $\mathbf{t}_{\text{trans}}$ и масштабировать на число s так, чтобы она накладывалась на другую оптимальным образом.

Решение этой задачи было предложено П. Шёнеманном и Р. Кэрроллом [25]. Его можно найти в явном виде:

1) Матрицы центрируются по столбцам:

$$\bar{A} = \frac{1}{n}\mathbf{1}^T A, \quad \bar{B} = \frac{1}{n}\mathbf{1}^T B,$$

$$A_0 = A - \mathbf{1}\bar{A}^T, \quad B_0 = B - \mathbf{1}\bar{B}^T.$$

2) Находится оптимальный вектор переноса:

$$\mathbf{t}_{\text{trans}} = \bar{B} - s\bar{A}R.$$

3) Из сингулярного разложения произведения матриц A_0^T, B_0 находится матрица R:

$$M = A_0^T B_0$$

$$M = U\Sigma V^T$$

$$R = VU^T$$

Если $det(R) < 0$, то имеем преобразование с отражением, но отражения не подходят по задаче. В этом случае последний столбец матрицы V умножается на -1. Это позволяет дополнить алгоритм так, чтобы учитывать ограничение $det(R) = 1$ [20; 27].

4) В общем случае далее находится оптимальный масштаб s, однако для трехмерных поз ранее была выполнена нормализация, а потому в подборе этого параметра нет необходимости. В таком случае s полагается равным единице [25].

Итоговое приближение:

$$B \approx AR + \mathbf{1}\mathbf{t}_{\text{trans}}^T.$$

В случае более чем двух матриц (т.е. более чем двух камер) формулируется похожая задача, и решение затем находится из итерационного процесса [11]. В рассматриваемом случае, однако, в датасете отсутствуют видео, записанные в конфигурации с более чем двумя камерами.

Матрицы A и B могут быть, вообще говоря, произвольного размера (но такими, что размер обеих совпадает), потому в задаче сравнения поз можно брать сразу целое окно из кадров, выполняя конкатенацию матриц, представляющих позу в кадре, так, чтобы увеличивалось количество строк. Т.е. решаемая задача будет рассматриваться как задача подбора преобразования, наиболее качественно приближающего набор трехмерных точек 1 к набору трехмерных точек 2, где оба набора выражены матрицами из $\mathbb{R}^{n \times d}$, $n = k \cdot 17$ (17 – число точек в формате Human3.6M), $d = 3$, k – число кадров в окне.

Техника подбора оптимального преобразования по окну из нескольких кадров на практике позволила получить более плавное выравнивание поз с уменьшенным влиянием случайных шумов и особенностей отдельных кадров в сравнении с покадровым выравниванием.

Итак, метрика, описывающая разницу между последовательностями поз из видео 1 и 2 при смещении видео 1 на $\tilde{t} - 1$ кадров вперед во времени относительно начала видео 2, задается следующим образом (считаем начало видео 2 моментом времени $t = 1$):

Пусть, как и ранее, $\tilde{P}_{t_1}^{(1)}$ и $\tilde{P}_{t_2}^{(2)}$ есть нормализованные предсказания трехмерных поз, полученных с видео 1 и видео 2 в моменты времени t_1 и t_2 соответственно. Введем матрицы, включающие предсказания на окне из кадров длины k:

$$M_{t_1} = \begin{pmatrix} \tilde{P}^{(1)}_{t_1} \\ \tilde{P}^{(1)}_{t_1+1} \\ \vdots \\ \tilde{P}^{(1)}_{t_1+k-1} \end{pmatrix} \in \mathbb{R}^{17k\times 3}, \qquad N_{t_2} = \begin{pmatrix} \tilde{P}^{(2)}_{t_2} \\ \tilde{P}^{(2)}_{t_2+1} \\ \vdots \\ \tilde{P}^{(2)}_{t_2+k-1} \end{pmatrix} \in \mathbb{R}^{17k\times 3}$$

Для матриц M_{t_1} и N_{t_2} можно найти матрицу поворота R и вектор переноса $\mathbf{t}_{\text{trans}}$ такие, что на них достигается минимум:

$$\min_{R,\mathbf{t}_{\text{trans}}} \|N_{t_2} - M_{t_1}R - \mathbf{1}\mathbf{t}^T_{\text{trans}}\|^2_F$$

Пусть рассматривается смещение видео 1 на $\tilde{t}-1$ кадров относительно начала видео 2. С шагом единица обрабатываются окна из поз:

$$M_j, N_{\tilde{t}+j}: \quad j = 1, \ldots, (\tilde{T} - \tilde{t} + 1) - k + 1$$

Обозначим за $R^{\tilde{t}}_j$ и $(\mathbf{t}_{\text{trans}})^{\tilde{t}}_j$ соответственно оптимальные матрицу поворота и вектор смещения в задаче

$$\min_{R,\mathbf{t}_{\text{trans}}} \|N_{\tilde{t}+j} - M_j R - \mathbf{1}\mathbf{t}^T_{\text{trans}}\|^2_F$$

На каждом шаге находятся оптимальные поворот и смещение позы из видео 1 так, чтобы та максимально качественно накладывалась на позу из видео 2 на окне из кадров. Поскольку окно состоит из нескольких кадров, а шаг равен единице, у окон есть пересечение. Для каждого кадра $t = 1, \ldots, \tilde{T} - \tilde{t} + 1$ усредняем выровненные позы по всем окнам, содержащим этот кадр:

$W_t = \{j : j \le t \le\ j + k - 1\}$ – количество окон, в которых есть поза на данном кадре t,

$$\hat{P}^{(1)}_t(\tilde{t}) = \frac{1}{|W_t|} \sum_{j\in W_t} \left(\tilde{P}^{(1)}_t R^{\tilde{t}}_j + \mathbf{1}((\mathbf{t}_{\text{trans}})^{\tilde{t}}_j)^T \right) \tag{12}$$

Заметим, что здесь производится умножение не набора поз на матрицу поворота, как ранее, а лишь позы в одном кадре.

Так, с помощью прокрустова преобразования получили набор поз $(\hat{P}^{(1)}_1(\tilde{t}), \ldots, \hat{P}^{(1)}_{\tilde{T}-\tilde{t}+1}(\tilde{t}))$. Эти позы, в числе прочего, зависят от рассматриваемого смещения $\tilde{t}\ -\ 1$. Неформально, это набор нормализованных поз из видео 1 $(\tilde{P}^{(1)}_1, \ldots, \tilde{P}^{(1)}_{\tilde{T}-\tilde{t}+1})$, преобразованный с сохранением расстояний и углов и сдвинутый в пространстве так, чтобы наилучшим образом подходить под набор поз $(\tilde{P}^{(2)}_{\tilde{t}}, \ldots, \tilde{P}^{(2)}_{\tilde{T}})$ из видео 2.

3.7.2 Синхронизация

Метрика ошибки для данного смещения $\tilde{t}-1$ задаётся как среднее евклидова расстояния между позами из видео 2 и усреднёнными выровненными позами из видео 1:

$$\text{score}(\tilde{t}) = \frac{1}{(\tilde{T} - \tilde{t} + 1) \cdot 17} \sum_{t=1}^{\tilde{T}-\tilde{t}+1} \sum_{j=1}^{17} \left\| (\tilde{P}^{(2)}_{t+\tilde{t}-1})_j - (\hat{P}^{(1)}_t(\tilde{t}))_j \right\|_2, \tag{13}$$

где за $(\tilde{P}^{(2)}_t)_j$ и $(\hat{P}^{(1)}_t(\tilde{t}))_j$ обозначены строки матриц соответственно $(\tilde{P}^{(2)}_t)$ и $(\hat{P}^{(1)}_t(\tilde{t}))$ с номерами j. Т.е. метрика есть отклонение точек позы 1 относительно соответствующих им точек позы 2, усредненное по всем кадрам и по всем точкам.

На рис. 2 представлен результат вычисления метрики на одном из видео. Заметим,

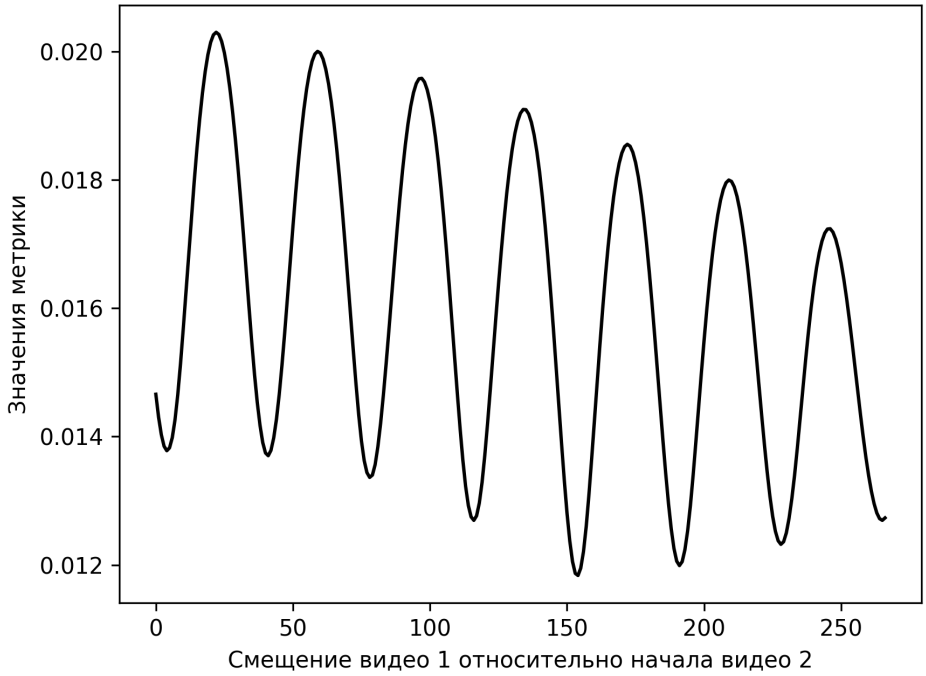

Рис. 2: Пример результата вычисления метрики

что в силу периодичности движения, значения метрики имеют осциллирующий характер; локальные минимумы соответствуют смещению примерно на длительность (в кадрах) одного полного цикла движения (один левый и один правый шаг), а максимумы, соответственно, смещению на половину длительности цикла. График построен на основе видео с частотой 30 кадров в секунду. Смещение выражено в кадрах. Выравнивание

произведено с помощью прокрустова преобразования с окном в 100 кадров. Минимум в районе смещения на 150 кадров отражает реальную задержку одного видео относительно другого.

Для оценки смещения видео 1 относительно видео 2 вычисляется:

$$t^* = \arg\min_{\tilde{t}} \text{score}(\tilde{t})$$

Заметим, что автоматическая оценка смещения не является полноценной заменой ручной синхронизации видео, так как может ошибаться, особенно при использовании данных с существенным шумом. Но даже при ручной синхронизации информация о локальных минимумах метрики может помочь определить точное смещение и снизить стоимость разметки данных, так как человек-разметчик уже выбирает из сокращенного набора возможных смещений.

Далее считаем смещение восстановленным.

3.7.3 Слияние

В качестве результата автоматической синхронизации видео имеем две последовательности поз:

$$(\hat{P}_1^{(1)}(t^*), \ldots, \hat{P}_{\tilde{T}-\tilde{t}+1}^{(1)}(t^*)) \quad \text{и} \quad (\tilde{P}_{\tilde{t}}^{(2)}, \ldots, \tilde{P}_{\tilde{T}}^{(2)})$$

Пусть, для простоты обозначений:

$$P_t^{(1)} := \hat{P}_t^{(1)}(t^*), \quad P_t^{(2)} := \hat{P}_{\tilde{t}+t}^{(2)}, \quad t = 1, \ldots, \tilde{T} - \tilde{t} + 1$$

$$l := \tilde{T} - \tilde{t} + 1$$

l – длина обоих видео после синхронизации и обрезки видео 2 по длине видео 1.

Требуется скомбинировать набор поз $(P_1^{(1)}, \ldots, P_l^{(1)})$ и $(P_1^{(2)}, \ldots, P_l^{(2)})$.

В случае множества камер, расположенных случайным образом, при неизвестной структуре движения, позы усредняются с весами $\frac{1}{m}$, где m – число камер. Усреднение – это универсальное решение, применимое для любой задачи подобного типа.

В случае используемого в рамках данной работы датасета структура движения и приблизительное расположение камер известны, следовательно нельзя исключать, что веса могут быть подобраны динамически так, чтобы уменьшить ошибку предсказания по

сравнению с одинаковыми константными весами. Известно, что камеры расположены по двум концам коридора. Человек на одном видео движется по направлению к выбранной камере, а на другом, соответственно – от этой камеры. Оба видео заканчиваются, когда человек находится вблизи противоположной камеры. Ошибка предсказания в среднем больше, когда человек находится дальше от камеры. Если ввести динамические веса на основе близости позы к камерам, то можно было бы ожидать уменьшение ошибки, вызванной ухудшением качества детекции и восприятия глубины на расстоянии. В то же время присвоение большего веса одной модели непременно уменьшит эффект устранения шума при объединении независимых наблюдений.

Эксперименты показали, что положительный эффект от ансамблирования превосходит негативный эффект от уменьшения качества предсказания на расстоянии (конкретно на датасете, использованном в этой работе), и веса $w_1(t) = w_2(t) = \frac{1}{2}$ дают наилучший результат.

Так, финальная оценка трехмерной позы человека в конфигурации с двумя камерами и в условиях описанной природы данных:

$$P_t = w_1(t)P_t^{(1)} + w_2(t)P_t^{(2)} \tag{14}$$

$$w_1(t) = w_2(t) = \frac{1}{2} = const,$$

и последовательность, описывающая позу,

$$P = (P_1, \ldots, P_l)$$

может быть использована для извлечения кинематических характеристик движения.

Таким образом, описанный подход, основанный на применении прокрустова преобразования для выравнивания поз между видеопоследовательностями, позволяет синхронизировать данные с нескольких камер и объединять их в единую трёхмерную оценку позы.

Описанный подход может быть обобщен на большее число камер.

4 Эксперименты

В данном разделе представлены метрики, позволяющие оценить качество метода оценки трехмерной позы в условиях отсутствия эталонных (ground truth) данных. Основная цель — численно проанализировать влияние различных элементов пайплайна на улучшение точности и стабильности предсказаний.

Для экспериментов использовались данные, описанные в разделе 3.2. Количество видео в датасете – 11 штук, из которых 2 пары видео записаны в конфигурации с двумя камерами.

В рамках данного раздела нумерация видео везде совпадает с той, что приведена в таблице 2: одно и то же видео далее сохраняет свой номер.

4.1 Оценка 2D детекции

Сравним уверенность 2D детекции для всех видео. Выберем лишь точки, которые далее используются при переводе из формата двумерной позы Halpe26 в формат Human3.6M. По каждому видео приведем данные как для всего видео, так и для той половины видео, где человек находится ближе к камере. Результаты приведены в таблице 2 и отражают уверенность в детекции именно пациента (главная траектория уже выделена).

RTMPose демонстрирует высокую уверенность (>0.7) в подавляющем большинстве кадров. При этом наблюдается вариативность качества между видео, обусловленная ракурсом съёмки: так, пары видео 2–3 и 4–5, записанные в конфигурации с двумя камерами, показывают различия в уровне уверенности внутри пары. Визуальный анализ результатов показал отсутствие систематических ошибок, таких как ошибки идентификации левой и правой сторон тела и слияние двух ног в одну в рамках позы.

При более строгих порогах ($\text{conf} > 0.8$ и выше) доля кадров с соответствующей уверенностью снижается, однако для области «ближе к камере» она в большинстве случаев составляет более 50%. В целом, уверенность выше, когда человек ближе к камере. Это, в частности, оправдывает применение описанного в разделе 3.7 метода слияния, при котором в мультикамерной конфигурации предпочтение в любой момент времени отдается предсказанию на основе тех кадров, где человек ближе к камере (чем точнее 2D детекция, тем, при прочих равных, точнее результаты методов на дальнейших этапах).

Номер видео	Область видео	Всего кадров	% кадров с conf > 0.7	% кадров с conf > 0.75	% кадров с conf > 0.8	% кадров с conf > 0.85
1	Все кадры	2245	100.00%	92.78%	28.33%	0.09%
1	Ближе к камере	1122	100.00%	99.64%	56.24%	0.18%
2	Все кадры	520	91.54%	51.15%	33.08%	0.00%
2	Ближе к камере	260	100.00%	94.62%	66.15%	0.00%
3	Все кадры	586	100.00%	96.08%	64.85%	31.06%
3	Ближе к камере	293	100.00%	100.00%	99.66%	60.75%
4	Все кадры	313	100.00%	93.93%	40.26%	0.00%
4	Ближе к камере	156	100.00%	100.00%	57.69%	0.00%
5	Все кадры	337	100.00%	100.00%	51.63%	11.57%
5	Ближе к камере	169	100.00%	100.00%	71.60%	22.49%
6	Все кадры	420	95.95%	65.71%	25.95%	0.24%
6	Ближе к камере	210	93.81%	88.10%	51.90%	0.48%
7	Все кадры	315	100.00%	95.87%	52.38%	3.17%
7	Ближе к камере	158	100.00%	98.10%	78.48%	6.33%
8	Все кадры	342	100.00%	86.55%	38.89%	0.88%
8	Ближе к камере	171	100.00%	100.00%	71.93%	1.75%
9	Все кадры	299	100.00%	97.66%	48.49%	0.67%
9	Ближе к камере	150	100.00%	97.33%	88.00%	1.33%
10	Все кадры	382	100.00%	90.84%	19.63%	0.00%
10	Ближе к камере	191	100.00%	98.95%	31.94%	0.00%
11	Все кадры	422	100.00%	90.05%	63.51%	3.32%
11	Ближе к камере	211	100.00%	100.00%	97.16%	6.16%

Таблица 2: Уверенность в 2D детекции

Примечание: Значение conf (уверенность) — среднее значение уверенности в детекции по всем выбранным ключевым точкам в кадре. Процент означает долю кадров (в %), для которых средний conf превышает указанный порог.

4.2 Оценка трекинга и выбора траектории

Для каждого из 11 видео вычислена метрика важности траектории (раздел 3.4) для всех обнаруженных траекторий. Результаты приведены на рис. 3.

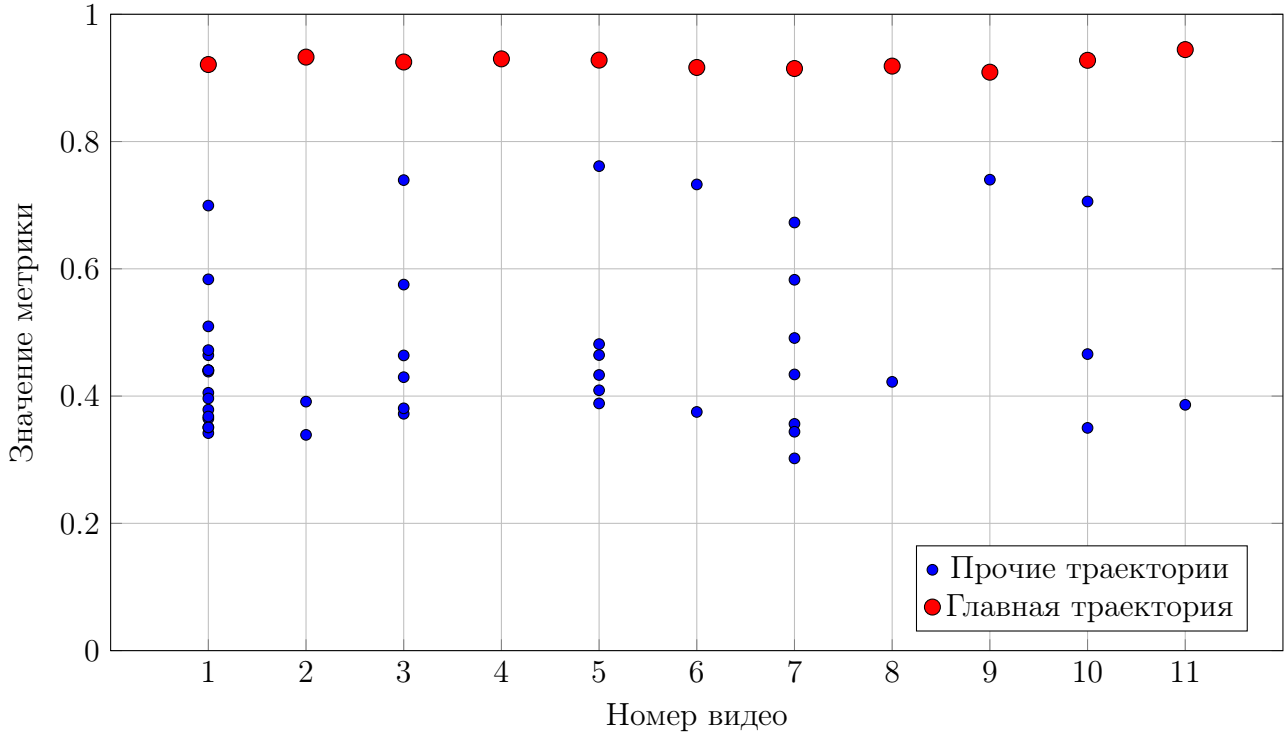

Рис. 3: Метрика важности траектории

Главная траектория была выделена верно автоматически во всех случаях.

Для оценки работы выбранного метода трекинга на датасете было проведено сравнение количества кадров, на которых пациент был обнаружен, с общим их числом. На всех 11 видео пациент был обнаружен на всех кадрах без исключения.

4.3 Оценка лифтинга, слияния и синхронизации. Примеры

В этом подразделе оценим влияние ансамблирования на предсказание позы. Для этого возьмем комбинацию предсказаний с различными весами (7):

1) $w_1 = 1, w_2 = w_3 = 0$ (без ансамблирования)

2) $w_1 = w_2 = w_3 = \frac{1}{3}$ (одинаковые веса)

3) с весами как в утв. из раздела 3.5 (динамически изменяющиеся веса)

Введем метрики:

$$\mathrm{RMSV} = \sqrt{\frac{1}{17(T-1)} \sum_{t=2}^{T} \sum_{i=1}^{17} \left\| (\tilde{P}_t)_i - (\tilde{P}_{t-1})_i \right\|^2} \tag{15}$$

где T – число кадров, 17 – число ключевых точек,

$\tilde{P}_t \in \mathbb{R}^{17\times 3}$ – трехмерная поза в кадре t после нормализации (9) с однородными весами (т.е. $\tilde{w} = (\frac{1}{n}, \ldots, \frac{1}{n})$),

$(\tilde{P}_t)_i \in \mathbb{R}^3$ – i-я строка матрицы $\tilde{P}_t$, т.е. трехмерные координаты точки i в кадре t,

$\|\cdot\|$ — евклидова норма.

Далее, работая в обозначениях раздела 3.6, введем метрику AV (average variance) наподобие функционала $J(\tilde{w})$ (11) в точке $(\frac{1}{n}, \ldots, \frac{1}{n})$. Возьмем вектор основных расстояний точно так же, как в (8), за тем лишь исключением, что вместо позы P_t теперь используем $\tilde{P}_t$. Так, $d(t; i, j)$ – это уже нормализованные расстояния, т.к. взяты относительно позы $\tilde{P}_t$ вместо P_t.

Введем среднюю дисперсию AV на основе формул (10), (11):

$$\mathrm{AV} = \sum_{k=1}^{n} \mathrm{Var}_t(d(t; i_k, j_k)) \tag{16}$$

Метрика RMSV (15) – среднее квадратическое евклидовых расстояний между одними и теми же точками в соседних кадрах. Эта метрика покажет, насколько сильно изменяется позиция точки между кадрами. Бóльшие значения будут означать меньшую плавность.

Метрика AV (16) – средняя дисперсия длин некоторых расстояний между точками. Т.к. длины костей не изменяются с течением времени, то ожидается, что чем точнее предсказание, тем меньше значения AV.

Результаты вычисления метрики при отсутствии ансамблирования, одинаковых и динамически изменяющихся весах представлены в таблице 3.

Эксперименты показали, что метрика RMSV на всех видео максимальна при использовании динамически изменяющихся весов. Худшие результаты наблюдаются при полном отсутствии ансамблирования.

Номер видео	RMSV (без анс.)	RMSV (одинак. веса)	RMSV (динамич.)	AV (без анс.)	AV (одинак. веса)	AV (динамич.)
1	0.01478	0.01163	**0.01017**	0.02438	**0.02110**	0.02164
2	0.03222	0.02878	**0.02737**	0.01888	0.01319	**0.01237**
3	0.04245	0.03685	**0.03567**	0.02218	**0.01474**	0.01497
4	0.07002	0.04108	**0.03587**	0.07882	0.03138	**0.02362**
5	0.04073	0.03504	**0.03070**	0.02694	0.02014	**0.01944**
6	0.02830	0.02531	**0.02409**	0.02705	**0.02144**	0.02153
7	0.03897	0.04506	**0.02356**	**0.12399**	0.16994	0.15756
8	0.09007	0.04218	**0.02647**	0.19841	0.07011	**0.06436**
9	0.03336	0.03745	**0.02990**	0.03976	0.04183	**0.03876**
10	0.07009	0.04154	**0.03449**	0.08065	0.02886	**0.02250**
11	0.06221	0.04055	**0.03571**	0.07940	**0.04367**	0.04587

Таблица 3: Плавность предсказаний при ансамблировании

Значения метрики AV менее последовательны. Можно заметить, однако, что на большинстве видео есть существенный разрыв между значениями AV при отсутствии ансамблирования и значениями AV при двух других вариантах. При этом разрыв между двумя другими вариантами – при использовании предсказаний с одинаковыми весами и с динамически изменяющимися весами – уже меньше, а при использовании динамически изменяющихся весов минимум метрики принимается в наибольшем числе случаев (в сравнении с остальными вариантами).

Данные из таблицы 3, а также визуальный анализ показывают, что наибольшая плавность и временная связность предсказаний достигаются в случае весов, зависящих от времени.

Теперь приведем данные, позволяющие оценить эффект слияния в мультикамерной конфигурации. Пары видео, записанные в мультикамерной конфигурации: 2-3 и 4-5. Обозначим предсказания, полученные слиянием видео 2 и 3, как «2+3», а 4 и 5 – «4+5».

Пусть известно, что человек на видео 1 движется по направлению к камере, а на видео 2 – от камеры. Используем эвристический прием: введем веса так, чтобы слияние было гладким, и одновременно больший вес был бы у позы, соответствующей видео с той камеры,

к которой эта поза ближе. Природа данных позволяет сказать, что изначально поза ближе к камере 2, а далее, начиная с момента вблизи середины видео, поза становится ближе к камере 1.

Введем веса на основе сигмоиды:

$$w_1(t) = \frac{1}{1 + \exp\big(-k(t - t_0)\big)}$$

$$w_2(t) = 1 - \frac{1}{1 + \exp\big(-k(t - t_0)\big)}$$

$$t = 1, \ldots, l$$

t_0 берется равным $\lfloor l/2 \rfloor$. Константа k определяет скорость изменения веса со временем, в частности, при преодолении отметки в $\lfloor l/2 \rfloor$ кадров. Результирующая поза получается подстановкой этих весов в (14).

Сравним слияние с динамическими ($k > 0$) и константными ($k = 0$) весами, а также оценим влияние на RMSV и AV различных размеров окна в рамках прокрустова преобразования (см. раздел 3.7.1).

Видео	Всего кадров	RMSV (динамич.)	AV (динамич.)
2	520	0.02737	0.01237
3	586	0.03567	0.01497
4	313	0.03587	0.02362
5	337	0.03070	0.01944

Таблица 4: Данные для сравнения

Видео	Кадры	k	RMSV	AV
2+3	420	0.05	0.03615	0.02327
4+5	237	0.05	0.03819	0.02664

Видео	Кадры	k	RMSV	AV
2+3	420	0.02	0.03489	0.01784
4+5	237	0.02	0.03385	0.01719

Видео	Кадры	k	RMSV	AV
2+3	420	0.01	0.03293	0.01157
4+5	237	0.01	0.03097	0.01198

Видео	Кадры	k	RMSV	AV
2+3	420	0.00	0.02890	0.01395
4+5	237	0.00	0.02838	0.00889

Таблица 5: Плавность предсказаний при слиянии для различных k

Видео	Размер окна	RMSV	AV
2+3	1	0.03183	0.00535
4+5	1	0.03089	0.00889

Видео	Размер окна	RMSV	AV
2+3	10	0.03100	0.00534
4+5	10	0.02958	0.00894

Видео	Размер окна	RMSV	AV
2+3	40	0.02936	0.00536
4+5	40	0.02847	0.00896

Видео	Размер окна	RMSV	AV
2+3	200	0.02936	0.00531
4+5	200	0.02839	0.00878

Видео	Размер окна	RMSV	AV
2+3	100	0.02936	0.00535
4+5	100	0.02838	0.00889

Таблица 6: Плавность предсказаний при слиянии для различных размеров окна в прокрустовом преобразовании ($k = 0$)

Данные в таблице 5 представлены для размера окна 100 в прокрустовом преобразовании.

Из таблицы 5 видно, что при резком переключении предсказаний с тех, что получены по видео 1, на те, что получены по видео 2 (т.е. при относительно больших k), результаты метрик наихудшие. Метрика RMSV, описывающая плавность движения, принимает наименьшие значения при $k = 0$, т.е. при весах $w_1 = w_2 = \frac{1}{2}$, на обеих парах видео. Метрика AV разброса длин расстояний между точками принимает наименьшие значения при $k = 0$ и $k = 0.01$ в зависимости от пары. Эти результаты косвенно показывают, что ансамблирование дает преимущество даже несмотря на то, что результаты базовых моделей по отдельности хуже, чем предсказания, полученные слиянием с динамическими весами. Ожидается, что при большем числе камер в конфигурации метрики можно улучшить еще больше.

Таблица 6 показывает, что значения AV практически неизменны в зависимости от ширины окна, использующегося в рамках прокрустова преобразования. Одновременно с этим, при увеличении ширины окна (начиная с единицы) значения RMSV уменьшаются

до тех пор, пока ширина окна не будет сравнима с длиной шага (в кадрах). После этого RMSV практически не изменяется (окна ширины 40, 100, 200).

При совпадении метрик разумно взять как можно меньшую ширину окна в силу формулы (12). В датасете, использованном в данной работе, пациент движется в конкретном направлении, но если бы человек в кадре часто менял направление, это бы привело к ухудшению качества преобразованной позы, т.к. усреднение может приводить к искажениям (утрате жесткости скелета).

Примечание: в таблицах 5, 6 использованы предсказания поз в мультикамерной конфигурации, для которых синхронизация выполнена корректно.

Важной частью исследования являются анализ надежности предлагаемого алгоритма синхронизации и анализ возможности использования данных о движении для этого. Заметим, что не во всех случаях предложенный алгоритм синхронизации может в автоматическом режиме верно выбрать смещение. Это связано с периодичностью движения и шумом, всегда наличествующим в данных. Этот алгоритм – компромиссный вариант. Он сужает множество возможных смещений примерно в 30 раз (см. рис. 2: выбирается один из экстремумов), но требует дополнительного анализа результатов вручную.

Приведем результаты работы алгоритма синхронизации при использовании различных фрагментов видео в рамках синхронизации и окон различной ширины для прокрустова преобразования при выравнивании поз.

Видео	Размер окна	Верно ли определено смещение?
2+3	1	Нет
4+5	1	Да

Видео	Размер окна	Верно ли определено смещение?
2+3	20	Нет
4+5	20	Да

Видео	Размер окна	Верно ли определено смещение?
2+3	100	Да
4+5	100	Да

Видео	Размер окна	Верно ли определено смещение?
2+3	200	Да
4+5	200	Да

Таблица 7: Работа алгоритма синхронизации. Первое видео пары урезано на 50 кадров с начала и конца

Данные таблиц 7, 8 показывают ненадежность синхронизации без дополнительного выбора из экстремумов вручную, т.к. ошибки на обоих парах возникают независимо от размера окна и количества вырезанных кадров.

Видео	Размер окна	Верно ли определено смещение?
2+3	1	Да
4+5	1	Нет

Видео	Размер окна	Верно ли определено смещение?
2+3	20	Да
4+5	20	Нет

Видео	Размер окна	Верно ли определено смещение?
2+3	100	Да
4+5	100	Нет

Видео	Размер окна	Верно ли определено смещение?
2+3	200	Да
4+5	200	Нет

Таблица 8: Работа алгоритма синхронизации. Первое видео пары урезано на 100 кадров с начала и конца

В рамках экспериментального исследования в числе прочего были выполнены: визуальный анализ графиков, показывающих работу алгоритма синхронизации; попытка коррекции на скользящую среднюю для устранения систематического увеличения и уменьшения ошибок с течением времени из-за удаления (приближения) субъекта от камеры (к камере); анализ результатов при иных размерах окна; а также применение среднего квадратического вместо среднего арифметического в (13). Это также показало наличие ошибок при синхронизации.

Вывод: наличие дополнительных механизмов синхронизации необходимо, т.к. ошибки предложенного алгоритма являются систематическими и отражают тот факт, что при имеющихся природе данных (движение периодично) и шумах, вносимых в предсказание в процессе оценки позы, данные об одном лишь движении не могут быть надежно использованы для согласования камер во времени.

Хотя предложенный алгоритм призван сократить множество возможных смещений, его надежность недостаточна для полностью автоматизированного применения. Наиболее надежным решением остается синхронизация на этапе записи (временные метки) или применение иных механизмов синхронизации на аппаратном или программном уровнях.

Визуализированные примеры восстановленных поз

Ниже представлены иллюстрации результатов работы пайплайна, изображающие полностью восстановленную трехмерную позу.

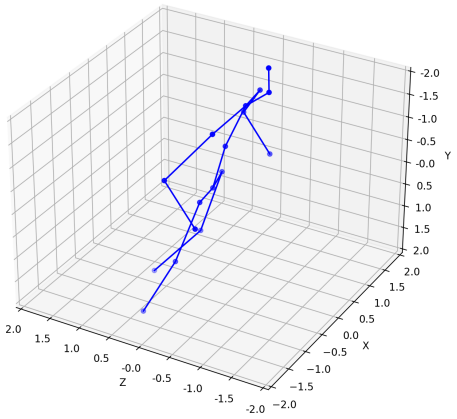

Рис. 4: Восстановленная поза, соответствующая кадру №350 из видео 2

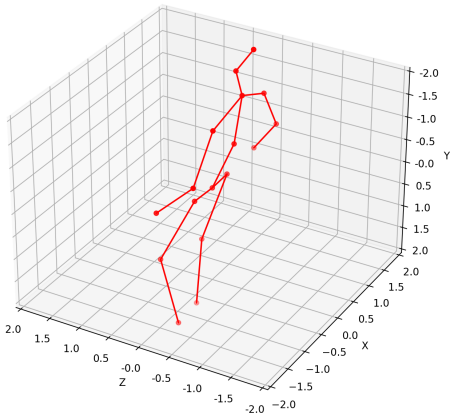

Рис. 5: Восстановленная поза, соответствующая кадру №404 из видео 3

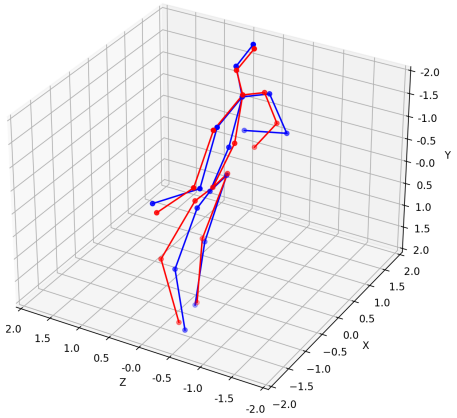

Рис. 6: Наложение поз с помощью прокрустова преобразования

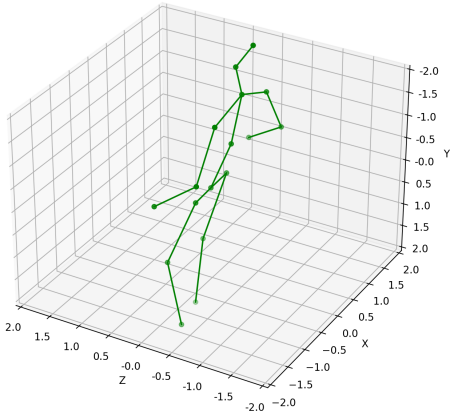

Рис. 7: Результирующая уточненная поза

Заключение

В данной работе был реализован пайплайн для восстановления 3D позы человека из видео с использованием одной и нескольких камер. За основу метода восстановления трехмерной позы взяты SOTA-модели. Также были предложены методы сочетания различных элементов пайплайна таким образом, чтобы сделать итоговое предсказание пригодным для дальнейшей обработки.

Проведённые эксперименты подтвердили корректность работы отдельных компонентов пайплайна. В условиях отсутствия эталонных (ground truth) данных, оценка итоговой 3D позы проводилась визуально и показала плавность и правдоподобие реконструкции движений человека.

Исходный код проекта доступен в GitHub-репозитории[1]

[1] `https://github.com/oscar-foxtrot/PoseBrew` (дата обращения: 19.06.2025)

A Приложение

A.1 Производная

$$\text{Для } t \in \{1, \ldots, 243\} \quad G(t) = \exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)$$

$$\text{Для } t \in \{1, \ldots, 81\} \quad G(t-81) = \exp\left(-\frac{1}{2}\left(\frac{(t+162-1)-121}{40}\right)^2\right)$$

$$\text{Для } t \in \{82, \ldots, 243\} \quad G(t-81) = \exp\left(-\frac{1}{2}\left(\frac{(t-81-1)-121}{40}\right)^2\right)$$

$$\text{Для } t \in \{1, \ldots, 162\} \quad G(t-162) = \exp\left(-\frac{1}{2}\left(\frac{(t+81-1)-121}{40}\right)^2\right)$$

$$\text{Для } t \in \{163, \ldots, 243\} \quad G(t-162) = \exp\left(-\frac{1}{2}\left(\frac{(t-162-1)-121}{40}\right)^2\right)$$

Получим производную для $w_1(t)$ на $t \in \{82, \ldots, 162\}$, рассматривая данную функцию как заданную на вещественных числах $t \in [82, 162]$:

$$w_1(t) = \frac{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)}{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right) + \exp\left(-\frac{1}{2}\left(\frac{(t-1-81)-121}{40}\right)^2\right) + \exp\left(-\frac{1}{2}\left(\frac{(t-1+81)-121}{40}\right)^2\right)}$$

$$w_1'(t) = \frac{\left(\frac{61}{800} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}}{e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}} +$$

$$+\frac{\left(-\left(\frac{41}{1600} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} - \left(\frac{61}{800} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} - \left(\frac{203}{1600} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}\right) e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}}{\left(e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}\right)^2} =$$

$$= e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} \cdot \left(\frac{\frac{61}{800} - \frac{x}{1600}}{e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}} + \right.$$

$$\left. + \frac{-\left(\frac{41}{1600} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} - \left(\frac{61}{800} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} - \left(\frac{203}{1600} - \frac{x}{1600}\right) e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}}{\left(e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}} + e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}\right)^2}\right)$$

Домножая на $\left(e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}}+e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}+e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}\right)^2$,

а также сокращая на $e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}$ получаем:

$$-\left(\frac{61}{800}-\frac{x}{1600}\right)e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}-\left(\frac{203}{1600}-\frac{x}{1600}\right)e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}+$$
$$+\left(\frac{x}{1600}-\frac{41}{1600}\right)e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}}+\left(\frac{61}{800}-\frac{x}{1600}\right)\left(e^{-\frac{\left(\frac{x}{40}-\frac{41}{40}\right)^2}{2}}+e^{-\frac{\left(\frac{x}{40}-\frac{61}{20}\right)^2}{2}}+e^{-\frac{\left(\frac{x}{40}-\frac{203}{40}\right)^2}{2}}\right)$$

Слагаемые с x сокращаются. Остается:

$$\frac{81\left(-e^{\frac{81x}{800}}+e^{\frac{4941}{400}}\right)e^{-\frac{x^2}{3200}+\frac{41x}{1600}-\frac{41209}{3200}}}{1600} \tag{1A}$$

Это выражение равно нулю $\iff \left(-e^{\frac{81x}{800}}+e^{\frac{4941}{400}}\right)=0 \iff \frac{81x}{800}=\frac{4941}{400} \iff$
$81x=4941\cdot 2 \iff x=122$.

Заметим, что при получении выражения (1A) были использованы операции, не изменяющие нулей производной и ее знака. Отсюда, если $x>122$, тогда $\frac{81x}{800}>\frac{4941}{400} \implies$ выражение (1A) <0. Аналогично, при $x<122$ выражение >0.

Для $t\in\{1,\ldots,81\}$ (w_1 рассматривается как заданная на вещественных числах):

$$w_1(t)=\frac{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)}{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)+\exp\left(-\frac{1}{2}\left(\frac{(t-1+162)-121}{40}\right)^2\right)+\exp\left(-\frac{1}{2}\left(\frac{(t-1+81)-121}{40}\right)^2\right)}$$

$$w_1'(t)=\frac{81\left(e^{\frac{81x}{1600}+\frac{1}{2}}+2e^{\frac{1681}{3200}}\right)e^{-\frac{x^2}{3200}-\frac{x}{40}-\frac{3281}{3200}}}{1600}$$

Это выражение всегда положительно.

Для $t\in\{163,\ldots,243\}$

$$w_1(t)=\frac{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)}{\exp\left(-\frac{1}{2}\left(\frac{(t-1)-121}{40}\right)^2\right)+\exp\left(-\frac{1}{2}\left(\frac{(t-1-81)-121}{40}\right)^2\right)+\exp\left(-\frac{1}{2}\left(\frac{(t-1-162)-121}{40}\right)^2\right)}$$

Выполняя шаги, аналогичные предыдущим, получаем:

$$w_1'(t)=\frac{81\left(-e^{\frac{203x}{1600}+\frac{5041}{200}}-2e^{\frac{71x}{400}+\frac{41209}{3200}}\right)e^{-\frac{x^2}{3200}-\frac{24373}{640}}}{1600}$$

Это выражение всегда отрицательно.

Заметим также, что $w_1(81)<w_1(82)$ и $w_1(162)>w_1(163)$

A.2 Формат Human3.6M

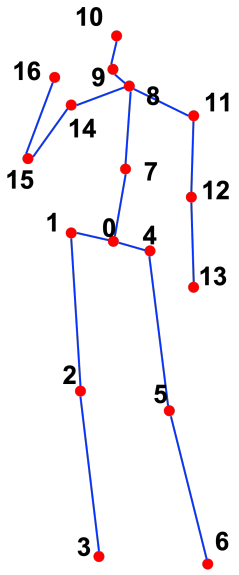

Рис. 8: Схема трехмерной позы в формате Human3.6M

На данном рисунке приведено схематичное изображение трехмерной позы в формате Human3.6M. Под множеством D «основных расстояний» подразумеваются пары точек, между которыми на данной схеме проведены отрезки.

Конкретно:

$$D = \{(3,2),(2,1),(1,0),(0,4),(4,5),(5,6),(13,12),(12,11),$$
$$(11,8),(8,14),(14,15),(15,16),(8,9),(9,10),(8,7),(7,0)\}$$

Список литературы

1. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis / M. Andriluka [et al.] // 2014 IEEE Conference on Computer Vision and Pattern Recognition. — 2014. — P. 3686–3693.
2. 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training / D. Pavllo [et al.] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2019. — P. 7745–7754.
3. *Aharon N.*, *Orfaig R.*, *Bobrovsky B.-Z.* BoT-SORT: Robust Associations Multi-Pedestrian Tracking. — 2022. — arXiv: `2206.14651 [cs.CV]`.
4. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time / H.-S. Fang [et al.] // IEEE Transactions on Pattern Analysis & Machine Intelligence. — Los Alamitos, CA, USA, 2023. — Vol. 45, no. 06. — P. 7157–7173.
5. Anatomy-Aware 3D Human Pose Estimation With Bone-Based Pose Decomposition / T. Chen [et al.] // IEEE Transactions on Circuits and Systems for Video Technology. — 2022. — Vol. 32, no. 1. — P. 198–209.
6. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [et al.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) / ed. by J. Burstein, C. Doran, T. Solorio. — Minneapolis, Minnesota : Association for Computational Linguistics, 2019. — P. 4171–4186.
7. Deep High-Resolution Representation Learning for Visual Recognition / J. Wang [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2021. — Vol. 43, no. 10. — P. 3349–3364.
8. Diagnosis of disease affecting gait with a body acceleration-based model using reflected marker data for training and a wearable accelerometer for implementation / M. Takallou, F. Fallahtafti, M. Hassan, [et al.] // Scientific Reports. — 2024. — Vol. 14. — P. 1075.
9. Exploiting Temporal Contexts With Strided Transformer for 3D Human Pose Estimation / W. Li [et al.] // IEEE Transactions on Multimedia. — 2023. — Vol. 25. — P. 1282–1293.

10. Gait-based Parkinson's disease diagnosis and severity classification using force sensors and machine learning / M. P. Navita, Y. Sharma, [et al.] // Scientific Reports. — 2025. — Vol. 15. — P. 328.

11. *Gower J. C.* Generalized Procrustes analysis // Psychometrika. — 1975. — Vol. 40, no. 1. — P. 33–51.

12. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments / C. Ionescu [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2014. — Vol. 36, no. 7. — P. 1325–1339.

13. *Kocabas M.*, *Athanasiou N.*, *Black M. J.* VIBE: Video Inference for Human Body Pose and Shape Estimation // 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2020. — P. 5252–5262.

14. *Kraft D.* A software package for sequential quadratic programming : Tech. Rep. / DLR German Aerospace Center – Institute for Flight Mechanics. — Cologne, Germany, 1988. — DFVLR-FB 88–28.

15. Large-Scale Datasets for Going Deeper in Image Understanding / J. Wu [et al.] // 2019 IEEE International Conference on Multimedia and Expo (ICME). — IEEE, 2019.

16. *Lin K.*, *Wang L.*, *Liu Z.* End-to-End Human Pose and Mesh Reconstruction with Transformers // 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2021. — P. 1954–1963.

17. Microsoft COCO: Common Objects in Context / T.-Y. Lin [et al.] // Computer Vision – ECCV 2014 / ed. by D. Fleet [et al.]. — Cham : Springer International Publishing, 2014. — P. 740–755.

18. MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking / P. Dendorfer [et al.] // International Journal of Computer Vision. — 2021. — Vol. 129, no. 4. — P. 845–881.

19. MotionBERT: A Unified Perspective on Learning Human Motion Representations / W. Zhu [et al.] // 2023 IEEE/CVF International Conference on Computer Vision (ICCV). — 2023. — P. 15039–15053.

20. *Myronenko A.*, *Song X.* Point Set Registration: Coherent Point Drift // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2010. — Vol. 32, no. 12. — P. 2262–2275.

21. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking / J. Cao [et al.] // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2023. — P. 9686–9696.

22. Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera / T. von Marcard [et al.] // Computer Vision – ECCV 2018 / ed. by V. Ferrari [et al.]. — Cham : Springer International Publishing, 2018. — P. 614–631.

23. RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose / T. Jiang [et al.]. — 2023. — arXiv: 2303.07399 [cs.CV].

24. *Schönemann P. H.* A Generalized Solution of the Orthogonal Procrustes Problem // Psychometrika. — 1966. — Vol. 31, no. 1. — P. 1–10.

25. *Schönemann P. H.*, *Carroll R. M.* Fitting one matrix to another under choice of a central dilation and a rigid motion // Psychometrika. — 1970. — Vol. 35, no. 2. — P. 245–255.

26. *Stanojević V.*, *Todorović B.* BoostTrack++: using tracklet information to detect more objects in multiple object tracking. — 2024. — arXiv: 2408.13003 [cs.CV].

27. *Umeyama S.* Least-squares estimation of transformation parameters between two point patterns // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1991. — Vol. 13, no. 4. — P. 376–380.

28. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation / Y. Xu [et al.] // Advances in Neural Information Processing Systems. Vol. 35 / ed. by S. Koyejo [et al.]. — Curran Associates, Inc., 2022. — P. 38571–38584.

29. WHAM: Reconstructing World-Grounded Humans with Accurate 3D Motion / S. Shin [et al.] // 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2024. — P. 2070–2080.

30. Whole-Body Human Pose Estimation in the Wild / S. Jin [et al.] // Computer Vision – ECCV 2020 / ed. by A. Vedaldi [et al.]. — Cham : Springer International Publishing, 2020. — P. 196–214.

31. *Broström M.* BoxMOT: pluggable SOTA tracking modules for object detection, segmentation and pose estimation models [Электронный ресурс]. — 2023. — URL: `https://github.com/mikel-brostrom/boxmot` (дата обращения: 19.06.2025).

32. *MMPose Contributors.* OpenMMLab Pose Estimation Toolbox and Benchmark [Электронный ресурс]. — 2020. — URL: `https://github.com/open-mmlab/mmpose` (дата обращения: 19.06.2025).