УДК 004.932:004.42

КОНВЕЙЕР ОБРАБОТКИ ПОЗЫ И НОРМАЛИЗАЦИЯ КООРДИНАТ В AR-ПРИЛОЖЕНИИ ДЛЯ ВИРТУАЛЬНОЙ ПРИМЕРКИ ОДЕЖДЫ

Заяц А.А.

Национальный детский технопарк, г. Минск, Республика Беларусь

Научные руководители: Прудник А.М. – к. т. н, доцент кафедры ИПиЭ;
Ильясова М.С. – магистр техн. наук, ассистент кафедры ИПиЭ

Аннотация. Рассмотрена архитектура конвейера обработки видеокадров в Android-приложении для виртуальной примерки одежды. Описаны методы нормализации координат ключевых точек тела, обеспечивающие корректное отображение 3D-моделей на устройствах с различным разрешением экрана. Приведены результаты тестирования на нескольких моделях смартфонов.

Ключевые слова: дополненная реальность, распознавание позы, нормализация координат, скелетная анимация, CameraX, ML Kit, Kotlin, Android.

Введение. Технологии дополненной реальности (AR) постепенно занимают своё место в розничной торговле, позволяя покупателю оценить товар до покупки [1]. Главная техническая сложность при переносе AR на мобильные устройства – огромный разброс экранных разрешений и пропорций. Нейросеть выдаёт координаты ключевых точек тела в пикселях конкретного кадра, и если не привести их к единой системе, виртуальная одежда на одном телефоне будет позиционироваться корректно, а на другом – сместится относительно фигуры пользователя.

Существующие сервисы, например IDM VTON [2], обрабатывают изображение на удалённом сервере, что порождает заметную задержку и не поддерживают взаимодействие в реальном времени. В настоящей статье описывается локальный конвейер обработки видеопотока, реализованный в приложении виртуальной примерочной для магазина с атрибутикой Минского автомобильного завода (МАЗ) [3]. Особое внимание уделено алгоритму нормализации двумерных координат и их проецированию на скелет 3D-модели одежды.

Основная часть. В качестве среды разработки использовалась Android Studio, язык программирования – Kotlin. Альтернативный вариант на базе движка Unity был отвергнут из-за дополнительных накладных расходов при обращении к камере. Графический интерфейс реализован средствами декларативного фреймворка Jetpack Compose [4], который заменил классическую XML-разметку экранов и упростил управление состояниями. Внешний вид главного экрана приложения показан на рисунке 1.

Рисунок 1 – Пользовательский интерфейс приложения

Обработка каждого кадра проходит три последовательных этапа. На первом этапе класс CameraXViewModel формирует два объекта UseCase из библиотеки CameraX [5]: Preview для предварительного просмотра и ImageAnalysis для анализа кадров. ImageAnalysis использует стратегию KEEP_ONLY_LATEST, а выбор разрешения делегирован ResolutionSelector. Метод bindToCamera связывает оба UseCase с жизненным циклом Activity.

На втором этапе кадр поступает в класс AndroidPoseDetector – обёртку над библиотекой Google ML Kit Pose Detection [6]. В зависимости от источника изображения детектор работает в режиме STREAM_MODE (камера) или SINGLE_IMAGE_MODE (фотография из галереи). Метод processImage преобразует входной PlatformImage в формат InputImage с учётом поворота сенсора и запускает асинхронный анализ. Результатом является массив ключевых точек тела – плечи, локти, бёдра – с координатами в пикселях исходного кадра.

Третий этап – нормализация координат. Пиксельные значения напрямую зависят от физического разрешения камеры, и без пересчёта позиция 3D-модели оказывается некорректной на устройствах с иным соотношением сторон. В функции convertToBones рассчитывается вектор масштабирования maxValue: его компоненты x и y равны отношению фиксированных опорных констант к ширине и высоте кадра соответственно. Далее координаты каждой точки покомпонентно умножаются на maxValue, а компонента z обнуляется, поскольку наложение одежды происходит в плоскости камеры.

Для определения положения виртуальной футболки вспомогательная функция average вычисляет среднее двух трёхмерных точек, что позволяет получить центр плечевого пояса и центр бёдер. Угол наклона корпуса находится функцией calculateAngle: она принимает две точки, рассчитывает арктангенс отношения разности их y-координат к разности x-координат, добавляет смещение 90° и переводит результат в радианы. Полученные значения Position и Rotation передаются движку SceneView [7], который в режиме реального времени деформирует скелет 3D-модели футболки (рисунок 2).

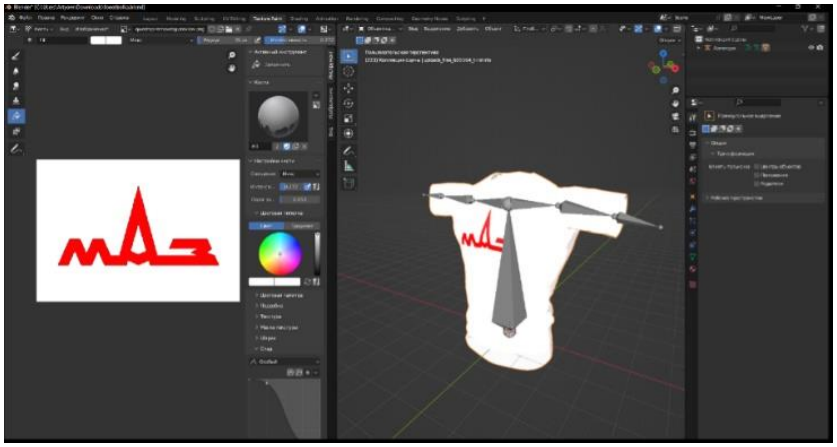

Рисунок 2 – 3D-модель футболки с атрибутикой МАЗ и встроенный скелет в среде Blender

Трёхмерные модели футболок подготовлены в редакторе Blender, выбранном благодаря открытому исходному коду и удобным средствам настройки скелетной анимации. Каждая модель содержит иерархию костей: корневая кость устанавливается в центр торса, а дочерние кости отвечают за плечи и нижнюю часть. При поступлении новых данных от нейросети корневая кость перемещается, а дочерние поворачиваются на вычисленные углы, что обеспечивает визуально плавную деформацию ткани.

Тестирование проводилось на трёх устройствах с разрешениями экрана 1080×2400, 1080×2340 и 720×1600 пикселей. Без нормализации горизонтальное отклонение модели от ожидаемого положения достигало 18 %, вертикальное – 12 %. Наибольшая погрешность наблюдалась на устройстве с разрешением 720×1600, имеющем нестандартное соотношение сторон 20:9. После внедрения описанного коэффициентного пересчёта максимальное отклонение не превысило 2 % ни на одном из устройств. Среднее время обработки одного кадра составило около 35 мс, что соответствует частоте обновления приблизительно 28 кадров

в секунду – этого достаточно для визуально плавного наложения одежды при типичных движениях пользователя перед камерой.

Дополнительно было отмечено, что точность распознавания ключевых точек заметно зависит от условий освещённости и расстояния пользователя до камеры. При слабом освещении детектор ML Kit периодически терял отдельные точки на один-два кадра, из-за чего 3D-модель кратковременно дёргалась. Для смягчения этого эффекта в конвейер был добавлен промежуточный буфер, который сохраняет координаты последнего успешного распознавания и подставляет их вместо отсутствующих данных. Оптимальная дистанция между пользователем и устройством в ходе экспериментов составила от 50 до 120 сантиметров: при меньшем расстоянии фигура выходила за границы кадра, при большем – падала детализация точек и ухудшалась точность наложения. Данное ограничение было учтено в интерфейсе приложения: пользователю выводится текстовая подсказка с просьбой отойти или приблизиться к камере. Кроме того, на устройствах с фронтальной камерой разрешением ниже 5 мегапикселей время обработки возрастало до 42 мс на кадр, что тем не менее оставалось в допустимых пределах для комфортного использования.

Заключение. В работе представлен трёхэтапный конвейер обработки видеопотока для мобильного AR-приложения виртуальной примерки одежды с атрибутикой МАЗ. Коэффициентная нормализация координат снизила ошибку позиционирования 3D-модели с 18 % до 2 % при смене устройства. Связка библиотек CameraX, ML Kit и SceneView обеспечивает задержку около 35 мс на кадр, достаточную для комфортного восприятия. Предложенный подход может быть перенесён на другие категории товаров, включая головные уборы и аксессуары, при условии дополнения модели соответствующими ключевыми точками. В дальнейшем планируется реализовать временно́й фильтр координат для подавления колебаний модели, расширить каталог доступной атрибутики предприятия, а также добавить функцию экспорта снимков примерки в социальные сети.

Список литературы

1. *Kipper G. Augmented Reality / G. Kipper, J. Rampolla. – Waltham: Syngress, 2013. – 176 p.*

2. *Прорывная технология IDM VTON [Электронный ресурс]. Режим доступа: https://idmvton.com/ru. Дата доступа: 01.03.2026.*

3. *Пальчевский, М. Д. Мобильное приложение для виртуальной примерки одежды с использованием ar-технологий = Mobile application for virtual fitting of clothes using ar-technology / М. Д. Пальчевский, А. А. Заяц, К. Р. Лыфарь // Электронные системы и технологии : сборник материалов 61-й научной конференции аспирантов, магистрантов и студентов БГУИР, Минск, 21–25 апреля 2025 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Д. В. Лихаческий [и др.]. – Минск, 2025. – С. 539–541.*

4. *Jetpack Compose [Электронный ресурс]. Режим доступа: https://developer.android.com/compose. Дата доступа: 01.03.2026.*

5. *Официальная документация CameraX [Электронный ресурс]. Режим доступа: https://developer.android.com/media/camera/camerax/overview. Дата доступа: 01.03.2026.*

6. *ML-Kit Pose Detection [Электронный ресурс]. Режим доступа: https://developers.google.com/ml-kit/vision/pose-detection. Дата доступа: 01.03.2026.*

7. *SceneView Android [Электронный ресурс]. Режим доступа: https://github.com/SceneView/sceneview-android. Дата доступа: 01.03.2026.*

UDC 004.932:004.42

POSE ESTIMATION PIPELINE AND COORDINATE NORMALIZATION IN AN AR APPLICATION FOR VIRTUAL CLOTHING FITTING

Zayats A.A.

National Children's Technopark, Minsk, Republic of Belarus

Prudnik A.M. – Cand. of Sci. (Tech.), Associate Professor at the Department of EPE;
Ilyasova M.S. – M.Sc. (Tech.), Assistant at the Department of EPE

Annotation. The architecture of a video frame processing pipeline in an Android application for virtual clothing fitting is described. Body landmark coordinate normalization methods ensuring correct 3D model rendering on devices with varying screen resolutions are presented. Test results on several smartphone models are provided.

Keywords: augmented reality, pose estimation, coordinate normalization, skeletal animation, CameraX, ML Kit, Kotlin, Android.