

Крусач.

Аннотация

В данной курсовой работе исследуются современные технологии WebAR и их применение в индустрии электронной коммерции на примере разработки интерактивного каталога одежды с виртуальной примеркой на манекене. Проект демонстрирует практическое использование дополненной реальности в веб-среде без необходимости установки мобильного приложения. В работе рассматриваются ключевые механики AR — трекинг по изображению, работа с 3D-ассетами, визуализация объектов в реальном времени, а также особенности взаимодействия пользователя с AR-интерфейсом.

На основе анализа существующих инструментов был выбран оптимальный стек технологий, включающий MindAR.js, Three.js, формат 3D-моделей glb и Node.js-хостинг. Разработка реализована с учётом ограничений мобильных браузеров, особенностей доступа к камере и производительности устройства. В практической части представлен прототип WebAR-каталога, позволяющий пользователю просматривать базовый манекен и переключать варианты одежды через удобное интерфейсное меню. Приложение работает в стандартном браузере смартфона, что обеспечивает высокую доступность и лёгкость демонстрации.

Работа носит как исследовательский, так и прикладной характер: она описывает современные подходы к WebAR-разработке и демонстрирует реальное применение AR-технологий в коммерческих задачах, создавая основу для дальнейшего расширения и интеграции подобных решений в онлайн-торговлю.

Введение

Современное развитие цифровых технологий всё активнее интегрируется в повседневную жизнь человека, изменяя способы восприятия, коммуникации и потребления информации. Одной из наиболее

перспективных областей цифровой трансформации является дополненная реальность (Augmented Reality, AR), которая позволяет объединять реальный и виртуальный мир в едином пространстве взаимодействия. В отличие от виртуальной реальности, создающей полностью искусственную среду, AR дополняет реальный мир цифровыми элементами, обеспечивая пользователю расширенное восприятие окружающего пространства. Благодаря стремительному развитию мобильных устройств и веб-технологий, дополненная реальность становится доступной не только для специализированных решений, но и для массового применения в коммерческих и образовательных сферах.

Одним из актуальных направлений внедрения AR является сфера электронной коммерции и, в частности, **индустрия моды**. Онлайн-покупки одежды стали неотъемлемой частью современной экономики, однако до сих пор сохраняется ключевая проблема — невозможность примерить вещь перед покупкой. Пользователь вынужден ориентироваться лишь на фотографии и описание товара, что нередко приводит к ошибкам в выборе размера, фасона или стиля. Это снижает удовлетворённость клиентов и увеличивает количество возвратов. Дополненная реальность предоставляет инновационное решение этой проблемы, позволяя пользователю **взаимодействовать с одеждой в трёхмерном пространстве**, рассматривать её со всех сторон и визуально оценивать посадку на виртуальном манекене.

В последние годы технологии WebAR (веб-дополненной реальности) открыли новые возможности для розничной торговли и онлайн-каталогов. В отличие от традиционных мобильных приложений, WebAR не требует установки дополнительного программного обеспечения — достаточно стандартного браузера с поддержкой камеры и WebXR. Это значительно упрощает доступ к контенту, повышает удобство и охват аудитории. Благодаря библиотекам вроде **Three.js**, **Zappar WebAR** или **8th Wall**, разработчики могут реализовывать высококачественные 3D-взаимодействия

прямо в веб-среде, обеспечивая кроссплатформенность и лёгкость интеграции с существующими интернет-магазинами.

Целью данной курсовой работы является **разработка WebAR-каталога одежды с возможностью виртуальной примерки на манекене**, демонстрирующего перспективы внедрения технологий дополненной реальности в онлайн-торговлю. Проект направлен на исследование современных подходов к реализации WebAR-приложений, анализ инструментов разработки и практическое создание интерактивного прототипа, позволяющего пользователю взаимодействовать с 3D-моделями одежды в браузере.

Практическая значимость проекта заключается в демонстрации реальных возможностей использования WebAR-технологий в коммерческих целях. Создание интерактивного каталога одежды с виртуальной примеркой позволяет улучшить пользовательский опыт, повысить доверие к онлайн-покупкам и сократить количество ошибок при выборе одежды. С теоретической точки зрения, работа также иллюстрирует актуальные подходы к реализации 3D-интерфейсов в браузере, интеграции с базами данных и работе с графикой в реальном времени.

1. Изучение новых механик для разработки AR-приложений

1.1. Трекинг и режимы дополнённой реальности

В основе созданного WebAR-каталога одежды лежит технология **image-tracking**, то есть трекинг по заранее подготовленному изображению-маркеру. Такой подход применяется, когда необходимо обеспечить максимально предсказуемое и стабильное поведение виртуальных объектов на широком спектре мобильных устройств.

Для реализации трекинга в проекте используется библиотека **MindAR.js**, специально адаптированная для работы в браузере. Её механизм основан на распознавании уникальных особенностей изображения (feature points). Когда камера определяет совпадение с обученным маркером, движок вычисляет его положение в кадре и передаёт координаты в трёхмерную сцену. Таким образом, виртуальный манекен закрепляется в пространстве относительно маркера и «движется» вместе с ним без дрожания и смещения.

В отличие от markerless-подхода (где требуется анализ плоскостей и SLAM-алгоритмы), image-tracking имеет ряд преимуществ:

- корректная работа даже на старых смартфонах;
- отсутствие необходимости в WebXR и сложных сенсорах;
- высокая скорость и точность распознавания при правильном выборе маркера;
- возможность использования в любых браузерах, включая Safari на iOS, где WebXR всё ещё ограничен.

В рамках разработки каталог ориентировался на стабильность работы, поэтому был выбран именно такой подход. При наведении камеры на обученное изображение система активирует **anchor** — точку привязки для виртуальной модели. На этот якорь помещаются 3D-объекты: базовый манекен и дополнительные варианты одежды.

Изучение устройства MindAR позволило понять основы:

- структуры файла targets.mind, содержащего данные маркера;
- принципов работы anchor-группы;

- особенностей запуска AR-сессии через `MindARThree.start()`;
- поведения событий `onTargetFound` и `onTargetLost`, позволяющих динамически отображать меню и заменять модели.

Таким образом, проект использует **классический marker-based AR**, обеспечивая простоту внедрения и стабильную работу на большинстве устройств без сложных библиотек и фреймворков.

1.2. Отрисовка и материалы (рендеринг)

Визуальная составляющая каталога реализована на основе трёхмерного движка **Three.js**. Он отвечает за отображение моделей, работу с камерой, освещение и анимацию. Загруженные модели представлены в формате **glb (glTF-binary)**, который является оптимизированным вариантом для WebAR.

Three.js позволяет управлять материалами, световым окружением и рендерингом. В проекте использованы:

- **HemisphereLight** — мягкий рассеянный свет;
- **DirectionalLight** — направленный свет для усиления объёмности;
- анимация вращения через `renderer.setAnimationLoop()`.

Материалы моделей используют физически корректный рендеринг (**PBR**), который обеспечивает реалистичное взаимодействие ткани с освещением. Формат glTF поддерживает:

- **baseColor** — цвет поверхности;
- **roughness** — степень матовости;
- **metallic** — отражающая способность;
- **normal map** — мелкие детали складок;
- **ambient occlusion** — затенение в углублениях.

Благодаря этому манекены выглядят естественно и одинаково отображаются на разных устройствах.

1.3. Оптимизация 3D-ассетов

Одним из ключевых аспектов WebAR является производительность на мобильных устройствах. Задержки при загрузке моделей или низкий FPS

могут негативно сказаться на качестве пользовательского опыта. Поэтому в проекте проводится оптимизация 3D-ассетов ещё до публикации.

Основные меры оптимизации включают:

- снижение количества полигонов в Blender;
- уменьшение разрешения текстур;
- экспорт в формат .glb, который быстрее обрабатывается;
- корректный выбор масштаба модели, чтобы избежать тяжёлой геометрии.

Несмотря на то что в данном проекте не применялись Draco, KTX2 или другие методы продвинутого сжатия, использование компактных моделей и единообразной структуры позволило добиться быстрой загрузки и стабильной работы даже с четырьмя вариантами манекена.

2. Инструментарий

После изучения современных технологий дополненной реальности важно выбрать инструменты, которые позволят реализовать проект максимально эффективно. Для веб-приложений, работающих прямо в браузере, приоритетом является стабильность, кроссплатформенность и простота интеграции. Именно на этих принципах и строится выбранный стек технологий.

2.1. Основной технологический стек

Основой проекта стала библиотека **MindAR.js**, которая реализует AR-трекинг по изображению напрямую в браузере. Простота интеграции и отсутствие требований к WebXR сделали её практически идеальным вариантом. Для решения нашей задачи — показать манекен и одежду при наведении камеры на заранее подготовленный маркер — не требуются сложные SLAM-алгоритмы или определение плоскостей. Поэтому marker-based подход оказался логичным выбором. При тестировании на разных устройствах MindAR показал себя стабильнее, чем многие современные фреймворки.

Важно и то, что MindAR отлично работает на iPhone, где WebXR до сих пор остаётся ограниченным, а доступ к камере иногда блокируется без HTTPS. Именно поэтому технологию markerless AR пришлось исключить: она требует полноценного сенсорного стека, который Safari просто не предоставляет. Мы также рассматривали популярное решение 8th Wall, однако от него пришлось отказаться по объективным причинам: платформа объявила о закрытии, ограничила доступ к инструментам и перестала поддерживать новые проекты. Выбирать закрывающееся решение для разработки учебного каталога было бы рискованно.

Для визуализации трёхмерных моделей выбрали **Three.js**. Это одна из самых надёжных и давно проверенных библиотек для WebGL, которая обеспечивает стабильный рендеринг сцены в реальном времени. Она легко совмещается с MindAR, не требует сложных настроек и не конфликтует с DOM. В проекте Three.js отвечает за освещение, позиционирование моделей, анимацию и общую структуру 3D-сцены. Такой подход позволяет загружать манекены и одежду в формате glb, вращать их и динамически заменять при выборе пользователем нового образа.

При создании пользовательского интерфейса было решено использовать **чистый JavaScript, HTML**, без фреймворков. React или Vue могли бы дать удобную структуру, но в контексте WebAR они создают дополнительные проблемы: повышенную нагрузку на оперативную память, лишние абстракции и снижение производительности в мобильном браузере, который уже испытывает давление из-за WebGL. Поскольку интерфейс приложения состоит всего из нескольких кнопок выбора наряда, применение тяжёлых фреймворков было бы неоправданным.

Деплой проекта был выполнен на платформе **Render**, которая обеспечивает автоматический HTTPS — это критично, потому что без него доступ к камере просто не работает на iOS. Render корректно раздаёт статические файлы, включая бинарные модели glb, и не вызывает ошибок MIME-типов, с которыми часто сталкиваются на Netlify или GitHub Pages.

Кроме того, Render позволяет в будущем добавить полноценный сервер, если это понадобится.

2.2. Инструменты для подготовки и оптимизации ассетов

Перед тем как интегрировать модели в WebAR, нужно было убедиться, что они оптимизированы под мобильные устройства. Основная работа велась в **Blender**, где модели приводились к единому масштабу, очищались от лишней геометрии и экспортировались в формат glb. Этот формат выбран не случайно — он загружается быстрее, чем FBX или OBJ, не требует отдельных файлов материалов и поддерживает PBR-текстуры, которые three.js рендерит без проблем.

Оптимизация в нашем проекте не требовала использования более тяжёлых инструментов, таких как Draco, gltf-transform или KTX2. Обычно они применяются в тех случаях, когда вес моделей превышает 20–30 мегабайт или сцена включает сложные высокополигональные объекты. Наши манекены и варианты одежды достаточно лёгкие, и их загрузка происходит быстро даже на слабых смартфонах.

Тем не менее мы придерживались важных принципов:

- избегали лишних текстур большого разрешения;
- следили за плотностью меша;
- убирали скрытую геометрию, которая не влияет на финальный рендер;
- контролировали масштаб, чтобы все модели корректно переключались на одном и том же anchor.

Такой подход обеспечил быструю загрузку модели при выборе каждого нового образа и достаточную частоту кадров в AR-сцене.

2.3. Обоснование выбора инструментов

Каждый инструмент в проекте выбран не “просто так”, а из-за конкретных технических причин. MindAR оказался лучшим вариантом из-за своей надёжности, отсутствия сложных зависимостей и стабильности на iOS. Markerless-решения были отброшены из-за ограничений Safari и из-за того,

что многие современные SLAM-платформы стали либо недоступны, либо требуют слишком мощного оборудования. Особенно показательной оказалась ситуация с 8th Wall, которая долгое время считалась почти стандартом WebAR-рынка, но в итоге объявила о закрытии. Такой сценарий полностью исключает возможность использовать её в учебных или долгосрочных проектах.

Three.js был выбран как естественный спутник MindAR: лёгкий, быстрый, с огромным сообществом и полным набором возможностей для работы с PBR-материалами и glb-моделями. Альтернативы вроде Babylon.js и PlayCanvas рассматривались, но их преимущества не перевешивали рисков: они тяжелее, требуют больше ресурсов и имеют более сложный порог интеграции с image-tracking.

Использование чистого JavaScript позволило избежать накладных расходов, которые создают React и другие SPA-фреймворки. В WebAR важен каждый мегабайт памяти, а каждый лишний фрейм может приводить к фризам. Поэтому отказ от фреймворков — это не упрощение, а осознанная оптимизация.

Хостинг на Render обеспечил стабильную работу камеры, корректную отдачу моделей и возможность дальнейшего развития проекта без ограничений. Это намного практичнее, чем статические хостинги, которые иногда блокируют доступ к `getUserMedia` или некорректно обслуживают бинарные файлы.

В итоге весь инструментарий подобран с учётом реальных условий работы WebAR: ограничений браузеров, производительности мобильных телефонов и необходимости максимально лёгкой интеграции. Такой подход обеспечивает проекту стабильность и позволяет пользователю получать предсказуемый и комфортный опыт дополнённой реальности прямо в браузере.