FashionBlend: Virtual Try-On System Using Augmented Reality
Technical Analysis Report

Submitted by: Vinay Kumar

Course: AR/VR

Department: Computer Science

Institution: Rajiv Gandhi Institute of Petroleum Technology, Jais

Course Instructor: Dr. Susham Biswas

Date: May 4, 2025

1 Executive Summary

FashionBlend is an innovative augmented reality (AR) virtual try-on system designed to revolutionize the online shopping experience. Developed as a mobile application, FashionBlend enables users to virtually try on clothing items in real-time using only a smartphone camera, eliminating the need for specialized hardware. By integrating advanced computer vision, pose estimation, and 3D modeling, the system provides a seamless and immersive experience, addressing key challenges in e-commerce, such as inaccurate sizing and high return rates. This report provides a detailed technical analysis of FashionBlend, highlighting its architecture, pipeline, and potential for integration with online retail platforms.

The project's significance lies in its accessibility and scalability. By leveraging smartphone-based AR, FashionBlend makes virtual try-on technology widely available, offering a cost-effective solution for retailers and an engaging experience for consumers. The system's robust pipeline ensures accurate garment visualization, paving the way for enhanced customer satisfaction and reduced environmental impact through fewer returns.

2 Introduction

The rise of e-commerce has transformed the retail industry, but challenges such as sizing inaccuracies and the inability to try on clothing before purchase persist. FashionBlend addresses these issues through an AR-based virtual try-on system that allows users to visualize garments on their bodies in real-time. The project combines cutting-edge technologies, including pose estimation, 3D garment modeling, and AR rendering, to deliver a user-friendly and efficient solution.

This report examines the technical architecture, development pipeline, and performance of FashionBlend, with a focus on its AR-driven approach. Drawing from the GitHub repository (`https://github.com/vinu0404/AR-FashionBlend.git`), the analysis highlights the system's innovative features, challenges overcome, and recommendations for future enhancements.

3 Technical Background

3.1 Evolution of Virtual Try-On Systems

Virtual try-on technology has progressed through distinct phases:

1. **Static 2D Overlays**: Early systems relied on manual alignment of 2D garment images over user photos.

2. **Automated 2D Mapping**: Introduction of pose estimation for automatic garment alignment.

3. **Hardware-Dependent 3D Solutions**: Use of depth-sensing cameras (e.g., Microsoft Kinect) for enhanced accuracy.

4. **Smartphone-Based AR**: Current systems leverage neural networks and smartphone cameras for real-time 3D visualization.

FashionBlend represents the latest generation, utilizing ARCore-supported smartphones to deliver a hardware-agnostic solution.

3.2 Key Technical Challenges

The development of FashionBlend addressed several challenges:

- **Accurate Pose Estimation**: Detecting and tracking human body joints in real-time.
- **Garment Scaling and Alignment**: Ensuring garments fit the user's body proportions and align dynamically with movements.
- **Real-Time Rendering**: Achieving smooth AR visualization on resource-constrained mobile devices.
- **User Accessibility**: Designing an intuitive interface for seamless interaction.
- **Performance Optimization**: Balancing processing speed with visual quality.

4 Technical Architecture

FashionBlend employs a client-server architecture optimized for AR applications, with distinct components handling data acquisition, processing, and visualization.

4.1 System Components

1. **Mobile Client Application (Android)**
   - Captures user video via smartphone camera.
   - Provides an intuitive interface for clothing selection and AR visualization.
   - Integrates Unity for real-time AR rendering.
   - Stores processed data locally for quick access.
2. **Server-Side Processing**
   - Handles computationally intensive tasks, including pose estimation and coordinate transformation.
   - Processes video frames and generates 3D pose data.
   - Manages garment model alignment and scaling.
3. **Data Storage**
   - Maintains a repository of 3D garment models and patterns.
   - Stores user authentication data and processed pose information.
   - Uses SQLite for lightweight, efficient data management.

4.2 Development Technologies

- **Frontend**: Android (API 24+), Unity with ARCore for AR rendering.
- **Backend**: Python with Flask for server-side processing.
- **Machine Learning**: TensorFlow for 2D and 3D pose estimation.
- **3D Modeling**: Blender for creating realistic garment models.
- **Database**: SQLite for storing user and garment data.
- **Hardware Requirements**: Android device with ARCore support, ARMv7 CPU, minimum 2GB RAM.

5 AR-Driven Technical Pipeline

FashionBlend's pipeline is designed to deliver a seamless AR experience through a series of interconnected stages.

5.1 Data Acquisition

1. Users record a short video using the mobile app's camera.
2. The video is uploaded to the server for processing.
3. Video frames are standardized to a resolution of 480×848 for consistent processing.

5.2 Pose Estimation

1. **2D Pose Estimation**:
    - Utilizes a TensorFlow-based model (inspired by OpenPose) to detect key body joints.
    - Outputs (x, y) coordinates for joints in pixel space.
    - Achieves 90–95% accuracy based on standard benchmarks.
2. **3D Pose Estimation**:
    - Converts 2D joint coordinates to 3D (x, y, z) space using a neural network model.
    - Applies preprocessing to enhance Z-coordinate accuracy, reducing average error to $\sim$45mm.
    - Outputs 3D joint positions for garment mapping.
3. **Coordinate Transformation**:
    - Normalizes 2D and 3D coordinates to align with Unity's coordinate system.

- Uses Unity's Camera.ScreenToWorldPoint for accurate AR overlay.

5.3 Garment Mapping

1. **Model Preparation**:
   - 3D garment models are created in Blender with defined attachment points.
   - Models include realistic properties (e.g., fabric texture, draping behavior).
   - Supports user-uploaded patterns via UV mapping.
2. **Dynamic Mapping**:
   - Aligns garment attachment points with 3D body joints.
   - Scales garments based on estimated user measurements.
   - Updates garment position and orientation in real-time to match user movements.

5.4 AR Visualization

1. Processes video frames for AR overlay.
2. Superimposes 3D garment models onto the user's body.
3. Renders the composite view in the mobile app using Unity's AR engine.
4. Ensures smooth visualization at 30+ FPS on supported devices.

6 Performance Optimization

To enhance user experience, FashionBlend incorporates several optimization strategies:

- **Asynchronous Processing**: Offloads pose estimation and garment mapping to the server, allowing users to continue interacting with the app.
- **Local Data Storage**: Caches processed pose data to avoid redundant computations.
- **Resolution Standardization**: Balances video quality and processing speed.
- **Notification System**: Alerts users when processing is complete, improving usability.

6.1 Performance Metrics

On reference hardware (Intel i5, 8GB RAM):

Module	Processing Time (300 Frames)
2D Pose Estimation	$\sim$7 minutes
3D Pose Estimation	$\sim$1.5 minutes
Total Processing	$\sim$8.5 minutes

Table 1: Processing Time Analysis

7 Technical Challenges and Solutions

7.1 Coordinate System Alignment

Challenge: Integrating 2D pixel coordinates, 3D world coordinates, and Unity's coordinate system for accurate AR rendering. **Solution**: Developed a transformation pipeline using normalization and Unity's ScreenToWorldPoint function to ensure seamless coordinate mapping.

7.2 3D Pose Accuracy

Challenge: Inaccurate Z-coordinates led to improper garment orientation. **Solution**: Implemented a preprocessing step to align torso and neck Z-coordinates, apply differential Z-movement, and smooth transitions between frames.

7.3 Processing Latency

Challenge: Long processing times impacted user experience. **Solution**: Introduced asynchronous processing with push notifications and local storage to minimize wait times and redundant computations.

8 Architecture Diagram

The system architecture consists of a user device running the mobile application and a server handling processing tasks. The mobile app captures video, visualizes AR content using Unity and ARCore, and stores processed data locally. The server standardizes video frames, performs 2D and 3D pose estimation, and maps garments to user poses, returning results to the client.

9 Future Enhancements

9.1 Technical Improvements

1. **Real-Time Processing**: Optimize the pipeline for on-device processing using TensorFlow Lite.
2. **Expanded Garment Library**: Include diverse clothing types (e.g., dresses, accessories).
3. **Cross-Platform Support**: Extend to iOS and web-based AR platforms.
4. **Physics-Based Rendering**: Integrate cloth physics for realistic draping and movement.
5. **Size Recommendation**: Implement AI-driven algorithms to suggest optimal garment sizes.

9.2 Strategic Recommendations

1. Partner with e-commerce platforms to integrate FashionBlend as a try-on feature.
2. Explore edge computing to reduce server dependency and latency.
3. Conduct user testing to refine the interface and enhance accessibility.
4. Investigate advanced neural networks (e.g., MobileNetV3) for improved pose estimation.

10 Conclusion

FashionBlend represents a significant advancement in AR-based virtual try-on technology, offering a scalable and accessible solution for the e-commerce industry. By leveraging smartphone cameras, computer vision, and 3D modeling, the system delivers an immersive experience that bridges the gap between online and in-store shopping. Despite challenges such as processing latency, FashionBlend's optimized pipeline and innovative architecture lay a strong foundation for future development.

With potential applications in retail, fashion design, and personalized shopping, FashionBlend has the opportunity to transform how consumers interact with clothing online, driving engagement and reducing return rates. Continued investment in real-time processing and cross-platform support will further enhance its impact.

11 References

1. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Field. https://arxiv.org/abs/1812.08008
2. 3D Human Pose Estimation Baseline. https://arxiv.org/pdf/1705.03098.pdf
3. Project Repository: https://github.com/vinu0404/AR-FashionBlend.git