INTERNSHIP REPORT

Title: Internship Report at Algoarn

Name: Pooja Shekharagouda Doddagoudar

USN: 2SD22CS130

Course: Computer Science

Duration of Internship: April 2025 – October 2025

Project Title: VTR (Virtual Trial Room)

Project using AI and Computer Vision

Submitted to: Mr. Prajwal Gaddigoudar

(Mentor, Algoarn)

Acknowledgement

I would like to express my sincere gratitude to **Algoarn** for giving me the invaluable opportunity to pursue my internship in their esteemed organization. This internship has been a highly enriching and transformative experience, allowing me to practically apply my academic knowledge in real-world projects while exploring the dynamic fields of **Artificial Intelligence, Computer Vision, and Generative AI**.

I am deeply indebted to my mentor, **Mr. Prajwal Gaddigoudar**, for his constant guidance, support, and encouragement throughout the internship. Even in a virtual setup, his mentorship was highly effective; he provided timely feedback, shared resources, and ensured that I was able to overcome challenges in the development of the **Virtual Try Room (VTR) project**. His guidance strengthened my technical skills in algorithm development, problem-solving, and professional practices, and taught me the importance of self-discipline, communication, and proactive learning in a remote environment.

I would also like to acknowledge the **entire Algoarn team** for providing a collaborative and motivating environment. Even though the internship was conducted remotely, their support, willingness to share knowledge, and timely guidance ensured that I could actively participate in project development and learn effectively.

I am grateful to my **college/university** for facilitating this internship and encouraging practical learning that bridges the gap between theory and application.

Lastly, I extend my thanks to my **peers, colleagues, and friends** who provided continuous motivation, guidance, and support during this internship. Their encouragement helped me complete my tasks effectively and make the most of this learning experience.

This internship at Algoarn has been a remarkable milestone in my academic and professional journey, and I remain sincerely thankful to everyone who contributed to this valuable experience.

Table of contents

Introduction

Algoarn is an **AI-based company** dedicated to embedding intelligence into everyday business processes. Its motto, *"Smart AI for Smarter Business"*, reflects the organization's commitment to leveraging **Artificial Intelligence, Machine Learning, and Computer Vision** to deliver innovative and practical solutions across various industries. The company focuses on developing AI-driven tools that solve complex business challenges while enhancing operational efficiency, customer experience, and decision-making processes.

During my internship at Algoarn, I had the opportunity to work on the **Virtual Try Room (VTR) project**, a pioneering initiative designed to transform the online retail experience. The VTR project enables users to virtually try on clothes using advanced AI algorithms, helping them make more informed purchasing decisions and bridging the gap between physical and online shopping experiences. My role in the project involved working with sophisticated technologies such as **pose estimation, depth estimation, 3D mesh generation, and texture mapping**, which are critical for creating realistic virtual fittings. Through this project, I gained hands-on experience in developing AI-based solutions that combine computer vision, graphics, and optimization techniques.

I chose Algoarn for my internship because of its **focus on cutting-edge AI solutions**, its commitment to innovation, and the exposure it offers to real-world projects at the forefront of AI research and application. The opportunity to work under the mentorship of experienced professionals, contribute to a product with tangible impact, and learn advanced technical concepts in a structured environment made this internship an ideal choice for my professional growth.

Moreover, this internship allowed me to understand the **complete lifecycle of AI product development**, from conceptualization and algorithm design to implementation and testing, providing invaluable insights into industry practices and standards. Working on the VTR project not only enhanced my technical expertise but also strengthened my problem-solving, analytical, and project management skills, preparing me for future challenges in the field of Artificial Intelligence.

Company Profile

Algoarn is an **AI-based company** dedicated to transforming industries by embedding intelligent solutions into everyday business processes. The company focuses on developing **practical and scalable AI technologies** that enhance operational efficiency, improve decision-making, and provide innovative solutions across multiple sectors. With expertise in **Computer Vision, Predictive AI, and Generative AI**, Algoarn bridges the gap between advanced AI research and real-world applications, helping businesses adopt intelligent solutions that deliver tangible results.

Major Focus Areas: Computer Vision, Generative AI, Predictive AI.

Notable Innovations

- **Virtual Try Room (VTR):** An AI-powered platform that allows users to virtually try on clothing, using technologies like **pose estimation, depth mapping, 3D mesh generation, and texture mapping** to create a realistic experience.
- **SUCA Project.**

Mission: To create **AI-powered systems** that address real-world problems, improve operational efficiency, and enable businesses to make smarter, data-driven decisions.

Vision: To **revolutionize industries** by embedding intelligence into everyday processes, helping organizations leverage AI for enhanced productivity, innovation, and superior customer experiences.

Internship Objectives

The objectives of my internship were as follows:

1. Gain exposure to **computer vision algorithms** and their practical applications.
2. Apply **machine learning and depth estimation models** to build real-world systems.
3. Improve skills in **Python programming, debugging, and 3D mesh processing**.
4. Contribute directly to the **Virtual Try Room (VTR) project** by implementing algorithms for realistic shirt wrapping and fitting.
5. Develop teamwork, problem-solving, and communication skills through active collaboration.

Work & Tasks Assigned

During my internship at Algoarn, I was actively involved in the development of the **Virtual Try Room (VTR) pipeline**, which required integrating computer vision, depth estimation, and 3D mesh processing techniques. The internship was structured with specific tasks assigned on a timeline, which helped me progress from theoretical understanding to practical implementation.

Timeline of Tasks and Expected Outputs

Task	**Expected Output**
Literature survey, use case diagrams, UML diagrams	Documented survey with use case and UML diagrams
Body detection, pose estimation, mesh model, depth estimation (study + coding with example)	Initial working code and understanding of body detection & depth estimation
Shirt overlay on the output of pose estimation or mesh model	Prototype overlay on detected model
Capture picture of object/person → convert into 2.5D map using depth estimation	2.5D map representation
Real-time body detection, pose estimation, depth estimation using webcam → create mesh model	Live system generating body mesh
Wrap shirt on the person using Blender	Shirt-wrapped 3D model demo

□ Method 1: Virtual Trial Room — Python AI-Based 3D Shirt Fitting

□Overview

This Python script enables a quick **virtual try-on** experience for an **upper-body shirt** using a combination of **computer vision**, **AI-based depth estimation**, and **3D mesh processing**.
It performs a realistic shirt fitting by:

- Capturing a live **webcam frame** of the user.
- Estimating **depth maps** for both the person and the shirt using **MiDaS**.
- Building **3D meshes** from the depth data and alpha masks.
- Detecting **body keypoints** using **MediaPipe Pose Estimation**.
- **Projecting and wrapping** the shirt mesh onto the person's body using **KDTree-based surface mapping**, along with scaling, rotation, and neck-hole adjustment.
- Exporting combined **OBJ/MTL files** and textures to the output/ directory.

□ Libraries and Main Components

Library	Purpose
os	File and directory management (creating output/, handling paths).
cv2 (OpenCV)	Webcam capture, image I/O, resizing, thresholding, and texture saving.
torch	Loads and runs the MiDaS depth estimation model (GPU/CPU).
numpy	Numerical computation and geometric transformations.
trimesh	Mesh loading, editing, and exporting for 3D models.
mediapipe	Pose detection (shoulders, elbows, nose, hips) and selfie segmentation for alpha masks.
ctypes	Reads screen size to scale preview window (Windows-specific).
warnings	Suppresses unnecessary runtime warnings.
scipy.spatial.KDTree	Performs fast nearest-neighbor lookups to snap the shirt mesh onto the person's mesh.

⚙ High-Level Pipeline (Step-by-Step)

1. Start / Setup

- Create the output/ folder.
- Load **MiDaS small** model and transformations via torch.hub.
- Initialize **MediaPipe Pose** and **Selfie Segmentation** modules.

2. Capture

- Open a resizable webcam preview window.
- Press **SPACE** to capture the frame.

3. Foreground Mask & Person Texture

- Generate an alpha mask using **Selfie Segmentation**.
- Combine RGB and alpha channels to form an **RGBA texture** for the person.

4. Depth Estimation (MiDaS)

- Run MiDaS on both the person frame and the shirt image to produce depth_person and depth_shirt.

5. Mesh Creation from Depth + Alpha

- create_mesh_from_depth() builds:
  - **Vertices (x, y, z)** where z is scaled depth.
  - **UV coordinates** based on image-space projection.
  - **Triangle faces** for all valid alpha pixels.
- Exports a simple, textured OBJ using save_obj().

6. Pose Keypoints Extraction

- extract_pose_keypoints() returns 2D pixel coordinates for major landmarks (shoulders, elbows, wrists, hips, nose, neck).

7. Shirt Projection / Wrapping

- project_shirt_to_upper_keypoints() performs the main wrapping process:
  - Matches 3D person vertices to 2D pose keypoints.
  - Scales the shirt mesh to match shoulder width.
  - Rotates the shirt for correct shoulder alignment.
  - Uses **KDTree** to “snap” shirt vertices to the nearest body surface points.
  - Extends sleeves toward elbows.
  - Cuts a **neck opening** based on proximity to neck center.
  - Smooths and recalculates UVs.

8. Combine & Export

- Merges person and shirt meshes.
- Saves final combined .obj and .mtl models with separate materials and textures inside output/.

💾 Files Created

- output/person_mesh.obj + .mtl + person_texture.png
- output/shirt_mesh.obj + .mtl + shirt_texture.png
- output/person_with_upperbody_shirt.obj + .mtl — **final fitted model**

□ Key Functions

Function	Description
capture_image_from_webcam()	Opens webcam, captures frame on key press.
extract_pose_keypoints(image)	Returns 2D coordinates of body landmarks.
create_mesh_from_depth(depth, rgba, ...)	Builds 3D mesh from depth + alpha data.
save_obj(path, verts, faces, uvs, texture)	Exports OBJ/MTL with texture references.
project_shirt_to_upper_keypoints(...)	Core wrapping and alignment logic.

save_combined_obj_with_shirt_wrap(...)	Merges meshes and saves final model.

⚙ Important Parameters & Heuristics

Parameter	Purpose
depth_scale = 200.0	Converts normalized MiDaS depth to Z-units.
Alpha threshold = 50	Filters valid mesh pixels.
Shirt scale = (shoulder_dist * 1.18) / shirt_width_px	Adjusts size based on body proportions.
KDTree alpha blending	Controls how strongly the shirt adheres to the surface.
Neck radius = 0.24 * shoulder_pixel_distance	Defines the neck-hole size.
Laplacian smoothing	Refines final mesh for smoother appearance.

⚠ Limitations & Issues

◈ Depth / Geometry

- MiDaS produces **relative depth**, not metric — scaling can vary.
- No **camera intrinsics**, so 3D positions are approximate.
- Meshes may be **dense or noisy**, not ideal for physics or animation.
- Normals not exported — shading may look inconsistent.

◈ Wrapping / Alignment

- KDTree snapping is **naïve** — can clip through body.
- No physics (gravity, drape, collisions).
- Neck-hole carving is **2D heuristic**, may over/under-cut.
- Sleeve alignment is rough, not topology-aware.
- Rotation limited to **Y-axis** only — torso tilt may misalign.

◈ Pose & Landmarks

- Uses static MediaPipe mode — no temporal smoothing.
- Neck point is an estimated average, may vary across poses.

◈ Texturing / UV Mapping

- Bounding-box UV projection — may stretch shirt textures.
- Minimal material separation and surface color settings.

◈ Reliability / Compatibility

- Windows-specific (uses ctypes.windll.user32).
- Needs internet on first run to download MiDaS model.
- High memory usage for large meshes.
- Unity/Three.js may require rotation/scaling fix.

🚀 Potential Improvements

- Integrate **camera intrinsics** for real 3D coordinate reprojection.
- Use **normal-aware projection** for better surface conformity.
- Replace KDTree with **closest-point-on-triangle** mapping.
- Add **cloth simulation** (mass-spring or physics-based).
- Detect neck hole in 3D instead of 2D projection.
- Perform **retopology** for clean, manifold meshes.
- Preserve original **shirt UVs** via UV-transfer.
- Extend to **MediaPipe Holistic** for full-body keypoints.
- Add **temporal smoothing** for real-time webcam mode.

📦 Runtime Notes & Requirements

- Requires E:\VTR PROJECT\output\blackshirt.jpg (shirt image).
- First run downloads **MiDaS model** via torch.hub.
- Outputs saved to output/ directory.

▯ Common Runtime Issues

Issue	Description
Shirt floating or clipping	Incorrect depth scaling or alignment.
Neck hole too large / misplaced	Neck heuristic mismatch.
Sleeves look distorted	Elbow projection not aligned.
OBJ imports incorrectly	Axis mismatch or missing normals.
Slow performance	High-res frames or CPU-only MiDaS inference.

✅ Result:

Generates a realistic 3D shirt fitting mesh aligned to the user's pose and anatomy, exportable to Unity, Blender, or Three.js for visualization and further processing.

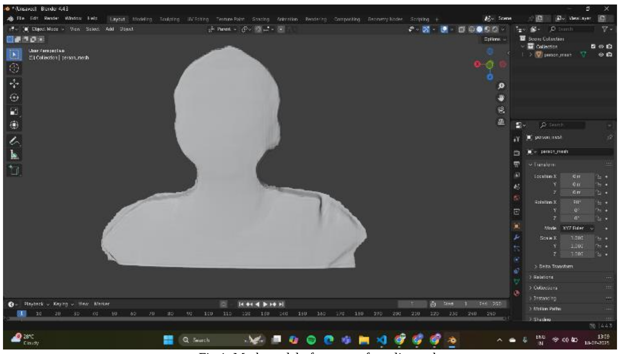

Fig 1. Mesh model of a person from live web cam.

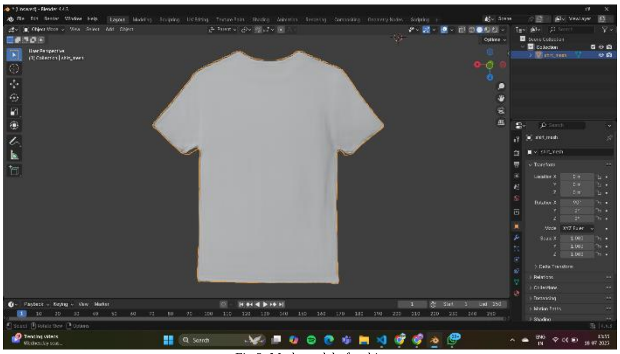

Fig 2. Mesh model of a shirt.

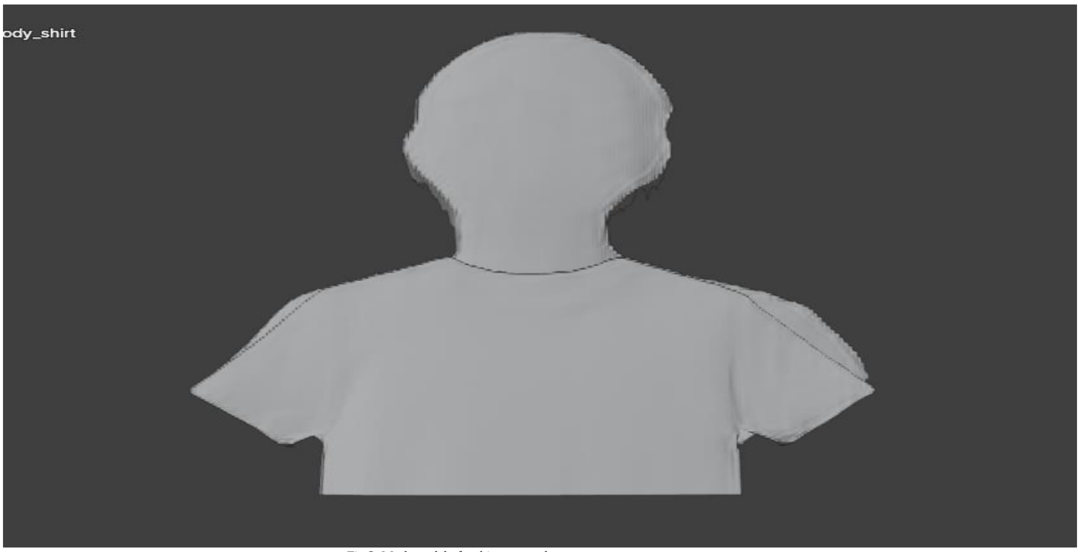

Fig 3. Mesh model of a shirt wrapped on a person

□ Method 2: Full-Body Virtual Try-On with OOTDiffusion

OOTDiffusion is a latent diffusion model designed for controllable virtual try-on experiences. This method allows users to visualize how a garment will look on their full body, including tops, bottoms, or dresses.

□ Overview

This method enables a full-body virtual try-on by:

- Uploading a photo of the user (person image).
- Uploading a photo of the garment (clothing image).
- Selecting the appropriate garment category (e.g., upper-body, lower-body, or full-body).
- Running the model to generate the try-on image.

⚙ Workflow

1. **Prepare Input Images**
    - **Person Image**: A clear, high-resolution photo of the user, preferably with a neutral background.
    - **Garment Image**: A standalone image of the clothing item to be tried on, without a model.

2. **Select Garment Category**
    - Choose the appropriate category for the garment:
        - **Upper-body**: For tops, shirts, jackets.
        - **Lower-body**: For pants, skirts, shorts.
        - **Dress**: For full-body garments.
3. **Run the Model**
    - Upload both images to the OOTDiffusion interface.
    - Select the garment category.
    - Click on the "Run" button to generate the virtual try-on image.

4. **Review and Save**
   - Once the image is generated, review the result.
   - Save the image if satisfied with the output.

🗀 Output Files

- **Generated Image**: The final image showcasing the user wearing the selected garment.

▯ Key Functions

Function	Description
`upload_person_image()`	Uploads the user's photo.
`upload_garment_image()`	Uploads the garment's photo.
`select_garment_category()`	Selects the garment category (upper-body, etc.).
`generate_virtual_tryon()`	Runs the model to generate the try-on image.
`save_generated_image()`	Saves the generated image.

⚙ Important Parameters & Heuristics

Parameter	Purpose
`image_resolution`	Sets the resolution for input images.
`output_format`	Specifies the format for the output image.
`category_selection`	Determines the garment category for try-on.

⚠ Limitations & Issues

- **Image Quality**: Low-resolution images may result in poor-quality outputs.
- **Background Interference**: Complex or cluttered backgrounds can affect the model's performance.
- **Garment Transparency**: Semi-transparent garments may not render accurately.
- **Pose Variability**: Extreme poses may lead to misalignment in the try-on image.

🚀 Potential Improvements

- **Enhanced Pose Estimation**: Integrate advanced pose detection for better alignment.
- **Background Removal**: Implement automatic background removal for cleaner inputs.
- **Multi-Garment Try-On**: Allow users to try on multiple garments simultaneously.
- **Real-Time Processing**: Optimize the model for faster, real-time virtual try-on experiences.

Learning Outcomes

Technical Learnings

- Gained **hands-on proficiency** in **MediaPipe Pose** for detecting key body landmarks such as shoulders, neck, and elbows, which was critical for realistic virtual try-on.
- Learned to implement **MiDaS depth estimation models** to convert 2D images into accurate **3D representations**, enabling depth-aware clothing overlay.
- Developed expertise in **3D mesh construction**, including **vertex manipulation, Laplacian smoothing**, and **UV mapping** for aligning digital shirts onto human meshes.
- Enhanced understanding of **coordinate transformations, scaling, and rotation** techniques for fitting objects to human body models.
- Improved **debugging and problem-solving skills** in complex AI pipelines that integrated multiple models and processing stages.
- Gained exposure to **real-time computer vision applications**, including **webcam integration**, efficient data handling, and optimization for smooth performance.
- Understood the process of **image segmentation** and **masking**, crucial for separating subjects from the background in virtual try-on systems.
- Developed knowledge in **nearest-neighbour mapping** using **KDTree**, improving mesh alignment and accuracy of overlays.
- Learned the importance of **data preprocessing, normalization, and scaling** to ensure accurate depth estimation and mesh generation.

Professional Learnings

- Learned to **collaborate effectively** with a mentor and team members, including sharing progress updates and integrating feedback.
- Developed strong **time management skills**, planning tasks according to deadlines and prioritizing key project milestones.
- Enhanced **documentation and reporting abilities**, maintaining clear records of experiments, algorithm changes, and outputs.
- Strengthened **analytical thinking and problem-solving skills** by iteratively testing algorithms, identifying bottlenecks, and refining results for accuracy and realism.
- Improved **communication skills**, explaining technical concepts clearly and discussing challenges and solutions with the mentor.
- Gained insight into **project workflow and methodology**, including breaking down complex problems into manageable tasks and following a systematic approach to development.
- Learned to **adapt to new tools and frameworks** quickly, including PyTorch, OpenCV, NumPy, MediaPipe, and Trimesh, in a project-driven environment.
- Developed the ability to **translate theoretical concepts into practical applications**, understanding how AI algorithms can be applied to real-world systems.

Challenges Faced

1. Depth Estimation Errors

- **Challenge:** Early attempts at generating 3D meshes produced **distorted or inaccurate models** due to inconsistent tensor dimensions and improper scaling of depth maps.
- **Solution:** This issue was resolved by **normalizing input images**, adjusting tensor shapes to match model requirements, and ensuring consistent preprocessing for depth estimation. These steps improved the accuracy of the generated 3D meshes, providing a reliable foundation for subsequent shirt overlay and fitting tasks.

2. Shirt Misalignment

- **Challenge:** Initial shirt wrapping often resulted in **misaligned meshes around the shoulders, chest, and neckline**, leading to unrealistic virtual try-on results.
- **Solution:** The problem was addressed by **directly integrating pose keypoints from MediaPipe** with mesh deformation algorithms. By using shoulder, neck, and elbow landmarks, the shirt mesh could be scaled, rotated, and translated to match the anatomical features of the user, significantly improving the realism of the virtual fitting.

3. Performance Bottlenecks

- **Challenge:** Integrating **real-time webcam input** for live try-on initially slowed the system, causing delays in pose estimation, depth calculation, and mesh rendering.
- **Solution:** Optimization was achieved by switching to the **MiDaS_small model** for depth estimation, which required fewer computational resources, and by **refactoring calculations** to reduce redundant operations. These improvements allowed smoother real-time interaction without compromising accuracy.

4. Texture and Fitting Realism

- **Challenge:** Ensuring **natural appearance and realistic fitting** of the shirt around the chest, shoulders, and sleeves was complex, as minor misalignments could break the illusion of proper fitting.
- **Solution:** Applied **UV mapping techniques** and **fine-tuned KDTree blending** to project shirt vertices onto the body mesh accurately. Iterative testing and adjustments to vertex positions enhanced visual fidelity and made the virtual try-on appear more lifelike.

Conclusion

- The internship at **Algoarn** was a highly **rewarding and enriching experience**.
- Worked on the **Virtual Try Room (VTR) project**, applying knowledge of **AI, Computer Vision, and 3D modeling**.
- Gained hands-on experience in **pose estimation, depth estimation, 3D mesh generation, and texture mapping**.
- Enhanced **technical expertise**, including debugging complex AI pipelines and integrating multiple models for real-time applications.
- Developed **professional skills** such as **time management, documentation, problem-solving, and teamwork**.
- Learned to overcome challenges like **depth estimation errors, mesh misalignment, and performance bottlenecks** through iterative testing and optimization.
- Obtained valuable **exposure to real-world AI project workflows**, bridging the gap between academic theory and practical implementation.
- Strengthened **confidence and interest in pursuing a career in AI**, particularly in **Computer Vision, AR, and VR applications**.
- Expressed sincere gratitude to **Algoarn** and **mentor Mr. Prajwal Gaddigoudar** for guidance, support, and this valuable learning opportunity.
- Overall, the internship served as a **pivotal step in professional and technical growth**, preparing for more advanced challenges in AI development.