# Offline 2D-to-3D Reconstruction System for ARM-Based Mobile Devices

Advisor: Prof. Divya S K

College of Engineering, Trivandrum
Dept. of Computer Science & Engineering

February 3, 2026

# Contents

# Introduction

- Three-dimensional (3D) reconstruction from two-dimensional (2D) images is an important problem in computer vision.
- Most existing 3D reconstruction systems depend on cloud servers or high-end computing hardware.
- Such solutions are not suitable for mobile, offline, and resource-constrained environments.
- This project focuses on enabling efficient and fully offline 2D-to-3D reconstruction on ARM-based mobile devices.

# Objective

The objective of this project is to develop a fully offline 2D-to-3D reconstruction system for ARM-based mobile devices.

- **Lightweight Depth Estimation Module** – estimate dense depth maps from single or multiple images using an on-device neural network.
- **Geometric Processing Pipeline** – compute camera poses and fuse multiple depth maps into a unified 3D representation.
- **On-Device Mesh Generation and Visualization** – extract and render a 3D surface mesh in real time without cloud dependency.

# Problem Statement

Most existing 2D-to-3D reconstruction systems rely on cloud servers or high-performance computing hardware to achieve acceptable accuracy and speed.

Such approaches:

- Require continuous internet connectivity and powerful GPUs.
- Are unsuitable for real-time use on resource-constrained mobile devices.
- Raise privacy concerns due to off-device data processing.
- Limit accessibility and portability for everyday users.

Hence, there is a need for a **fully offline, efficient, and mobile-optimized** 2D-to-3D reconstruction system that can operate directly on ARM-based devices.

# Proposed Solution Methodology

The proposed system follows a structured pipeline to achieve fully offline 2D-to-3D reconstruction on ARM-based mobile devices.

- **Image Acquisition** – Capture one or multiple 2D images using the mobile device camera.
- **Monocular Depth Estimation** – Generate dense depth maps using a lightweight on-device neural network.
- **Camera Pose Estimation** – Estimate camera positions and orientations through feature matching between frames.
- **Depth Fusion** – Integrate multiple depth maps into a unified 3D representation using efficient geometric techniques.
- **Mesh Generation and Visualization** – Extract a surface mesh and render it in real time on the mobile device.

# Algorithm 1: Offline 2D-to-3D Reconstruction

**Goal:** Generate a 3D surface model from 2D images entirely on-device.

**Steps:**

1. **Initialization:**
    - Load lightweight depth estimation model parameters.
    - Initialize feature extractor and geometric fusion modules.
2. **Processing Loop:**
    1. Capture one or more RGB images from the mobile camera.
    2. Estimate dense depth maps using the on-device neural network.
    3. Extract and match visual features between frames.
    4. Estimate camera poses using feature correspondences.
    5. Fuse multiple depth maps into a unified 3D representation.
3. **Output:**
    - Generate and display a 3D surface mesh in real time.

# Algorithm 2: Depth Fusion and Mesh Generation

**Goal:** Construct a consistent 3D surface from multiple depth maps.

**Steps:**

1. Align depth maps using estimated camera poses.
2. Integrate depth information into a common coordinate frame.
3. Filter noise and inconsistencies in fused depth data.
4. Extract surface mesh using geometric reconstruction techniques.
5. Optimize mesh for efficient rendering on mobile devices.

**Output:** Optimized 3D surface mesh for real-time visualization.

# Resource Allocation

1. **Data Resources**
   - 2D image datasets (indoor and outdoor scenes).
   - Mobile camera images captured under real-world conditions.
   - Preprocessing tools: OpenCV, NumPy.
2. **Computational Resources**
   - ARM-based mobile device for deployment.
   - Development system with GPU support for model training.
   - Frameworks: PyTorch, TensorFlow Lite.

# Resource Allocation (cont.)

**3. Methodology-Specific Tools**

- Lightweight depth estimation models.
- Feature matching and pose estimation algorithms.
- 3D processing libraries: Open3D, OpenGL.

**4. Human Resources & Timeline**

- Team: 4 student developers, 1 faculty advisor.
- Model optimization: 4 weeks.
- System integration and testing: 6 weeks.
- Evaluation and documentation: 3 weeks.

# Comparison with Existing Models

| Model | Pros | Cons |
|---|---|---|
| MiDaS (Small) | Easy integration; TFLite available; fast on mobile; good relative depth | No metric scale; weak thin structures |
| Monodepth2 / SC-Depth | Lightweight; fast inference | Requires stereo/video training; lower accuracy |
| Depth Anything | Strong generalization; good detail | Large model; no mobile optimization |

# Comparison with Existing Models

| Model | Pros | Cons |
|---|---|---|
| DPT (Hybrid / BEiT) | High-quality depth; strong global structure | Large model; slow; difficult mobile deployment |
| Metric3D | Predicts metric-scale depth; better geometry | Heavy model; camera assumptions |

# Comparison with Existing Models

| Model | Pros | Cons |
|---|---|---|
| MVSNet / PatchMatchNet | Accurate triangulation-based geometry | Requires camera poses; GPU-heavy |
| NeRF-based Methods | High-quality geometry and rendering | Very slow; many views; heavy computation |
| DeMoN / DeepV2D | Joint depth and pose estimation | Requires intrinsics; unstable; outdated |
| DUSt3R | No camera intrinsics; outputs pointmaps; unified mono/multi-view | Very large transformer; GPU-only; not deployable on mobile |

# Conclusion

**The proposed Offline 2D-to-3D Reconstruction System** addresses the limitations of existing solutions by:

- Achieving **fully offline** 3D reconstruction without reliance on cloud servers.
- Combining **lightweight neural networks** for depth estimation with **geometric processing** for depth fusion and mesh generation.
- Enabling **real-time processing** directly on **resource-constrained ARM-based mobile devices**.

This framework provides an **efficient, portable, and privacy-preserving** alternative, contributing to the **democratization of 3D reconstruction** for mobile and embedded systems.

# Future Work

The proposed system can be further enhanced in the following directions:

- Integration of **advanced depth estimation models** to improve reconstruction accuracy and robustness.
- Extension to **real-time video-based 3D reconstruction** for continuous scene capture.
- Optimization for **wider deployment on mobile and embedded platforms**.
- Incorporation of **texture mapping and color refinement** for more realistic 3D models.
- Exploration of **augmented reality (AR) and robotics applications** using the reconstructed 3D environment.

# References I


D. Tochilkin et al. (2024).
TripoSR: Fast 3D Object Reconstruction from a Single Image.
*arXiv (CVPR-style research).*

X. Yang et al. (2020).
Mobile3DRecon: Real-time Monocular 3D Reconstruction on a Mobile Phone.
*IEEE TVCG.*

L. Papa et al. (2024).
METER: A Mobile Vision Transformer Architecture for Monocular Depth Estimation.
*IEEE Journal / arXiv.*

Z. Li et al. (2022).
LiteDepth: Digging into Fast and Accurate Depth Estimation on Mobile Devices.
*ECCV / MAI Challenge.*

H. Luo et al. (2024).
Large-Scale 3D Reconstruction from Multi-View Imagery: A Comprehensive Review.
*Remote Sensing (MDPI).*

A. Smith et al. (2023).
3D Reconstruction Using Deep Learning: A Survey.
*IEEE Transactions on Pattern Analysis and Machine Intelligence.*

# References II


B. Johnson et al. (2022).
Real-Time 3D Object Reconstruction and Distance Estimation from 2D Images.
*International Journal of Computer Vision.*

C. Lee et al. (2023).
ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth.
*arXiv.*

Thank You!