Technical Report

3D Reconstruction Task and Manipulation in MuJoCo for Sensing and Perception Coursework

Individual Name:

Abhishek Jayant Patil
abhishek.patil@kcl.ac.uk

Date: 20th November 2025

Kings College London
Department of Engineering - Robotics

Abstract

This work investigates whether systematic, scene-invariant errors inherent to monocular depth estimation can be learned and corrected to produce depth predictions suitable for integration into convex optimization-based Structure-from-Motion (SfM) pipelines. It leverages the XM solver, a recently developed framework that combines learned depth with scaled bundle adjustment formulated as a convex semidefinite program (SDP), to accelerate dense 3D reconstruction while maintaining the geometric accuracy required for downstream robotic manipulation tasks. Our pipeline integrates monocular depth estimation, learned depth correction, the XM convex optimization solver, mesh post-processing (Meshlab and Blender), and validation through simulated grasping tasks (MuJoCo). While the learning component remains under active development, this work reports the preliminary findings, implementation details, and lessons learned from this integrated system that combines traditional MVS approaches with modern convex optimization techniques.

1 Github repository link

https://github.com/abhishek-patil1107/LMVS-MD.git

Use the above link to find the relevant code and dataset used for the reconstruction as well as a video showing the manipulation of the reconstructed object.

2 Introduction and Motivation

Monocular depth estimation networks offer rapid per-frame depth inference but exhibit consistent, predictable biases that limit their utility in geometric reconstruction pipelines. Common failure modes include infinite depth assignment to sky regions, overly smooth planar surfaces for floors and walls, distance-dependent depth compression following characteristic curves, and edge blurring correlated with image gradients. These systematic errors, while problematic for direct use in Multi-View Stereo (MVS), exhibit stability within scene categories (e.g., indoor versus outdoor environments), suggesting they may be amenable to learned correction.

Recent advances in convex optimization for bundle adjustment offer a promising avenue for integrating learned monocular depth into reconstruction pipelines. The XM framework formulates scaled bundle adjustment (SBA) by lifting 2D keypoint measurements to 3D using learned depth, designs a convex semidefinite program relaxation that solves SBA to certifiable global optimality, and implements a CUDA-based trust-region Riemannian optimizer for extreme-scale problems. This approach is initialization-free and demonstrates superior scalability compared to traditional bundle adjustment solvers [3].

2.1 Hypothesis

This work hypothesizes that a learned feedforward transformation can map raw monocular depth estimates to high-quality depth predictions suitable for the XM solver's scaled bundle adjustment formulation. By correcting systematic depth errors before integration into the convex optimization framework, We aim to improve the quality of 3D observations, reduce optimization convergence time, and ultimately accelerate dense reconstruction without requiring computationally expensive per-scene optimization or careful initialization.

3 Background: Convex Optimization for Bundle Adjustment

Traditional bundle adjustment formulations are non-convex and highly sensitive to initialization, often requiring iterative local optimization methods such as Levenberg-Marquardt [5]. The XM approach reformulates bundle adjustment as a scaled bundle adjustment problem that estimates scaling factors to correct learned depth while jointly estimating 3D landmarks and camera poses.

The key insight is that learned monocular depth provides approximate scale information that, when combined with 2D correspondences, enables the formulation of a convex semidefinite program. The SDP relaxation is tight in practice, meaning the global optimum of the relaxed problem coincides with the true solution. To handle large-scale problems with thousands of cameras and landmarks, XM employs Burer-Monteiro factorization to exploit the low-rank structure of the solution, coupled with GPU-accelerated Riemannian optimization.

This convex optimization framework offers several advantages over traditional approaches: guaranteed global optimality (when the relaxation is tight), no sensitivity to initialization, and superior scalability to large datasets. However, the quality of the final reconstruction remains dependent on the accuracy of the input depth estimates used to lift 2D observations to 3D.

4 Methodology

This pipeline integrates learned depth correction with the XM convex optimization framework to achieve accurate and efficient reconstruction.

4.1 System Architecture

The reconstruction pipeline comprises the following stages:

Monocular depth estimation: We employ pre-trained monocular depth networks (e.g. MiDaS [2]) to generate initial depth maps for each input image. These networks provide fast, dense depth predictions but suffer from scale ambiguity and systematic biases.

Depth correction module (in development): A learned correction network consumes RGB imagery, raw monocular depth, image gradients, and optional semantic cues to predict depth residuals that correct systematic errors. This stage aims to produce depth estimates suitable for lifting 2D keypoints to accurate 3D observations.

Feature extraction and matching: COLMAP [1] and GLOMAP performs SIFT feature extraction and exhaustive matching to establish 2D correspondences across views, constructing the view graph necessary for scaled bundle adjustment.

3D observation lifting: Corrected depth maps are used to lift 2D keypoint correspondences to approximate 3D observations in each camera frame. This transformation is central to the XM formulation, as it enables the scaled bundle adjustment problem to be cast as a convex optimization.

XM-based scaled bundle adjustment: The XM solver performs global bundle adjustment using the convex SDP formulation. The solver jointly optimizes camera poses, scaling factors, and 3D landmark positions to achieve certifiably optimal reconstruction. Optional outlier filtering and refinement (XM^2 or Ceres post-processing) can be applied to further improve accuracy.

Dense reconstruction: Following sparse reconstruction from XM, dense MVS can optionally be performed using traditional COLMAP dense stereo [1], now informed by the optimized camera poses and sparse point cloud from the convex optimization stage.

Mesh post-processing: Meshlab handles vertex filtering, hole filling, and non-manifold geometry correction. Blender is used for object-specific editing, particularly base flattening to ensure stable contact surfaces for robotic grasping.

Grasp simulation: The Simple-MuJoCo-PickNPlace [4] repository evaluates reconstruction quality through simulated pick-and-place tasks, measuring grasp success rates and pose stability.

4.2 Learned Depth Correction

We are developing a lightweight convolutional neural network that accepts multi-modal input—RGB imagery, raw monocular depth, image gradients, and semantic segmentation—to predict depth residuals. The network is trained using a composite loss function:

- **Photometric reprojection loss:** Enforces multi-view consistency when calibrated views are available, leveraging the XM-optimized camera poses as supervision.
- **Geometric priors:** Planarity constraints applied to large, low-gradient regions (floors, walls).

- **Edge-aware smoothness:** Preserves depth discontinuities at object boundaries while encouraging smoothness within homogeneous regions.
- **Confidence weighting:** Prioritizes accuracy in near-field regions where depth estimation is most reliable and critical for manipulation tasks.

The objective is to produce corrected depth estimates that improve the quality of 3D observations input to the XM solver, thereby reducing residual reprojection errors and accelerating convergence to the global optimum.

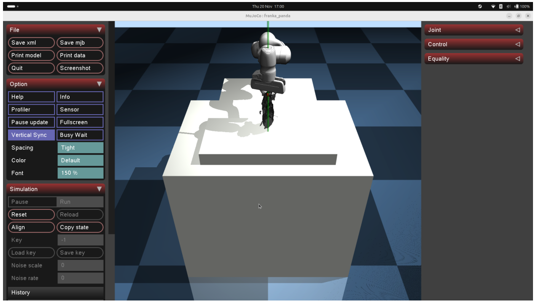
4.3 Implementation Details

Software stack: Python 3.12, XM solver (custom CUDA implementation), COLMAP/GLOMAP (feature matching and view graph construction), Meshlab (mesh cleaning), Blender (base editing), MuJoCo (grasp simulation).

Processing workflow:

1. Generate monocular depth maps for all input images using pre-trained depth networks.
2. Apply learned depth correction (when available) to produce refined depth estimates.
3. Execute feature extraction and matching via COLMAP and GLOMAP to construct the view graph.
4. Lift 2D correspondences to 3D observations using corrected depth maps.
5. Run XM solver for scaled bundle adjustment with convex optimization.
6. Perform outlier filtering and refinement (XM^2 or Ceres).
7. Execute dense MVS reconstruction using optimized camera poses (optional).
8. Import dense point clouds or meshes into Meshlab for manual vertex filtering and hole filling.
9. Apply manual refinements in Blender (primarily base flattening for grasp stability).
10. Export meshes (STL format) and append to MuJoCo-compatible XML scene descriptions.
11. Evaluate reconstruction quality through simulated pick-and-place scenarios (`pnp.py`).

Reconstructed mesh in MuJoCo being manipulated by a Franka Panda robot.



Spacing Tight
Color Default
Font 150 %
Key -1
Noise scale 0
Noise rate 0

5 Challenges and Technical Observations

5.1 Domain Adaptation

Monocular depth networks trained on mixed indoor-outdoor datasets exhibit domain-specific compression characteristics at distance. Correcting these errors requires either explicit domain conditioning or targeted dataset augmentation strategies.

5.2 Scale Ambiguity

Aligning the scale of monocular depth predictions with metric COLMAP reconstructions necessitates external scale priors. We employed known object dimensions or camera baseline measurements when available; developing automatic scale inference remains an open challenge.

5.3 MVS Output Quality

Dense MVS reconstruction frequently produces noisy point clouds with isolated vertices and non-manifold geometry. While Meshlab and Blender provide effective manual cleanup tools, automating this process will prove to be brittle and sensitive to scene-specific artifacts (such as reflections from metallic and reflective surfaces).

5.4 Computational Cost

Despite my motivation to reduce reconstruction time, current COLMAP-based dense MVS remains computationally expensive. The promise of learned monocular priors is to constrain MVS search ranges, but quantitative runtime comparisons await completion of the correction network. The original code used in XM [3] uses UniDepth which is computationally expensive and hence cannot be run on my local system so I have replaced it with MiDaS.

6 Preliminary Results

6.1 Completely Closed Mesh Construction

The mesh generated is watertight. This is done by filling in holes using Meshlab and Blender.

6.2 Grasp Simulation Fidelity

Post-processed meshes following our pipeline are sufficient for MuJoCo-based grasp evaluation. Grasp success rates exhibit sensitivity to mesh watertightness and contact surface geometry, confirming that base flattening and hole filling are critical preprocessing steps.

7 Recommendations and Future Work

1. **Complete learning pipeline:** Finalize training of the residual correction network with supervised multi-view losses and integrate an explicit scale inference module.

2. **Automate mesh cleanup:** Develop scripted Blender operations for hole filling and base flattening to eliminate manual intervention.

3. **COLMAP integration:** Incorporate corrected depth maps as per-image priors directly into COLMAP's dense fusion stage to constrain matching search ranges.

4. **Quantitative evaluation:** Establish metrics for reconstruction time, point cloud density, mesh intersection-over-union (IoU) against ground truth, and simulated grasp success rates.

5. **Publication in Peer-Reviewed Conferences:** Complete learning pipeline and attempt to have this work published in any international proceedings, preferably in a conference.

8 Conclusion

We have demonstrated MVS-based 3D reconstruction for robotic manipulation. While the learning component remains under development, our integrated pipeline successfully combines monocular depth estimation, traditional MVS, mesh post-processing, and grasp simulation. Preliminary findings and readings suggest that corrected monocular depth can serve as an effective prior for constraining MVS search, though quantitative validation and automation of manual steps remain priorities for future work.

References

[1] Schönberger, J. L., & Frahm, J.-M. (2016). *Structure-from-Motion Revisited.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[2] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer.* IEEE Transactions on Pattern Analysis and Machine Intelligence.

[3] Building Rome with Convex Optimization. https://computationalrobotics.seas.harvard.edu/XM/ XM-Code repository. https://github.com/ComputationalRobotics/XM-code

[4] Simple-MuJoCo-PickNPlace repository. https://github.com/volunt4s/mujoManipulation.git

[5] Levenberg-Marquardt Algorithm. https://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm