Towards Clean 3D Scene Reconstruction by Filtering Dynamic Objects

Omkar Sargar
Northeastern University
Boston, MA
sargar.o@northeastern.edu

Jitesh Sonkusare
Northeastern University
Boston, MA
sonkusare.j@northeastern.edu

Srimathnath Thejasvi Vondivillu
Northeastern University
Boston, MA
vondivillu.s@northeastern.edu

Haoyu Li
Northeastern University
Boston, MA
li.haoyu2@northeastern.edu

Abstract

Dynamic objects in video sequences, such as moving vehicles or pedestrians, cause mismatches in point correspondence, leading to artifacts and degraded quality in 3D scene reconstructions. This project addresses the challenge by proposing a pipeline to detect and remove dynamic objects from video sequences, yielding cleaner input data for reconstruction algorithms like Neural Radiance Fields (NeRF) and Gaussian Splatting. This report details our methodology, implementation, results, and future directions.

1. Introduction

Dynamic objects pose significant hurdles in the field of 3D scene reconstruction due to their non-static behavior, which introduces substantial noise and inconsistencies into algorithms that rely on spatial coherence. These inconsistencies often manifest as errors in the mapping of spatial correspondences, causing artifacts and a degradation of overall scene quality in the reconstructed outputs. For instance, moving objects such as pedestrians or vehicles can distort the point cloud data or render photometric calculations inaccurate, making it challenging to produce reliable reconstructions.

To address this pressing challenge, the primary aim of this project is to develop a comprehensive framework for detecting and removing dynamic objects from video sequences. By systematically isolating and eliminating these noise-inducing elements, the project aims to create a "cleaned" dataset that serves as a robust foundation for advanced 3D reconstruction pipelines. This cleaned input data ensures that reconstruction algorithms, which often assume static environments, can operate under more controlled conditions, thereby improving their accuracy and reliability.

This study also evaluates the broader implications of state-of-the-art 3D reconstruction approaches in real-world scenarios, highlighting their resilience to residual noise and computational efficiency. These findings not only underscore the importance of dynamic object removal but also illuminate the strengths and limitations of modern reconstruction techniques, providing a comprehensive understanding of their applicability in diverse settings such as robotics, urban planning, and augmented reality.

2. Related Work

Techniques like NeRF [8] excel at high-quality photorealistic reconstructions but are computationally expensive and sensitive to noisy data. Gaussian Splatting [5] offers a robust alternative, representing scenes as sparse sets of 3D Gaussians. Prior work has explored segmentation and optical flow-based methods for dynamic object detection but often lacks integration with modern 3D reconstruction frameworks.

3D scene reconstruction from video frames has become increasingly sophisticated, with techniques like NeRF achieving impressive photo realism. However, the presence of dynamic objects in videos introduces significant challenges. These objects cause mismatches in point correspondences, leading to artifacts and degraded quality in reconstructions. Several prior works have addressed dynamic object removal for 3D scene reconstruction:

- **Segmentation-based methods** - These methods utilize deep learning models like DeepLab [1] or U-Net [10] to segment objects in each frame. These methods can effectively identify state and dynamic regions, but their accuracy can be very limited by complex objects, shapes, occlusions, and challenging lighting conditions.
- **Optical flow techniques** - These estimate motion between consecutive frames, often relying on algorithms

like FlowNet [3]. While these methods capture object motion, they can struggle to differentiate object movement from camera motion. Inaccurate camera motion estimation can lead to incomplete masking of dynamic objects which is like the most common issue in this approach.

- **Background subtraction techniques** - In this type of method we identify foreground objects by analyzing differences between frames. These methods, often based on statistical models, can be computationally efficient but are sensitive to lighting changes.

Existing work majorly struggles with camera motion estimation. This can lead to incomplete dynamic object removal and negatively impact 3D reconstruction quality. Our Proposed methods addresses these limitations by:

- **Combining segmentation and optical flow:** We leverage the strengths of both approaches. Segmentation identifies objects, and optical flow captures their motion.
- **Advanced camera motion estimation:** We employ techniques like Scale-Invariant Feature Transform (SIFT) [7] and RANSAC [4] to accurately separate camera motion from object motion, leading to more precise dynamic object masking.
- **Classical Computer Vision Techniques:** In addition to the above, we explore a simpler approach using SIFT, Fast Library for Approximate Nearest Neighbors (FLANN) [9], edge detection, and Structural Similarity Index (SSIM) [12] for dynamic object detection. This offers an efficient alternative for applications requiring real-time processing.

By combining these techniques, our approach aims to achieve more robust and accurate dynamic object removal, ultimately leading to higher-quality 3D scene reconstructions.

3. Methods

The method we devised to address the problem of clean 3D scene reconstruction is divided into two major steps. The first step involves detecting and removing dynamic objects from video sequences, while the second step focuses on leveraging these cleaned sequences to perform high-quality 3D reconstruction. Each of these steps was explored through distinct approaches, and this section provides a detailed theoretical foundation and description of the implementation of these methodologies.

3.1. Dynamic Object Detection and Removal

Dynamic object detection and removal form the crux of the first step in our method. To ensure robustness and effectiveness, we implemented and evaluated two separate approaches for this step: a segmentation and optical flow-based approach and a classical computer vision-based approach.

Figure 1. **Input Image:** The original image frame with a person and cellphone detected and segmented using YOLOv7. **Person Mask:** A binary mask is generated by removing the person class from the segmentation output. **Masked Image:** The original image with the person region masked out using the generated mask. **Optical Flow:** The optical flow is computed on the masked image using the RAFT algorithm, highlighting the residual motion in the scene.

3.1.1. Segmentation and Optical Flow

The segmentation and optical flow approach begins by partitioning each video frame into meaningful regions using advanced segmentation models like SAM [6] or YOLOv7 [11]. These models are adept at identifying objects and delineating them within each frame. Once segmented, optical flow is computed for each segment using the RAFT [13] algorithm, which provides a dense representation of motion between consecutive frames, as illustrated in Figure 1. Optical flow is particularly advantageous for capturing the relative motion of objects in a scene, but it also includes motion due to the camera itself, as illustrated in Figure 4.

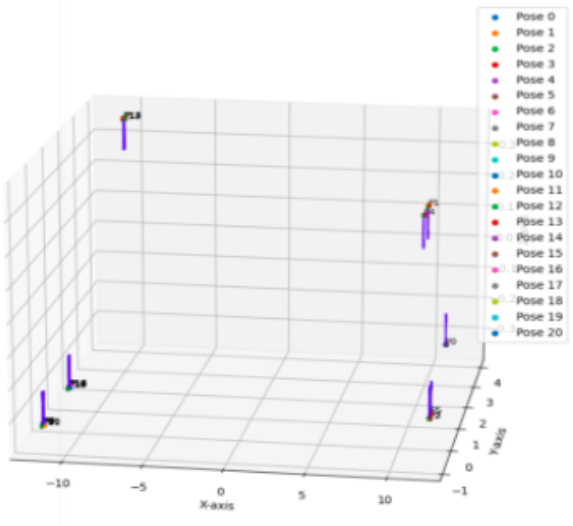


Figure 2. Figure showing refined camera poses after Bundle Adjustment using GTSAM.

We employed the SIFT algorithm to detect robust feature points across consecutive frames. To efficiently match these feature points, we utilized the FLANN algorithm. This ac-

Figure 3. The image demonstrates the output of applying the SIFT algorithm to detect robust feature points between consecutive frames. These detected features are then matched using the FLANN, which efficiently finds corresponding points between the frames. The matches shown in this figure are just the inliers that are obtained after outlier rejection using RANSAC.

celerated the matching process and improved the accuracy of feature correspondence. Subsequently, the RANSAC algorithm was employed to identify and eliminate outlier matches, ensuring the reliability of the estimated camera motion, as illustrated in Figure 3. With the inlier matches, we estimated the fundamental and essential matrices, which proved constraints on the relative camera motion between frames. 3D points were then triangulated from the inlier matches and camera poses.

To further refine the camera poses and 3D point cloud, we utilized the GTSAM [2] optimization framework for Bundle Adjustment. The refined camera poses after applying bundle adjustment can be seen in Figure 2. This iterative optimization process minimizes the reprojection error of 3D points onto the image plane, resulting in a more accurate and consistent representation of the scene geometry. By subtracting the refined camera motion from the overall optical flow, we effectively isolated the movement of dynamic objects, enabling a more accurate reconstruction of the static scene.

3.1.2. Classical Computer Vision-Based Approach

The classical computer vision-based approach relies on traditional image processing techniques, emphasizing efficiency and simplicity. Initially, feature detection and matching are performed using SIFT and FLANN respectively to establish correspondences across frames. Motion detection is then conducted by applying edge detection algo-

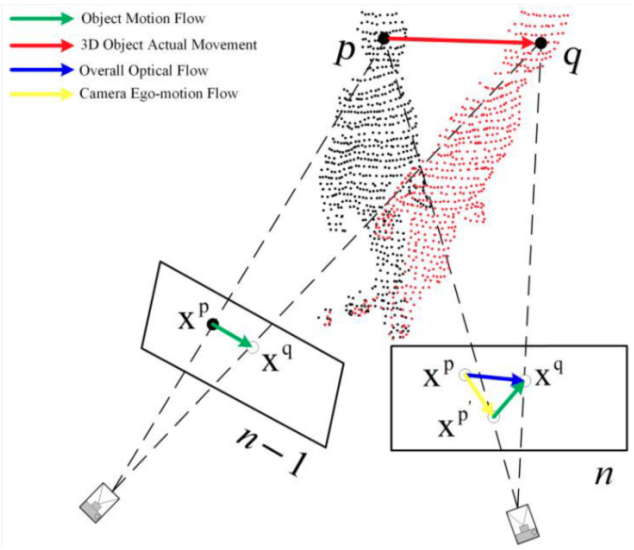


Figure 4. Camera flow removal from the overall optical flow.

rithms such as Canny or Sobel, which highlight regions of change. To improve accuracy, the SSIM is calculated between frames, pinpointing areas where significant alterations have occurred.

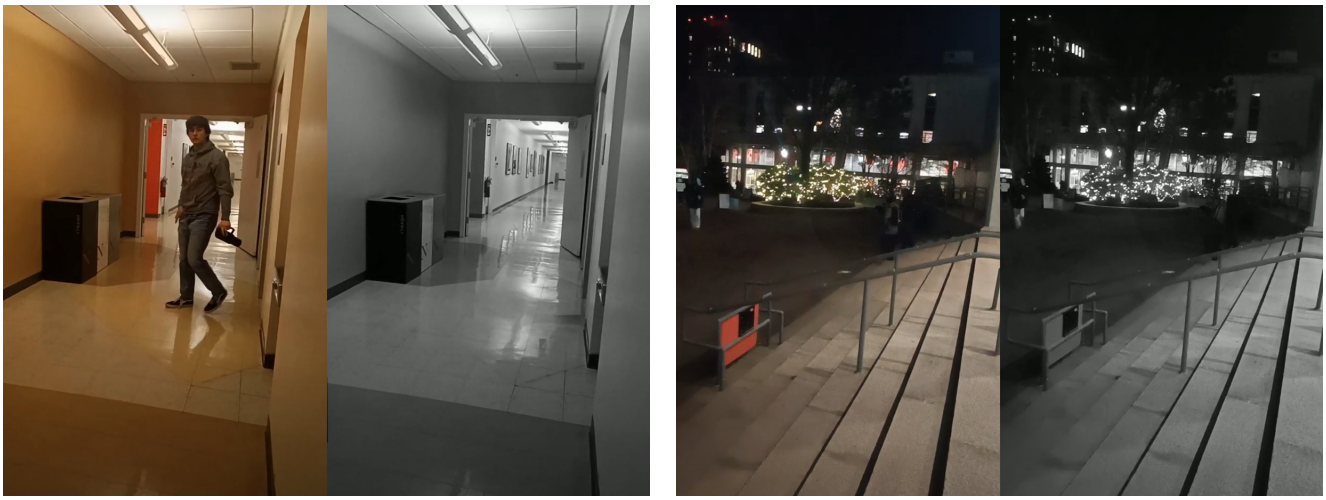
To further refine these detections, additional filtering steps, including morphological operations and motion blur analysis, are applied. Morphological operations help eliminate noise and enhance the contours of detected regions, while motion blur analysis identifies objects that exhibit elongated streaks due to motion. Bounding rectangles are subsequently drawn around the identified dynamic areas, encapsulating these regions for processing. Finally, inpainting techniques are used to fill the areas from which dynamic objects have been removed, ensuring that the visual and structural integrity of the video frames is preserved.

3.2. 3D Reconstruction

With dynamic objects removed, the cleaned video sequences form the basis for the second step, which involves 3D reconstruction. To explore the efficacy of modern reconstruction methods, we implemented two cutting-edge techniques: Neural Radiance Fields (NeRF) and Gaussian Splatting.

3.2.1. Neural Radiance Fields (NeRF)

NeRF represents the scene as a continuous volumetric function, modeled by a neural network. It maps 3D spatial coordinates and viewing directions to their corresponding colors and densities. By rendering the scene through volumetric integration, NeRF produces high-quality photorealistic images. However, its computational demands are significant, and it is highly sensitive to noise and imperfections in the input data. Any residual artifacts from dynamic object re-



(a) Indoor well-lit environment.

(b) Outdoor low-lit environment.

Figure 5. Raw video frames along with the output frame after dynamic object removal.

moval can lead to noticeable degradation in the reconstruction quality.

3.2.2. Gaussian Splatting

Gaussian Splatting, in contrast, models the scene as a sparse collection of 3D Gaussians, each defined by parameters such as position, shape, color, and opacity. This representation is computationally efficient and robust to noise, making it particularly well-suited for datasets that contain minor residual artifacts. Gaussian Splatting achieves this robustness by leveraging the probabilistic nature of its representation, which smooths over imperfections in the data. Furthermore, its flexibility allows for efficient rendering in real-time or near-real-time scenarios, making it an excellent choice for applications requiring high-speed processing.

Both approaches were rigorously evaluated to understand their strengths and limitations in achieving clean and accurate 3D reconstructions. NeRF excels in visual fidelity under ideal conditions, while Gaussian Splatting demonstrates superior robustness and efficiency, particularly in handling cleaned datasets with residual noise.

3.3. Implementation

The pipeline implementation combines open-source libraries and frameworks to ensure an efficient and modular design for dynamic object removal and 3D reconstruction. The project was developed in Python, utilizing state-of-the-art tools.

Dynamic object detection employed segmentation models such as SAM and YOLOv7 and optical flow computation using RAFT, with camera motion-compensated using SIFT and RANSAC. For 3D reconstruction, Neural Radiance Fields (NeRF) and Gaussian Splatting were implemented using PyTorch. NeRF focused on photorealistic scene generation, while Gaussian Splatting offered computational efficiency and robustness.

The hardware setup included NVIDIA RTX 4080 GPU and an i9 14th-gen CPU, leveraging acceleration for computationally intensive tasks. Comprehensive implementation details, including preprocessing and training scripts, are available in the repository. This streamlined design ensures flexibility and adaptability for diverse datasets and scenarios.

4. Results

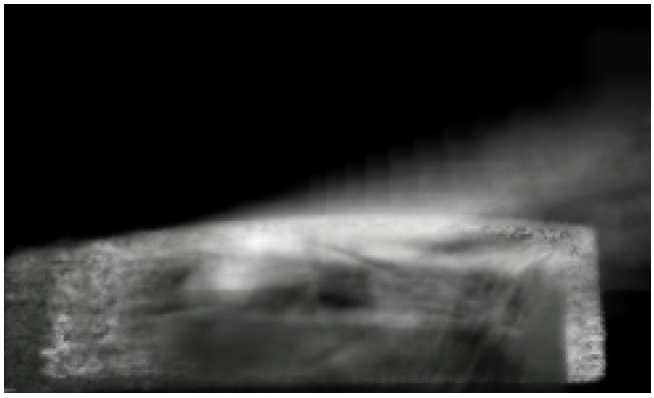
The evaluation of our pipeline was conducted on two distinct video datasets: one recorded indoors in a university hallway with pedestrians walking during the day, and the other captured outdoors at night near the university library, Figure 5. These datasets were selected to test the pipeline's performance under diverse conditions, including varying lighting and dynamic object scenarios.

4.1. Dynamic Object Removal

The first approach, which combined segmentation and optical flow, did not perform well in differentiating between dynamic objects and the static background. The main limitation was the inaccurate estimation of camera optical flow, which resulted in improper masking of dynamic objects.

The second approach, based on classical computer vision techniques, performed comparatively better. While it effectively removed most dynamic objects as seen in Figure 5a, some frames exhibited partial remnants of dynamic entities as seen in Figure 5b. Despite these imperfections, this approach produced cleaner inputs for subsequent 3D reconstruction.

(a) NeRF reconstruction of the indoor well-lit environment.



(b) NeRF reconstruction of the outdoor low-lit environment.

(c) Gaussian Splat of the indoor well-lit environment.

(d) Gaussian Splat of the outdoor low-lit environment.

Figure 6. 3D reconstruction results.

4.2. 3D Reconstruction

For 3D reconstruction, the effectiveness of NeRF and Gaussian Splatting was evaluated both with and without dynamic object removal. NeRF failed to produce usable outputs when applied to video sequences with dynamic objects, highlighting its sensitivity to noise. After dynamic object removal, NeRF was able to generate reconstructions, though the quality remained suboptimal as seen in Figures 6a and 6b.

Gaussian Splatting exhibited better robustness. While it sometimes failed with unprocessed videos containing dynamic objects, it consistently delivered photorealistic results when used with cleaned video sequences. Moreover, Gaussian Splatting demonstrated resilience to residual noise, handling frames with incomplete dynamic object removal far better than NeRF as evident in Figures 6c and 6d.

5. Future Work

Building on the limitations and observations from this study, the following directions are proposed for future research:

- Advanced Camera Motion Estimation: Explore existing computer vision literature to identify more effective techniques for accurate camera motion estimation. This would address the key limitation of the segmentation and optical flow-based approach.
- Enhanced Inpainting Techniques: Investigate generative models and machine learning-based methods for more seamless and realistic inpainting of removed dynamic objects.

- Real-Time Optimization: Optimize the pipeline for real-time applications, ensuring scalability for use in dynamic environments where immediate feedback is necessary.
- Broader Dataset Evaluation: Extend testing to larger and more diverse datasets to validate the pipeline's robustness and generalizability.

6. Conclusion

Filtering dynamic objects from video sequences significantly enhances the quality of 3D scene reconstruction. NeRF, despite its visual fidelity, remains sensitive to noise and computationally demanding. Gaussian Splatting, with its robustness and efficiency, emerges as a practical alternative for real-world applications. Advancements in dynamic object detection and inpainting hold the potential to further elevate reconstruction capabilities, enabling broader adoption across various domains such as robotics, augmented reality, and urban planning.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 1

[2] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. 2012. 3

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 2

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 2

[7] Tony Lindeberg. *Scale Invariant Feature Transform*. 2012. 2

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1

[9] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009. 2

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[11] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. 2

[12] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 2

[13] Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. Raft: Adapting language model to domain specific rag, 2024. 2

Towards Clean 3D Scene Reconstruction by Filtering Dynamic Objects

Supplementary Material

7. Presentation and Github Link

- **Presentation: -** Presentation Link
- **Github: -** Github Link
- **3D Reconstruction Video: -** Gaussian Splatting Link
- **Optical Flow 1 Video: -** Optical Flow Indoor
- **Optical Flow 2 Video: -** Optical Flow Outdoor