

*Робастный SLAM с 3D Gaussian Splatting для динамических сцен:
открытые проблемы и гипотезы*

Я.С. Дуль

Санкт-Петербург

2026

СОДЕРЖАНИЕ

1	Обзор	3
2	Существующие проблемы SLAM в динамических средах	4
2.1	Статические алгоритмы	4
2.2	Anti-dynamic алгоритмы	4
3	Формулировка проблемы и эксперимент	6
3.1	Проблема	6
3.2	Экспериментальная проверка влияния низкого числа повторных наблюдений	6
4	Предлагаемая гипотеза	10
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

1 Обзор

С развитием автономной навигации и компьютерного зрения задача одновременной локализации и картографирования (SLAM) вошла в ряд фундаментальных задач робототехники. Цель алгоритма SLAM — достаточно точно предсказать положение камеры и построить карту неизвестной окружающей среды.

В последние несколько лет активно развивается направление SLAM с использованием технологии 3D Gaussian Splatting (3DGS) [1], которое позволяет получать явное представление карты и отличается высоким качеством выходного рендера. Большинство современных алгоритмов, работающих с последовательностью входных RGB(-D) кадров, например, SplatAM [2], MonoGS [3] и др., показывают отличные результаты на статических сценах. Однако, в динамических средах, где положение объектов непостоянно, появляются серьезные проблемы: дрейф оценки положения камеры (трекинга) и низкое качество реконструкции.

Озадачившись данным вопросом, исследователи разработали множество решений для наделения SLAM робастностью к динамике. Можно выделить два основных подхода: 4DGS и anti-dynamic. 4DGS-подход подразумевает моделирование динамики при ее обнаружении. Некоторые из представителей — DynaGSLAM [4] и 4DGS-SLAM [5]. Anti-dynamic подход отличается тем, что он находит динамические объекты и вырезает их из конечного представления сцены. Это особенно полезно в задачах, где динамические объекты не представляют полезной информации, а только мешают качественному SLAM в статической среде. Одним из ярких представителей такого подхода выступает WildGS-SLAM [6], разработанный исследователями из Стэнфордского Университета и Швейцарской высшей технической школы Цюриха.

В данной работе предлагается рассмотреть существующие проблемы и варианты их решения, в особенности для 3DGS алгоритмов SLAM, использующих подход избавления от динамики.

2 Существующие проблемы SLAM в динамических средах

2.1 Статические алгоритмы

С помощью анализа существующих решений и их результатов можно выделить их основные слабые стороны.

Рассмотрение базовых 3DGS-SLAM алгоритмов в динамической среде носит информативный характер, поскольку они изначально предполагают статичность сцены и не предлагают инструментов борьбы с динамикой.

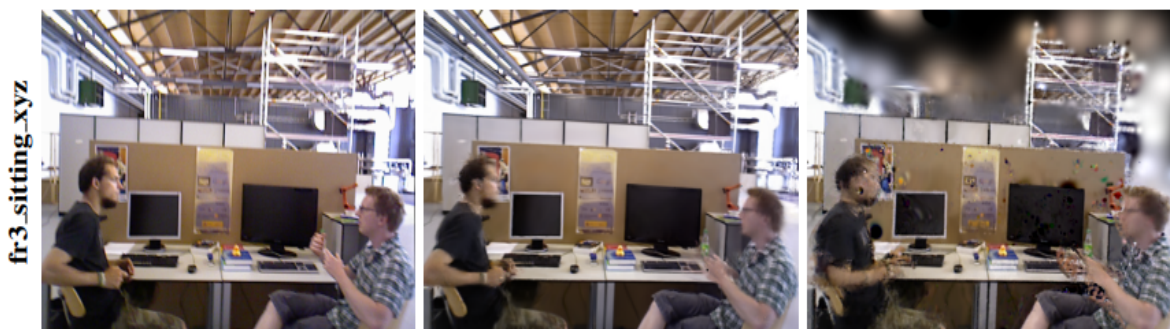


Рисунок 2.1 — Визуальное сравнение качества рендера MonoGS и SplatAM на датасете TUM fr3_sitting_xyz [5]

На рис. 2.1 видно, что даже на датасете с низким количеством динамики и отсутствием резких изменений среды алгоритмы деградируют: в области движущихся объектов появляются артефакты и «призраки», а ошибки локализации накапливаются. С ростом доли динамических пикселей эта деградация усиливается.

2.2 Anti-dynamic алгоритмы

В отличие от статических 3DGS-SLAM систем, динамические решения учитывают присутствие движущихся объектов. anti-dynamic алгоритмы не пытаются промоделировать динамику, а стремятся исключить её влияние на оценку позы и обновление карты. Механизм выглядит как поиск и подавление “помех” (distractors) в изображении.

В стандартных алгоритмах, Когда в оптимизацию попадают пиксели, несущие информацию о динамических объектах, SLAM разрушается: траектория разрывается, появляется дрейф, карта деградирует. Мотивация

данного класса алгоритмов — снизить влияние динамических областей на функцию потерь в трекинге и исключение их из маппинга.

Обычно рассматривают два пути. Первый — использование сегментации для выбора потенциально подвижных классов. Второй — обучение предсказателя неопределенности (uncertainty) на-лету: модель оценивает доверие к каждому пикселю, и затем этот сигнал используется для взвешивания фотометрического лосса и ограничения обновления гауссиан. Данный механизм позволяет подавлять не только очевидную динамику (люди, машины), но и отражения, экраны, движущиеся детали, которые сложно описать фиксированными правилами. Так работает и один из новейших и прорывных инструментов WildGS-SLAM. Ключевой механизм алгоритма — карта неопределенности (uncertainty map), которая хранит информацию о том, каким пикселям можно доверять при трекинге и маппинге. Карта получается таким образом: DINOv2-признаки проходят через небольшой MLP-декодер, который предсказывает неопределенность по пикселям.

Однако у anti-dynamic подходов есть слабость. Более агрессивное подавление динамики повышает робастность к выбросам, но в то же время оно уменьшает число информативных пикселей, участвующих в оценке позы. Это может приводить к срывам трекинга, особенно если значительная часть сцены постоянно закрыта движущимися объектами. Также методы, которые обучаются на-лету, чувствительны к числу повторных наблюдений одних и тех же областей. Если ракурсов не хватает, алгоритму становится трудно накопить знания, чтобы отличить статические объекты от динамических помех. По этой причине даже такие робастные алгоритмы могут демонстрировать деградацию качества локализации при высокой динамике и низком уровне покрытия сцены входными кадрами.

3 Формулировка проблемы и эксперимент

3.1 Проблема

Одним из основных ограничений anti-dynamic подходов в 3DGS-SLAM является зависимость робастности трекинга от наблюдаемости статической сцены. В частности, в WildGS-SLAM предсказатель неопределенности (uncertainty) обучается онлайн на входных кадрах. Это усложняет распознавание движущихся объектов и других несогласованных областей в условиях, когда ограниченное число ракурсов покрывает одни и те же области сцены. В таком режиме модель получает недостаточно повторных наблюдений, чтобы надёжно отделить динамические компоненты от статических. В результате появляется снижение устойчивости локализации: либо динамические пиксели недостаточно подавляются и начинают смещать оценку позы камеры, либо подавление оказывается чрезмерным и в оптимизацию попадает слишком мало информативных пикселей, ухудшая результаты задачи трекинга.

3.2 Экспериментальная проверка влияния низкого числа повторных наблюдений

Цель эксперимента. Проверить, приводит ли снижение overlap к деградации качества локализации в WildGS-SLAM.

Идея. Чтобы уменьшить overlap, не меняя геометрию сцены, входная последовательность была прорежена по времени: из исходного датасета оставлялся каждый k -й кадр, где $k \in \{1, 2, 3\}$. Такое преобразование уменьшает плотность наблюдений и число повторных проекций одних и тех же областей.

Для эксперимента была арендована виртуальная машина на платформе Intel Ice Lake with T4i GPU в облачном сервисе Yandex Cloud. Далее был клонирован репозиторий WildGS-SLAM. После сборки производились запуски: тестовый запуск демо и 3 экспериментальных запуска на датасете Wild-SLAM Mocap Dataset Umbrella с разным коэффициентом прореживания входных кадров. Датасет снят с высокой частотой кадрирования, в

нем мало резкой динамики и перемещений камеры, поэтому прореживание в данном случае оправдано.

Процедура.

а) Для эксперимента была арендована виртуальная машина на платформе Intel Ice Lake with T4i GPU в облачном сервисе Yandex Cloud.

б) Производилась сборка всего проекта.

в) Были сформированы три версии датасета:

- $k = 1$: исходная последовательность (без прореживания);
- $k = 2$: оставлен каждый второй кадр;
- $k = 3$: оставлен каждый третий кадр.

г) Для каждой версии датасета выполнен прогон WildGS-SLAM с одинаковыми параметрами.

д) производилась оценка качества по статистикам абсолютной ошибки позы (*Absolute Pose Error, APE*) после сравнения оценённой траектории с ground truth: rmse, mean, median, std, max, scale.

Результат. Итоговые метрики для трёх прогонов приведены в табл. 3.1.

Таблица 3.1 — Статистики APE для разных коэффициентов прореживания k (датасет Umbrella)

k	scale	rmse, м	mean, м	median, м	std, м	max, м
1	0.9054	0.002214	0.002010	0.001826	0.000930	0.005214
2	0.9571	0.002612	0.002174	0.002026	0.001448	0.018347
3	0.9634	0.003040	0.002444	0.002000	0.001809	0.017175

Визуализация результатов.

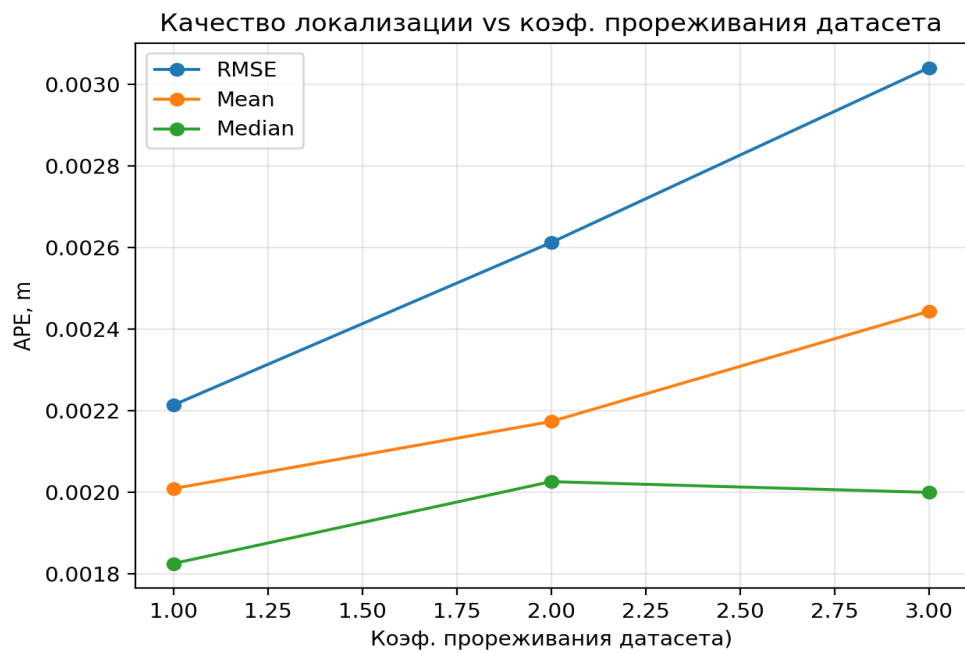


Рисунок 3.1 — Зависимость $\text{rmse}(\text{APE})$, $\text{mean}(\text{APE})$ и $\text{median}(\text{APE})$ от k

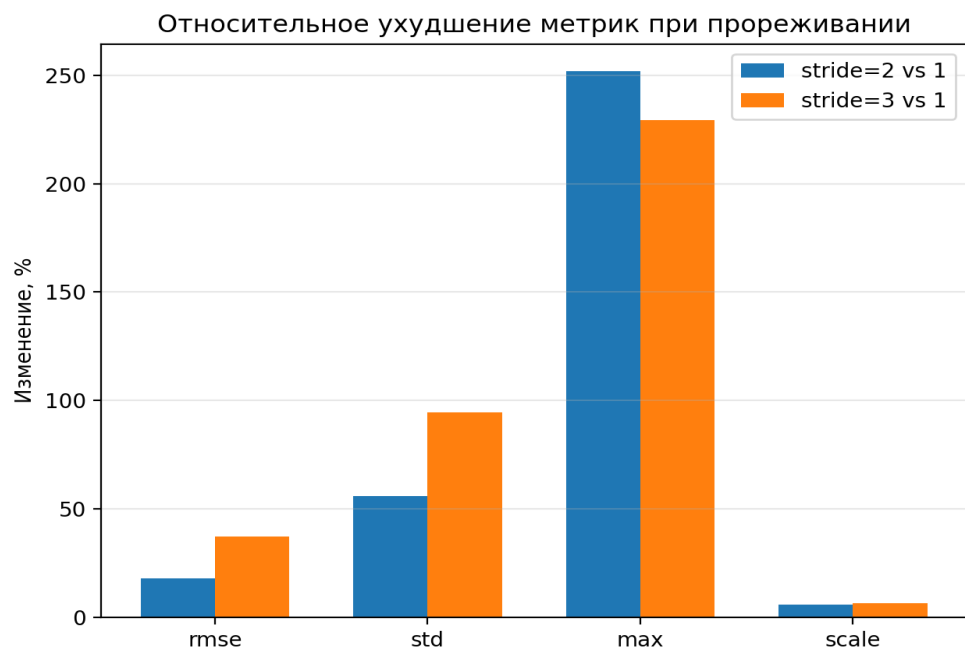


Рисунок 3.2 — Столбчатая диаграмма относительного изменения метрик относительно базового прогона ($k = 1$)

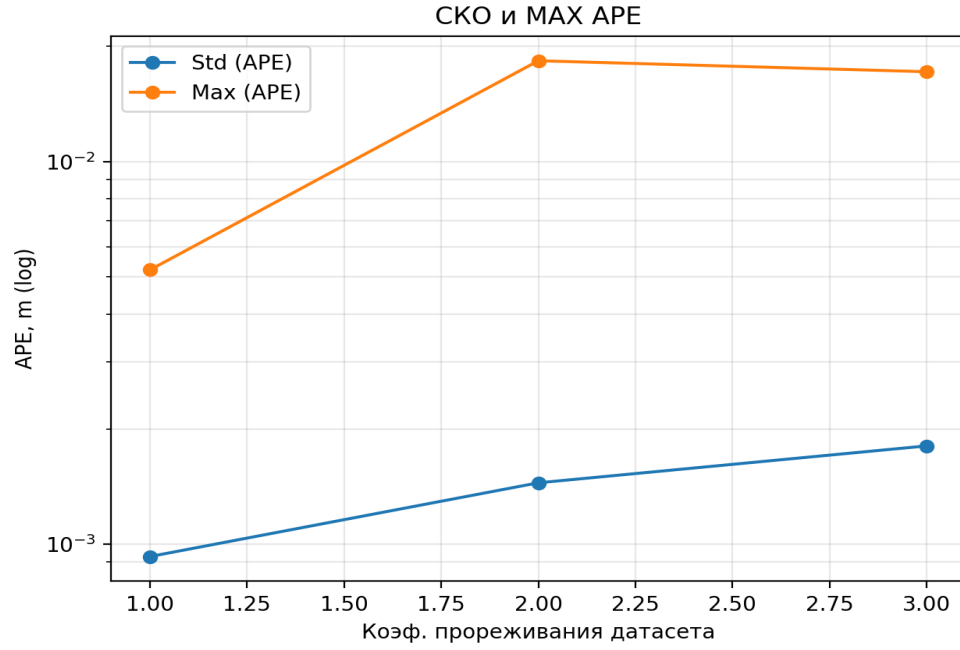


Рисунок 3.3 — Зависимость СКО и $\max(\text{APE})$ от k с логарифмической шкалой по оси Y

Вывод. Полученные результаты подтверждают сформулированную проблему: уменьшение перекрытия наблюдений (прореживание) приводит к снижению робастности к динамике трекинга WildGS-SLAM. В частности, растут разброс и максимальные ошибки APE, что указывает на появление провалов локализации в режиме низкого overlap, что подтверждает сформулированную выше проблему.

4 Предлагаемая гипотеза

Если в систему добавить нейросетевой модуль движения, который по короткому временному окну (2–3 кадра) оценивает пиксельную вероятность принадлежности к движению или шуму на основе несоответствия между наблюдаемым оптическим потоком и потоком, объясняемым текущей оценкой движения камеры, и использовать этот сигнал для взвешивания фотометрических/геометрических ошибок в трекинге и обучения online-предиктора неопределённости, то система будет существенно лучше распознавать дистракторы при малом числе повторных наблюдений области и, как следствие, демонстрировать более устойчивый трекинг и меньшую ошибку траектории в динамических сценах с низким overlap.

Вход. Два соседних кадра, текущая оценка позы, глубина.

Выход. Карта $p_{dyn} \in [0,1]$ — вероятность “динамики”.

Обучение. Нейросетевой модуль движения предполагается предобучать на стандартных наборах для оптического потока/видеосегментации движения, чтобы он давал устойчивый сигнал p_{dyn} уже по двум кадрам и не зависел от повторных наблюдений одной и той же области. Встраивание в SLAM выполняется без явного моделирования динамики: карта p_{dyn} используется для подавления вклада подозрительных пикселей в лоссе трекинга и как априорная подсказка для ускорения online-обучения.

На шаге трекинга итоговый вес пикселя задаётся как

$$w(x) = w_{unc}(x) \cdot (1 - p_{dyn}(x))^\beta, \quad (4.1)$$

где $w_{unc}(x)$ — текущий вес из online-неопределённости, а β регулирует агрессивность подавления.

Это позволяет использовать p_{dyn} как слабую разметку: пиксели, несогласованные с жёстким движением камеры, должны получать повышенную неопределённость уже в первые итерации, что снижает вероятность редких провалов трекинга.

Ожидаемый эффект. В режиме low-overlap, когда кадров участка сцены ограниченное количество, предложенный модуль должен прежде всего уменьшить ошибку локализации, а также в уменьшении частоты редких провалов трекинга, сохраняя anti-dynamic подход.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 3D Gaussian Splatting for Real-Time Radiance Field Rendering / Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis // *ACM Transactions on Graphics*. — 2023. — July. — Vol. 42, no. 4. <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>.
2. SplatTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM / Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula et al. // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2024.
3. Gaussian Splatting SLAM / Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, Andrew J. Davison. — 2024.
4. *Li, Runfa Blark*. DynaGSLAM: Real-Time Gaussian-Splatting SLAM for Online Rendering, Tracking, Motion Predictions of Moving Objects in Dynamic Scenes. — 2025. <https://arxiv.org/abs/2503.11979>.
5. *Li, Yanyan*. 4D Gaussian Splatting SLAM. — 2025. <https://arxiv.org/abs/2503.16710>.
6. WildGS-SLAM: Monocular Gaussian Splatting SLAM in Dynamic Environments / Jianhao Zheng, Zihan Zhu, Valentin Bieri et al. // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2025.