A Neurosymbolic Approach To Adaptive Feature Extraction In SLAM

Authors: Yasra Chandio, Momin Ahmed Khan, Khotso Selialia, Luis Antonio Garcia, Joseph DeGol, and Fatima M. Anwar

Yasra Chandio
University of Massachusetts Amherst

Webpage

Preliminary Analysis: Feature Tracking Performance Across Environmental Conditions

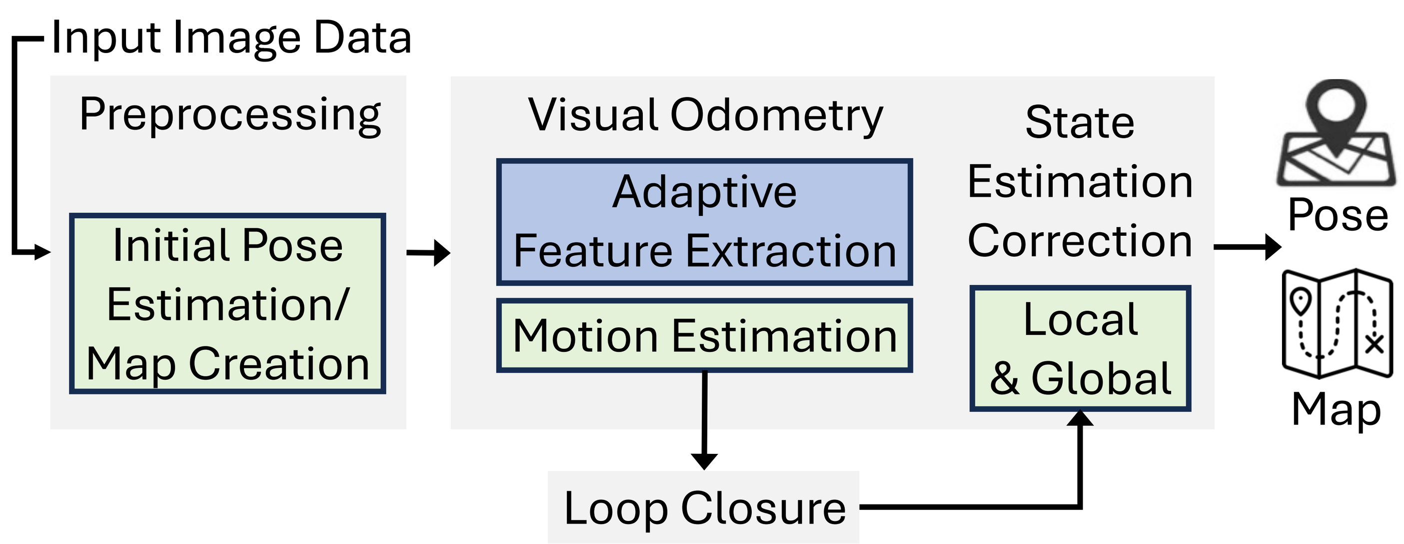
Setup: EuRoC dataset, KTL Sparse Optical Flow (VINS-Fusion) Extractor

Observation: The no. of features tracked depends on the scene and extractor

Idea: Adaptive Feature Extraction in SLAM Pipelines

Traditional SLAM pipeline

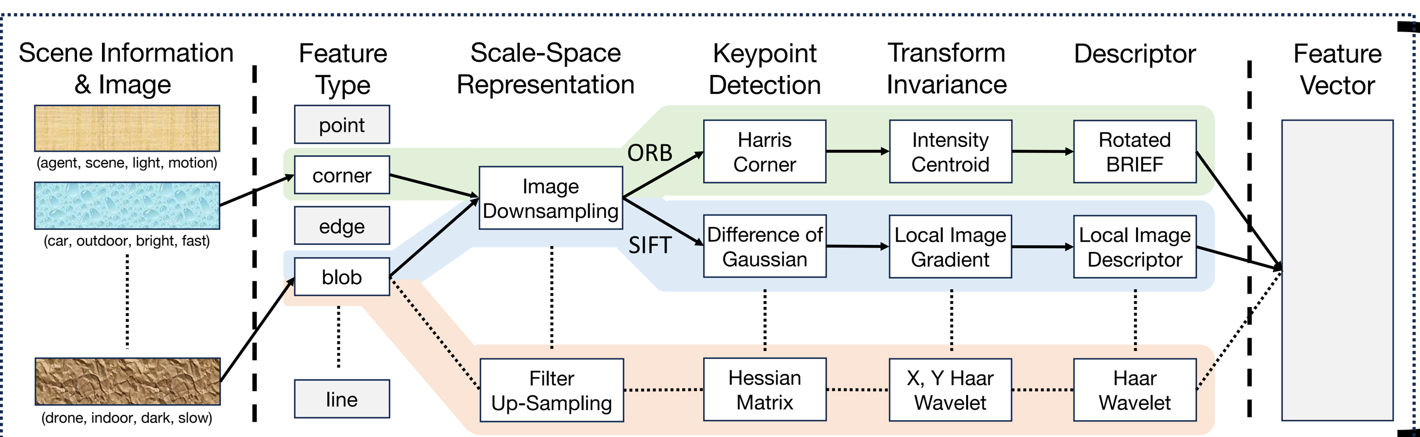
A given feature extractor component detects & tracks features over time, typically with manually tuned parameters.



Key Insight

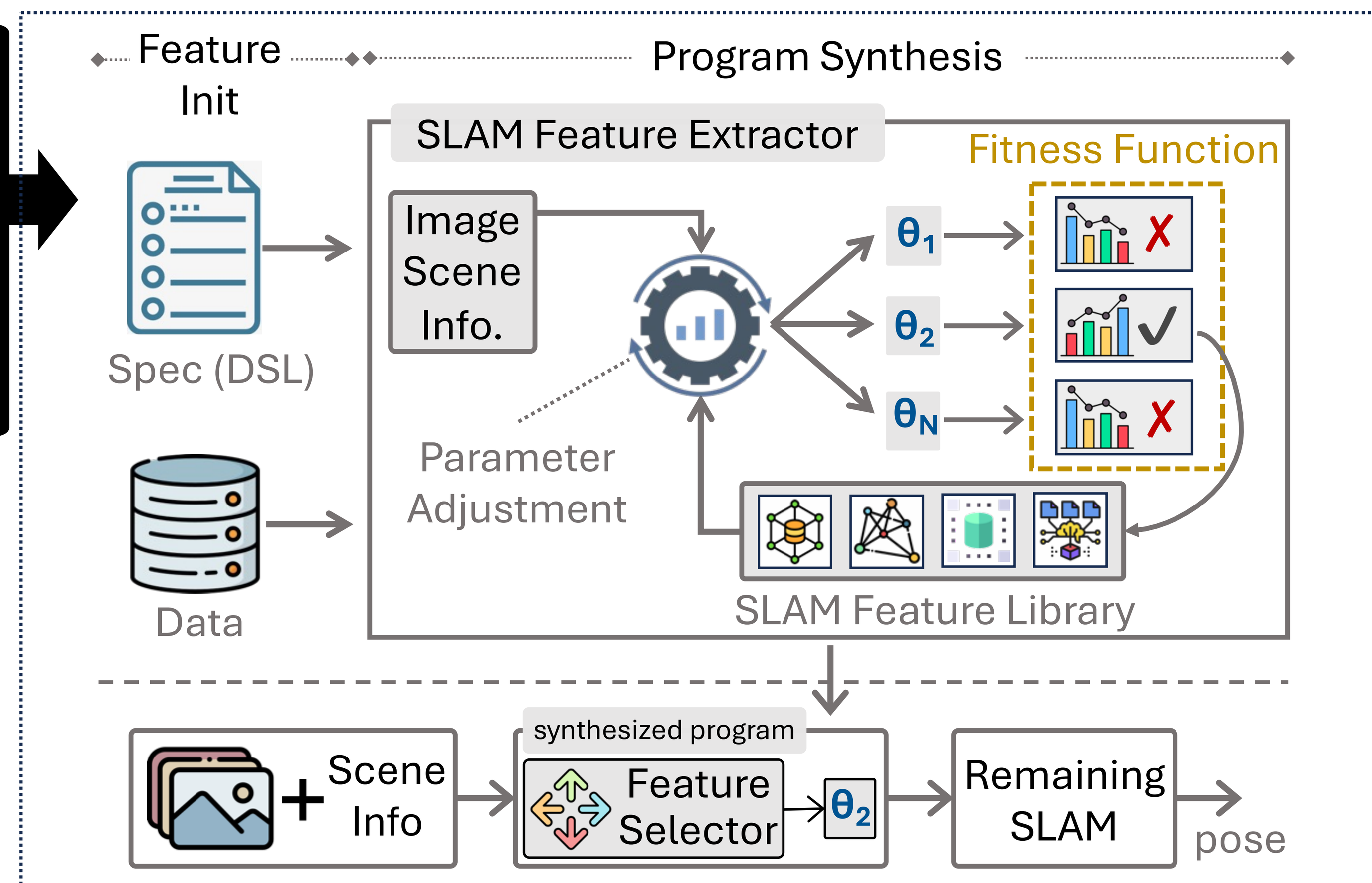
Different feature extractors suit different application scenarios. Dynamically choosing and configuring extractors can improve tracking.

Execution: Leverage Neurosymbolic Program Synthesis



Domain Knowledge Graph

We encapsulate domain insights into the feature extraction module with a knowledge graph that acts as a database of domain information that helps understand and navigate the module's operation and parameters.



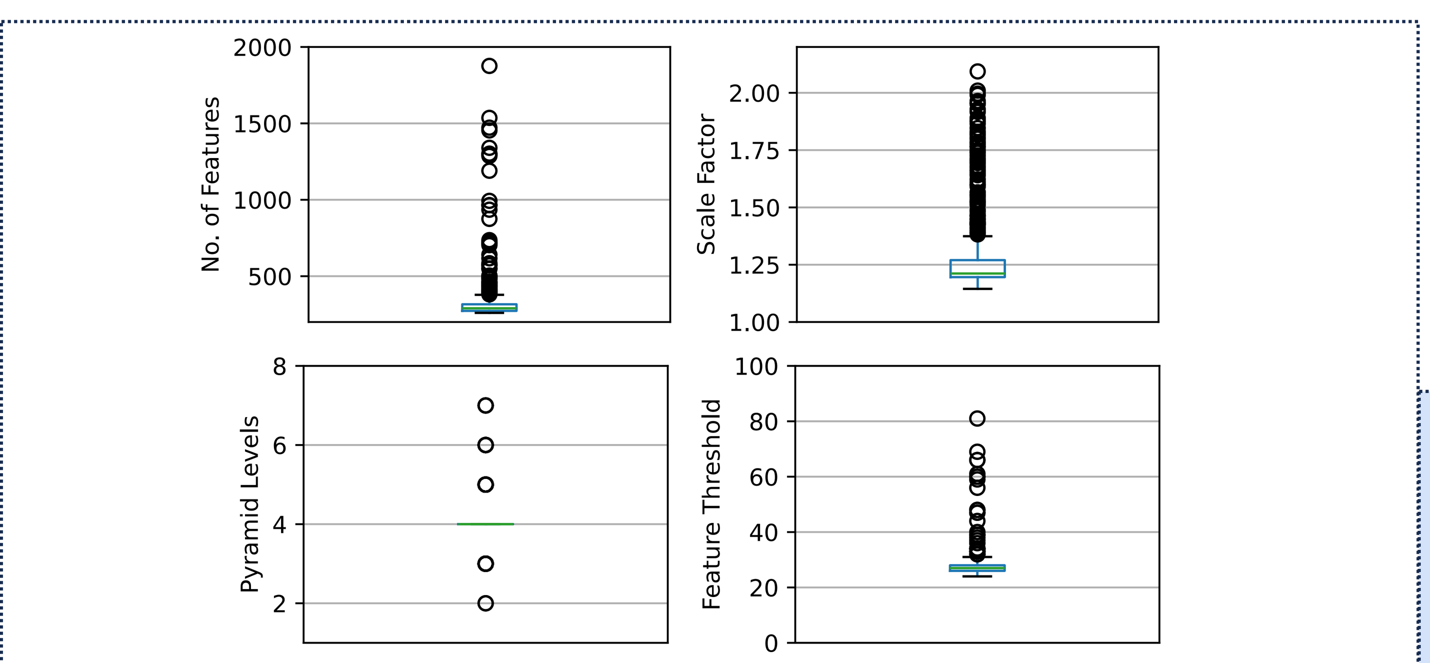
nFEX Overview

An overview of the high-level architecture for our approach, its components, and how it fits into the SLAM pipeline.

```
InputConditions {
    Scene: "Indoor" | "Outdoor";
    Agent: "Car" | "Human" | "Drone";
    LightType: "Bright" | "Dark";
    MotionType: "Fast" | "Slow";
    reflectiveSurface: "Yes" | "No";
    texture: "High" | "Low";
}
FeatureExtractionParameters {
    NF: Int = 1000;
    NL: Int = 4;
    SF: Float = 0.8;
    ST: Float = 6.0;
}
ParameterAdjustment {
    If LightType=="Bright" && MotionType=="Fast":
        If reflectiveSurface=="Yes" && texture=="High":
            NF: 338;
        ElseIf reflectiveSurface=="No" &&
        ↪   texture=="Low":
            NF: 1200;
        Else:
            NF: default;
    ElseIf LightType=="Dark" && MotionType=="Slow":
        NF: 800;
    Else:
        NF: default;
}
FeatureFeatureExtractorSelectionMetrics {
    Metrics:["texturedness", "stability", "motion",
    ↪   "dissimilarity", "spatialDensity",
    ↪   "distinctiveness"];
}
ComputeFeatureExtractorScore {
    InputImage: Image;
    Method: "SomeMethod";
}
FeatureExtractorSelection {
    If ComputeFeatureExtractorScore["SIFT"] >
    ↪   ComputeFeatureExtractorScore["ORB"]:
        Extractor: "SIFT";
    Else:
        Extractor: "ORB"; // default extractor
}
```

Output DSL Program

Example DSL Program based on the knowledge graph and empirical data.



Empirical Data

The distribution of best values on training data for each of the four parameters we configure. Each point is a parameter value that gives the lowest error on a given frame.

Key Findings

Performance Comparison of Different Feature Extractors
(mean absolute trajectory error in meters)

		EuRoC		KITTI		HoloSet	
		MH01	MH05	KITTI-1	KITTI-6	Campus-Center-1	Suburb-Jog-2
ORB	Default	0.855	0.952	2.955	1.173	11.789	11.604
	Dynamic	0.792	0.815	**0.565**	0.126	5.903	5.845
SIFT	Default	0.860	1.038	*fail*	*fail*	13.789	12.825
	Dynamic	0.859	0.882	6.426	5.875	7.049	6.984
nFEX		**0.704**	**0.761**	**0.565**	**0.115**	**4.729**	**5.800**

Evaluation Results

A preliminary implementation of a neurosymbolic adaptive feature extractor (nFEX) outperforms the default ORB and SIFT extractors and their dynamic variants (with parameter adjustment).

Key Takeaway

Leveraging symbolic reasoning and data-driven learning to construct and configure SLAM pipelines improves tracking.