HELLENIC REPUBLIC
MINISTRY OF DEVELOPMENT AND INVESTMENT
GENERAL SECRETARIAT FOR RESEARCH AND INNOVATION
HELLENIC FOUNDATION FOR RESEARCH AND INNOVATION

This project is carried out within the framework of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union – NextGenerationEU (Implementation body: HFRI)

Greece 2.0
Basic Research Financing Action
(Horizontal support of all Sciences)
Sub-action 1
Funding New Researchers

LEARNER

Project Title

SLAM AND PATH PLANNING MIDDLEWARE PACKAGE FOR ROBOTS IN CHALLENGING ENVIRONMENTS

Project Duration

20 November 2023 – 19 November 2025
24 Months

Project Acronym

LEARNER

Project No

015339

Deliverable No

D2.3

Deliverable Title

Journal paper on AI-based environment representation for SLAM

Deliverable Completion Date

19 September 2025

Table of Contents

Document Revision History

Version	Date	Notes
1.0	19/09/2025	First version document describing the two papers submitted regarding the developed robust environment representations within SLAM.

List of Acronyms

Acronym	Meaning
ATE	Absolute Trajectory Error
ICA	Illumination Condition Adaptation
LEARNER	SLAM and Path Planning Middleware Package for Robots in Challenging Environments
PP	Path Planning
PRSI	Photo-Realistic Synthetic Illumination
RL	Reinforcement Learning
RMSE	Root Mean Square Error
SLAM	Simultaneous Localization and Mapping

1. Introduction

This deliverable consolidates the main findings from two key research outputs developed within the context of Task 2.3. These works focus on the data illumination-robust detection of local feature points and the on-the-fly optimization of keypoint selection through deep RL. Together, they provide a major advancement in enabling a robot to build reliable maps and localize itself within them (SLAM), even in challenging environments characterized by poor visibility, dynamic lighting changes, or texture-sparse surfaces.

2. Core Contributions

2.1. Data-Augmented Illumination-Invariant Feature Detection

In the first study ("*Increasing Illumination Invariance of Learning-Based Local Features using Photo-Realistic Simulated Environments*"), we introduced a Photo-Realistic Synthetic Illumination (PRSI) dataset combined with an Illumination Condition Adaptation (ICA) training procedure.

- **PRSI Dataset**: 74k high-fidelity day–night image pairs rendered with Unreal Engine 5, covering indoor, outdoor, and urban scenes.
- **ICA Training Procedure**: Combines feature points from fully-lit and low-light frames, filtered by non-maximum suppression and weighting toward high-confidence daytime features, to produce pseudo ground-truths for detector refinement.
- **Homographic Adaptation**: Further boosts detector invariance by applying geometric transformations, increasing repeatability across viewpoint changes.

Results demonstrated significantly higher matching scores and reduced trajectory errors during visual odometry on KITTI-derived night-time datasets, yielding up to 83% reduction in Mean ATE and RPE.

2.2. Reinforcement Learning for Persistent Keypoint Detection

The second study (*"A Deep Actor-Critic Reinforcement Learning Framework for Persistent Keypoint Detection under Challenging vSLAM Conditions"*) introduced a learning-based feature selection mechanism:

- **Actor-Critic RL Architecture:** The Actor trains a modified SuperPoint detector to favor keypoints that remain trackable across frame sequences, while the Critic evaluates the long-term contribution of the Actor's decisions using discounted rewards based on geometric consistency (RANSAC + 8-point algorithm).
- **Ground-Truth-Free Training:** The approach eliminates the need for absolute pose ground truth, enabling learning in previously unseen or unlabeled environments.
- **Quality-over-Quantity Principle:** The network learns to prefer fewer but more persistent keypoints, reducing computational load and improving localization accuracy in low-light and texture-sparse conditions.

Extensive experiments on KITTI, Oxford RobotCar (night subset), and 4Seasons datasets showed superior Absolute Trajectory Error (ATE) and RMSE compared to the original SuperPoint, particularly with as few as 500 selected features.

3. Integration with LEARNER Middleware

These contributions directly support the semantic mapping layer described in Deliverable D2.2 and feed into WP3's social-aware PP in the following manner:

- **Illumination-Robust Mapping:** Ensures that the hybrid map remains populated with reliable landmarks even in dark or dynamically lit environments, allowing consistent localization.
- **Adaptive Feature Prioritization:** RL-driven keypoint selection enables the robot to adapt its perception policy on the fly, dynamically improving map quality in areas where standard detectors would fail.

- **Foundation for Social Semantics:** The stable and high-confidence feature maps produced here act as the backbone for higher-level semantic segmentation (e.g., human presence, object detection) used in D2.4 and WP3 PP**.**

Together, these results form a robust perception layer that guarantees the consistency of the topological and metric maps used for socially compliant navigation.

4. Conclusion

Deliverable D2.3 provides a dual-strategy solution to the problem of robust semantic mapping under difficult visual conditions: (i) pre-trained illumination-adapted detectors via data augmentation, and (ii) adaptive reinforcement-learning-based feature prioritization for persistent tracking. These advancements ensure that LEARNER's navigation stack can maintain localization accuracy and semantic awareness in real-world scenarios with poor or changing lighting.

The two research papers are provided in Annexes A for detailed methodological and experimental insights.

Annex A

Increasing Illumination Invariance of Learning-Based Local Features using Photo-Realistic Simulated Environments

Anastasios Agakidis[a], Antonios Gasteratos[a], Loukas Bampis[b]

[a]*Production and Management Engineering, Democritus University of Thrace, Xanthi, Greece*

[b]*Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece*

Abstract

Reliable local feature detection is crucial for autonomous robotics, yet dynamic lighting conditions often undermine performance. While traditional algorithms struggle, deep learning has set new standards in accuracy and adaptability for key point extraction. However, challenges persist in ensuring robustness under variable illumination. In this paper, we propose a novel method to enable illumination adaptation of learned feature detectors and descriptors which can increase the applicability of existing mapping and localization techniques. Our approach combines Photo-Realistic Synthetic Illumination (PRSI) dataset with an Illumination Conditions Adaptation (ICA) approach, designed to improve generalization across diverse lighting scenarios by leveraging robust pseudo-ground truths. Extensive evaluation is performed using HPatches and KITTI subsets for visual odometry. Results highlight significant improvements in feature detection and description robustness, particularly in low-light conditions and abrupt lighting transitions leading to increased localization accuracy compares to the state-of-the-art.

Keywords: Feature Point Detection, Computer Vision, Deep Neural Networks, Illumination Invariance, Synthetic Image Dataset

1. Introduction

In the rapidly evolving field of robotics, the drive towards autonomy has magnified the importance of robust robot localization mechanisms.

One of the most established approaches for real-time localization relies on the detection and tracking of distinct local feature points using RGB camera sensors [1, 2, 3, 4, 5], due to their affordability, compactness, and low energy consumption. These features serve as markers for robots to measure movement and orientation in space, enabling tasks such as Simultaneous Localization and Mapping (SLAM) and Visual Place Recognition (VPR), which are crucial for applications like domestic robots and autonomous vehicles [6, 7]. SLAM builds maps while navigating, and VPR allows recognition of previously visited locations, both relying on robust feature detection under diverse conditions.

Some of the most acknowledged methods use hand-crafted algorithms to detect key points in images [8, 9]; however, deep learning methods, particularly Convolutional Neural Networks (CNNs), have significantly advanced feature extraction [10]. Nevertheless, feature detection under low and highly varying illumination remains an open research challenge, as lighting changes can impair reliability, especially in dynamic environments [11]. To address this challenge, synthetic datasets have emerged as a solution, providing diverse training data in a cost-effective and controlled manner [12, 13, 14, 15].

The motivation behind our proposal lies in enhancing the capabilities of RGB cameras to their fullest potential, ensuring that even the most energy-restrictive devices can achieve high levels of autonomy and environmental awareness. To this end, we developed a new Photo-Realistic Synthetic Illumination (PRSI) dataset that combines the advantages of synthetic data while also imitating real-world conditions through hyper-realistic lighting and textures. This allows the integration of an Illumination Condition Adaptation (ICA) step, which guides the training process of any learning-based local feature extraction technique towards consistent detections and descriptions (Fig. 1).

A preliminary version of our work was presented in [16], where the concept of ICA was first introduced, providing sufficient matching accuracy under severe lighting changes but struggling with general applicability across real-world datasets, such as HPatches [17]. This paper expands upon our previous method by providing the following novel contributions:

- Refinement of the original ICA method by utilizing a greater amount of available information to provide better associations between features under different illumination conditions.

- A thorough hyperparameter tuning process to boost the model's performance and generalization capabilities.
- Enhancement of the PRSI dataset with increased image samples covering a wider variety of environmental changes.
- Rigorous evaluation in multiple different scenarios, including visual odometry runs in day and night conditions.
- Release of PRSI in a publicly available repository[1] to further support future research in the field of robotics vision and localization.

The rest of this paper is organized as follows: Section 2 reviews the existing literature and methodologies relevant to our research. Section 3 provides a detailed description of our methodology and its practical applications. Section 4 details the experiments conducted to validate our approach, while Section 5 presents the results of these experiments. Finally, Section 6 concludes our findings and suggests directions for future research.

2. Literature Review

Local-feature-based SLAM architectures are typically based on the detection of repeatable key points in the environment, which are tracked among consecutive frames to compute an estimation of the camera's locomotion and the environment's structure [5]. Traditional methods for feature detection, such as SIFT [8], SURF[18], and ORB [19], have served as cornerstones in the field. SIFT is probably the most acknowledged method for extracting features from images, and it can be used to perform reliable matching between different views of an object or scene. These features are invariant to image scale and rotation. ORB is designed to be a faster alternative to floating point features, offering both efficiency and performance by combining a fast feature point detector (FAST [20]), with a robust descriptor (BRIEF [21]), while also incorporating orientation and scale invariance. However, these algorithms rely on gradient-based handcrafted rules; therefore, their performance is significantly impacted under conditions of extreme illumination variations, and low lighting [6].

[1]The PRSI dataset will be published upon acceptance of this paper.

In recent years, the field has shifted towards deep-learning-based methods. Deep learning models showed improvements in the performance of feature detection and description across a broad range of conditions. SuperPoint[22] and D2-Net[23] are notable examples of these. SuperPoint employs a self-supervised approach with pre-training on simple images to learn basic feature detection, followed by self-supervised training to match features between different images of the same scene. D2-Net uses a single CNN for joint detection and description with dense feature extraction for each pixel, maintaining robustness across scales and transformations. Moreover, LF-Net [24] provides an end-to-end model for simultaneous feature detection and description, training on simulated real-world changes in viewpoint and lighting. R2D2 [25] focuses on repeatable and reliable feature points, using a specially designed loss function to ensure consistency across viewpoint changes. Finally, ASLFeat [26] integrates attention mechanisms to focus on informative regions, enhancing dense feature extraction by dynamically adjusting the importance of different image areas. These models advance feature detection and description in terms of accuracy, robustness, and efficiency by leveraging deep learning to address the challenges that traditional algorithms face. However, the illumination invariance problem persists and calls for specifically designed learning procedures that dictate common landmark features to be detected, despite any appearance changes of the scene.

Synthetic datasets are pivotal in advancing computer and robotics vision by simulating real-world variability on demand, under extensively-controlled conditions. Notable publicly available datasets include SYNTHIA [12], which focuses on urban scenarios with diverse layouts, weather, and lighting conditions, that are valuable for autonomous driving research. SUNCG [13] provides detailed indoor scenes with various lighting and furniture arrangements, essential for indoor navigation and object recognition. Virtual KITTI[14] replicates real-world KITTI[27] scenarios with controlled variability, useful for object detection and tracking in driving contexts. As a final note, CARLA [15], an open-source simulator, allows the creation of custom scenarios with varying weather, lighting, and traffic conditions. Despite their strengths, none of these datasets explicitly combine realistic lighting conditions and image pairs from the same scene sharing exact camera poses and locomotion.

3. Methodology

This section is structured into two primary parts to address the development and implementation of our feature detection enhancements using the ICA method. The first one focuses on our proposal's architecture (Fig. 2), detailing the ICA method and how it can be integrated into an existing deep feature extraction pipeline, which for this work is based on SuperPoint [22]. This combination is critical for testing and refining the ICA's capability to enhance feature extraction under varying lighting conditions. The second part of the presented methodology describes the creation and characteristics of the PRSI dataset, designed specifically to include image pairs that capture the essential lighting condition transitions, and thus, enabling effective training of ICA.

3.1. Architecture

3.1.1. ICA

With ICA, we aim to enhance the performance of feature detection under varying illumination conditions; particularly, by providing consistent associations among fully-lit and low-lit or nighttime scenarios. Given a feature point detection input within a trainable pipeline, ICA makes use of feature points as ground truths for the subsequent learning phases, viz., the detector's and descriptor's refinement. This whole process is inspired by the principles of data adaptation, specifically targeting the challenges posed by different lighting conditions.

To implement ICA, a dataset that contains pairs of identical images captured under different lighting conditions for every scene p_i is needed. Each $p_i = \{I_{f_i}, I_{l_i}\}$ contains a camera measurement of a fully-lit version of the scene (I_{f_i}) and one of low lighting (I_{l_i}). Both images are captured from the same position and orientation, ensuring that the geometric structures and scene elements remain constant across the pair, with only the illumination conditions being changed. The PRSI dataset, described in Section 3.2, fulfills these requirements, targeting specifically the day-to-night challenge.

ICA involves several steps to adapt the detection capabilities to varying illumination conditions. Feature points are first extracted from I_{f_i} using any type of feature detector. These points ($\mathcal{F}_{f_i}$) are assumed to be more reliable than the low-lighting ones, due to the better visibility and contrast provided by the corresponding frames [28]. Feature points from I_{l_i} are also extracted ($\mathcal{F}_{l_i}$), to capture landmarks usually visible during the night (e.g., a lit light

bulb). Subsequently, we combine $\mathcal{F}_{f_i}$ and $\mathcal{F}_{l_i}$ and filter out duplicate points, as well as points in very close proximity using Non-Maximum Suppression (NMS) [29], of value 4. We apply a threshold giving more weight on the features detected in the daily image. The final set of combined and filtered points $\mathcal{F}_{fl}$, from all the available p_i pairs will be used as pseudo ground truths for the subsequent detector and descriptor training.

In the following subsections, we describe the network structure adopted within this work [22]. However, different architectures can also be adapted to include the ICA module.

3.1.2. Network backbone

The process initiates with a series of synthetic images composed of basic geometric shapes such as circles, squares, and triangles. These shapes act as the foundational elements for constructing more complex patterns and structures. Initially, the model is trained on this synthetic dataset, enabling it to learn how to detect feature points within a controlled environment. The training employs the following loss function, using ground truth points generated from the edges of the synthetic shapes:

$$\mathcal{L}_{det}(\mathcal{X}, G) = \frac{1}{H_e W_e} \sum_{\substack{h=1,\\ w=1}}^{H_e, W_e} l_{det}\left(\mathbf{x}_{hw}; G_{hw}\right), \tag{1}$$

where

$$l_{det}\left(\mathbf{x}_{hw}; g\right) = -\log\left(\frac{\exp\left(\mathbf{x}_{hwg}\right)}{\sum_{k=1}^{65}\exp\left(\mathbf{x}_{hwk}\right)}\right). \tag{2}$$

In the above, H_e and W_e refer to the downsampled dimensions of the images, which are divided into 8×8 pixel regions. The detector operates on X, a tensor with dimensions $R^{(H_e \times W_e \times 65)}$, producing an output of $R^{(H \times W)}$. After applying a softmax function to each channel, the dustbin compartment (indicating the absence of a feature point) is removed, and a reshaping operation converts $R^{(H_e \times W_e \times 64)}$ to $R^{(H \times W)}$. The detector's loss function uses a fully convolutional cross-entropy loss applied to elements $\mathbf{x}_{hw}$ within X. The ground truth labels for the feature points, collectively termed G, have individual components denoted as G_{hw}.

This generates a heatmap that indicates the likelihood of each pixel being a feature point for any given input image. However, due to accuracy issues in real-world tests, a homographic adaptation step is additionally employed.

Homographic adaptation adjusts an image I using a predefined homography or transformation. This process involves applying various transformations $\mathcal{T}$, such as rotations, translations, warping, and scaling, to diversify the detection process. The original image I and the transformed ones $I_{\mathcal{T}}$ are processed by the feature detector, and the resulting heatmaps are combined to produce the final set of feature points $\mathcal{F}$. This method has been shown to significantly enhance the feature detector's accuracy [22]. By applying the above homographic adaptation procedure over the I_{f_i} and I_{l_i} samples, the corresponding $\mathcal{F}_{f_i}$ and $\mathcal{F}_{l_i}$ points described in Section 3.1.1 are produced.

3.1.3. Training

The training phase involves developing a network utilizing both real and synthetic datasets. To enhance the diversity and realism of our learning samples, we integrate the Common Objects in Context (COCO) dataset [30]. COCO is highly regarded in the computer vision community for its utility in tasks such as object detection, segmentation, and captioning, owing to its wide range of complex and varied images that feature numerous objects and scenes. Although COCO includes annotations, we utilized the images without these labels for our training. The dataset is split into approximately $82k$ training samples and $40k$ validation samples. For each sample I_j from the COCO dataset, feature points $\mathcal{F}_j$ are extracted after homographic adaptation. These real-world data are combined with the synthetic ones (I_{fl} and $\mathcal{F}_{fl}$) to finally form our overall learning samples I and labels $\mathcal{F}$

Our approach employs both a detection and a description encoder for feature points. This involves a concurrent refinement process for both components of the network. Training is guided by a multi-task loss function that balances the tasks of detection and description. The overall loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda \cdot \mathcal{L}_{desc} \, . \tag{3}$$

In the above, the detector's loss $\mathcal{L}_{det}$ uses the same function defined in equation 1, while the descriptor's loss is computed as:

$$\begin{aligned} &\mathcal{L}_{desc}\left(\mathcal{D}, \mathcal{D}', S\right) = \\ &\frac{1}{\left(H_e W_e\right)^2} \sum_{\substack{h=1 \\ w=1}}^{H_e, W_e} \sum_{\substack{h'=1 \\ w'=1}}^{H_e, W_e} l_{desc}\left(\mathbf{d}_{hw}, \mathbf{d}'_{h'w'}; s_{hwh'w'}\right), \end{aligned} \tag{4}$$

where

$$l_{desc}\left(\mathbf{d},\mathbf{d}';s\right)=\lambda\cdot s\cdot\max\left(0,m_p-\mathbf{d}^T\mathbf{d}'\right)\\+(1-s)\cdot\max\left(0,\mathbf{d}^T\mathbf{d}'-m_n\right). \tag{5}$$

$\hat{H}p_{hw}$ denotes the transformation of the cell location p_{hw} by the homography H, divided by the final coordinate, a standard procedure when transitioning between Euclidean and homogeneous coordinates. The entire set of correspondences for a pair of images is denoted with S. Finally, a weight factor λ is introduced to balance the discrepancy due to the presence of more negative correspondences compared to positive ones, and a hinge loss with positive (m_p) and negative (m_n) margins are applied.

3.2. PRSI dataset

To effectively train our proposed ICA methodology, a specialized dataset including image pairs capturing day-to-night transitions is proposed. The dynamic nature of these transitions presents unique challenges in feature detection, making it important to utilize images that mirror real-world conditions as closely as possible, while still maintaining low size to improve training times and save computational power. This necessity leads to the requirements of the PRSI dataset, which is designed to produce high-quality yet low-resolution synthetic images (namely 640x640). The dataset is formed with full control over camera poses, transformations, and objects within the scene. Samples can be seen in Fig. 3. Additionally, we maintain complete supervision over the lighting conditions. An overview of one of the sample maps formulated for this study, along with the camera path is shown in Fig. 4.

The PRSI dataset is created using Unreal Engine 5[2] and Unreal Marketplace assets. To increase its applicability, three different types of scenes are included, namely: i) indoors, ii) outdoors, and iii) urban scenes. These settings are used to test and refine our training methods for feature detection. PRSI includes $37k$ images for each of the day and night segments, leading to a total of $74k$ image samples. To achieve high realism, high-definition textures (up to $8k$ resolution) are used. Rendering is done either with Lumen or Ray-Tracing, both supported natively by Unreal Engine 5.

[2]Unreal Engine 5 is, at the time of writing, the latest graphics engine developed by Epic Games (`https://www.unrealengine.com/en-US/unreal-engine-5`).

The day-to-night image associations are achieved through a scripted camera-based automation system, which precisely replicates the exact sensor transformations across various scenes. This systematic approach ensures that each pair of images shares the same camera position, orientation, and environmental structure setup, yet significantly differs in lighting conditions.

4. Experiments

Our current implementation builds upon and significantly enhances the previous approach [16], through several key improvements. In our earlier work, which will we refer to as **ICA v0** for the rest of this paper, the focus was primarily on reducing irrelevant features, leading to higher matching accuracy among images with significant lighting differences. However, **ICA v0** struggled with general applicability, particularly under changes in the camera's viewpoint, resulting in notably fewer feature point detections. In the current implementation (**ICA v1**), hyperparameter tuning, threshold adjustment, and an extended PRSI dataset are introduced to guide the training process for both detection and description. In this section, we provide the list of experiments we conducted to enhance the training procedure and the trained models.

4.1. HPatches

We utilize HPatches [17] to tune and then evaluate the models on it. HPatches include over $1k$ sample patches collected from various scenes, each comprising a reference image and five variations that represent distinct transformations, viewpoints, and illumination. Our experiments are divided into two testing cases: i) one that uses the full version of HPatches and ii) one that uses only the illumination subset.

The HPatches evaluation employs the metrics below:

- **Repeatability:** Calculates the ratio of correctly matched feature points (with a distance threshold of 3 pixels) to the total number of detected feature points. High repeatability indicates that the detector consistently identifies the same points despite possible changes in the appearance of the scene or the camera pose.
- **Mean Localization Error (MLE):** Computes the Euclidean distance between corresponding feature points detected in different images. This

distance represents the localization error for each feature point, and it is computed as the average among all feature point distances of the evaluation set.

- **Nearest-Neighbor mean Average Precision (mAP):** Assesses the accuracy of feature descriptors by measuring the average precision of the nearest-neighbor matching process.

- **Matching Score:** Evaluates the proportion of correctly matched feature points between image pairs, showing the overall effectiveness of the feature descriptors.

4.2. Visual Odometry

In order to assess our final system, we make use of `PySlam`[3], an open-source visual odometry (VO) and SLAM framework.

To evaluate the models under different lighting conditions, we use two subsets of the KITTI dataset, namely `kitti00` and `kitti06` [27], which offer precise trajectory ground truth data.

We generate low-light and night-time equivalents of the above subsets by drawing inspiration from the approach outlined in [31]. We found that [32] had the best results in transforming a day-time image into a night-time one. This allowed us to generate three new datasets: two using the aforementioned method (`night-kitti00` and `night-kitti06`), and a third one (`darker-night-kitti06`), generated with an image darkening algorithm we created using OpenCV and lookup tables (LUTs) to resemble near complete darkness without light sources. Representative image samples are presented in Fig. 5.

Through the above, our system evaluation within the context of VO and SLAM was performed across five distinct sequences: (i) `kitti06`, (ii) `night-kitti06`, (iii) `night-kitti00`, (iv) `darker-night-kitti06`, and (v) `day-to-night-kitti06`. The `day-to-night-kitti06` sequence transitions between `kitti06` and `darker-night-kitti06` every 30 frames.

To assess the VO performance achieved through the proposed feature extraction approach, several widely used key metrics [33] were utilized:

- **Root Mean Squared Error (RMSE) in X and Y**: Measures deviation in the ‘x‘ and ‘y‘ coordinates.

[3]https://github.com/luigifreda/pyslam

- **Mean Absolute Trajectory Error (ATE)**: Quantifies the global deviation of the trajectory from the ground truth.
- **Incremental Translation Error (ITE)**: Assesses errors in incremental movements between consecutive frames.
- **Relative Pose Error (RPE)**: Measures relative errors between consecutive trajectory estimates.

5. Results

To properly evaluate the ICA method offering a direct measurement for the provided performance improvement, we retrained **Baseline** Model on the PRSI dataset in two distinct ways: one with the use of ICA (**ICA v1**), and one without (**no ICA**). Both models are trained using the same images, ensuring identical inputs.

5.1. Hyperparameter Tuning

A wide set of hyperparameters were evaluated before training our final ICA-enabled model. Specifically, we tested different thresholds and hyperparameters to maximize the number of reliable feature points detected before training the network with the proposed ICA module. We observed that the repeatability and matching score both increase up to a certain point and then decline(as shown in Fig. 6). Maximum performance is reached at a detection threshold of 0.01 and a learning rate of 0.00007. Based on the above, we were able to fine-tune the rest of the training process, ensuring that ICA was experiencing the most robust set of input local image features.

5.2. Evaluation in HPatches

The HPatches dataset (illumination and camera transformation) is used to evaluate the general performance of our models across a variety of conditions, utilizing the evaluation metrics described in Section 4.1. Alongside **ICA v1** and **no ICA**, we also provide the results of the **Baseline** Model (the initial model without retraining or applying ICA) [22].

Table 1: Detector Metrics (HPatches)

Model	Repeatability	MLE
Baseline Model	0.63	1.07
no ICA	0.58	1.10
ICA v1	0.62	1.10

Table 2: Descriptor Metrics (HPatches)

Model	mAP	Matching Score
Baseline Model	0.78	0.45
no ICA	0.77	0.45
ICA v1	0.82	0.51

5.2.1. Full dataset

The results of the evaluation on the whole HPatches dataset are summarized in Table 1 and Table 2. **ICA v1**, using our proposed method (ICA), is performing better than the similarly trained model without the use of ICA. Although the Repeatability and MLE show moderate improvement, the gains in mAP and Matching scores highlight the effectiveness of ICA in enhancing the model's performance.

5.2.2. Illumination only subset

The results on the illumination subset of HPatches are summarized in Table 3 and Table 4. **ICA v1** achieves significantly higher mAP and Matching Scores than **no ICA** and **Baseline** model's, showcasing our method's effectiveness in improving feature reliability under challenging illumination conditions.

5.3. Visual Odometry Evaluation

Our final system is evaluated within the context of a SLAM architecture for computing the visual odometry of an autonomous robot. As evidenced in Table5 model trained with our proposed ICA architecture, offers improved performance results in all evaluated metrics, reaching over 60% RMSE reduction in 'x' and 'y' dimensions and up to 83% for the case of Mean ATE, ITE, and RPE metrics. These improvements highlight ICA's ability in minimizing trajectory errors, reducing global drift, and improving local accuracy in low

Table 3: Detector Metrics (HPatches illumination-only)

Model	**Repeatability**	**MLE**
Baseline Model	0.68	0.95
no ICA	0.66	0.95
ICA v1	0.68	0.93

Table 4: Descriptor Metrics (HPatches illumination-only)

Model	**mAP**	**Matching Score**
Baseline Model	0.81	0.55
no ICA	0.83	0.56
ICA v1	0.86	0.61

visibility and night-time scenarios. In the `kitti-06` dataset, both models exhibited similar performance, showing that the observed improvements are attributed to the method itself rather than the characteristics of the dataset. For the case of `day-to-night-kitti06` specifically, ICA effectively handles extremely dynamic lighting transitions, with substantially reduced error metrics, showing that the same local features cannot only be used at extreme -through static- illumination conditions; but also in cases where the lighting significantly changes over time.

Furthermore, Fig.7 presents qualitative results of the estimated trajectories (green) as compared to the ground through (red) of the KITTI dataset. As it can be seen, the **ICA v1** is capable of computing the platform's path significantly more accurately, proving the significance of our method for autonomous robotic missions. Finally, Figure 8 shows feature matching results between day and night images from VO evaluation sequences. The night images are two frames ahead in the sequence. We include three examples: `kitti06` with `dark-kitti06`, `kitti06` with `darker-kitti06`, and `kitti00` with `dark-kitti00`, highlighting the system's robustness in detecting and matching features under varying illumination.

6. Conclusions

In this paper, we presented a comprehensive study on enhancing feature detection and description under varying illumination conditions, targeting

Table 5: Comparison of **ICA v1** and **No ICA** models across the test sequences.

Model	Metric	kitti06	night-kitti06	darker-night-kitti00	day-to-night-kitti06	night-kitti00
ICA v1	RMSE in X	4.38	16.32	22.92	11.14	11.94
	RMSE in Y	4.68	2.77	4.28	4.17	5.13
	Mean ATE	4.84	13.42	18.75	8.63	11.84
	Mean ITE	1.97	36.75	112.07	19.12	9.95
	Mean RPE	0.07	0.41	1.12	0.34	0.09
No ICA	RMSE in X	5.99	45.98	138.59	79.29	35.52
	RMSE in Y	2.84	3.45	31.24	38.96	21.44
	Mean ATE	5.04	29.42	109.26	71.17	36.30
	Mean ITE	0.93	44.85	338.40	154.27	46.12
	Mean RPE	0.05	0.83	4.46	1.74	0.45

autonomous robot applications that operate with a single RGB camera. We started by expanding our preliminary implementation of the PRSI dataset, which provides high-quality synthetic images with controlled lighting conditions, ensuring reliable training data. We then expanded and refined our ICA method, which leverages the reliable feature points detected in fully lit images with features from the low-lighting samples to guide the training process.

By comparing models trained with and without the use of ICA, we highlighted its critical role in significantly enhancing local feature detection and matching, in addition to the visual localization performance of a SLAM architecture especially when lighting conditions became progressively darker. Our experiments showed improvements in key metrics such as MLE, mAP, and matching score on the evaluation set of HPatches, as well as in the trajectory errors, RMSE, mean ATE, ITE, and RPE, of the PySlam toolkit. Our future work will explore the integration of additional sensors, such as LIDAR, and the use of more complex datasets to further improve the robustness and applicability of our approach in a wider range of environmental conditions.

Acknowledgments

This research is implemented in the framework of H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union – NextGenerationEU (H.F.R.I. Project Number: 15339).

References

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, J. D. Tardós, ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM, IEEE Transactions on Robotics 37 (6) (2021) 1874–1890.

[2] K. A. Tsintotas, L. Bampis, A. Gasteratos, The Revisiting Problem in Simultaneous Localization And Mapping: A Survey on Visual Loop Closure Detection, IEEE Transactions on Intelligent Transportation Systems 23 (11) (2022) 19929–19953.

[3] I. T. Papapetros, V. Balaska, A. Gasteratos, Visual loop-closure detection via prominent feature tracking, Journal of Intelligent & Robotic Systems 104 (3) (2022) 54.

[4] L. Bampis, A. Amanatiadis, A. Gasteratos, Fast Loop-Closure Detection using Visual-Word-Vectors from Image Sequences, The International Journal of Robotics Research 37 (1) (2018) 62–82.

[5] S. Li, S. Liu, Q. Zhao, Q. Xia, Quantized Self-Supervised Local Feature for Real-Time Robot Indirect VSLAM, IEEE/ASME Transactions on Mechatronics 27 (3) (2022) 1414–1424.

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A Benchmark for the Evaluation of RGB-D SLAM Systems, in: Proceedings of the IEEE/RSJ Int. Conf. on intelligent robots and systems, 2012, pp. 573–580.

[7] K. M. Oikonomou, I. Kansizoglou, A. Gasteratos, A Hybrid Reinforcement Learning Approach with a Spiking Actor Network for Efficient Robotic Arm Target Reaching, IEEE Robotics and Automation Letters (2023).

[8] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[9] E. Rosten, T. Drummond, Machine Learning for High-Speed Corner Detection, in: Proceedings of the European Conf. on Computer Vision, 2006, pp. 430–443.

[10] C. Deng, K. Qiu, R. Xiong, C. Zhou, Comparative Study of Deep Learning Based Features in SLAM, in: Proceedings of the IEEE Asia-Pacific Conf. on Intelligent Robot Systems, 2019, pp. 250–254.

[11] X. Wu, C. Sun, L. Chen, T. Zou, W. Yang, H. Xiao, Adaptive orb feature detection with a variable extraction radius in roi for complex illumination scenes, Robotics and Autonomous Systems 157 (2022) 104248. doi:https://doi.org/10.1016/j.robot.2022.104248.
URL https://www.sciencedirect.com/science/article/pii/S0921889022001439

[12] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. M. Lopez, The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 3234–3243.

[13] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, T. Funkhouser, Semantic Scene Completion from a Single Depth Image, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2017, pp. 1746–1754.

[14] A. Gaidon, Q. Wang, Y. Cabon, E. Vig, Virtual Worlds as Proxy for Multi-Object Tracking Analysis, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 4340–4349.

[15] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An Open Urban Driving Simulator, in: Proceedings of the Conf. on Robot Learning, 2017, pp. 1–16.

[16] A. Agakidis, L. Bampis, A. Gasteratos, Illumination Conditions Adaptation for Data-Driven Keypoint Detection under Extreme Lighting Variations, in: Proceedings of the IEEE Int. Conf. on Imaging Systems and Techniques, 2023, pp. 1–6.

[17] V. Balntas, K. Lenc, A. Vedaldi, K. Mikolajczyk, HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2017, pp. 5173–5182.

[18] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, in: Proceedings of the European Conf. on Computer Vision, 2006, pp. 404–417.

[19] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An Efficient Alternative to SIFT or SURF, in: Proceedings of the IEEE Int. Conf. on Computer Vision, 2011, pp. 2564–2571.

[20] A. Angeli, D. Filliat, S. Doncieux, J.-A. Meyer, Fast and Incremental Method for Loop-Closure Detection using Bags of Visual Words, IEEE Transactions on Robotics 24 (5) (2008) 1027–1037.

[21] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary Robust Independent Elementary Features, in: Proceedings of the European Conf. on Computer Vision, 2010, pp. 778–792.

[22] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-Supervised Interest Point Detection and Description, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2018, pp. 224–236.

[23] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, T. Sattler, D2-NET: A Trainable CNN for Joint Description and Detection of Local Features, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 2019, pp. 8092–8101.

[24] Y. Ono, E. Trulls, P. Fua, K. M. Yi, LF-Net: Learning Local Features from Images, Advances in Neural Information Processing Systems 31 (2018).

[25] J. Revaud, C. De Souza, M. Humenberger, P. Weinzaepfel, R2d2: Reliable and repeatable detector and descriptor, Advances in neural information processing systems 32 (2019).

[26] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, L. Quan, ASLFeat: Learning Local Features of Accurate Shape and Localization, in: Proceedings of the IEEE/CVF Conf. on computer vision and pattern recognition, 2020, pp. 6589–6598.

[27] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision Meets Robotics: The KITTI dataset, The International Journal of Robotics Research 32 (11) (2013) 1231–1237.

[28] L. V. Lozano-Vázquez, J. Miura, A. J. Rosales-Silva, A. Luviano-Juárez, D. Mújica-Vargas, Analysis of Different Image Enhancement and Feature Extraction Methods, Mathematics 10 (14) (2022) 2407.

[29] A. Neubeck, L. Van Gool, Efficient Non-Maximum Suppression, in: Proceedings of the IEEE Int. Conf. on Pattern Recognition, 2006, pp. 850–855.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: Proceedings of the European Conf. in Computer Vision, 2014, pp. 740–755.

[31] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, S. Yogamani, FuseMODNet: Real-Time Camera and LiDAR Based Moving Object Detection for Robust Low-Light Autonomous Driving, in: Proceedings of the IEEE/CVF Int. Conf. on Computer Vision Workshops, 2019, pp. 0–0.

[32] G. Parmar, T. Park, S. Narasimhan, J.-Y. Zhu, One-Step Image Translation with Text-to-Image Models, arXiv preprint arXiv:2403.12036 (2024).

[33] V.-J. Štironja, J. Peršić, L. Petrović, I. Marković, I. Petrović, Movro2: Loosely coupled monocular visual radar odometry using factor graph optimization, Robotics and Autonomous Systems 184 (2025) 104860. doi:https://doi.org/10.1016/j.robot.2024.104860.
URL https://www.sciencedirect.com/science/article/pii/S0921889024002446

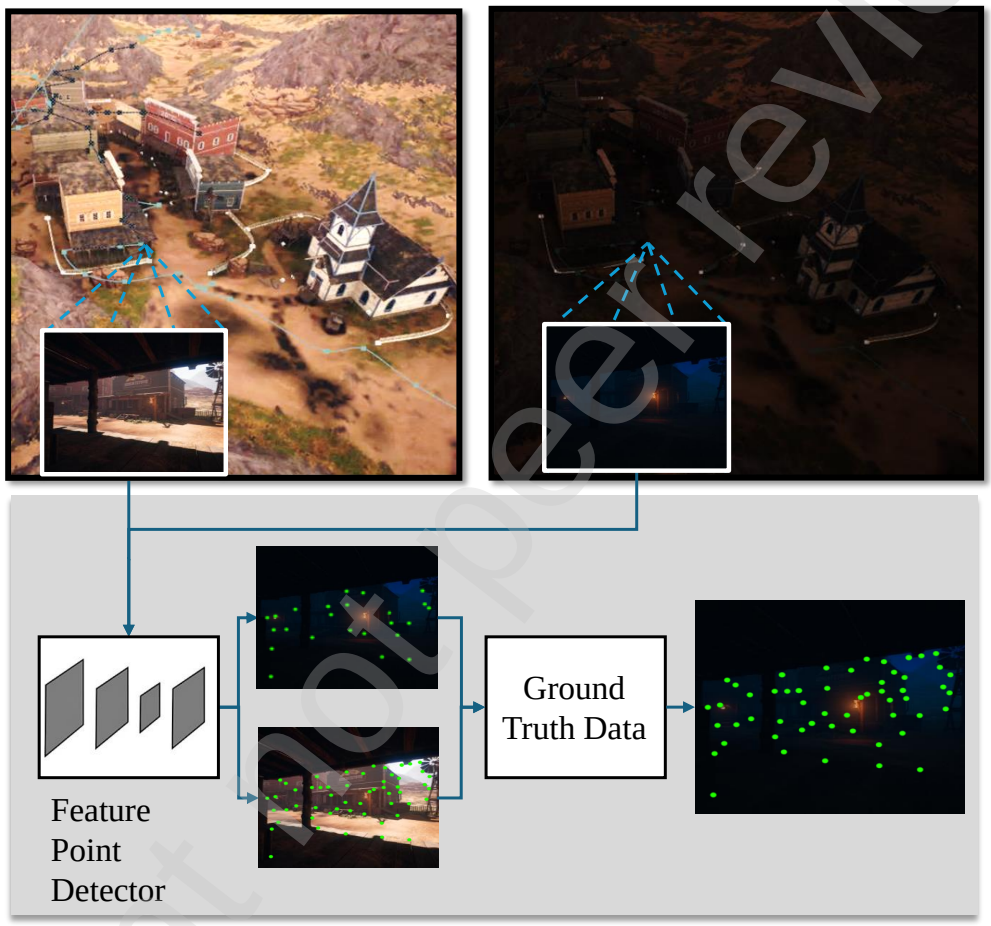


Figure 1: Illustration of our proposed Illumination Conditions Adaptation (ICA) method. Features are detected on two identical views with different illumination conditions. They are combined, filtered, and then used as ground truths for the subsequent training.

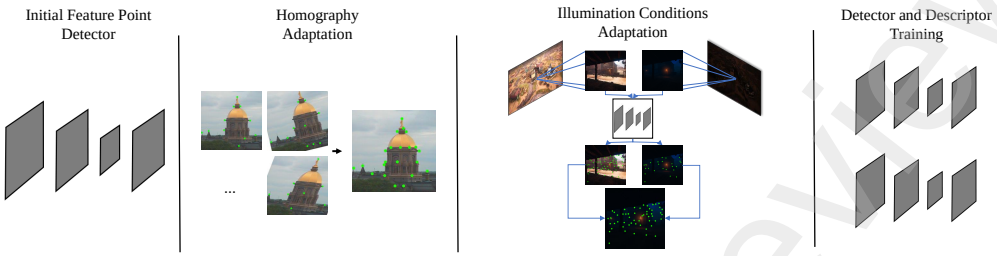


Figure 2: Schematic representation of Illumination Condition Adaptation (ICA) integration into the selected deep local feature extraction model. Initially, keypoints are detected from the daytime and the equivalent nighttime image using the pre-trained detector. Homographic adaptation is then applied to each input image, generating multiple transformed versions through rotations, translations, and scalings. The resulting heatmaps are aggregated, and ICA is used to filter, combine, and impose the feature points as ground truths for the subsequent training of our final system/

Figure 3: Sample images from our dataset demonstrating corresponding day (left) and night (right) recordings of the same scene.

Figure 4: One of the maps used to render images for the proposed PRSI dataset.

Figure 5: Sample images from (from top to bottom): `kitti06`, `night-kitti06`, `darker-night-kitti06`, and `night-kitti00`.

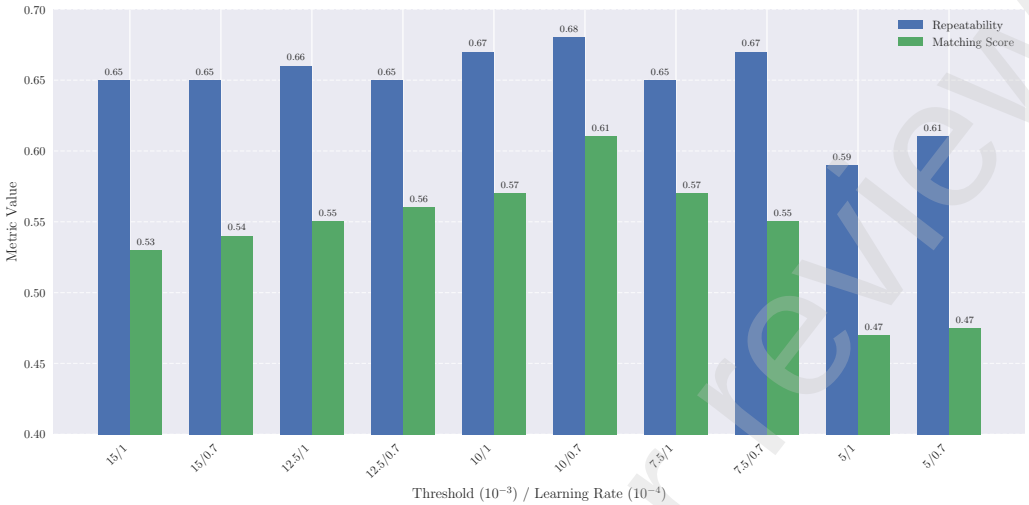
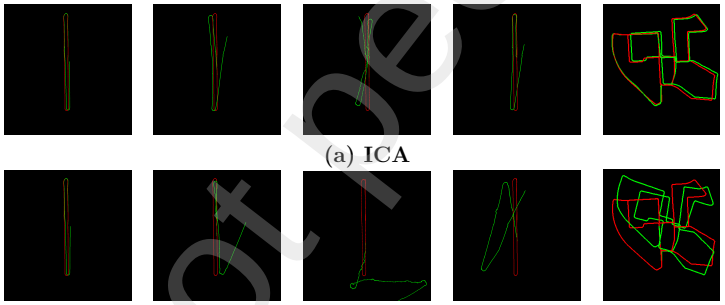


Figure 6: Performance metrics across different thresholds and learning rates.



(a) ICA

(b) No ICA

Figure 7: Comparison of trajectories generated by (a) **ICA v1** and (b) **No ICA**. From left to right: `kitti06`, `night-kitti06`, `darker-night-kitti06`, `day-to-night-kitti06`, and `night-kitti00`. The red line represents the ground truth trajectory, while the green line depicts the estimated trajectory.

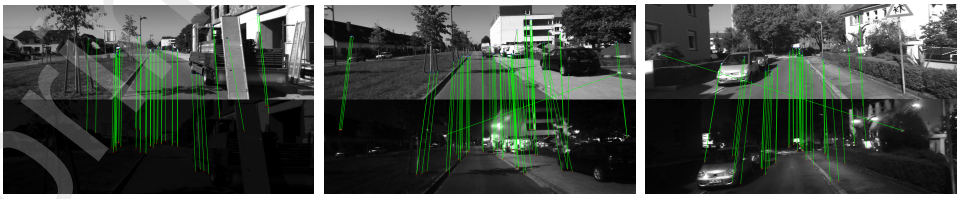


Figure 8: Feature matching results between day and night images, where the night image is two frames ahead. From left to right: `kitti06` with `dark-kitti06`, `kitti06` with `darker-kitti06`, and `kitti00` with `dark-kitti00`.

A Deep Actor-Critic Reinforcement Learning Framework for Persistent Keypoint Detection under Challenging vSLAM Conditions

Panagiotis Bakirtzis[1], Ioannis Kansizoglou[2], and Loukas Bampis[1]

***Abstract*—Existing visual Simultaneous Localization and Mapping (vSLAM) methods heavily rely on visual feature detection algorithms, which become unreliable or indistinguishable under poor illumination conditions. When visual perception does not meet the requirements of each specific environment, effective navigation should not rely on higher resolution sensing, but on smarter learning algorithms and predictive models. In this paper, we present a novel approach by introducing a Reinforcement Learning framework, utilizing an Actor-Critic scheme, which can be used by any type of deep feature detector. Our proposal prioritizes the extraction of fewer, more robust, and reliable subsets of keypoints that are trackable over time, leading to improved matching and, in turn, more accurate localization. The goal is quality over quantity by assigning to the RL the task of objectively deciding which keypoints are considered important for vSLAM. Our experimental results demonstrate that our method outperforms the original state-of-the-art models in a variety of publicly available challenging datasets for localization.**

***Index Terms*—vSLAM, Localization, Keypoint Detection, Deep Learning, Reinforcement Learning**

I. Introduction

WHETHER guiding robots successfully through crowded cities with Autonomous Navigation Systems or interacting with digital objects in the real-world through Augmented Reality [1], [2], [3], the core functionality relies on accurate and efficient visual Simultaneous Localization and Mapping (vSLAM) [4] solutions, where robust, low-cost, and passive sensing is essential. The general SLAM [5], [6], [7] algorithms in autonomous systems aim to estimate an agent's pose, while concurrently building a map of the environment by fusing sensory data. On the contrary, vSLAM has gained attention for its simplicity, by solely utilizing camera-derived measurements as input to leverage rich visual information. However, it comes with unique challenges since it depends on the environment's scene appearance, which can degrade feature tracking and subsequently lead to trajectory and mapping failures.

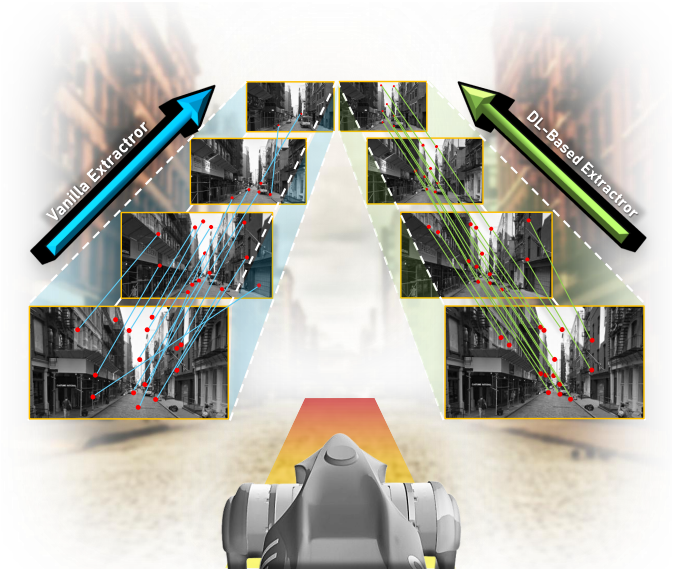


Fig. 1. An overview of the proposed system. The core idea includes the deployment of a Reinforcement Learning (RL) pipeline to train a deep feature extractor on detecting robust visual keypoints that would "survive" within visual Simultaneous Localization and Mapping (vSLAM) [4] architecture.

While conventional vSLAM approaches have proven themselves effective in controlled environments, their strict dependence on carefully designed hand-crafted visual features, in combination with their demanding geometric modeling makes them fragile in challenging real-world tasks with weak textures, varied and harsh illumination conditions or unpredictable structures. In such cases, classic feature extraction methods produce short-lived or weak keypoints, leading to deficient vSLAM performance. Based on the above interpretation, it becomes necessary to revisit the keypoints selection and exploitation strategies, in order to address feature stability and improve long-term localization.

Inspired by the necessity to fill this gap, our proposed solution comes as an addition in visual feature extractor techniques specifically targeting vSLAM architectures. Our guiding principle is to maximize long-term feature trackability, rather than raw feature count, as seen in Fig. 1. To achieve this, we developed a learning-based feature selection pipeline [8], making use of the well-established SuperPoint [9] architecture

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor FirstName A. EditorName upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union – NextGenerationEU

[1]Authors are with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Kimmeria Campus, GR-67100, Xanthi, Greece pbakirtz@ee.duth.gr, lbampis@ee.duth.gr

[2]Author is with the Department of Production and Management Engineering, Democritus University of Thrace, 12 Vas. Sophias, GR-67132, Xanthi, Greece iokansizo@pme.duth.gr

Digital Object Identifier (DOI): see top of this page.

as a case-study. Traditionally, local keypoints are detected, paired, and refined via geometry verification techniques, such as the 8-Point [10] and the Random Sample Consensus (RANSAC) [11] algorithms. Through reverse engineering, we exploit this procedure in order to systematically train a deep Reinforcement Learning (RL) [12] architecture to discern local features in the detection state, that are more likely to survive the above filtering scheme in a sequence of images.

Through a reward-penalty policy, RL has the ability to teach the whole system on how to prioritize visual features that contribute the most to long-term localization accuracy. We take RL a step further by employing an Actor-Critic [13], [14] framework on it, which operates by jointly training a policy (Actor) on choosing stable features and a value function (Critic) on evaluating their long-term contribution to localization. Unlike conventional and existing Deep Learning (DL) methods [15], [16], we introduce the following key novelties:

- **No Ground Truth Required During Training:** Our method enables autonomous adaptability, allowing real-time vSLAM in unknown complex conditions without the need of predefined pose data.
- **Reliable and Persistent Keypoints Through Time:** The detected features serve as robust landmarks for consistent vSLAM across multiple observations, allowing for fewer keypoints and reducing the computational complexity.
- **Environment-Aware vSLAM:** The above integration allows the generation of environment-specific representations, which are particularly valuable in challenging conditions, such as low-lighting.

Finally, to systematically evaluate the effectiveness of the proposed method and ensure fair and consistent comparison with the latest existing approaches, we follow a structured protocol to support reliable benchmarking through *training*, *evaluation*, and *testing*, demonstrating competitive performance compared to the state-of-the-art.

The paper's structure is outlined as follows. In Section II, we investigate the existing literature in the domain of vSLAM implementations and visual keypoint-based extraction, highlighting key trends and limitations. Section III introduces in detail our proposed method to facilitate understanding. The experimental framework and our proposal's evaluation within vSLAM is presented in Section IV. The paper concludes in Section V, where our findings are summarized and potential extensions are discussed.

II. Related Work

In the typical cases, vSLAM architectures rely on the detection of hand-crafted local keypoints from each frame, which are matched among consecutive images. These matches are used to extrapolate both the camera transformation and the coordinates of those keypoints in the 3D world. Among the available literature, ORB-SLAM3 [17] is one of the most advanced and widely used feature-based vSLAM architectures. It presents enhanced control over multi-map localization, loop closure, and global optimization through cost function minimization. Its efficient relocalization ensures repositioning while operating in real-time. Evidently, ORB-SLAM3 exhibits great results in rich textured environments, offering competitive results against other similar methods in its category. Despite their computational efficiency, there are significant shortcomings for vSLAM architectures that are based on hand-crafted rules for detecting local keypoints. Under low-texture, dynamic, and challenging lighting conditions they become sensitive, leading to tracking failures and unreliable feature extraction [18].

Learning Invariant Feature Transform-SLAM (LIFT-SLAM) [19], as its name suggests, is an advanced hybrid monocular vSLAM system that seamlessly integrates DL-based feature extraction and traditional model-based SLAM techniques into a single framework. It harnesses the power of deep Convolutional Neural Networks (CNNs) to learn discriminative features. Moreover, the network's output is a set of keypoints with associated descriptors for tracking, localization, and mapping, with the ability to remain invariant to scaling, rotation, and illumination alternations. Though robust, it still faces certain challenges. The LIFT network requires large amounts of labeled data for training, making it both resource intensive and time consuming. This dependency on large-scale data can result in system behaviour shifts, making it vulnerable to overfitting and unable to generalize well to unseen environments or novel conditions. Additionally, LIFT-SLAM faces challenges in dynamic environments cause of its versatility in dynamic object segmentation or tracking.

Conversely, modern DL-based techniques have come to remodel the existing SLAM methodologies. BEV-DWPVO [20] is a DL-driven approach to Visual Odometry (VO). By using DL, it combines a Bird's Eye View (BEV) perspective with depth estimation that attempts to calculate and infer both the relative camera motion-pose and depth information. The process involves the transformation of images into a top-down view, simplifying the terrain's spatial representation. Its architecture is based on end-to-end optimization that leads to even better integration and performance of the entire pipeline of camera motion estimation. While the current design improves efficiency, it still relies on ground truth pose data, which cannot be guaranteed for each targeted environment.

As evident by the above, local feature detection plays a significant role in the performance of each individual vSLAM architecture. Traditional feature detectors, such as Oriented FAST and Rotated BRIEF (ORB) [21], Speed Up Robust Features (SURF) [22], Scale-Invariant Feature Transform (SIFT) [23], and more recent self-trained methods like SuperPoint [9] are usually designed and trained to improve low-level matching rates. ORB is well-known for its lightweight algorithmic design and the ability to extract keypoints fast and efficiently, achieving an equilibrium between process speed and quantity. SURF and SIFT have both similar architectures, ideal for robust detections and accurate matching. Nevertheless, the above hand-crafted optimizations, under challenging conditions, do not always assure the corresponding expected enhanced performance in high-level vision tasks, becoming often sensitive to lighting changes, and therefore unreliable to detect consistent keypoints in low illumination environments. SuperPoint, similarly to its derivatives [24], [25], overcomes

the classical feature detection methods. Its special fully-convolutional model has made it increasingly popular to be integrated in modern DL-based SLAM systems. Nevertheless, it requires additional time-consuming training, with guidelines that focus on detecting corners in a generic manner, without considering tracking persistent keypoints from prominent world landmarks.

Among the existing literature, probably the most closely aligned with our proposal is the one in [26]. The authors present a pre-trained DL feature extractions network aimed at optimizing the detection and description of keypoints across consecutive frames. Keypoint selection and descriptor matching are handled as sampling probabilistic operations. The keypoints are matched across the frames and used in a 5-Point algorithm scheme to compute relative transformations, which are subsequently compared against the localization ground truth, yielding the error metric to "feed" an RL's reward function. Despite its dynamic training scheme, this approach still depends on ground truth data, which are yet hard to be obtained. On the contrary, our method, computes directly the RL's reward as the percentage deviation of accurately matched inlier keypoints and the total keypoints detected, based on the computed camera transformation among consecutive frames. Furthermore, beyond implementing the baseline of the RL architecture, we incorporated a more sophisticated and efficient training method from the broader RL's algorithmic family, the Actor-Critic. We opt for this approach with the insight of letting the Actor to learn a full action set, while the Critic constantly evaluates its totality, providing real-time feedback to refine feature selection for a complete sequence of input camera measurements.

III. Methodology

In our work, we adopt a hybrid methodology that leverages the power of DL to explicitly learn the detection of local features that are more likely to contribute to the vSLAM engine, by respecting the geometrical constrains of a moving camera. The subsequent subsections, along with Fig. 2, describe the implementation details of our Actor-Critic RL framework. Each network plays an independent role within the framework. In general terms, the Actor learns the policy, while the Critic evaluates the Actor's performance over a series of implemented actions. Within the context of our research, the Actor decides which keypoints are persistent and trackable through a sequence of frames, while the Critic estimates the value of those decisions for long-term reward. In addition, the reward is defined as the percentage of the detected keypoints that respect a computed camera transformation, among the total keypoints found by the Actor.

A. Actor Network for Feature Points Detection

The primary objective of our methodology is to train a feature extraction framework capable of detecting robust, stable, and distinctive keypoints for vSLAM, even under adverse environmental conditions. To achieve this, we adopt SuperPoint as our Actor's network; a widely acknowledged architecture in the literature. The output of SuperPoint corresponds two heads: i) a heatmap of local features scores, which can be thresholded to provides direct sets of feature points, and ii) a fully connected layer of the corresponding descriptors. This architecture allows it to be straightforwardly applied on the majority of vSLAM architectures. However, alternative end-to-end trainable models can also be adopted without any modifications to the described approach.

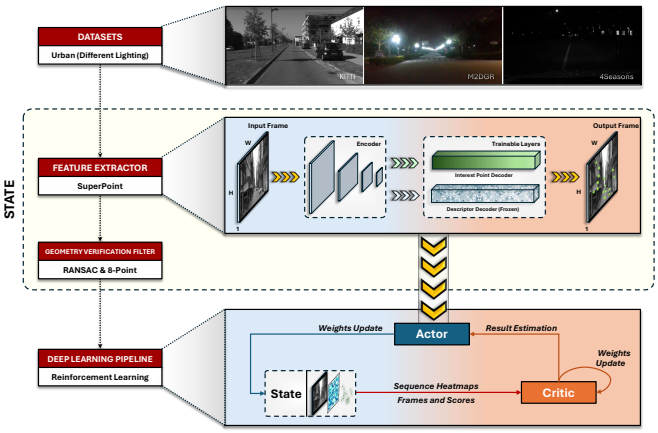


Fig. 2. An overview of the proposed methodology. Given the input camera stream from a specific environment, feature extraction is performed based on a pre-trained model of SuperPoint. The detected keypoints are matched with the subsequent ones from the respective image sequence, based on a geometry validation step that employs the 8-Point algorithm under the RANSAC scheme. Finally, the percentage of surviving keypoints is used as a reward in an Actor-Critic architecture to refine the weights.

To better suit our objectives, we deliberately break the joint optimization between the detector's and the descriptor's weight branches during learning, by freezing the descriptor's weights and leaving only the keypoint detectors available for training. One key insight of the above decision is to stabilize training and offer faster convergence with less chance of overfitting the whole network. Another benefit is that by freezing the descriptors, the network model will not need to adapt and respond to weak descriptors during training, and it will be forced to produce more stable and distinctive keypoints. Finally, we incorporate a pre-trained variant of the aforementioned network, available in [1].

B. RL Reward Function Based on Geometrically Verified Matches

The incoming frames of a robotic agent's RGB camera stream (state of the RL scheme) are partitioned into sequences of size s with overlap. Upon completion of sequence S, the Actor's action is to detect the local keypoints and their descriptors on every image-member (I_1, I_2, ..., I_s) in S. Those features are then matched among the image-members and the 8-Point is used for estimating the fundamental matrix from their point correspondences. To refine this estimation, the RANSAC framework is utilized for identifying the fundamental matrix assumption with the most inliers. Since the computation of the fundamental matrix is directly linked to the inter-frame camera transformations, the above procedure

[1] SuperPoint repo: github.com/magicleap/SuperPointPretrainedNetwork

serves as a filter in order to exclude the detected outlier keypoints and award the RL pipeline with the inlier ones that are most likely to contribute to a vSLAM module.

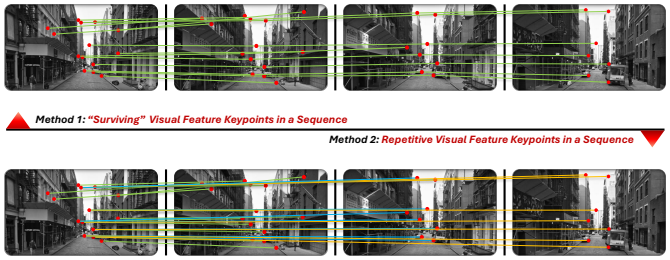


Fig. 3. An overview of the proposed reward approaches. The first method evaluates the keypoints survive from the first to the last image within a recorded sequence of size s. The second one measures the keypoints are matched among the first image and the rest sequence members.

To formulate the reward calculation, two distinct approaches are assessed as shown in Fig. 3. In the first one, starting with I_1, we establish correspondences with the subsequent frame, by following the geometrically consistent matching described above. The resulting set of keypoints is then matched to I_3, and the process is repeated until the final frame of the sequence I_s is reached. The following recursive function describes the denotation of the subset of features that have survived the geometrically consistent matching, from I_1 up to I_s:

$$\begin{aligned} T_{srv}(I_1) &= D(I_1), \\ T_{srv}(I_i) &= M(T_{srv}(I_{i-1}), D(I_i)), \, for \, i \in \{2, ..., s\}, \end{aligned} \tag{1}$$

where $D(I_i)$ corresponds to the set of detected keypoints from frame I_i, and $M(A, B) \subseteq B$ returns the subset of B, whose elements admits a geometrically verified match with at least one element of A. Based on this interpretation, the reward function is computed as:

$$R_{srv_{i,i+1}} = \lambda \frac{|T_{srv}(I_{i+1})|}{|T_{srv}(I_i)|}, \, for \, i \in \{1, ..., s-1\} \, and \, \lambda \in \mathbb{R}^+, \tag{2}$$

where $|...|$ denotes the cardinality of each set, and λ is a weighting term that balances the contribution of this component to the overall objective.

Our second approach for computing the reward function is more lenient, and it is based on a pair-wise matching among the first frame I_1 and the rest of the remain images in S.

$$\begin{aligned} T_{cmp}(I_1) &= D(I_1), \\ T_{cmp}(I_i) &= M(T_{cmp}(I_1), D(I_i)), \, for \, i \in \{2, ..., s\}. \end{aligned} \tag{3}$$

More specifically, the number of geometrically consistent matches from I_1 and every other member of S is computed, and the reward function is formed as:

$$R_{cmp_{i,i+1}} = \lambda \frac{|M(I_1, I_{i+1})|}{|T_{cmp}(I_1)|}, \, for \, i \in \{1, ..., s-1\} \, and \, \lambda \in \mathbb{R}^+. \tag{4}$$

C. Critic Network

We introduce a novel model design for implementing the Critic network's architecture. A CNN pipeline is proposed for the evaluation of the reward's quality. The algorithmic structure consists of convolutional layers followed by batch normalization layers, activation function, max-pooling layers, and with fully connected layers at the end to reward or penalize the Actor's actions. Figure 4 provides a detailed representation of the Critic's network configuration.

In order to evaluate Actor's actions, the Critic's pipeline initiates with the acquisition phase and proceeds through several transformative steps, culminating in the final output. After the completion of each sequence S, where the processes of extracting keypoint detections from the frames, matching geometrically verified feature pairs, and calculating their respective rewards, it is time for these values to be fed into the Critic network in order to evaluate the current action. Step one entails computing the Tempolar Difference Error (TDE). The equation below represents a discounted sum of future rewards, where the goal is to estimate how beneficial each action is considering both immediate and future rewards in a recursive way:

$$\begin{aligned} \delta(t-1) &= R_X(t-1), \\ \delta_{i,i+1} &= R_{X_{i,i+1}} + \gamma \cdot \delta(i+1), \, for \, i \in \{t-1, ..., 0\}, \end{aligned} \tag{5}$$

where R_X represents either reward functions described in Section III-B, $\delta_{i,i+1}$ are the discounted returns, $\delta(t-1)$ is essentially a starting point for the recursion loop, and γ is the discount factor controlling the future rewards' weighting in the calculation. After calculating the returns, normalization is applied to achieve a mean of 0 and a standard deviation of 1. This helps to stabilize training by ensuring a confined range of values:

$$\|\delta_{i,i+1}\| = \frac{\delta_{i,i+1} - \mu_\delta}{\sigma_\delta + c}, \, for \, i \in \{t-1, ..., 0\}, \tag{6}$$

where $\|\delta_{i,i+1}\|$ is the normalized discounting sum term, μ_δ is its mean value, σ_δ is its standard deviation, and c is a small constant to avoid division by zero (in our implementation, this value was set to $1 \cdot 10^{-5}$). Subsequently, the above computed normalized discount sum, together with the heatmap scores of each detected local feature, are used as inputs into the Critic network, in order to estimate the Actor's performance in facilitating vSLAM keypoints extraction:

$$P_{i,i+1} = C(H_i, H_{i+1}, \|\delta_{i,i+1}\|), \, for \, i \in \{1, ..., s-1\}, \tag{7}$$

where P is the predicted reward from the Critic, C is the Critic's network, and H is the heatmap scores of each detected feature point.

Once the predicted reward is extracted, the losses for the Critic and Actor Networks are computed. For the Critic loss l_{cr}, the Smooth L1 Loss, also known as Huber Loss, is used to ensure that the loss function will be less sensitive to outliers and prevent exploding gradients:

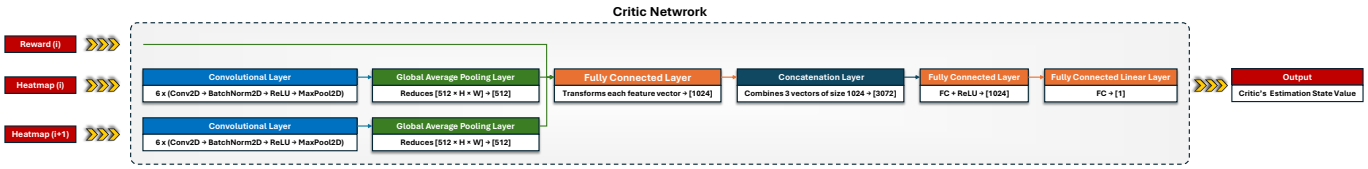


Fig. 4. Graphical representation of Critic's architecture. After the initial Convolutional Layers, a Global Average Pooling Layer reduces the spatial dimensions, and the output is passed through three Fully Connected Layers, combining features from the heatmaps and quality score inputs. The final layer outputs a scalar value, used for the Actor's evaluation during training.

$$l_{cr_{i,i+1}} = \begin{cases} 0.5\left(P_{i,i+1} - \|\delta_{i,i+1}\|\right)^2/\beta, & |P-\|\delta\||<\beta \\ |P_{i,i+1} - \|\delta_{i,i+1}\|| - 0.5\beta, & \text{otherwise,} \end{cases} \tag{8}$$

where β typically represents a threshold or scale parameter that determines when the loss behaves like L1 or L2 loss.

For the Actor's loss l_{ac}, we first need to calculate the Advantage Function A, which measures how much better or worse the Action is compared to the Critic's predicted reward. To encourage exploration by discouraging deterministic policies, our method integrates policy entropy distribution as part of the optimization process and policy's behavior shape. The entropy is calculated using the Softmax Function, which gives a probability distribution over the possible actions based on the heatmaps scores:

$$\begin{aligned} A_{i,i+1} &= \|\delta_{i,i+1}\| - P_{i,i+1}, \, for\, i \in \{1, ..., s-1\}, \\ E &= -\sum_{i}^{s-1} H_i \cdot \log H_i, \, for\, i \in \{1, ..., s-1\}. \end{aligned} \tag{9}$$

Then, the loss is a combination of two terms. The first term represents the policy gradient where the advantage is multiplied by the action probabilities. The second term is the entropy regularization, scaled by a factor b (in our model, the value is set to 0.2), to encourage exploration by discouraging deterministic policies:

$$l_{ac_{i,i+1}} = -A_{i,i+1} \cdot H_i - bH_i, \, for\, i \in \{1, ..., s-1\}. \tag{10}$$

In the final step, each image-pair's individual loss in the sequence, both for the Actor and the Critic, is summed and backpropagated through each network. This step ensures that both networks are optimized based on the gradients of their respective losses.

IV. Experiments

A. Datasets Selection

To evaluate our approach, we made use of several publicly available datasets relevant to the real-world vSLAM applications under challenging conditions. The criteria considered include factors such as environmental diversity, purity and illumination, dynamic objects, and sequence length.

The KITTI Dataset [27] is chosen as an essential starting point, providing a foundational perspective on vSLAM's performance and evaluation of our proposed framework. It provides high-quality recorded frame sequences and accurate localization ground truth, making it highly relevant for training our keypoint detector.

To assess the performance of our learned features in the context of datasets characterized by low lighting conditions, a night-time subset of the Oxford RobotCar Dataset [28] was chosen. It introduces the necessary diversity and challenge, including poor visibility, low textures, and lighting inconsistencies, through the inclusion of artificial light sources, such as streetlights, vehicle headlights, and shadows. Additionally, it contains significant occlusions, like cars, pedestrians, or street signs, that can disrupt feature detection.

Finally, the 4Seasons Dataset [29], and more specifically the night-time segment of the dataset, represents the most challenging case of our evaluation. The subset provides extremely near-dark and night-time sequences conditions under which most vSLAM and feature detection methods are stretched to their limits. The primary visual content consists of frames that are dominated by darkness, with minimal features apart from sparse illumination from streetlights, vehicle headlights, and faint lane markings.

We configured SuperPoint's algorithm to disregard the cars' hood from both the Oxford RobotCar Dataset and 4Seasons, avoiding any distortion of the results. Therefore, the bottom 160 pixel rows were cropped from all frames of the Oxford RobotCar Dataset, and 100 pixel rows from the 4Seasons.

B. Experimental Protocol

To provide conclusive evidence supporting the validity of our novel methodology, we applied a comprehensive and tightly regulated experimental procedure.

1) Dataset Partitioning: Each dataset underwent partitioning into *training*, *validation*, and *testing* components. The *training* sequences were further divided into two partitions such that 80% was allocated for actual model training and the remaining 20% for validation, allowing us to monitor convergence behavior and ensure the model's performance remained consistent. In the *training* stage, the KITTI 00 (KITTI Dataset) and Old Town 03 (4Seasons Dataset) subsets were selected.

In order to provide a fair evaluation scheme and report representative results that are not directly affected by the implementation choices of our method, an independent *evaluation* sequence for each dataset was employed to assess the influence of our system's components on its final performance; namely, the two reward functions described in Section III-B, the frame sequences size s, and the number of features required to successfully achieve vSLAM. In such a way, the reported performance is not directly influenced by our method's optimization. During this stage, the KITTI 06 (KITTI Dataset)

TABLE I
PERFORMANCE EVALUATION OF THE MOST REPRESENTATIVE CONFIGURATIONS FOR THE PROPOSED RL ARCHITECTURE ON THE EVALUATION DATASETS. NOTATIONS IN **BOLD** HIGHLIGHT THE BEST PERFORMING CONFIGURATIONS AND RESPECTIVE RESULTS.

RMSE of ATE	**KITTI Dataset** KITTI 06								**Oxford RobotCar Dataset** Night 2014-11-14 (Seq. 06)			
	$\lambda = 1$				$\boldsymbol{\lambda = 0.6}$				$\lambda = 1$		$\boldsymbol{\lambda = 0.6}$	
	2000 Features		500 Features		2000 Features		**500 Features**		500 Features		**500 Features**	
	R_{srv}	R_{cmp}	R_{srv}	R_{cmp}	R_{srv}	R_{cmp}	R_{srv}	$\boldsymbol{R_{cmp}}$	R_{srv}	R_{cmp}	$\boldsymbol{R_{srv}}$	R_{cmp}
Orig. SuperPoint	6.0949		3.7821		6.0949		3.7821		142.2156			
RL Seq. 4	4.7951	7.5716	2.5214	3.1006	2.1022	1.1145	2.8254	6.1134	107.1950	103.9883	108.6744	104.3625
RL Seq. 6	6.9793	7.5397	1.0067	1.2850	6.8276	2.2506	3.6193	5.8046	126.2964	111.7400	100.2854	108.2382
RL Seq. 8	3.2170	3.1215	6.5994	3.2047	2.5788	3.5799	5.9269	1.9975	93.7988	121.4565	120.9922	113.9244
RL Seq. 10	5.9257	6.0429	6.6710	3.2257	3.4640	6.0180	4.9629	5.4939	102.6827	126.7368	**87.6867**	103.8103
RL Seq. 12	3.9553	7.2241	7.0075	3.4064	8.5982	2.6419	5.1632	4.9357	97.2748	107.2545	109.5952	107.3652
RL Seq. 14	1.6735	2.8097	6.1458	4.4746	1.3075	6.7910	7.4187	**0.8408**	88.5257	101.2120	102.3565	116.2311
RL Seq. 16	7.9809	4.5950	6.5548	3.3162	2.4412	4.6457	2.4943	1.6361	119.6435	109.1168	115.0317	115.6323
RL Seq. 18	4.2875	5.5911	4.7913	1.3925	8.0202	7.7058	3.8707	5.3165	89.3586	110.3494	106.3562	113.6092

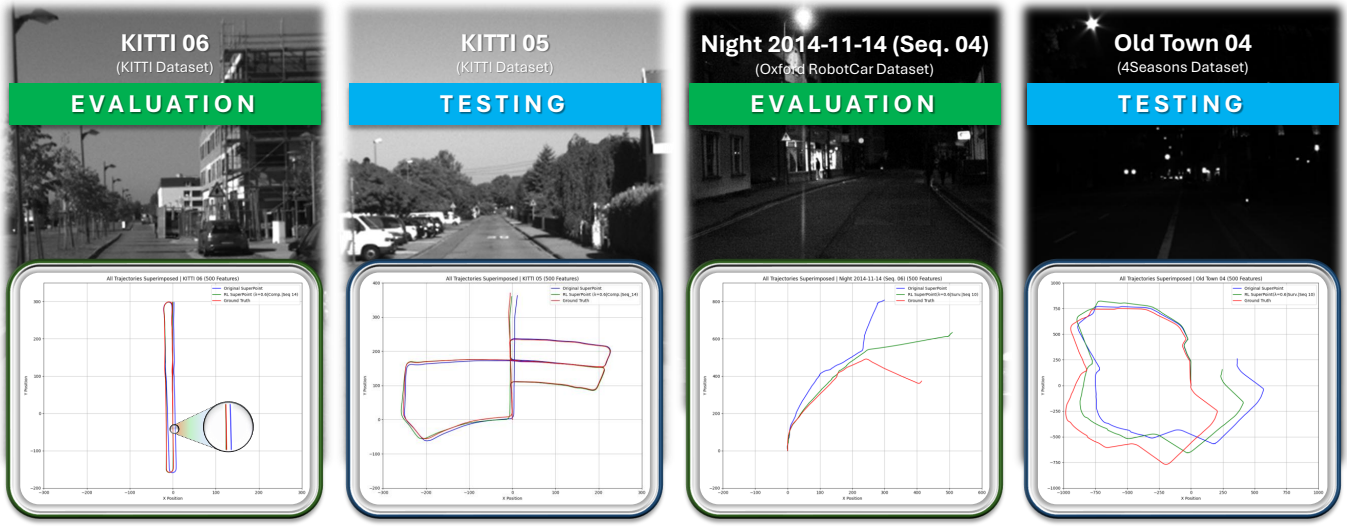


Fig. 5. Qualitative comparative results for the *evaluation* and *testing* datasets. The localization results, obtained thought the best performing parameters found during our *evaluation* experiments, are shown in green, while the ones produced through the original Superpoint implementation are presented in blue. The ground truth trajectories of the respective datasets are presented in red.

and Night 2014-11-14 (Seq. 04) (Oxford RobotCar Dataset) subsets were chosen. For the case of 4Seasons Dataset, only one of the provided sequences offers localization ground truth data; thus, the validation stage was once again performed on the above sequence of Oxford RobotCar since its subset closely resembles the characteristics of the 4Seasons one. Note that the ground truth information is only used for assess the performance of each approach and not for the actual keypoint detection training.

The *testing* dataset sequences were reserved in order to assess the generalization capabilities of our best performing trained architecture in different sequences of similar conditions. In addition, they also serve as comparative scenarios between the proposed RL-based SuperPoint feature extraction method and the original one. Specifically, the KITTI 05 sequence was selected for measuring the final testing accuracy of the KITTI Dataset and the Old Town 04 for the 4Seasons one.

2) Training Procedure: The training process is preceded each time by the selection of the batch size, learning rates, number of warm-up epochs, step schedulers, and accuracy score type. The three different batch sizes were evaluated, viz., 16, 32, and 64, allowing the model to accommodate sequences of different lengths. We additionally assessed two separate learning rates a in our Actor-Critic architecture, by implementing Adam Optimizer. Typically, α_A is set lower than α_C due to the higher variance in policy gradient estimates. In our setup, the α_A was set to $2.5 \cdot 10^{-10}$ and α_C to $2.5 \cdot 10^{-8}$. To prevent large, unstable updates in the early stages of training, where the agent's policy is still unrefined, warm-up epochs with even smaller learning rate than the starting one are used. In this way, the agent makes controlled adjustments, reducing the risk of diverging due to random and noisy reward signals. Our warm-up phase consist of 2 epochs, where the learning rate is incrementally increased from a smaller value to the desired one. Furthermore, during

TABLE II
COMPARATIVE RESULTS BETWEEN THE ORIGINAL SUPERPOINT AND THE WEIGHTS PRODUCED THROUGH THE PROPOSE RL FRAMEWORK.

Dataset	**KITTI 05**		**Old Town 04**	
SuperPoint Ver.	**Original**	**RL***	**Original**	**RL****
RMSE in axis X	3.9888	2.8403	250.5834	130.6972
RMSE in axis Y	3.9330	3.4396	113.6348	65.2899
Mean ATE	4.1954	3.0803	206.2941	119.4631
RMSE of ATE	5.6017	4.4608	275.1452	146.0928

*max features: 500, seq. size: 14, $\lambda = 0.6$, R_{cmp}
**max features: 500, seq. size: 10, $\lambda = 0.6$, R_{srv}

the training procedure, we further applied step-schedulers in order to strike a balance between fast exploration and stable convergence. In our configuration, α_A is updated according to a fixed rate of 60% every 10 epochs, while α_C is updated one epoch later, with the same percentage reduction, allowing the Critic to adapt on the new Actor's behavior. Finally, training is guided by a convergence threshold and terminated once the Actor Loss falls quietly below a predefined boundary of 0.005, indicating sufficient convergence.

3) vSLAM Evaluation: For systematic benchmarking and comparative analysis across the aforementioned different configurations and datasets, we made use of the pySLAM tool[2]. The tool is designed with rapid testing in mind, offering a comprehensive evaluation pipeline that supports multiple feature extractors and SLAM configurations and integration with modern deep learning models. In our experiments, we employed the monocular pySLAM's VO module to estimate camera motion trajectory, by using the proposed RL SuperPoint features and measuring the Root Mean Square of the Absolute Trajectory Error (RMS ATE) of each trained configuration. Using the above-stated *evaluation* sequences from Sections IV-A and IV-B1, we cross-evaluated 8 different sequence sizes $s \in \{4, 6, 8, 10, 12, 14, 16, 18\}$, as well as the reward function described in Section III-B and Eq. 2, 4 with two different weight contribution λ values 1.0 and 0.6. Moreover, we measured the VO SLAM performance by imposing two thresholds on the maximum number of features extracted, namely 2000 and 500, to evaluate capacity of our approach for learning and promoting the most suitable keypoints for vSLAM applications. Finally, all these parameters were applied to both proposed rewards functions described in Section III-B. Table I summarizes the results we obtained from each *evaluation* dataset. As it can be seen, the system remained remarkably effective with even fewer features, occasionally yielding superior results, demonstrating that our geometrically verified training scheme is able of identified the most suitable local keypoints for reliable localization.

C. Results

The best performing configurations, as identified in Section IV-B3 were used to test our final system's performance in the designated *testing* datasets. Table II presents the obtained results, as compared against the original SuperPoint weights for feature selection. Qualitative results are also shown in Fig. 5 for all the selected datasets, where the estimated trajectories are included, together the localization ground truth. In every case under consideration, the evidence consistently supports that our method successfully estimates the overall trajectory of the moving platform, outperforming the outcome of the original SuperPoint network and evidently managing to identify less, but more consistent and geometrically valuable local keypoints for vSLAM. Evidently, the localization performance improvement is more profound in the sequences of dark environmental conditions since the detection of local keypoints (especially during abrupt turns) is more challenging compared to the rest of the cases.

[2]pySLAM repo: github.com/luigifreda/pyslam

V. CONCLUSION

Framed within the context of modern vSLAM challenges, the presented research establishes a sound methodological approach for robust feature extraction under light and adverse-dark visual conditions, by providing a coherent integration of RL and vSLAM. Our proposed system takes advantage of the Actor-Critic learning scheme's capacity, while considers the feature detections within a sequence of constituent images as the network's action. Then, a reward function, specifically designed to inherit well-established 3D-geometry principles, is used to assess these actions and allow the final network to explicitly learn the detection of the most meaningful vSLAM keypoints, tailored to each individual environment's conditions. Our experimental results demonstrate the superiority of our approach as a learning scheme, surpassing the corresponding performance of the original feature point detection architecture.

Further efforts will be directed towards the implementation and inclusion of the previously "frozen" descriptors in the RL training process. The integration of the proposed approach into a continuous learning architecture will also be evaluated in order to support the deployment of robotic agents across a wider variety of sites, by adapting on-the-fly their vSLAM capabilities to the characteristics of each individual environment.

ACKNOWLEDGMENT

This research is implemented in the framework of H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union – NextGenerationEU (H.F.R.I. Project Number: 15339).

REFERENCES

[1] R. T. Azuma, "A Survey of Augmented Reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 08 1997.
[2] Z. Makhataeva and H. A. Varol, "Augmented Reality for Robotics: A Review," *Robotics*, vol. 9, no. 2, 2020.
[3] H. Liu, G. Zhang, and H. Bao, "Robust Keyframe-based Monocular SLAM for Augmented Reality," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2016, pp. 1–10.
[4] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual Simultaneous Localization and Mapping: A Survey," *Artificial Intelligence Review*, vol. 43, pp. 55–81, 2015.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
[6] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
[7] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The Revisiting Problem in Simultaneous Localization and Mapping: A Survey on Visual Loop Closure Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19 929–19 953, 2022.
[8] A. Agakidis, L. Bampis, and A. Gasteratos, "Illumination Conditions Adaptation for Data-Driven Keypoint Detection under Extreme Lighting Variations," in *Proceedings of the IEEE International Conference on Imaging Systems and Techniques (IST)*, 2023, pp. 1–6.
[9] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
[10] R. Hartley, "In Defense of the Eight-Point Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.
[11] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981.
[12] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.
[13] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.
[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870.
[15] I. Kansizoglou, L. Bampis, and A. Gasteratos, "Deep Feature Space: A Geometrical Perspective ," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 44, no. 10, pp. 6823–6838, Oct. 2022.
[16] C. Kenshimov, L. Bampis, B. Amirgaliyev, M. Arslanov, and A. Gasteratos, "Deep Learning Features Exception for Cross-Season Visual Place Recognition," *Pattern Recognition Letters*, vol. 100, pp. 124–130, 2017.
[17] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
[18] Z. Zhao, C. Wu, X. Kong, Q. Li, Z. Guo, Z. Lv, and X. Du, "Light-SLAM: A Robust Deep-Learning Visual SLAM System Based on LightGlue under Challenging Lighting Conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 7, pp. 9918–9931, 2025.
[19] H. M. S. Bruno and E. L. Colombini, "LIFT-SLAM: A Deep-Learning Feature-Based Monocular Visual SLAM Method," *Neurocomputing*, vol. 455, pp. 97–110, 2021.
[20] Y. Wei, S. Lu, W. Lu, R. Xiong, and Y. Wang, "BEV-DWPVO: BEV-Based Differentiable Weighted Procrustes for Low Scale-Drift Monocular Visual Odometry on Ground," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4244–4251, 2025.
[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2564–2571.
[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008, similarity Matching in Computer Vision and Multimedia.
[23] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[24] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," in *Proceedings of Neural Information Processing Systems*, vol. 31, 2018.
[25] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "ASLFeat: Learning Local Features of Accurate Shape and Localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6589–6598.
[26] A. Bhowmik, S. Gumhold, C. Rother, and E. Brachmann, "Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI Vision Benchmark Suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
[28] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.
[29] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers, "4Seasons: A Cross-Season Dataset for Multi-Weather SLAM in Autonomous Driving," in *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020.

Panagiotis Bakirtzis received the Diploma degree from the Department of Electrical and Computer Engineering, Democritus University of Thrace (DUTh), Xanthi, Greece, in 2023. Currently, he is working towards the Ph.D. degree and as a Research Assistant in the Laboratory of Mechatronics and Systems Automation (MeSA), DUTh. His research is focused on Robotics Vision, SLAM, and Deep Learning for autonomous systems, with a long-term focus on advancing next-generation solutions for the defense and aeronautics industry.

Ioannis Kansizoglou received the Diploma degree in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2017, and his Ph.D. in deep representation learning and computer vision from the Laboratory of Robotics and Automation, Department of Production and Management Engineering, Democritus University of Thrace, Xanthi, Greece, in 2021. He is currently a Post-Doctoral Researcher within the same Laboratory, and his research interests include deep representation learning, emotion analysis, and human-robot interaction. His work is supported by several research projects funded by the European Commission and the Greek Government.

Loukas Bampis received the Diploma degree in Electrical and Computer Engineering and his Ph.D. in machine vision and embedded systems from the Democritus University of Thrace (DUTh), Xanthi, Greece, in 2013 and 2019, respectively. He is currently an Assistant Professor with the Laboratory of Mechatronics and Systems Automation (MeSA), Department of Electrical and Computer Engineering, DUTh. His work has been supported through several research projects funded by the European Space Agency, the European Commission, and the Greek Government. His research interests include real-time mapping, localization, and place recognition techniques for unmanned autonomous vehicles.