Highlights

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

Anastasios Agakidis, Antonios Gasteratos, Loukas Bampis

- A novel Illumination Condition Adaptation (ICA) method aimed to improve illumination invariance.
- A new Photo-Realistic Synthetic Illumination (PRSI) dataset enhances training.
- ICA refines feature points for learning-based detection under extreme lighting changes.
- ICA improves SLAM-based localization night conditions and dynamic illumination shifts.
- Extensive experiments validate ICA's impact on VO accuracy and real-world datasets.

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

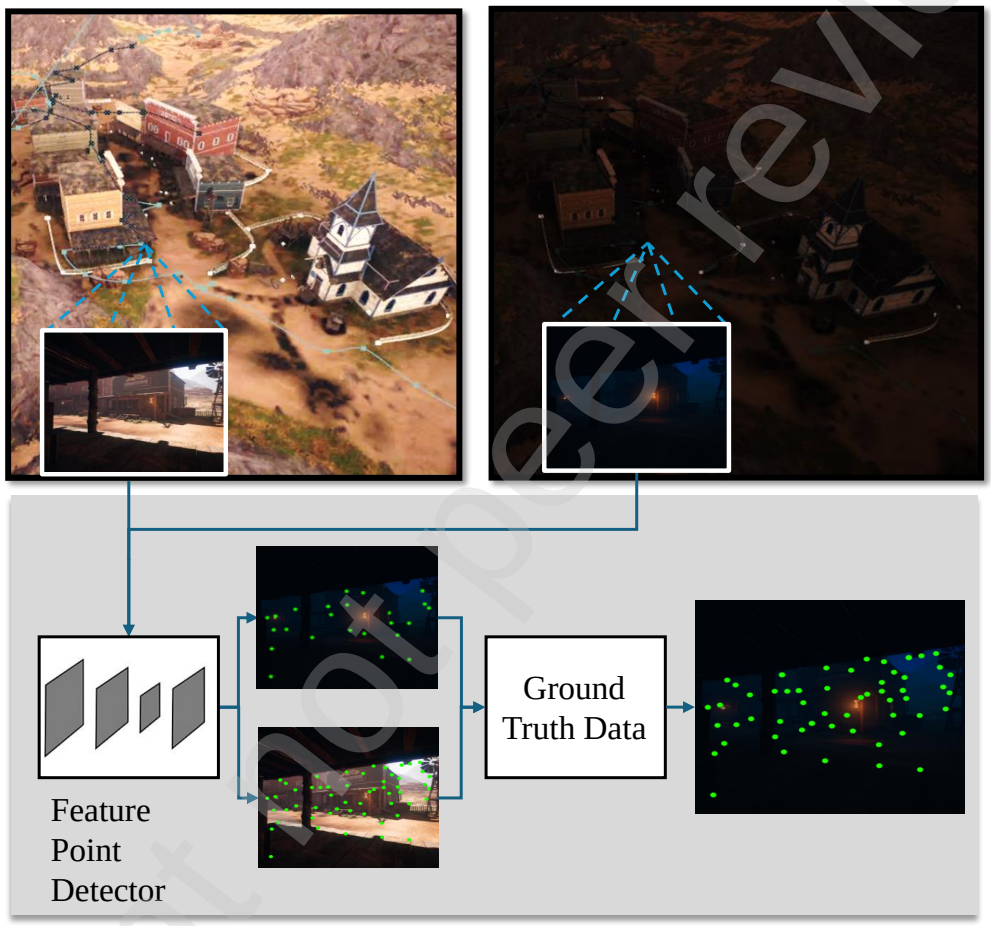


Figure 1: Illustration of our proposed Illumination Conditions Adaptation (ICA) method. Features are detected on two identical views with different illumination conditions. They are combined, filtered, and then used as ground truths for the subsequent training.

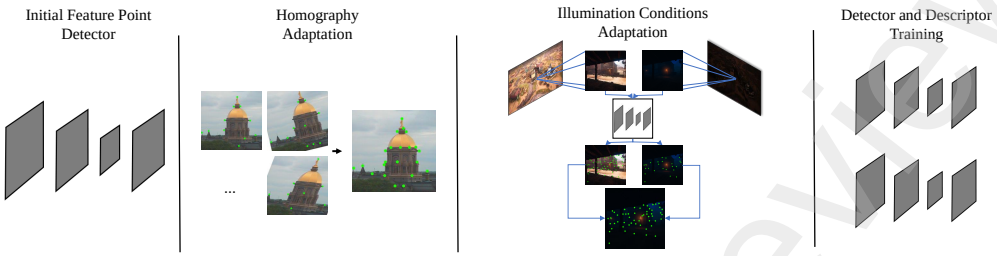


Figure 2: Schematic representation of Illumination Condition Adaptation (ICA) integration into the selected deep local feature extraction model. Initially, keypoints are detected from the daytime and the equivalent nighttime image using the pre-trained detector. Homographic adaptation is then applied 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`.

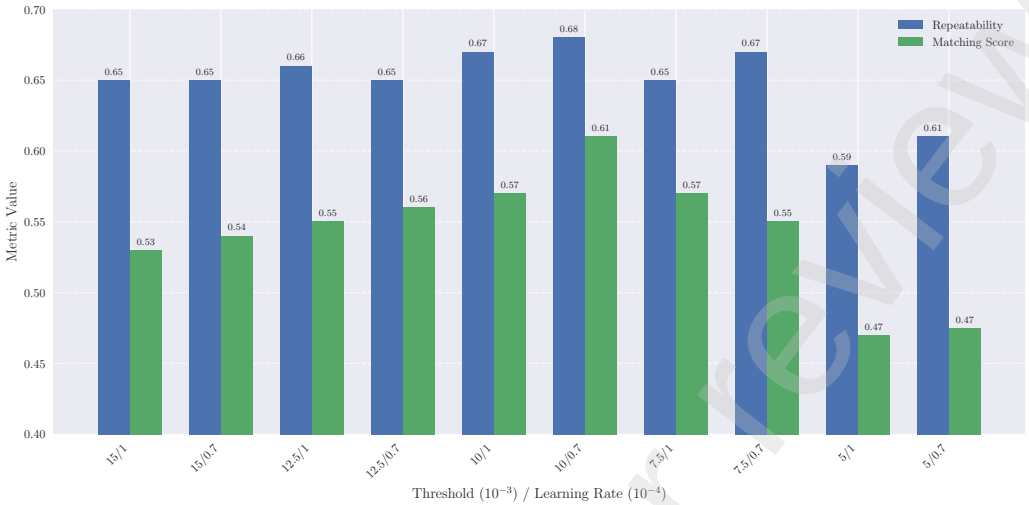
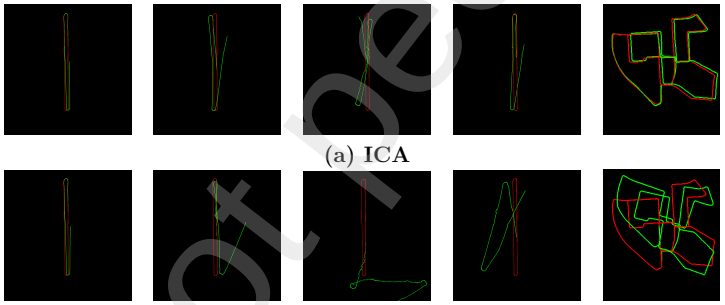


Figure 6: Performance metrics across different thresholds and learning rates.



(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.

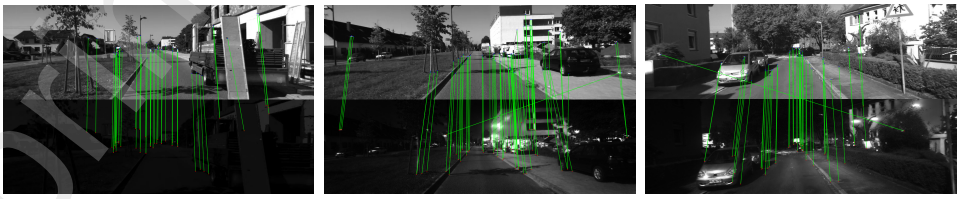


Figure 8: Feature matching results between day and night images, where the night image is two frames ahead. From left to right: `kitti06` with `dark-kitti06`, `kitti06` with `darker-kitti06`, and `kitti00` with `dark-kitti00`.