博 士 学 位 論 文

論文題目　Deep Learning Methods for Robust Image Matching and Visual Localization

提　出　者　　東北大学大学院情報科学研究科

システム情報科学専攻　　専　攻

学籍番号　　B9ID2506

氏　名　　Wenzheng SONG

.

TOHOKU UNIVERSITY
Graduate School of Information Sciences

Deep Learning Methods for Robust Image Matching and Visual Localization
(頑健な画像間対応付け及び視覚的位置推定のための深層学習手法)

A dissertation submitted to the department of
System Information Science in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
in
Information Sciences

by

Wenzheng SONG (ID No.: B9ID2506)

December 27, 2022

Deep Learning Methods for Robust Image Matching and Visual Localization

Wenzheng SONG (ID No.: B9ID2506)

Abstract

Structure-from-motion (SfM) and visual simultaneous localization and mapping (SLAM), which seek to address the issues of localization and 3D reconstruction, have been utilized in real-world applications for an extended period. These techniques play a vital role in numerous applications, including indoor sweeping machines and mobile robots, unmanned aerial vehicles (UAVs), self-driving cars, virtual reality (VR), and augmented reality (AR). For instance, robots utilize these methods to determine their position in an unfamiliar environment and generate a map. In recent years, with the rapid advancement of deep learning and computer vision, new approaches based on deep neural networks (DNNs) have been developed for SfM and visual SLAM, yielding tremendous success.

Although visual simultaneous localization and mapping (SLAM) has achieved a certain level of success, numerous challenges remain. Specifically, the expanding range of applications requires reliable operation indoors and outdoors, regardless of weather, illumination, seasonal changes, or extreme environments such as low light conditions at night. The key to addressing these issues lies in the development of robust visual odometry (VO) techniques that can function in these challenging situations, such as VO utilizing a robust image-matching method. On the other hand, advances in image sensor technology have enabled the recording of incoming light with over eight bits (e.g., 14 bits). However, standard RAW image processing on many cameras cannot fully exploit the information contained in all bits of RAW signals. This limitation arises due to the need for adaptability in diverse scenarios with varying lighting conditions and the requirement to reduce the number of bits in order to compress image file size. Previous studies have demonstrated that the use of RAW images can yield significantly better results than RGB in image processing tasks such as image denoising and enhancement. With the advancement of hardware, the direct utilization of RAW images in downstream tasks is becoming increasingly promising.

In the first part of this work, we are interested in fully utilizing such high-precision information to match extremely low-light scene images that conventional methods cannot handle. For extremely low-light scenes, even if some of their brightness information is present in the low bits of RAW format images, standard image processing pipelines for

cameras fail to properly utilize it. As was recently demonstrated by Chen et al., convolutional neural networks (CNNs) can learn to produce images with a natural appearance from such RAW format images. To investigate the feasibility and efficacy of utilizing information stored in RAW format images for image matching, we have created a new dataset called MID (matching in the dark). Using this dataset, we experimentally evaluated combinations of eight image-enhancing methods and eleven image-matching methods comprising classical/neural local descriptors and classical/neural initial point-matching methods. The results show the advantage of using RAW format images and the strengths and weaknesses of the aforementioned component methods. They also suggest that there is potential for further research in this area.

In the second part of this work, we investigate the potential of utilizing the information stored in RAW images of low-light scenes to improve performance for downstream tasks related to 3D reconstruction. We observe that standard image enhancers are designed to produce images that appear the most natural, which may not necessarily be the best images for downstream tasks. To explore the potential of RAW images, we introduce a novel dataset comprising multi-view RAW images of extremely low-light scenes along with corresponding long-exposure versions and ground-truth depth. To the best of our knowledge, this is the first dataset created for 3D reconstruction that includes low-light RAW images. Existing datasets for image matching do not contain dark RAW format images of low-light scenes, whereas those that do include RAW format images do not meet the requirements for training image-matching models, such as a lack of relevant images, intrinsic/extrinsic parameters, and ground truth scene depths. In addition, we propose a novel image enhancer called SuperISP that enhances low-light RAW images to produce favorable inputs for downstream tasks. We further evaluate our method on two typical downstream tasks, namely image matching and monocular depth estimation, and demonstrate the advantage of SuperISP compared to other image enhancement methods.

Finally, in the third part of this work, we focus on the flexible use of image-matching models in a related field, namely visual localization. Advanced visual localization approaches address two distinct problems: image retrieval and six degree-of-freedom (DoF) camera pose estimation. Given a query image of the current view, these approaches first identify relevant candidates by performing image retrieval in a database, then perform local feature matching between the query image and relevant candidates to establish point correspondences while imposing the epipolar constraint and estimating the six DoF camera pose of the query image. Since these two problems can be conflicting at the feature level, previous studies have attempted to reconcile them. We propose a method that satisfies all requirements by addressing the feature-level conflicts for improved visual localization. The

HF-Net proposed by Sarlin et al. integrates two models for visual localization using model distillation. However, distillation can harm localization accuracy, particularly in challenging situations such as night-time scenes. Experimental results show the effectiveness of our method compared to existing methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an introduction about the problem we will discuss throughout this dissertation. We introduce an overview of background to illustrate the territory of the problem in Sec. 1.1. To encourage more understanding, we include the preliminary knowledge related to our study in Sec. 1.2. Lastly in this chapter, the outline of each individual chapter is described in Sec. 1.3.

1.1 Background

1.1.1 Image Matching

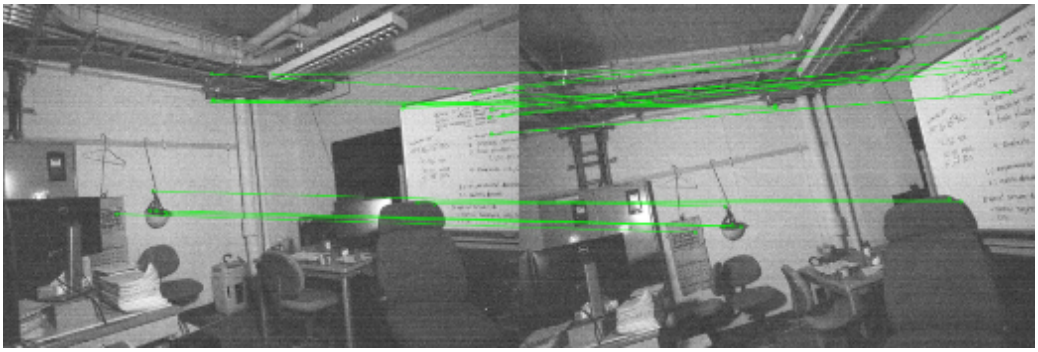
Matching two or more images of a scene is at the core of fundamental computer vision problems, including image retrieval [6, 8, 9], 3D reconstruction [10–13], visual localization [14–19], and SLAM [20–22]. Despite decades of research, image matching remains unsolved in the general, wide-baseline scenario. Image matching is a challenging problem with many factors that need to be taken into account, e.g., viewpoint, illumination, occlusions, and camera properties. The main object of image-matching is to identify corresponding points in two or more images. Give images of the same scene, features of detection and description are extracted from each image, establish initial matching of the corresponding points between the images, then obtain final matching results while imposing the epipolar constraint on them and removing outliers. Finally, these determined correspondences are used for estimating geometric transformations between images, such as homography estimation and 6 DoF (Degree-of-Freedom) camera pose estimation. In short, the stan-

dard approach of image matching encapsulates two steps, i.e., image feature detection and description and feature matching. There are many classical methods that do not rely on learning data. As with other computer vision problems, deep neural networks (DNN) have been applied to each step in recent years; see Sec. 1.2.

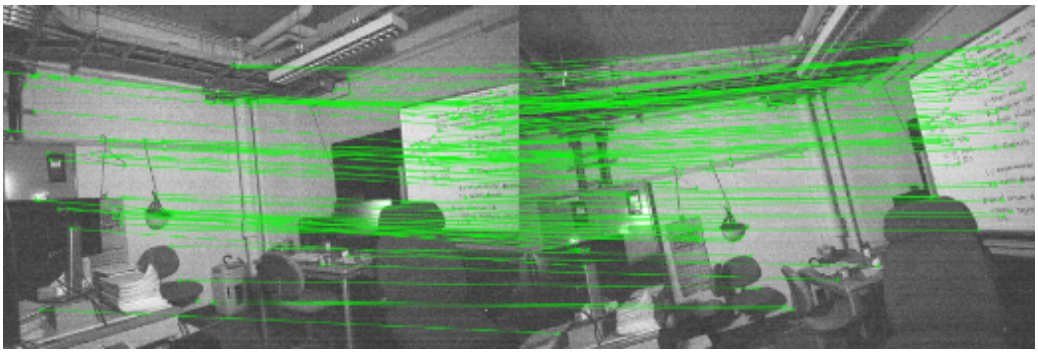
Image Feature Detection and Description This step's algorithm typically follows a *detect-then-describe* approach, which starts by extracting low-level features from the image, such as edges, corners, and blobs, by a detector, denoted as keypoints or interest points commonly. Then, a descriptor provides a compact representation for each patch around the keypoints. It has therefore been traditionally approached with sparse methods with local features, such as SIFT [8], SURF [23], and ORB [24], .etc. The local features provided by this step are typically expected for providing robustness or invariance against effects such as scale, rotation, or viewpoint changes. Due to requirements to improve such problems, recent efforts have moved towards holistic, learnable end-to-end solutions [2, 25–27].

Feature Matching The purpose of feature matching is to establish corresponding points between images. A typical approach is to perform the NN (Nearest Neighbor) or FLANN (Fast Library for Approximate Nearest Neighbors) [28] search in the descriptor's space for obtaining initial correspondences across images, with an optional *ratio-test* step for filtering out unreliable matches [8], and RANSAC for outlier removal. Morden's approaches for feature matching and robust estimation tend to be learned by DNNs [29, 30]. Recently, transformers have been applied to learn robust matchers and led to state-of-the-art results, e.g. SuperGlue [3] and LoFTR [31].

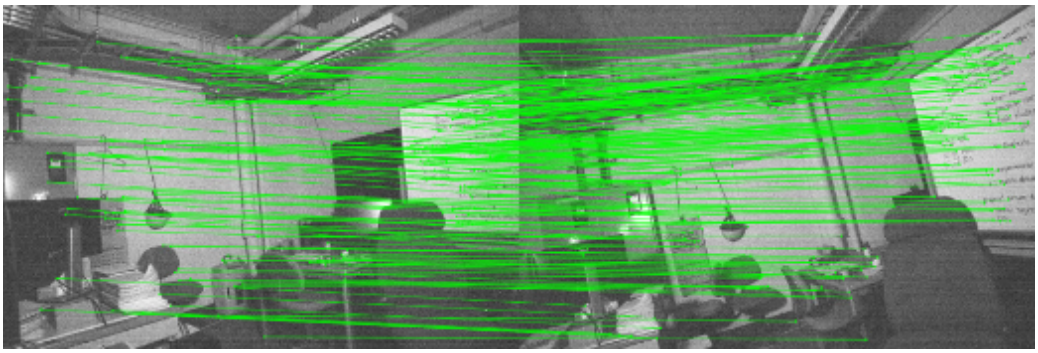
To meet the needs of practical applications in the real world, advanced image-matching methods are required to cope with kinds of challenging conditions, such as appearance changes, illumination changes, and noise. Recent studies [2, 26, 27] show significant advantages of learning-based methods in dealing with this problem. Figure 1.1 shows an example of a comparison between typical traditional and learning-based image-matching methods under noise conditions.



(a)



(b)



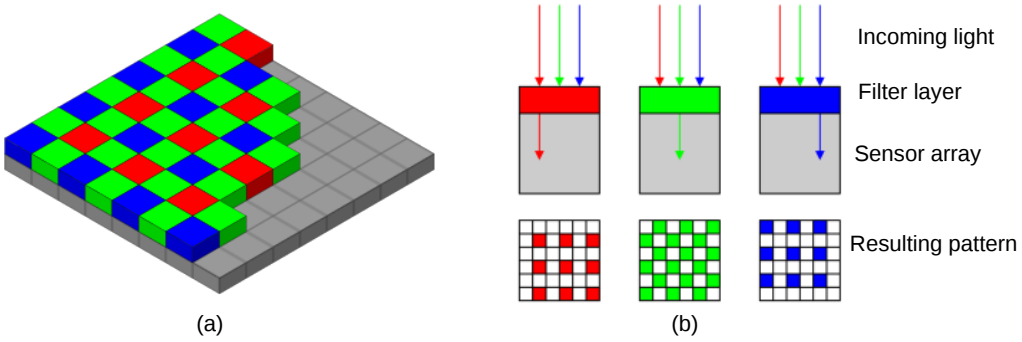
(c)

Figure 1.1: Comparison between image-matching methods; traditional: (a) RootSIFT [1]; and learning-based: (b) SuperPoint [2] and (c) SuperPoint+SuperGlue [3]. Point correspondences judged as inliers are shown in green lines.

1.1.2 Image Enhancement

Image enhancement technology is a commonly used way in digital image processing, which can improve image quality, highlight the useful information in the image according to people's actual needs, and suppress the redundant information in the image. So it has applications in many computer vision fields, which aim to provide high-quality images. Image enhancement methods are expected to cope with complex illumination conditions, external noises, and environmental disturbances such as ambient pressure and temperature fluctuations. Numerous image enhancement methods have been proposed over the last decades. Besides basic image processing such as histogram equalization, there are many methods based on different assumptions and physics-based models, etc., such as global analysis and processing based on the inverse dark channel prior [32, 33], the wavelet transform [34], the Retinex model [35], and illumination map estimation [36]. Recently, deep learning-based methods have dominated the image enhancement community [37–44].

In addition, the advancement of image sensors allows them to record incoming light with more than eight bits (e.g., 14 bits), and the unprocessed data from the camera is first stored in a RAW image. The standard approach to obtaining RGB images is to convert RAW images to RGB vair an Image Signal Processor (ISP). Although many optimization processes or enhancement methods are included in camera ISPs, this transformation process irreversibly converts RAW images with high-precision information into RGB images with lower-bit space and causes information loss. As a result, RGB images do not keep the complete information in RAW images, especially the information existing in the lower bits. This limitation arguably comes from the conflict between versatility against various lighting conditions and reducing the number of bits. To deal with more severely underexposed images, recent studies applied proposed learning-based methods that directly convert a low-light RAW image to a good-quality RGB image [5, 45, 46]. These methods can handle more severe image noise and color distortion emerging in underexposed images than the previous methods. Figure 1.2 shows the illustration of the RAW-format image and a comparison between RAW and RGB. Figure 1.3 shows an example of the output of RAW image-based enhancer, i.e., SID [5].



(a) (b)

	Channels	Bit	Data Size	Color Gradation
RGB	3	8	big	256
RAW	1	14	small	16348

(c)

Figure 1.2: (a) The Bayer arrangement of color filters on the pixel array of an image sensor; (b) Profile/cross-section of the sensor; (c) Comparison between RAW and RGB. (Image source: `https://en.wikipedia.org/wiki/Bayer_filter`)

(a) Camera output with ISO 8,000 (b) Camera output with ISO 409,600 (c) Result of SID from RAW data of (a)

Figure 1.3: An example of extreme low-light imaging with SID. Dark indoor environment. The illuminance at the camera is < 0.1 lux. The Sony α7S II sensor is exposed for 1/30 second. (a) Image produced by the camera with ISO 8,000. (b) Image produced by the camera with ISO 409,600. The image suffers from noise and color bias. (c) The result of SID applied to the raw sensor data from (a).

1.1.3 Visual Localization

Visual localization is the problem of estimating the 6 Degree-of-Freedom (DoF) camera pose from which a given image was taken relative to a reference scene representation. Robust and accurate visual localization is a fundamental capability for numerous applications, such as autonomous driving, mobile robotics, or augmented reality. It remains, however, a challenging task, particularly for large-scale environments and in presence of significant appearance changes. The approaches to visual localization can be summarized as below:

Visual localization using 3D maps. Most existing approaches to visual localization mostly rely on estimating correspondences between 2D keypoints in the query and 3D points in a sparse model using local descriptors. The map is usually composed of a 3D point cloud constructed via Structure-from-Motion (SfM) where each 3D point is associated with one or more local feature descriptors. The query pose is then obtained by feature matching and solving a Perspective-n-Point problem (PnP). However, the direct matching methods are either robust but intractable on mobile [47–49], or optimized for efficiency but fragile [50]. In both cases, the robustness of classical localization methods is limited by the poor invariance of hand-crafted local features [51, 52].

Image-retrieval-based localization. Visual localization in large-scale urban environments is often an image retrieval problem. The location of a given query image is predicted by transferring the geotag of the most similar image retrieved from a geotagged database [6, 53–58]. This approach scales to entire cities thanks to compact image descriptors and efficient indexing techniques [59–65] and can be further improved by spatial re-ranking [66], informative feature selection [60, 67] or feature weighting [56, 58, 68, 69]. Most of the above methods are based on image representations using sparsely sampled local invariant features. While these representations have been very successful, outdoor image-based localization has recently also been approached using densely sampled local descriptors [57] or (densely extracted) descriptors based on convolutional neural networks [6, 55, 70, 71]. However, all the above methods are not competitive in terms of accuracy because they output only an approximate location of the query, not an exact 6-DoF pose.

Visual localization using a hierarchical approach. Tackling the visual localization problem in a hierarchical manner encapsulates the advantages of both approaches [18], which can achieve efficient and accurate visual localization results even on large-scale databases. Prior retrieval allows feature matching to be restricted to a reasonable range rather than globally. It firstly saves computational resources and makes large-scale visual localization on mobile possible. Secondly, limiting feature matching to relevant frames can reduce the number of outliers of feature matching, thus increasing the probability of successful localization. Sarlin et al. distillate the NetVLAD network (the teacher) into a smaller one for prior retrieval. In the part of feature matching, they used hand-crafted descriptors [18]. Recent features emerging from convolutional neural networks (CNN) exhibit unrivaled robustness at low computing costs [2, 72, 73]. Taira *et al.* applied learning-based features to camera pose estimation of the visual localization problem, but in a dense, expensive manner [17]. Recently, HF-Net integrated NetVLAD and a learned sparse descriptor [19] by model distillation that simultaneously predicts keypoints as well as global and local descriptors for accurate 6-DoF localization.

1.2 Preliminaries

To facilitate understanding, we provide an overview of the relevant background knowledge in this section.

1.2.1 Epipolar Geometry

Epipolar Constraints For example, epipolar constraints of a pair of matched feature points from two images are illustrated in Figure 1.4. If there are several pairs of such matching points, the camera motion between the two frames can be recovered through the correspondence between these two-dimensional image points.

As shown in Figure 1.4, our goal is to find the motion between two frames, i.e., I_1, I_2. Define the motion from the first frame to the second frame as $\mathbf{R}$, $\mathbf{t}$, and the centers of the two cameras as O_1, O_2. Now, consider that there is a feature point p_1 in I_1, which corresponds to the feature point p_2 in I_2, obtained through feature matching. If the matching is correct, it means that they are indeed the projection of the same point. Now we need some

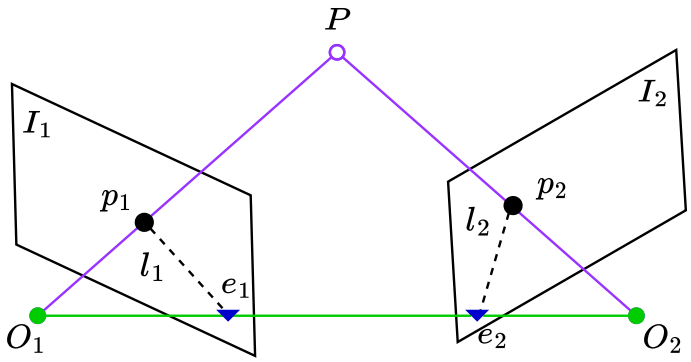


Figure 1.4: Example of epipolar constraints.

terms to describe the geometric relationship between them. First, the line $\overrightarrow{O_1p_1}$ and the line $\overrightarrow{O_2p_2}$ will intersect at the point P in the 3D space. The three points O_1, O_2, and P can determine a plane, and it is called the epipolar plane. The intersection of the line of O_1O_2 and the image plane I_1, I_2 is e_1, e_2, respectively. The e_1, e_2 is called epipoles, and O_1O_2 is called the baseline. We call the intersecting line l_1, l_2 between the polar plane and the two image planes I_1, I_2 as the epipolar line.

From the first frame, the ray $\overrightarrow{O_1p_1}$ represents the possible spatial locations where a pixel may appear since all points on the ray will be projected to the same pixel. Meanwhile, suppose we don't know the location of P. When we look at the second image, the connection $\overrightarrow{e_2p_2}$ (i.e., the epipolar line in the second image) is the possible projected positions of the point P , as well as the projection of the ray $\overrightarrow{O_1p_1}$ in the second image. Now, since we have determined the pixel location of p_2 through feature matching, we can infer the spatial location of P and the camera movement, as long as the feature matching is correct. If there is no feature matching, we can't determine where the p_2 is on the epipolar line. At that time, we must search on the epipolar line l_2 to obtain the correct match.

Now, let's look at the geometric relationship algebraically. Define the spatial position of P in the first frame to be:

$$\mathbf{P} = [X, Y, Z]^T \tag{1.1}$$

Once camera intrinsics are known, we can obtain the pixel positions of the two pixels $\mathbf{p}_1$, $\mathbf{p}_2$ as bellow:

$$s_1\mathbf{p}_1 = \mathbf{KP}, \quad s_2\mathbf{p}_2 = \mathbf{K}(\mathbf{RP} + \mathbf{t}) \tag{1.2}$$

where $\mathbf{K}$ is the camera intrinsic matrix, and $\mathbf{R}$, $\mathbf{t}$ are the camera motions between two frames. Specifically, they are $\mathbf{R}_{21}$ and $\mathbf{t}_{21}$, i.e., transformation from the first frame to the second. We can also write them in Lie-algebra form.

Sometimes, we use homogeneous coordinates to represent pixels. When using homogeneous coordinates, a vector will be equal to itself multiplied by any non-zero constant. This is usually used to express a projection relationship. e.g., $s_1\mathbf{p}_1$ and $\mathbf{p}_1$ form a projection relationship, and they are equal in the sense of homogeneous coordinates. We call this equal up to a scale, denoted as:

$$s\mathbf{p} \simeq \mathbf{p} \tag{1.3}$$

Then, the relationship between two projections can be written as:

$$\mathbf{p}_1 \simeq \mathbf{K}\mathbf{P}, \quad \mathbf{p}_2 \simeq \mathbf{K}(\mathbf{R}\mathbf{P} + \mathbf{t}) \tag{1.4}$$

Now, define:

$$\mathbf{x}_1 = \mathbf{K}^{-1}\mathbf{p}_1, \quad \mathbf{x}_2 = \mathbf{K}^{-1}\mathbf{p}_2 \tag{1.5}$$

Here, $\mathbf{x}_1$, $\mathbf{x}_2$ are the coordinates on the normalized plane of two pixels. Substituting to the above equation, we get:

$$\mathbf{x}_2 \simeq \mathbf{R}\mathbf{x}_1 + \mathbf{t} \tag{1.6}$$

Left multiply both sides by $\mathbf{t}^\wedge$. Recalling the definition of $\wedge$, this is equivalent to the outer product of both sides with $\mathbf{t}$:

$$\mathbf{t}^\wedge\mathbf{x}_2 \simeq \mathbf{t}^\wedge\mathbf{R}\mathbf{x}_1 \tag{1.7}$$

Then, left multiply $\mathbf{x}_2^T$ on both sides:

$$\mathbf{x}_2^T\mathbf{t}^\wedge\mathbf{x}_2 \simeq \mathbf{x}_2^T\mathbf{t}^\wedge\mathbf{R}\mathbf{x}_1 \tag{1.8}$$

On the left side, $\mathbf{t}^\wedge\mathbf{x}_2$ is a vector orthogonal to both $\mathbf{t}$ and $\mathbf{x}_2$. So its inner product with $\mathbf{x}_2$ will get 0. Since the left side of the equation is strictly zero, it is also zero after multiplying by any non-zero constant, so we can turn $\simeq$ back to the usual equal sign. Therefore, we have a concise equation:

$$\mathbf{x}_2^T\mathbf{t}^\wedge\mathbf{R}\mathbf{x}_1 = 0 \tag{1.9}$$

Substituting the $\mathbf{p}_1$, $\mathbf{p}_2$ again, we have:

$$\mathbf{p}_2^T \mathbf{K}^{-T} \mathbf{t}^\wedge \mathbf{R} \mathbf{K}^{-1} \mathbf{p}_1 = 0 \tag{1.10}$$

Both equations are called epipolar constraint, which is famous for its conciseness. Geometrically, it means O_1, P, O_2 are coplanar. The epipolar constraint encodes both translation and rotation. We define two matrices: Fundamental matrix $\mathbf{F}$ and essential Matrix $\mathbf{E}$ from this equation:

$$\mathbf{E} = \mathbf{t}^\wedge \mathbf{R}, \quad \mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}, \quad \mathbf{x}_2^T \mathbf{E} \mathbf{x}_1 = \mathbf{p}_2^T \mathbf{F} \mathbf{p}_1 = 0 \tag{1.11}$$

The epipolar constraint gives the spatial relationship of two matching points concisely. Therefore, the camera pose estimation problem can be summarized as the following two steps:

1. Find $\mathbf{E}$ or $\mathbf{F}$ based on the pixel positions of the matched points.
2. Find $\mathbf{R}$, $\mathbf{t}$ based on $\mathbf{E}$ or $\mathbf{F}$.

Since $\mathbf{E}$ and $\mathbf{F}$ only differ from the camera internal parameters, and the internal parameters are assumed to be known in SLAM problem, so the simpler form E is often used in practice. Let's take $\mathbf{E}$ as an example to introduce how to solve the above two problems.

Essential Matrix By definition, the essential matrix $\mathbf{E} = \mathbf{t}^\wedge \mathbf{R}$. It is a matrix of 3×3 with 9 unknown variables. From the structure of $\mathbf{E}$, there are the following points worth noting:

- The essential matrix is defined by the epipolar constraint. Since the epipolar constraint is the constraint of an *equal-to-zero* equation, after multiplying $\mathbf{E}$ by any non-zero constant, the constraint is still satisfied. We call this $\mathbf{E}$'s equivalence under different scales.
- According to $\mathbf{E} = \mathbf{t}^\wedge \mathbf{R}$, it can be proved that the singular value of the essential matrix $\mathbf{E}$ must be in the form of $[\sigma, \sigma, 0]^T$. This is called internal properties of essential matrix [74].

- On the other hand, since translation and rotation each have 3 degrees of freedom, $\mathbf{t}^\wedge\mathbf{R}$ has 6 degrees of freedom. But due to the equivalence of scales, $\mathbf{E}$ actually has 5 degrees of freedom.

The fact that $\mathbf{E}$ has 5 degrees of freedom indicates that we can use at least 5 pairs of points to solve $\mathbf{E}$. However, the internal property of $\mathbf{E}$ is nonlinear, which could cause trouble in the estimation. Therefore, it is also possible to consider only its scale equivalence and use 8 pairs of matched points to estimate $\mathbf{E}$. This is the classic *eight-point-algorithm* [75, 76]. The eight-point method only uses the linear properties of $\mathbf{E}$, so it can be solved under the framework of linear algebra. Let's take a look at how the eight-point method works.

Consider a pair of matched points, their normalized coordinates are $\mathbf{x}_1 = [u_1, v_1, 1]^T$, $\mathbf{x}_2 = [u_2, v_2, 1]^T$. According to the polar constraints, we have:

$$(u_2, v_2, 1) \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0 \tag{1.12}$$

We rewrite the matrix $\mathbf{E}$ in the vector form:

$$\mathbf{e} = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T \tag{1.13}$$

Then the epipolar constraint can be written in a linear form related to $\mathbf{e}$:

$$[u_2u_1, u_2v_1, u_2, v_2v_1, v_2u_1, v_1, 1] \cdot \mathbf{e} = 0 \tag{1.14}$$

Analogously, we stack all the points into one equation and obtain a linear equation system

(where u^i, v^i represent the i-th feature point):

$$\begin{pmatrix} u_2^1u_1^1 & u_2^1v_1^1 & u_2^1 & v_2^1u_1^1 & v_2^1v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2u_1^2 & u_2^2v_1^2 & u_2^2 & v_2^2u_1^2 & v_2^2v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^8u_1^8 & u_2^8v_1^8 & u_2^8 & v_2^8u_1^8 & v_2^8v_1^8 & v_2^8 & u_1^8 & v_1^8 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{pmatrix} = 0 \tag{1.15}$$

These eight equations form a linear equation system. Its coefficient matrix is composed of 2D feature point positions, and its size is 8×9. $\mathbf{e}$ is located in the null space of this matrix. If the coefficient matrix is of full rank (i.e., 8), then its null space dimension is 1, meaning that e forms a line. This is consistent with the scale equivalence of e. If the matrix composed of 8 pairs of matching points meets the condition of rank 8, then the elements of $\mathbf{E}$ can be solved uniquely by the above equation.

The next question is how to recover the movement of the camera $\mathbf{R}$, $\mathbf{t}$ according to the estimated essential matrix $\mathbf{E}$. This process is obtained by singular value decomposition (SVD). Let the SVD decomposition of $\mathbf{E}$ be:

$$\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{1.16}$$

where $\mathbf{U}$, $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}$ is the singular value matrice. According to the internal properties of $\mathbf{E}$, we know that $\mathbf{\Sigma} = diag(\sigma, \sigma, 0)$. In SVD decomposition, for any

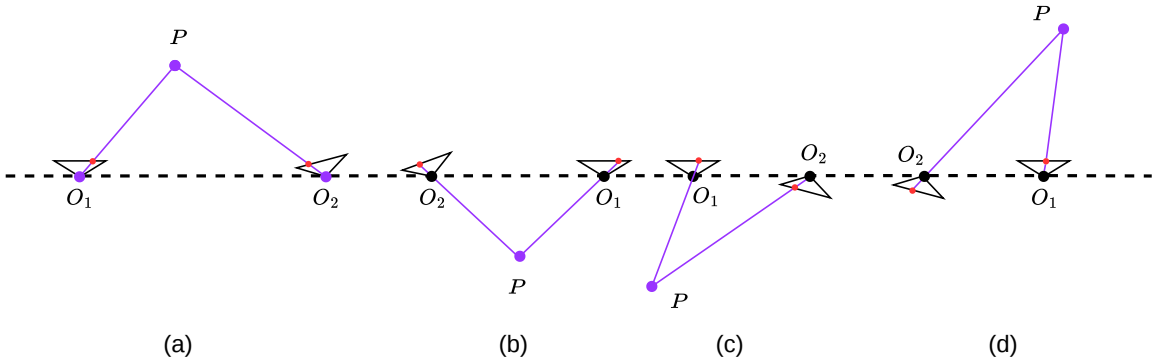


Figure 1.5: We get four solutions when decomposing the essential matrix.

$\mathbf{E}$, there are two possible $\mathbf{t}$, $\mathbf{R}$:

$$\begin{aligned} \mathbf{t}_1^\wedge &= \mathbf{U}\mathbf{R}_Z(\frac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \quad \mathbf{R}_1 = \mathbf{U}\mathbf{R}_Z^T(\frac{\pi}{2})\mathbf{V}^T \\ \mathbf{t}_2^\wedge &= \mathbf{U}\mathbf{R}_Z(-\frac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \quad \mathbf{R}_2 = \mathbf{U}\mathbf{R}_Z^T(-\frac{\pi}{2})\mathbf{V}^T \end{aligned} \tag{1.17}$$

Among them, $\mathbf{R}_Z(\frac{\pi}{2})$ represents the rotation matrix obtained by rotating $90°$ along the Z axis. Since $\mathbf{E}$ is equivalent to $\mathbf{E}$, taking the minus sign for any $\mathbf{t}$ will also get the same result. Therefore, when decomposing from $\mathbf{E}$ to $\mathbf{t}$, $\mathbf{R}$, there are a total of four possible solutions.

Figure 1.5 shows the four solutions obtained by decomposing the essential matrix. We know the projection (red) of the space point on the camera (blue line) and want to solve the camera's motion. In the case of keeping the red point unchanged, four possible situations can be drawn. Fortunately, only in the first solution, P has positive depths in both cameras. Therefore, we can substitute any points into the four solutions and check the depth's sign to determine which solution is correct.

If we use the internal properties of $\mathbf{E}$, then it has only five degrees of freedom. So at least five pairs of matched points can be used to solve the camera motion [77,78]. However, this approach is more complicated. Since there are usually dozens or even hundreds of matching points for engineering realization, it is often not helpful to reduce from 8 pairs to 5 pairs. To keep it simple, we only introduce the basic eight-point method here. There is one remaining problem: $\mathbf{E}$ solved by linear equations may not satisfy the internal properties, i.e., its singular value is not necessarily in the form of σ, σ, 0. At this time, we will

deliberately adjust the $\mathbf{\Sigma}$ matrix to look like the above. The usual procedure is to perform SVD decomposition on the $\mathbf{E}$ obtained by the eight-point method. Assume the singular value matrix is $\mathbf{\Sigma} = diag(\sigma, \sigma, 0)$, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$. We may take:

$$\mathbf{E} = \mathbf{U} diag\left(\frac{\sigma_1 + \sigma_2}{2}, \frac{\sigma_1 + \sigma_2}{2}, 0\right)\mathbf{V}^T \tag{1.18}$$

This is equivalent to projecting the calculated essential matrix onto the manifold where $\mathbf{E}$ is located. A simpler approach is to take the singular value matrix as $diag(1, 1, 0)$, due to $\mathbf{E}$'s scale equivalence, so it is also reasonable.

1.2.2 3D – 2D: Perspective-n-Point

PnP (Perspective-n-Point) is a method to solve 3D to 2D motion estimation. It describes how to estimate the camera's pose when the n 3D space points and their projection positions are known. As mentioned earlier, the 2D-2D epipolar geometry method requires eight or more point pairs (take the eight-point method as an example), and there have problems with initialization, pure rotation, and scale. However, if the 3D position of one of the feature points in the two images is known, then we need at least three pairs (and at least one additional point to verify the result) to estimate the camera motion. The 3D position of the feature point can be determined by triangulation or the depth map of an RGB-D camera. Therefore, in binocular or RGB-D visual odometry, we can directly use PnP to estimate camera motion. While in the monocular case, initialization must be conducted before using PnP. The 3D-2D method does not require epipolar constraints and can obtain better motion estimation in a few matching points. It is the most important pose estimation method.

There are many ways to solve PnP problems, for example, P3P [79], direct linear transformation (DLT), EPnP (Efficient PnP) [80], UPnP [81], etc. In addition, non-linear optimization can be used to construct a least-square problem and iteratively solve it, which is commonly called bundle adjustment.

Direct Linear Transformation Consider such a problem: we know the 3D positions of a point set and their projections in the camera, now we want to find the camera's pose. This problem can be used to solve the camera pose when a given map and image are given. If the

3D point is regarded as a point in another camera coordinate system, it can also solve the two cameras' relative motion problem. We will start with simple questions. Consider a 3D spatial point P, its homogeneous coordinates are $\mathbf{P} = (X, Y, Z, 1)^T$. In the image I_1, it is projected to the feature point $\mathbf{x}_1 = (u_1, v_1, 1)^T$ (expressed as the normalized homogeneous coordinates). At this time, the pose of the camera $\mathbf{R}$, $\mathbf{t}$ is unknown. We define the 3×4 augmented matrix $[\mathbf{R}|\mathbf{t}]$, encoding rotation and translation information. We will write its expanded form as follows:

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = [\mathbf{R}|\mathbf{t}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad [\mathbf{R}|\mathbf{t}] = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \tag{1.19}$$

Eliminate s with the last row to get two constraints:

$$u_1 = \frac{t_1 X + t_2 Y + t_3 Z + t_4}{t_9 X + t_{10} Y + t_{11} Z + t_{12}}, \quad v_1 = \frac{t_5 X + t_6 Y + t_7 Z + t_8}{t_9 X + t_{10} Y + t_{11} Z + t_{12}} \tag{1.20}$$

To simplify the representation, define $\mathbf{T}$ as a row vector:

$$\mathbf{t}_1 = (t_1, t_2, t_3, t_4)^T, \quad \mathbf{t}_2 = (t_5, t_6, t_7, t_8)^T, \quad \mathbf{t}_3 = (t_9, t_{10}, t_{11}, t_{12})^T \tag{1.21}$$

Now we have:

$$\mathbf{t}_1^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} u_1 = 0, \quad \mathbf{t}_2^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} v_1 = 0 \tag{1.22}$$

Please note that $\mathbf{t}$ is the variable to be determined. As you can see, each feature point provides two linear constraints on $\mathbf{t}$. Assuming there are a total of N feature points, the

following linear equations can be constructed:

$$\begin{pmatrix} \mathbf{P}_1^T & 0 & -u_1\mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N\mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N\mathbf{P}_N^T \end{pmatrix} \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix} = 0 \tag{1.23}$$

Since $\mathbf{t}$ has a total dimension of 12, the linear solution of the matrix $\mathbf{T}$ can be achieved by at least six pairs of matching points. This method is called Direct Linear Transform (DLT). When the matching points are greater than 6 pairs, methods such as SVD can also be used to find the least-square solution of the overdetermined equation.

In the DLT solution, we directly regard the T matrix as 12 unknowns, ignoring the correlation between them. Because the rotation matrix $\mathbf{R} \in SO(3)$, the solution obtained by DLT does not necessarily satisfy the $SE(3)$ constraint. It is just a general matrix. The translation vector is easier to handle. It belongs to the vector space. For the rotation matrix $\mathbf{R}$, we must look for the best rotation matrix to approximate the matrix block of 3×3 on the left of $\mathbf{T}$ estimated by DLT. This can be done by QR decomposition [82, 83], or it can be calculated like [84, 85]:

$$\mathbf{R} \leftarrow (\mathbf{R}\mathbf{R}^T)^{-\frac{1}{2}}\mathbf{R} \tag{1.24}$$

This can be seen as reprojecting the result from the matrix space onto the $SE(3)$ manifold and converting it into two parts: rotation and translation.

What needs to be mentioned is that our $\mathbf{x}_1$ here uses normalized plane coordinates and neglects the influence of the intrinsic matrix $\mathbf{K}$. This is because $\mathbf{K}$ is usually assumed to be known. Even if the intrinsic parameters are unknown, PnP can still be used to estimate the three quantities $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$. However, due to the increase in the number of unknown variables, the result's quantity may be worse.

P3P P3P is another way to solve PnP. It only uses 3 pairs of matching points and requires less data. P3P requires establishing geometric relationships of the given 3 points. Its input

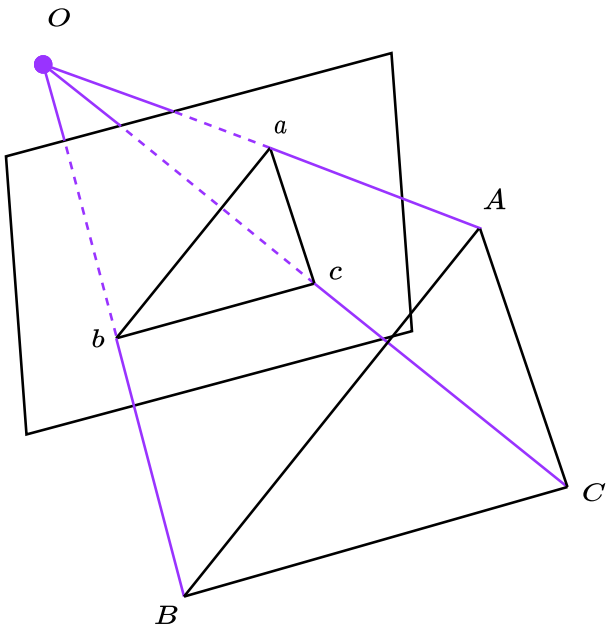


Figure 1.6: The P3P problem.

data is 3 pairs of 3D-2D matching points. Define 3D points as A, B, C, 2D points as a, b, c, where the point represented by the lowercase letter is the projection of the point on the camera image plane represented by the corresponding uppercase letter, as shown in Figure 1.6. Also, P3P needs a pair of verification points to select the correct one from the possible solutions (similar to epipolar geometry). Denote the verification point pair as D - d and the principal camera point as O. Suppose that A, B, C are in the world coordinate frame, not the camera coordinate. Once the coordinates of the 3D point in the camera coordinate system can be calculated, we get the 3D3D corresponding point and convert the PnP problem to the ICP problem. Obviously, there is a relationship between triangles:

$$\Delta Oab - \Delta OAB, \quad \Delta Obc - \Delta OBC, \quad \Delta Oac - \Delta OAC \tag{1.25}$$

Consider the relationship between Oab and OAB. Using the law of cosines, there are:

$$\begin{aligned} \overline{OA}^2 + \overline{OB}^2 - 2.\overline{OA}.\overline{OB}.cos\left\langle a, b\right\rangle &= \overline{AB}^2 \\ \overline{OB}^2 + \overline{OC}^2 - 2.\overline{OB}.\overline{OC}.cos\left\langle b, c\right\rangle &= \overline{BC}^2 \\ \overline{OA}^2 + \overline{OC}^2 - 2.\overline{OA}.\overline{OC}.cos\left\langle a, c\right\rangle &= \overline{AC}^2 \end{aligned} \tag{1.26}$$

Divide all the above three equations by $\overline{OC}^2$ on both sides, and consider $x = \overline{OA}/\overline{OC}$, $y = \overline{OB}/\overline{OC}$, we get:

$$\begin{aligned} x^2 + y^2 - 2xy \cdot cos\left\langle a, b\right\rangle &= \overline{AB}^2/\overline{OC}^2 \\ y^2 + 1^2 - 2y \cdot cos\left\langle b, c\right\rangle &= \overline{BC}^2/\overline{OC}^2 \\ x^2 + 1^2 - 2x \cdot cos\left\langle a, c\right\rangle &= \overline{AC}^2/\overline{OC}^2 \end{aligned} \tag{1.27}$$

Let $v = \overline{AB}^2/\overline{OC}^2$, $uv = \overline{BC}^2/\overline{OC}^2$, $wv = \overline{AC}^2/\overline{OC}^2$, then we have:

$$\begin{aligned} x^2 + y^2 - 2xy \cdot cos\left\langle a, b\right\rangle - v &= 0 \\ y^2 + 1^2 - 2y \cdot cos\left\langle b, c\right\rangle - uv &= 0 \\ x^2 + 1^2 - 2x \cdot cos\left\langle a, c\right\rangle - wv &= 0 \end{aligned} \tag{1.28}$$

Please distinguish the known from the unknown quantities in these equations. Since we know the positions of the 2D points in the image, the 3 cosine angles $cos\left\langle a, b\right\rangle$, $cos\left\langle b, c\right\rangle$, $cos\left\langle a, c\right\rangle$ can be calculated. Meanwhile, $u = \overline{BC}^2/\overline{AB}^2$, $w = \overline{AC}^2/\overline{AB}^2$ can be calculated by the coordinates of A, B, C in the world frame. Transforming to the camera frame does not change the ratio. The x and y are unknown and will change as the camera moves. Therefore, the equations are quadratic equations about two unknowns x, y. Analytically solving the equations is complicated and requires Wu's elimination method. It will not be introduced here. Interested readers could refer to the literature [79]. Analogous to the case of decomposing $\mathbf{E}$, this equation may get 4 solutions at most. Still, we can use the verification points to select the most probable solution and get the 3D of A, B, C in the camera frame. Then, based on the 3D - 3D point pair, the camera movement $\mathbf{R}$, $\mathbf{t}$ can be calculated.

Solve PnP by Minimizing the Reprojection Error Other than the linear method, we can also construct the PnP problem as a nonlinear least-square problem about reprojection errors. The linear method mentioned above is often divided into many steps, such as estimating the camera pose first and then the point's position. While nonlinear optimization treats them as optimization variables and optimizes them together. This is a very general solution method. We can use it to optimize the results given by PnP or ICP. This type of problem, putting the camera and 3D points together to minimize, is generally referred to as

Bundle Adjustment (BA).

We can build a bundle adjustment problem in PnP to optimize the camera pose. Suppose we have n 3D space points P and their projection p, we want to calculate the pose $\mathbf{R}$, $\mathbf{t}$ of the camera. Suppose the coordinates of a point are $\mathbf{P}_i = [X_i, Y_i, Z_i]^T$, and their projected pixel coordinates are $u_i = [u_i, v_i]^T$. According to 1.2.1, the relationship between the 2D pixel position and the 3D spatial position is:

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{KT} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{1.29}$$

This equation includes a conversion from homogeneous coordinates to non-homogeneous coordinates implicitly. Or we can also use $\mathbf{RP} + \mathbf{t}$. Now, due to the unknown camera pose and the noise of the observation points, there is a residual in the equation. Therefore, we sum up the residuals, construct a least-square problem, and then minimize it to find the most possible camera pose:

$$\mathbf{T}^* = argmin \frac{1}{2} \sum_{i=1}^{n} \left\| \mathbf{u}_i - \frac{1}{s_i} \mathbf{KTP}_i \right\|_2^2 \tag{1.30}$$

The residual term is the error of the projected position and the observed position, which is called the reprojection error. With homogeneous coordinates, this error has three dimensions. However, since the last dimension of $\mathbf{u}$ is 1, the error of this dimension is always zero, so we normally use non-homogeneous coordinates. Therefore, the error has only 2 dimensions. As shown in Figure 1.7, we know that p_1 and p_2 are projections of the same space point P through feature matching, but we don't know the pose of the camera. In the initial value, there is a certain distance between the projection of $P\hat{p}_2$ and the observed p_2. So we adjusted the pose of the camera to make this distance smaller. However, since this adjustment needs to consider many points, the goal is to reduce the overall error, and the error of each point usually can not be exactly zero.

Using Lie algebra, we can construct an unconstrained optimization problem on the

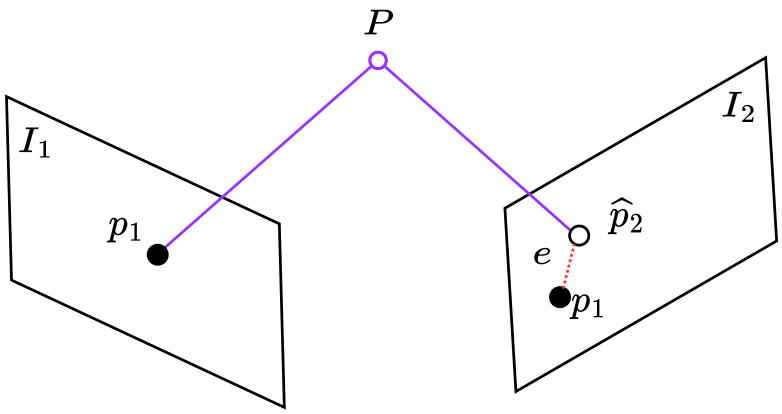


Figure 1.7: The reprojection error.

manifold, easily solved using optimization algorithms such as the Gauss-Newton method and Levenberg-Marquardt method. However, we need to calculate the derivative of each error term with respect to the optimization variable, which is also the linearization:

$$\mathbf{e}(\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}) \approx \mathbf{e}(\mathbf{x}) + \mathbf{J}^T \boldsymbol{\Delta}\mathbf{x} \tag{1.31}$$

The form of $\mathbf{J}^T$ is worth discussing. Definitely, we can use numerical derivatives, but if we can derive an analytical form, we will prefer the analytical derivatives. Now, $\mathbf{e}$ is the pixel coordinate error (2d) and $\mathbf{x}$ is the camera pose (6d), $\mathbf{J}^T$ is a matrix of 2×6. Let's derive the form of $\mathbf{J}^T$.

First, define the coordinates of the space point in the camera frame as $\mathbf{P}'$, and take out the first 3 dimensions:

$$\mathbf{P}' = (\mathbf{TP})_{1:3} = [X', Y', Z']^T \tag{1.32}$$

Then, the camera projection model with respect to $\mathbf{P}'$ is:

$$s\mathbf{u} = \mathbf{KP}' \Rightarrow \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{1.33}$$

Use the third row to eliminate s (actually it is the distance of $\mathbf{P}'$), we get:

$$u = f_x \frac{X'}{Z'} + c_x, \quad v = f_y \frac{Y'}{Z'} + c_y \tag{1.34}$$

When we find the error, we can compare the u, v here with the measured value to find the difference. After defining the intermediate variables, we left multiply $\mathbf{T}$ by a disturbance quantity $\delta\boldsymbol{\xi}$, and then consider the derivative of the change of $\mathbf{e}$ with respect to the disturbance quantity. Using the chain rule, it is:

$$\frac{\partial \boldsymbol{e}}{\partial \delta \boldsymbol{\xi}} = \lim_{\delta \boldsymbol{\xi} \to 0} \frac{\boldsymbol{e}(\delta \boldsymbol{\xi} \oplus \boldsymbol{\xi}) - \boldsymbol{e}(\boldsymbol{\xi})}{\delta \boldsymbol{\xi}} = \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P'}} \frac{\partial \boldsymbol{P'}}{\partial \delta \boldsymbol{\xi}} \tag{1.35}$$

Here $\oplus$ refers to the disturbance left multiplication in Lie algebra. The first item is the derivative of the error with respect to the projection point. The relationship between the variables in E.q. 1.34, and it is easy to get:

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P'}} = - \begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \tag{1.36}$$

The second term is the derivative of the transformed point with respect to the Lie algebra. we get:

$$\frac{\partial (\boldsymbol{TP})}{\partial \delta \boldsymbol{\xi}} = (\mathbf{TP})^{\odot} = \begin{bmatrix} \mathbf{I} & -\mathbf{P}'^{\wedge} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \Rightarrow \frac{\partial (\boldsymbol{P'})}{\partial \delta \boldsymbol{\xi}} = [\mathbf{I}, -\mathbf{P}'^{\wedge}] \tag{1.37}$$

Multiply these two items together, we get the 2×6 Jacobian matrix:

$$\frac{\partial (\boldsymbol{e})}{\partial \delta \boldsymbol{\xi}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \tag{1.38}$$

This Jacobian matrix describes the first-order derivative of the reprojection error with respect to the left perturbation model. We keep the negative sign in front of it because the error is defined by the observed value minus the predicted value. It can also be reversed and defined in the form of the predicted value minus the observed value. In that case, just

remove the negative sign in front. Besides, if the definition of $se(3)$ is a rotation followed by translation, just swap the first 3 columns and the last 3 columns of this Jacobian matrix.

On top of optimizing the pose, we want to optimize the spatial position of the feature points. Therefore, we also need to discuss the derivative of $\mathbf{e}$ with respect to the space point $\mathbf{P}$. Fortunately, this derivative matrix is relatively easy. Still using the chain rule, there are:

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P}} = \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P'}} \frac{\partial \boldsymbol{P'}}{\partial \boldsymbol{P}} \tag{1.39}$$

The first item has been deduced before, and the second item is defined as:

$$\mathbf{P}' = (\mathbf{TP})_{1:3} = \mathbf{RP} + \mathbf{t} \tag{1.40}$$

So only $\mathbf{R}$ is left in the derivative:

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{P}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \mathbf{R} \tag{1.41}$$

We have derived the two Jacobian matrices of the observation camera equation with respect to the camera pose and feature points. They are very important to provide gradient directions in the optimization and guide the iteration of optimization.

1.2.3 Deep Neural Network

A Deep Neural Network (DNN) is a type of artificial neural network that is composed of many layers of interconnected nodes. These layers allow the network to learn and represent complex patterns in data, and they make it possible for the network to perform highly accurate predictions on new data. Deep neural networks are widely used in many applications, such as image recognition, natural language processing, and even self-driving cars. They are also at the heart of many recent advances in artificial intelligence and machine learning. The process of training a deep neural network involves adjusting the weights of the connections between the nodes in order to improve the network's ability to make accurate predictions. This is done using a variety of algorithms and techniques, such as backpropagation and stochastic gradient descent. One of the key advantages of deep neural

networks is that they are highly flexible and can be applied to a wide range of problems. They can learn to make predictions on data that is structured or unstructured, and they can be fine-tuned to perform well on specific tasks. Additionally, deep neural networks are able to learn from large amounts of data, which allows them to make highly accurate predictions.

The development of deep neural networks can be traced back to the 1940s and 1950s, the Artificial Neural Network (ANN) was introduced firstly by Frank Rosenblatt in 1958 [86]. Early DNNs had so many problems that they needed to be more challenging to apply in practice, such as the difficulty of training, poor generalizability, a limited amount of data, etc. David *et al.* proposed Multi-Layer Perception (MLP) and training strategy with Backward Propagation (BP) [87]. It was significant progress in artificial neural networks that allowed the ANN to learn effectively from its error. It becomes a core idea in any neural network up to now. Moreover, MLPs learn to recognize the patterns from the data using one perceptron for each input, which is quite expensive and more easily leads to overfitting problems. It is especially different from applying to process data with a grid-like structure, such as an image. Yann [88] proposed the Convolutional Neural Network (CNN) for better utilizing neural networks to process data of images which leads to a significant development in computer vision fields. CNNs are composed of multiple layers of interconnected nodes, and they are able to learn and represent complex patterns in data by adjusting the weights of the connections between these nodes; see Sec. 1.2.3.

With the booming development of IT technology, deep neural networks have become a powerful tool for solving many complex problems in artificial intelligence and machine learning. They have contributed to many recent breakthroughs in these fields, and they are likely to continue to play a key role in the future development of AI and machine learning.

This section introduces the background knowledge about DNNs involved in this dissertation, including components of the Neuron in DNN, the convolutional neural network, and the transformer.

Components of the Neuron in DNN

The neuron is an individual and fundamental unit of a neural network. The components of a neuron are similar to those of a biological neuron. A DNN neuron consists of a set of inputs, weights, a bias term, normalization, and an activation function. Mathematically,

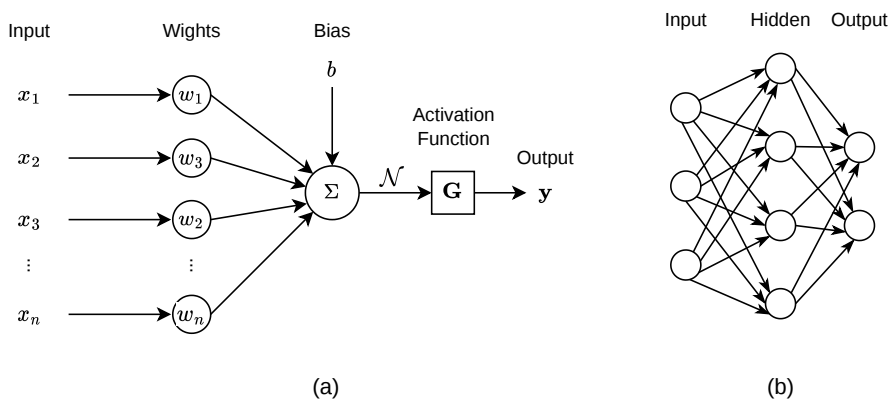


Figure 1.8: Example of (a) Neuron and (b) Neural network.

given an input $\mathbf{x} : x_i \in \mathbf{x}$ where $i \in [1, n]$, the output $\mathbf{y}$ of a neuron is denoted as:

$$\mathbf{y} = \mathbf{G}(\mathcal{N}(\sum_{i=1}^{n} w_i \cdot x_i + b)) \tag{1.42}$$

where w_i denotes weights, b denotes bias, $\mathcal{N}$ denotes normalization function, and $\mathbf{G}$ denotes activation function. Figure 1.8 shows an example of a neuron and a neural network.

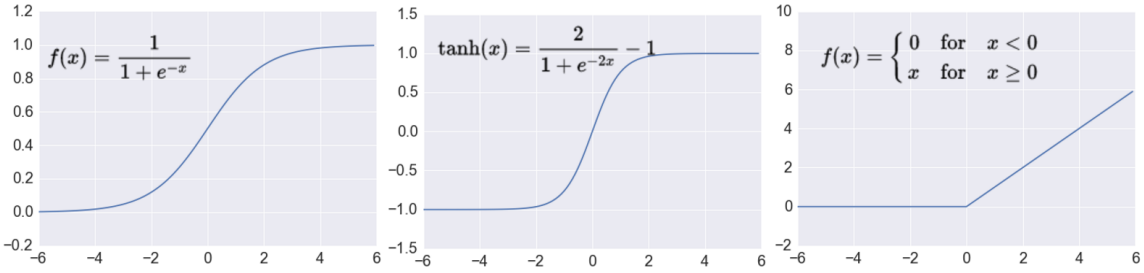
Normalization in DNN. Appling normalization can improve the performance and stability of training deep learning models. It normalizes the output of each neuron in a layer so that they have a mean of zero and a standard deviation of one. This helps to prevent the inputs to the next layer from becoming too large or too small, which can cause the model to become unstable or perform poorly.

Batch normalization is widely used in training CNN models. It performs normalization in the batch size dimension, which can be denoted as:

$$\hat{x} = \frac{x - \mu}{\sigma}, \quad y = \gamma\hat{x} + \beta \tag{1.43}$$

where x is input, μ and σ are mean and standard deviation (std) of x in mini-batch. γ and β are scale and shift parameters that are learnable.

Layer normalization is similar to batch normalization, but instead of normalizing the outputs of each neuron based on the statistics of the current batch of data. It is applicable in the situation of input data with unequal lengths, such as texts, which be widely used in



(a) (b) (c)

Figure 1.9: Activation Function; (a) Sigmoid (b) TanH and (c) ReLU.

transformers.

Activation Function. A activation function is used to introduce non-linearity into the network, allowing the model to learn and make more complex decisions. The most commonly used activation function is the rectified linear unit (ReLU), which returns the input if it is positive, and returns zero if it is negative. Other common activation functions include the sigmoid function, which outputs a value between 0 and 1, and the hyperbolic tangent function, which outputs a value between -1 and 1. The choice of activation function can have a significant impact on the performance of the network, so it is important to select the right one for the task at hand.

Convolutional Neural Network

The default input to a convolutional neural network is an image, which allows us to encode specific properties into the network structure, making our feedforward function more efficient and reducing a large number of parameters. It is a deep neural network with a convolutional structure, which can reduce the amount of memory occupied by the deep network. Its three key operations, one of which is the local perceptual field, the other is weight sharing, and the third is the pooling layer, effectively reduce the number of parameters of the network and alleviate the overfitting problem of the model. A standard convolutional neural network is composed of three main parts, i.e., convolution layer, pooling layer, and fully connected layer.

Convolution Layer. The convolution layer is the first layer of CNN and is responsible for applying a set of filters to the input data to extract features. This is done by sliding the filters across the input data and computing the dot product between the values in the filter

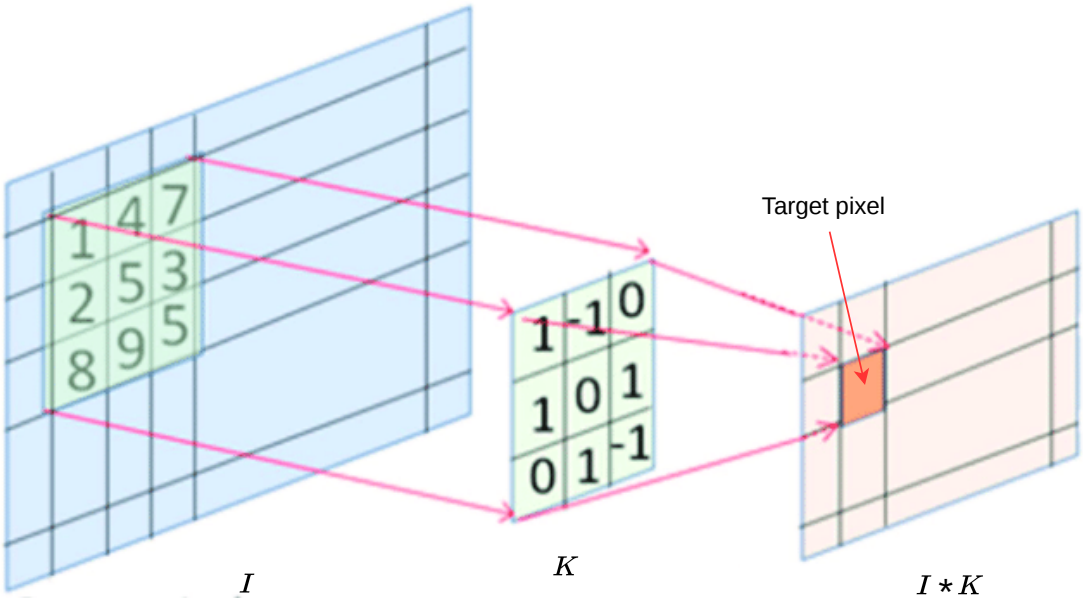


Figure 1.10: Illustration of the convolution layer. K denotes a learnable convolution kernel with size of 3×3.

and the input data at each position. This results in a set of feature maps that represent the output of the filters at each position in the input data. These feature maps are then fed as input to the next layer of the network. The convolution layer is an essential part of a CNN and is critical for its ability to extract meaningful features from images. Figure 1.10 shows an illustration of the convolution layer.

Pooling Layer. Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a

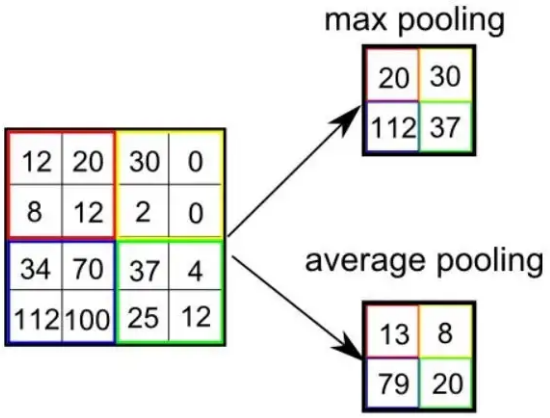


Figure 1.11: Types of pooling layer.

noise-suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling. Figure 1.11 shows the two types of pooling layers.

Fully Connected Layer. Fully connected (FC) layer is a linear approximation module that takes all elements in the input into account. This module obtains the input and applies the linear equation by its learnable parameters, i.e., weight and bias, to approximate the output. The input and output can be a vector of any dimension. Suppose the input $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ is the vector of m dimensions and the output $\mathbf{y}$ contains n dimensions, the calculation is

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} + \mathbf{b} = \begin{bmatrix} w_{1,1}x_1 + w_{2,1}x_2 + \cdots + w_{m,1}x_m + b_1 \\ w_{1,2}x_1 + w_{2,2}x_2 + \cdots + w_{m,2}x_m + b_2 \\ \vdots \\ w_{1,n}x_1 + w_{2,n}x_2 + \cdots + w_{m,n}x_m + b_n \end{bmatrix}$$

where $\mathbf{x}$ is the input, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight parameter, and $\mathbf{b} \in \mathbb{R}^n$ is the bias parameter. The special case of the FC layer when $n = 1$ is the linear regression model.

Stacking multiple FC layers increases the ability to approximate the value. However, naively stack the layers has an issue in mathematical aspect. It is necessary to apply the activation function between each layer. The detail will be described in the following section.

Transformer

The transformer was introduced in 2017 [89], which is used in the field of NLP at first. It is based on the idea of using self-attention mechanisms to process input data, allowing the network to learn dependencies between different parts of the input without the need for recursion or convolution. This makes it well-suited for natural language processing tasks, where the order and context of words in a sentence can be important for understanding the meaning of the sentence. Recently, the application of transformer is applied extended to computer vision fields and achieved state-of-art performances. The self-attention mechanisms allow the model to learn to focus on the most relevant features of the image. Vision transformers are mainly working based on transformer encoders which are composed of a Multi-Head Attention Module (MHA) and MLP.

Self-Attention. Self-attention operates on a set of input tokens, which are mapped to a set of query Q, K key, and V value vectors. The attention weights are then computed as the dot product of the query and key vectors, normalized by the square root of the key vector length. The final output is computed as the weighted sum of the value vectors, using the attention weights as the weights. This allows the model to focus on different parts of the input when computing the output.

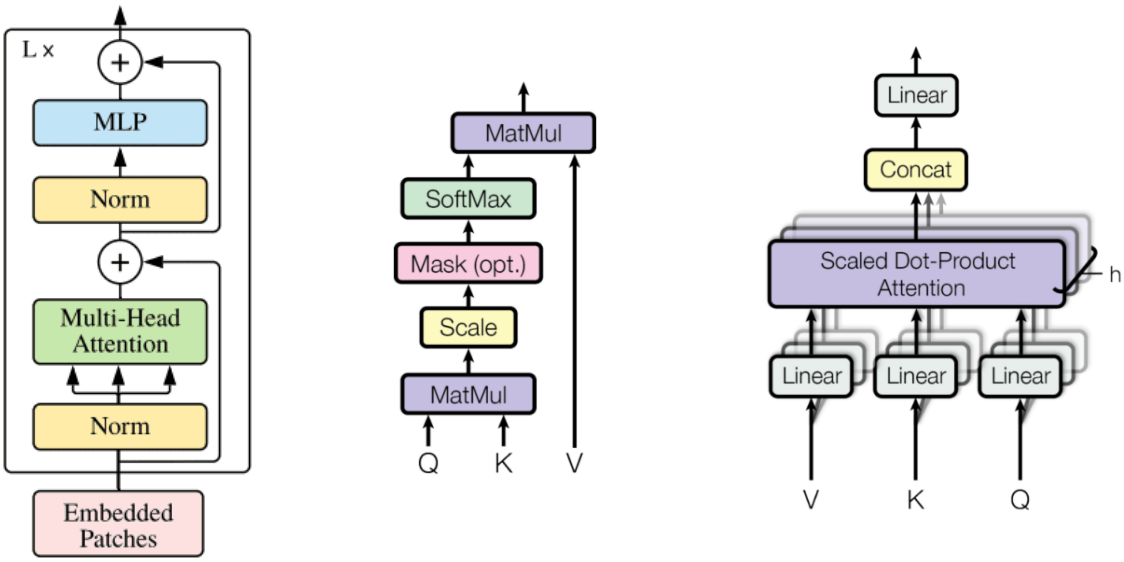
In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. Q, K, and V are learned from input data, respectively. We compute the matrix of outputs as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.44}$$

where d_k is a scaling factor.

Multi-Head Attention. Instead of performing a single attention function with Q, K, and V, it is beneficial to linearly project the Q, K, and V h times with different, learned linear projections, respectively. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

$$\begin{aligned} &MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \\ &head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{1.45}$$



(a) (b) (c)

Figure 1.12: Illustration of the transformer; (a) Structure of transformer encoder, (b) Self-Attention, and (c) Multi-Head Attention

where the projections are parameter matrices W_i^Q, W_i^K, W_i^V and W^O.

1.2.4 Deep Metric Learning

Deep Metric Learning is a method of learning a transformation of input data into a feature space so that samples are separated into classes based on their distance (metric) or similarity. The goal of distance learning is to increase the distance between negative samples while decreasing the distance between positive samples. With the development of deep learning, various deep distance learning methods have been proposed that combine deep learning and distance learning. Typical learning methods, such as contrastive loss and triplet loss are widely used for training deep-learning models of numerous computer vision tasks, such as classification, image-matching, and image retrieval.

Contrastive Loss. The contrastive loss was proposed for dimensionality reduction in 2006 [90] and the typical network structure used with it is the Siamese network [91]. When a positive pair is an input, the network is trained to minimize the distance, and when a negative pair is an input, the network is trained to maximize the distance. The loss function is as follows:

$$
\begin{aligned}
L_{Contrastive} &= (1-Y)\frac{1}{2}(D)^2 + (Y)\frac{1}{2}max(0, m-D)^2 \\
D(x_i^a, x_i^b) &= \|f(x_i^a) - f(x_i^b)\|_2
\end{aligned}
\tag{1.46}
$$

where x_i^a and x_i^b denote the i-th input image pair, Y denotes the label of the pair, labeled 0 for positive pairs and 1 for negative pairs. m denotes the margin which is a hyperparameter of a positive constant. According to the above equation, positive pairs are trained to $D = 0$ in the case of positive pairs, and $D > m$ in the case of negative pairs. However, it would be inappropriate to apply a constant margin to all negative pairs, since the distances within or between classes vary from class to class and can vary widely.

Triplet Loss. Triplet loss addresses the issue of margins in contrastive loss because a margin is applied to the relative distance. Negative pairs in the trio of input images and apply the margin to the relative distance between the positive and negative pairs. In other words, while the contrastive loss trains the negative pair to be further away than the distance of the margin m, the triplet loss trains the negative sample to be further away than the distance of the positive sample by more than the distance of the margin m. Triplet loss is to learn the distance from the negative sample to be more than the distance from the positive sample. At the same time, the positive samples can be attracted to the negative samples and try to leave the negative samples. It is a widely used distance learning method in current studies.

In previous studies, several slightly different equations were used as loss functions for triplet loss, the most commonly cited of which seems to be Wang *et al.* [92]. Their loss function is expressed by the following equation:

$$
L_{Triplet} = max(0, D^2(x_i^a, x_i^p) - D^2(x_i^a, x_i^n) + m) \tag{1.47}
$$

where x_i^a, x_i^p, and x_i^n denote the standard for sample pairs of anchor, positive, and negative images. It was called Hinge loss, but in recent research, a similar expression is often referred to as triplet loss. FaceNet [93] proposed triplet loss as the loss function that is minimized based on the above equation. In the equation below, the minimum loss is set to

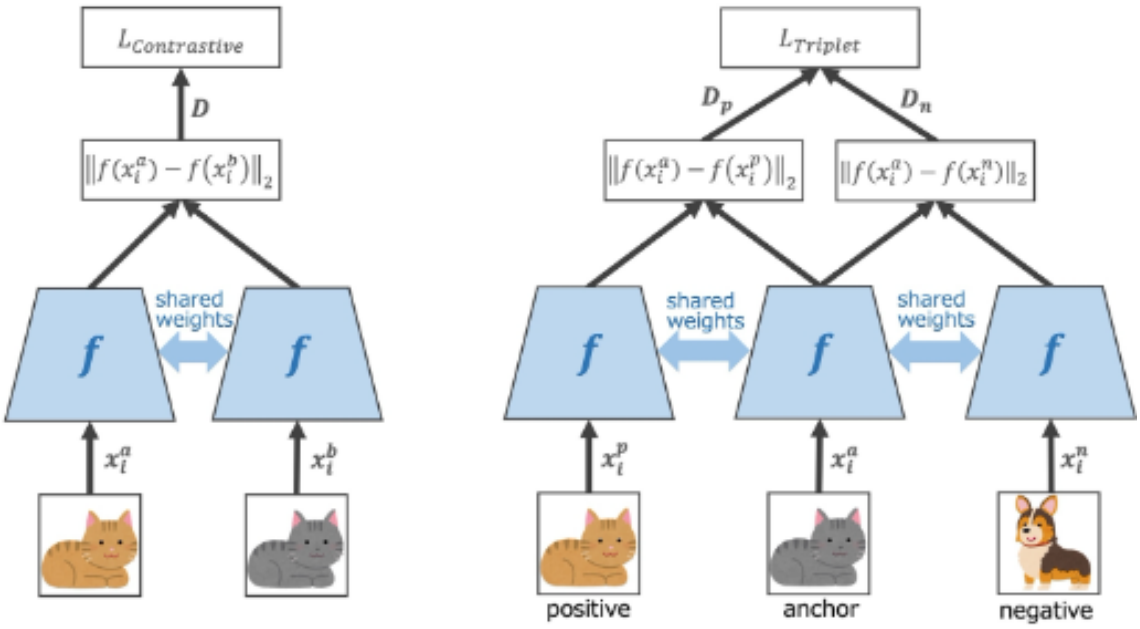


Figure 1.13: Illustration of neural networks trained with (a) Contrastive loss and (b) Triplet loss

zero.

$$L_{Triplet} = \sum_{i}^{N} [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + m]_+ \tag{1.48}$$

Note that the above two losses use normalized features to compare distances between samples. Figure 1.13 shows the illustration of neural networks trained with the above two losses, respectively.

1.3 Outline of the Dissertation

There are many challenges in the applications of visual SLAM or SfM. As the essential component in Visual Odometry (front-end) of visual SLAM or SfM, a robust image-matching approach is a key to overcoming these challenges. In this dissertation, we focus on the image-matching problem, aiming to widen the frontier of SfM and visual SLAM applications, as well as to improve downstream components such as visual localization. Specifically, we developed a robust image-matching pipeline for extremely low-light environments. To achieve this purpose, we presented two databases for evaluation and model

training. In addition, we proposed a method that unified related tasks, i.e., image retrieval and image-matching, for better visual localization. This dissertation also includes analyses in many aspects. The outline of the content is described in what follows.

In Chapter 2, we experimentally evaluate and verify the gain of utilizing high-precision information recorded in RAW images to match extremely low-light scene images. For extreme low-light scenes, even if some of their brightness information exists in the RAW format images' low bits, the standard raw image processing on cameras fails to utilize them properly. To promote further studies, we create a dataset for image-matching evaluation which include multi-level underexposure low-light RAW images, aiming at widening their application field toward lower-light scenes. Using this dataset, we experimentally evaluate several existing component methods for image-matching applied for camera pose estimation. They further provide the strengths and weaknesses of the above component methods, also showing that there is room for further improvement.

In Chapter 3, we are interested in better better-utilizing information in low-light RAW images for downstream tasks related to 3D reconstruction, since the standard image enhancers are designed to yield images that appear the most natural, which should differ from the best image for downstream tasks. However, the existing database does not meet the above requirements, i.e., existing datasets for 3D reconstruction don't contain dark RAW images of low-light scenes; on the other hand, existing datasets containing RAW images don't meet the requirements for training downstream task models, e.g., lack of relevant images, intrinsics/extrinsics and ground truth of depth information. To promote further studies, we collected a new dataset for 3D reconstruction which include low-light RAW images, MID-V2. In addition, we introduce a novel enhancer for RAW image processing, SuperISP, designed/trained to utilize the information stored in RAW images of low-light scenes to yield better performance for downstream tasks related to 3D reconstruction. Then, we experimentally evaluate the SuperISP on two typical downstream tasks related to 3D reconstruction. The SuperISP achieves better performance on the two downstream tasks than other enhancers that aim to obtain a high visual quality image.

In chapter 4, we focus on the flexible use of image-matching models in visual localization. Advanced visual localization approaches work with both global and local image features. i.e., the former is used to retrieve relevant frames, and the latter is used for local feature matching for 6 Degree-of-Freedom (DoF) camera pose estimation. However, exist-

ing studies are struggling to unify these two tasks due to conflicts at the feature level. For better visual localization, we propose a novel model which unifies local and global features for visual localization. In this model, we solve the feature-level conflict based on the attention mechanism of a transformer. The results show the advantage of our method compared to existing methods.

Finally, in Chapter 5, we summarize the content and provide the conclusion of this dissertation. Furthermore, we state the open problem and the possible future work.

Chapter 2

Matching in Low-light Scenes Leveraging RAW Images

2.1 Introduction

Structure-from-motion (SfM) [94,95] and visual SLAM (simultaneous localization and mapping) [96, 97] have been used for real-world applications for a while. The mainstream methods use point correspondences between multiple views of a scene. They first detect keypoints and extract the descriptor of the local feature at each keypoint [8, 24, 98, 99]. They then find initial point correspondences between images and eliminate outliers from them, finally estimating the geometric parameters such as camera poses, etc.

SfM and visual SLAM have the potential to widen their application fields. One important target is the application to extremely low-light environments, such as outdoor scenes at night under moonlight or indoor scenes with insufficient illumination. Making it possible to use SfM and visual SLAM in these environments is essential for real-world applications, such as autonomous vehicles that can operate at night.

Owing to the advancement of image sensors, they can record incoming light with more than eight bits (e.g., 14 bits). However, standard raw image processing employed on many cameras cannot make full use of the information existing in the lower bits of the sensor signal; it reduces mosaic artifacts on the sensor signal, adjusts the white balance and contrast, and then converts the processed signal into the standard format of eight-bit RGB images (we will refer to this raw image processing as RIP in this paper). This limitation arguably

comes from the requirement for versatility against all sorts of scenes with various lighting conditions in addition to reducing the number of bits. In extreme low-light scenes, even when some details of the scenes' brightness are stored in the low bits of their RAW signals, the standard RIP often yields mostly black images. The study of SID (see-in-the-dark) [5] well proves such limitation of the image pipeline, in which the authors show that a CNN can learn to convert such RAW-format images of dark scenes into brightened images with a natural appearance.

It is very likely that we can do the same with SfM and visual SLAM applied to low-light scenes, i.e., extracting the information present in the lower bits of the RAW signals to make SfM/visual SLAM work. The question is how to do this. It is noteworthy that the goal is not to generate natural looking bright images as SID does but to achieve the optimal performance for SfM and visual SLAM.

There are potentially several directions to achieve the goal. One is to develop a keypoint detector and a feature descriptor that work directly on the RAW-format images. Even if keypoint detectors and descriptors are not good enough, it could be possible to attain the necessary level of matching performance by strengthening the subsequent steps in the pipeline. Recently, CNNs have been applied to these steps, leading to promising results, such as outlier removal in the initial correspondences [29, 30] and establishing initial matching [3]. In parallel to these, the application of image enhancement methods for RAW-format low-light images in a pre-processing stage of image matching could be useful, e.g., SID [5] and others [45, 46]. Methods for more general image restoration would be applied to the RAW-format images [100, 101].

As above, we can think of multiple different approaches to making SfM and visual SLAM methods applicable to low light environments. To promote further studies, we need a dataset to evaluate the above approaches in a multi-faceted fashion. Aiming at widening their application field toward lower-light scenes, it is necessary to examine how underexposed the image will be that each approach can deal with. There is currently no dataset that can be used for this purpose. Considering these, we create a dataset having the following features:

- To examine each method's limit with underexposed images, we acquire multiple RAW-format images at each scene's position with $48 = (6 \text{ shutter speeds} \times 8 \text{ ISO settings})$ exposure settings ranging from extreme to mildly underexposure settings. The cam-

era is mounted on a tripod while capturing all the images.

- We additionally provide long-exposure images, using which as the ground truth, one can evaluate image restoration methods on the task of estimating it from one of the underexposed images.

- The current standard for the evaluation of image matching methods is to measure the accuracy of the downstream task, i.e., the estimation of geometric parameters, as was pointed out in recent studies [102]. Therefore, we acquire images from two positions to form stereo pairs for each scene along with their ground truth relative pose. To obtain the ground truth pose, we capture a good quality image with a long-exposure setting for each scene position.

- The dataset contains diverse scenes consisting of 54 outdoor and 54 indoor scenes.

Using this dataset, we experimentally evaluate several existing component methods for the SfM pipeline, i.e., detecting keypoints and extracting descriptors [8], finding initial point correspondences, and removing outliers from them [3, 29, 103]. We choose classical methods and learning-based methods for each. We also evaluate the effectiveness of image enhancement, including classical image-enhancing methods with/without denoising [104], and a CNN-based method [5, 105]. The results show the importance of using the RAW-format images instead of using the processed images by the standard RIP. They further provide the strengths and weaknesses of the above component methods, also showing that there is room for further improvement.

2.2 Related Work

2.2.1 Matching Multi-view Images

Matching multi-view images of a scene is a fundamental task of computer vision, and its research has a long history. It generally performs the following steps: detecting keypoints/computing local descriptors, establishing initial point correspondences, and removing outliers to finding correct correspondences. A baseline of this pipeline, built upon traditional methods, consists of SIFT [8], SURF [23], etc. for detecting interest points and

extracting their local descriptor, nearest neighbor search in the descriptor's space for obtaining initial correspondences across images, with an optional 'ratio test' step for filtering out unreliable matches [8], and RANSAC for outlier removal [103, 106].

A recent trend is to use CNNs to detect keypoints and/or extract local descriptors. Early studies attempted to learn either keypoint detectors [107–109], or descriptors [110–114]. In contrast, in recent studies, researchers have proposed end-to-end pipelines [2, 25–27, 115, 116] that can perform the two at once. Despite the success of CNNs in many computer vision tasks, it remains unclear that these learning-based methods have surpassed the classical hand-crafted methods. In parallel to the developments of methods for keypoint detectors and descriptors, several recent studies have developed learning-based methods for initial point matching and outlier removal [3, 29].

2.2.2 Datasets for Image Matching

There are many datasets created for the research of image matching [107, 117–122]. Many recent studies of image matching employ HPatches [123]. There are also a number of datasets for visual SLAM and localization/navigation [4, 124–127].

Some of these datasets provide challenging cases, including illumination changes, matching daylight and nighttime images, motion blur in low-light conditions, etc. However, all these datasets provide only images in the regime where the standard RIP can successfully yield RGB images with a well-balanced brightness histogram. This is also the case with a recent study [128] that analyzes image retrieval under varying illumination conditions. Our dataset contains the images of very dark scenes all in a RAW-format with 14-bit depth. In fact, while we have verified the authors' findings in [128] with 8-bit images converted from our RAW-format images using the standard RIP, they do not hold in the case of directly using the RAW-format images, as we will show later.

There are also many evaluation methods for image matching, which are developed aiming at a more precise evaluation [117, 118, 129–131]. A recent study has introduced a comprehensive benchmark for image matching [102]. As in this study, the current trend is to focus on the downstream task; the accuracy of the reconstructed camera pose is chosen as a primary metric for evaluation. Following this trend, our dataset provides the ground truth for the relative camera pose between every stereo image pair.

2.2.3 Image Enhancement

There are many image-enhancing methods that improve the quality of underexposed images. Besides basic image processing such as histogram equalization, there are many methods based on different assumptions and physics-based models, etc., such as global analysis and processing based on the inverse dark channel prior [32, 33], the wavelet transform [34], the Retinex model [35], and illumination map estimation [36]. These methods are proven to be effective for images that are mildly underexposed.

To deal with more severely underexposed images, Chen *et al.*proposed a learning-based method that uses a CNN to directly convert a low-light RAW image to a good quality RGB image [5]. Creating a dataset containing pairs of underexposed and well-exposed RAW images (i.e., the SID dataset), they train the CNN in a supervised fashion. Their method can handle more severe image noise and color distortion emerging in underexposed images than the previous methods. For the problem of enhancing extreme low-light videos, Chen *et al.*extended this method while creating a dataset for training [45]. In parallel to these studies, Wei *et al.*have developed a model of image noises, making it possible to synthesize realistic underexposed images [46]. They demonstrated that a CNN trained on the synthetic dataset generated by their model performs denoising equally well or even better than a CNN trained on pairs of real under/well-exposed images.

While these studies aim sorely at image enhancement, our study considers the problem of matching images of extremely low-light scenes. Our dataset contains stereo image pairs of multiple scenes; there are 48 low-light RAW images with different exposure settings and one long-exposure reference image for each camera position of each scene. It is noteworthy that they include much more underexposed images than the datasets of [5, 45].

2.2.4 Datasets for Image Matching

There are many datasets created for the research of image matching [107, 117–122]. Many recent studies of image matching employ HPatches [123]. There are also a number of datasets for visual SLAM and localization/navigation [4, 124–127].

Some of these datasets provide challenging cases, including illumination changes, matching daylight and nighttime images, motion blur in low-light conditions, etc. However, all these datasets provide only images in the regime where the standard RIP can successfully

yield RGB images with a well-balanced brightness histogram. This is also the case with a recent study [128] that analyzes image retrieval under varying illumination conditions. Our dataset contains the images of very dark scenes all in a RAW-format with 14-bit depth. In fact, while we have verified the authors' findings in [128] with 8-bit images converted from our RAW-format images using the standard RIP, they do not hold in the case of directly using the RAW-format images, as we will show later.

There are also many evaluation methods for image matching, which are developed aiming at a more precise evaluation [117, 118, 129–131]. A recent study has introduced a comprehensive benchmark for image matching [102]. As in this study, the current trend is to focus on the downstream task; the accuracy of the reconstructed camera pose is chosen as a primary metric for evaluation. Following this trend, our dataset provides the ground truth for the relative camera pose between every stereo image pair.

2.2.5 Image Enhancement

There are many image-enhancing methods that improve the quality of underexposed images. Besides basic image processing such as histogram equalization, there are many methods based on different assumptions and physics-based models, etc., such as global analysis and processing based on the inverse dark channel prior [32, 33], the wavelet transform [34], the Retinex model [35], and illumination map estimation [36]. These methods are proven to be effective for images that are mildly underexposed.

To deal with more severely underexposed images, Chen *et al.*proposed a learning-based method that uses a CNN to directly convert a low-light RAW image to a good quality RGB image [5]. Creating a dataset containing pairs of underexposed and well-exposed RAW images (i.e., the SID dataset), they train the CNN in a supervised fashion. Their method can handle more severe image noise and color distortion emerging in underexposed images than the previous methods. For the problem of enhancing extreme low-light videos, Chen *et al.*extended this method while creating a dataset for training [45]. In parallel to these studies, Wei *et al.*have developed a model of image noises, making it possible to synthesize realistic underexposed images [46]. They demonstrated that a CNN trained on the synthetic dataset generated by their model performs denoising equally well or even better than a CNN trained on pairs of real under/well-exposed images.

While these studies aim sorely at image enhancement, our study considers the problem of matching images of extremely low-light scenes. Our dataset contains stereo image pairs of multiple scenes; there are 48 low-light RAW images with different exposure settings and one long-exposure reference image for each camera position of each scene. It is noteworthy that they include much more underexposed images than the datasets of [5, 45].

2.3 Dataset for Low-light Image Matching

2.3.1 Design of the Dataset

We built a dataset of stereo images of low-light scenes and named it the MID (Matching In the Dark) dataset. It contains stereo image pairs of 54 indoor and 54 outdoor scenes (108 in total). We used a high-end digital camera to capture all the images; they are recorded in a RAW format with 14-bit depths. Figure 2.1 shows example scene images. For each of the 108 scenes, we captured images from two viewpoints with 49 different exposure settings, i.e., 48 exposure settings in a fixed range plus one long exposure setting to acquire a reference image. Note that most of the images are so underexposed that the standard RIP cannot yield reasonable RGB images from them.

Some of the 48 images of each scene captured with the most underexposed settings are so underexposed that they appear to store only noises; it will be impossible to perform image matching using them, even if we try every one of the currently available methods. Nevertheless, we keep these images in the dataset to assess the lower limit of exposure to which image matching and restoration methods work, not only existing methods but those to be developed in the future. We designed the dataset primarily to evaluate image matching methods in low-light conditions, but the users can also evaluate image-enhancing methods. Our 48 images of each scene contain more severely underexposed ones than any existing datasets for low-light image enhancement (e.g., [5]).

2.3.2 Detailed Specifications

The dataset contains 10,584 ($= 108$(scenes) $\times$ 2(stereo) $\times$ $(48 + 1)$(exposure settings)) images in total. They are of $6,720 \times 4,480$ pixels and in a RAW format of 14 bits per pixel;

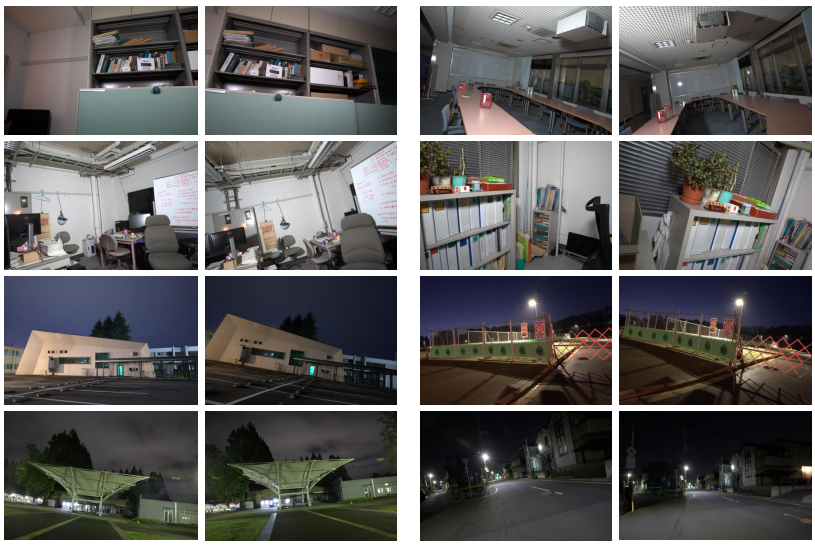


Figure 2.1: Example stereo image pairs (long exposure versions) of four indoor scenes (upper two rows) and four outdoor scenes (lower two rows).

its Bayer pattern is RGGB. We used Canon EOS 5D Mark IV with a full-frame CMOS sensor and EF24-70mm f/2.8L II USM to capture these images.

For each scene, we set up the camera in two positions to capture stereo images. For each position, we mounted the camera on a sturdy tripod while capturing 49 images. We first captured a long-exposure image, which serves as a reference image; we use it to compute the ground-truth camera poses of the stereo pair, as will be explained in Sec. 2.3.3. To capture the reference image, we choose exposure time from the range of 10 to 30 seconds, while fixing ISO to 400.

We then captured the low-light images in 48 different exposure settings that are combinations between six exposure times and eight ISO values. The exposure time is chosen from the range of $[1/200, 1]$ seconds for the indoor scenes and $[1/200, 0.5]$ seconds for the outdoor scenes. The ISO value is chosen from $\{100, 200, 400, 800, 1600, 3200, 6400, 12800\}$.

The indoor scene images were captured in closed rooms with regular lights turned off; the illuminance at the camera is in the range of 0.02 to 0.3 lux. The outdoor scene images were captured at night under moonlight or street lighting. The illuminance at the camera is in the range of 0.01 to 3 lux.

2.3.3 Obtaining Ground Truth Camera Pose

To compare various image matching methods with different local descriptors and keypoint detectors, we need to evaluate the accuracy of the camera poses estimated from their matching results. We consider stereo matching in our dataset, and an image matching method yields the estimate of the relative camera pose between the stereo images. To obtain its ground truth, we use the pairs of the reference images to perform image matching, from which we estimate the relative camera pose for each scene. Following [132], we use it as the ground truth after manual inspection along with correction, if necessary, which we will explain later.

The detailed procedure for obtaining the ground truth camera pose for each scene is as follows.

We first convert the two reference images in the RAW format into RGB space[1]. We then covert each RGB image into grayscale and compute keypoints and their descriptors using the difference of Gaussian (DoG) operator and the RootSIFT descriptor [1]. We next establish their initial point matches using nearest neighbor search with Lowe's ratio test [8] with a threshold of 0.8.

We then estimate the essential matrix by using the 5-point algorithm with the pretrained neural-guided RANSAC (NG-RANSAC) [29]. We employed the authors' implementation for it. We employ NG-RANSAC over conventional RANSAC, since we found in our experiments that it consistently yields more accurate results. Calibrating the camera with the standard method using a planar calibration chart, we decompose the estimated essential matrix and obtain the relative camera pose (i.e., translation and rotation) between the stereo pair.

As mentioned above, we performed a manual inspection of the estimated essential matrix, ensuring they are reliable enough to be used as the ground truths. We did this by checking if any image point on the paired images satisfies the epipolar constraint given by the estimated essential matrix. To be specific, we manually select a point on either left or right image and draw its epipolar line on the other image.

We then visually check if the corresponding point lies on the epipolar line with a deviation less than one pixel. We chose a variety of points having different depths for this check.

[1]Following [5], we used rawpy (`https://pypi.org/project/rawpy/`), a python wrapper for *libraw* that is a raw image processing library (`https://www.libraw.org/`)

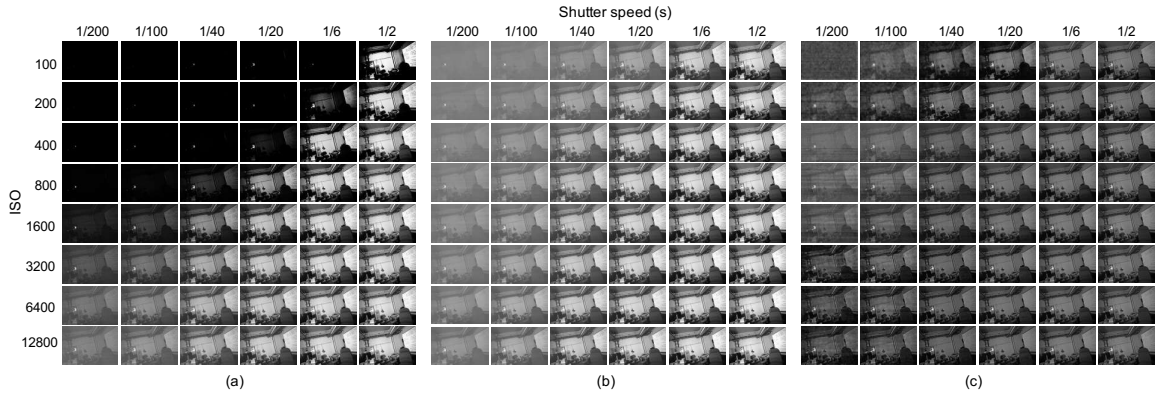


Figure 2.2: Images of a scene captured from the same camera pose that are converted from their RAW-format originals by three conversion methods. (a) **RIP-HistEq**. (b) **Direct-BM3D**. (c) **SID**. See text for these methods.

If an estimated essential matrix fails to pass this test, we either remove the scene entirely or manually add several point matches to get a more accurate estimate of the essential matrix and perform the above test again. All the scenes in our dataset have passed this test.

2.4 Matching Images in Low-light Scenes

This section discusses what methods are applicable to matching low-light images in our dataset. We evaluate those in our experiments.

2.4.1 Conversion of RAW Images to RGB

As there is currently no image matching method directly applicable to RAW-format images, we consider existing keypoint detectors and local descriptors that receive grayscale images. To cope with the low-light condition, we plug image enhancement methods before keypoint detectors and local descriptors, which we will describe later.

It is first necessary to convert RAW-format images into RGB/grayscale images. We have two choices here. One is to use the standard RIP that converts RAW to RGB. As mentioned in Sec. 2.1, the standard RIP often fails to make use of brightness information stored in the lower bits of RAW signals of dark scenes, due to the requirement for versatility against a variety of scenes with different illumination conditions and also the limit of computational resources available to on-board RIP. To confirm its limitation, we evalu-

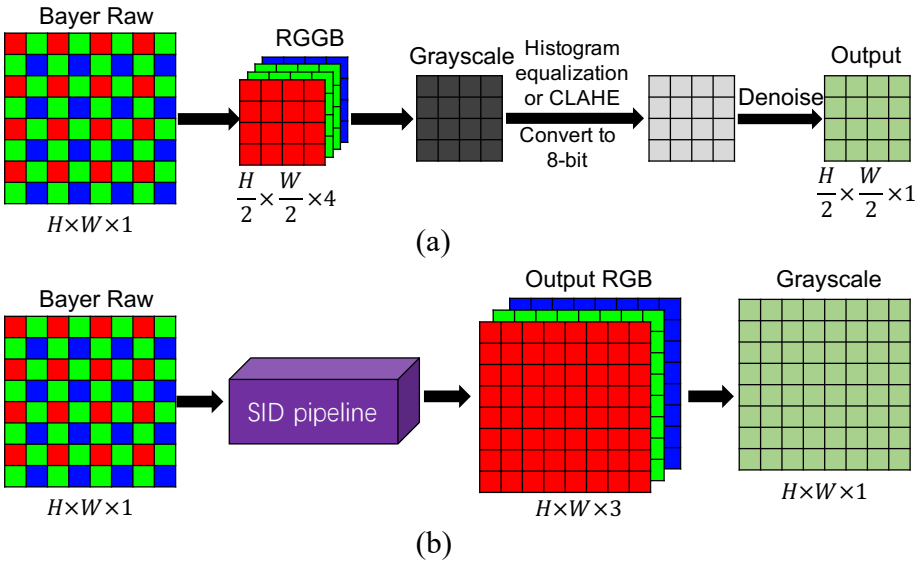


Figure 2.3: The pipelines of two image-enhancing methods. (a) **Direct-HistEq** or **Direct-CLAHE**. (b) **SID**.

ate this standard-camera-pipeline-based conversion in our experiment; we use the LibRaw library using rawpy, a Python image processing module.

The other choice is to do the conversion without using the standard RIP. We will explain this below, because it is coupled with the image enhancement step.

2.4.2 Image Enhancement

Thus, we consider two methods, i.e., using the standard RIP for the RAW-to-RGB conversion and directly using RAW-format images. For each, we consider three different image enhancing methods.

Conversion by standard camera pipeline

When using the standard RIP to convert RAW images, we consider applying the following four methods to its outputs: none, a classical histogram equalization, a contrast limited adaptive histogram equalization (CLAHE), and a CNN-based image enhancement, MIRNet [105]. We choose MIRNet because it is currently the best image-enhancing method applicable to RGB/grayscale images. Figure 2.2(a) shows examples of the standard RIP with histogram equalization. We will refer to the four methods as **standard RIP**, **RIP-**

HistEq, **RIP-CLAHE**, and **RIP-MIRNet** in Sec. 3.5.

Direct Use of RAW-format Images

We consider two approaches. One is to use standard image processing methods to convert RAW to RGB/grayscale images; see Fig. 4.2(a). For this, we employ the following simple approach. Given a Bayer array containing the input RAW data, we first apply black level subtraction to it and then split the result into four channels; the pixel values are now represented as floating point numbers. We then take the average of the two green channels to obtain an RGB image and convert it to grayscale using the OpenCV function `cvtColor`. Next, we perform histogram equalization or CLAHE to improve the brightness of the image. We map the brightness in the range $[m - 2d, m + 2d]$, where m is the average brightness and d is the mean absolute difference from m to each pixel value, to the range $[0, 255]$. Finally, we quantize the pixel depth to 8 bits. We will call this method **Direct-HistEq** or **Direct-CLAHE**.

We optionally apply denoising to the converted image at the final step. We employ BM3D [104] with a noise PSD ratio of 0.08 in our experiments. The resulting image will be transferred to the second step of image matching. Figure 2.2(b) shows examples of the converted images by the method. We will call this method **Direct-BM3D**.

In parallel to the above, we consider a CNN-based image enhancing method that directly works on RAW-format images; see Fig. 4.2(b). We employ SID [5], a CNN trained on the task of converting an underexposed RAW image of a low-light scene to a good quality image. It is designed to receive the RAW data of an image and output an RGB image. We calculate the amplification ratio of SID using shutter speed and ISO values between the underexposed and the reference images. As the output of SID is twice as large as others, we downscale the image size by $2 : 1$ and then convert it into grayscale for image matching; see Fig. 2.2(c). We used the pretrained model provided by the authors , which is trained on the SID dataset. We call this method **SID** in what follows.

2.4.3 Image Matching

We consider matching a pair of images of a scene here. It is to establish point correspondences between images while imposing the epipolar constraint on them and estimate the

camera pose (., a essential or fundamental matrix) encoded in the constraint. The standard approach to the problem is to first extract keypoints and their local descriptors from each input image, establish initial matching of the keypoints between the images, and finally estimate the camera pose from them.

There are at least several methods for each of the three steps. There are many classical methods that do not rely on learning data. As with other computer vision problems, neural networks have been applied to each step. They were first applied to the first step, ., keypoint detectors [107] and descriptors [110, 112–114], to name a few. The next was the third step of robust estimation [29, 30]. Recently, SuperGlue [3] was proposed, which deals with the step of establishing initial point correspondences.

2.5 Experiments

We experimentally evaluate the combinations of several methods discussed in Sec. 2.4 using our dataset.

2.5.1 Experimental Configuration

Compared Methods

We choose both classical methods and neural network-based methods for each step of image matching. As for keypoint detection and local descriptors, we choose RootSIFT [1] and ORB [24] as representative classical methods; we consider ORB because it has been widely used for visual SLAM. We use their implementation of OpenCV-3.4.2. We use SuperPoint [2] [2], Reinforced SuperPoint [3] [133], GIFT [4] [134], R2D2 [5] [27], and RF-Net [6] [135] as representative neural network-based methods. Furthermore, we employ L2-Net [7] [136] and SOSNet [8] [137] as hybrid methods of classical and neural-based methods; they compute local descriptors based on the SIFT keypoints and neural networks. For them, we use the authors' implementation and follow the settings recommended in their papers.

[2] https://github.com/magicleap/SuperPointPretrainedNetwork
[3] https://github.com/aritra0593/Reinforced-Feature-Points
[4] https://github.com/zju3dv/GIFT
[5] https://github.com/naver/r2d2
[6] https://github.com/Xylon-Sean/rfnet
[7] https://github.com/ubc-vision/image-matching-benchmark
[8] https://github.com/ubc-vision/image-matching-benchmark

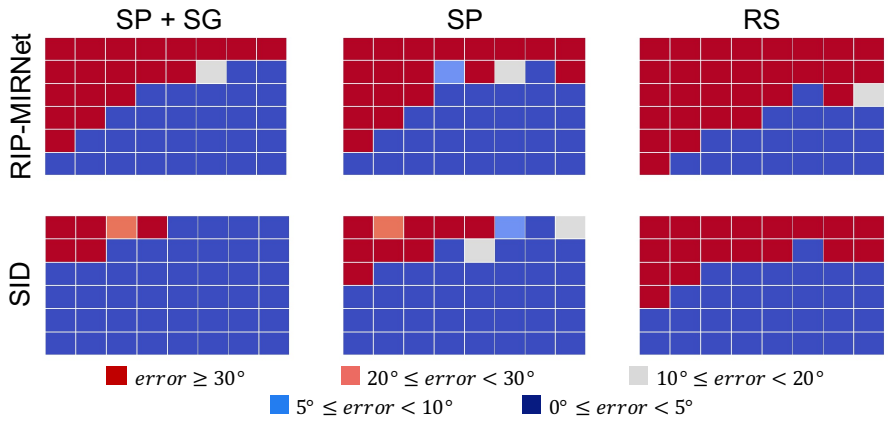


Figure 2.4: Angular errors of the camera pose estimated by several methods for a scene from images with 6×8 different exposure settings. The number of cells with an error lower than a specified threshold quantifies the robustness of the method.

As for outlier removal of point correspondences, we choose RANSAC and NG-RANSAC [29]. We use the OpenCV-3.4.2 implementation of RANSAC with `threshold` $= 0.001$, `probability` $= 0.999$, and `maxIters` $= 10,000$ with the five point algorithm and use the authors' code for the latter. For obtaining initial point correspondences, we use the nearest neighbor search and also SuperGlue [9] [3]. We apply Lowe's ratio test [8] with a threshold of 0.8 to RootSIFT, L2-Net, SOSNet, and RF-Net.

To summarize, we compare the following eleven methods: **SP:** Superpoint + NN + RANSAC, **RSP:** Reinforced SuperPoint + NN + RANSAC, **GIFT:** GIFT + NN + RANSAC, **SP + SG:** SuperPoint + SuperGlue + RANSAC, **R2D2:** R2D2 + NN + RANSAC, **RF:** RF-Net + NN + RANSAC, **L2:** L2-Net + NN + RANSAC, **SOS:** SOSNet + NN + RANSAC, **RS:** RootSIFT + NN + RANSAC, **RS + NG:** RootSIFT + NN + NG-RANSAC, **ORB:** ORB + NN + RANSAC. As for image enhancers, we use the eight methods explained in Sec. 2.4.2., i.e., **standard RIP**, **RIP-HistEq**, **RIP-CLAHE**, **RIP-MIRNet**, **Direct-HistEq**, **Direct-CLAHE**, **Direct-BM3D**, and **SID**. We combine these eight image enhancers with the above eleven image matching methods and evaluate each of the 88 pairs. We resize the output images from each image enhancer to 960×640 pixels and feed it to the image matching step.

[9] `https://github.com/magicleap/SuperGluePretrainedNetwork`

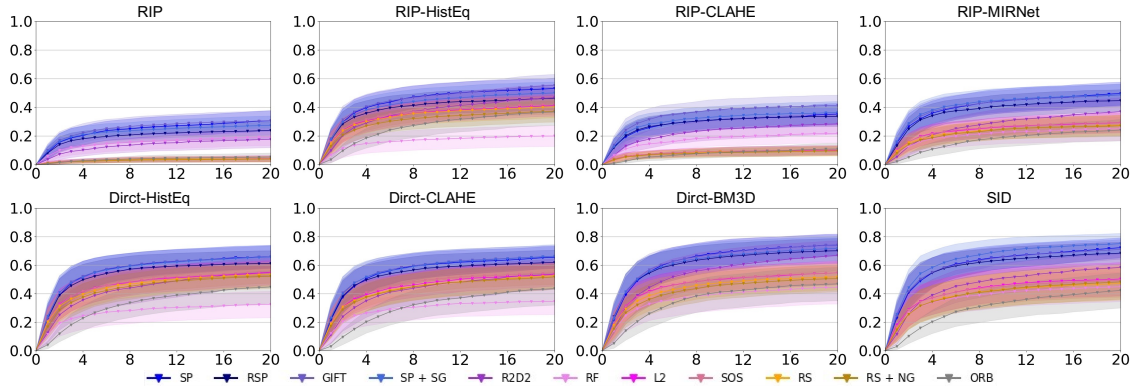


Figure 2.5: The normalized number N_τ of the exposure settings (the vertical axis) for which the estimation error of each method is lower than threshold τ (the horizontal axis). Each panel shows the means and standard deviations over 54 *indoor* scenes for the eleven image matching methods for an image-enhancing method.

Evaluation

We compare these methods by evaluating the accuracy of their estimated relative camera pose. We apply each pair of an image enhancer and an image matching method to the stereo images of each scene. We consider only pairs of stereo images with the same exposure setting; there are 48 pairs per each scene. Thus, we have 48 estimates of relative camera pose for each scene.

To evaluate the accuracy of these estimates, we follow the previous work [3, 29, 30]. Specifically, we measure the difference between the rotational component of the ground truth camera pose and its estimate, as well as the angular difference between their translational components. We use the maximum of the two values as the final angular error. Figure 2.4 shows examples of the results. Each of the colored 6×8 matrices indicate the above angular errors of one of the compared methods for a scene and the 48 exposure settings.

We are interested in how robust each method will be for underexposed images. To measure this, we count the exposure settings (out of 48) for which each method performs well. To be specific, denoting the above angular error of i-th exposure setting by e_i $(i = 1, \ldots, 48)$, we set a threshold τ and count the exposure settings with an error lower than τ as $N_\tau = \sum_{i=1}^{48} 1(e_i < \tau)$, where $1(\texttt{True}) = 1$ and $1(\texttt{False}) = 0$. We normalize N_τ dividing by the total number of exposure settings. As shown in Fig. 2.4, the angular error decreases roughly in a monotonic manner from well-exposed toward underexposed settings. Thus, a larger N_τ means that the method is more robust to underexposure.

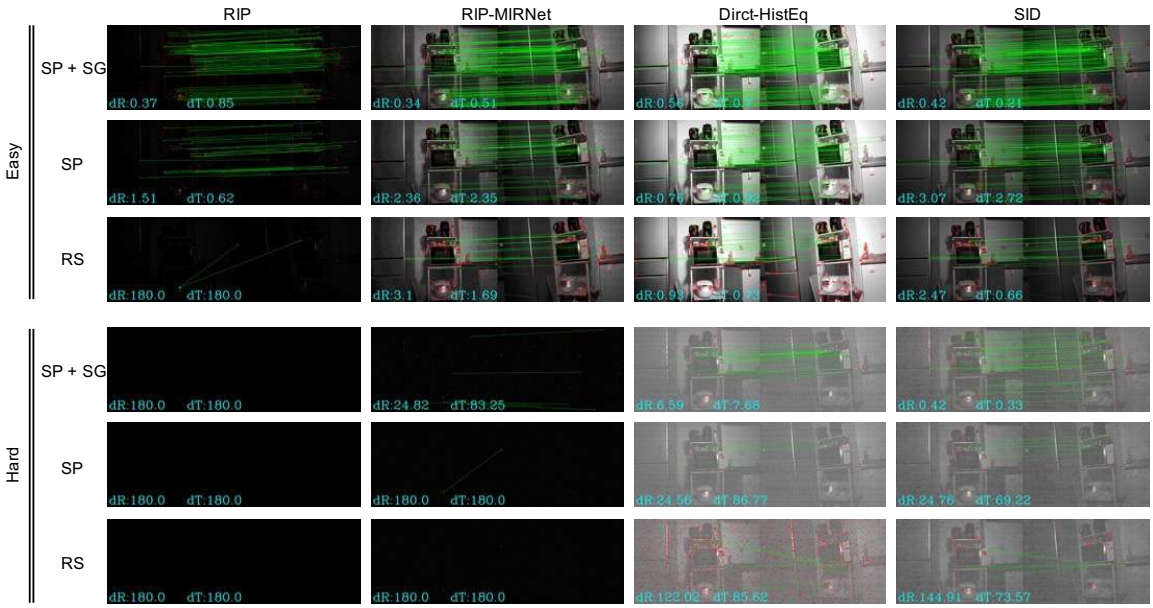


Figure 2.6: Visualization of the matching results for one of the 54 indoor scenes. Point correspondences judged as inliers are shown in green lines. The combination of three matching methods and the four image enhancing methods are applied to two image pairs with different levels of exposure (i.e., 'Easy' and 'Hard').

2.5.2 Results

Figure 2.5 shows the results for the indoor scenes; see Fig. 11 in the supplementary for the outdoor scenes. Table 2.1 shows the mean of N_τ with $\tau = 5°$ over 54 scenes for indoor and outdoor scenes, i.e., the values of the curve in Fig. 2.5 and Fig. 11 at the error threshold $\tau = 5°$. It can be used as a summary of Fig. 2.5 and Fig. 11. We can make the following observations.

First, the overall comparison of the image enhancers indicates the following: *i) Using the standard RIP to convert RAW-format images to 8-bit RGB images before enhancing and matching is inferior to the direct use of RAW-format images.* This shows that the standard RIP cannot utilize the information stored in the low bits of the RAW signals. This fact forms a basis for our dataset.

Next, the overall comparison of the image matching methods yields the following: *ii) SP and its variants are clearly better than the other methods.* For example, SP and GIFT outperform RS and R2D2 in all cases. This may somewhat contradict previous reports [102, 133] that while SP is superior to SIFT in the homography-based evaluation using the HPatches dataset, the superiority is not observed in the evaluation with non-planar scene

matching. Additionally, *iii) SP+SG performs the best in many cases.* However, the gap to other methods considerably differs between the indoor and the outdoor scenes. For the outdoor scenes, the gap to the second-best methods tends to be large, whereas, for the indoor scenes, it is not so large.

The comparison within standard-camera-pipeline-based enhancers indicates the following. *iv) The results of the standard RIP (without any enhancement) are the worst. Comparing RIP-HistEq and RIP-MIRNet, the former is comparable or even better than the latter.* This agrees with the results reported in the recent study of Jenicek and Chum [128], where the authors use 8-bit RGB images outputted from the standard RIP.

Finally, the comparison within the enhancers using RAW-format images shows the following. *v) For the outdoor scenes, the four enhancers show similar performance in many cases. When used with SP+SG, both BM3D and SID perform better than Direct-HistEq and Direct-CLAHE; the two show the best performance. For the indoor scenes, while there is a similar tendency, SID shows a good margin with others only when used with SP+SG.* It is noteworthy that the superiority of SG depends on the chosen image enhancer, regardless of whether they are applied to indoor or outdoor scenes; this tendency cannot be predicted sorely from the performance of SP.

We conclude that if we use SG, we should choose SID for the image enhancer, which achieves the best performance; if we do not, we should use BM3D since it achieves good performance overall. This conclusion differs from that with the standard-camera-pipeline-based enhancers (i.e., (iv)), which is another evidence that the proposed dataset offers what is unavailable in the previous datasets providing only low-bit depth images. Figure 2.13 shows the visualization of a few matching results for an indoor scene.

2.6 Summary and Discussion

This paper has presented a dataset created for evaluating image matching methods for low-light scene images. It contains stereo images of diverse low-light scenes (54 indoor and 54 outdoor scenes). They are captured with 48 different exposure settings, including from mildly to severely underexposed ones. The dataset provides ground truth camera poses to evaluate image matching methods in terms of the accuracy of estimated camera poses.

We have reported the experiments we conducted to test multiple combinations of exist-

ing image-enhancing methods and image-matching methods. The results can be summarized as follows.

- The direct use of the RAW-format images shows a clear advantage over the standard RIP. Using the standard RIP yields only suboptimal performance, as it cannot utilize information stored in the lower bits of RAW-format signals. Moreover, when using the standard RIP, using classical histogram equalization or the state-of-the-art CNN-based image-enhancing method does not make a big difference, as reported in [128].

- SuperPoint and its variants work consistently better than RootSIFT.

- SID is the best image enhancer when using SuperPoint+SuperGlue. Otherwise, BM3D and SID perform equally well and better than the sole use of histogram equalization.

While the above is our conclusion about the combinations of currently available methods, we think there remains much room for improvement. For instance, we manually chose the range of 14-bit RAW signal and converted it into 8-bit images, and applied Superpoint to them. It is observed that the manual method yields significantly better results than the image enhancers tested in this paper, showing that none of the tested methods can choose the best range in the 14-bit RAW signals for image matching; see Sec. B in the supplementary for details. The standard image enhancers are designed to yield images that appear the most natural, which should differ from the best image for image matching. We will explore this possibility in a future study.

2.7 Appendix

2.7.1 Distinction from Existing Datasets

Figure 2.7 shows example images of our dataset and RobotCar [33]. Most of our images are darker than the darkest one of RobotCar. The standard raw image processing yields mostly black images from them. Nevertheless, one can derive sufficient info from their RAW signals, when treating them properly.

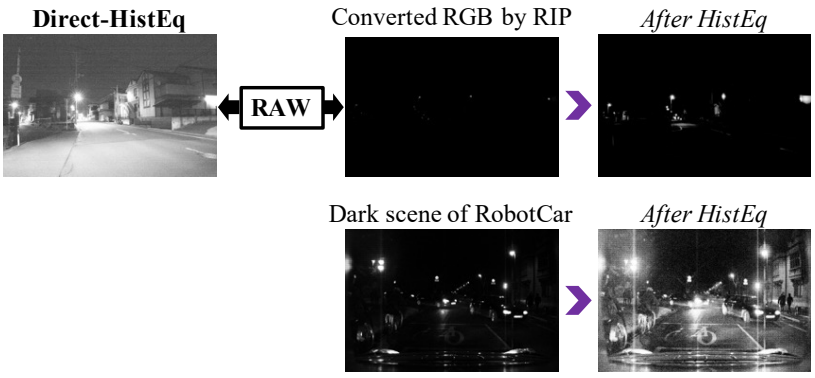


Figure 2.7: Comparison between RobotCar [4] and our dataset.

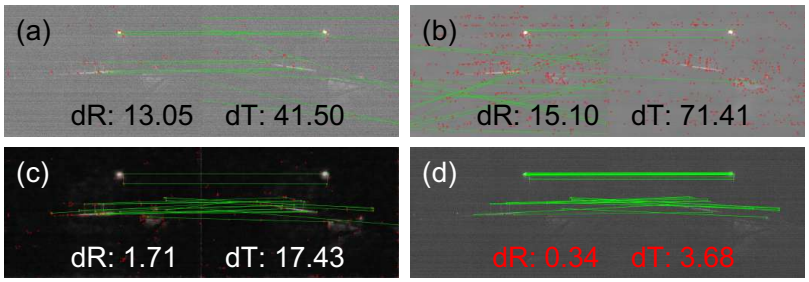


Figure 2.8: Matching results of **SP** with (a) **Direct-HistEq**, (b) **Direct-BM3D**, (c) **SID**, and (d) Images obtained by manual adjustment of brightness range in 14-bits RAW signals.

2.7.2 Performance Comparison to a Manual Adjustment

Figure 2.8 shows the results of using SuperPoint with the three image-enhancing methods and the result obtained by using SuperPoint on a manually converted 8-bit image from the same RAW image. To be specific, we manually chose the range of 14-bit RAW signal and converted it into an 8-bit image. The values of 'dR' and 'dT' indicate the rotation and translation errors for each method. It is observed that the manual method yields significantly better results than others and indicates that there is stil much room for improvement.

2.7.3 All Samples of Scene Images in the Dataset

Figures 2.9 and 2.10 show all samples of indoor and outdoor scenes in our dataset, respectively. All images are obtained from the long exposure RAW-format images by the standard RIP.

2.7.4 More Results of Image Matching

Figure 2.11 shows the normalized number N_τ of the exposure settings for which the estimation error is lower than threshold τ averaged over 54 *outdoor* scenes.

Figure 2.12 shows the average angular errors of the camera pose estimated by the compared 88 methods (i.e., eight image enhancers with eleven image matching methods) over all scenes for each of the 6×8 exposure settings.

2.7.5 Visualization of Matching Results

Figure 2.13 and 2.14 show examples of the visualization of the matching results by the 88 methods for an indoor and an outdoor scene, respectively.

Table 2.1: Averaged number N_τ over 54 scenes of exposure settings for which each method yields a better result than error threshold $= 5°$. Extracted from Fig. 2.5 and Fig. 2.11 in the supplementary. 'R-' means 'RIP-' and 'D-' means 'Direct-.'

	Indoor							
	RIP	R-HistEq	R-CLAHE	R-MIRNet	D-HistEq	D-CLAHE	D-BM3D	SID
SP	0.223	0.421	0.275	0.381	0.548	0.540	0.596	0.583
RSP	0.190	0.379	0.277	0.365	0.523	0.523	0.581	0.577
GIFT	**0.238**	**0.427**	**0.338**	0.390	**0.552**	**0.550**	**0.602**	0.583
SP + SG	0.219	0.400	0.292	**0.404**	0.548	0.544	0.585	**0.619**
R2D2	0.113	0.317	0.192	0.229	0.388	0.383	0.483	0.421
RF	0.138	0.154	0.152	0.192	0.256	0.275	0.346	0.358
L2	0.027	0.323	0.077	0.227	0.442	0.415	0.444	0.394
SOS	0.029	0.333	0.077	0.229	0.438	0.429	0.440	0.392
RS	0.025	0.317	0.071	0.210	0.423	0.404	0.410	0.369
RS + NG	0.023	0.288	0.073	0.202	0.404	0.398	0.388	0.363
ORB	0.029	0.210	0.056	0.125	0.267	0.238	0.296	0.238
	Outdoor							
	RIP	R-HistEq	R-CLAHE	R-MIRNet	D-HistEq	D-CLAHE	D-BM3D	SID
SP	0.233	0.379	0.269	0.352	0.460	0.475	0.502	0.500
RSP	0.215	0.363	0.277	0.335	0.435	0.448	0.494	0.477
GIFT	0.254	0.375	0.321	0.358	0.475	0.477	0.506	0.492
SP + SG	**0.302**	**0.419**	**0.365**	**0.410**	**0.525**	**0.527**	0.577	**0.575**
R2D2	0.104	0.240	0.163	0.188	0.267	0.277	0.373	0.321
RF	0.160	0.146	0.183	0.202	0.225	0.244	0.323	0.325
L2	0.052	0.331	0.096	0.258	0.410	0.423	0.427	0.406
SOS	0.054	0.325	0.096	0.256	0.417	0.413	0.423	0.402
RS	0.046	0.317	0.094	0.242	0.410	0.413	0.404	0.406
RS + NG	0.048	0.296	0.102	0.229	0.388	0.392	0.396	0.375
ORB	0.069	0.213	0.094	0.144	0.265	0.233	0.277	0.217

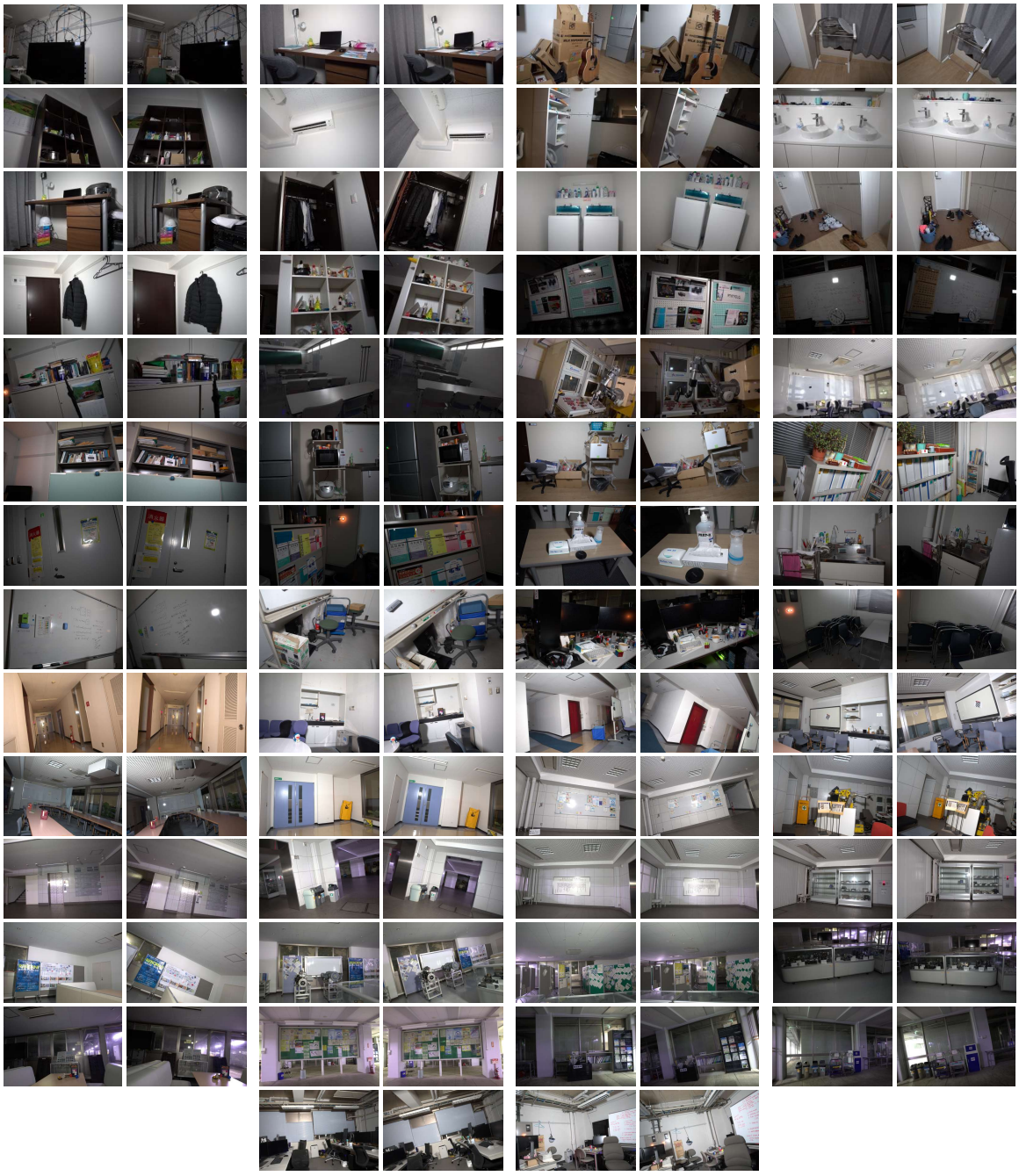


Figure 2.9: Samples of all image pairs (long exposure versions) of the indoor scenes.

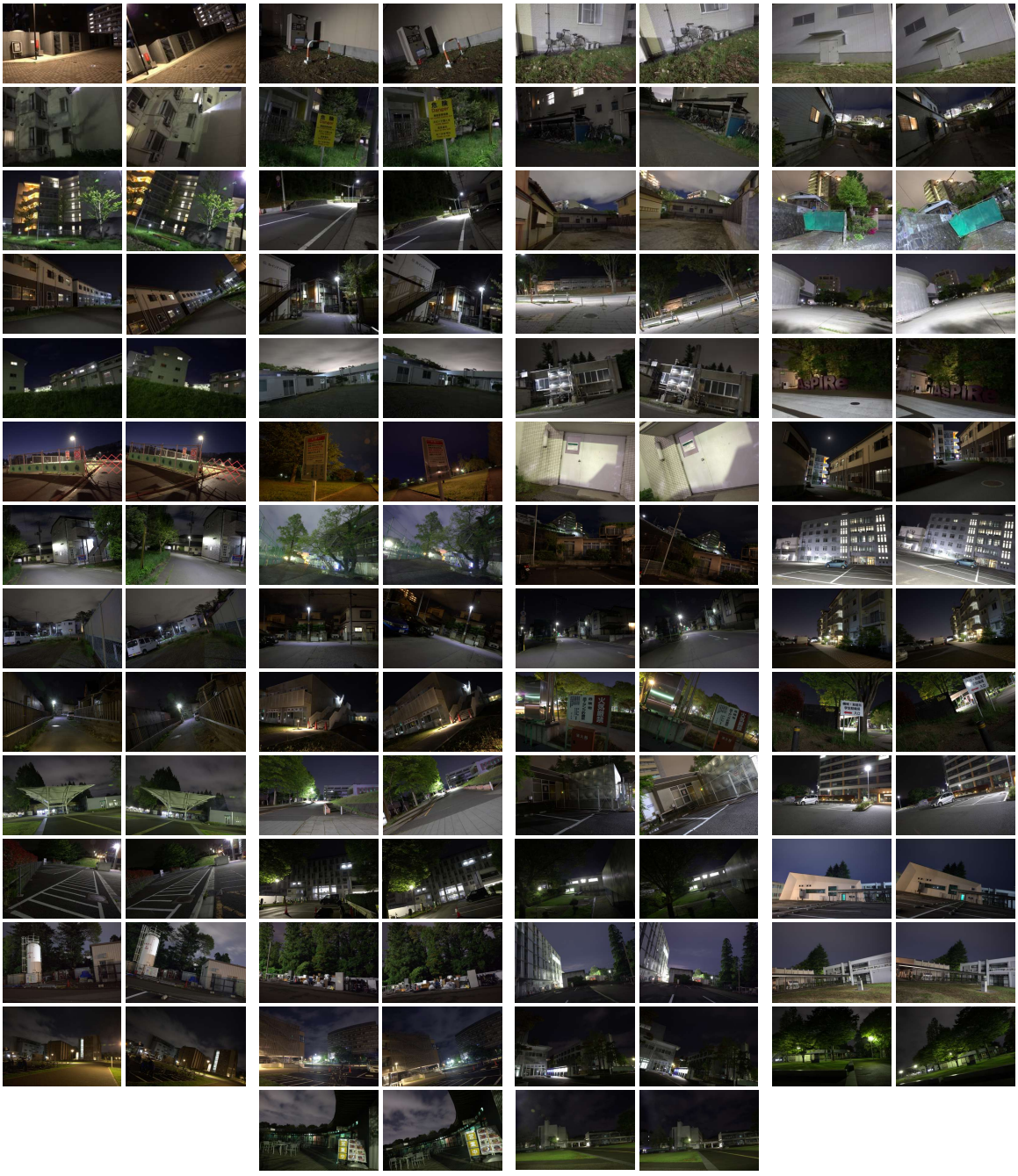


Figure 2.10: Samples of all image pairs (long exposure versions) of the outdoor scenes.

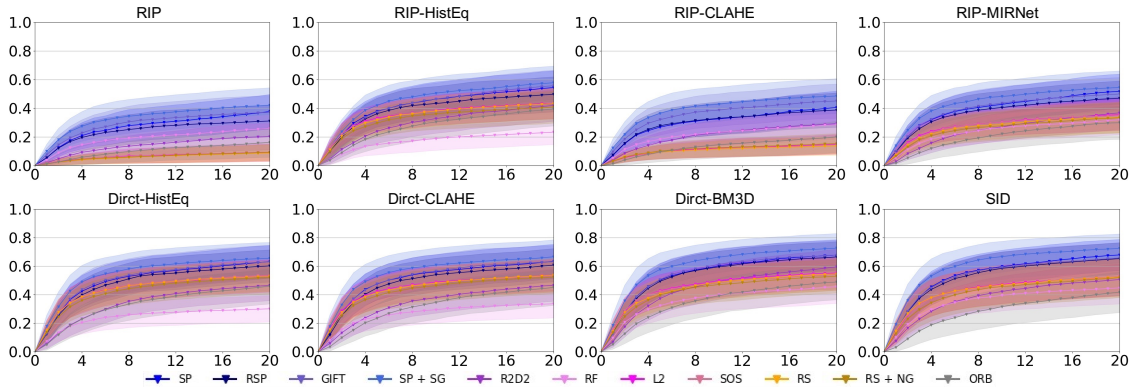


Figure 2.11: The normalized number N_τ of the exposure settings (the vertical axis) for which the estimation error of each method is lower than threshold τ (the horizontal axis). Each panel shows the means and standard deviations over 54 *outdoor* scenes for the eleven image matching methods for an image-enhancing method.

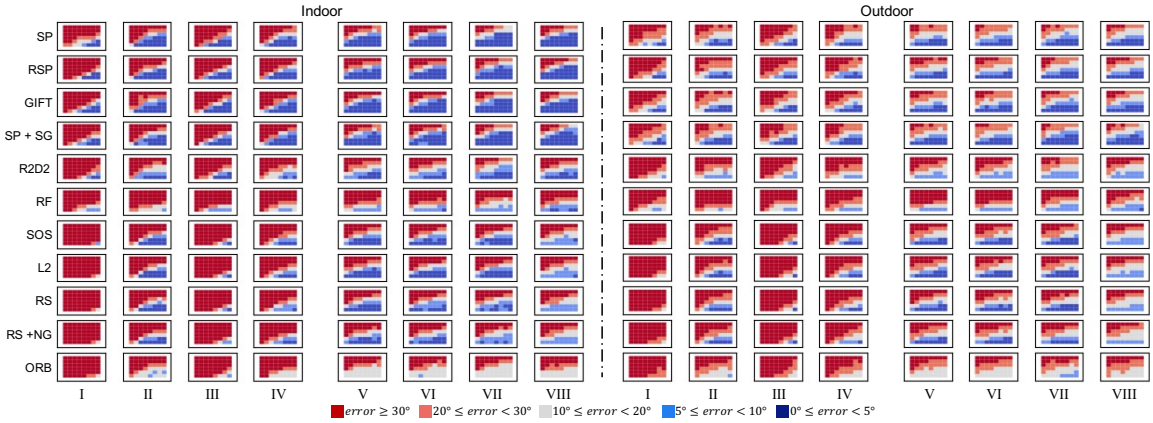


Figure 2.12: Average angular errors of the camera pose estimated by the 88 methods (i.e., eight image enhancers with eleven image matching methods) over all the 54 scenes for each of the 6×8 exposure settings. (I) **RIP**. (II) **RIP-HistEq**. (III) **RIP-CLAHE**. (IV) **RIP-MIRNet**. (V) **Direct-HistEq**. (VI) **Direct-CLAHE**. (VII) **Direct-BM3D**. (VIII) **SID**.

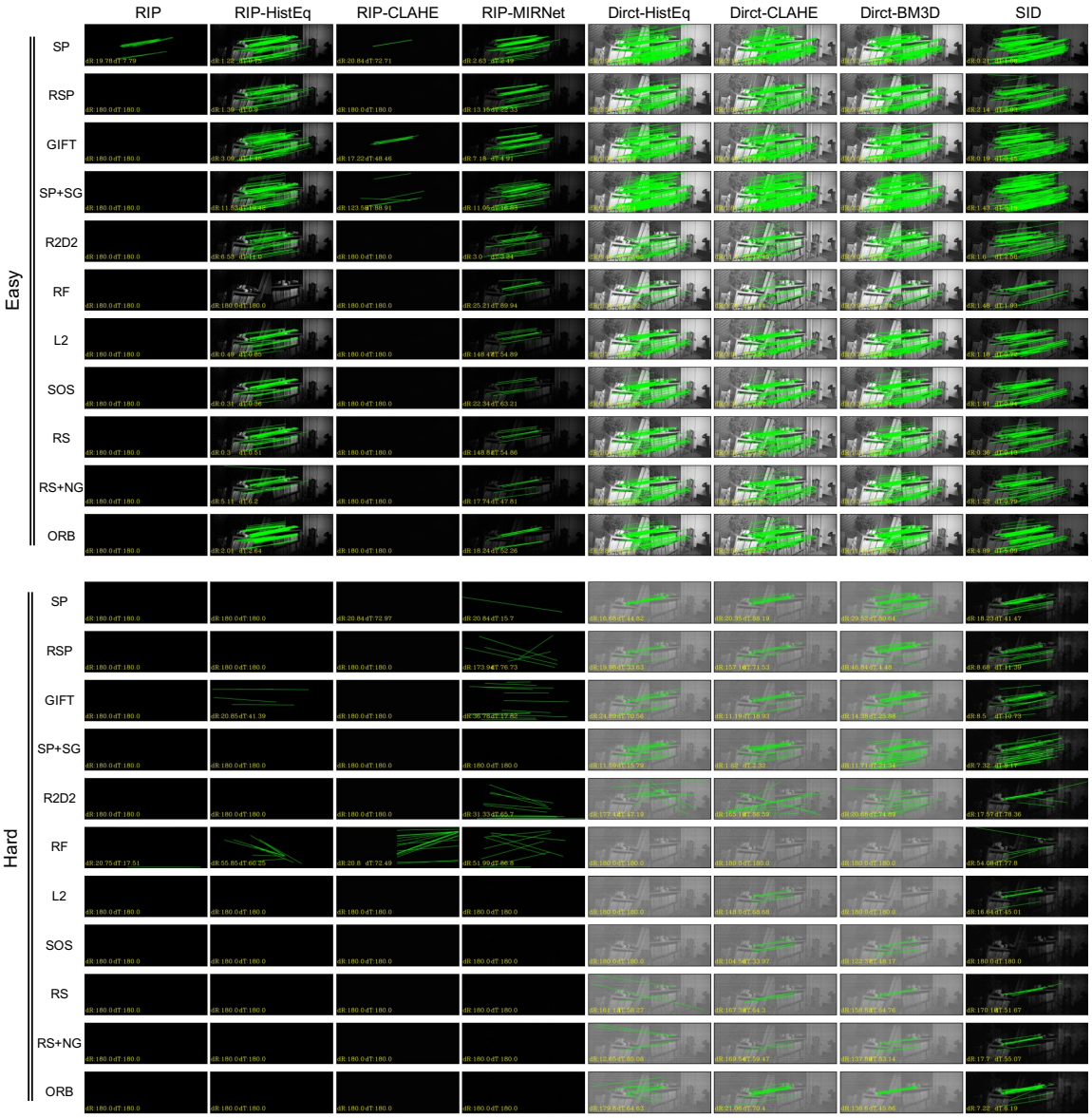


Figure 2.13: Visualization of the matching results for one of the 54 indoor scenes. Point correspondences judged as inliers are shown in green lines. The combination of eleven matching methods and the eight image enhancing methods are applied to two image pairs with different levels of exposure (i.e., 'Easy' and 'Hard').

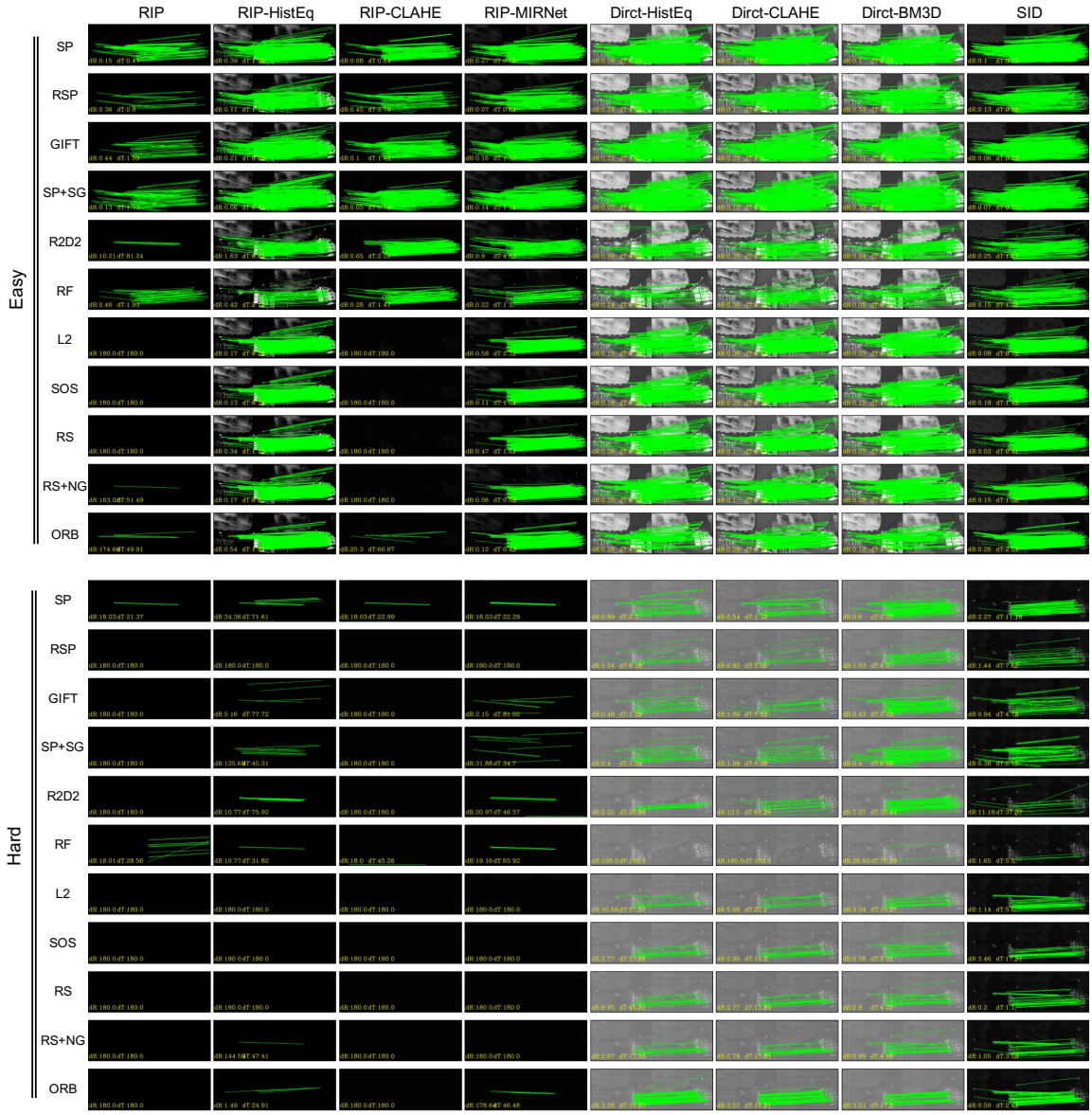


Figure 2.14: Visualization of the matching results for one of the 54 outdoor scenes. Point correspondences judged as inliers are shown in green lines. The combination of eleven matching methods and the eight image enhancing methods are applied to two image pairs with different levels of exposure (i.e., 'Easy' and 'Hard').

Chapter 3

Better Utilizing RAW Images of Low-light Scene

3.1 Introduction

Coping with complex lighting, especially extreme low-light conditions, has always been an important and challenging topic for computer vision tasks related to 3D reconstruction. Such as Structure-from-motion (SfM) or visual SLAM. Although, to a limited extent, we can widen their application fields by improving the generalizability of the model to low-light conditions. However, the essence of this issue is how to obtain high-quality images for downstream tasks.

The advancement of image sensors allows them to record incoming light with more than eight bits (e.g., 14 bits), and the unprocessed data from the camera is first stored in a RAW image. The standard approach to obtaining RGB images is to convert RAW images to RGB vair an Image Signal Processor (ISP). Although many optimization processes or enhancement methods are included in camera ISPs, this transformation process irreversibly converts RAW images with high-precision information into RGB images with lower-bit space and causes information loss. As a result, RGB images do not keep the complete information in RAW images, especially the information existing in the lower bits. This limitation arguably comes from the conflict between versatility against various lighting conditions and reducing the number of bits.

In extreme low-light environments, such as outdoor scenes at night under moonlight or

indoor scenes with insufficient illumination, even when some details of the scenes' brightness are stored in the low bits of their RAW signals, camera ISPs often yield mostly black images. Although many physical settings can be adjusted to obtain brighter images in modern cameras, e.g., opening the aperture, extending exposure time, using higher ISO, or using flash; however, each option is a trade-off. For example, increasing exposure time or using a bigger aperture can introduce blur, high ISO can amplify noise, and flash does not work in open scenes.

Recently, CNNs have been widely applied to various computer vision tasks, leading to promising results. The study of SID (see-in-the-dark) [5] shows that a CNN can learn to convert such RAW images of low-light scenes into brightened images with a natural appearance. It is very likely that we can skip the camera ISP and directly use the information present in RAW images to gain downstream models applied to low-light scenes. The question is how to do this. *It is noteworthy that the goal is not to generate natural-looking bright images as SID does but to achieve the optimal performance for downstream tasks.*

Potentially, we can develop downstream task models that work on RAW images directly. In parallel to these, another idea is that we can add some priors during the training of the image enhancement model to induce the model to produce images that are favorable for downstream tasks. However, there are currently few RAW image datasets that we can use to train downstream task models, especially for 3D reconstruction. Specifically, existing datasets containing RAW images [5, 7, 138] don't meet the requirements for training models for 3D reconstruction, such as image matching and monocular depth estimation. For instance, lack of co-visibility image pairs of one scene, camera intrinsics, and ground truth of depth information.

To promote further studies, we need a dataset that can support model training for more computer vision tasks, including 3D reconstruction. Aiming to widen their application field toward lower-light scenes, the dataset must include underexposed images of various degrees. There is currently no dataset that can be used for this purpose. Considering these, we collected a new dataset, MID-v2. The characteristics of this dataset are as follows:

- We provide data of several indoor and outdoor low-light scenes. For each scene, we selected multiple viewpoints to capture. And we provide a set of RAW images for each viewpoint. The camera is mounted on a tripod while capturing all the images, so that each group of images corresponds strictly.

- There are 12 dark RAW images captured in each set with exposure settings ranging from extreme to mildly underexposure settings. And we additionally provide one long-exposure image in each set, using which as the ground truth.

- We provide ground truths of intrinsics, extrinsics, and depth for each set, which is obtained from a powerful Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline using long-exposure images.

Even though we have such a dataset, as mentioned above, implementing the downstream task methods on such low-light RAW images is still a problem. In this paper, we introduce a novel enhancer for RAW image processing, SuperISP, designed/trained to utilize the information stored in RAW images of low-light scenes to yield better performance for downstream tasks related to 3D reconstruction. It is trained jointly with a weakly supervised signal about feature extraction. Specifically, we consider the global and spatial invariance of the features extracted from the CNN encoder and use it as a weakly supervised signal to train the SuperISP. Then, we experimentally evaluate the SuperISP on two typical downstream tasks related to 3D reconstruction. The SuperISP achieves better performance on the two downstream tasks than other enhancers that aim to obtain a high visual quality image.

3.2 Related Work

3.2.1 Datasets for 3D Reconstruction

There are many datasets created for the research of 3D reconstruction, which include ground truth of depth. Some of them are created based on SfM pipelines [139–142], and also some are based on RGB-D cameras [143–146], or LiDARs [147, 148]. However, few datasets include samples of low-light scenes. Compared to these datasets, our dataset provides the images of very dark scenes all in a RAW with 14-bit depths. On the other hand, there are also a number of datasets created for visual SLAM and localization/navigation [4,124,126,127,148]. Some of these datasets provide challenging cases of low-light scenes. However, all these datasets provide only images and omit ground truth of depths.

3.2.2 Low-light image enhancement

Generally, the image enhancement technique improves the visual quality of the low-quality low-light images. It supports further high-level computer vision tasks to extract valuable information from the captured images. Numerous low-light image enhancement methods have been proposed over last decades, which can be roughly grouped into traditional and deep learning-based methods.

Traditional Methods. The most intuitive and simplest way is to linearly adjust the value rage or execute a nonlinear Gamma correction on inputs. Based on the retinex theory [149], several models were proposed for low-light image enhancement. For instance, SSR [150] first uses the Gaussian blurred input as its illumination map, and then removes the estimated illumination from the input as its final result. MSR [151] extends SSR [150] by fusing the results of multiple Gaussian blur functions with different variances.

Deep learning-based Methods. Recently, deep learning-based methods have dominated the image enhancement community [37–44]. MSR-Net [152] integrates the MSR mechanism into a deep neural network and uses the BM3D [153] for denoising. SID [5] creates a dataset containing pairs of underexposed and well-exposed RAW images, which are used to train a CNN model in supervised fashion. SID [5] can handle more severe image noise and color distortion emerging in underexposed images than the previous methods.

3.2.3 3D Reconstruction tasks

Image matching. Matching multi-view images of a scene is a fundamental computer vision task, which firstly detects keypoints / computes local descriptors, then establish initial point correspondences, and finally remove outliers to find the correct correspondences. The traditional methods consist of SIFT [154], SURF [155], *e.t.c*. With the deep learning developing, numerous methods [2, 25, 105, 108–110, 112, 115, 116, 156] are proposed to use Convolutional Neural Networks (CNNs) to detect keypoint or extract local detectors. Moreover, Several studies [3, 29] are proposed for initial point matching and outlier removal.

Depth estimation. The current depth estimation methods can be roughly grouped into supervised and self-supervised methods. As supervised methods, combining local predictions [157, 158], non-parametric scene sampling [159], through to end-to-end super-

vised learning [160–162] have been explored. The supervised methods require ground truth depth during training. However, this is challenging to acquire the sufficient ground truth in the real-world. Consequently, the various self-supervised approaches are proposed [163, 163–169]. Here, the model is given a set of images as inputs, either in the form of stereo pairs or monocular sequences. By hallucinating the depth for a given image and projecting it into nearby views, the model is trained by minimizing the image reconstruction error.

3.3 Dataset of Low-light Scenes for 3D Vision Tasks

3.3.1 Overview of Dataset

We created a dataset of low-light scenes for 3D vision tasks, named as MID-V2 (Matching In the Dark V2). It contains multi-view images of eight indoor and five outdoor scenes that can be used to reconstruct the scenes with SfM/MVS. We selected static scenes without moving objects to ensure an ideal setting for SfM/MVS.

Each scene contains between 34 and 572 multi-view images (127 on average).We used a high-end digital camera with a lens with minimum distortion to capture all the images; they are recorded in a RAW format with 14-bit per pixel. For each of the viewpoints, we capture a set of images with 13 different exposure settings, i.e., 12 underexposed settings in a fixed range plus one long exposure setting to acquire a reference image. We selected these underexposure settings so that the captured images cannot be converted to normal-quality RGB images by standard camera ISPs.

We create the depth map for each viewpoint of each scene using a state-of-the-art SfM/MVS software, providing it as a ground-truth depth map as a part of the dataset; see Sec. 3.3.3 for details. It also contains the intrinsics and extrinsics of the camera at each viewpoint.

Figure 3.1 shows example images of two indoor and outdoor scenes with their depth maps. Users can train and evaluate methods for 3D vision tasks such as SfM/MVS, image matching, image-enchancement, etc. using this dataset.

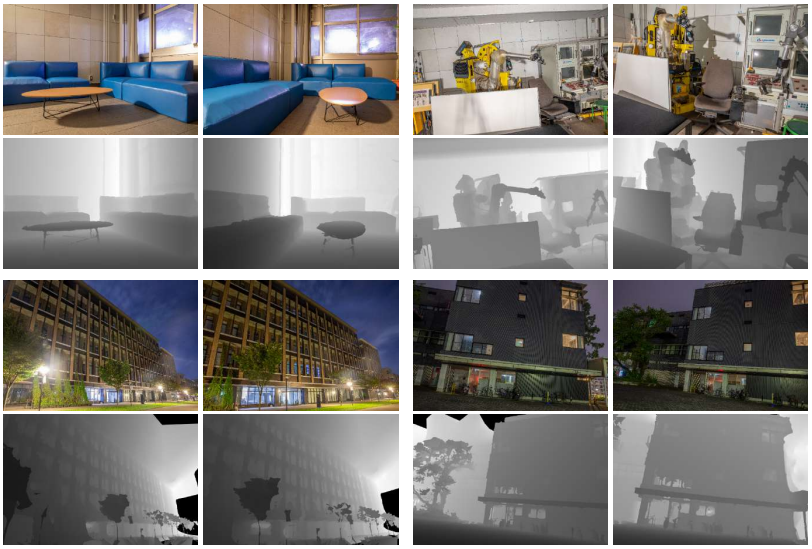


Figure 3.1: Examples of stereo image pairs (long exposure versions) and the corresponding ground truth depth maps.

3.3.2 Detailed Specifications

The dataset contains 19,032 images (17,568 underexposed + 1,464 long exposure) in total. They are of 6,720 $\times$ 4,480 pixels and in a 14 bpp RAW format; its Bayer pattern is RGGB. We used Canon EOS 5D Mark IV with a full-frame CMOS sensor and EF24-70mm f/2.8L II USM to capture these images.

We set up the camera for each scene in multiple viewpoints to capture stereo images. For each view, we mounted the camera on a sturdy tripod while capturing 13 images. We first captured a long-exposure image, which serves as a reference image. We set the exposure time to 20 seconds while fixing ISO to 400 and the aperture to f/8.0. We then captured low-light images at 13 different exposure settings, combining six different exposure times and eight different ISO values. The exposure time is chosen from the range of $[1/200, 0.5]$ seconds and the ISO value is chosen from $\{100, 200, 400, 800, 1600, 3200, 6400\}$.

The indoor scene images were captured in closed rooms at night with regular lights turned off; the illuminance at the camera is in the range of 0.02 to 0.3 lux. The outdoor scene images were captured at night under moonlight or streetlights. The illuminance at the camera is in the range of 0.01 to 3 lux.

Table 3.1: The composition of MID-V2 dataset.

	Indoor									**Outdoor**					
	Scn-1	Scn-2	Scn-3	Scn-4	Scn-5	Scn-6	Scn-7	Scn-test	Total	Scn-1	Scn-2	Scn-3	Scn-4	Scn-test	Total
Views	150	149	71	41	42	34	99	75	661	572	79	100	127	112	990
Depth maps	150	107	55	40	40	22	88	67	569	516	79	98	117	112	922
Images	1,950	1,937	923	533	546	442	1,287	975	8,593	7,436	1,027	1,300	1,651	1,456	12,870

3.3.3 Obtaining Ground Truths

As mentioned above, MID-V2 dataset contains the camera intrinsics / extrinsics and the scene depth maps at each viewpoint of each scene. To estimate them with the quality of ground truth data, we employ a state-of-the-art commercial software for SfM/MVS to estimate them using the reference long-exposure images.

We first convert the reference images in the RAW to high-quality RGB images to obtain the best 3D reconstruction result. We use *Adobe Lightroom*, an image processing software, to process each image with manual adjustments. Specifically, starting with automatically selected settings, we manually adjusted the brightness histogram to decrease excessive highlights and shadows and minimize missing tones.

We then create a 3D model of each scene using the converted references in the RGB format. We use *RealityCapture* to perform SfM to obtain sparse point clouds and camera intrinsics/extrinsics and MVS to obtain dense depth maps seen at each viewpoint. The sparse point cloud for each scene consists of 3D coordinates of keypoints and their camera visibility. The software's good performance was able to produce high-quality results;

see Figure 3.1. *RealityCapture* often fails to estimate scene depths in some local regions or sometimes completely fails recover the depths over the entire image. We do not provide depth maps for the latter viewpoints. Table 3.1 shows the detailed statistics of the dataset.

We completed capturing images at once for all the scenes but outdoor scene-1. For outdoor scene-1, following MegaDepth [139], a dataset containing scene depths, we used semantic filtering to process the obtained depth maps to ensure accuracy. That is, we first detect transient objects in the scene, such as cars, using a pre-trained segmentation network, PSPNet [170]. We then remove the estimated depths in their regions.

3.4 SuperISP: ISP for 3D Reconstruction

3.4.1 Utilizing RAW Images of Low-light Scenes

There are several methods to convert an underexposure RAW image into an RGB-format image, i.e., standard camera ISPs, classical image enhancers based on brightness normalization [149, 151, 171], and modern learning-based image enhancers, such as SID [5]. Previous studies [7] point out that standard camera ISPs cannot make full use of the information stored in RAW data and are empirically shown to be inferior to the other two approaches. The classical image enhances that use normalization operations, such as histogram equalization (abbreviated as HistEq in this paper) and Local Contrast Normalization (LCN), work fairly well to brighten underexposure RAW images. However, they tend to suffer from strong noises in the resulting images.

The CNN-based approaches enhance the RAW images, which provide a better signal-to-noise ratio (SNR) than the classical methods. While many methods have been developed following this approach, they are designed to achieve the highest visual quality. That is, the networks are trained to make the output images as close to their reference image (long-exposure image) as possible with supervised by $L1$ loss.

However, our goal here is to achieve good quality in 3D reconstruction, not high visual quality. While there is some overlap between the two, there are also differences. As shown in previous studies [7], SID and its variants do not show a clear advantage over the classical approach using HistEq followed by denoising with BM3D.

3.4.2 Architectural Design

Based on the above consideration, we introduce a novel CNN-based enhancer for RAW image processing, named SuperISP. SuperISP is designed/trained to utilize the information stored in RAW images of low-light scenes to yield best 3D reconstruction. Example outputs from SuperISP and SID are shown in Figure 3.2.

SuperISP consists of three components, *i.e.*, normalization processing, denoiser and channel attention fusion. The structure of SuperISP is shown in Figure 3.3.

Normalization. Given a Bayer RAW image $\mathcal{X} \in \mathbb{R}^{H \times W \times 1}$, we first perform black-level subtraction and unpack it into four channels (RGGB) denoted as $\boldsymbol{R} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 4}$.

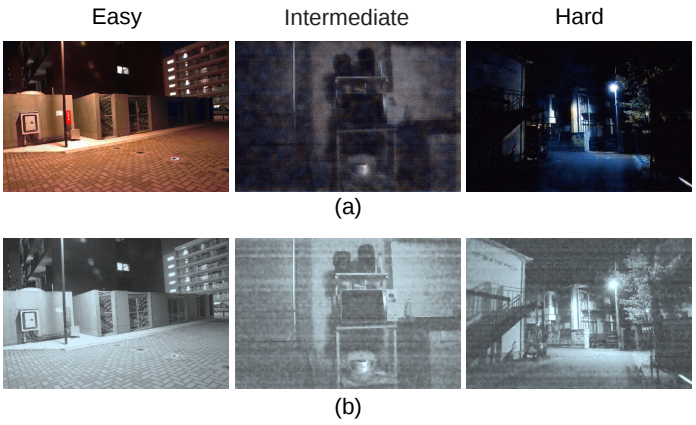


Figure 3.2: Comparison of outputs between (a) SID [5] and SuperISP under easy, mild intermediate and hard exposure settings.

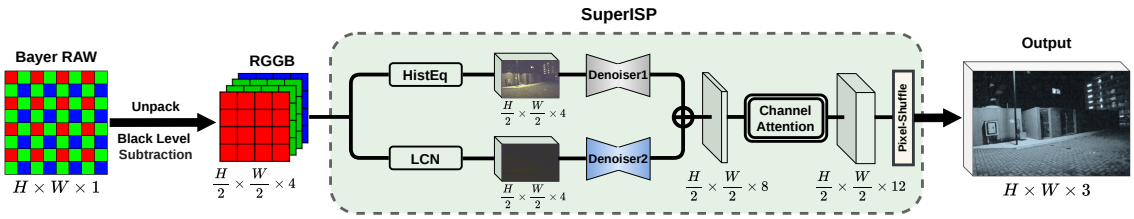


Figure 3.3: The framework of SuperISP. The Bayer Raw images are first performed "unpacking" and Black Level Subtraction options to obtain the RGGB images, which then are input into SuperISP. SuperISP consists of normalization, denoiser and channel attention fusion. The outputs of SuperISP are used for the downstream tasks learning.

Then we process the $\boldsymbol{R}$ by using two kinds of normalization, *i.e.*, histogram equalization (HistEq) denoted as $\mathcal{N}_H$ and local contrast normalization (LCN) denoted as $\mathcal{N}_L$. Formally:

$$T_H, T_L = \mathcal{N}_H(R), \mathcal{N}_L(R) \tag{3.1}$$

where $T_H, T_L \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 4}$, denotes the outputs of $\mathcal{N}_H$ and $\mathcal{N}_L$, respectively. It is worth noting that the normalization operation used in SID [5] is that multiply an amplification ratio to $\boldsymbol{R}$, where the ratio is set to be the exposure difference between the input and reference images. However, it isn't easy to estimate a suitable ratio in practice due to complex lighting conditions and camera parameter settings. *e.g.*, the brightness of an image depends not only on the exposure time but also on ISO and aperture.

Denoiser. Since $\mathcal{N}_H$ and $\mathcal{N}_L$ output different noise distributions on T_H and T_L, we design two denoisers for them, respectively. Specifically, we use two scaled-down versions

(half-channels) of U-Net [157] as denoisers, denoted as $\mathcal{U}_H$ and $\mathcal{U}_L$, respectively. Note that the two denoisers perform denoise in a RAW-to-RAW manner, which is different from SID [5] performing RAW-to-RGB manner. The output channels of $\mathcal{U}_H$ and $\mathcal{U}_L$ are $\frac{H}{2} \times \frac{W}{2} \times 4$. Then $\mathcal{U}_H(T_H)$ and $\mathcal{U}_L(T_L)$ are concatenated as the input of the channel attention fusion module. Formally:

$$\boldsymbol{P} = Concat\left[\mathcal{U}_H(T_H), \mathcal{U}_L(T_L)\right] \tag{3.2}$$

where $\boldsymbol{P} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 8}$. In practice, these two denoisers are pre-trained on MID-V2 dataset supervised by L_1 loss, respectively.

Channel attention fusion. This module fuses information from $\mathcal{U}_H(T_H)$ and $\mathcal{U}_L(T_L)$. The fused information is input to a pixel-shuffle module to generate the final output. Formally:

$$\boldsymbol{I} = \mathcal{S}(\mathcal{A}_C(\boldsymbol{P})) \tag{3.3}$$

where $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$, $\mathcal{A}_C$ denotes the channel attention layer and $\mathcal{S}$ denotes the operation of efficient sub-pixel convolution with a stride of $1/3$.

We end up with the output of SuperISP $\boldsymbol{I}$. We keep the $\boldsymbol{I}$ have the three channels so that it can be unified with downstream task models trained on RGB images. The difference is that SuperISP does not compress it to 8-bit space.

3.4.3 Weakly Supervised Training Strategy

Different from the previous image enhancement approaches, *e.g.*, SID [5], in which, a reconstruction loss is used to induce the output of model only look like the reference image, this work aims to make the output of SuperISP good for downstream tasks learning. Consequently, SuperISP is trained in a weakly supervised strategy. Our approach is illustrated in Figure 3.4.

Given a pair of RAW image patches$(\mathcal{X}_1, \mathcal{X}_2)$, SuperISP outputs a pair of images (I_1, I_2). In this process, we first employ a reconstruction loss to supervise the distribution of I_1 and I_2 similar with the reference images, *i.e.*, long-exposure images.

$$\mathcal{L}_R^i(I_i, G_i) = w_1 \cdot L_1(I_i, G_i) + w_2 \cdot SSIM(I_i, G_i) \tag{3.4}$$

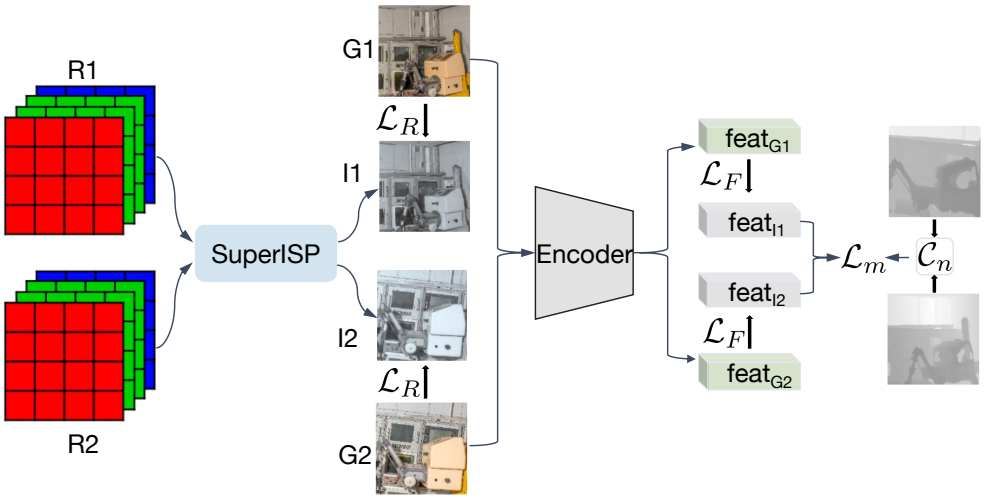


Figure 3.4: SuperISP train strategy.

where G_i denotes the corresponding reference RGB image, $L_1(\cdot)$ and $SSIM(\cdot)$ represent the L_1 loss and structural similarity index measure (SSIM) loss, respectively. w_1 and w_2 are weighting factors.

The output pair of SuperISP (I_1, I_2) and the reference image pair (G_1, G_2) are input into an encoder $\mathcal{F}$, *i.e.*, ResNet-18 [172]. Then we obtain the feature maps of I_1, I_2, G_1 and G_2, denoted as $\mathcal{F}(I_1), \mathcal{F}(I_2), \mathcal{F}(G_1)$ and $\mathcal{F}(G_2)$. Specifically, we discard the last three blocks of ResNet-18 [172]. Consequently, the channel of $\mathcal{F}(\cdot)$ is $\frac{H}{16} \times \frac{W}{16} \times 256$. Then the mean squared error (MSE) loss is applied between I_i and G_i, which aims to align the feature maps between low-light images and long-exposure images. The loss function is formulated as:

$$\mathcal{L}_F^i(\mathcal{F}(I_i), \mathcal{F}(G_i)) = MSE(\mathcal{F}(I_i), \mathcal{F}(G_i)) \tag{3.5}$$

In practice, we detach the gradient of $\mathcal{F}(G_i)$ during the training process to eliminate the ambiguity of feature alignment. Intuitively, the loss $\mathcal{L}_F$ can be regarded as a global supervision signal.

In order to supervise the generation of local features, we employ a cross matching loss that emphasizes the spatial invariance of local features, which is important for 3D vision tasks. Specifically, we perform a ranking loss, *i.e.*, Average Precision (AP) loss [**?**, **?**] to train our model. $\mathcal{C} : I_1 \leftrightarrow I_2$ is defined as the ground-truth correspondences between I_1 and I_2. Specifically, if the pixel (i, j) of I_1 corresponds to pixel (i', j') of I_2, then $\mathcal{C}_{ij} = (i', j')$. Actually, $\mathcal{C}$ can be estimated by using the ground-truth depth maps and extrinsics of I_1 and

I_2

If there are N corresponding local features between $\mathcal{F}(I_1)$ and $\mathcal{F}(I_2)$, *i.e.*, $\mathcal{C}_n : I_1 \leftrightarrow I_2 \in N$, the cross matching loss is defined as:

$$\mathcal{L}_M(\mathcal{F}(I_1), \mathcal{F}(I_2)) = \frac{1}{N}\sum_{n=1}^{N}[1 - AP(\mathcal{F}(I_1), \mathcal{F}(I_2), \mathcal{C}_n)] \tag{3.6}$$

where $AP(\cdot)$ denotes the ranking metric AP for N corresponding local features. The final loss function is formulated as:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_R + \beta\mathcal{L}_F + \gamma\mathcal{L}_M \tag{3.7}$$

α, β and γ are weighting factors. We use $\mathcal{L}_{total}$ supervise the SuperISP and encoder in the end-to-end manner. As for the downstream tasks learning, we discard the encoder and use the output I of SuperISP as the input of the downstream tasks directly.

3.5 Experimental Settings

We experimentally evaluate the proposed SuperISP and some other RAW enhancers on two downstream tasks, *i.e.*,image matching and monocular depth estimation.

3.5.1 Datasets

We evaluate the image matching and monocular depth estimation on MID dataset [7] and MID-V2 test set. Totally, the MID dataset [7] includes 108 scenes, *i.e.*, 54 indoor scenes and 54 outdoor scenes. MID-V2 test set includes 1 indoor scene and 1 outdoor scene. As for one scene, the MID [7] contains RAW images captured from one pair of viewpoints, and MID-V2 test set contains images captured from multi-viewpoints. As for one viewpoint, the MID dataset [7] includes 48 images of different exposure settings, and MID-V2 test set contains 13 images. Both of them include the ground truth of camera poses and intrinsics that is used for evaluating image matching task. As for monocular depth estimation, the existing evaluation datasets do not provide low-light RAW images, therefore we perform the evaluation on MID-V2 test set. Specifically, the ground truth of depth is obtained via SfM+MVS. The obtained depth cannot be used directly, because it

is only up to an unknown scale factor. However, as mentioned in Eigen *et al.* [161], the ratios of depths are preserved under scaling. Consequently, we can perform the evaluation on such ambiguous depths after fine-tuning the model.

In the evaluation of image matching, we consider pairs of stereo images with different exposure settings. We fix one image with a middle-level exposure setting (*i.e.*, *ISO: 400 Exposure time: 0.05* in MID [7] and *ISO: 800 Exposure time: 0.025* in MID-V2 test set). Then we perform image matching with the co-visible keypoints on its paired images taken under all other different settings. We perform the image matching evaluation on all co-visible viewpoints provided by the MID dataset [7]. However, In MID-V2 test set, in order to provide the enough difference of viewpoint as that in MID [7], we randomly select 100 co-visible viewpoints to perform the evaluation for both the indoor and outdoor scenes. In the evaluation of monocular depth estimation, all the images with depth ground truth are used for evaluation.

3.5.2 Evaluation Metrics

Image Matching. We evaluate the image matching methods by measuring the camera pose error between the results estimated by methods and the ground truth. Specifically, under one of the exposure settings, we measure the rotational and the translational component of camera pose. We use the maximum of the two values as the final angular error [**?**] under the current exposure setting. We set a threshold τ and count all the exposure settings with an error lower than τ, denoted as N_τ. We normalize N_τ as the final score, dividing by the total number of exposure settings.

Monocular Depth Estimation. We adopt six evaluation metrics which are commonly used in the previous work [139, 161, 173], *i.e.*, Abs Rel, Sq Rel, RMSE, RMSE log, δ<1.25, δ<1.25^2 and δ<1.25^3.

3.5.3 Compared methods

Conversion of RAW Images to RGB

The RAW images cannot be directly used for the downstream tasks. We select two ways to convert RAW image to RGB images in order to compare with SuperISP.

The first one is to use the camera ISP to convert RAW images to RGB images Specifically, we use the LibRaw library, *i.e. rawpy*, a Python image processing module.

The second one is to do the conversion plug image enhancement methods that work on RAW images. Specifically, we consider two kinds of approaches, *i.e.*, the traditional method and the CNN-based method. In traditional method, we follow the work [153]. We first apply black-level subtraction to the Bayer RAW image, then perform HistEq to improve the brightness, and split the result into RGGB. We further perform BM3D denosie [153] with PSD ratio of 0.08 on each channel. The two green channels are then merged into one to obtain the RGB image which pixel range is mapped into $[0, 255]$. Finally, we quantize the pixel depth to 8 bits. We name it as Hist-BM3D. In CNN-based method, we consider a recent work SID [5]. We provide two kinds of SID model. (1) The pre-trained SID model, denoted as SID. (2) Considering the domain shift between SID dataset and ours, we retrain the SID model on MID-V2 train set. We name it as SID-R. The difference in the implementation of the study [5] is that we calculate the amplification ratio using shutter speed and ISO values between underexposed and reference images.

Image Matching

We compare five popular methods: SP-SG: SuperPoint [2] + SuperGlue [3] + RANSAC, D2-Net: D2-Net [26] + Nearest Neighbor (NN) + RANSAC, LoFTR: LoFTR [31] + RANSAC, D2-Net-f and LoFTR-f. " *-f " indicates that the method is respectively finetuned by using the outputs RGB images provided by the enhancement methods (as mentioned in Sec 3.5.3). Because the ISP is difficult to yield significant RGB images, we only use the outputs of the other four enhancement methods for finetuning, *i.e.*, His-BM3D, SID, SID-R and SuperISP. When we finetune the image mathching methods, the input images are resized to 960×640.

Monocular depth estimation

In practice, we adopt the method proposed in MegaDepth [139] which is trained in supervised way by using ground truth depth obtained from SfM/MVS. We further use the images provided by the enhancement methods as described in Sec. 3.5.3 (except ISP) to finetune the model. All images are resized to 640×480.

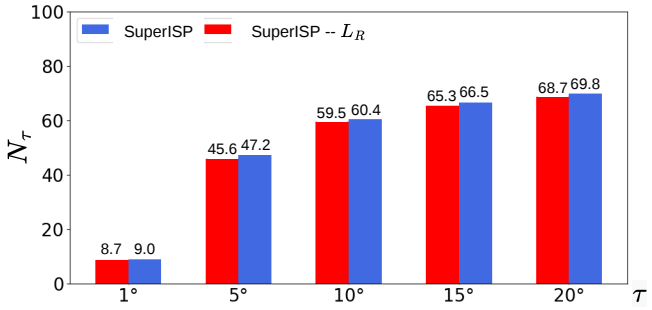


Figure 3.5: Camera pose estimation results of SP-SG with SuperISP and SuperISP – $\mathcal{L}_R$, respectively. "SuperISP – $\mathcal{L}_R$" means the SuperISP only supervised by $\mathcal{L}_R$.

3.5.4 Results Comparison

As for evaluation results for the image-matching task, Table 3.2 and Table 3.3 shows the evaluation results on MID [7] and MID-V2 test set, respectively. We provide the mean results of N_τ with $\tau = 5°$, $10°$ and $20°$ over indoor and outdoor scenes.

Second, when we compare the performance of different image matching methods, we observe that (1) SP-SG and LoFTR, which cover the matching process, are clearly better than D2-Net that only provide keypoint detection and description. Moreover, SP-SG and LoFTR perform differently under the indoor and the outdoor scenes. Specifically, LoFTR outperforms SP-SG under the indoor scene of MID [7], and is inferior to SP-SG under the outdoor scenes of MID [7] and MID-V2 test set. It is because LoFTR is better suited for the indoor scene which consist more indistinctive regions with low texture and repetitive patterns. (3) The performance of D2-net obtains significant improvement after fine-tuning. However, under all image enhancement methods, the performance of LoFTR turns bad after fine-tuning. The reason is that MID-V2 training set provide more hard samples, which is unfriendly to the LoFTR learning. Even so, we can still observe the advantage of SuperISP compared to other methods.

Table 3.4 shows evaluation of monocular depth estimation task. We observe that SuperISP obtains the best or the second best performance on all metrics, even it is not designed for the specific task. The observation is consistent with that in image matching task.

Table 3.2: Camera pose estimation results on MID [7] dataset. The number N_τ with the error threshold $\tau = 5^\circ$, 10° and 20°

	Indoor				
	ISP	Hist-BM3D	SID	SID-R	**SuperISP**
	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°
SP-SG	0.1 / 0.2 / 0.5	36.2 / 40.5 / 45.0	53.5 / 58.5 / 63.1	48.7 / 52.2 / 56.2	**56.1 / 61.2 / 67.0**
D2-Net	0.0 / 0.0 /1.0	22.4 / 27.6 / 34.6	25.2 / 30.9 / 37.8	22.0 / 27.6 /34.2	**27.9 / 34.0 / 41.2**
D2-Net-f	–	37.0 / 44.4 / 52.9	40.6 / 48.0 / 55.8	35.7 / 42.5 / 49.9	**43.7 / 51.1 / 59.0**
LoFTR	0.2 / 0.5 / 1.2	50.5 / 54.5 / 59.2	54.8 / 58.9 / 63.9	51.0 / 55.1 / 59.9	**57.7 / 61.9 / 66.6**
LoFTR-f	–	45.8 / 50.1 / 55.3	50.1 / 55.2 / 60.5	48.5 / 53.0 / 57.2	**55.5 / 59.9 / 64.8**
	Outdoor				
	ISP	Hist-BM3D	SID	SID-R	**SuperISP**
	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°
SP-SG	5.7 / 7.8 / 10.6	33.3 / 38.8 / 45.2	46.2 / 52.0 / 58.6	41.2 / 47.7 / 54.5	**47.2 / 53.8 / 61.0**
D2-Net	0.4 / 0.8 / 3.3	10.2 / 15.1 /22.6	**13.3 / 18.9 / 26.9**	11.3 / 17.0 / 24.9	10.6 / 16.4 / 24.7
D2-Net-f	–	18.8 / 27.0 / 37.1	20.9 / 28.5 / 38.0	17.9 / 25.2 / 35.3	**19.6 / 27.7 / 38.1**
LoFTR	2.3 / 3.2 / 5.1	37.1 / 43.3 / 50.2	**42.5 / 48.4 / 54.9**	41.2 / 47.7 / 54.5	40.9 / 47.1 / 54.5
LoFTR-f	–	30.6 / 36.5 / 43.2	32.5 / 38.7 / 46.1	30.2 / 36.8 / 43.8	**34.9 / 41.9 / 49.7**

Table 3.3: Camera pose estimation results on test dataset. The number N_τ with the error threshold $\tau = 5°$, $10°$ and $20°$.

	Indoor				
	ISP	Hist-BM3D	SID	SID-R	**SuperISP**
	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°
SP-SG	0.1 / 0.2 / 0.3	19.5 / 23.1 / 27.4	22.3 / 27.3 / 32.9	24.5 / 29.1 / 34.3	**27.5 / 32.5 / 38.0**
D2-Net	0.0 0.2 0.9	5.3 / 7.3 / 10.8	4.7 / 7.5 / 12.0	**5.9 / 8.7 /** 13.1	5.5 / 8.6 / **13.2**
D2-Net-f	–	9.4 / 13.1 / 18.7	8.8 / 12.5 / 17.6	**20.2 / 26.4 / 34.1**	12.2 / 16.4 / 22.3
LoFTR	0.2 / 0.3 / 0.8	17.1 / 20.1 / 24.4	19.5 / 22.9 / 27.2	18.8 / 22.9 / 27.8	**20.7 / 24.5 / 29.6**
LoFTR-f	–	11.8 / 14.8 / 18.6	12.0 / 14.9 / 18.9	**18.6** / **22.2** / 26.3	18.5/ **22.2 /26.8**
	Outdoor				
	ISP	Hist-BM3D	SID	SID-R	**SuperISP**
	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°	5° / 10° / 20°
SP-SG	39.2 / 44.1 / 50.5	75.4 / 79.8 / 84.1	81.8 / 84.8 / 88.0	83.3 / **87.2** / **90.6**	**83.8** / **87.2** / 90.4
D2-Net	6.5 / 10.1 / 15.9	29.1 / 37.2 / 46.4	37.5 / 45.1 / 53.4	**37.0 / 44.2 / 52.5**	32.1 / 39.5 / 48.7
D2-Net-f	–	52.5 / 59.8 / 68.2	55.5 / 62.8 / 70.3	**59.5 / 66.7 / 74.3**	55.7 / 64.0 / 71.9
LoFTR	24.1 / 28.8 / 35.3	63.5 / 67.8 / 73.8	66.2 / 70.2 / 75.2	66.5 / 70.0 / 74.7	**68.8 / 72.3 / 77.3**
LoFTR-f	–	57.8 / 62.5 / 68.8	56.8 / 60.7 / 66.1	62.9 / 67.4 / 73.1	**66.7 / 70.8 / 76.8**

Table 3.4: Monocular depth estimation results on MID-V2 test dataset. "↓" means the lower the better, "↑" means the bigger the better.

	Indoor						
	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Hist-BM3D	0.62	5.27	8.17	1.59	0.05	0.14	0.29
SID	0.68	6.06	8.80	2.07	0.02	0.07	0.16
SID-f	0.61	5.24	8.20	1.53	0.06	0.15	0.29
SuperISP	**0.58**	**4.80**	**7.86**	**1.39**	**0.06**	**0.16**	**0.31**
	Outdoor						
	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Hist-BM3D	0.38	1.57	3.86	**0.91**	0.37	0.63	0.79
SID	0.40	1.60	3.97	1.48	0.34	0.59	0.74
SID-f	**0.37**	**1.43**	3.71	1.17	0.38	0.65	0.79
SuperISP	0.38	1.48	**3.62**	0.94	**0.40**	**0.67**	**0.81**

Figure 3.6: Comparison of reconstructed detail. (a) SuperISP – $\mathcal{L}_R$ (b) SuperISP

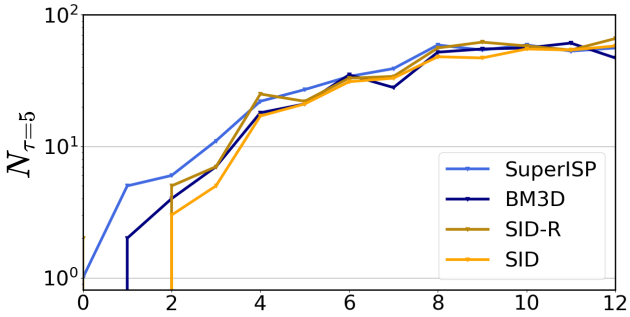


Figure 3.7: Camera pose estimation results of D2-net-f with different exposure settings. Abscissa axis is the exposure settings from hard to easy.

3.6 Ablation study and Analysis

We analyze the importance of weakly supervised training to downstream task learning. Specifically, We compare SuperISP and SuperISP – $\mathcal{L}_R$ supervised only by $\mathcal{L}_R$. Experimentally, we evaluate SP-SG on the the outdoor dataset of MID [7] with more detail τ. The result is shown in Figure 3.8. We observe that the performance of SuperISP – $\mathcal{L}_R$ shows a "decrease" phenomenon. It further confirms that image enhancement methods that aim to obtain a high visual quality image cannot better utilize information in low-light RAW images for downstream tasks. Furthermore, the visualization results in the detail of output images of different enhancers are given in Figure 3.6. We can observe that the SuperISP tends to out an image with more apparent textures than it supervised only by $\mathcal{L}_R$.

We further evaluate the image enhancers on image matching task under different exposure settings. The result is shown in Figure 3.9. To eliminate the possible effects for the

image matching model caused by the domain shift of training data, we select D2-Net-f as the baseline. Specifically, we evaluate the four enhancers on MID-V2 indoor test set. The results show that the advantage of SuperISP is obvious in severe exposure settings. This indicates the limitation of other enhancers that aim to obtain a high visual quality image.

Please refer to supplementary materials for more ablation studies, including more image matching and monocular depth estimation results on other scenes and visualization results of SuperISP and other image enhancers.

3.7 Summary and Discussion

This paper present a low-light RAW image dataset for 3D vision construction. The dataset is captured with 13 different exposure settings, ranging from mildly to severely underexposed. The dataset provides the ground truth of camera pose and depth. Moreover, we propose an image enhancement method, *SuperISP*, which outperform the other enhancers under two downstream tasks, *i.e.*, image matching and monocular depth estimation.

We evaluate multiple combinations of image enhancement methods and task-specific methods. The experimental results show that compared with other image enhancers, SuperISP obtains the best performances on two downstream tasks, which indicates that SuperISP exploits the information in low-light images and provides effective information for downstream tasks.

Limitation. In this paper, we provide a large scale low-light RAW dataset, *i.e.* MID-V2. Although it provides images from different viewpoints under multi scenes, The current scale still cannot meet the needs of practical applications in the real world. Even so, MID-V2 provides a reliable benchmark to the research of 3D reconstruction tasks under low-light scene.

3.8 Appendix

3.8.1 Implementation Details

As explained in Sec. 4.2, we apply two normalization methods to the RGGB inputs. For the first one, HistEq, we perform the histogram equalization on each of the four channels, so that output is in the range of [0,1]. For the second normalization, LCN, we employ the PyTorch implementation[1] that uses a Gaussian filter.

Then, we pre-train the two denoisers on our training set. The reference RAW images (*i.e.,* the long-exposure images) are used as the ground truth, for which the same normalization as the RGGB patches is applied. We follow the training procedure of SID [10], which uses image patches cropped randomly from the RGGB; the patches are resized to 512×512.

We train SuperISP on image patches with the size of 256×256, randomly cropped from the RGGB. Specifically, we first determine the corresponding image pairs using the sparse 3D point clouds derived from SfM. To do this, we compute the overlap between a pair of images by counting the number of 3D points they share; then, we find the corresponding image pairs if it exceeds a pre-defined threshold. Next, we randomly choose one of the shared 3d points and crop the patches whose centers are its projections in the two images. We also crop the corresponding patches from the reference RGB images and depth maps. Note that the cropped patches have different sizes; the patch size for the RGB references and the depth images is 512×512.

To start the training, we initialize the parameters of the two denoisers with their pre-trained weights, as explained in Sec. 4.2. We initialize the parameters of the CNN encoder using pre-trained weights with ImageNet. We freeze its parameters in the first training epoch and unfreeze them from the second epoch when the outputs of SuperISP become relatively stable.

We pre-train the two denoisers and SuperISP on the training set of MID-V2, where we employ the Adam optimizer. We set the learning rate to 10^{-3}, batch size to 40. We update the two denoisers for 200,000 iterations. For the training of SuperISP, we set the learning rate to 10^{-4}, batch size to 4, and train it for 20 epochs.

[1] https://github.com/shufanwu/SegNet-PyTorch/blob/master/lcn.py

3.9 Details of the Experimental Setting

As explained in Sec. 5, we first convert the RAW images to RGB in the experiment using the image enhancers explained in Sec. 5.3.1. We further use these converted images to evaluate several pre-trained downstream models directly or fine-tune them on the output of each enhancer before evaluation.

Image matching. We use a brute-Force matcher to establish initial matching between the images for D2-Net. Regarding the camera pose estimation, we adopt RANSAC as a robust estimator for outlier removal with the five-point algorithm. We use the OpenCV-4.5.5 implementation of RANSAC with $\texttt{threshold} = 0.001$, $\texttt{probability} = 0.999$, and $\texttt{maxIters} = 10,000$. For SuperGlue and LoFTR, we used the pre-trained models for outdoor scenes that are provided by their authors. We fine-tune D2-Net and LoFTR on the MID-V2 training set, where we use the implementations provided by the authors.

Monocular Depth Estimation. We fine-tune the pre-trained model provided by MegaDepth [32] on the MID-V2 training set. It is worth noting that we adopted the loss function that combines $\mathcal{L}_{data}$ and $\mathcal{L}_{gard}$; see [32] for details.

3.9.1 More Ablation Studies

We report experimental results for different scenes in the main paper. Figure 3.8 shows the results of the pose estimation by SP-SG on the indoor dataset of MID [49] with a larger set of τ's, which corresponds to Figure 5 of the main paper. Figure 3.9 shows the pose estimation results of D2-net-f with different enhancers on the outdoor test dataset of MID-V2, which corresponds to Figure 7 in the main paper.

3.9.2 More Examples of Scenes in the Dataset

Figures 3.10 and 3.11 show more examples of our dataset, including indoor and outdoor scenes with ground truth depth maps. All images are obtained from the long exposure RAW-format images by the *Adobe Lightroom*.

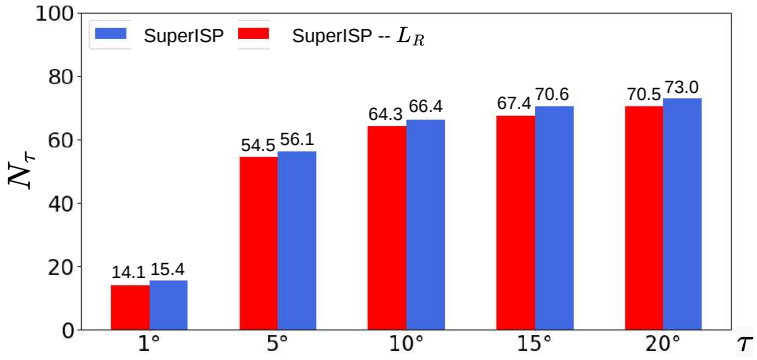


Figure 3.8: Results of camera pose estimation by SP-SG with SuperISP and SuperISP – $\mathcal{L}_R$. "SuperISP – $\mathcal{L}_R$" means the SuperISP trained with only $\mathcal{L}_R$.

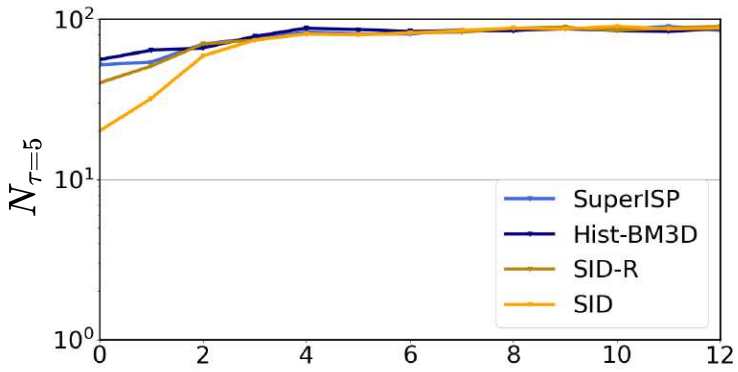


Figure 3.9: Results of the camera pose estimation by D2-net-f with different exposure settings. The abscissa represents exposure settings from hard to easy.

3.9.3 Visualization Results of Image Matching and Monocular Depth Estimation

Figure 3.12 shows several examples of the results of image-matching by each of the combinations of SP+SG and five image-enhancing methods. Figure 3.13 shows visualization of the results of monocular depth estimation by the combinations of the fine-tuned model and four enhancers.

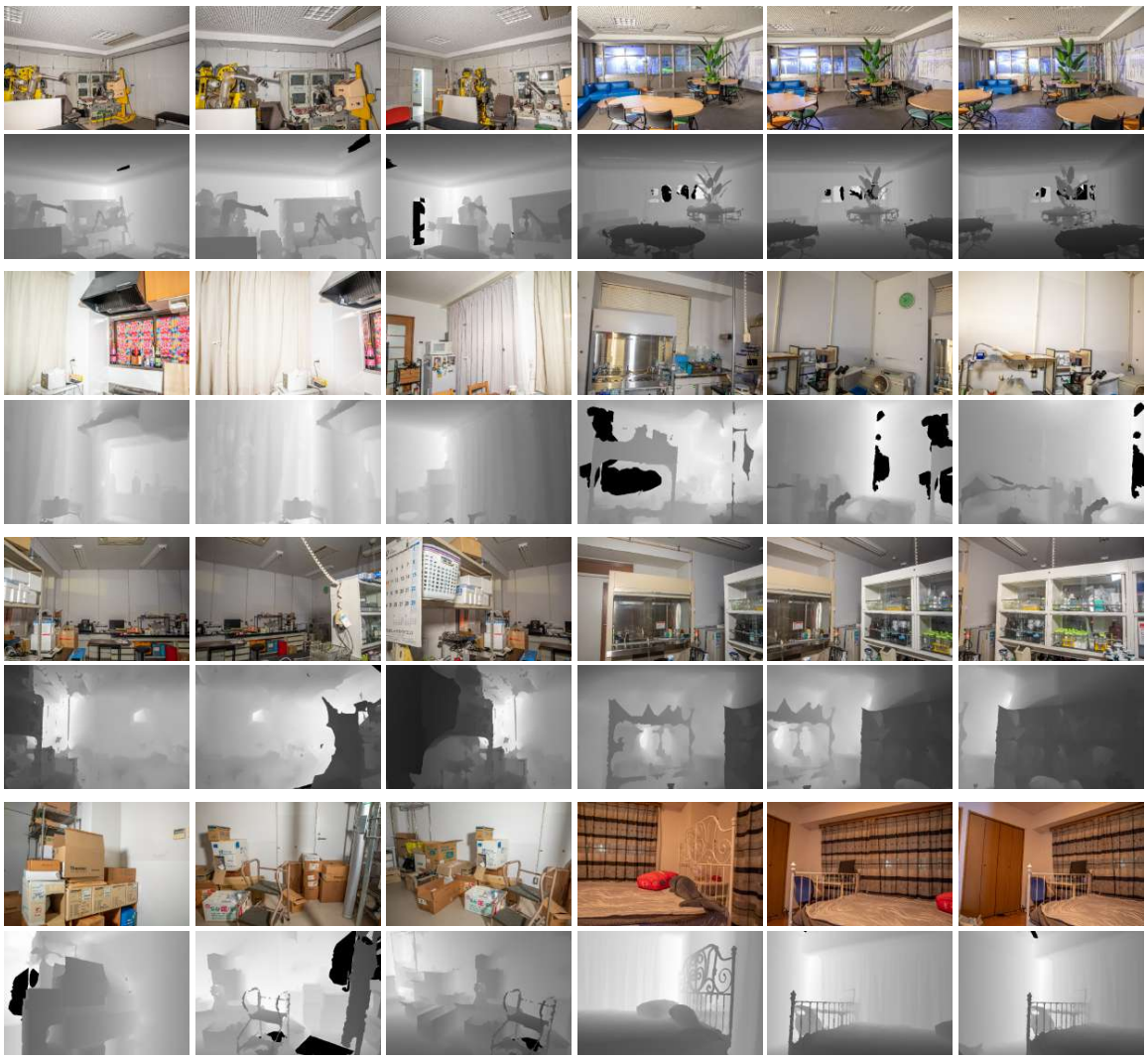


Figure 3.10: Examples of the indoor scenes in the MID-V2 dataset. The reference images (long exposure versions) and the depth maps are shown.

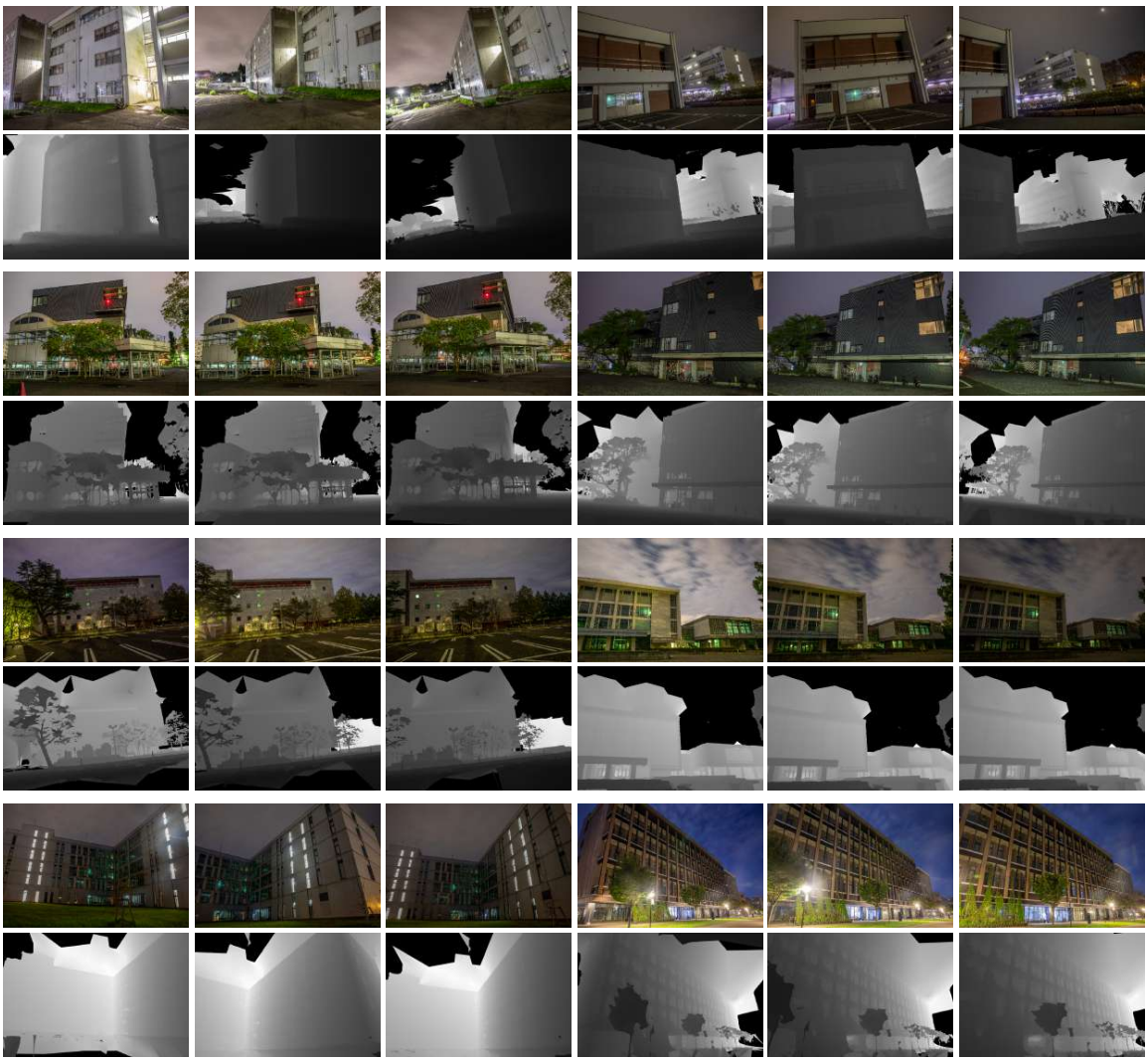


Figure 3.11: Examples of the outdoor scenes in the MID-V2 dataset. The reference images (long exposure versions) and the depth maps are shown .

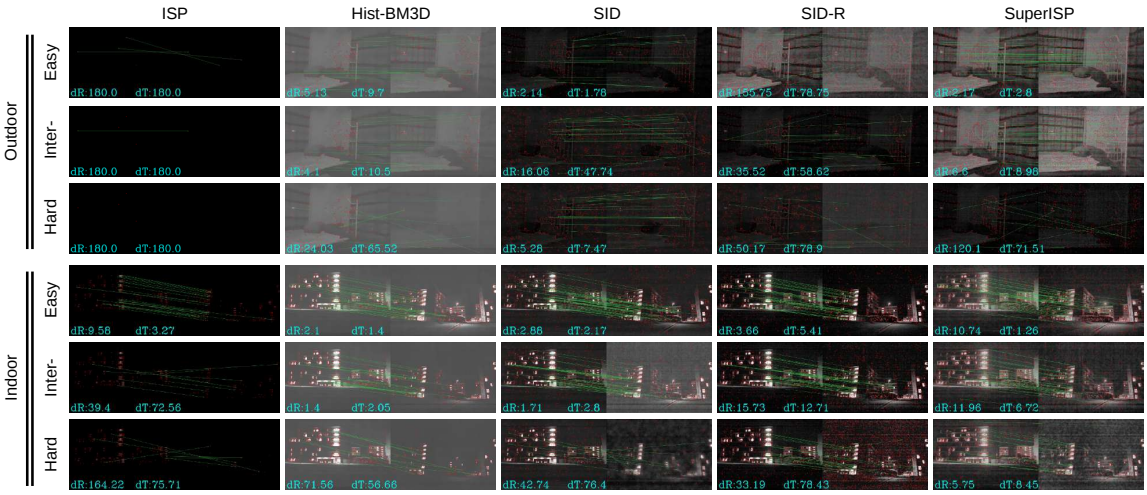


Figure 3.12: Visualization of image matching for one indoor and one outdoor scenes. Point correspondences judged as inliers are shown with green lines. The combination of SP+SG and the five image enhancing methods are applied to three image pairs with different levels of exposure (*i.e.,* 'Easy', 'Inter-', and 'Hard').'Inter-' means 'Intermediate'.

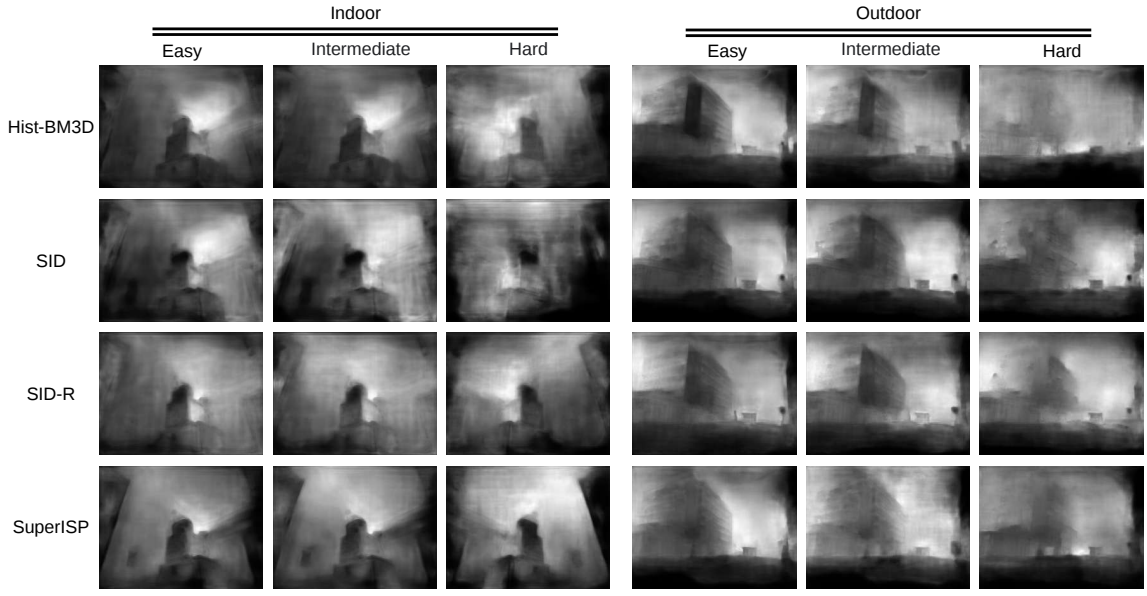


Figure 3.13: Visualization of monocular depth estimation for one indoor and one outdoor scenes. Note that black means near and white means far. The combination of the fine-tuned model and the four image enhancing methods are applied to three images with different levels of exposure (*i.e.,* 'Easy', 'Intermediate', and 'Hard').

Chapter 4

Unifying Local and Global Features for Visual Localization

4.1 Introduction

Visual localization is a key component in computer vision tasks such as Structure-from-Motion (SfM) or SLAM, which is a fundamental problem in numerous applications, such as autonomous driving, mobile robotics, or augmented reality. This growing range of applications of visual localization calls for reliable operation in both large-scale changing indoor and outdoor environments, irrespective of the weather, illumination, or seasonal changes. Visual localization is the problem of estimating the 6 Degree-of-Freedom (DoF) camera pose from which a given image was taken relative to a reference scene representation.

Advanced visual localization approaches are hierarchical, encapsulating image retrieval problems and 6-DoF camera pose estimation [17, 19, 164]. i.e., given a query image of the current view, related candidates are determined by performing image retrieval in a database; then, perform local feature matching between the query image and related candidates to establish point correspondences between 2D to 3D, solving a PnP and estimate 6 DoF camera pose of the query image.

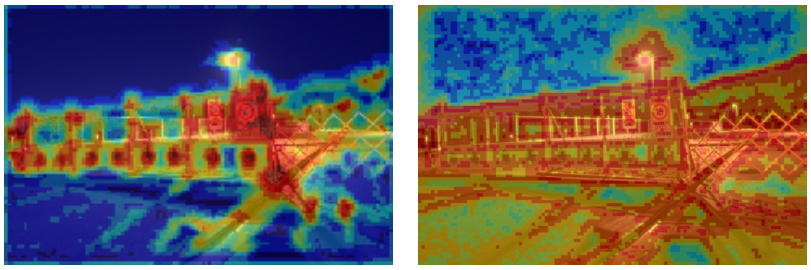
Naturally, two types of image features are needed to perform such a hierarchical visual localization approach, i.e., global image features for retrieval and local image features for image-matching. However, existing studies struggle to unify these two types of image features. For instance, the image retrieval task also involves these two types of features for

the two-stage retrieval strategy [174], i.e., global features are used for large-scale coarse retrieval to get prior candidates, then re-rank it based on the number of inliers obtained from local feature matching with geometry verification. However, such local features tend to deviate from needs from 6-DoF pose estimation. Specifically, local features yielded by a retrieval model always tend to be semantically rich and lose spatial invariance [175, 176]. Moreover, retrieval models don't provide key-point detection, so these local features lack corresponding exact pixel coordinate information.

Most studies perform two models perform hierarchical visual localization. i.e., one for image retrieval, such as NetVLAD [6], and another for image matching, such as SIFT [8] or SuperPoint [2]. However, such a method is not competitive as far as efficiency is concerned due to the absence of shared computing. In addition, it is difficult to trade-off efficiency and accuracy in model selection. Specifically, accurate image retrieval can improve the accuracy and efficiency of localization within a certain range. However, advanced image retrieval models tend to be heavy to pursue higher accuracy, which causes an increase in computational cost. Still, these improvements do not result in corresponding gains for localization.

A recent study unifies global and local features for visual localization using multitask distillation, i.e., HF-Net [19]. It uses a CNN that jointly estimates local and global features with a shared encoder. Although it is improved in terms of computational efficiency by using a compression model, i.e., MobileNetVLAD (MNV). However, the accuracy decrease after distillation, especially in challenging cases, especially at night scenes. One possible reason is that they did not consider the conflict between the two tasks, i.e., image retrieval and image-matching, at the feature level; see Figure 4.1. Specifically, learning global features for image retrieval tends to induce features in the backbone to be more semantic and less localizable, eventually making it more sparse. Conversely, learning local features for image-matching emphasizes feature invariance, which leads the features to focus more on local information and lacks global semantics. This issue has been demonstrated earlier [174] but needs to be handled more appropriately.

Most recently, transformers have been used for feature extraction in computer vision tasks and have led to state-of-the-art results [175, 177]. It benefits from the desirable property of the self-attention mechanism, which can break through spatial constraints to establish global correlations, and aggregate task-relevant features naturally.



(a) (b)

Figure 4.1: Illustration of the feature-level gap between the two tasks. i.e., (a) image retrieval and (b) image matching. Where (a) and (b) are activation maps of local features generated by NetVLAD [6] and SuperPoint [2], respectively.

In this paper, we propose a novel holistic model, SuperGF, which unifies global and local features for visual localization. It works directly on the local features generated by the image-matching model and aggregates to a global image feature, similar to BoW [178,179] or Fisher Vector [180]. A transformer is adopted to perform feature aggregation, which is more accurate and resource-friendly. It integrates global contextual information and establishes global correlations between feature tokens by the self-attention mechanism; thus, semantic cues can be learned automatically from local features of the image-matching model. As a result, the transformer module can bridge the feature-level gap between the two tasks and yield robust global features for image retrieval in an efficient manner. We experimentally evaluate global features yield by SuperGF on several benchmarks. Then we also assess the performance of visual localization using the holistic model combining different kinds of local features. The results show the advantage of our model compared to existing methods.

4.2 Related Work

4.2.1 Approaches for Visual Localization

The approaches of previous works on visual localization can be summarized as follow:

Structure-based Localization. Previous visual localization approaches mainly rely on estimating correspondences between 2D keypoints in the query and 3D points in a sparse model using local descriptors. The map is usually composed of a 3D point cloud con-

structed via Structure-from-Motion (SfM), where each 3D point is associated with one or more local feature descriptors. The query pose is obtained by feature matching and solving a Perspective-n-Point problem (PnP) [80]. However, direct matching methods tend to be resource-intensive or fragile and challenging to apply in large-scale localization.

Image-based Localization. Visual localization in large-scale urban environments is often approached as an image retrieval problem. Specifically, the location of a given query image is predicted by transferring the geotag of the most similar image retrieved from a geotagged database [6, 53–58]. This approach scales to entire cities thanks to compact image descriptors and efficient indexing techniques [1, 3, 26, 27] and can be further improved by spatial re-ranking [174], informative feature selection [60, 67] or feature weighting [56, 58, 68, 69]. Image-based localization approaches have recently shown promising results in terms of robustness and efficiency but are not competitive in terms of accuracy [6, 181], which output only an approximate location of the query, not an exact 6-DoF pose.

Hierarchical Localization. Hierarchical localization takes an approach, dividing the problem into a global, coarse search followed by a fine pose estimation. It shows advantages in terms of efficiency and accuracy compared to the above two approaches, which can be applied to large-scale. The intermediate retrieval step of hierarchical localization limits the downstream feature matching to a reasonable range, which reduces the computational cost significantly while improving the localization performance by reducing the influence of feature repetition. [18] proposed to search at the map level using image retrieval and localize by matching hand-crafted local features against retrieved 3D points. However, its robustness and efficiency are limited by the underlying local descriptors and heterogeneous structure. Taira *et al.*applied learning-based features to camera pose estimation but in a dense, expensive manner [17]. Recently, HF-Net [19] integrated learning-based models of image retrieval and image-matching, by model distillation that simultaneously predicts keypoints as well as global and local descriptors for accurate 6-DoF localization.

4.2.2 Global and Local Image Features

Before the emergence of deep learning, hand-crafted local features, such as SIFT [8], ORB [24], and SURF [23], are widely applied in computer vision fields such as image

matching. Moreover, traditional aggregation methods [179, 180] are developed for generating global image features for image retrieval using these hand-crafted local features. However, hand-crafted local features are limited in invariance due to only involving low-level information.

Recent features emerging from convolutional neural networks (CNN) exhibit unrivaled robustness at a low computing cost. However, it tends to be task-specific. Specifically, task-specific local or global image features are generated using different models in an end-to-end manner. Even though they achieve superior performances in their respective domain, such as image retrieval [174, 175, 177, 182] or image matching [2, 3, 25–27], there are still problems in unifying for multi-task.

More recently, transformers have been adopted for feature extraction in computer vision fields and achieved state-of-art performances [175, 177]. It benefits from the desirable property of the self-attention mechanism, which can naturally aggregate task-relevant features. Recent studies have applied transformers to each component method for visual localization, i.e., image retrieval [175, 177, 183] and image matching [3, 31].

4.3 Unify Local and Global features for Visual Localization.

4.3.1 Overview

SuperGF essentially plays the role of feature aggregation, i.e., aggregate local image features used for image-matching, which are more localizable but lack semantics, into glocal image features with rich semantic information used for image retrieval. Figure. 4.2 illustrates the framework of SuperGF.

Given an input image, local features are extracted by a detector and descriptor at first. In practice, SuperGF is designed for both a hand-crafted descriptor and a learning-based descriptor, i.e., SIFT [8] and SuperPoint [2], respectively. Moreover, we consider two different forms of the descriptors separately, i.e., dense or sparse. Specifically, in the case of the sparse version, only local descriptors where keypoints are taken into count. In the case of the dense version, we adopt dense descriptors, i.e., dense SIFT or feature map output by the encoder of SuperPoint.

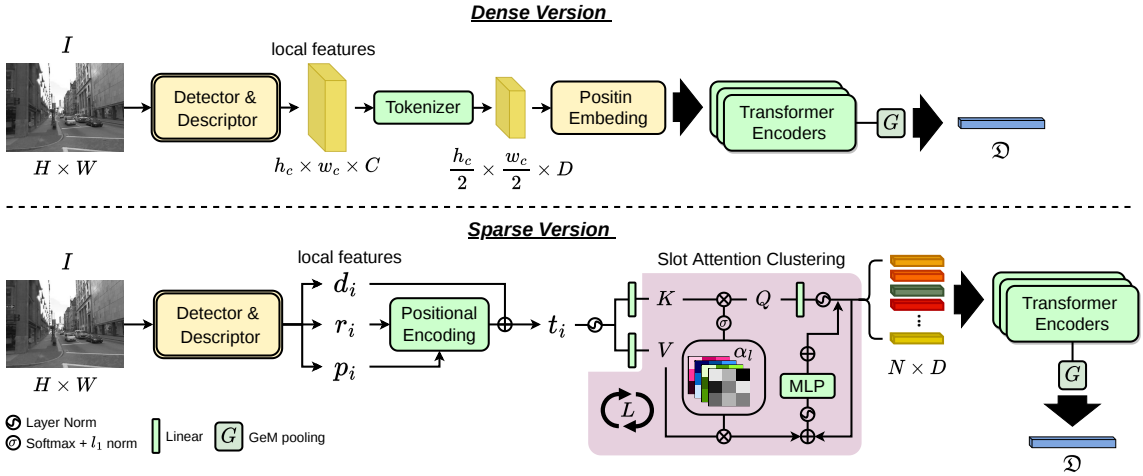


Figure 4.2: The framework of SuperGF. We provide two implementations of SuperGF, i.e., the dense version and the sparse version. The former works on dense local descriptors, such as dense SIFT or feature maps output by the SuperPoint encoder. The latter works on sparse local features used for image matching, i.e., keypoints (p_i), descriptors (d_i), and confidence scores (r_i). The modules indicated by green baskets contain learnable parameters.

Then, we process the input of local features into tokens. In the case of the dense version, we perform the local descriptors by a tokenizer consisting of a one-layer CNN with downsampling operations. Then, we perform position embedding for these tokens. In the case of the sparse version, the input of local features is a variable-length sequence. To save computational costs, we downsample them by performing clustering. Before that, we integrate three features of descriptors, keypoints, and confidence scores by performing positional encoding.

Finally, these tokens generated by the previous step are input into the three layers of the vision transformer encoder. And a global image feature is aggregated from the transformer's output using a GeM pooling layer.

4.3.2 Local Feature Processing

Dense version. Given an input image $I \in \mathbb{R}^{H \times W}$, the feature map of local descriptors of I can be denoted as $\mathbf{F} \in \mathbb{R}^{128 \times \frac{H}{8} \times \frac{W}{8}}$. Let's define $\mathcal{F} : \xi \rightarrow \mathbb{R}^{D \times \frac{H}{16} \times \frac{W}{16}}$ to denote a CNN layer encodes feature map $\mathbf{F} \in \xi$ into a feature with the shape of $(D, \frac{H}{16}, \frac{W}{16})$, then, the output of

the tokenizer can be denoted as:

$$\boldsymbol{T}_{raw} = ReLU(\mathcal{F}(\mathbf{F})) + \boldsymbol{\mathcal{P}} \tag{4.1}$$

where $\boldsymbol{T}_{raw}, \boldsymbol{\mathcal{P}} \in \mathbb{R}^{D \times \frac{H}{16} \times \frac{W}{16}}$, $\boldsymbol{\mathcal{P}}$ is the tokens of *Position Embedding*, D means the number of channels of raw patch tokens, specifically, $D = 512$. Finally, the raw patch tokens $\boldsymbol{T}_{raw}$ is reshaped into a sequence of flattened 2D pattern $(\frac{H}{16} \times \frac{W}{16}, D)$ as input tokens for transformer encoder layers. In practice, 7×7 convolutional kernel is used, and the stride is 2.

Sparse version. Given an input image $I \in \mathbb{R}^{H \times W}$, the sparse local features of I can be denoted as $d_i \in \mathbb{R}^{i \times d}$: local descriptors, $p_i \in \mathbb{R}^{i \times 2}$: keypoints position, and $r_i \in \mathbb{R}^{i \times 1}$: confidence scores. $d = 128$ when using SIFT, and $d = 256$ when using SuperPoint.

The initial representation $t_i \in \mathbb{R}^{i \times d}$ for each keypoint i combines its visual appearance and location. We embed the keypoint position into a high-dimensional vector with a Multilayer Perceptron (MLP) as:

$$t_i = d_i + MLP_{enc}(p_i, r_i) \tag{4.2}$$

This encoder enables the graph network to later reason about both appearance and position jointly, especially when combined with attention, called *Positional Encoding*.

Inspired by recent studies [183, 184], we propose a clustering module base on slot attention, which takes an input of t_i and outputs an ordered set of features of clusters, denoted by $\boldsymbol{\zeta}$. Let the clustering module be represented by function $\Phi(\bullet) : \mathbb{R}^{i \times d} \rightarrow \mathbb{R}^{N \times D}$, which can be defined as an *iterative* module:

$$\Phi(\bullet) = \boldsymbol{\zeta}^L, \quad \boldsymbol{\zeta}^l = \phi(\bullet; \boldsymbol{\zeta}^{l-1}) \tag{4.3}$$

where ϕ denotes the core function of the module applied L times, and $\boldsymbol{\zeta}^0 \in \mathbb{R}^{N \times D}$ denotes a set of learnable templates of clusters, which are initialized randomly. In practice, we set $N = 512$ and $L = 6$. The final output is progressively formed by iterative refinement of the templates and each token of $\boldsymbol{\zeta}$ is a function of all input of t_i.

The architecture of the core function ϕ is inspired by [89,185] and is composed of a dot-

product attention function ψ, followed by an MLP. The function ψ receives three inputs, i.e., *key*, *value*, and *query*, represented by K, Q, and V, respectively. The corresponding input generates the K, Q, and V, passing through layer normalization and fed to three linear projection functions that project them to dimensions d_k, d_q, and d_v, respectively. In practice, we set $d_k = d_q = d_v = D = 512$, K and V are generated by t_i, and Q is generated by $\boldsymbol{\zeta}$. The functions ϕ and ψ is given by:

$$\phi(\bullet) = \psi(\bullet) + MLP(\psi(\bullet)), \quad \psi(\bullet) = \boldsymbol{\alpha} \cdot V + \boldsymbol{\zeta}^l \tag{4.4}$$

where $\boldsymbol{\alpha}$ denotes the attention maps over t_i. Thus, we have N attention maps in total. i.e., $\boldsymbol{\alpha} \in \mathbb{R}^{i \times N}$ $\boldsymbol{\alpha}$ can be calculated as follow:

$$\boldsymbol{\alpha} = \mathcal{N}_{L_1}(\text{softmax}(\frac{QK^\top}{\sqrt{d_k}})) \tag{4.5}$$

where $\mathcal{N}_{L_1}$ denotes L_1 normalization.

4.3.3 Aggregate to Global Image Features

SuperGF employs vision transformer encoders to achieve feature aggregation, integrating global contextual information and establishing global correlations between feature tokens by self-attention mechanism, and semantic cues can be learned automatically from the input of local features. In practice, we follow the implementation of the vision transformer encoder, which consists of a stack of Multi-headed Self-Attention (MSA) and Multi-Layer Perceptron (MLP) modules [89,185], and add skip-connection between two transformer encoders. Moreover, we discarded the operation of *patch embedding* and instead it by the operation of the previous step; see Sec. 4.3.2. Let's define the input as $\mathbf{T}_0$, where $\mathbf{T}_0$ denotes the output of the previous step, i.e., $\mathbf{T}_{raw}$ or $\boldsymbol{\zeta}^L$. Then define function $\boldsymbol{\Theta}_n(\bullet) : n \in [1, 4]$ to denote the n-layer of vision transformer encoders. The output tokens $\mathbf{T}_{out}$ is given by:

$$\begin{aligned} \boldsymbol{T}_n &= \sum_{n=1}^{3} \left[\boldsymbol{T}_{n-1} + \boldsymbol{\Theta}_n(\boldsymbol{T}_{n-1})\right], \\ \mathbf{T}_{out} &= MLP(\mathcal{L}(\boldsymbol{T}_3)) \end{aligned} \tag{4.6}$$

where $\mathcal{L}$ denotes the layer normalization, In practice, we double the dimension of the input in the MLP. Thus, $\mathbf{T}_{out} \in \mathbb{R}^{(\frac{H}{16} \times \frac{W}{16}) \times 1024}$ in the case of the dense version or $\mathbf{T}_{out} \in \mathbb{R}^{N \times 1024}$ in the case of the sparse version. As a result, $\mathbf{T}_{out}$ tends to focus on task-relevant local features.

Finally, $\mathbf{T}_{out}$ is aggregated to global image features by performing Generalized Mean (GeM) pooling [], which is a flexible way to aggregate local features with a learnable parameter p. The aggregated global image feature $\mathfrak{D}$ is given by:

$$\begin{aligned} \mathfrak{D} &= \mathcal{N}_{L_2}(GeM(\mathbf{T}_{out}, p)), \\ GeM(\mathbf{T}_{out}, p) &= \left[\frac{1}{\lambda} \sum_{i=0}^{\lambda} (T_{out}^{i})^{p} \right]^{\frac{1}{p}} \end{aligned} \tag{4.7}$$

where $\mathfrak{D} \in \mathbb{R}^{1 \times 1024}$, $\mathcal{N}_{L_2}$ denotes L_2 normalization, and λ denotes the special size of $\mathbf{T}_{out}$, i.e., $\lambda = \frac{H}{16} \times \frac{W}{16}$ in the case of the dense version and $\lambda = N$ in the case of the sparse version. In practice, we set $p = 3$ in parameter initialization.

4.3.4 Training Strategy

Loss Function

In most studies, image retrieval or Visual Place Recognition (VPR) is treated as a binary problem. Models are trained using pairs or triplets of image samples labeled as either positive or negative with a metric learning loss, such as the triplet loss [186, 187] or a contrastive loss [188]. However as mentioned in existing studies [181, 189], since the distribution of scenes in the real world is continuous, in the VPR task, the similarity between images in VPR cannot be strictly defined in a binary fashion. Thus as we observed, in most VPR datasets, there is a large ambiguity between the positive and negative samples of the training data.

Moreover, VPR is essentially a problem of ranking, not classification. For training a VPR model, metric learning losses distinguish between positive and negative samples by defining a margin, which easily leads to a distance ambiguity. This results in the performance being very sensitive to the loss settings so it is often difficult to achieve optimal performance by optimizing a metric learning loss directly.

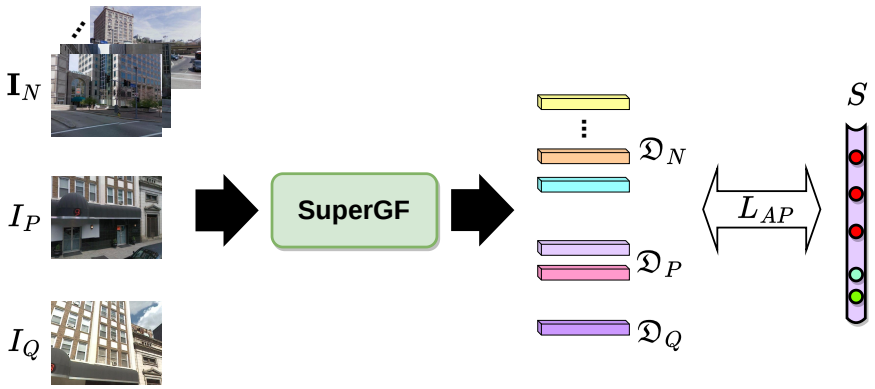


Figure 4.3: Illustration of the training strategy. Each sample of training data includes a batch composed of one query image I_Q, one positive sample I_P, and $\alpha + \beta$ negatives; of these, α soft negative samples and β negative samples. i.e., $I_N^i \in \boldsymbol{I}_N : i = [1, ..., \alpha + \beta]$. In practice, $\alpha = 2$ and $\beta = 100$.

Considering the above, we adopt a listwise loss. i.e., Average Precision (AP) loss [27, 189, 190], to train our model. The AP loss train models by optimizing ranking results directly. We follow the implementation of AP loss as Chen *et al.* [189]. As illustrated in Fig. 4.3, after inference an image batch of $[I_Q, I_P, \boldsymbol{I}_N]$, we obtain global image descriptors $[\boldsymbol{\mathfrak{D}}_Q, \boldsymbol{\mathfrak{D}}_P, \boldsymbol{\mathfrak{D}}_N]$. Then we calculate cosine similarities $\boldsymbol{\mathcal{S}} = \mathcal{S}_i : i \in [1, \alpha+\beta]$ between $\boldsymbol{\mathfrak{D}}_Q$ and $\boldsymbol{\mathfrak{D}}_P \cup \boldsymbol{\mathfrak{D}}_N$:

$$\mathcal{S}_i = sim(\boldsymbol{\mathfrak{D}}_Q, \boldsymbol{\mathfrak{D}}^i) = \boldsymbol{\mathfrak{D}}_Q^\top \boldsymbol{\mathfrak{D}}^i \tag{4.8}$$

where $\boldsymbol{\mathfrak{D}}^i \in \boldsymbol{\mathfrak{D}}_P \cup \boldsymbol{\mathfrak{D}}_N$, $\mathcal{S}_i \in [-1, 1]$. Let's define the function $AP(\cdot)$ denotes the ranking metric AP for $\boldsymbol{\mathfrak{D}}_Q$, The final loss function can be defined as:

$$L_{AP} = 1 - AP(\boldsymbol{\mathcal{S}}, \boldsymbol{S}) \tag{4.9}$$

where $\boldsymbol{S} \in [0, 1]$ denotes the label of image similarity scores between I_Q and $I_P \cup \boldsymbol{I}_N$.

In the case of the sparse version, we further adopt an attention decorrelation loss which aims at reducing the spatial correlation between attention maps. It makes the output of the slot attention clustering module, i.e., $\boldsymbol{\zeta}$, as complementary as possible. Specifically, we encourage them to attend to different local features, i.e., different locations of the image.

Let's define $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]$, The attention decorrelation loss is given by:

$$L_{attn} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{\alpha_i \cdot \alpha_j}{\|\alpha_i\|_2 \|\alpha_j\|_2} \tag{4.10}$$

where $i, j \in \{1, ..., N\}$. In practice, We train our sparse version model with the above two losses jointly.

Training Setting

First, we aim to design robust global image descriptors that can be used for place recognition for SuperPiont dependencies. So we fixed the SuperPoint encoder during the training process. We perform image-level supervision using the AP loss. In practice, we adopt the data annotation provided by Vallina *et al.*. [181] as the soft label, i.e, $\boldsymbol{S}$ for our model training, which represents Field-of-View (FoV) overlap between query and database images.

Specifically, for each query image, we split the database images into three kinds based on FoV overlap, i.e., positive samples where $S \in [0.5, 1]$, soft negative samples where $S \in (0, 0.5)$, and negative samples where $S = 0$. For each query image I_Q, we compute the AP loss between it with 105 images in the database, which contains 1 positive sample, 2 soft negative samples and 100 negative samples. For each iteration of the training process, we load 4 batches to compute loss and optimize the parameters of our model. We observe that choosing more negative samples for each query image tends to result in better performance. However, selecting too many negative samples takes up more computational resources and provides limited gain to the model.

4.4 Experiments

We experimentally evaluate the proposed SuperGF on several benchmark datasets compared with some state-of-the-art methods. We evaluate our method on two tasks, i.e., VPR and visual localization. The former aims mainly to examine the performance of global image features generated by SuperGF. We give the details of experimental settings, datasets, evaluation metrics, and compared methods in the following.

4.4.1 Implementation details

Model Settings. We set the latent embedding dimension of the transformer encoders as 512, and the hidden dimension of MLP blocks as 1024. We trained our model on the MSLS training set with settings reported in Sec. 4.3.4. We selected 10,000 query samples randomly from all sub-cities for one epoch and trained our model on 350 epochs in total. We adopted the AdamW optimizer and set the parameter of weight_decay $= 10^{-4}$ in training. We set an initial learning rate to 10^{-4} and it will finally decline to 10^{-6}. We set the input image size to 640×480.

Experiment Settings for VPR. We perform the retrieval based on the L_2 distance between global image features to obtain the prior candidates. We further perform geometry verification by matching sparse descriptors provided by the front-end model, i.e., SIFT or SuperPoint. In practice, given an image pair of the query and database, we obtain keypoints and sparse local descriptors for both. Then, we perform feature matching to obtain initial point correspondences, using Nearest Neighbor searching (NN) or SuperGlue [3]. Finally, we estimate the homography with RANSAC as a robust estimator and re-rank the prior candidates based on the number of inliers. We keep the top 100 candidates retrieved by global image features and perform the geometry verification.

Experiment Settings for Localization. Firstly, a sparse reference model is built with SfM using database images. Specifically, we adopt prebuilt reference models with COLMAP using RootSIFT or SuperPoint, provided by [4] and [19]. They are each used in the case of the corresponding descriptor employed in SuperGF to avoid the influence of using different detectors on feature matching. For localization, given a query image, we retrieve the top-n candidates from the database using global image features generated by SuperGF and match them one by one. In practice, we adopt NN and also SuperGlue when SuperPoint is selected. Then, we keep all matches where a corresponding 3D point is included in the reference SfM model. Finally, we can lift them to 2D-3D matches, and the 6-DoF camera pose of the query image is obtained by solving PnP with RANSAC.

4.4.2 Evaluation Datasets

We evaluated our method on several public benchmark datasets: MSLS [191], Pitts30k [58], Nordland [192] and Tokyo247 [57] for VPR; The RobotCar Seasons dataset [4] for localization. All of these datasets contain different challenging appearance variations, such as changes in day-night, weather, and season. More details of dataset usage are given in Supplementary Material. All images are resized to 640×480 while evaluation.

4.4.3 Metrics

As illustrated in Sec. 4.4.2, for VPR datasets, we use Recall@N metric, which computes the percentage of query images that are correctly localized. A query image is considered retrieved successfully if at least one of the top N ranked reference images is within a threshold distance from the ground truth location of the query image. Default threshold definitions are used for all datasets [6, 177, 182].

For the localization dataset, i.e., the RobotCar Seasons dataset, we report the pose recall at position and orientation thresholds different for each sequence, as defined by the benchmark [4].

4.4.4 Compared Methods

We experimentally compared SuperGF against several models on both VPR and visual localization tasks with settings introduced in Sec. 4.4.1.

VPR methods. We compare methods in the VPR task considering both two approaches, i.e., single-pass and two-stage retrieval. The former only uses global image features, and the latter performs re-ranking with geometry verification additionally. For the experiment of single-pass retrieval, we compare with a widely used baseline, i.e., NetVLAD [6], several state-of-the-art CNN-based methods, including SFRS [176], ResNet50-GeM-GCL [181], GeM-AP [189], and also two state-of-the-art transformer-based methods, i.e., SOLAR [175] and TransVPR [177]. For the experiment of two-stage retrieval, we compare with DELG [174], and two state-of-the-art methods, i.e., Patch-NetVLAD [182] and TransVPR. As mentioned earlier in the paper, local features generated by all the above methods don't support 6-DoF camera pose estimation. In addition, we further compare with a strong hybrid baseline,

NV-SP-SG, which re-ranks NetVLAD retrieved candidates by using SuperGlue matcher to match SuperPoint local features.

Localization methods. Since image-based localization approaches lack accuracy, we only consider localization approaches that can obtain 6-DoF camera poses; see Sec. 4.2.1. In practice, we compare ours with two methods based on structure-based localization approaches, i.e., Active Search (AS) [49] and City Scale Localization (CSL) [47], hierarchical localization approach based on hybrid pipelines including NV+SIFT, NV+SP, and also HF-Net [19]. NV+SIFT and NV+SP mean using global features extracted by NetVLAD and local features extracted by SIFT or SuperPoint.

4.5 Results and Discussion

4.5.1 Single-pass Retrieval

Table 4.1 shows the experimental results for single-pass retrieval. We can make the following observations.

First, the performance of single-pass retrieval only based on global image features is highly relevant to the dataset, which denotes global features as a compact representation of the entire image, which exhibits undesirable generalizability, especially for high-precision retrievals, such as R@1 and R@5.

Next, in general, even if we do not use separate generated task-specific local features as other methods do, i.e., we adopt local features designed for image-matching. Our method based on learned local features, such as SuperGF-denseSP or SuperGF-sparseSP, can still generate global image features on par with state-of-the-art retrieval models. It demonstrates the advantages of the transformer in feature aggregation.

Finally, from the comparison between different implementations of our method, we can observe that using learned local features for the aggregation, i.e., SuperGF-denseSP and SuperGF-sparseSP, show significantly better results than using hand-craft local features, i.e., SuperGF-denseSIFT and SuperGF-sparseSIFT. In addition, using dense local features show better results than using sparse local features, especially for hand-craft local features.

Table 4.1: Single-pass retrieval results with three metrics of recall, where 'R-' means 'Recall'.

	MSLS val			MSLS challenge			Pitts30k test			Nordland test			Tokyo247 test		
Method	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD	47.7	62.8	70.9	30.7	41.9	46.4	82.1	91.4	93.8	12.5	21.4	26.1	57.1	72.1	77.1
SFRS	69.2	80.3	83.1	41.5	52.0	56.3	**89.4**	**94.7**	**95.9**	18.8	32.8	39.8	**85.4**	**91.1**	**93.3**
ResNet50-GeM-GCL	66.2	78.9	81.9	43.3	59.1	65.0	72.3	87.2	91.3	27.2	41.1	49.2	44.1	61.0	66.7
GeM-AP	64.1	75.0	78.2	33.7	44.5	49.4	80.7	91.4	94.0	11.8	18.4	22.7	11.4	22.9	30.5
SOLAR	**78.3**	**87.2**	**89.6**	45.2	60.5	69.3	85.4	92.6	94.8	36.6	51.3	58.8	76.2	84.4	88.3
TranVPR	70.8	85.1	**89.6**	48.0	**67.1**	**73.6**	73.8	88.1	91.9	15.9	38.6	49.4	–	–	–
SuperGF-DenseSP	72.8	81.5	85.2	**50.8**	66.0	71.6	68.4	82.5	87.3	**58.4**	**78.6**	**85.5**	23.2	37.8	46.0
SuperGF-SparseSP	71.3	81.3	85.0	48.2	65.3	70.4	60.2	77.6	83.1	57.1	75.4	82.0	25.7	39.4	44.4
SuperGF-DenseSIFT	60.2	72.8	75.3	39.7	56.2	63.9	53.2	71.6	78.1	35.6	50.7	56.7	14.7	26.8	36.3
SuperGF-SparseSIFT	49.7	67.5	74.4	11.2	22.7	37.5	37.7	59.1	69.2	15.5	30.5	38.4	8.6	15.9	21.0

4.5.2 Two-stage Retrieval

Table 4.2 shows the experimental results for two-stage retrieval. We can make the following observations.

First, compared to single-pass retrieval, the robustness is significantly improved by performing re-ranking based on geometry verification, which illustrates the importance of spatial information for image retrieval, which is discarded by global image features.

Next, using sparse local features designed for image-matching for the re-ranking show on par even better performance than using those task-specific local features. It is noteworthy that the former performs matching sparse local features, which is more competitive in terms of efficiency than the latter, which performs matching local features densely.

Finally, our methods with the SuperGlue matcher, i.e., SuperGF-denseSP-SG and SuperGF-sparseSP-SG, show state-of-the-art level performance compared to others.

4.5.3 Visual Localization

We report the pose recall at position and orientation thresholds different for each sequence, as defined by the benchmark [4]. Table 4.3 shows the localization results for the different methods. We can make the following observations.

First, compared to methods that intermediate a retrieval step, structure-based methods, i.e., AS and CSL, show competitive results on the dusk sequence, where the accuracy tends to saturate. In the more challenging sequences, methods based on a hierarchical

Table 4.2: Two-stage retrieval results with three metrics of recall, where 'R-' means 'Recall'.

	MSLS val			MSLS challenge			Pitts30k test			Nordland test			Tokyo247 test		
Method	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
DELG	83.2	90.0	91.1	52.2	61.9	65.4	**89.9**	**95.4**	**96.7**	51.3	66.8	69.8	78.8	86.7	90.0
NV-SP-SG	78.1	81.9	84.3	50.6	56.9	58.3	87.2	94.8	96.4	29.1	33.5	34.3	71.2	82.0	78.8
Patch-NetVLAD	79.5	86.2	87.7	48.1	57.6	60.5	88.7	94.5	95.9	46.4	58.0	60.4	**95.9**	**96.8**	**97.1**
TransVPR	86.8	91.2	92.4	63.9	74.0	77.5	89.0	94.9	96.2	58.8	75.0	78.7	–	–	–
SuperGF-DenseSP	**87.7**	**92.0**	**93.2**	**65.2**	**76.5**	**80.1**	84.6	92.1	94.0	**87.4**	**95.3**	**97.2**	68.9	72.4	73.7
SuperGF-SparseSP	86.4	91.5	92.9	64.4	74.3	79.1	83.3	90.8	93.8	85.3	94.7	97.1	63.2	70.6	72.8
SuperGF-DenseSIFT	71.3	81.3	85.0	50.2	57.1	59.3	60.2	77.6	83.1	57.1	75.4	82.0	25.7	39.4	44.4
SuperGF-SparseSIFT	49.7	67.5	74.4	45.5	50.2	59.8	55.2	67.1	78.9	46.8	54.1	60.2	25.2	37.9	43.1

approach tend to work significantly better than structure-based methods, which suffer from the increased ambiguity of the matches.

Next, in contrasting the two pipelines of NV+SIFT and NV+SP, we can observe that NV+SIFT shows slightly better results than NV+SP on easy cases, such as dusk. In contrast, the NV+SP shows significantly better results than NV+SIFT in those challenging cases, especially at night. It indicates that the SIFT descriptors perform better than SuperPoint on camera pose estimation in normal cases, but the conclusion is the opposite in the situation of challenging cases. It is consistent with the findings of previous studies [7, 102].

Finally, by comparing our method with baselines, we can observe that although the two-stage localization fuses image retrieval and camera pose estimation, it is clear that the latter plays a dominant role. Especially for simple cases, the use of SuperGF, which can be regarded as a robust global feature, for retrieval has limited improvement for hierarchical localization. But for challenging cases, we can observe the localization improvement using SuperGF. Overall, by implanting SuperGF, our pipelines achieve state-of-art performance on 6-DoF localization compared with others.

4.5.4 Latency and Memory

In real-world applications, latency and scalability are important factors that need to be considered for real-time visual localization. We report the computational time and memory requirements for compared methods of extracting glocal image features to process a single query image; see Table 4.4. In practice, we unify the input image size as 480×640 for

Table 4.3: Localization results on the RobotCar Seasons dataset. We report the recall [%] at different distance and orientation thresholds, i.e., Δt and Δr, on different conditions. i.e., dusk, sun, night, and night-rain.

		dusk	sun	night	night-rain
	Δt@	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0
Method	Δr@	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AS		44.7 / 74.6 / 95.9	25.0 / 46.5 / 69.1	0.5 / 1.1 / 3.4	1.4 / 3.0 / 5.2
CSL		56.6 / 82.7 / 95.9	28.0 / 47.0 / 70.4	0.2 / 0.9 / 5.3	0.9 / 4.3 / 9.1
NV+SIFT		55.6 / 83.5 / 95.3	46.3 / 67.4 / 90.9	4.1 / 9.1 / 24.4	2.3 / 10.2 / 20.5
NV+SP		54.8 / 83.0 / 96.2	51.7 / 73.9 / 92.4	6.6 / 17.1 / 32.2	5.2 / 17.0 / 26.6
NV+SP+SG		62.4 / 83.5 / 97.2	52.8 / 80.2 / 96.1	29.0 / 66.9 / 90.2	46.4 / 76.1 / 92.0
HF-Net		53.9 / 81.5 / 94.2	48.5 / 69.1 / 85.7	2.7 / 6.6 / 15.8	4.7 / 16.8 / 21.8
SuperGF-SparseSIFT		55.8 / 83.6 / 95.0	45.8 / 67.2 / 91.1	3.6 / 8.8 / 20.8	2.6 / 11.5 / 19.8
SuperGF-DenseSIFT		56.2 / 84.2 / 97.2	48.4 / 68.9 / 91.2	6.9 / 17.9 / 33.1	6.2 / 17.3 / 32.6
SuperGF-SparseSP		55.1 / 83.0 / 96.7	52.2 / 75.7 / 96.1	8.0 / 19.9 / 36.8	8.2 / 22.0 / 33.9
SuperGF-DenseSP		56.6 / 83.5 / 97.0	53.1 / 76.4 / 95.8	12.4 / 26.3 / 45.3	13.0 / 25.2 / 35.8
SuperGF-SparseSP-SG		64.2 / 84.3 / 98.5	54.2 / 81.5 / 97.1	33.6 / 69.9 / 92.6	49.2 / 78.1 / 92.4
SuperGF-DenseSP-SG		**65.1**/**84.5**/**98.5**	**55.8**/**82.5**/**97.5**	**34.2**/**70.3**/**93.0**	**49.9**/**78.2**/**92.8**

all methods. In the case of SuperGF-sparse, we randomly selected 100 images to generate sparse local features for the test. SuperGF shows clear advantages in both terms than others.

Combining the results of previous sections, we can conclude that SuerGF uses minimal resources but generates SOTA-level global image features for image retrieve and lead to better performance of 6-DoF localization. In particular, the sparse version of SuperGF causes a loss of retrieval accuracy to some extent, but it has obvious advantages in terms of feature extraction efficiency. It is worth noting that SuperGF is affected by the adopted local descriptor significantly. We prefer to use learning-based local features, such as SuperPoint, joined with SuperGF. We tried to use SuperGF to generate global features for hand-craft local features, i.e., SIFT, but the results were unsatisfactory. One possible reason is the limited representative ability of the hand-craft local features, which only jointly low-level information of images.

Table 4.4: Comparisons of model size and latency. Latency is measured on an NVIDIA TITAN RTX GPU.

Method	Memory (MB)	Extraction latency (ms)
NetVLAD	14.8	15.5
SFRS	149.0	16.4
ResNet50-GeM	23.5	11.1
SOLAR	56.2	19.5
SuperGF-dense	6.1	4.4
SuperGF-sparse	2.2	2.3

4.6 Summary and Conclusion

This paper has presented a novel method for global feature extraction, i.e., SuperGF. It is transformer-based and designed for 6-DoF localization, which acts directly on local features provided by descriptors of image-matching. The results show our method's advantages in terms of accuracy and efficiency.

We provide different implementations of SuperGF. i.e., different types of local features with both learning-based and hand-craft descriptors. Users can choose different versions according to their needs. Moreover, we encourage using the learning-based descriptor, i.e., SuperPoint, joint with SuperGF. In addition, if you pursue higher retrieval accuracy while localization, e.g., using for loop closure, then the dense version is more suitable. For 6-DoF localization alone, the sparse version is better in efficiency.

Chapter 5

Conclusions

In this dissertation, we focus on the image-matching problem, aiming to widen the frontier of SfM and visual SLAM applications. We are interested in fully using such high-precision information in RAW images to match extremely low-light scene images that conventional methods cannot handle. In Chapter 2, we presented a dataset created for evaluating image matching methods for low-light scene images. Then, we have reported the experiments we conducted to test multiple combinations of existing image-enhancing methods and image-matching methods. The results show that The direct use of the RAW-format images shows a clear advantage over the standard RIP. Using the standard RIP yields only suboptimal performance, as it cannot utilize information stored in the lower bits of RAW-format signals. Moreover, when using the standard RIP, using classical histogram equalization or the state-of-the-art CNN-based image-enhancing method does not make a big difference, as reported in [128]. We further conclude that SuperPoint [2] and its variants work consistently better than RootSIFT on such challenging conditions and SID [5] is the best image enhancer when using SuperPoint+SuperGlue [3]. Otherwise, BM3D [104] and SID perform equally well and better than histogram equalization.

Next, we consider better utilizing the information in RAW images under low-light scenes. The current use of RAW images stops at the image enhancement stage, i.e., it yields a high visual quality image but not for obtaining optimal performance for downstream tasks. We aim to utilize the information stored in RAW images of low-light scenes to yield better performance for downstream tasks of 3D reconstruction in Chapter 3. To promote further study, we introduce a novel dataset that includes multi-view RAW images under

extremely low-light scenes, with corresponding long-exposure versions and ground-truth depth. Moreover, we propose an image enhancement method, SuperISP, which outperforms the other enhancers under two downstream tasks, i.e., image matching and monocular depth estimation. We evaluate multiple combinations of image enhancement methods and task-specific methods. The experimental results show that compared with other image enhancers, SuperISP obtains the best performances on two downstream tasks, which indicates that SuperISP exploits the information in low-light images and provides effective information for downstream tasks.

Finally, for better visual localization, we presented a novel method for global feature extraction in Chapter 3, i.e., SuperGF. It is transformer-based and designed for 6-DoF localization, which acts directly on local features provided by descriptors of image-matching. The results show our method's advantages in terms of accuracy and efficiency. We provide different implantations of SuperGF. i.e., different types of local features with both learning-based and hand-craft descriptors. Users can choose different versions according to their needs. Moreover, we encourage using the learning-based descriptor, i.e., SuperPoint, joint with SuperGF. In addition, if you pursue higher retrieval accuracy while localization, e.g., using for loop closure, then the dense version is more suitable. For 6-DoF localization alone, the sparse version is better in efficiency.

In addition, there are remaining problems in the study of this field. Although we provide two datasets, as mentioned. The current scale still cannot meet the needs of practical applications in the real world. Even so, we provide reliable benchmarks for the research of 3D reconstruction tasks under low-light scenes. Even deep learning technology has been widely used in this field. Due to various limitations, only a few deep models can be used for real-world applications. Finally, we hope our research will contribute to the community and developments in other related fields.

Bibliography

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops*, pages 337–33712, 2018. `https://github.com/magicleap/SuperPointPretrainedNetwork`.

[3] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 4937–4946, 2020. `https://github.com/magicleap/SuperGluePretrainedNetwork`.

[4] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.

[5] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018. `https://github.com/cchen156/Learning-to-See-in-the-Dark`.

[6] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[7] Wenzheng Song, Masanori Suganuma, Xing Liu, Noriyuki Shimobayashi, Daisuke Maruta, and Takayuki Okatani. Matching in the dark: A dataset for matching image pairs of low-light scenes. In *Proceedings of International Conference on Computer Vision*, pages 6029–6038, 2021.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Conflict and Violence*, 60(2):91–110, 2004.

[9] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of European Conference on Computer Vision*, pages 3–20. Springer, 2016.

[10] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[11] Heinly Jared, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3287–3295, 2015.

[12] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[13] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018.

[14] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Proceedings of European Conference on Computer Vision*, pages 752–765. Springer, 2012.

[15] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al.

Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018.

[16] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *The International Journal of Robotics Research*, 39(9):1061–1084, 2020.

[17] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018.

[18] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, pages 456–465. PMLR, 2018.

[19] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[20] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[21] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*, 2017.

[22] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops*, pages 337–33712, 2018.

[23] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of European Conference on Computer Vision*, pages 346–359, 2006.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of International Conference on Computer Vision*, pages 2564–2571, 2011.

[25] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-net: learning local features from images. In *Proceedings of Conference on Neural Information Processing Systems*, pages 6237–6247, 2018.

[26] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 8084–8093, 2019.

[27] R. Jerome, W. Philippe, R. S. César, and H. Martin. R2D2: repeatable and reliable detector and descriptor. In *Proceedings of Conference on Neural Information Processing Systems*, pages 12405–12415, 2019. `https://github.com/naver/r2d2`.

[28] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *Proceedings of International Conference on Computer Vision Theory and Applications*, 2(331-340):2, 2009.

[29] E. Brachmann and C. Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *Proceedings of International Conference on Computer Vision*, pages 4321–4330, 2019. `https://github.com/vislearn/ngransac`.

[30] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to find good correspondences. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018.

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021.

[32] H. Malm, M. Oskarsson, E. Warrant, P. Clarberg, J. Hasselgren, and C. Lejdfors. Adaptive enhancement and noise reduction in very low light-level video. In *Proceedings of International Conference on Computer Vision*, pages 1–8, 2007.

[33] X. Dong, G. Wang, Y. Pang, W. Li, J. Wen, W. Meng, and Y. Lu. Fast efficient algorithm for enhancement of low lighting video. In *Proceedings of International Conference on Complex Medical Engineering*, pages 1–6, 2011.

[34] A. Łoza, D. R. Bull, P. R. Hill, and A. M. Achim. Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients. *Digital Signal Processing*, 23(6):1856–1866, 2013.

[35] S. Park, S. Yu, B. Moon, S. Ko, and J. Paik. Low-light image enhancement using variational optimization-based retinex model. *IEEE Transactions on Consumer Electronics*, 63(2):178–184, 2017.

[36] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2016.

[37] R Priyadarshini, Arvind Bharani, E Rahimankhan, and N Rajendran. Low-light image enhancement using deep convolutional network. In *Innovative Data Communication Technologies and Application*, pages 695–705. Springer, 2021.

[38] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3063–3072, 2020.

[39] Li-Wen Wang, Zhi-Song Liu, Wan-Chi Siu, and Daniel PK Lun. Lightening network for low-light image enhancement. *IEEE Transactions on Image Processing*, 29:7984–7996, 2020.

[40] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129(4):1013–1037, 2021.

[41] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018.

[42] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. Gladnet: Low-light enhancement network with global awareness. In *2018 13th IEEE international con-*

ference on automatic face & gesture recognition (FG 2018), pages 751–755. IEEE, 2018.

[43] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019.

[44] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4):2049–2062, 2018.

[45] C. Chen, Q. Chen, M. N. Do, and V. Koltun. Seeing motion in the dark. In *Proceedings of International Conference on Computer Vision*, pages 3291–3300, 2019.

[46] K. Wei, Y. Fu, J. Yang, and H. Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2755–2764, 2020.

[47] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, 2016.

[48] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision*, pages 383–399, 2018.

[49] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2016.

[50] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, volume 1, page 1, 2015.

[51] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Proceedings of European Conference on Computer Vision*, pages 778–792. Springer, 2010.

[52] Mathias Bürki, Lukas Schaupp, Marcin Dymczyk, Renaud Dubé, Cesar Cadena, Roland Siegwart, and Juan Nieto. Vizard: Reliable visual localization for autonomous vehicles in urban outdoor environments. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1124–1130. IEEE, 2019.

[53] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Proceedings of Asian Conference on Computer Vision*, pages 188–204. Springer, 2014.

[54] David Chen, Sam Tsai, Vijay Chandrasekhar, Gabriel Takacs, Huizhong Chen, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod. Residual enhanced visual vectors for on-device image matching. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 850–854. IEEE, 2011.

[55] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2136–2145, 2017.

[56] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1582–1590, 2016.

[57] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015.

[58] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2013.

[59] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[60] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[61] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of European Conference on Computer Vision*, pages 304–317. Springer, 2008.

[62] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011.

[63] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. Ieee, 2006.

[64] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of International Conference on Computer Vision*, volume 3, pages 1470–1470. IEEE Computer Society, 2003.

[65] Jan C Van Gemert, Cor J Veenman, Arnold WM Smeulders, and Jan-Mark Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2009.

[66] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[67] Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas. Total recall ii: Query expansion revisited. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 889–896. IEEE, 2011.

[68] Petr Gronat, Guillaume Obozinski, Josef Sivic, and Tomas Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 907–914, 2013.

[69] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1169–1176. IEEE, 2009.

[70] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of International Conference on Computer Vision*, pages 1449–1457, 2015.

[71] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *Proceedings of European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[72] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2d and 3d geometric correspondences. In *Proceedings of European Conference on Computer Vision*, pages 803–819, 2018.

[73] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *Proceedings of Conference on Neural Information Processing Systems*, 30:4826–4837, 2017.

[74] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[75] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[76] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[77] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 630–633. IEEE, 2006.

[78] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[79] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[80] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[81] Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2387–2400, 2013.

[82] R Hartley and A Zisserman. Multiple view geometry in computer vision (cambridge university, 2003). *C1 C3*, 2, 2013.

[83] Liang Chen, Charles W Armstrong, and Demetrios D Raftopoulos. An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique. *Journal of biomechanics*, 27(4):493–500, 1994.

[84] Tim D. Barfoot. State estimation for robotics. 2017.

[85] Bert F Green. The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika*, 17(4):429–440, 1952.

[86] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[87] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[88] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of Conference on Neural Information Processing Systems*, 30, 2017.

[90] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006.

[91] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of European Conference on Computer Vision*, pages 850–865. Springer, 2016.

[92] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[93] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[94] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[95] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *Proceedings of European Conference on Computer Vision*, pages 230–247, 2020.

[96] R. Mur-Artal, J. Maria M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[97] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 6565–6574, 2017.

[98] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Conflict and Violence*, 60(1):63–86, 2004.

[99] P. F. Alcantarilla and T. Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2011.

[100] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu. Residual non-local attention networks for image restoration. In *Proceedings of The International Conference on Learning Representations*, 2019.

[101] X. Liu, M. Suganuma, Z. Sun, and T. Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 7000–7009, 2019.

[102] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Conflict and Violence*, page 517–547, 2020.

[103] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[104] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[105] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Learning enriched features for real image restoration and enhancement. *arXiv preprint arXiv:2003.06792*, 2020.

[106] R. Raguram, J. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *Proceedings of European Conference on Computer Vision*, pages 500–513, 2008.

[107] Y. Verdie, K. Yi, P. Fua, and V. Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 5279–5288, 2015.

[108] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3929–3937, 2017.

[109] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of International Conference on Computer Vision*, pages 698–711, 2019.

[110] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of International Conference on Computer Vision*, pages 118–126, 2015.

[111] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[112] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[113] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016.

[114] D. Yoo, S. Park, J. Lee, and I. S. Kweon. Multi-scale pyramid pooling for deep convolutional representation. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops*, pages 71–80, 2015.

[115] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *Proceedings of European Conference on Computer Vision*, pages 467–483, 2016.

[116] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of International Conference on Computer Vision*, pages 3476–3485, 2017.

[117] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[118] H. Aanæs, A. L. Dahl, and K. S. Pedersen. Interesting interest points. *International Journal of Conflict and Violence*, 97(1):18–35, 2012.

[119] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *Proceedings of International Conference on Computer Vision*, pages 359–366, 2011.

[120] M. Pultar, D. Mishkin, and J. Matas. Leveraging outdoor webcams for local descriptor learning. In *Proceedings of Computer Vision Winter Workshop*, pages 6–8, 2019.

[121] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[122] L. Karel, G. Varun, and V. Andrea. Vlbenchmarks, 2011. `http://www.vlfeat.org/benchmarks/`.

[123] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2017.

[124] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *Proceedings of The British Machine Vision Conference*, pages 76.1–76.12, 2012.

[125] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[126] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018.

[127] V. Balntas. Silda: A multi-task dataset for evaluating visual localization, 2018. `https://research.scape.io/silda/`.

[128] T. Jenicek and O. Chum. No fear of the dark: Image retrieval under varying illumination conditions. In *Proceedings of International Conference on Computer Vision*, pages 9695–9703, 2019.

[129] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 6959–6968, 2017.

[130] A. Crivellaro, M. Rad, Y. Verdie, K. Moo Yi, P. Fua, and V. Lepetit. Robust 3d object tracking from monocular images using stable parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1465–1479, 2017.

[131] J. Bian, W. Lin, Y. Matsushita, S. Yeung, T. Nguyen, and M. Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2828–2837, 2017.

[132] J. Bian, Y. Wu, J. Zhao, Y. Liu, L. Zhang, M. Cheng, and I. Reid. An evaluation of feature matchers for fundamental matrix estimation. In *Proceedings of British Machine Vision Conference*, page 25, 2019.

[133] 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 Conference on Computer Vision and Pattern Recognition*, pages 4947–4956, 2020. `https://github.com/aritra0593/Reinforced-Feature-Points`.

[134] Y. Liu, Z. Shen, Z. Lin, S. Peng, H. Bao, and X. Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. In *Proceedings*

of Conference on Neural Information Processing Systems, pages 6990–7001, 2019. https://github.com/zju3dv/GIFT.

[135] X. Shen, C. Wang, X. Li, Z. Yu, J. Li, C. Wen, M. Cheng, and Z. He. Rf-net: An end-to-end image matching network based on receptive field. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 8132–8140, 2019. https://github.com/Xylon-Sean/rfnet.

[136] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 6128–6136, 2017. https://github.com/ubc-vision/image-matching-benchmark.

[137] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. https://github.com/ubc-vision/image-matching-benchmark.

[138] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2011.

[139] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[140] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014.

[141] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *Proceedings of European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[142] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *Proceedings of European Conference on Computer Vision*, pages 130–143. Springer, 2012.

[143] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[144] Reza Abazari, Ali Morsali, and Deepak P Dubal. An advanced composite with ultrafast photocatalytic performance for the degradation of antibiotics by natural sunlight without oxidizing the source over tmu-5@ ni–ti ldh: mechanistic insight and toxicity assessment. *Inorganic Chemistry Frontiers*, 7(12):2287–2304, 2020.

[145] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of International Conference on Computer Vision Workshops*, pages 2109–2118, 2019.

[146] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[147] Yingqian Wang, Longguang Wang, Jungang Yang, Wei An, and Yulan Guo. Flickr1024: A large-scale dataset for stereo image super-resolution. In *Proceedings of International Conference on Computer Vision Workshops*, pages 3852–3857, 2019.

[148] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[149] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[150] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing*, 6(3):451–462, 1997.

[151] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965–976, 1997.

[152] Liang Shen, Zihan Yue, Fan Feng, Quan Chen, Shihao Liu, and Jie Ma. Msr-net: Low-light image enhancement using deep convolutional network. *arXiv preprint arXiv:1711.02488*, 2017.

[153] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[154] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[155] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[156] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 5279–5288, 2015.

[157] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[158] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005.

[159] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[160] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In

Proceedings of 2016 Fourth international conference on 3D vision, pages 239–248. IEEE, 2016.

[161] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Proceedings of Conference on Neural Information Processing Systems*, 27:2366–2374, 2014.

[162] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[163] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 155–163, 2018.

[164] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision*, pages 817–833, 2018.

[165] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[166] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[167] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision*, pages 690–706, 2018.

[168] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of Conference on Computer Vision and Pattern Recognitionn*, pages 1983–1992, 2018.

[169] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv preprint arXiv:1711.03665*, 2017.

[170] Jingchun Zhou, Mingliang Hao, Dehuan Zhang, Peiyu Zou, and Weishi Zhang. Fusion pspnet image segmentation based method for multi-focus image fusion. *IEEE Photonics Journal*, 11(6):1–12, 2019.

[171] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[172] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[173] Zuria Bauer, Zuoyue Li, Sergio Orts-Escolano, Miguel Cazorla, Marc Pollefeys, and Martin R Oswald. Nvs-monodepth: Improving monocular depth prediction with novel view synthesis. In *Proceedings of 2021 International Conference on 3D Vision*, pages 848–858. IEEE, 2021.

[174] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proceedings of European Conference on Computer Vision*, pages 726–743. Springer, 2020.

[175] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. Solar: second-order loss and attention for image retrieval. In *Proceedings of European Conference on Computer Vision*, pages 253–270. Springer, 2020.

[176] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Proceedings of European Conference on Computer Vision*, pages 369–386. Springer, 2020.

[177] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggrega-

tion. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022.

[178] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of European Conference on Computer Vision*, pages 3–20. Springer, 2016.

[179] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of European Conference on Computer Vision*, pages 3–20, 2016.

[180] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[181] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Generalized contrastive optimization of siamese networks for place recognition. *arXiv preprint arXiv:2103.06638*, 2021.

[182] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021.

[183] Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning super-features for image retrieval. *arXiv preprint arXiv:2201.13182*, 2022.

[184] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Proceedings of Conference on Neural Information Processing Systems*, 33:11525–11538, 2020.

[185] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[186] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[187] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision*, pages 459–474, 2018.

[188] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2021.

[189] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of International Conference on Computer Vision*, pages 5107–5116, 2019.

[190] Kean Chen, Weiyao Lin, Jianguo Li, John See, Ji Wang, and Junni Zou. Ap-loss for accurate one-stage object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3782–3798, 2020.

[191] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2020.

[192] Daniel Olid, José M Fácil, and Javier Civera. Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*, 2018.

List of Publications

Research Papers

1. **Wenzheng Song**, Masanori Suganuma, Xing Liu, Noriyuki Shimobayashi, Daisuke Maruta, Takayuki Okatani, "Matching in the Dark: A Dataset for Matching Image Pairs of Low-light Scenes", *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

2. **Wenzheng Song**, Jialun Liu, Xing Liu, Takayuki Okatani, "MID-V2: A Dataset for 3D Reconstruction from Images of Low-light Scenes", *Under submission to the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

3. **Wenzheng Song**, Ran Yan, Boshu Lei, Takayuki Okatani, "SuperGF: Unifying Local and Global Features for Visual Localization", *arXiv preprint arXiv:2212.13105*, 2023. (To be submitted to the International Conference on Computer Vision (ICCV) 2023).

Invited Talk

1. **Wenzheng Song**, Masanori Suganuma, Xing Liu, Noriyuki Shimobayashi, Daisuke Maruta, Takayuki Okatani, "Matching in the Dark: A Dataset for Matching Image Pairs of Low-light Scenes", *The 25th Meeting on Image Recognition and Understanding (MIRU)*, 2022/7/18, Himeji, Japan.

Acknowledgments

Firstly, I'd like to express my thanks to my patient and supportive supervisor, Professor Okatani Takayuki, who has supported me throughout this research project. Before entering this lab, I was a layman of computer vision or deep learning. At that time, I wondered how I could be accepted into this laboratory. Fortunately, Professor Okatani Takayuki allowed me to learn and improve myself. As my supervisor, he often needed to be patient with my stupidity. I still remember when I presented my ideas with poor presentation skills. However, he has never given up pushing me in the right direction. Without his support and supervision, I could not imagine finishing this research. I thank him so much.

Secondly, this dissertation couldn't have been completed without the help of many people surrounding me. It is not only physical but also those who send their warmth from far away. It must be said that I survived a difficult situation with their kindness and support.

I want to thank Research Assistant Professor Liu Xing, even though he has left the research lab. He is a friend and teacher who give many helpful suggestions about research and daily life. I also want to thank Assistant Professor Suganuma Masanori, because he manages our lab server; this has helped me a lot in my research.

I got a lot of help from Mrs. Sakane Akemi, the secretary of the laboratory. She facilitates every member of the laboratory with her kindness. All the activities in the laboratory have never been done without her support.

My senior, Junjie Hu, helped me a lot at the beginning when I entered this lab. He introduced me to many things about this lab and basic knowledge about deep learning.

My contemporaries, Qian Ye, Zhijie Wang, Kangjun Liu, and Xiangyong Lu, also greatly help me during my studies. Zhijie Wang helped me to sever a lot of environmental bugs. I learned a lot during the discussions with Qian Ye and Kangjun Liu. Xiangyong Lu enabled me to collect the MID dataset. I also want to thank Kittitouch Areerob, who also helped me a lot in collecting the dataset.

I thank all of my friends in the laboratory, regardless of their nationality. Also, I thank all the friends I have outside the campus. Although I do not mention everyone, please be reminded that I am so grateful to spend time with them. A perfect moment cannot be possible without them.

Not only the people who physically stay near me, but I also thank my friends and family from far away. They were forced to listen to me whenever I was in trouble. Many lessons

have been taught by failure. I also thank myself for the past, which produces the mistakes. And I will not make as many mistakes as in the past. That is the meaning of *learning*, for any errors or inadequacies that may remain in this work.