ROBUST LOCAL IMAGE FEATURE EXTRACTION AND MATCHING FOR CORRESPONDENCE ESTIMATION AND 3D RECONSTRUCTION

by

Usama Sadiq

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering

Examination Committee:	Dr. Rehan Hafiz Dr. Mohsen Ali
External Examiner:	Dr. Khawar Khurshid School of Electrical Engineering and Computer Science NUST, Pakistan
Previous Degree:	Bachelors of Science in Electrical Engineering University of Engineering & Technology, Lahore, Pakistan

Information Technology University of the Punjab
Electrical Engineering Department
Lahore, Pakistan
August 2020

All models are wrong, but some are useful.

George E.P. Box (1919-2013)

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every possible effort has been made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. I also declare that this work is the result of my own investigations and findings, except where identified by references and free from plagiarism of the work of others.

Signature: _____

Usama Sadiq

The undersigned hereby certify that they have read and recommend the thesis entitled "Robust Local Image Feature Extraction and Matching for Correspondence Estimation and 3D Reconstruction" by Usama Sadiq for the degree of Master of Science in Electrical Engineering.

Prof. Dr. Rehan Hafiz (Supervisor)
Department of Electrical & Computer Engineering
Information Technology University of Punjab

Dr. Mohsen Ali (Co-Supervisor)
Department of Computer Science
Information Technology University of Punjab

Dr. Khawar Khurshid (Co-Supervisor)
School of Electrical Engineering & Computer Science
National University of Science & Technology

Dr. Muhammad Zubair
Chairperson, Department of Electrical Engineering
Information Technology University of Punjab

Dedicated to my father, my mother and Tehreem Sadiq.
For their unconditional love & support.

Acknowledgments

I would like to thank my advisor Dr. Rehan Hafiz and co-advisor Dr. Mohsen Ali for their constant support, help and technical assistance throughout my research. This work would not have been conceivable without their guidance and supervision. They have provided me with all the freedom, technical support and resources to complete this thesis work. Throughout the time I have spent under their supervision, I have deeply benefited from their knowledge and experience. I am so grateful for their valuable time, constructive criticism and insightful comments on my work. Their positive attitude towards their profession, supportive and encouraging behavior was a big help during this journey.

I am also thankful to Dr. Khawar Khurshid of National University of Science and Technology for his valuable time and astute comments to my work. I am also grateful to Muhammad Faisal, a good friend and colleague, for his support and help. It would not be just if I fail to mention the unwavering support and love of my family during this journey. Lastly, I am also thankful to Department of Electrical Computer Engineering at Information Technology University for the award of Graduate Student Fellowship during my master studies.

Abstract

Matching two views of the same underlying scene and establishing the robust correspondences is the long-standing fundamental problem in computer vision due to its numerous applications such as Structure from Motion and Multi-view Stereo etc. Quality and discriminative power of local features play a pivotal role in determining the quality of established correspondences. Many effective methods have been devised for learned local descriptors using the powerful machinery of deep neural networks. However, the work on the learned keypoint extractors is quite limited even in this era of the deep learning. Some efforts have been made to employ the handcrafted features as anchors in learning a keypoint extractor. In this work, we argue that features from pre-trained deep convolutional neural networks work better as compared to handcrafted features. We propose *Multi-Scale Keypoint (MSK)* extractor, a stand-alone keypoint extractor, which employs a pre-trained feature extractor such as ResNet-50 for learning multi-scale keypoints using the effective receptive field of convolutional neural networks. We introduce the concept of information change across varying receptive field within the same depth in a convolutional neural network to associate the scale of a keypoint with the receptive field. We evaluate our proposed approach on numerous evaluation metrics and tasks using various datasets to show the improvement in performance gains over the current state of the art keypoint extractors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter starts with a gentle introduction to the correspondence estimation problem and motivates the reader about the complexity of problem and its applications in computer vision. Second, it gives a generic introduction to structure from motion (SfM) and 3D image reconstruction pipeline. Third, it states the problem statement of this thesis and describes the research challenges. Finally, this chapter concludes by stating the major contributions of this thesis work and gives a brief description of the organization of the thesis.

1.1 Motivation

Matching two or more views of the same scene is a long-standing fundamental problem in computer vision. Also known as correspondence estimation refers to the problem of determining which parts of one image correspond to which parts of another image of the same scene. All solutions of matching different views of the same scene eventually lead to correspondence estimation problem i.e. how a particular area in one image match with an area in another image. Correspondence estimation problem lies at the centre of many key problems in computer vision e.g. structure from Motion (SfM), 3D image reconstruction, object recognition, camera calibration, image alignment & stitching, image retrieval, optical flow, re-localisation and visual odometry and simultaneous localisation & mapping (SLAM) to name a few. Although this is a key problem in computer vision, yet, so far no general dependable solution exists to solve it with high accuracy.

Although numerous different techniques have been employed since the dawn of the research in establishing correspondences between two views of a scene [19], but recently, for last two decades, image matching and resulting correspondence estimation problem has been tackled using sparse methods i.e. *local features*. Local features [1] further consists of *local keypoint* or *interest point*, which represents a localized region of an image, and its corresponding *descriptor*, which is a representation of that region of image in the form of a vector of real numbers. The work on local features can be classified into two main categories. Traditionally, the majority of local feature extraction methods were based on *handcrafted* techniques, but most recently, with the advances in the ability of convolutional neural networks (CNNs) to learn *efficient* feature representations directly form the imaging data has given

[1] Although local feature is a collective term used for a keypoint and its corresponding descriptor, but in this work, we use term local feature for keypoint or interest point extractor unless stated otherwise.

birth to a new and rather superior class of *learned local features*.

Despite decades of research, the image matching problem remains unsolved in the general wide-baseline scenarios. There are three general challenges associated with this problem. First, there is no generic definition or constraint to reduce the ambiguity of problem. Second, it suffers from high complexity due to the huge dimensionality of the combinatorial search space in the general setting of wide-baseline stereo. Finally, the presence or absence of new local features in a sequence of wide baseline images and due to inherent inaccuracies of local feature extraction methods, outliers naturally occurs and this natural existence of outliers must be considered carefully in this matching. In addition to that, extreme variations in the conditions under which images are taken e.g. viewpoint and illumination changes, occlusions and camera properties, are major factors that make correspondence estimation a very hard problem to tackle in most practical conditions.

1.2 Structure from Motion (SfM)

Incremental structure from motion, simply known as structure from motion (SfM), is a process of reconstructing the 3D structure from a large sequence of unordered images taken under different imaging conditions e.g. viewpoint and illuminations [20]. It is a sequential process with iterative reconstruction stages repeated over many iterations. The first and fundamental step in this process is the estimation of robust and stable correspondences. This step is also known as correspondence search. This step begins with feature extraction and subsequent feature matching to extract the robust correspondences which are subsequently used in geometric verification [2, 3]. Correspondences obtained from this step are further used in incremental image registration and eventually employed in triangulation to get 2D-3D correspondences. Finally, these triangulated 2D-3D correspondences are fed to bundle adjustment, which re-adjusts these correspondences in such a way that minimizes the re-projection error which is the main goal of bundle adjustment.

1.3 Problem Statement

A prevalent approach for image matching and correspondence estimation involves the local features extraction and subsequent matching. Consequently, the overall performance of feature matching strongly depends upon the quality and distinctive power of local features. Traditionally, the local feature extractor attempts to localize the visual and geometrical structures present in the images through some engineered algorithms [21], which are often termed as *handcrafted* local feature extractor. Recently, many learning based methods, that learn to predict features directly from imaging data, have

been proposed for feature extraction and description which are termed as *learned* local features extractor. Although, local features based solutions to image matching problem have been proved quite popular due to their comparatively good performance, flexibility and robustness as well as due to modular nature of such feature extraction methods and applicability in wide range of applications. Still, their performance is only limited to the controlled experimental settings and intermediate tasks. In more practical conditions and downstream tasks e.g. pose estimation, performance of traditional local features based methods appears to be deteriorating [4, 11]. Although, recently proposed deep learning based methods have exhibit superior performance as compared to their handcrafted counterparts. Still, the ability of all these methods to perform under challenging real world condition is quite limited [4].

Deep convolutional neural networks (CNNs) have shown extremely remarkable performances on many challenging tasks like image classification [22–24], object detection and image segmentation [25]. However, in this era of deep learning the focus of the most of work in local features remained specifically on metric learning i.e. descriptor generation. Very limited amount of work is found on learning to predict the keypoints directly from data [12, 26]. Since, regions of deep concavity, simply local features, plays a key role in human and machine visual recognition tasks [1]. *Thus, a question arises that how can we use the already learned representation of deep convolutional neural networks for finding robust low level local features.* Moreover, how can we utilize the other properties of convolutional neural networks e.g. receptive field for establishing properties of local features e.g. locality of region. Therefore, we model a local feature extractor that employs transfer learning of already learned knowledge of CNN's to predict robust keypoints. Moreover, we also study that how can we use the receptive fields of CNN's to extract the locality of local feature's region.

1.4 Research Challenges

In recent years, growing literature has emerged on the methods describing local feature extraction and description or both. Several engineered algorithms as well as learned methods have been presented that guarantee to achieve the state-of-the-art results in the challenging conditions, for instance see [**?**]. Although all these methods claim to perform well on intermediate tasks and performance metrics, yet their performance in practical conditions, such as wide base-line scenario, deteriorates rapidly, which ultimately results non-modularity and application specificity. As in many real-world applications, the images are usually taken under extremely varying conditions e.g. under viewpoint, illumination and temporal variations. Therefore, a fundamental research challenge is to design a local feature extractor which works well not only in limited experimental settings but it's performance must generalize to real world conditions as well.

Another serious challenge in developing a deep learning based local feature extractor is its training approach. Since, it is impossible to annotate the imaging data for keypoints location and shape through human or machine assistance, therefore, it is very hard to train a deep learning based local feature extractor in supervised manner. Many recent works [12, 27, 28] uses the output of handcrafted detectors to train the their networks in self-supervised manner, which in turn limits the ability of the feature detector to predict the location of new and novel keypoints. Unsupervised training is important for local feature extractors to perform well on real-world conditions. We explore the unsupervised training approach with loss function that helps the network to enforce the covariant constraints. In this way, our network not only learn to predict keypoints from wide range of features but also constraint their locations under geometric transformations. Finally, we evaluate the performance of our local feature extractor on many intermediate as well downstream tasks to test its performance in large set of different tasks.

1.5 Thesis Outline

In Chapter 1, we motivate the reader about the correspondence estimation problem and describe its applications and complexity. Later, we state the problem statement of this thesis work and describe the potential research challenges. Thesis outline and chapter-wise major contributions of this thesis work has also been provided.

In Chapter 2, we give a detailed overview of problem of correspondence estimation and local features. We start by introducing the local features and its types. Later, we detail out the major works done on the local features in last two decades.

In Chapter 3, we present a keypoint extractor, *Multi-Scale Keypoint Extractor (MSK)*, as our major contribution of this thesis work. We describe the basic ideas behind this feature extractor and then explain its all modules, loss function, training strategy and training dataset.

In Chapter 4, we evaluate our proposed keypoint extractor and discuss the results. We first describe all the evaluation tasks, strategies, testing dataset, We also describe the evaluation metrics and their importance. At last, we analyze and discuss our results and indicates the performance gains of our keypoint extractor as compared to state-of-the-art.

Finally, in Chapter 5, we conclude this thesis by highlighting the major findings of this work and potential future research directions.

1.6 Technical Contributions

The main contribution of this thesis work is to develop a convolutional neural network based keypoint extractor. First, we interpret a keypoint in the context of a convolutional neural network and information change. Second, we develop a multi-scale keypoint extractor which exploits the idea of transfer learning and utilize the receptive field of CNN's to build scale space. Finally, we show the validation of our proposed approach by highlight the performance gains on numerous intermediate metrics and downstream tasks e.g. structure from motion and camera pose estimation. Chapter-wise major contributions of this thesis work are given below.

Chapter 2

- We take a comprehensive review of all the work done in the field of local features and image matching over the last two decades. We detail out strengths and weaknesses of all major handcrafted as well as learned local features extractors. We also highlights the performance deterioration and limitations of these detectors in some real world applications.

Chapter 3

- First, we define a keypoint in the context of convolutional neural network using the concept of *information change* across varying receptive field.
- As our main contribution, we develop, *Multi-Scale Keypoint (MSK) extractor*, a keypoint extractor by exploiting the already learned representation of features in pre-trained CNN's. We combine the idea of *information change* and *receptive field* to build the *continuous scale space* for representing the keypoint's locality.
- We develop a *differentiable* spatial softmax layer (SSL) for implementing the covariant constraint cost function to train our proposed network in unsupervised settings.
- Finally, we present a learned keypoint extractor that predicts keypoints using pyramid representation without incorporating the pyramid in training.

Chapter 4

- We present a complete set of evaluation tasks including intermediate metrics as well as downstream tasks to validate the performance of our proposed approach.

- First, we compute the *repeatability* and *matching score* of our keypoint extractor on numerous challenging datasets. We refer these performance metrics as intermediate metrics or tasks.
- Second, we extensively evaluate the performance of *MSK* on *Incremental Structure from Motion*. It is iterative and very extensive evaluation that measures how well a keypoint extractor works in real-world applications.
- Finally, we test our proposed keypoint extractor on a crucial task of *camera pose estimation* by competing in open *Image Matching Challenge.* We discuss the results of all these evaluation in details.

Chapter 2

Literature Review

This chapter provides an overview of local features and describes their applications in computer vision. We start by giving a brief introduction to local features and their two main components. We also highlight the major applications of image matching and local features. Later, we show how the overall performance of the image matching and subsequent tasks strongly depends upon the quality and robustness of the underlying local features. Finally, we take a comprehensive overview of all the research work related to local feature extraction methods. We classify the major research work on the local features detectors into handcrafted and learned algorithms and discuss their strengths and weaknesses, separately.

2.1 Local Features

Matching two or more views of the same scene is a fundamental problem in computer vision. Almost all important problems in computer vision eventually lead to the key problem of correspondence estimation between two views of the underlying scene. Although numerous different techniques have been employed since the dawn of the research in establishing correspondences between two views of a scene [19], but for last two decades, image matching and resulting correspondence estimation problem has been tackled using sparse methods i.e. *local invariant features*.

Simply speaking, local features are image regions of deep concavity. These are salient image regions which play a key rule in human and machine vision [1]. In computer vision, we often meet the local features in two main contexts i.e. local regions of an image and its description is the from of a vector of real numbers. Generally, a local features consists of a *local keypoint* or *interest point*, which represents a localized distinctive region of an image, and its corresponding *descriptor*, which is a representation of description of that region of image in the form of a high dimensional vector of real numbers. The process of finding the precise location and the locality of a *salient* image region i.e. a keypoint is termed as keypoint detection or extraction, while, the process of describing a keypoint in the form of a high dimensional vector of real numbers is called *description* or *descriptor generation*. Generally, in literature, the term local feature is collectively utilized to refer both keypoint and its corresponding descriptor.

Local features detection and description is a pivotal step in many computer vision applications. Therefore, it is one of the most extensively studied problem in computer vision due to the numerous im-

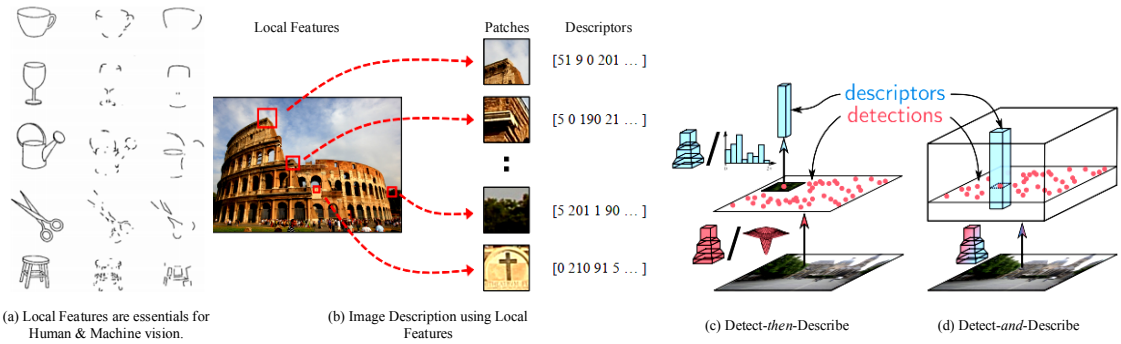


Figure 2.1:: Local features play a pivotal role in human and machine vision tasks. (a) Theory of recognition by components [1] states that regions of deep concavity i.e. *local features* are essential components of human visual recognition tasks. (b) Image description using local features i.e. keypoints and their corresponding descriptors. Two different approaches are employed in literature for the local features extraction (c) detect-*then*-describe and (d) detect-*and*-describe.

portant applications. Local features have been successfully employed in wide range of systems and applications including face as well as object recognition, large scale image retrieval (LSIR), motion detection & tracking, optical flow, depth maps generation, panorama stitching, camera pose estimation, structure from motion (SfM), 3D reconstruction, visual odometry, simultaneous localisation and mapping (SLAM) etc. Studies [1] have demonstrated the extreme important rule of local features in object detection and differentiation in human visual recognition system as well.

In almost, all major computer vision applications especially iterative image-based 3D modelling pipelines such as incremental Structure from Motion (SfM), Multi-view Stereo (MVS), visual Odometry(VO), simultaneous localisation and mapping (SLAM), image retrieval, camera pose estimation and image-based camera localisation, the most performance-defining & result effecting step is local feature extraction and subsequent matching. All the aforementioned processes are iterative and of incremental nature. In all these application, numerous 2D projections of the underlying scene are used for retrieving the scene geometry. Local features matching is the first and foremost step in all these processes. Error accumulated in this stage propagate through subsequent stages and effects the end goal. Therefore, *robust* local feature extraction and correspondence estimation is of fundamental importance in all of these applications.

2.2 Related Work

To achieve the best performance in robust local image feature extraction and matching is of significant importance to large part of computer vision community. Therefore, keypoints extraction and

(a) Structure from Motion (b) Point Cloud Generation (c) Wide Baseline Image Matching

(d) Dense Reconstruction & Multi-view Stereo (e) Camera Pose Reconstruction and Sparse Reconstrcton

Figure 2.2:: Local features have numerous application in computer vision such as (a) Structure from Motion [2, 3] (b) Point Cloud Generation (c) Wide Baseline Image Matching [4] (d) Dense Reconstruction and Multi-view Stereo [5] (e) Large Scale 3D Modelling from 2D Images [3]. All these applications are of iterative nature and have incremental components. Correspondence search is first and foremost step in all these applications. Robust correspondences are established by extracting local features and subsequent matching.

description is an extensively studied problem in computer vision literature. Consequently, this area has received a continuous attention over the last two decades, there are many surveys [8, 29–31] that discuss the advancement of feature extraction methods over the years.

Generally, local feature extraction pipeline consists of two main components i.e. keypoint detection and its descriptor generation. In literature, keypoints are the distinctive image regions with a defined locality also called *local interest points*. Whereas a descriptor is the description of the localized image region surrounded by a keypoint. Specifically, a descriptor is a representation of localized image region of a keypoint in the form of a high-dimensional vector of real numbers and usually called as *local descriptor*. The work on the local features is too vast to review in single stream. Therefore, the work on local features can be classified into two main categories. Traditionally, the majority of local feature extraction methods were based on *handcrafted* techniques, but most recently, with the advances in the ability of convolutional neural networks (CNNs) to learn efficient feature representations directly form the imaging data has given birth to a new and rather superior class of *learned local features*.

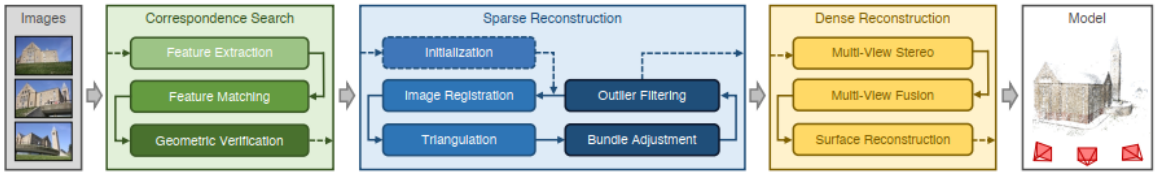


Figure 2.3:: Traditional image based 3D reconstruction pipeline using spare methods i.e. local features for the correspondence search, an incremental algorithm for the sparse reconstruction and multi-view stereo for dense reconstruction. This traditional pipeline is of iterative nature and have incremental components for sparse reconstruction. Correspondence search is first and foremost step in all these applications. The error accumulated in this step propagates to all components and effects the overall end goal. Therefore, the nature of established correspondences determines the end goal which in turn depends upon the nature and robustness of the underlying local features.

In literature, there are principally two different approaches for feature extraction i.e. 1) *detect-then-describe* and 2) *detect-and-describe*. In *detect-then-describe* technique, first keypoints are detected and subsequently corresponding descriptors are generated from the patches extracted around the keypoints. In contrast, the *detect-and-describe* approach prefers to simultaneously extract the keypoints and their corresponding descriptors. Here, we take a detailed overview of major *handcrafted* and *learned* local features extractor, separately.

2.2.1 Handcrafted Local Feature Extractor

Before the advent of deep learning, handcrafted techniques were employed to extract local image features. The main idea behind the handcrafted techniques was to describe the local image regions as closely as possible using tools from the signal processing, image processing and pattern recognition such as edges, corners, blobs, circle, gradients, wavelet transform and other known affine shapes in general. The methods of describing the local image regions using such elements and their associated role were not always identical, which resulted in numerous different kinds of keypoints extractors and descriptors.

The earliest example of handcrafted keypoint detector is Harris [32] corner detector which uses combination of first order derivatives of an image to locate the specific image regions i.e. corners that fulfills the certain uniqueness and stability criteria. Hessian [33] is an extension of Harris corner detector for locating the blobs. It employs a combination of second order derivatives of an image to localize the blobs in images. [34, 35] combined Harris and Hessian detectors to tackle the problem of multi-scale and affine transformations in images. This is first method which introduces the affine regions to handle the generic viewpoint changes in real world applications. Wavelet-transform, another tool from signal processing, based methods were also explored in the problem of feature points

extraction and description but performance gains from such methods were negligible [31], therefore, such methods did not get popularity.

Scale Invariant Feature Transform (SIFT) [21], is in all likelihood the most popular handcrafted combined local feature extractor and descriptor. SIFT has become the staple merit in numerous computer vision applications since dawn of research in local features for last two decades. It provides both local keypoints and corresponding descriptors. In general, SIFT generates a gaussian scale-space and using the concepts from heat diffusion equation it proves that normalized difference-of-gaussian function has property of scale invariance. Subsequently, SIFT searches for all the potential keypoints in this efficiently implemented difference-of-gaussian function, which is though an approximation. SIFT also generates the corresponding descriptor by computing and combining the gradient magnitude and orientations at each image keypoint in a specified region. SIFT was the first widely successful effort for designing local feature detection and description. SIFT employs *detect-then-describe* technique i.e. it first detects local keypoints and then uses regions around these keypoints to generate corresponding descriptors. Subsequent methods attempted to improve its computational requirements and efficiency. Speed-ed Up Robust Features (SURF) [36] is second most prominent traditional local feature feature extractor i.e. it extracts both keypoints and generates corresponding descriptors. It was the first attempt, to improve the efficiency and computational cost of SIFT. Basically it uses image convolution to extract local interest points, but for lessened computational cost its employs *integral images* for computing convolution. For orientation assignment and subsequent descriptor generation, SURF uses Haar-Wavelet responses. On the other hand, KAZE [37] and it's extension AKAZE [38] improve the scale-invariance by employing the Hessian detector on non-linear diffusion scale-space instead of using long-established gaussian pyramid as in SIFT [21]. These two methods provides a unique and noble method of building a scale-space, but their computational costs heavily affect their applicability in real-world applications.

High curvature points usually exhibits the high probability of being an interest point. Intensity variation based methods apply mathematical morphology to extract such points [31]. One of such method is SUSAN [39], which computes a low level features such as edges & corners through voting strategy based on the fraction of pixels within a pre-defined regions which have identical intensity to the central pixel. The advancements in machine learning also influenced the way of designing the algorithms for local features extraction. Features from Accelerated Segment Test (FAST) [40], which is an extension of SUSAN, is the first machine learning based high-speed corner detector. It chooses the keypoints by comparing the intensity of central pixel with intensities of pixels on the circular pattern. Simply speaking, it runs a consensus problem in a predefined image region to select a candidate pixel as a keypoint. The central pixel is considered to be a good candidate for being a keypoint if a number of consecutive pixels on circular region around are consistently brighter than central pixel. Image segmentation techniques have also been studied in the context of the features extraction. A method coined as Maximally-Stable-External-Regions (MSER) is proposed in [41], it exploits the segmen-

tation technique and watershed like algorithm to find homogeneous intensity regions in the images, which fulfills a stability criterion over a wide range of pre-defined thresholds.

Numerous applications such as motion tracking, visual odometry and visual simultaneous localization and mapping (VSLAM) usually require real time feature detection and matching. In order to meet the requirements of such applications so-called *binary local features* have been successfully introduced in the literature to reduce the computational cost, time and memory footprint. Binary Robust Independent Elementary Feature (BRIEF) is a stand-alone binary descriptor. It is a 512 bit descriptor which uses random distribution pattern for point pairs in a local image region for comparison to compute a binary descriptor. Oriented FAST and Rotated Brief (ORB) [42] is a first complete binary local features extractor which has a detector, binary descriptor and a procedure for orientation assignment. ORB combines keypoints extracted by FAST detector and add rotational invariance to BRIEF descriptors [43] to show that said combination exhibits two times faster local feature extraction than SIFT. Binary Robust Invariant Scalable Keypoint (BRISK) [44], an another popular local feature extractor, that extract keypoints across both image and scale-space using a saliency criterion, and forms a binary descriptor through the gray scale relationship of sampled points pairs in the neighborhood detected keypoint. Matching speed of the BRISK features is faster than conventional approaches and also its descriptors requires less storage memory but at the cost of reduced robustness.

2.2.2 Learned Local Feature Extractor

Deep convolutional neural networks (CNNs) have shown extremely remarkable performance on numerous challenging tasks like image classification [22–24], object detection and image segmentation [25]. Consequently, in last few years many, deep learning based approaches have been proposed to either learn local descriptors directly from patches or to learn keypoint extractor or both i.e. a end-to-end matching pipeline. However, in this era of deep learning the focus of the most of work in local features remained specifically on descriptor generation or collective matching pipelines [16, 27, 28, 45–47]. Very limited amount of work is found on learning to predict the keypoints directly from data [12, 26]. Here we first take a detailed review of some major work of stand-alone keypoint detector and descriptors, then we review all the major works of combined keypoint detectors and descriptors i.e. matching pipeline.

Convolutional neural networks based keypoint detector attempts to predict the spatial location of a keypoint and optionally its locality i.e. scale while minimizing symmetric transfer error [20]. TILDE (Temporally-Invariant-Learned-Detector) [12] is probably the pioneering deep learning based stand-alone keypoint detector. TILDE actually works by training a regressor by using keypoints generated from the SIFT [21] as ground truth during training. It predicts a score map which is used to regress the location of the keypoints by applying the Non-maximum Suppression (NMS) in specified win-

dow size. DNet [48] is the first CNN based stand-alone keypoint detector which defines the features covariant constraints in context of deep neural network. It also trains a keypoint regressor by converting the feature covariant constraints into differentiable covariant constraint loss. Main limitation of DNet is that it applies covariant constraint in a pre-defined patch. The additional process of patch extraction limits its applications. By leveraging the pre-detections on the training data similar to TILDE, TCDet [26] builds on DNet's framework and exhibits increased training stability and improved features covariance. TCDet trains a regressor that predicts a group of affine transformations. The consensus of these transformations is used to predict the salient locations which are keypoints. Same as TILDE, TCDet also works on patches. Using ranking loss, Quad-Net [49] trains a two layered shallow network for keypoint detection in an un-supervised manner. The Quad-Net technique is sluggish and impracticable because of the patch-based prediction mechanism.

Recently, KeyNet [6] introduced a multi-scale learned stand-alone keypoint detector that combines the responses of conventional handcrafted filters as learning anchors with learned convolutional filters in a hybrid framework. KeyNet argues that the output features from these time-tested handcrafted filters [7] work as anchors structure during training of subsequent convolutional filters which uses the output feature map of these filters to learn a covariant keypoint detector. KeyNet uses handcrafted filters from the *local-jet* [7] which approximates the image representation in local neighborhood using Taylor series expansion [6]. KeyNet uses first and second order image derivatives along with gaussian smoothing to suppress the noise. Higher order image derivatives are not used as these are quiet sensitive to noise and require large kernels. Therefore, KeyNet only includes first & second order image derivatives and their combinations. For gradient based features extraction, KeyNet employs convolutional layers with batch-normalization and ReLU activation to minimize the modified the covariant constraint loss. KeyNet argues that using the output of handcrafted filters helps during the training of learnable filters during training.

Ideally, a keypoint detector must be robust against scale and viewpoint changes in the imaging conditions. In order to induce the scale-invariance, KeyNet uses the scale space representation using multi-scale pyramid of images within their network during training as well as inference. KeyNet makes an image pyramid containing down and up-sampled images and passes the each image from handcrafted and learned convolutional filters and finally concatenates the output feature map from all levels of the pyramid to make a features volume which is further passed from single convolutional layer to produce a single output score map.

KeyNet trains in an unsupervised manner without any ground truth using a siamese pipelines. It uses an image pair with original and transformed image with known ground truth geometric transformation. KeyNet uses a modified covariant constraint loss which constraints the locations of the keypoint, within a spatial window of specified size, in both original and transformed image. For extracting the location of keypoint within a window in a differentiable way, KeyNet proposes a Index Proposal (IP)

layer which extracts the location of a keypoint within a window of size NxN. Since spatial softmax function is a differentiable method for projecting the soft maximum within a window/region. Therefore, to ensure the differentiability of IP layer, KeyNet uses soft-arg-max function for extracting the location of a candidate keypoint as weighted average of all the coordinates within a window. Due to exponential scaling, the location with highest score in score map dominates. In this way, the expected location of strongest keypoint is calculated as the weighted average which gives the approximation of coordinates of strongest keypoints. A detector is said to be covariant feature detector if it detects same local features under varying image transformations. Therefore, the covariant constraint loss, within a window, is computed as euclidean distance between the coordinates returned by the IP layer and actual maximum coordinates extracted through Non-maximum suppression (NMS) in the same window. This euclidean distance is weighted by the sum of the scores at the both coordinates to give high loss values to strongest keypoints in both windows. In this way, KeyNet tackles the problem of applying the covariant constraint loss on the patches as in TCDet [26] and DNet [48]. KeyNet computes the loss within different window sizes. KeyNet extends the loss for multi-scale by extending the IP layer for different window sizes. KeyNet refers it as multi-scale index proposal (MSIP) layer. The final loss is computed by adding the individual loss values for all the window sizes across all the scales.

With the exception of Quad-Net, all of these above mentioned methods still rely on handcrafted detectors to create ground truth or to act as anchors in learning process. Although anchor structures increase training stability and prove helpful during training, but anchors also stop the network from predicting new keypoints when there isn't an anchor in vicinity. Due to these constraints, the learned detector can only learn features that are analogous to handcrafted features.

2.2.2.1 End-to-End Matching Pipeline

The second class of learned local features extractor is a combined unified pipeline for joint keypoint detection and description which is termed as *end-to-end matching pipeline*.

LIFT (Learned Invariant Features Transform) [28] is the first deep learning based end-to-end matching pipeline that merges both keypoint detection an description into a single unified pipeline. It leverages ground truth produced by SIFT based structure from motion (SfM) and merges previously developed keypoint detector [12], orientation estimator [50] for patch extraction, and descriptor [51] into a single unified pipeline. Since this pipeline consists of different neural network for keypoint detection, orientation estimation and descriptor generation which makes it slow for practical applications. SuperPoint [47] proposes a CNN based pipeline for the joint keypoint detection and description. It has a similar structure like D2Net [46] i.e. it works on *detect-and-describe* technique for parallel keypoint detection and descriptor generation. It generates a score map for extracting the keypoints using non-

maximum suppression (NMS) and features volume having similar spatial resolution as score map but having depth equal to 128 channels for generating descriptors. It is trained on using a synthetic dataset with known corner points locations as ground truth. Since it works on *detect-and-describe* technique, therefore, it does not have scale and orientation estimation modules as LFIT [28] or LF-Net [27]. LF-Net [27] is an extension of LIFT. It trains a single convolutional neural network for keypoint detection, scale & orientation estimation for patch extraction and subsequent descriptor generation in a single forward pass. It works on *detect-and-describe* technique that first detects the keypoint first and then extract a patch around the keypoint using estimated scale and orientation which is further fed to descriptor head. LF-Net uses the first few blocks from the Residual Network (ResNet) [23] as base network and resizes its features to create a scale space. LF-Net uses the SIFT based structure from motion(SfM) output during training.

RF-Net [52] is another unique end-to-end matching pipeline which encompasses both keypoint detection and description. It is the first method which employs the receptive field to estimate the locality of keypoint. It actually associates the scale of the keypoint with the receptive field of the convolutional neural network. RF-Net's structure is similar to the LF-Net, except the method of scale estimation. It also works of the *detect-then-describe* technique. However, in contrast to LF-Net, RF-Net trains in an unsupervised manner using the real ground truth homography i.e. HPatches [9] dataset. RF-Net trains an *N* keypoint detector, hierarchical convolutional layer, where $N = 10$. The proposed method can only detect the keypoints up to the scale equal to receptive field of last filter. More recently, D2Net [46], a joint detection and description pipeline for local features, attempts to the tackle problem of illumination changes in the imaging conditions using a single CNN based framework. It's structures is similar to the SuperPoint [47]. D2Net argues that a single CNN based framework can be employed for the both detection and description. More precisely, D2Net first generates the dense local descriptors which are further processed for generating the keypoints. R2D2 [53], (Reliable and Repeatable Detector and Descriptor, hence the name), is another recent work based on the *detect-and-describe* technique, which performs joint interest point detection and description in a single unified pipeline. However, it argues that not all regions in an image can generate descriptors which can be matched with high confidence. Therefore, along side the repeatability map for keypoints and dense features map for descriptors, R2D2 also generates a third map, termed as reliability map to increase the discriminative power of the local descriptors. Reliability map is a predictor of the discriminative power of the local descriptor at that location. R2D2 is trained using the correspondences estimated from optical flow and keypoints verified by the SfM pipeline.

ELF [54] (Embedded Localisation of Features) is a recent novel work that proposes to develop a keypoint detector by distilling the information embedded in a convolutional neural network trained of some other task e.g.classification or segmentation. It also generates the descriptors as well using interpolation. First it generates the image level features using a pre-trained CNN and saliency maps are generated using the gradient of images features. These saliency map provides the local maxima

as locations of the keypoints. However, performance of the proposed approach is not up to the mark. GLAM [55] (Greedily Learned Accurate Match points) is another instance of recent work similar to LF-Net. It is trained in semi-supervised settings by using pseudo ground-truth. It is application specific framework for medical applications e.g. medical image matching. It trains using a differentiable version of image matching using descriptors and uses the locations of true positive matches as a candidate keypoint. Domain-specific nature of GLAM places a limit on its application to other difficult tasks, for instance 3D reconstruction.

It's widely accepted that convolutional neural networks trained on a substantial datasets are capable of representing an input image accurately. Pre-trained CNNs are frequently used as base feature extractor or for fine-tuning on tasks other than those for which those were trained. Recently, Deep Image Prior [56] have shown that the structure of the convolutional neural networks works as prior in numerous vision tasks. Even prior to learning or training, the structure of the CNN allows it to capture the significant low level image features in numerous inverse-problems such as image denoising, inpainting and super resolution. We argue that features map generated from the pre-trained deep CNN has enough low and high level information which can act as prior for distilling these features for extracting keypoints using the concept of information change within the network. We use a akin argument like the one used in Deep image Prior(DIP) [56], that the structure of a convolutional neural network is sufficient to capture a great deal of low-level image features without any prior training and it learns high-level image features with proper training.

In summary, we argue that all the above mentioned methods somehow uses some kind of anchors to help train their networks. LIFT [28], LF-Net [27], D2Net [46] rely on output of SIFT based structure from motion (SfM) for training. R2D2 [53] relies on the correspondences generated from the optical flow and keypoint verified from the SfM for learning. SuperPoint [47] uses hand annotated corner locations as ground truth and TCDet [26] reckons the output of TILDE [12] during training. KeyNet employs output of hand-crafted filters as input to its learnable module. We argue that the use of keypoints from traditional feature detectors, SfM verified points or or any other kind of anchors limit the network's ability to generalize and propose new keypoints. Unlike these said approaches, our network formulation allows us not to rely on any of the above mentioned ground truths or anchors.

Conclusions

We took a detailed overview of the local features extraction methods and different approaches employed in literature. We started from handcrafted techniques and discussed all methods based on the detect-then-describe approach. We also discussed the re-emergence of the problem of local feature extraction in this era of deep learning. We highlighted the major stand-alone keypoint detectors, descriptors and combined matching pipelines. We noted that the inherent structure of convolutional

neural networks helps to learn the low level representation of such local features and provide an opportunity to distill these features which has robustness and stability to be used in establishing the robust correspondences in image matching.

Chapter 3

Multi-Scale Keypoints

In this chapter, we present Multi-Scale Keypoint (MSK) Extractor [57] as our main contribution. We start by describing the main concept and motivation behind our proposed approach. Then we discuss the architecture of MSK in detail and highlight the working of all of its modules. We also address the problem of selecting the loss function to guide the training of proposed network. Finally, we give the training and implementation details and discuss the two variants of MSK to indicate the stability of our idea.

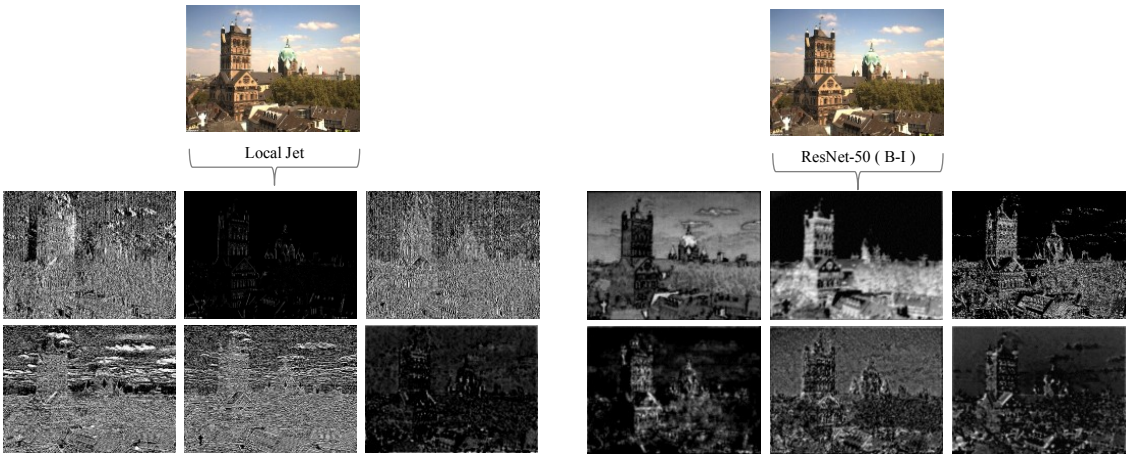
3.1 Motivation

Regions of deep concavity in images, that is salient features, play a key role in human and machine visual recognition tasks [1]. It has been shown through various studies that deep convolutional neural networks trained on large datasets have a remarkable ability to automatically learn a *reasonable* representation of an input image or data. Moreover, convolutional neural networks have ability inherited in their structure to learn image priors from large imaging dataset [56]. One other reason for their superior performance is that their structure *resonate* with the underlying structure of the data i.e. fundamental building blocks of the images. Since local features are like fundamental building blocks of the images [1] and visual recognition. Thus, it is common practice to use the pre-learned representation or features of CNN's from one task and use in another task after a little which is referred as transfer learning [58]. This is main the idea behind our proposed model to utilize the learned representation of pre-trained CNN and use its features to further distill its pre-trained features using a trainable CNN based architecture. In addition to that, it is very challenging to incorporate a method of scale extraction in a keypoint detector. SIFT [21] builds a gaussian scale space to assign scale to keypoints and KAZE [37] uses non-linear diffusion scale space to extract the scales. In most deep learning based methods, there is no method for specifying the scale of the extracted keypoints. In contrast, we propose to build a scale space using a receptive field of convolutional neural network.

Transfer Learning

Key.Net [6] is a recently proposed stand alone keypoint detector which combines the conventional handcrafted filters from *local Jet* [7] and learned CNN filters to predict the interest points. It uses a

multi-scale pyramid for scale assignment. Key.Net argues that using handcrafted filters approximate the image in local neighborhood in a similar manner as Taylor expansion. These filters act as anchors during training and helps to reduce the number of parameters in subsequent learnable module, improve the stability and convergence during training.



(a) Key.Net Handcrafted Features (a) ResNet-50 Pre-Trained Features

Figure 3.1:: Comparison of features from the handcrafted filters [6, 7] and features from the learned filters from ResNet-50. (a) Features obtained by applying the handcrafted filters from *local jet* [7] on an image as used by KeyNet [6] and (b) Few randomly selected features maps from the feature volume obtained from pre-trained ResNet-50 (block-I). Clearly, the learned features from ResNet-50 has much *dense* low and high level information which helps in learning a keypoint extractor.

In contrast, we argue that using handcrafted filters as priors to network place an upper bound on the ability of the network to predict the novel features during inference. Instead, convolutional neural networks learn a very deep low-level image representation due to their natural structure [56] and due to the generalization ability. Therefore, we propose a novel keypoint extractor that employs the pre-trained CNN's features as priors and uses a learnable module to extract the keypoints by minimizing a symmetric covariant constraint loss.

Receptive Field

Unlike the gaussian scale space [21] or non-linear diffusion scale space [37], we build a scale space using the effective receptive field of the convolutional neural network. We define a local feature in terms of information change across a varying receptive field around an image region. Moreover, we extract the scale of keypoint as a *weighted average*, which results in a continuous scale estimation process which is much closer to real world concept of scale.

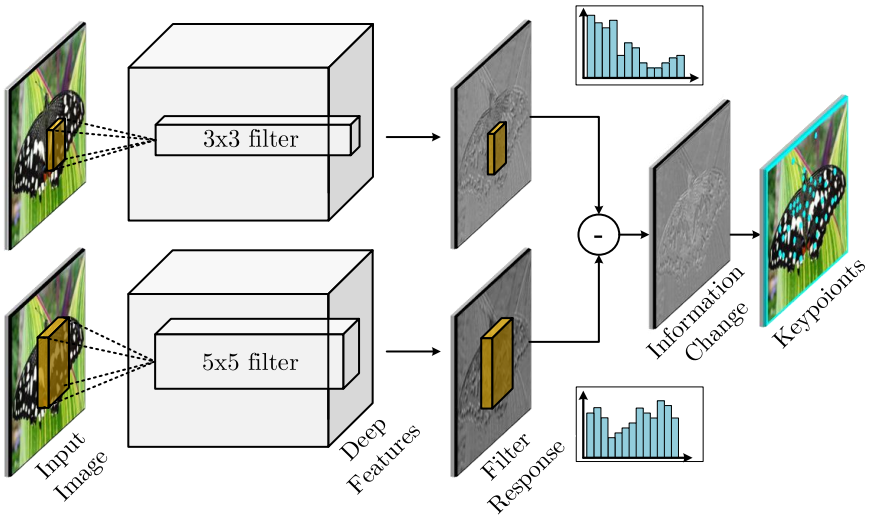


Figure 3.2:: Our proposed approach Multi Scale Keypoint (MSK) Extractor learns to regress the salient locations as keypoints which have maximum information change across the varying receptive field. Two filters, with different kernel size, having different receptive field at the same depth within the convolutional neural network fetch the different information from the image. We compute the change in information between two filters having different receptive field by subtracting their responses and identify the locations having maximum change in information as keypoints. This definition allows us to associate the scale of a keypoint with the receptive field.

3.2 Method

3.2.1 Keypoint in the Context of CNN

We hypothesize that not all the areas are equal in the representation of an image. Image representation through local features states that salient image regions can be represented through local keypoints or interest points and can be described using local descriptors around these interest points. Our objective is to find the sparse *salient* image regions to represent the image using sparse local features which are termed as local keypoints or interest points. With an objective to define a keypoint in context of a deep CNN, we define a keypoint as the location with maximum change in information across varying receptive field within the network. Specifically, we define a keypoint as a tuple (x, y, s, θ) where (x, y) is the center of the keypoint with locality or neighborhood of the keypoint specified with scale s and θ is the orientation of the keypoint. We argue that (x, y, s, θ) is a keypoint with location (x, y) if there is a maximum change of information with the change of the locality of the keypoint from scale s to $s + \delta s$ (δs being infinitesimally small change). In context of convolutional neural networks, the spatial location and locality of a keypoint, i.e. *information*, is defined as the learned weighted pooling of the deep hierarchical representation of the area generated by the network. Our definition of the keypoint in context of deep CNN gives us a chance to link the keypoint location and

size of its neighborhood with the receptive field of the network. It allows us to define information to be learnable transformation of deep features extracted at a specified level in network. We define *change in information* to be the change/variation in the learned responses of two convolutional filters, with different kernel sizes, hence, with different receptive fields within the same depth in the network.

Since, we define a keypoint to be change/variation in information of two filters having different receptive field. This change will be higher if the location is very different from its surrounding and vice versa. Our definition of the keypoint gives our network freedom to detect the same salient image regions across numerous different illumination and viewpoint changes in imaging condtions. The association of the locality or neighborhood (i.e. scale) of a keypoint with the receptive field of the network gives us a chance to build a scale-space representation using receptive field in contrast to KeyNet's [6] gaussian scale space.

3.2.2 Multi-Scale Keypoint (MSK) Architecture

We propose a stand alone learned keypoint extractor based on the concept of information change across varying receptive field. We name our proposed network "*Multi-Scale Keypoint (MSK)*" extractor. Our proposed keypoint extractor, MSK, consists of the five modules (a) A Base (pre-trained) Feature Extractor (b) an Information Accumulation Pyramid (IAP) module (c) an Information Change Detection (ICD) module, (d) Keypoint Detection module, and (d) a Continuous Scale Estimation module.

KeyNet [6] posits that utilizing the responses of the conventional handcrafted filters such as local-jet [7] as input to the gradient-based learnable convolutional filters acts as anchors and helps the convolutional filters to learn more stable and repeatable keypoints as well as helps convergence during training. On the contrary, we argue that using the learned features maps as input to the subsequent learnable module from the pre-trained CNN models, trained for other tasks such as classification [22, 23] and segmentation, work far better than the conventional handcrafted filters. Therefore, in MSK, we use the first block of the ResNet-50 [23] as our pre-trained base feature extractor. Figure 3.1 compares the features from the handcrafted filters [6, 7] and features from the learned filters from ResNet-50. Features obtained by applying the handcrafted filters from *local jet* [7] on an image as used by KeyNet [6] are shown on the left whereas few randomly selected features maps from the feature volume obtained from pre-trained ResNet-50 (block-I) are shown on right. Clearly, the learned features from ResNet-50 has much dense low and high level information which helps in detecting stable, distinct and repeatable keypoints. Although, we have employed the block-I of ResNet-50 as our pre-trained base feature extractor in MSK, but there are no restrictions on using any other base features extractor like [13, 22]. This gives our proposed approach *modularity*. It can be attached with any pre-trained base feature extractor which makes it conformable to a particular application. We

employ such low & mid-level representation since various studies have shown that it enhances the predictive ability of deep convolutional neural networks in various learning tasks [59] and therefore it aids in extracting a diverse set of features. We have discussed each sub-module of *MSK* in details below.

3.2.2.1 Base Feature Extractor

Recently proposed stand alone keypoint detector, KeyNet [6], which is a state-of-the-art learned feature extractor, argues that using the features form the conventional handcrafted filters such as *local-jet* [7] as input to the gradient based learnable convolutional filters acts as anchors and helps the CNN filters to learn more stable and repeatable keypoints as well as helps convergence during training. On the contrary, we argue that using the learned feature maps as input to the subsequent learnable convolutional layers from the pre-trained CNN models, trained for other tasks such as classification [22,23] and segmentation, work far better than the conventional handcrafted filters as depicted in figure 3.1. Therefore, we employ such low & mid-level representation since various studies have shown that it enhances the predictive ability of deep convolutional neural networks in various learning tasks [59] and hence it helps in extracting a diverse set of features. Therefore, in MSK, we use the first block of the ResNet-50 [23] as our pre-trained base feature extractor as shown in figure 3.3.

3.2.2.2 Information Accumulation Pyramid (IAP) Module

Information Accumulation Pyramid (IAP) module is the main part of the MSK. It further comprises of two sub-modules 1) Feature Transformation Block and 2) Information Accumulation Pyramid. Feature transformation block contains a non-learnable bi-linear interpolation and two convolutional layers each subsequently followed by Batch Normalization [60] & ReLU [61] activation function. Feature Transformation Block performs two primary tasks, upscale the input features and transform the features. Our pre-trained base feature extractor, ResNet-50 Block-I, produces a features volume which contains 256 channels having reduced spatial resolution as compared to the input image. This output feature volume is fed to feature transformation block which first of all restores the original spatial resolution of this feature volume using a non-learnable bi-linear up-sampling.

After restoring the original spatial resolution of this feature volume, it is fed to convolutional layers. We argue that many invariancies might have been learned at this part of the network. These convolutional layers serve two purposes. First, these layers are trained to transform this feature volume to one that can enhance its usefulness in identifying the keypoints. Second, these convolutional layers allow the network to learn to transform these features into more useful form that helps in minimizing

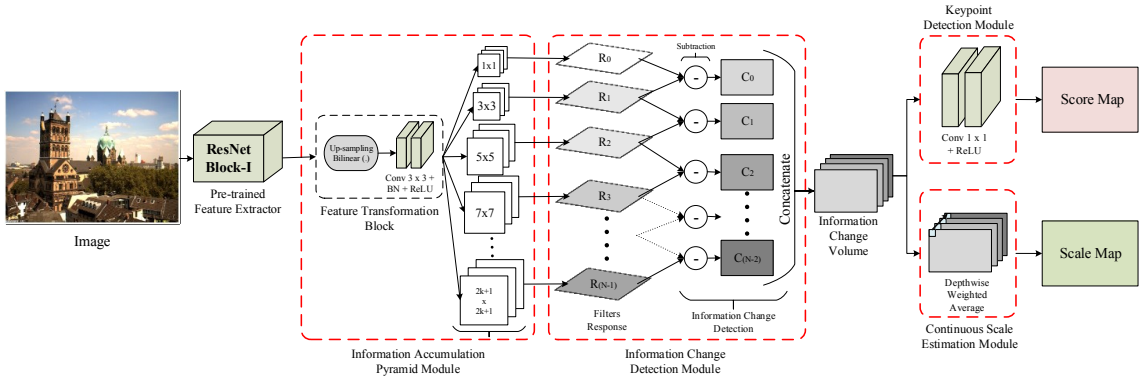


Figure 3.3:: Our propose approach, *Multi-Scale Keypoint (MSK)* detector's architecture. MSK mainly consists of five parts i) **Base Feature Extractor:** We use pre-trained ResNet-50 as our base feature extractor which is truncated after first block. ii) **Information Accumulation Pyramid Module:** This module accumulates the information across varying receptive field. iii) **Information Change Detection Module:** It computes the change in information across varying receptive field using pixel-wise subtraction. Finally, iv) **Keypoint Detection Module** and v) **Continuous Scale Estimation Module** produce score map and corresponding scale map.

the loss function without disturbing the base network. In addition to that, we make up-sampling and convolutional block customizable to introduce variants of MSK i.e. MSK-Tiny & MSK-Tiny++, for details see 3.3.3.

The output of feature transformation block is the transformed feature volume which is further fed to Information Accumulation Pyramid (IAP). Information Accumulation Pyramid (IAP) is a pyramid of N parallel convolutional layers. Each convolutional layer is subsequently followed by Batch Normalization [60] layer & ReLU [61] activation. These convolutional layers are ordered according to their increasing kernel size, thus, creating a scale-space, hence named pyramid. Assuming, there are N parallel convolutional layers in pyramid, size of k^{th} filter is set to $2k+1$, where $k = 0, 1, 2, 3, ...N-1$. All the convolutional layers in the pyramid are parallel within the same depth in the network, moreover, the size of convolutional filters increase downwards which results in increasing receptive field and varying receptive field as we move downwards in the pyramid. Therefore the output of the each filter different from the neighboring filter and it captures different information of the input image. Each filter accumulates the information at different scale due its different receptive field. Small changes in receptive field among these parallel convolutional layers allow our proposed approach to be robust against small scale changes.

3.2.2.3 Information Change Detection Module

Information Change Detection module computes the *variation* in information accumulated in the adjacent layers of information accumulation pyramid. The output of each convolutional layer of information accumulation pyramid is fed to information change detection module. This module computes the change across information of adjacent convolutional layers. Finite difference between adjacent filters responses is computed to highlight salient spatial locations where information variations across the scales are significant. Let R_k & R_{k+1} be the output of the k^{th} and $(k+1)^{th}$ filter, where $k = 0, 1, 2, 3, ..., N-1$ and N is total number of filters, in information accumulation pyramid module, respectively. Then, we define C_k be the change between responses of R_k and R_{k+1} as:

$$C_k = R_{k+1} - R_k \qquad \text{(Equation 3.1)}$$

Note that scale at which information is accumulated in R_{k+1} is greater than R_k, therefore, C_k computes the change in accumulated information across different receptive fields i.e. scales. The output of change detection module is a feature volume $V \in \mathbb{R}^{H \times W \times N-1}$, called the *Information Change Volume*. For the simplicity of implementation, we use subtraction as a forward difference function, with assumption that layers before it will learn to adjust for even this simple choice of function. We argue that change detection volume has important information regarding the significant locations in the image. This feature volume is further fed to keypoint detection and continuous scale estimation modules for subsequent keypoint detection and scale estimation, respectively.

3.2.2.4 Keypoint Detection Module

Keypoint Detection module produces the final score map for MSK. Change detection module's output is a feature volume $V \in \mathbb{R}^{H \times W \times N-2}$ referred as *Information Change Volume*. This feature volume contains information of salient spatial locations at each scale or channel. In order to map this information change volume $V \in \mathbb{R}^{H \times W \times N-1}$ to single score map, we use the technique of *cross channel parametric feature pooling* [62]. In cross channel parametric feature pooling, each pooling layer computes the weighted linear combinations of the input channels, which then are fed to some activation function. This cross channel parametric pooling technique allows learnable and complex interactions of cross channel information. We argue that cross channel information on the each channel of the *Information Change Volume* contains spatial locations of the keypoints. The cross channel parametric pooling layer is also equivalent to a convolution layer with 1×1 convolution kernel. In order to implement this parametric pooling, we use two convolutional layers each with kernel size of 1×1. First convolutional layer consists of $N - 1$ channels, which attempts to compute the complex

and learnable interactions of cross channel information in input feature volume. Second convolutional layer comprises of only single filter of size 1×1 and outputs a single channel. The final output $S \in \mathbb{R}^{H \times W}$, named score map, predicts keypoint score for each candidate pixel.

3.2.2.5 Continuous Scale Estimation Module

Continuous Scale Estimation module returns a scale map. Final scale map $U \in \mathbb{R}^{H \times W}$, having same spatial resolution as score map, gives keypoint scale for each candidate pixel. Since we define the scale of the keypoint in terms of the receptive field of the convolutional filters. Therefore, in contrast to discrete hierarchical scale space of RF-Net [52] or KeyNet [6], we build a continuous scale space using receptive field of convolutional filters and weighted average.

We use N parallel convolutional layers to generate feature maps $\{F^n\}, 1 \leq n \leq N$ with increasing receptive fields in Information Accumulation Pyramid module 3.2.2.2. Therefore, each response map in $\{F^n\}$ describes the abstracted features extracted from a certain region of the image and this range increases as we move down to aforementioned pyramid. We feed these feature maps to Information Change Detection 3.2.2.3 module which returns a feature volume $V \in \mathbb{R}^{H \times W \times N-1}$. This feature volume is fed to Continuous Scale Estimation module. This module returns a Scale Map $U \in R^{H \times W}$ where the value of each pixel $u(i,j)$ represents the scale of corresponding candidate pixel as

$$u(i,j) = \frac{\sum_{k=0}^{N-2} w_{i,j}^k \cdot r^k}{\sum_{k=0}^{N-2} w_{i,j}^k} \qquad \text{(Equation 3.2)}$$

where $w_{i,j}^k = V(i,j,k), k = 0,1,2,....N-2$ and r_k is receptive field of k_{th} filter. This makes our scale space continuous i.e. $u(i,j) \in [\, r_0, r_{N-1}]$. This continuous scale estimation allows the detector to extract keypoints whose locality is not restricted to some discrete levels (RF-Net [52], LF-Net [27], KeyNet [6]) rather to infinite many continuous levels within a bounded region.

3.2.3 Loss Function

In supervised learning, ground-truth data is needed to compute the loss and perform learning through backpropagation. But in case of keypoint detection, first, it is unclear that which locations are important, second, it is expensive to generate ground-truth through human labelling or annotation. A robust keypoint detector must be able to detect the same keypoints regardless of geometric and photometric transformations i.e. a robust detector must detect covariant [48] features.

Therefore, we train *MSK* in an unsupervised manner with an objective to minimize the symmetric transfer error [20]. For each image I_A in the training dataset, we generate a training image pair (I_A, I_B) by applying the random perspective and photometric transformation on the original image I_A and produce a transformed image I_B with known ground truth homography $H_{A,B}$. Training pipeline of MSK is based on siamese pipeline with weight sharing between its two instances. In siamese pipeline, the MSK takes the image pair (I_A, I_B) as input and generates two corresponding output score maps (S_A, S_B).

In the absence of ground-truth, most of the keypoint detection networks train in an unsupervised manner or semi-supervised manner. Some methods [27, 28, 46] rely on output of SfM for learning. Some learned detectors [12, 26, 27, 48, 49, 52] train their networks in an end-to-end fashion without constraining the location of keypoints. While other methods [12,26,47,49] show the anchor guidance leads to more stable training. Although anchors help the training to be more stable and produce better results in terms of convergence, but these anchors limits the ability of network to predict new keypoints.

Some learned detectors, LF-Net [27] and RF-Net [52], train their networks to identify key points without constraining their locations. But these networks are complete matching pipelines i.e. these networks embed both keypoint detector and descriptor head in their pipelines. LF-Net and RF-Net use a combination of image-level loss and patch-level loss to train their pipelines. This image-level loss tries to make the two score maps similar without constraining the location of keypoints. Since MSK is a stand-alone keypoint detector and it is trained in siamese training pipeline without any descriptor head, so training with image-level loss or similarity loss results in bad localization and loss of distinctiveness of keypoints. We constrain the locations of the keypoints using covariant constraint loss [48] to ensure better keypoint localisation. We use a covariant constraint [48] loss to train MSK for detecting keypoints.

3.2.3.1 Covariant Constraint Loss

For standalone keypoint detectors, it is hard to train network without constraining the location of keypoints. Extracting the location of a keypoint through a differentiable method is also a hard task. LIFT [28] introduces soft-arg-max function which gives the average location of keypoint in a patch or a window. Since soft-arg-max is differentiable, therefore, it approximates the non-maximum-suppression algorithm, which, on the contrary, is non-differentiable.

Similarly to hand-crafted techniques, keypoint locations are indicated by the locations of local maxima in score map output by MSK. Spatial softmax function is an effective method for suppressing the non-maxima in a window and highlighting the *soft* maximum. Soft-arg-max function returns the

average location of the scaled soft maxima in a window. Soft-arg-max function takes a score map as input and gives the average location of each dominant keypoint in a non-overlapping window. Spatial soft-arg-max function is an differentiable method for extracting the average location of a soft maximum within within a window [28, 47, 48]. So in order to ensure the differentiability of loss function, we employ soft-arg-max function similar to LIFT [28]. In LIFT [28], soft-arg-max function is used to extract the location of a keypoint in a patch, where only one keypoint location is returned per patch. In contrast to LIFT [28], we use soft-arg-max function on the output score map. For this, we divide our score map in non-overlapping windows and we apply soft-arg-max function in each window, similar to KeyNet [6].

In order to compute the covariant constraint loss between two score maps. We first compute the average location of the soft maximum in a window. For this purpose, we first divide the each score map into $N_s \times N_s$ non-overlapping windows, where subscript s represent the scale of the window. We compute the average location of the soft maximum using the modified soft-arg-max function similar to LIFT.

Consider a window w_i of size $N_s \times N_s$ in each score map. The score value at each coordinate $[u_i, v_i]^T$ within window w_i in the score map is exponentially scaled and normalized using spatial softmax function within that window as:

$$\widehat{w}_i(u, v) = \frac{e^{\beta \cdot w_i(u,v)}}{\sum_{j,k}^{N_s} e^{\beta \cdot w_i(j,k)}} \qquad \text{(Equation 3.3)}$$

where $\widehat{w}_i$ is window w_i with score value at each coordinate exponentially scaled & normalized and β is a hyper-parameter controlling the smoothness of the softmax function. Softmax function performs the exponential scaling and normalization consequently only maximum dominates. The expected location of the soft maximum within a window is computed as weighted average using the coordinates of the window which gives an approximation of the maximum coordinates extracted through non-maximum suppression as:

$$[x_i, y_i]^T = [\bar{u}, \bar{v}]^T = \frac{\sum_{j,k}^{N_s} e^{\beta \cdot w_i(j,k)} \cdot [j, k]^T}{\sum_{j,k}^{N_s} e^{\beta \cdot w_i(j,k)}} \qquad \text{(Equation 3.4)}$$

or this soft-arg-max function can be written as

$$[x_i, y_i]^T = [\bar{u}, \bar{v}]^T = \sum_{u_i, v_i}^{N_s} \Big[W \odot \widehat{w}_i(u_i, v_i), W^T \odot \widehat{w}_i(u_i, v_i) \Big]^T \qquad \text{(Equation 3.5)}$$

where W is $N_s \times N_s$ kernel and $\odot$ is element-wise product. This is soft-arg-max function which is similar to non-maximum suppression (NMS) with only difference that former is differentiable in context of the gradient based training. Soft-arg-max function gives the *expected* location of the global maximum in the window rather than exact location of the maximum. The base of the exponential function in the soft-arg-max function demands a ablation study to see the effect of changing the exponential base on overall performance on training.

A feature detector is said to be the *covariant* detector if the same features are detected irrespective of the underlying geometric or photometric transformation. For instance, let $\boldsymbol{x}$ be an image and let $T\boldsymbol{x}$ is the translated version of it with $T \in R^2$. A feature detector detects an affine feature $\boldsymbol{f}$. The feature detector is said to be *covariant* feature detector if it is applied to the translated image $T\boldsymbol{x}$, it detects the translated feature $\boldsymbol{f} + T$. DNet [48] formulates the complete covariant constraints for image patches and TCDet [26] extends these constraints for predicting the transformations.

Since the training strategy of MSK is based on siamese pipeline. Therefore, MSK takes the image pair (I_A, I_B), with known ground truth homography $H_{A,B}$, as input and outputs two corresponding score maps (S_A, S_B). (Symmetric) Covariant constraint loss $\mathcal{L}_{cc}$ between two score maps is computed as distance Equation 3.6, using l_2-norm, between the average coordinates of the soft local maximum within a window, extracted through soft-arg-max layer, and actual coordinates of the maximum, extracted through non-maximum suppression (NMS), within same window and warped using ground truth homography $H_{A,B}$.

$$\mathcal{L}_{cc}(S_A, S_B, H_{A,B}, N_s) = \sum_{i=1}^{n_s} \gamma_i \| [x_i, y_i]_A^T - H_{A,B}^{-1} [\hat{x_i}, \hat{y_i}]_B^T \|_2^2 \qquad \text{(Equation 3.6)}$$

and γ_i is the weighting parameter for ith window given as

$$\gamma_i = S_A(x_i, y_i)_A + S_B(\hat{x}_i, \hat{y}_i)_B$$

Where $i = 1, \ldots, n_s$ is index for non-overlapping windows in score maps, N_s is the size of non-overlapping window. The learned parameter γ_i controls the share of each distance computed between the coordinates of original and warped keypoint in each window. It gives high value for those keypoints which are more prominent and reduces the contribution of less dominant keypoints in all windows. It helps computing loss for significant features only. Since NMS is non-differentiable, therefore, the gradients are only computed for the original keypoints locations.

Since Soft-arg-max function operates on non-overlapping windows, therefor, it returns only one location per window. Hence, total number of keypoints, in the score map, strongly depends upon the size of this window N_s. Therefore, with increasing window size N_s only small number of dominant

keypoints are left in the score map. Whereas for small window size N_s, large number of keypoints are detected in the score map. Therefore, the size of this non-overlapping window acts as scale for the keypoints indirectly. In [63], authors have shown the improvements in the performance of detected local features when these features are not only accumulated within spatial window but also from the neighboring or adjacent scales as well. Therefore, we also extend our covariant constraint loss for multi-scale by employing the soft-arg-max function for different window sizes N_s and then accumulating the features from different scales. Varying window size N_s forces the network to detect strong keypoint across different scales.

We, therefore, propose Multi-Scale Covariant Constraint (MSCC) loss to incorporate the scale within loss function. We apply soft-arg-max function on each score map with varying window sizes N_s. Then we compute covariant constraint loss $\mathcal{L}_{cc}$ for all these window sizes N_s and finally, multi-scale covariant constraint (MSCC) loss is computed as weighted sum of these individual loss values, that is, our proposed MSCC loss function is the weighted average of covariant constraint losses from all individual scale levels, which is given as follows

$$\mathcal{L}_{MSCC}(S_A, S_B, H_{A,B}) = \sum_s \lambda_s \mathcal{L}_{cc}(S_A, S_B, H_{A,B}, N_s) \qquad \text{(Equation 3.7)}$$

where s is index for different scale levels, $\mathcal{L}_{cc}$ is covariant constraint loss for constant scale level s and window size N_s. The hyper parameter λ_s is control parameter for scale level s. It controls the contribution of individual covariant constraint losses towards final multi-scale covariant constraint (MSCC) loss $\mathcal{L}_{MSCC}$. The value of hyper parameter λ_s decreases proportionally with increasing value of scale level s and window size N_s, because the larger scale value s results in larger window size which in turn increases the individual covariant constraint loss. Therefore, this proportional decrease in the value of λ_s is similar to scale-space normalization [34].

3.3 Experimental Settings

In this section, we discuss all the experimental setting for training of our proposed approach *MSK*. We discuss the training & validation dataset for training and hyper parameter tuning of *MSK*. We also performs ablation study over the architecture of the *MSK* and details out two of its variants as well.

3.3.1 Training Dataset

In supervised learning, ground-truth data is needed to compute the loss and perform learning using gradient based methods. But in case of keypoint detection, ground-truth data is not available or expensive to generate using human labelling or annotation. Therefore, we train *MSK* in an unsupervised manner, without any true ground-truth data, with an objective to minimize the symmetric transfer error [20] 3.2.3.1.

There are numerous datasets [4] available for training and testing local feature detectors and descriptor heads. They have numerous different features and nature of available (pseudo) ground truths. However, we expostulate that a truly representative dataset must contain a wide range of photometeric and geometric transformation along with numerous different imaging conditions, viewpoint and temporal variations in imaging, partial occlusions etc. That is, the training dataset must contain wide-baseline images. Therefore, we use *PhotoTourism* [64,65] dataset for training of MSK. PhotoTourism dataset is a very large collection of images containing almost 30,000 images of landmark buildings and natural scenes from around the world. In general, this dataset consists of 25 classes or sequences of images. Number of images in each sequence are ranging from 75 to 4000 images. We have used five sequences [4] of PhotoTourism dataset for training of MSK i.e. *Brandenburg Gate*, *Buckingham Palace*, *Sacre Coeur*, *Tajmahal* and *Temple Nara Japan*, which total to 6,434 images.

For each image I_A in the Photo Tourism training dataset, we generate a training image pair (I_A, I_B) by applying the *random* geometric (perspective) 3.3.1.1 and photometric 3.3.1.2 transformation on the original image I_A and produce a transformed image I_B with known ground truth homography $H_{A,B}$. Therefore, for each image in the Photo Tourism, we synthetically generate a 3-tuple, $(I_A, I_B, H_{A,B})$, with an original image I_A, a transformed image I_B and the (known) ground truth homography $H_{A,B}$. Training pipeline of MSK is based on siamese training pipeline with weight sharing between its two instances. In siamese pipeline, the MSK takes an image pair (I_A, I_B) as input and generates two corresponding output score maps (S_A, S_B).

3.3.1.1 Geometric Transformations

For a given image I_A, we apply random geometric (perspective) transformation on the original image to generate transformed image I_B. We apply perspective transformation on the original image. For generating random perspective transformation, we employ random *uniform* sampling for rotation angle, scale and skew factor from $[-\pi/4, +\pi/4]$, $[0.5, 2.0]$ and $[-0.5, 0.5]$, respectively. After applying geometric transformation on the original image, we use center cropping to extract a central region from the original image & subsequently corresponding region from transformed image. If

the corresponding region of transformed image falls out of image boundaries, then a new geometric transformation is sampled. We also generate similar binary masks for original and transformed image to identify common regions in both images.

3.3.1.2 Photometric Transformations

For inducing the robustness against illumination changes in the images, we apply a robust set of photometric transformation on the original image I_A to generate transformed image I_B. We apply photometric transformation on the original image in both RGB-color space as well as in HSV-color space. In HSV-color space, we vary brightness, contrast and saturation of the original image randomly from a pre-defined range of values. Similar photometric transformation are applied in RGB-color space as well. We apply both geometric and photometric transformation on the (same) original image simultaneously to make the training set as hard as possible for the robust training. We also generate a similar synthetic validation dataset for the hyper-parameter tuning and computing the validation loss. Our validation consist of total 1074 images taken from Photo Tourism dataset.

3.3.2 Implementation Notes

Training pipeline of MSK is based on siamese pipeline with weight sharing between its two instances. In siamese pipeline, the MSK takes the image pair (I_A, I_B) as input and generates two corresponding output score maps (S_A, S_B). During training, weights of the both instances of MSK are updated simultaneously during each iteration.

In our implementation of MSK, we have used ResNet-50 [23], pre-trained on Image-Net [66] dataset, truncated after first block as our pre-trained base feature extractor. We have implemented our network, MSK, in PyTorch [67]. A sub-module of PyTorch, Torch-Vision, offers pre-trained weights for every provided architecture. Therefore, we have used Torch-Vision's implementation for ResNet-50 which provides the readily available pre-trained model along with its weights. The weights of the ResNet-50 are not updated during training & are kept frozen. The output of the pre-trained base feature extractor is a feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$. ResNet-50 contains Max-Pooling layers, due to this max pooling operation the spatial resolution of the output feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$ is $1/4th$ of the original image's resolution. In order to restore the original spatial resolution of the output feature volume , we use bi-linear interpolation in the Feature Transformation Block of the Information Accumulation Pyramid Module. In feature transformation block, we use two convolutional layers, each having filter size of 3×3, with number of output channels in each filter set to 64 in both layers, each layer is followed by Batch Normalization & ReLU function. We have also trained two variants

of MSK, discussed in 3.3.3, by changing the number of channels in the output feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$ and number of output channels of the convolutional layers in the Feature Transformation block.

In Information Accumulation Pyramid (IAP) module, we use $N = 5$ parallel convolutional layers to produce feature maps $\{F^n\}, 1 \leq n \leq N$ with increasing receptive fields to build an information accumulation pyramid. Then, receptive field pyramid is constructed using convolutional layers with increasing filter size i.e. $F^1 = 1 \times 1$, $F^2 = 3 \times 3$, $F^3 = 5 \times 5$, $F^4 = 7 \times 7$ and $F^5 = 9 \times 9$. Each convolutional layer in IAP is followed by the batch normalization & ReLU activation function and each layer has only one output channel. The output of these convolutional filters from Information Accumulation Pyramid is further fed to information change detection module, which computes the change in information between adjacent feature maps. Output from the change detection module is concatenated and fed to keypoint detection module & continuous scale estimation module to produce score and scale maps, respectively.

Each convolutional layer has been initialized with He-Kaiming weight initialization and $l2$-kernel regularization. A batch size of 16 with PyTorch built-in Adam Optimizer is used for training with a learning rate of 10^{-3} and decay factor of 0.5 after every 8 epochs. Each input image during training is resized to 300×300 keeping computational aspects in view. MSK and its variant converges on average in 40 epochs. We have trained MSK and its variants on a machine with an i7-8600k processor running at 3.60GHz and with NVIDIA GeForce GTX 1080-Ti GPUs.

We have used synthetic validation dataset for the hyper parameter tuning and computing validation loss. The hyper-parameters of the Multi-Scale Covariant Constraint (MSCC) loss are tuned using validation dataset. We compute multi-scale covariant constraint loss by dividing the each score map into $s = 7$ different window sizes with each window size $N_s \times N_s$, where $N_s \in \{5, 10, 15, 20, 25, 30\}$. The weighting parameter for each window size is, λ_s, where $\lambda_s \in \{32, 16, 8, 4, 2, 1\}$. As mentioned earlier, the larger the window size N_s has in turn larger corresponding covariant constraint loss for that window size. Therefore, the the weighting parameter for larger window size is proportionally smaller.

At the *inference time*, MSK produces score and scale map having same spatial resolution as of input image. We apply non-maximum suppression (NMS) in a 15×15 window on score map to generate a list of keypoints which are aggregated in decreasing order of their strength. We also make a gaussian image pyramid containing 7 levels (with one up-sampling and five down-sampling) for detecting and accumulating the keypoints from multi-scales.

3.3.3 MSK Variants

Output of the pre-trained base feature extractor, ResNet-50, is a feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$. Before feeding this feature volume to Information Accumulation Pyramid (IAP) module, bi-linear up-sampling is employed to restore the original spatial resolution of this feature volume. Since this feature volume has 256 channels, which makes it memory intensive to handle such large feature volume. Therefore, we have trained two variants of MSK by changing the number of channels in the output feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$ and number of the output channels of the convolutional layers in the feature transformation block.

Bottleneck convolutional layer having filter size of 1×1 is applied on feature volume $V \in \mathbb{R}^{H' \times W' \times 256}$. This convolutional layer has 256 input channels and only 64 output channels. Therefore, this bottleneck convolutional layer produces as output feature volume $V' \in \mathbb{R}^{H' \times W' \times 64}$, which has only 64 channels, thus reducing memory footprint of MSK. We call this variant of MSK as **MSK-Tiny**.In order to further lessen the memory footprint of MSK-Tiny, we further lower the number of output channels in first convolutional layer of feature transformation block to 32 and output channels of second convolutional layer to 16 channels. Thus, reducing the overall memory footprint of MSK. We name this version of MSK as **MSK-Tiny++**. This compression not only makes MSK computationally efficient but also increases the performance of these two variants which validates the accuracy of our main architecture.

Conclusions

In this chapter, we described our proposed approach in details. We started by discussing that how the natural inherited structure of the convolutional neural network learns better image priors and low-level representation directly from the imaging data. Moreover, how this learned representation assist in learning and distilling better local features from this pre-learned representation. Also, we demonstrated how can we combine the concept of information change with the varying receptive field to build a scale space which is further employed to extract the scale of the keypoints. Finally, we discuss the necessary implementation details of our network and how modifying its architecture generates its performance-wise superior variants.

Chapter 4

Experimental Evaluations & Results

In this chapter, we undertake a detailed experimental evaluation of our proposed approach i.e. Multi-Scale Keypoint (MSK) keypoint extractor and its variants using numerous challenging datasets and standard performance metrics. We start by computing the repeatability and matching score on numerous different datasets and compare the results with current state-of-the-art methods (SOTA). Secondly, we move to more challenging downstream task i.e. 3D reconstruction task and evaluate its performance on three small scale and three large scale dataset and details all the evaluation metrics. Finally, we describe the most recent results of our proposed approach on the ultimate task of the accuracy of reconstructed camera poses including numerous associated metrics on standard CVPR benchmark and challenge provided by [4]. Finally, we conclude the chapter by analyzing all the results and highlights the major findings of this section.

4.1 Evaluation Metrics

We have evaluated the performance of our *Multi-Scale Keypoint (MSK)* extractor and its variants, MSK-Tiny & MSK-Tiny++, on numerous different datasets using standard benchmarks for different intermediary as well as complex downstream tasks. Since MSK is a stand-alone keypoint detector without any descriptor head, therefore, we mainly evaluate its performance on repeatability and matching score, which are two standard metrics for assessing the performance of stand alone keypoint extractor. The main purpose of developing of a stand alone keypoint extractor is to use the powerful machinery of the deep neural networks to estimate the robust correspondences for subsequent use in camera pose estimation, structure from motion, point cloud generation and 3D reconstruction. Therefore, secondly, we evaluate the performance of MSK on ETH 3D Reconstruction benchmark [11]. We compare the performance of MSK with numerous state-of-the-art methods for the fair judgement of performance and robustness in practical conditions. Finally, we compute the *accuracy of reconstructed camera poses* using the keypoints extracted from the MSK on CVPR Image Matching Benchmark [4] and also participate in associated CVPR image matching challenge (IMC). MSK consistently exhibits robustness and performance stability across the all aforementioned metrics without any prior fine tuning of hyper parameters. In this chapter, we evaluate & discuss the performance of MSK on the following standard evaluation metrics along side the their datasets and evaluation settings:

- *Repeatability [8]*

- *Matching Score [4, 68]*
- *ETH 3D Reconstruction Benchmark [11]*
- *CVPR Image Matching Benchmark [4]*

4.2 Repeatability

Multi Scale Keypoint extractor (MSK) is a stand alone keypoint detector without any descriptor head. The performance of stand alone keypoint extractor is assessed in terms of *repeatability* [8, 35]. Repeatability measures the robustness of the extracted keypoints against the nuisance effects of the imaging conditions such as viewpoint and illumination variations. The repeatability score, for the detected keypoints, is defined as the ratio of features i.e. keypoints that *match* between images with a sufficient geometric overlap to minimum number of keypoints found within image regions after warping using the ground truth homography [8]. This evaluation procedure ensures that each keypoint is matched only once. For matching the keypoints, *Intersection-over-Union*, ε_{IoU} is computed between the supporting region of the original keypoint and warped keypoint using the ground truth homography. Conventionally, the performance of keypoint extractor and descriptor generator was collectively assessed. However, in order to get the valuable insights to the nature and working of the keypoint extractor and descriptor under practical conditions, it is usually important to evaluate theses individually. Hpatches [9] provide a very challenging and robust benchmark for the stand alone descriptors. However, the until recently there was no such standardized benchmark and datasets were available for the evaluation of stand alone keypoint extractor. Most recently, the work in [8] provides the solution to this problem. It does not only improve the standard definition of repeatability but also provides a standard benchmark with all datasets and evaluation strategies. Therefore, we also follow the standard evaluation protocol defined in [8], which provides a relatively robust and scale independent definition of repeatability as compared to previous works such as [13, 35]. Basically, it address the issue of bias from scaling factor that is applied to enhance the speed of computation of overlapping regions in case of multi-scale keypoints, by normalizing with a common scale factor. For all our evaluations, we use publicly available code[1] without any modifications per se. In our evaluations, we select the top 1024 keypoints that falls within the common regions of both images and call these repeatable iff the ε_{IoU} error is less than a preset threshold i.e. 0.5, i.e. a keypoint is considered repeatable if the geometric overlap between the corresponding supporting regions of the original and warped keypoint is at least 50%.

[1] `https://github.com/lenck/vlb-deteval`

4.2.1 Datasets for Repeatability

A large number of different datasets have been proposed in literature for the training and evaluation of local image features extractors [4]. All these datasets have different classes or sequences of images which test particular characteristics of local image features extractor. These datasets have a diverse collection of images including natural scenes & landmarks, buildings and real world objects etc. All of these datasets mostly represent the mild to extreme illumination and viewpoint changes in images. We have used total 5 publicly available datasets for the computation of repeatability score. We have used *Hpatches* [9], *Edge Foci* [14], *Webcam* [69], *Hannover* [15] and *VGGH* [13] datasets for evaluation of MSK on repeatability metric. These afore-mentioned datasets impart geometric and photometric transformations, however, these datasets are not capable of testing the scale and rotation invariance properties of the keypoint detectors. Therefore, We have also generated two variants of Hpatches [9] datasets, *Hpatches Rotation* and *Hpatches Scaling*. The main reason for generating these two variants of Hpacthes dataset is to create an opportunity to evaluate the performance of MSK and other keypoint detectors for translation, rotation and scale invariance. For *Hpatches Rotation* dataset, we have applied the rotation angles of [50°, 130°, 210°] on the reference image in each sequence of the Hpacthes dataset to generate the three rotated version of original image. Similarly, for *Hpatches Scaling* dataset, we have scaled the reference image in each sequence of the Hpatches dataset with a scale factor from [1.25, 1.50, 1.75] to generate its three scaled versions. We also evaluate MSK's repeatability on *Hpatches Rotation* and *Hpatches Scaling* dataset individually and discuss the results in analysis section.

Figure 4.1 shows first three images from only one randomly chosen sequence from all above mentioned datasets. It can be observed that how the geometric and photometric transformations intensify with each image in each sequence with first transformed image labeled as easiest and last image labeled as hardest. Table 4.1 shows important aspects of all datasets such as nature of ground truth, number of sequences and total images in above mentioned datasets. We have discussed the details of all the aforementioned datasets individually below as well.

4.2.1.1 Hpatches

Hpatches[2] [9] dataset is a state-of-the-art dataset available as benchmark for evaluation of local image features extractions method. The Hpatches dataset contains 696 images which are distributed in total 116 sequences. These 116 sequences are divided into two parts, 57 sequences have photometric (Illumination) transformation only and remaining 59 sequences have geometric (viewpoint) transfor-

[2] https://github.com/hpatches/hpatches-dataset

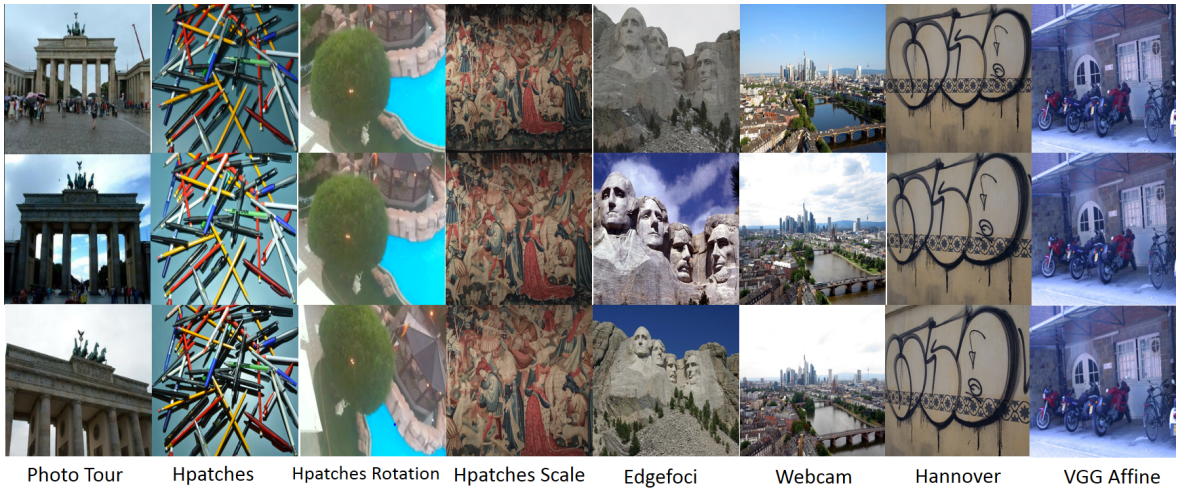


Figure 4.1:: Instances of datasets used in computation of Repeatability [8] ad Matching Score [4]. First three images from one randomly selected sequence from all datasets, i.e. *Phototour* [64, 65], *Hpatches* [9], *Hpatches rotation*, *Hpatches scaling*, *Edge Foci* [14], *Webcam* [69], *Hannover* [15] and *VGGH* [13], which are used for computing repeatability. We have used only Hpatches datasets for computing matching score.

Dataset	**Nature of Ground Truth**	**# Sequences**	**# Images**
Photo-tourism Test [64, 65]	Corresponding Images	26	29,707
Hpatches [9]	Corresponding Images, Homographies	116	696
Hpatches Rotation	Corresponding Images, Homographies	116	812
Hpatches Scaling	Corresponding Images, Homographies	116	580
Edge Foci [14]	Image Sequences	13	86
Webcam [69]	Image Sequences	6	250
Hannover [15]	Image Sequences, Homographies	8	48
VGGH [13]	Homographies	8	48

Table 4.1:: Important details of all the datasets used in Repeatability and Matching Score Computation are outlined here. This tables shows the nature of ground truth, number of sequences and number of images in each dataset. We use Photo-Tourism dataset for training of MSK and its variants.

mations alone. In each sequence, there are 6 images, one reference image and five transformed images with corresponding ground truth homorpagihes already provided between reference image and each transformed image. We have used Hpatches dataset for computing both repeatability and matching score for MSK and its variants.

4.2.1.2 Hpatches Rotation & Scaling

These Hpatches dataset provides both geometric and photometric transformations, but it is not capable of testing the scale and rotation invariance properties of a keypoint detectors robustly. Therefore, We have also derived two variants of Hpatches [9] datasets, *Hpatches Rotation* and *Hpatches Scaling*. The main reason for generating these two variants of Hpacthes dataset to create an opportunity to evaluate the performance of MSK and other keypoint detectors for translation, rotation and scale invariance. For *Hpatches Rotation* dataset, we have applied the rotation angles of $[50°, 130°, 210°]$ on the reference image in each sequence of the Hpacthes dataset to generate the three rotated images. Similarly, for *Hpatches Scaling* dataset, we have scaled the reference image in each sequence of the Hpatches dataset with a scale factor from $[1.25, 1.50, 1.75]$ to generate its three scaled versions. We also evaluate repeatability on *Hpatches Rotation* and *Hpatches Scaling* dataset individually and discuss the results in analysis section.

4.2.1.3 Edgefoci

Edgefoci[3] [14] dataset is used in [14] for evaluating edge foci of interest points in images. Edgefoci dataset consists of 13 sequences, where each sequence consists of color as well as gray scale images. Number of images in each sequence are not equal. Number of images in each sequence are varying from 6 to 9 images.

4.2.1.4 Webcam

Webcam[4] [69] dataset is derived from the DTU robot dataset for only testing the illumination changes. Webcam only consists of the illumination changes. Webcam dataset contains images captured using outdoor Webcams, hence the name. The images in Webcam dataset represents the temporal and seasonal variations in imaging conditions. This dataset also contains some panoramic images. This dataset consists of following sequences: *Chamonix*, *Courbevoi*, *Frankfurt*, *Mexico*, *St.Louis*. We have used Webcam [69] for computing repeatability only.

[3] http://www.robots.ox.ac.uk/~vgg/research/affine/

[4] https://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/keypoints/WebcamRelease.tar.gz

4.2.1.5 Oxford Affine

Oxford Affine[5] [13] is also known as VGG Affine dataset. (In our results, we have denoted this as VGGH dataset.) This dataset consists of 8 sequences with total 48 images in dataset. There is one reference image in each sequence and 5 transformed images. There are five different type of variations in imaging conditions: viewpoint variations, illumination variations, scale variations, JPEG compression and image blurring. Each sequence represents variations in one of these aforementioned imaging conditions. The homography matrix between reference image and each transformed image in a particular sequence is computed. We used Oxford Affine [13] dataset for computing repeatability only.

4.2.1.6 Hannover

Hannover[6] [15] is another dataset which represents change of different imaging conditions like viewpoint change, texture and pattern repetition. This dataset also consists of 7 sequences with total 42 images in dataset. The datasets contain image sequences in different resolutions and homographies for the accurate mapping between image pairs of a sequence. Hannover [15] dataset is mostly employed to evaluate the performance of different local image feature extractors in [8]. We use Hannover dataset for computing the repeatability only.

4.2.2 Analysis & Discussion on Repeatability Results

We compare the repeatability results of MSK and its variants with both learned and handcrafted keypoint extractors. We have computed repeatability on all the aforementioned datasets for the current state-of-the-art methods in both learned and handcrafted streams of keypoint detectors. Complete repeatability results on all the datasets and for all SOTA detectors are given in table 4.2. For the sake of reproducibility of results, we have computed repeatability using standard benchmark presented and improved in [8]. We have used publicly available code[7] for the benchmark in [8]. For sake of simplicity and fair comparison, we have divided the repeatability results in two main categories depending upon the nature of keypoints extracted by a particular keypoint detector. We have divided the keypoint detectors in i) Translation-Invariant **(TI)** detectors and ii) Scale-Invariant **(SI)** detectors.

[5] http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/

[6] http://www.tnt.uni-hannover.de/project/feature_evaluation/

[7] https://github.com/lenck/vlb-deteval

Translation-Invariant (TI) detectors do not include a mechanism to predict a scale for each keypoint. The supporting region of keypoints for these keypoint detector is considered a fix number for computing repeatability. Whereas, Scale-Invariant (SI) keypoint detectors do predict a unique scale for each keypoint. These detectors include a method for predicting a unique scale value for each keypoint like SIFT [21]. For fair comparison and reproducibility of the results, we use author's implementation, for learned keypoint detectors and matching pipelines, i.e. SuperPoint[8] [47], LF-Net[9] [27], LIFT[10] [28], RF-Net[11] [52], KeyNet[12] [6], TILDE[13] [12], TCDET[14] [12], D-Net[15] [48]. Whereas for handcrafted keypoint detectors, i.e. for FAST [40], BRISK [44] and SURF [36], we use the implementation provided by VLFeat MATLAB[16] for uniformity. Top 1024 features are selected from each image while computing repeatability score.

For Translation-Invariant (TI) detectors, MSK-TI exhibits best repeatability score on Hpatches (Illumination & Viewpoint), WebCam, VGGH, Hannover datasets and performs 2nd best on EdgeFoci dataset. Moreover, the mean repeatability score of MSK-TI on all these datasets is best. This indicates that MSK predicts stable keypoints across varying imaging conditions. For Scale-Invariant (SI) detectors, we have reported the results of all three variants of MSK i.e. MSK-SI, MSK-Tiny-SI and MSK-Tiny++-SI. On average repeatability score, MSK-SI exhibits best repeatability score, TCDET-SI performs 2nd best overall and MSK-Tiny-SI exhibits 3rd best score overall. On individual datasets, MSK-SI performs consistently best for repeatability score. Whereas TCDET-SI and D-Net-SI perform good repeatability score on individual datasets. The main reason of good performance of D-Net-SI is that it trans a regressor using loss function similar to our covariant constraint loss function. Repeatability score of MSK-Tiny++-SI is not up to the mark as of the MSK-SI and MSK-Tiny-SI. We do not report the repeatability score of Rf-Net for complete Hpatches dataset, because RF-Net uses Hpatches dataset for training purposes.

We have also reported the repeatability results for Scale-Invariant (SI) detectors on Hpatches Rotation and Hpacthes Scaling datasets in table 4.3. We do not report the repeatability results for the Translation-Invariant (TI) detectors on these datasets, because Translation-Invariant detectors do not exhibit the property of scale invariance. Therefore, it does not make sense to report these results

[8] https://github.com/MagicLeapResearch/SuperPointPretrainedNetwork

[9] https://github.com/vcg-uvic/lf-net-release

[10] https://github.com/cvlab-epfl/LIFT

[11] https://github.com/Xylon-Sean/rfnet

[12] https://github.com/axelBarroso/Key.Net

[13] https://github.com/cvlab-epfl/TILDE

[14] https://github.com/ColumbiaDVMM

[15] https://github.com/lenck/ddet

[16] https://www.vlfeat.org/

merely for Hpacthes rotation dataset. These two datasets are especially created to test the rotation and scale invariance properties of keypoint detectors. MSK-SI performs best on average repeatability score for Hpatches rotation dataset. Whereas MSK-SI performs 2nd best on average repeatability for Hpatches Scaling dataset. Surprisingly, on average, BRISK-SI (Binary Robust Invariant Scalable Keypoints) performs best for Hpatches scaling dataset, which may be due its ability to predict scalable keypoints, hence the name. On average, the performance of MSK and its variants on repeatability metric on all datasets is best with few exceptions.

Detector	**HP-Illum**	**HP-View**	**Webcam**	**VGGH**	**Edge-Foci**	**HannOver**	**Mean**
FAST-TI [40]	66.4	42.3	60.9	57.3	46.7	**43.1**	52.8
DNet-TI [48]	71.9	**45.3**	68.3	**58.3**	59.4	42.8	57.7
TILDE-TI [12]	**77.5**	**45.8**	**81.0**	**62.1**	**72.6**	**44.0**	**63.8**
SuperPoint-TI [47]	**77.2**	44.4	**78.1**	57.7	**63.7**	**43.1**	**60.7**
MSK-TI	**83.3**	**48.8**	**85.8**	**63.3**	**65.7**	**47.2**	**65.7**
BRISK-SI [44]	58.4	29.8	63.8	38.5	52.0	22.5	44.2
SURF-SI [36]	61.7	56.7	58.1	63.7	45.3	53.4	56.5
DNet-SI [48]	64.6	64.4	56.7	**71.9**	44.6	**58.7**	60.2
TCDET-SI [26]	**74.0**	63.6	73.7	**71.6**	59.6	54.7	**66.2**
LIFT-SI [28]	60.3	53.9	59.2	55.9	46.0	43.3	53.1
LF-Net-SI [27]	**72.7**	41.7	**75.9**	57.1	62.6	42.8	58.8
RF-Net-SI [52]	-	-	73.9	56.8	59.3	29.8	54.9
KeyNet-SI [6]	70.5	61.7	**75.4**	64.8	51.6	55.5	63.3
MSK-SI	**76.3**	**66.6**	**76.2**	**69.2**	**74.7**	**56.9**	**69.9**
MSK-Tiny-SI	68.6	**65.7**	68.7	63.8	**66.4**	56.1	**64.9**
MSK-Tiny++-SI	67.5	**65.9**	67.5	65.8	**65.0**	**57.1**	64.8

Table 4.2:: **Repeatability Score** (%) results and comparison for Scale-Invariant (SI) & Translation-Invariant (TI) keypoint extractors on HPatches [9], Webcam [12], VGG-Affine (denoted as VGGH) [13], EdgeFoci [14], and Hannover [15] datasets. For fair comparison and uniformity of results, we re-evaluate the repeatability score from scratch for all keypoint detectors on standard evaluation protocol provided in [8]. We have employed the author's implementation for all off-the-shelf detectors. We have computed repeatability score on full resolution images without any re-sizing. Only top 1024 keypoints are selected for computing repeatability in each image. Rf-Net uses Hpatches dataset for training purposes, hence, repeatability results of RF-Net are not discussed for Hpatches. Top 3 repeatability scores in each column are highlighted as red, green and blue, respectively. MSK-TI & MSK-SI outperform all other Translation-Invariant and Scale-Invariant detectors on individual and mean repeatability scores, respectively.

Detector	50°	130°	210°	Mean	1.25	1.50	1.75	Mean
BRISK-SI	74.4	74.1	76.5	75.0	88.4	91.5	93.1	**91.0**
SURF-SI	66.7	66.6	70.6	68.0	77.4	75.4	71.4	74.7
TCDET-SI	80.0	73.2	75.0	**76.1**	78.6	83.6	71.6	**77.9**
LFNet-SI	76.1	73.4	76.2	**75.2**	85.7	17.4	0.90	35.3
KeyNet-SI	69.8	60.8	62.6	64.4	76.2	78.9	68.6	74.6
MSK-SI	87.3	69.9	72.1	**76.4**	88.4	87.9	87.8	**88.1**

Table 4.3:: Repeatability score results and comparison on Hpatches Scaling and Hpatches Rotation Datasets for only Scale-Invariant (SI) keypoint detectors. Top 3 repeatability scores in each column are highlighted as red, green and blue, respectively.

4.3 Matching Score

Since local features consist of two components i.e. a local keypoint or interest point and a local descriptor. The overall performance of the image matching and correspondence estimation strongly depends upon on both the saliency of local keypoint as well as the discriminative power of the local descriptor. However, in sparse methods of image matching that use local features (a combination of keypoint and descriptor), it is hard to differentiate that which component of the local features play a crucial part in the final results of matching and correspondence estimation. The empirical and experimental studies show that both components play almost equal role in determining the quality of robustness of the final correspondences [4]. A good descriptor head with a non-discriminative set of keypoints will result in bad correspondences and vice versa. Matching Score and Repeatability are two fundamental performance assessing metrics [35] for any keypoint extractor and resultant matching. Therefore, we also report matching score on Hpatches [9] dataset 4.2.1.1. Although the final results of the matching score also depends upon the the distinctness and discriminative power of the underlying descriptor head used during matching. However, for uniformity of results and fair comparison between the quality of the matching, we fixed the descriptor head for all the stand alone keypoint detectors and study the performance of all the stand alone keypoint detectors on the matching task on the Hpatches dataset. Recently proposed stand alone descriptor Hard-Net [16] and Twin-Net [45] are considered the two state-of-the-art learned local descriptors. Therefore, we report matching score with the Hard-Net and Twin-Net descriptors, separately, with keypoints extracted from MSK and other keypoint extractors. Fixing the descriptor head gives us the opportunity for the fair comparison of the matching score without considering the biasness of the local descriptor head towards some keypoints extractors. Moreover, fixing the descriptor head gives us a opportunity to study the true effect of underlying keypoint extractor on image matching and correspondence estimation.

For computing matching score, we adopt same procedure as described in [4]. We only compute matching score for Hpacthes dataset as in KeyNet [6]. In Hpacthes dataset, each image sequence comprises of one reference or train image and five transformed or query images. The ground-truth homography between each reference image and transformed image is also given. We select top 1024 features for computing matching score for all detectors. We project the keypoints from train image to query image using ground truth homography and vice versa. After that, we use brute-force matching for finding nearest neighbors of descriptors from the source image to query image to create a set of putative matches. We repeat procedure for descriptors in query image as well. After finding the nearest neighbor of each descriptor in source image to query image and vice versa. We apply filter that selects only those matches which are one-to-one in both set of descriptors. At this stage, we have set of *tentative inlier matches*, which, in literature, are referred as one-to-one, bipartite or cycle consistent or mutual nearest neighbor matches. Finally, this set of tentative inlier matches is further filtered using *pixel thresholding*. We use pixel thresholding of $5 - pixels$. The ratio of number inlier matches left after pixel thresholding to total number of features used is matching score for an image pair, which is averaged over all pairs in the whole dataset.

4.3.1 Dataset for Matching Score

In order to evaluate the performance of MSK and its variants on image matching task, we compute matching score, which is a direct measure of the quality of established correspondences for image matching. In order to compute matching score, we have used the Hpatches dataset only. Main reason of evaluating the matching score only for Hpacthes dataset is that the nature of ground-truth provided by the Hpatches dataset assists to verify the quality of the matching. We report matching score for both Hpatches illumination and viewpoint sequences. Details about Hpatches dataset is give in section 4.2.1.1.

4.3.2 Analysis & Discussion on Matching Score Results

For evaluating the performance of extracted keypoints on image matching, we compute matching score. Table 4.4 shows the results of matching score on Hpacthes dataset. For unbiased comparison, we report matching score with 128-dimensional Hard-Net [16] descriptor head. We have used authors' provided publicly available implementation of Hard-Net[17]. We discard orientation for those matching pipelines which also predict the orientation for the keypoints. However, we have also re-

[17] https://github.com/DagnyT/hardnet

Detector & Descriptor	Matching-Score		
	HP-V	HP-I	Mean
MSER [41] + Hard-Net [16]	11.7	18.8	15.3
SIFT [21] + Hard-Net [16]	23.2	24.8	24.0
HarrisLaplace [32] + Hard-Net [16]	30.0	31.7	30.9
AKAZE [38] + Hard-Net [16]	36.4	41.4	38.9
TILDE-TI [12] + Hard-Net [16]	32.3	39.3	35.8
LIFT-SI [28] + Hard-Net [16]	30.3	32.8	31.6
DNet-SI [48] + Hard-Net [16]	33.5	34.7	34.1
TCDET-SI [26] + Hard-Net [16]	27.6	36.3	31.9
SuperPoint-TI [47] + Hard-Net [16]	37.4	43.0	40.2
LF-Net-SI [27] + Hard-Net [16]	26.9	**43.8**	35.4
KeyNet-SI [6] + Hard-Net [16]	38.4	39.7	39.1
LIFT-SI [28]	21.8	26.5	24.2
SuperPoint-TI [47]	38.0	41.5	39.8
LF-Net-SI [27]	23.0	29.1	26.1
MSK-SI + Hard-Net [16]	**42.8**	41.9	**42.4**
MSK-Tiny-SI + Hard-Net [16]	**44.4**	**44.4**	**44.4**
MSK-Tiny++-SI + Hard-Net [16]	**46.3**	**43.6**	**45.0**

Table 4.4:: **Matching Score on Hpatches Dataset with Hard-Net Descriptor**. We assess the performance of our proposed approach on image matching task. We follow the evaluation criterion of [11] as standard benchmark for computing the matching score under known ground truth homography for Hpatches dataset. For uniformity and fair comparison of the results of image matching, we use Hard-Net [16] as our fixed descriptor head. We generate 128-dimensional floating point descriptors using Hard-Net as our descriptor generator. We implement one-to-one matching with pixel threshold set to $5 - pixels$ for computing matching score. However, we also report the matching score for complete learned pipelines such as LIFT [28], SuperPoint [47] and LF-Net [27]. Top 3 scores in each column are highlighted as red, green and blue, respectively. MSK and its variants outperform all other SOTA keypoint extractors.

ported matching score for LIFT, SuperPoint and LF-Net with their own descriptor heads as these are complete matching pipelines. Similar to repeatability results, we have employed original authors' implementation for all the learned and handcrafted keypoints extractors, as described in 4.2.2. We have reported the matching score for Hpacthes illumination & viewpoint as well as mean score. Overall MSK and its variants (MSK-Tiny & MSK-Tiny++) outperform all other keypoint detectors and joint matching pipelines. On Hpacthes illumination, MSK-Tiny++, MSK-Tiny and MSK exhibits 1st,

2nd and 3rd best matching score overall, respectively. Similarly, on Hpacthes viewpoint, MSK-Tiny and MSK-Tiny++ produces 1st and 3rd best score, respectively, while LF-Net with Hard-Net produces 2nd best score. But on average matching score, MSK and its variants outperforms all state of art detectors. Figure 4.2 shows the final inlier matches left after pixel thresholding for MSK-Tiny++, KeyNet and SuperPoint along with Hard-Net descriptor on few randomly selected images from Hpatches illumination and viewpoint sequences. Clearly, the performance of MSK-tiny++ is consistent and far better than KeyNet and SuperPoint across all sequences. LIFT, SuperPoint and LF-Net are jointly optimized pipelines for image matching. However, their performance, even along with their own descriptor heads, is not up to the mark which highlights the weakness in their training mechanism.

4.4 3D Reconstruction

Matching Score and Repeatability are two rudimentary metrics for measuring the performance of the stand alone keypoint detectors and descriptor heads. Since problem of image matching and correspondence estimation is an intermediary and sub-problem of many core computer vision problems like camera pose estimation, structure from motion (SfM) and Multi-view stereo etc. Therefore, it is not always necessary that the performance gains in repeatability and image matching, i.e. correspondence estimation, will linearly translate to the subsequent stages in the aforementioned applications. Therefore, recently the focus on evaluating of different methods of keypoint detection, description, camera pose estimation and robust image matching techniques has moved from the intermediary metrics (repeatability or matching score) to downstream tasks like structure from motion, accuracy of the reconstructed camera poses and 3D reconstruction [3, 4, 11, 70].

Since the main purpose of developing a keypoint detector is to establish the robust correspondences in stereo and multi-view scenarios and use these correspondences to accurately reconstruct the camera poses and subsequently improve the performance on structure from motion and 3D reconstruction tasks. Therefore, we also focus to evaluate the performance of keypoints extracted from the MSK on ETH 3D reconstruction benchmark [5]. This is a well known benchmark which employs the COLMAP [10] pipeline on backend. This benchmark incorporates the sparse and dense reconstruction techniques i.e. structure from motion (SfM) [2] and Multi-view stereo [5] into a single unified pipeline.

For evaluating the performance of MSK, we wield the 3D Reconstruction benchmark of [11], which quantifies the quality of 3D reconstruction using multiple metrics on different stages of reconstruction pipeline such as *# registered images, # sparse points, # observations, mean track length, re-projection error, # inlier pairs, # inliers matches, # Dense Points*. [11] uses COLMAP's pipeline at backend for

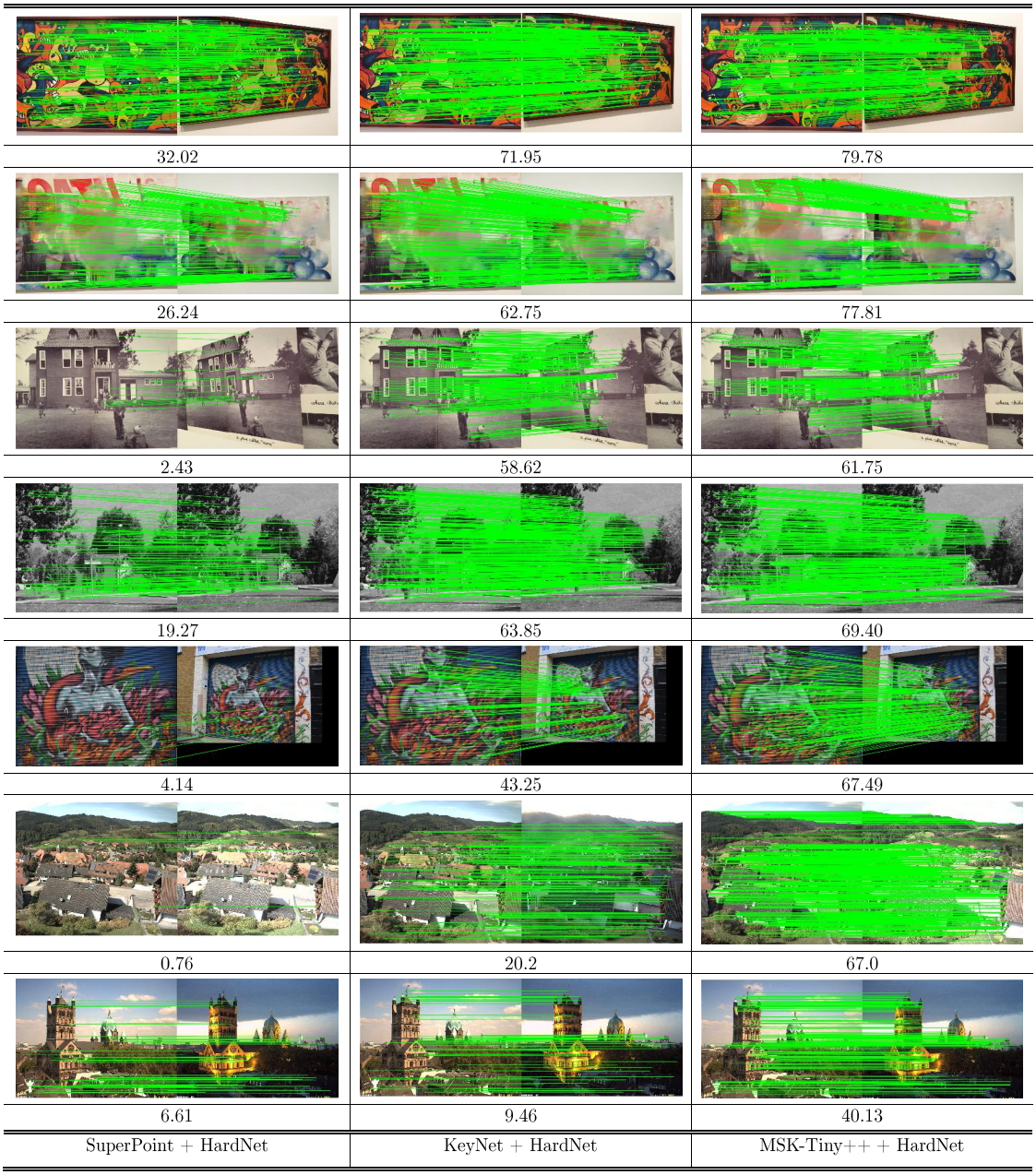


Figure 4.2:: Results of Image Matching on few randomly selected image pairs from Hpatches [9] dataset showing the final matches left after pixel thresholding. These image pairs represent the illumination and viewpoint variations for different scenes. For each keypoint detector, Hard-net [16] descriptor is used. Descriptors are matched using bi-directional, one-to-one matching [4,5] with pixel threshold of $5 - pixels$ is used to eliminate the outliers. Clearly, MSK-Tiny++ outperform the state-of-the-art detectors on almost all image pairs.

sparse and dense reconstruction.

COLMAP takes keypoint-locations (x, y) and corresponding descriptors for each image of the se-

quence in the dataset as inputs. We select top 5000 keypoints from each image of benchmark dataset for 3D reconstruction evaluation. Similarly to matching score computation, for uniformity and fair comparison of the state with the art stand alone keypoint detectors, we fix the descriptor head for all detectors. Descriptors are constructed using Hard-Net [16] for all of keypoint extractors as described in 4.3.2. For the sake of reproducibility of results and fair comparison, we re-evaluate all the keypoint detectors, with Hard-Net descriptors, under identical conditions to produce results on the aforementioned benchmark.

4.4.1 Datasets for 3D Reconstruction

For ETH 3D reconstruction benchmark [5], we use *Fountain, Herzjesu, South Building, Madrid Metropolis, Gendarmenmarkt and Tower of London* sequences as our dataset for this evaluation, as proposed in the aforementioned benchmark. These sequences of datasets have been collected from various existing benchmarks [71–73]. Sequences, *Fountain and Herzjesu*, are taken from famous MVS benchmark dataset Strecha [71]. Fountain and Herzjesu sequences contains 11 and 8 high resolution images, respectively, along with very accurate ground truth depth maps and underlying camera poses. South Building [72] sequences consists of 128 high-resolution images of a building, captured using a calibrated camera rig. All the images in this sequence are highly overlapping with high co-visibility ratio [4]. Fountain, Herzjesu and South Building are three small sequences with small number of images. We have also evaluated the said benchmark on three large sequences of images. *Madrid Metropolis, Gendarmenmarkt & Tower of London* are three extremely large sequences of images already used in [73] for 1-D structure from motion. These large sequences of images are actually extremely large and diverse collections of Internet images. These collections of internet images have a very diverse set of image features and are very challenging due to very high input data variance. Each sequence in Madrid Metropolis, Gendarmenmarkt and Tower of London contains 1344, 1463 and 1576 images, respectively, collected from *Flickr* website. These large collections of internet images have mixed with a set of distractor images to make them more challenging. These large collections of images, i.e. sequences, exhibit extreme illumination & viewpoint variations along with repetitive structures, image compression and distortion artifacts as well as distractor images embedded into them. Due to such heterogeneity of the these sequences, local feature extractor must generalize well to adapt such large variance of underlying scenes.

4.4.2 Evaluation Metrics for 3D Reconstruction

Structure from motion (SfM) is an iterative image based reconstruction pipeline. In SfM, local features extraction and robust correspondence estimation is first step to create a graph of matched fea-

tures in multiple views. Therefore, the performance of the all subsequent stages strongly depends upon the quality of these correspondences and error accumulated at this initial stage. Therefore, to verify the performance gains of these local features in subsequent stages of SfM pipeline, i.e. feature matching, geometric parameters estimation, camera pose reconstruction, image retrieval and sparse as well as dense modelling, we use numerous performance metrics. Any image based 3D reconstruction pipeline starts by using SfM to accurately estimate the camera poses of the input images, which are eventually used in inferring sparse model of the underlying scene. Afterwards, the output of this SfM stage serve as input to the multi-view stereo (MVS) step, which eventually results in obtaining the dense reconstruction i.e. in the form of dense point, depth maps, mesh surface models. The final goal of the this SfM and MVS is to generally produce a high quality dense 3D model of the underlying scene. Since local features extraction and matching is the first and foremost step in SfM, therefore, the overall quality of SfM strongly depends upon the two-view (stereo) correspondences. The quality of output of SfM is a good measure of the performance of the local features matching in initial stages. In addition to that, by combining the stereo correspondences into a graphs of matched features tracks in multiple views, SfM can use the multi-view redundancy of tracked features to verify the validity and reliability of the estimated correspondences.

In order to get the practical insights into the performance of the local features on the quality and accuracy of the multi-view based 3D reconstruction results, we determine a number of important metrics to evaluate the overall performance, which are as follow:

1. **# of Registered Images [11]**: Number of registered images during sparse reconstruction quantifies the *completeness* of the sparse modelling. More the number of registered images results in a more complete & dense MVS reconstruction and a larger number of 3D points with numerous image observations compose a more thorough and accurate representation of underlying scene.

2. **# Sparse Points [11]** Number of sparse points are total number of points resulted from the SfM. It also indicates the completeness of reconstruction. A larger number of sparse points means more complete MVS reconstruction and a larger number of 3D points.

3. **# Observations [11]** Number of observations per image is total number of geometrically verified (triangulation) projections of sparse points on the images.

4. **Mean Track length [11]** A track is a sequence of image positions of a feature (usually an image point) in subsequent images. Track length is the number of frames that feature is successfully tracked, from the frame the feature is detected to the one it is lost, e.g. due to occlusion, blurring, going out of the field of view, etc. Simply, the track length for each sparse point is the number of verified image observations. Mean track length and number of observations per image are two crucial metrics for an accurate cameras calibration and more reliable triangulation, because these two provide redundancy in the estimation.

5. **Re-Projection Error [11, 20]** Minimizing Re-projection error is main objective of the bundle adjustment. Bundle adjustment is combined adjustment of camera poses alongside the 3D points for minimizing the (symmetric) geometric or algebraic re-projection error. Bundle adjustment is a core problem of SfM. Bundle adjustment is about solving a non-linear joint optimization of the reconstructed camera poses along with 3-D points. The final re-projection error in bundle adjustment stipulates the accuracy of final reconstruction. It strongly depends upon the redundancy and accuracy of the input data, which, in turn, depends upon the completeness of underlying graph of established correspondences and keypoint localization accuracy.

6. **# Inlier Matches [11]** Number of inlier matches is total number of inliers found after geometric verification in, all image pairs, using robust model estimator.

7. **# Inlier Pairs [11]** In brute force matching, there are $N(N-1)/2$ image pairs for N images. Inlier pairs are those image pair in which at a fixed number of inliers are found after geometric verification.

8. **# Dense points [11]** Number of dense point indicates the number of 3D points after dense reconstruction using MVS. We compute the number of reconstructed dense points as a exclusive measure of the overall completeness of dense model of underlying scene and the accuracy of the SFM results.

4.4.3 Analysis & Discussion on 3D Reconstructions Results

Table 4.5 shows the results of 3D Reconstruction evaluation for few keypoint detectors along with Hard-Net descriptors on ETH 3D reconstruction benchmark dataset for three *small datasets* i.e. *Fountain, Herzjesu, South Building* and three *large datasets* i.e. *Madrid Metropolis, Gendarmenmarkt, Tower of London* sequences. In order to get the practical insights into the performance of the local features (keypoints and descriptors) on the quality and accuracy of the multi-view based 3D reconstruction results, we determine a number of key metrics such as *# registered images, # sparse points, # observations, mean track length, re-projection error, # inlier pairs, # inliers matches, # Dense Points* to evaluate the overall performance. ETH 3D reconstruction evaluation is very computationally intensive. Therefore, we have only evaluated the performance of the keypoints extracted from SIFT [21], TCDET [26], SuperPoint [47], KeyNet [6] and MSK-Tiny++. Evaluating the performance of almost all keypoint detectors is computationally and temporally impractical. Below we have detailed the performance of all keypoint extractors on the afore-mentioned performance metrics, separately.

1. **# of Registered Images [11]**: Number of registered images quantifies the *completeness* of the sparse model. On smaller datasets *(Fountain, Herzjesu, South Building)*, due to narrow

Datasets	Detector	# Images	# Registered Images	# Sparse points	# Observations	Mean Track length	Re-projection Error	# Inlier Pairs	# Inlier Matches	# Dense points
Fountain	**SIFT + Hard-Net**	11	11	5210	20234	3.88	0.2849	49	33530	293965
	TCDET + Hard-Net	11	11	1229	4952	4.03	0.7821	43	6678	334457
	SuperPoint + Hard-Net	11	11	6030	30059	4.98	0.7687	54	56160	296558
	KeyNet + Hard-Net	11	11	5067	28560	5.64	0.6018	53	45851	295598
	MSK-Tiny++ + Hard-Net	11	11	4492	24713	5.50	0.8476	54	41360	303990
Herzjesu	**SIFT + Hard-Net**	8	8	2857	10410	3.64	0.3222	26	15843	239103
	TCDET + Hard-Net	8	8	875	3157	3.61	0.7674	23	4355	235749
	SuperPoint + Hard-Net	8	8	4305	19222	4.47	0.6903	28	32921	242154
	KeyNet + Hard-Net	8	8	3166	13488	4.26	0.6046	28	18937	241962
	MSK-Tiny++ + Hard-Net	8	8	4266	17088	4.01	0.8454	28	25277	236026
South Building	**SIFT + Hard-Net**	128	128	37097	168749	4.54	0.4062	1419	365885	1997258
	TCDET + Hard-Net	128	62	6991	36220	5.18	0.8903	843	77920	986892
	SuperPoint + Hard-Net	128	128	47947	339807	7.09	0.6198	2760	1196878	2102471
	KeyNet + Hard-Net	128	128	31865	352915	6.67	0.6070	1962	1284841	2090784
	MSK-Tiny++ + Hard-Net	128	128	39100	268659	6.87	0.9156	2687	843206	2132983
Madrid Metropolis	**SIFT + Hard-Net**	1344	381	46723	304770	6.53	0.9753	12998	1096540	1270972
	TCDET + Hard-Net	1344	71	3062	14427	4.71	1.1229	1545	76920	366878
	SuperPoint + Hard-Net	1344	433	27359	257652	9.45	0.9131	12355	1231630	1612931
	KeyNet + Hard-Net	1344	319	23586	200516	8.50	0.9303	9376	800049	1198936
	MSK-Tiny++ + Hard-Net	1344	485	50945	438320	8.60	1.1012	23916	2297561	1693308
Gendarmenmarkt	**SIFT + Hard-Net**	1463	913	126254	716115	5.67	0.9687	40485	2442305	2793276
	TCDET + Hard-Net	1463	19	1413	6176	4.37	0.9321	3041	157284	114064
	SuperPoint + Hard-Net	1463	909	70115	582805	8.31	0.9409	41545	3245128	3649940
	KeyNet + Hard-Net	1463	901	84612	586714	6.93	0.9508	37664	2464561	3452597
	MSK-Tiny++ + Hard-Net	1463	949	129361	997401	7.71	1.0937	68124	5389160	3603918
Tower of London	**SIFT + Hard-Net**	1576	554	89083	528642	5.93	0.8671	17275	1616458	2438655
	TCDET + Hard-Net	1576	80	2584	15237	5.90	1.0649	1782	193530	497594
	SuperPoint + Hard-Net	1576	671	49040	445688	9.08	0.8678	18801	2024741	2773906
	KeyNet + Hard-Net	1576	563	55939	529949	9.47	0.8365	17114	1937675	2524027
	MSK-Tiny++ + Hard-Net	1576	710	72267	681773	9.43	1.0591	31328	3249820	2704831

Table 4.5:: **Comparative 3D Reconstruction Evaluation:** Comparative 3D reconstruction evaluation and analysis using keypoints extracted from different keypoint detectors with fixed Hard-Net [16] descriptor head. ETH 3D reconstruction benchmark [11] is used for evaluating the performance of different keypoint extractors on down stream task of 3D reconstruction using COLMAP at backend. Top 5000 keypoints are considered from each image. All other default setting are employed as described in [11] for fair comparative analysis.

baseline, almost all detectors registered all available images except TCDET on South-Building dataset. However, on larger datasets *(Madrid Metropolis, Gendarmenmarkt, Tower of London)*, due to wide baseline, number of registered images falls sharply for all detectors. However, on larger datasets, MSK-Tiny++ registers *highest* number of images as compared to all other keypoint detectors, even more than SIFT.

2. **# Sparse Points [11]** Number of sparse points are total number of points resulted from the SfM. It indicates the completeness of sparse reconstruction. On smaller datasets, SuperPoint generates highest number of sparse points and MSK-Tiny++ continuously produces 2^{nd} highest number of sparse point. However, on larger datasets, MSK-Tiny++ produces highest number of sparse points on all sequences. This indicates that under general wide baseline scenario,

MSK-Tiny++ produces a more complete sparse point cloud.

3. **# Observations [11]** Number of observations per image is total number of geometrically verified (triangulation) projections of sparse points on the images. Number of sparse points are proportional to number of observations. In case of number of observations, the trend is similar to sparse points exactly, with few minor exceptions. The overall performance of MSK-Tiny++ on number of sparse points and number of observations indicates it effectiveness in sparse reconstruction.

4. **Mean Track length [11]** Track length is the number of frames that feature is successfully tracked, from the frame the feature is detected to the one it is lost. On smaller dataset, SuperPoint produces longest track length for Herzjesu and South-Building, while KeyNet produces longest track length for Fountain dataset. MSK-Tiny++ produces the 2^{nd} longest mean track length on all these sequences. Similarly, on larger datasets, the same trend can be observed as well. However, the mean track length for SIFT and TCDET on all sequences of smaller as well as larger datasets is very short as compared to SuperPoint, KeyNet and MSK-Tiny++.

5. **Re-Projection Error [11, 20]** Minimizing Re-projection error is main objective of the bundle adjustment(BA). Final re-projection error in bundle adjustment stipulates the accuracy of final reconstruction. In terms of re-projection error, SIFT outperforms all keypoints detectors in both narrow as well as wide baseline scenarios, with few minor exceptions. On the contrary, MSK-Tiny++ produces highest re-projection error. This indicates that *sub-pixel localisation* is important for minimizing the re-projection error as used in SIFT [21].

6. **# Inlier Pairs [11]** In brute force matching, there are $N(N-1)/2$ image pairs for N images. Inlier pairs are those image pairs in which a fixed number of inliers are found after geometric verification. On smaller datasets, SuperPoint, KeyNet and MSK-Tiny++ verifies the maximum number of inlier image pairs. These detectors almost register maximum number of available image pairs. However, on larger datasets, MSK-Tiny++ produces highest number of inlier image pairs with large margins as compared to 2^{nd} and 3^{rd} highest number of inlier image pairs. This indicates the accuracy and redundancy of local features extracted by MSK-Tiny++ for geometric verification.

7. **# Inlier Matches [11]** Number of inlier matches is total number of inliers found after geometric verification using robust model estimator. On smaller datasets, SuperPoint produces maximum number of geometrically verified inlier matches whereas the number of inlier matches produced by MSK-Tiny++ is, on average, 2^{nd} highest. However, on larger datasets, the trend completely shifts and MSK-Tiny++ produces the highest number of inliers matches as compared to other keypoint detectors.

8. **# Dense points [11]** Number of dense points indicates the number of 3D points after dense reconstruction using Multi-view stereo. Number of dense points indicates the overall complete-

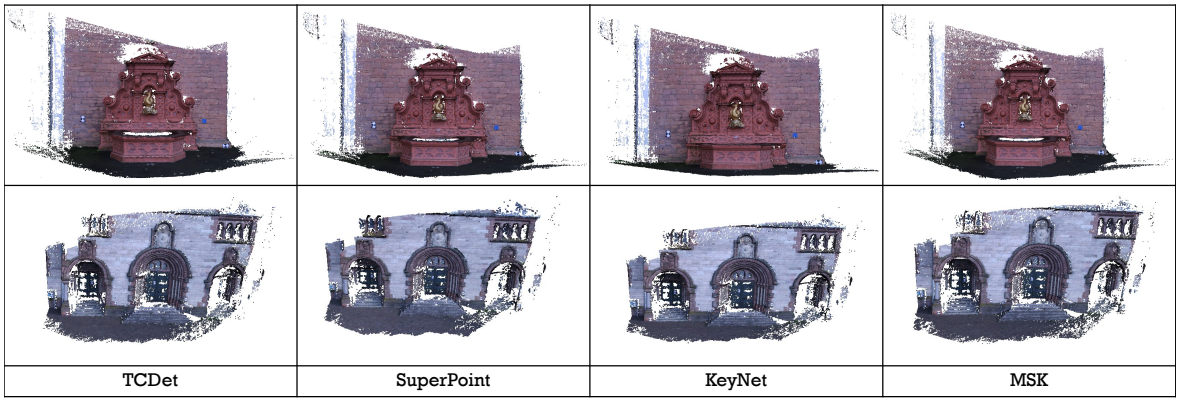


Figure 4.3:: **Reconstructed Dense Point Cloud.** Comparison of reconstructed dense point cloud using COLMAP [10] for *Fountain* and *Herzjesu* sequences of ETH 3D reconstruction benchmark [11] dataset for different keypoint detectors. Keypoints are extracted using TCDET [26], SuperPoint [47], KeyNet [6] and our proposed method, MSK-Tiny++. Hard-Net [16] descriptor head is used to generate descriptors for subsequent matching. All other default setting are employed as described in [11] for fair comparative analysis.

ness of dense model and the accuracy of the SFM results. In case of number of dense points, due to close margins, there is no regular trend on smaller as well as larger datasets. TCDET produces highest number of dense points on Fountain sequence, whereas, MSK-Tiny++ produces 2^{nd} highest number of dense points. On Herzjesu, SuperPoint and KeyNet produces 1^{st} and 2^{nd} highest number of dense points, respectively. MSK-Tiny++ produces the highest number of dense points for South-Building sequence. On Madrid Metropolis sequence, MSK-Tiny++ again produces the highest number of the dense points, whereas, MSK-Tiny++ produces 2^{nd} highest number of dense points for other two sequences. Figure 4.3 displays the final dense point cloud after meshing for *Fountain* & *Herzjesu* sequences. It shows the dense point cloud produced by COLMAP using keypoints from TCDET, SuperPoint, KeyNet and MSK-Tiny++ detectors with Hard-Net descriptors. Since the number of dense points produced by TCDET, SuperPoint, KeyNet and MSK-Tiny++, for *Fountain* & *Herzjesu*, are quite closer to each other, therefore, the final reconstructed scenes appear quite similar to each other.

On average, MSK-Tiny++ produces best performance on all afore-mentioned performance metrics, with an exception to Re-projection error. Overall performance of SuperPoint, KeyNet and MSK-Tiny++ is relatively close to each other on numerous performance metrics with few exceptions. However, performance of TCDET [26] is very subverting, keeping in view its performance on matching score and repeatability.

4.5 CVPR 2020 Image Matching Challenge

Establishing robust correspondences between two views is a fundamental problem in computer vision geometry. Though being a core problem of numerous important applications in machine vision and scene understanding, however, this problem in extremely challenging in generic wide baseline (stereo & multi-view) cases [4]. Numerous different solutions have been proposed to tackle this problem over last three decades. However, the solutions based on sparse methods i.e. local features have passed the test of time. Many learned and handcrafted local feature extraction methods show propitious performance on the repeatability and matching score. Especially, the recent data-driven *holistic* approaches have shown great improvement on these intermediary metrics. However, recent studies [4, 11] have shown that performance gains in these intermediate metrics usually does not translate linearly to the downstream and more important tasks like camera pose estimation which is an essential step in Structure from Motion and 3D reconstruction. Therefore, it is critical to look beyond intermediate metrics and focus on performance on downstream tasks i.e. accuracy of reconstructed camera poses. Since camera pose estimation is an iterative process with numerous hyper-parameters and components, which interact in complex ways, which effects the overall performance. Therefore, many *separatist* techniques such as SIFT [21] with properly fine tuned hyper-parameters and settings still outperforms the moderns techniques by a reasonable margin.

Therefore, in order to correctly evaluate the performance of local features & different matching strategies, they need to be properly embedded within the pipeline used to solve the given problem and the different components in said pipeline need to be tuned carefully and jointly, which requires engineering and domain expertise. To fill this need, CVPR 2020 workshop and challenge on *Wide-Baseline Image Matching: Local Features and Beyond* [18] [4] presents a unified modular pipeline and *benchmark* for image matching and subsequent camera pose reconstruction, providing an opportunity to analyze the performance of different methods impartially. This challenge provides a benchmark for the establishing standards in image matching and measuring the performance of all methods (local features and matching strategies) using a single universal metric i.e. *the accuracy of the reconstructed camera poses*. This benchmark provides an open source, easy-to-use and flexible framework for the benchmarking of local features, matching and robust estimation methods, both alongside and against top-performing methods.

We have evaluated the performance of keypoints extracted from MSK with descriptors generated using Hard-Net [16] with different matching strategies on this benchmark. We have also participated in subsequent Image Matching Challenge. In following, we have given a brief introduction of the afore-mentioned benchmark and challenge. For details regarding benchmark settings and challenge, please refer to the relevant article by organizers [4] and CVPR 2020 Image Matching Workshop

[18] https://www.cs.ubc.ca/research/image-matching-challenge/2020/

website [19].

4.5.1 Dataset for the Challenge: *Photo-Tourism*

For truly wide baseline image matching the existing dataset do not fulfil the purpose. Many hand-crafted and learned methods have shown promising performance on the many aforementioned existing datasets, but in real-world conditions their performance deteriorates very abruptly. A truly challenging dataset must contain wide range of geometric and photometric transformations including different imaging devices, temporal variation, weather, partial occlusions, etc. *Photo-Tourism* [64, 65] dataset is a very large collection of images containing almost 30,000 images of buildings and natural landmarks of around the world. This dataset consists of 26 sequences of images. Number of images in each sequence are ranging from 75 to 4000 images.Photo-Tourism dataset contains so diverse set of images that satisfy all these conditions and are readily available. This challenge and benchmark is thus build on Photo-Tourism dataset.

4.5.2 Benchmark Settings & Evaluation Strategy

CVPR 2020 workshop on *Wide-Baseline Image Matching: Local Features and Beyond* [4] provides a complete unified modular framework for benchmarking the various local features extraction methods, robust estimation methods, camera pose estimation and numerous other modules of structure from motion, both alongside and against top-performing methods. This benchmark provides basis for the CVPR 2020 Image Matching Challenge [20]. The benchmark pipeline along with ground truth generation is shown in figure 4.4. This pipeline provides a complete modular evaluation procedure, with easy integration from dataset to performance metrics and settings of the hyper-parameters within its different components. This framework uses Photo-Tourism [64, 65] dataset with thousands of images of 25 landmarks, taken from diverse viewpoints, with different cameras, in varying illumination and weather conditions, thus, providing a real world wide-baseline settings. For generating (pseudo) ground truth, this framework reconstruct the scenes with ,off-the-shelf technology, COLMAP [10]. It reconstructs the scenes with 26000 images from 25 different landmarks. COLMAP provides camera poses and depth maps for all images. This benchmark releases camera poses and depth maps for 26000 images from Photo-Tourism training set, however, it keeps the ground-truth camera poses and depth maps private for 4000 images of Photo-Tourism test set, which form the basis of Image match-

[19] https://www.cs.ubc.ca/research/image-matching-challenge/2020/benchmark/

[20] https://www.cs.ubc.ca/research/image-matching-challenge/2020/leaderboard/

ing Challenge. This modular benchmark provides an opportunity for camera pose estimation using the modern local features and robust estimation methods & compares with ground-truth camera poses under two different settings i.e. Stereo (using 2 images) and Multi-View (up to 25 images).

Stereo and multi-view reconstructions are evaluated with both downstream and intermediate metrics for comparison and in depth analysis. Since, the benchmark evaluates the accuracy of the reconstructed camera poses, hence, main error metric is based on the angular errors. Difference between the estimated and ground-truth translation and rotation vectors between two camera poses is computed. Different thresholds are applied over the computed error for all co-visible image pairs. Doing so over different angular thresholds renders a curve. Finally, it computes the *mean Average Accuracy (mAA)* by integrating this curve up to a maximum threshold, which we set to 5° and 10°. This *mean Average Accuracy (mAA)* is main evaluation metric. In addition to that, ground-truth pixel wise depth estimates are used to compute the classical pixel-wise intermediate metrics i.e. repeatability and matching score.

4.5.3 Benchmark Pipeline

Complete pipeline for CVPR Image Matching Benchmark & Challenge 2020 is shown in figure 4.4 along with its ground truth generation process. Feature extraction module takes a *subset* of $N = 100$ images from each scene [4]. This module computes up to $K \in \{2048, 8192\}$ features from each image. Feature matching module generates a list of putative matches using symmetric nearest neighbor matching for all *possible* image pairs i.e. $M = N(N-1)/2$ combinations. These matches can be (optionally) fed to outlier pre-filtering module, which filters these putative matches using context aware filtering. Eventually, these matches are fed to stereo and multi-view reconstruction modules which uses 2 and up to 25 images for reconstructing the camera poses, respectively. Here we have stated the crux of the each step briefly.

4.5.3.1 Feature Extraction

First and crucial step is local features extraction. Since this benchmark is generic so there is no restriction on the nature of local features. Any combination of local features i.e. keypoints and descriptors can be used for evaluating the performance of the *combination* of the local features on quality of reconstructed camera poses. However, the performance of the specific local features extractor strongly depends upon the number of local features fed to this pipeline. Therefore, CVPR 2020 Image Matching challenge allows submissions in two categories, *restricted features* ($K = 2048$) or *unlimited features* ($K = 8192$). Moreover, size and nature (binary or real-valued) of descriptor is also criti-

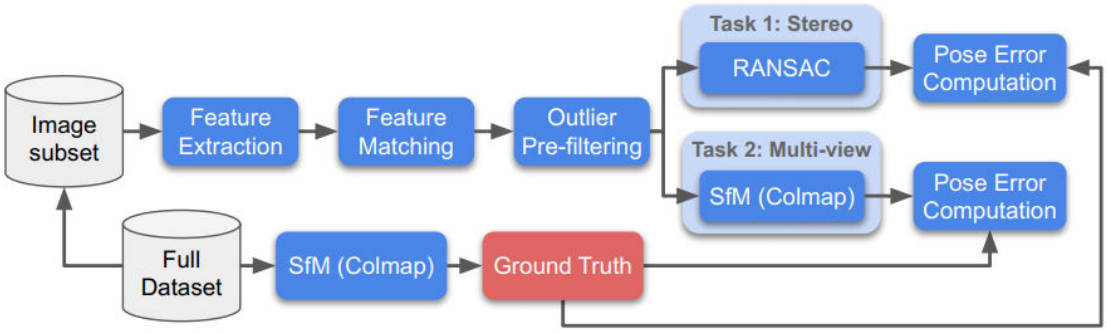


Figure 4.4:: **Complete pipeline for CVPR 2020 Image Matching Benchmark & Challenge** along with its (pseudo) ground truth generation process. Benchmark pipeline has feature extraction, feature matching, outlier pre-filtering, stereo and multi-view reconstruction modules. It takes a subset of images from each scene and extracts up to $K \in \{2048, 8192\}$ features per image. Feature matching module generates a list of putative matches using exhaustive or brute force matching. Eventually, these matches are fed to stereo and multi-view reconstruction modules for reconstructing the camera poses. Finally, mean average accuracy (mAA) is computed as error metric between estimated and ground-truth camera poses.

cal in assessing the quality of end goal. Therefore, CVPR 2020 Image Matching challenge allows submissions with either *small descriptors* $(128 - bytes)$ or *standard descriptor* $(512 - bytes)$. In order to optimize the computational cost and performance, we have evaluated the MSK-Tiny++ performance with Hard-Net [16] descriptors in *restricted features* $(K = 2048)$ and *standard descriptor* $(512 - bytes)$ category.

4.5.3.2 Feature Matching

This is second most crucial step in this benchmark evaluation. There are numerous hyper-parameters and heuristics involved in this step which strongly affects the performance of the subsequent tasks. This step is further divided in four sub-stages. Given images $\mathbf{I}_i$ and $\mathbf{I}_j$, $i \neq j$, we create an initial set of matches by *nearest neighbor (NN)* matching from $\mathbf{I}_i$ to $\mathbf{I}_j$, obtaining a set of matches $\mathbf{m}_{i \mapsto j}$. We optionally do the same in the opposite direction, which results in $\mathbf{m}_{j \mapsto i}$. Lowe's ratio test [21] is applied to each list to filter out non-discriminative matches, with a threshold $r \in [0, 1]$, creating *curated* lists $\tilde{\mathbf{m}}_{i \mapsto j}$ and $\tilde{\mathbf{m}}_{j \mapsto i}$. The final set of putative matches is lists intersection, $\tilde{\mathbf{m}}_{i \mapsto j} \cap \tilde{\mathbf{m}}_{j \mapsto i} = \tilde{\mathbf{m}}^{\cap}_{i \leftrightarrow j}$ (known in the literature as *one-to-one*, *mutual NN*, *bipartite*, or *cycle-consistent*), or union $\tilde{\mathbf{m}}_{i \mapsto j} \cup \tilde{\mathbf{m}}_{j \mapsto i} = \tilde{\mathbf{m}}^{\cup}_{i \leftrightarrow j}$ (known as *symmetric*). We refer to them as "both" and "either", respectively. Finally, the distance filter is optionally applied, removing matches whose distance is above a pre-defined threshold. "Both" strategy is similar to the *symmetrical nearest neighbor ratio (sNNR)*, proposed concurrently in [74], sNNR combines the nearest neighbor ratio in both directions into a single number by taking the harmonic mean, while here we take the maximum of the two values.

4.5.3.3 Outlier Pre-Filtering

Outlier pre-filter is an optional step. In this step outliers are further filtered using some context CNN network. We have skipped this step in our submissions and evaluations.

4.5.3.4 Stereo Reconstruction

List of tentative matches $\tilde{\mathbf{m}}^{\cap}_{i \leftrightarrow j}$ or $\tilde{\mathbf{m}}^{\cup}_{i \leftrightarrow j}$ is fed to a robust model estimator, which estimates Fundamental matrix $\mathbf{F}_{i,j}$, between $\mathbf{I}_i$ and $\mathbf{I}_j$. For robust model estimation, in addition to locally-optimized RANSAC [75] as implemented in Open-CV [76], and SKlearn [77], this benchmark also considers recent improvised robust model estimation algorithms such as:

- DEGENSAC (Degenerative RANSAC) [17]
- GC-RANSAC (Graph-Cut RANSAC) [78]
- MAGSAC (Marginalized RANSAC) [18]

Finally, given Fundamental matrix $\mathbf{F}_{i,j}$, with known intrinsics $\mathbf{K}_{i,j}$, these are used to compute the Essential matrix $\mathbf{E}_{i,j}$, as $\mathbf{E}_{i,j} = \mathbf{K}_j^T \mathbf{F}_{i,j} \mathbf{K}_i^T$. Finally, the *relative* rotation and translation vectors are recovered with OpenCV's *recoverPose*.

4.5.3.5 Multi-View Reconstruction

In stereo reconstruction, this benchmark only uses two images for reconstruction. In multi-view reconstruction, it reconstruct a scene from small image subsets, which are names as *bags*. It considers bags of 5, 10, and 25 images, which are randomly sampled from the original set of 100 images per scene. Specifically, for multi-view reconstruction, it creates 100 bags of bag size 5, 50of bag size 10, and 25 of bag size 25. It uses COLMAP [10], feeding it the matches computed by the previous module. COLMAP returns scene reconstruction for each bag of images. If multiple reconstructions are obtained, it considers the largest one. This benchmark also collect and report statistics such as the *number of landmarks* or the *average track length*. Both statistics and error metrics are averaged over the three bag sizes, each of which is in turn averaged over its individual bags.

4.5.4 Performance Metrics

Stereo and multi-view reconstructions are evaluated with both *downstream* and *intermediate* metrics for comparison and in depth analysis. In order to get the practical insights into the performance of the local features on the quality and accuracy of the reconstructed camera poses and depth maps for both stereo and multi-view reconstructions, CVPR 2020 image matching benchmark and challenge measures a number of important metrics to quantify the overall performance. Below we have briefly discussed these metrics separately for stereo and multi-view reconstructions:

- **Metrics for Stereo Reconstruction**: We evaluate the performance of local features on the accuracy of reconstructed camera poses in stereo settings using following metrics:

  - **Number of Inliers (NI)**: It is number of matches that are actually used to estimate the poses. Putative matches produced by nearest neighbor matching are fed to robust model estimator. Matches filtered by robust model estimator are fed for camera pose estimation, which are termed as inlier matches. Greater the number of inlier matches indicates the robustness of established correspondences.

  - **Repeatability & Matching Score**: This benchmark uses COLMAP's generated ground-truth depth estimates for computing repeatability and matching score using pixel thresholding. Repeatability and matching score is only reported for the stereo settings using fix threshold of $3 - pixels$.

  - **Mean Average Accuracy *(mAA)***: Since the benchmark evaluates the accuracy of the reconstructed camera poses, hence, main error metric is based on the angular errors. Since the reconstructed model is scale-agnostic, therefore, the difference between the estimated and ground-truth translation and rotation vectors between two camera poses is computed. Different thresholds are applied over the computed error for all co-visible image pairs. Repeating this for different angular thresholds renders a curve. Finally, it computes the *mean Average Accuracy (mAA)* by integrating this curve up to a maximum threshold, which we set to 5° and 10°. This *mean Average Accuracy (mAA)* is **main** evaluation metric for both **stereo** & **multi-view** reconstructions.

- **Metrics for Multi-view Reconstruction**: In addition of mean Average Accuracy, we evaluate the accuracy of reconstructed camera poses in multi-view settings using following metrics:

  - **Cameras Registration Ratio (CR)**: Cameras Registration ratio, simply registration ratio, is the percentage of cameras registered during pose estimation in a bag of images for multi-view reconstruction.

- **Mean Track Length (TL)**: Track length is the number of frames that feature is successfully tracked, from the frame the feature is detected to the one it is lost. Simply, it is number of observations per landmark.

- **Absolute Trajectory Error (ATE)**: Absolute trajectory error is absolute pose difference between the ground-truth poses and aligned estimated pose trajectory.

4.5.5 CVPR 2020 Image Matching Challenge Submissions

We have evaluated the performance of MSK's keypoints along with Hard-Net [16] descriptors on CVPR 2020 wide baseline image matching challenge and benchmark. Specifically, we have employed keypoints extracted from the MSK-Tiny++ along with standard $128-$dimensional descriptors generated using Hard-Net [16]. We have made three submissions to the associated challenge using standard evaluation protocol stated in [4] to evaluate the performance of our method in most practical conditions possible. We have evaluated our keypoint extractor using 2048 features. We have made submissions in *"Photo-Tourism : restricted keypoints, standard descriptors (512 bytes)"* category of the challenge. This benchmark pipeline has numerous settings and hyper-parameters, we have briefly discussed settings of all of our submissions below separately.

4.5.5.1 Submission ID: 00640

In our first submission, under submission ID: 00640, we have employed top 2048 keypoints extracted from MSK-Tiny++ and corresponding descriptors (128 float32: 512 bytes) are generated using Hard-Net [16]. For descriptor matching in both stereo and multi-view tracks, we have employed *symmetric* (one-to-one) nearest neighbor matching with l_2-distance and single neighbor. Final matches are filtered using pairwise filtering with 0.95 ratio [4]. For geometric verification in stereo track, we have used DEGENSAC [17] with degeneracy check on as our robust model estimator. For multi-view track, the subject benchmark uses built-in RANSAC of COLMAP [10] as robust model estimator for geometric estimation. We have not fine-tuned any hyper-parameters on validation set, rather, we have used default settings. For this submission, web link to detailed analysis, results and comparison with all other methods in challenge leader-board is here [21].

[21] https://www.cs.ubc.ca/research/image-matching-challenge/2020/submissions/sid-00640-mt-2-hardnet-pretrained-all-datasets-degensac-check-on/

4.5.5.2 Submission ID: 00641

In our second submission, under submission ID: 00641, we have employed top 2048 keypoints extracted from MSK-Tiny++ & corresponding descriptors (128 float32: 512 bytes) are generated using Hard-Net [16]. For descriptor matching in both stereo and multi-view tracks, we have employed *symmetric* (one-to-one) nearest neighbor matching with l_2-distance and single neighbor. Final matches are filtered using pairwise filtering with 0.95 ratio [4]. For geometric verification in stereo track, we have used MAGSAC [18] with default settings as our robust model estimator. For multi-view track, the subject benchmark uses built-in RANSAC of COLMAP [10] as robust model estimator for geometric estimation. We have not fine-tuned any hyper-parameters on validation set, rather, we have used default settings. For this submission, web link to detailed analysis, results and comparison with all other methods in challenge leader-board is here [22].

4.5.5.3 Submission ID: 00646

In our third submission, under submission ID: 00646, we have employed top 2048 keypoints extracted from MSK-Tiny++ & corresponding descriptors (128 float32: 512 bytes) are generated using Hard-Net [16]. For descriptor matching in both stereo and multi-view tracks, we have employed *symmetric* (one-to-one) nearest neighbor matching with l_2-distance and single neighbor. Final matches are filtered using pairwise filtering with 0.95 ratio [4]. For geometric verification in stereo track, we have used Graph-Cut RANSAC [78] with default settings as our robust model estimator. For multi-view track, the subject benchmark uses built-in RANSAC of COLMAP [10] as robust model estimator for geometric estimation. We have not fine-tuned any hyper-parameters on validation set, rather, we have used default settings. For this submission, web link to detailed analysis, results and comparison with all other methods in challenge leader-board is here [23].

4.5.6 CVPR 2020 Image Matching Challenge Results & Discussion

We evaluate the performance of MSK's keypoints along with Hard-Net [16] descriptors on CVPR 2020 wide baseline image matching challenge and benchmark. We have participated in *"Photo-Tourism : restricted keypoints, standard descriptors (512 bytes)"* category of the challenge. We have

[22] https://www.cs.ubc.ca/research/image-matching-challenge/2020/submissions/sid-00641-mt-2-hardnet-pretrained-all-datasets-magsac/

[23] https://www.cs.ubc.ca/research/image-matching-challenge/2020/submissions/sid-00646-mt-2-hardnet-pretrained-all-datasets-magsac/

		PyRANSAC		DEGENSAC		MAGSAC		
Method	NF	NI↑	mAA(10°)	NI↑	mAA(10°)	NI↑	mAA(10°)	Rank↑
CV-SIFT	2048.0	84.9	0.2489	79.0	0.2875	99.2	0.2805	13
CV-$\sqrt{SIFT}$	2048.0	84.2	0.2724	88.3	0.3149	106.8	0.3125	11
CV-SURF	2048.0	37.9	0.1725	72.7	0.2086	87.0	0.2081	16
CV-AKAZE	2048.0	96.1	0.1780	91.0	0.2144	115.5	0.2127	15
CV-ORB	2031.8	56.3	0.0610	63.5	0.0819	71.5	0.0765	20
CV-FREAK	2048.0	62.5	0.1461	65.6	0.1761	78.4	0.1698	18
L2-Net	1936.3	66.1	0.3131	92.4	0.3752	114.7	0.3691	6
DoG-Hard-Net	1936.3	111.9	0.3508	117.7	0.4029	150.5	0.4033	5
KeyNet-Hard-Net	2048.0	134.4	0.3272	174.8	**0.4139**	228.4	0.3897	1
KeyNet-SOS-Net	2048.0	88.0	0.3279	171.9	**0.4132**	212.9	0.3928	2
GeoDesc	1936.3	98.9	0.3127	103.9	0.3662	129.7	0.3640	7
ContextDesc	2048.0	118.8	0.2965	124.1	0.3510	146.4	0.3485	9
DoG-SOS-Net	1936.3	111.1	0.3536	132.1	0.3976	149.6	0.4092	4
LogPolarDes	1936.3	118.8	0.3569	124.9	**0.4115**	161.0	0.4064	3
MSK-Tiny++Hard-Net	2048.0	–	–	128.8	0.3313	219.7	0.3419	10
D2-Net(SS)	2045.6	107.6	0.1157	134.8	0.1355	259.3	0.1317	19
D2-Net(MS)	2038.2	149.3	0.1524	188.4	0.1813	302.9	0.1703	17
LF-Net	2020.3	100.2	0.1927	106.5	0.2344	141.0	0.2226	14
SuperPoint	2048.0	120.1	0.2577	126.8	0.2964	127.3	0.2676	12
R2D2	2048.0	191.0	0.2829	215.6	0.3614	215.6	0.3619	8

Table 4.6:: Table from CVPR 2020 Image Matching Benchmark and Challenge for Stereo Track [4] with 2048 features and standard descriptor with custom matches using PyRANSAC, DEGENSAC [17] and MAGSAC [18] as robust model estimators. We report i) Number of features **(NF)**, ii) number of inliers **(NI)** and iii) mean average accuracy **mAA**(10°) only. All methods are ranked according to the their mAA(10°) score. Top three column-wise best scores are colored as red, green and blue, respectively.

made three submissions in the associated challenge. However, for the sake of brevity and brachylogy, we only discuss the results of submission (ID:00641) 4.5.5.2, because this submission performs best overall. Below we have briefly discussed the results on stereo and multi-view tracks, separately.

4.5.6.1 Results on Stereo Task

Here we discuss the results of our CVPR 2020 Image Matching Challenge submission 00641 on Stereo track. In our best performing submission under submission ID 00641, we have used MAGSAC [18] as our robust model estimator for geometric verification in stereo reconstruction track. Table 4.6 shows the results from CVPR 2020 Image Matching Benchmark and Challenge for Stereo Track [4]

Method	NL↑	RR(%)↑	TL↑	mAA(10°)	mAA(10°)	ATE ↓	Rank↑
CV-SIFT	1081.2	87.4	3.70	0.3718	0.4562	0.6136	13
CV-$\sqrt{SIFT}$	1174.7	89.4	3.82	0.4074	0.4995	0.5589	12
CV-SURF	1186.6	88.6	3.55	0.3335	0.4184	0.6701	15
CV-AKAZE	1383.9	90.9	3.74	0.3393	0.4361	0.6422	14
CV-ORB	683.3	73.0	3.21	0.1422	0.1914	0.8153	19
CV-FREAK	1075.2	86.3	3.52	0.2578	0.3297	0.7169	17
L2-Net	1253.3	92.6	3.96	0.4369	0.5392	0.5419	9
DoG-Hard-Net	1338.2	93.7	4.03	0.4624	0.5661	0.5093	7
KeyNet-Hard-Net	1276.3	95.7	4.49	0.5050	0.6161	0.4902	2
KeyNet-SOS-Net	1475.5	96.5	4.42	0.5229	0.6340	0.4853	1
GeoDesc	1133.6	91.3	4.02	0.4246	0.5244	0.5455	10
ContextDesc	1504.9	93.3	3.92	0.4529	0.5568	0.5327	8
DoG-SOS-Net	1317.4	93.8	4.05	0.4739	0.5784	0.5194	6
LogPolarDes	1410.2	93.8	4.05	0.4794	0.5849	0.5090	5
MSK-Tiny++Hard-Net	1595.3	97.1	4.28	0.5029	0.6096	0.4991	4
D2-Net(SS)	2357.9	94.7	3.39	0.2875	0.3943	0.7010	16
D2-Net(MS)	2177.3	93.4	3.01	0.1921	0.3007	0.7861	18
LF-Net	1385.0	90.4	4.14	0.4156	0.5141	0.5738	11
SuperPoint	1184.3	92.4	4.34	0.4423	0.5464	0.5457	9
R2D2	1228.4	96.2	4.29	0.5045	0.6149	0.4956	3

Table 4.7:: **Table from CVPR 2020 Image Matching Benchmark and Challenge for Multi-view Track [4]** with 2048 features and standard descriptor with built-in matcher and baseline RANSAC as robust model estimator. For multi-view reconstruction, we report i) Number of landmarks **(NL)**, ii) Cameras registration ratio **(RR)**, which indicates the percentage of cameras registered in a bag, iii) track length **(TL)**, iv) Absolute Trajectory Error (ATE) and finally mean average accuracy **(mAA)** for 5° & 10°. All methods are ranked according to the their mAA(10°) score. Top three column-wise best scores are colored as red, green and blue, respectively.

with 2048 features and standard descriptor with custom matches using PyRANSAC, DEGENSAC [17] and MAGSAC [18] as robust model estimators for geometric verification in stereo track. In [4], if we turn off the degeneracy check in the DEGENSAC, it is termed as PyRANSAC. We have evaluated the results with DEGENSAC with degeneracy check on, therefore, we do not report results for PyRANSAC. Our proposed approach, MSK-Tiny++ produces mean average accuracy (mAA) score of 0.3419 at (10°) and it is ranked at $10th$ place.

In addition to that, we have discussed the results from CVPR image matching challenge leader board. Figure 4.5 shows a snapshot of the table of stereo reconstruction results from the associated challenge leader board. This table displays score and ranks for all performance metrics for individual sequences of the Photo-Tourism test set as well as average score for complete dataset. On repeatability, our

Scene	Features	Matches (matcher)	Matches (filter)	Matches (final)	Rep. @ 3 px.	MS @ 3 px.	mAA(5°)	mAA(10°)
bm	2048.0	310.1	310.1	238.7	0.592 Rank: 3/108	0.734 Rank: 95/108	0.02255 (±0.00168) Rank: 95/108	0.05244 (±0.00172) Rank: 95/108
fcs	2048.0	261.2	261.2	195.6	0.374 Rank: 29/108	0.840 Rank: 76/108	0.44613 (±0.00109) Rank: 75/108	0.58606 (±0.00052) Rank: 71/108
lms	2048.0	340.2	340.2	258.0	0.381 Rank: 36/108	0.616 Rank: 83/108	0.35760 (±0.00161) Rank: 76/108	0.47117 (±0.00219) Rank: 76/108
lb	2048.0	265.9	265.9	174.5	0.409 Rank: 10/108	0.626 Rank: 82/108	0.18848 (±0.00168) Rank: 95/108	0.26561 (±0.00079) Rank: 89/108
mc	2048.0	286.0	286.0	211.0	0.509 Rank: 16/108	0.879 Rank: 60/108	0.24253 (±0.00168) Rank: 95/108	0.38055 (±0.00340) Rank: 60/108
mr	2048.0	373.6	373.6	308.1	0.479 Rank: 5/108	0.881 Rank: 76/108	0.19968 (±0.00138) Rank: 58/108	0.29188 (±0.00082) Rank: 58/108
psm	2048.0	153.2	153.2	104.4	0.379 Rank: 3/108	0.420 Rank: 54/108	0.08373 (±0.00076) Rank: 70/108	0.17388 (±0.00114) Rank: 55/108
sf	2048.0	376.1	376.1	303.2	0.449 Rank: 2/108	0.769 Rank: 78/108	0.32756 (±0.00081) Rank: 65/108	0.46366 (±0.00018) Rank: 65/108
spc	2048.0	263.3	263.3	184.0	0.479 Rank: 1/108	0.736 Rank: 80/108	0.24725 (±0.00203) Rank: 79/108	0.39132 (±0.00059) Rank: 77/108
avg	2048.0	292.2	292.2	219.7	0.450 Rank: 4/108	0.722 Rank: 81/108	0.23506 (±0.00055) Rank: 79/108	0.34184 (±0.00032) Rank: 78/108

Figure 4.5:: **Snapshot of table of Stereo reconstruction results of CVPR 2020 Image Matching Challenge for Submission ID: 00641**. Table displays scores & ranks for i) Number of features, ii) Number of matches, iii) Number of inliers after filtering, iv) Repeatability score at threshold of 3-pixels, v) Matching score at threshold of 3-pixels and vi) Mean Average Accuracy (*mAA*) score for (5°) & (10°) threshold. This table reports score and overall rank against all methods for all performance metrics for individual sequences of the Photo-Tourism test set as well average score for complete test set.

submission 00641 produces an exceptional average score of 45% and ranked at $4th$ place in total 108 submissions. Similarly, our submission 00641, produces 0.23506 mAA(5°) and 0.34184 mAA(10°), which are ranked at 79 and $78th$ places, respectively. Figure 4.6 shows the final inliers survived after MAGSAC, robust model estimation loop. Ground-truth depth estimates from COLMAP are used to check whether survived inliers are correct or not. Matches with above 5-pixel error threshold are drawn in red and and those below 5-pixel error are color-coded by their error, from 0 (green) to 5 pixels (yellow). Matches for which depth-estimates are not given are drawn in blue.

4.5.6.2 Results on Multi-View Task

Here we discuss the results of our CVPR 2020 Image Matching Challenge submission 00641 on Multi-view track. In our submission 00641, for multi-view settings, we have used same matching strategy as in stereo track. Table 4.7 shows the results from CVPR 2020 Image Matching Benchmark and Challenge for Multi-view Track [4] with 2048 features and standard descriptor with built-in matcher and baseline RANSAC as robust model estimator. Our proposed approach, MSK-Tiny++ produces mean average accuracy (mAA) score of 0.6096 at (10°) and it is ranked at $4th$ place in

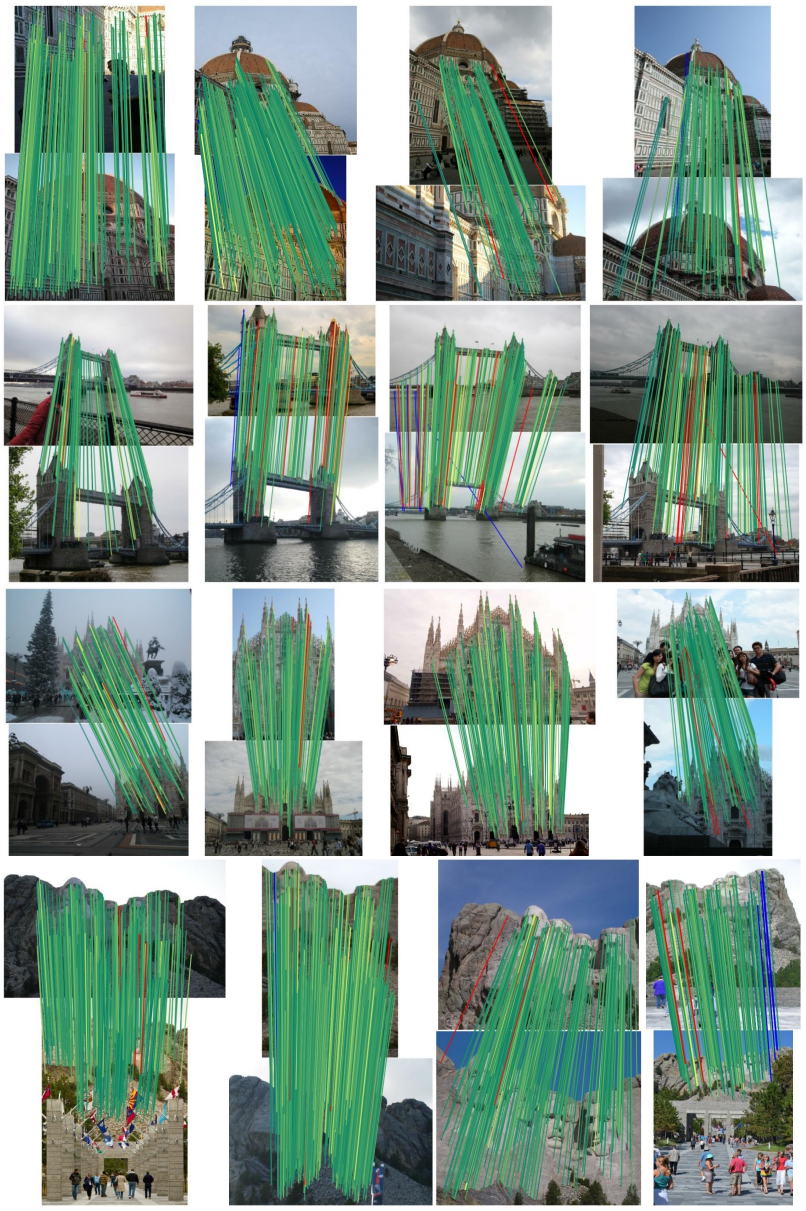


Figure 4.6:: **Visualization of inliers survived after robust model estimation during geometric verification of Stereo reconstruction for CVPR 2020 Image Matching Challenge submission 00641.** Figure shows the final inliers survived after MAGSAC, robust model estimation loop. Ground-truth depth estimates from COLMAP are used to check whether survived inliers are correct or not. Matches with above 5-pixel error threshold are drawn in red and and those below are color-coded by their error, from 0 (green) to 5 pixels (yellow). Matches for which depth-estimates are not given are drawn in blue.

overall ranking.

In addition to that, we have discussed the results from CVPR image matching challenge leader board.

Scene	Features	Matches (input)	Registration Ratio (%)	Number of Landmarks	Track Length	ATE	mAA(5^0)	mAA(10^0)
bm	2048.0	308.98	99.22 Rank: 84/108	1478.01 Rank: 39/108	4.488 Rank: 87/108	0.87681 Rank: 92/108	0.15086 (±0.00419) Rank: 93/108	0.25544 (±0.00603) Rank: 93/108
fcs	2048.0	269.42	98.17 Rank: 13/108	1707.09 Rank: 62/108	4.032 Rank: 79/108	0.25719 Rank: 27/108	0.70427 (±0.01029) Rank: 34/108	0.78063 (±0.01320) Rank: 29/108
lms	2048.0	368.35	99.38 Rank: 9/108	1897.38 Rank: 27/108	4.467 Rank: 83/108	0.30359 Rank: 40/108	0.75461 (±0.01266) Rank: 50/108	0.83856 (±0.01240) Rank: 45/108
lb	2048.0	323.44	98.33 Rank: 30/108	1425.79 Rank: 40/108	4.520 Rank: 41/108	0.50434 Rank: 40/108	0.51116 (±0.00167) Rank: 59/108	0.62105 (±0.00326) Rank: 60/108
mc	2048.0	280.36	99.62 Rank: 31/108	1281.45 Rank: 46/108	4.571 Rank: 36/108	0.43719 Rank: 39/108	0.44981 (±0.00660) Rank: 40/108	0.60467 (±0.00827) Rank: 38/108
mr	2048.0	373.89	94.24 Rank: 15/108	1676.48 Rank: 40/108	4.466 Rank: 28/108	0.53290 Rank: 24/108	0.35694 (±0.01030) Rank: 37/108	0.48028 (±0.00875) Rank: 32/108
psm	2048.0	148.70	87.24 Rank: 36/108	1476.26 Rank: 46/108	3.039 Rank: 54/108	0.67800 Rank: 44/108	0.29778 (±0.00443) Rank: 39/108	0.38924 (±0.00414) Rank: 37/108
sf	2048.0	371.50	98.20 Rank: 25/108	1857.26 Rank: 48/108	4.559 Rank: 36/108	0.32333 Rank: 31/108	0.68306 (±0.00270) Rank: 38/108	0.78044 (±0.00315) Rank: 35/108
spc	2048.0	270.15	99.38 Rank: 36/108	1557.97 Rank: 49/108	4.359 Rank: 46/108	0.57822 Rank: 75/108	0.61772 (±0.00421) Rank: 63/108	0.73637 (±0.00560) Rank: 56/108
avg	2048.0	301.64	97.09 Rank: 32/108	1595.30 Rank: 43/108	4.278 Rank: 62/108	0.49907 Rank: 60/108	0.50291 (±0.00122) Rank: 55/108	0.60963 (±0.00152) Rank: 53/108

Figure 4.7:: **Snapshot of table of Multi-view reconstruction track results of CVPR 2020 Image Matching Challenge for Submission ID: 00641**. Table displays scores & ranks for i) Number of features extracted, ii) Number of matches, iii) Cameras Registration Ratio (%) for a bag of images, iv) Number of Landmarks, v) Track Length vi) Absolute Trajectory Error (ATE) and vii) Mean Average Accuracy (*mAA*) score for (5°) & (10°) threshold. Table reports score and overall rank against all methods for all performance metrics for individual sequences of the Photo-Tourism test set as well average score for complete test set.

Figure 4.7 shows a snapshot of the table of Multi-view reconstruction results from the associated challenge leader board. This table displays score and ranks for all performance metrics for individual sequences of the Photo-Tourism test set as well as average score for complete dataset. On registration ratio, our submission 00641 produces an exceptional average score of 97.1% and ranked at $32nd$ place in total 108 submissions. Similarly, our submission 00641, produces 0.50291 mAA(5°) and 0.60963 mAA(10°), which are ranked at 55 and $53rd$ places, respectively. Figure 4.8 shows the extracted keypoints used during multi-view reconstruction. Bag of images are used in multi-view reconstruction using COLMAP. Keypoints have become part of final reconstruction are shown in blue and those which are not part of final model are shown in red.

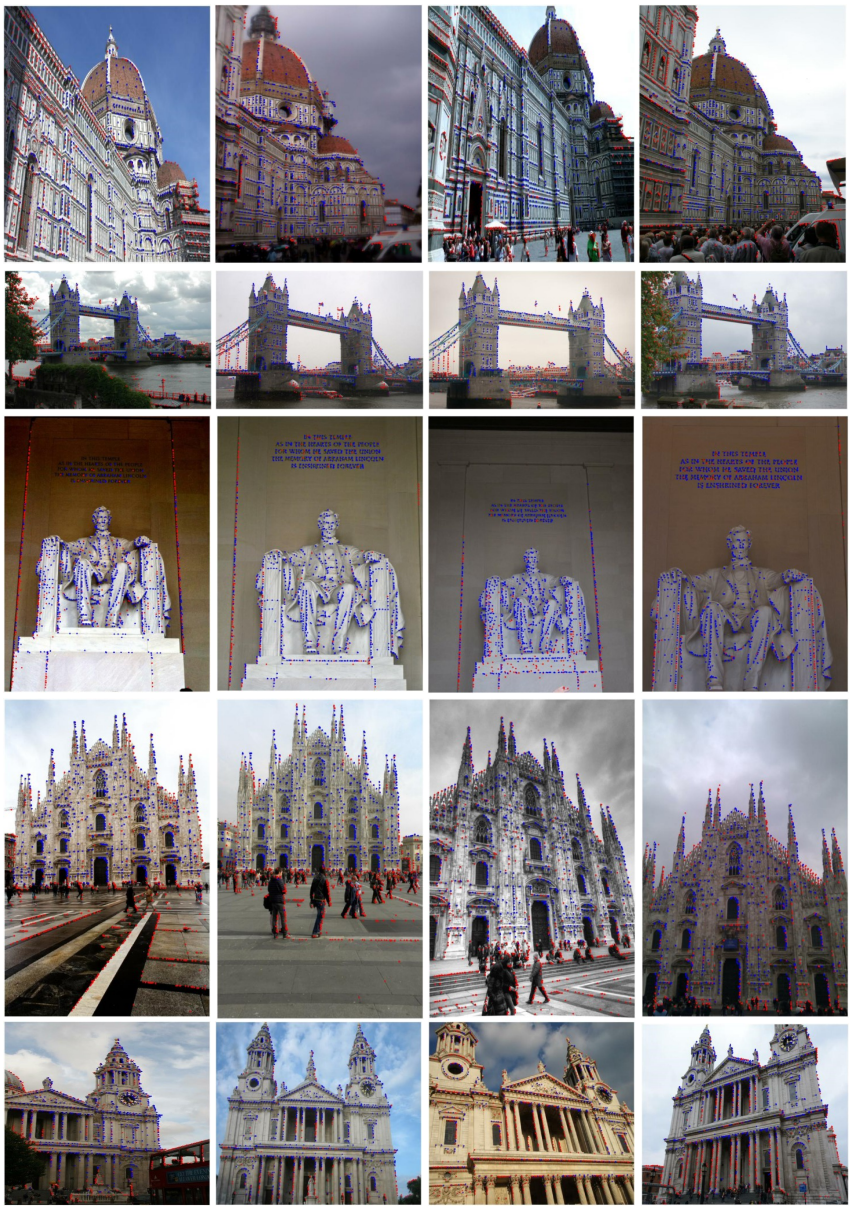


Figure 4.8:: **Visualization of keypoints which are part of finally reconstructed model for a bag of images for Multi-view reconstruction for CVPR 2020 Image Matching Challenge submission 00641.** Figure shows the extracted keypoints used during multi-view reconstruction. Bag of images are used in multi-view reconstruction using COLMAP. Keypoints have become part of final reconstruction are shown in blue and those which are not part of final model are shown in red.

Conclusions

In this chapter, we extensively studied the evaluation of our proposed approach, Multi-Scale Keypoint (MSK) extractor, for numerous intermediary metric as well downstream tasks. First of all, we evaluated the performance of MSK using repeatability score, which is fundamental metric for measuring the performance of the stand alone keypoint extractors. We see that MSK and its variants consistently performed better than all the state-of-the-art methods. Secondly, we evaluated MSK keypoints using image matching tasks which is main problem in many computer vision applications. For uniformity of the results, we fixed the descriptor head for all the detectors. MSK and its variants outperformed all other keypoint extractors by wide margin on matching score as well. Thirdly, we tested the performance of MSK on the downstream task of Structure from Motion and Multi-view Stereo for getting insights in to performance of MSK on the 3D reconstruction. We use numerous performance measuring reconstruction metric to evaluate the performance on this downstream task. MSK performed consistently best on almost all of these metric, which indicates the effectiveness of MSK for 3D reconstruction applications. Finally, we discussed the performance of MSK on extremely general task of wide baseline image matching on CVPR 2020 wide baseline Image Matching Benchmark and Challenge using the accuracy of reconstructed camera pose as main our metric. For all these evaluation, we used the publicly available datasets and benchmark evaluation methods for fair comparison and for getting meaningful insights into the working of MSK.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In essence, we undertook a detailed study of correspondence estimation problem in the context of sparse methods i.e. local features. First we introduced the fundamental role of robust correspondences in numerous key computer vision tasks. We took an in depth review of the all the prominent handcrafted as well as learned local features extraction methods. We also highlighted the strengths and weaknesses of these methods and also discussed their role in the establishing the robust correspondences in computer vision tasks. We noted that the inherent structure of convolutional neural networks helps to learn the low level representation of such local features and provide an opportunity to distill these features which has robustness and stability to be used in establishing the robust correspondences in image matching. We argued that how the natural inherited structure of the convolutional neural network learns better image priors and low level representation directly from the imaging data. Thus, we developed a stand-alone keypoint extractor, titled as *Multi-Scale Keypoint (MSK)* extractor, by first defining a keypoint in context of convolutional neural network and then associating the locality of keypoint with effective receptive field of a convolutional neural network. We effectively studied that learned convolutional filters can be used in place of handcrafted filters to efficiently extract and distill the low-level representation to extract salient image regions termed as keypoints. We demonstrated how can we combine the concept of information change with the varying receptive field to build a scale space which is further employed to estimate the scale of the keypoints.

Secondly, we carry out extensive evaluation of our proposed approach on numerous tasks. We evaluate the performance of the MSK on numerous intermediary as well as downstream tasks using various performance metrics. In all experimentation, we also comprehensively study the effect of various hyper-parameters of these pipelines on end goal. We start from most fundamental metric of repeatability of keypoints and finally discuss the performance of MSK in on a standard image matching benchmark using a single yet most important metric of accuracy of reconstructed camera poses. During all these evaluation and the analysis of the results, we point out that keypoints extracted from MSK perform reasonably well on multi-view stereo task in image matching as well as in dense reconstruction. This highlights the stability of the MSK's keypoints on wide baseline scenario. Finally, we conclude that there are numerous other parameters in camera pose estimation and 3D reconstruction which significantly effects the end goal.

5.2 Broader Impact

The main contribution of this thesis work is two fold. One, developing a stand alone keypoint detector, second, to study the performance of our proposed approach and other state of the art methods on downstream tasks i.e. accuracy of the reconstructed camera poses, structure from motion and multi-view stereo. In general, we developed first stand alone keypoint detector using transfer learning i.e. by employing the features extracted from a pre-trained feature extractor. Moreover, ours is the first method to build a conditioned scale space using pyramid based on receptive field, which is continuous bounded by minimum and maximum value of receptive field of the network. Moreover, our proposed approach is modular in nature and provide an opportunity to use any base feature extractor keeping other modules fixed, which in turn, gives us a chance to use application specific base feature extractor to enhance the generalization of keypoints to a specific application. However, this thesis works also highlights the sensitive dependence of the the correspondence estimation pipeline on numerous yet critical hyper parameters which strongly determines the quality of the end goal irrespective of the nature of learned features used during the process.

5.3 Future Work

In future, first, we intend to further reduce the computational intensity of MSK so that at inference time the pyramid computation does not increase the inference time. Secondly, compatibility of the descriptor head with the underlying keypoints strongly determines the quality of correspondences established during image matching. Therefore, a combined training of the keypoint extractor with descriptor head in inevitable for improved performance on image matching tasks. In addition to developing a matching pipeline, the robustness of the established correspondences also strongly depend upon the numerous hyper-parameters. Values of these hyper-parameters depend upon the underlying local features (keypoints and descriptors). Therefore, in order to obtain the best performance on the wide baseline image matching tasks like CVPR image matching challenge, it is eminent to set these hyper-parameters on their optimal setting. Fine tuning of these hyper-parameters is very important for specific local features on validation dataset. We also intend to develop a procedure for the fine tuning of these parameters on validation set of our image matching pipeline as well as performing the ablation study to get meaningful insights into working of these local features on downstream tasks.

REFERENCES

[1] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987. vii, 3, 7, 8, 18

[2] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113. vii, 2, 9, 45

[3] J. L. Schönberger, "Robust methods for accurate and efficient 3d modeling from unstructured imagery," Ph.D. dissertation, ETH Zurich, 2018. vii, 2, 9, 45

[4] 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 Computer Vision*, vol. 129, no. 2, pp. 517–547, 2021. vii, ix, 3, 9, 30, 34, 35, 36, 37, 42, 43, 45, 46, 47, 53, 54, 55, 59, 60, 61, 62, 63

[5] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016. vii, 9, 45, 46, 47

[6] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, 2019. vii, 13, 18, 19, 21, 22, 25, 27, 40, 41, 43, 44, 49, 52

[7] L. Florack, B. ter Haar Romeny, M. Viergever, and J. Koenderink, "The gaussian scale-space paradigm and the multiscale local jet," *International Journal of Computer Vision*, vol. 18, no. 1, pp. 61–75, 1996. vii, 13, 18, 19, 21, 22

[8] K. Lenc and A. Vedaldi, "Large scale evaluation of local image feature detectors on homography datasets," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 122. [Online]. Available: http://bmvc2018.org/contents/papers/0462.pdf vii, 9, 34, 35, 37, 39, 41

[9] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182. vii, ix, 15, 35, 36, 37, 38, 41, 42, 46

[10] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113. vii, 45, 52, 54, 57, 59, 60

[11] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1482–1491. vii, 3, 34, 35, 44, 45, 48, 49, 50, 51, 52, 53

[12] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: a temporally invariant learned detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5279–5288. ix, 3, 4, 12, 14, 16, 26, 40, 41, 44

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005. [Online]. Available: https://doi.org/10.1007/s11263-005-3848-x ix, 21, 35, 36, 37, 39, 41

[14] C. L. Zitnick and K. Ramnath, "Edge foci interest points," in *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 2011, pp. 359–366. [Online]. Available: https://doi.org/10.1109/ICCV.2011.6126263 ix, 36, 37, 38, 41

[15] K. Cordes, B. Rosenhahn, and J. Ostermann, "Increasing the accuracy of feature evaluation benchmarks using differential evolution," in *2011 IEEE Symposium on Differential Evolution (SDE)*. IEEE, 2011, pp. 1–8. ix, 36, 37, 39, 41

[16] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems*, 2017, pp. 4826–4837. ix, 12, 42, 43, 44, 46, 47, 50, 52, 53, 56, 59, 60

[17] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 772–779. ix, 57, 59, 61, 62

[18] D. Barath, J. Matas, and J. Noskova, "Magsac: marginalizing sample consensus," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 197–10 205. ix, 57, 60, 61, 62

[19] D. Forsyth and J. Ponce, *Computer vision: A modern approach.* Prentice hall, 2011. 1, 7

[20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 12, 26, 30, 49, 51

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 2, 11, 12, 18, 19, 40, 44, 49, 51, 53, 56

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 3102–3117. 3, 12, 21, 22

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3, 12, 15, 21, 22, 31

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 3, 12

[25] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020. 3, 12

[26] X. Zhang, F. X. Yu, S. Karaman, and S.-F. Chang, "Learning discriminative and transformation covariant local feature detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6818–6826. 3, 12, 13, 14, 16, 26, 28, 41, 44, 49, 52

[27] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: learning local features from images," in *Advances in Neural Information Processing Systems*, 2018, pp. 6234–6244. 4, 12, 15, 16, 25, 26, 40, 41, 44

[28] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483. 4, 12, 14, 15, 16, 26, 27, 40, 41, 44

[29] T. Tuytelaars, K. Mikolajczyk *et al.*, "Local invariant feature detectors: a survey," *Foundations and trends® in computer graphics and vision*, vol. 3, no. 3, pp. 177–280, 2008. 9

[30] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," in *IEEE International Conference on Industrial Technology, ICIT 2017, Toronto, ON, Canada, March 22-25, 2017*, 2017, pp. 1059–1063. [Online]. Available: https://doi.org/10.1109/ICIT.2017.7915508 9

[31] G. Csurka and M. Humenberger, "From handcrafted to deep local invariant features," *CoRR*, vol. abs/1807.10254, 2018. [Online]. Available: http://arxiv.org/abs/1807.10254 9, 11

[32] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244. 10, 44

[33] O. A. Zuniga and R. M. Haralick, "Corner detection using the facet model," in *CVPR 1983*, 1983. 10

[34] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000027790.02288.f2 10, 29

[35] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005. 10, 35, 42

[36] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417. 11, 40, 41

[37] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *European Conference on Computer Vision*. Springer, 2012, pp. 214–227. 11, 18, 19

[38] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011. 11, 44

[39] S. M. Smith and J. M. Brady, "Susan—a new approach to low level image processing," *International journal of computer vision*, vol. 23, no. 1, pp. 45–78, 1997. 11

[40] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443. 11, 40, 41

[41] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004. 11, 44

[42] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2. 12

[43] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792. 12

[44] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 IEEE international conference on computer vision (ICCV)*. Ieee, 2011, pp. 2548–2555. 12, 40, 41

[45] A. Irshad, R. Hafiz, M. Ali, M. Faisal, Y. Cho, and J. Seo, "Twin-net descriptor: Twin negative mining with quad loss for patch-based matching," *IEEE Access*, vol. 7, pp. 136 062–136 072, 2019. 12, 42

[46] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101. 12, 14, 15, 16, 26

[47] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236. 12, 14, 15, 16, 26, 27, 40, 41, 44, 49, 52

[48] K. Lenc and A. Vedaldi, "Learning covariant feature detectors," in *European Conference on Computer Vision*. Springer, 2016, pp. 100–117. 13, 14, 25, 26, 27, 28, 40, 41, 44

[49] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1822–1830. 13, 26

[50] K. Moo Yi, Y. Verdie, P. Fua, and V. Lepetit, "Learning to assign orientations to feature points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 107–116. 14

[51] 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 the IEEE International Conference on Computer Vision*, 2015, pp. 118–126. 14

[52] 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 the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8132–8140. 15, 25, 26, 40, 41

[53] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," *arXiv preprint arXiv:1906.06195*, 2019. 15, 16

[54] A. Benbihi, M. Geist, and C. Pradalier, "Elf: Embedded localisation of features in pre-trained cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, June 2019. 15

[55] P. Truong, S. Apostolopoulos, A. Mosinska, S. Stucky, C. Ciller, and S. D. Zanet, "Glampoints: Greedily learned accurate match points," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 16

[56] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454. 16, 18, 19

[57] Y. Cho, M. Faisal, U. Sadiq, T. Arif, R. Hafiz, J. Seo, and M. Ali, "Learning to detect local features using information change," *IEEE Access*, vol. 9, pp. 43 898–43 908, 2021. 18

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. 18

[59] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6129–6138. 22

[60] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. 22, 23

[61] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010. 22, 23

[62] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013. 24

[63] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5097–5106. 29

[64] *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015. [Online]. Available: https://ieeexplore.ieee.org/xpl/conhome/7293313/proceeding 30, 37, 54

[65] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, "YFCC100M: the new data in multimedia research," *Commun. ACM*, vol. 59, no. 2, pp. 64–73, 2016. [Online]. Available: http://doi.acm.org/10.1145/2812802 30, 37, 54

[66] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPRW.2009.5206848 31

[67] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037. 31

[68] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005. 35

[69] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: a temporally invariant learned detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5279–5288. 36, 37, 38

[70] 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 the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610. 45

[71] 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 *2008 IEEE conference on computer vision and pattern recognition*. Ieee, 2008, pp. 1–8. 47

[72] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3d scene reconstruction and class segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 47

[73] K. Wilson and N. Snavely, "Robust global translations with 1dsfm," in *European conference on computer vision*. Springer, 2014, pp. 61–75. 47

[74] F. Bellavia and C. Colombo, "Is there anything new to say about sift matching?" *International journal of computer vision*, vol. 128, no. 7, pp. 1847–1866, 2020. 56

[75] 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*, vol. 24, no. 6, pp. 381–395, 1981. 57

[76] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, p. 120, 2000. 57

[77] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011. 57

[78] D. Barath and J. Matas, "Graph-cut ransac," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6733–6741. 57, 60