

令和元年度 博士論文（要約）



**3D Reconstruction of Scattered Objects
within RGB-D Images**

(RGB-D 画像内に点在する物体の三次元再構成)

指導教員 太田 順 教授

東京大学大学院 工学系研究科 精密工学専攻

学生証番号 37-167268

サホロール ハムディ モハメド サレフ
Sahloul, Hamdi Mohammed Saleh

**3D Reconstruction of Scattered Objects
within RGB-D Images**

by

Hamdi Mohammed Saleh Sahloul

Submitted to Department of Precision Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering

at

THE UNIVERSITY OF TOKYO

September 2019

Certified by

Jun Ota
Professor
Thesis Supervisor

Accepted by

Jun Ota
Professor
Chairman, Thesis Committee

© Hamdi Mohammed Saleh Sahloul, MMXIX. All rights reserved.

The author hereby grants to The University of Tokyo permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Except where otherwise indicated, this thesis is my own original work.

Hamdi Mohammed Saleh Sahloul
August 15, 2019

To Allah The Almighty...
To my father and mother...
To my wife and two daughters...

Acknowledgments

It is my pleasure to acknowledge the assistance that I have received from many individuals and organizations, while conducting my research and preparing this Ph.D. thesis. First and foremost, I would like to express my deepest gratitude towards my supervisor Professor Jun Ota for his patience, time, and generous effort during this program. Professor Ota has been a key figure by giving lots of insightful comments and inquiries regarding this research. I am also aware of the generous help that I have received from MEXT (Ministry of Education, Culture, Sports, Science and Technology), by granting me the Monbukagakusho Scholarship that facilitated my study in Japan. Also, I would like to thank the staff and students of the Mobile Robotics Laboratory, and The University of Tokyo for providing an extremely excellent research environment. I owe a particular debt of gratitude to Assistant Professor Shouhei Shirafuji for his assistance. Furthermore, I appreciate the efforts of both Ahmed Al-Fusail and Abdullah Arafa for their iterative and extensive proofreadings of the initial thesis drafts. Last but not the least, I offer my humble gratitude to my parents for supporting me immeasurably, my brother and my three sisters for always being there throughout my life, my wife for her patience, sacrifice, and companionship during both the good and the bad, through our voyage in Japan; lastly, my little two daughters who brought happiness to my life, and gave it a meaning.

3D Reconstruction of Scattered Objects

within RGB-D Images

by

Hamdi Mohammed Saleh Sahloul

Submitted to Department of Precision Engineering
on August 15, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Engineering

Abstract

Constructing 3D structures of real-world objects is a scientific problem for a wide variety of fields, including computer vision and robotics. In recent decades, hardware utilized in reconstruction, such as depth sensors and TOF (time of flight) cameras, have become widely affordable, enabling reconstruction to gain huge attention. Nowadays, 3D reconstruction is in active research to achieve goals ranging from aiding computer graphics, virtual and augmented reality to surveillance, robotic navigation, manipulation and objects recognition.

Various and numerous reconstruction applications in home and office environments as well as industrial environments have motivated many researchers to introduce different reconstruction methods. Briefly, there are three types of reconstruction methods: MVS (multi-view stereo), SfM (structure from motion), and dense VSLAM (visual SLAM (simultaneous localization and mapping)). 3D reconstruction from 2D images is an old problem, which have been tackled by various algorithms such as SfM and MVS. More importantly, after the development of RGB-D sensors in the last decade, VSLAM methods have emerged and made great momentum in the 3D reconstruction research society. Notably, KinectFusion [1] is one of the early VSLAM that have achieved real-time 3D construction using RGB-D sensor data. However, real-time 3D construction methods usually depends on marginal camera assumptions, and thus fail to reconstruct from RGB-D images of different view. For such reasons, researchers are now revisiting MVS and SfM techniques and augmenting them with the new RGB-D images in order to achieve robustness and model completeness. MVS utilizes several extrinsically calibrated cameras system, while SfM depends on

motion cues within its input 2D or 2.5D images by estimating their correspondences.

Importantly, the objective is to reconstruct accurate models covering all objects' surfaces and to segment them from their environments within reasonable time. An input of several RGB-D images is considered, where each RGB-D image captures a scene of various same- or different-type objects from numerous viewpoints. However, MVS and VSLAM are unable to cope with such input due to their camera motion constraints. Instead, if each input RGB-D image is segmented according to the captured objects and viewpoints, such that each segment is considered as an independent RGB-D image of its own, then a virtual camera motion between these segments can be assumed, and the problem can be reduced into a SfM problem. Although SfM does not have a camera motion limitation, it suffers from a time-accuracy trade off. More importantly, all reconstruction methods cannot segment an object model from its environment without texture, geometry, or motion assumptions. Yet, the discussions of both the time-accuracy trade off and the segmentation issue are deferred until more intuition is developed. Indeed, 3D reconstruction problem embarrasses a great deal of techniques, and for simplicity purposes, its study is divided into three categories based on their correspondence complexity: 1-1 (one-to-one), 1-M (one-to-many), and M-M (many-to-many). In 1-1 or model-view correspondence, one input, i.e. an RGB-D image, captures a single object from a viewpoint while another captures the same physical object yet from a different viewpoint, where the problem involves finding which parts of one viewpoint represent the same physical point in the other viewpoint, i.e. correct or *inlier correspondences*, in order to register the inputs together into a 3D model. On the other hand, if one input involves several different objects, then the correspondence problem is a 1-M or model-scene, which is more involved than the 1-1 relation, due to the additional need to localize the correct object. If several objects are observed in each input, then the correspondence problem is a M-M or scene-scene, in which there is a need to segment the scenes in order to reduce the problem to several 1-M ones. Importantly, although two inputs are employed in the above demonstration, the correspondence problem can involve arbitrarily larger number of inputs.

Notwithstanding, each of these correspondence problems has its own challenges. Firstly, the success in solving 1-1 correspondence problems depends heavily on the input viewpoints, for which if two inputs have been taken from very different angles, i.e. relatively-large *viewpoint difference*, the common surfaces captured within these inputs drastically diminishes, rendering their matching unattainable. Even in the moderately overlapping-surface cases, algorithms desperately struggle to detect the same physical point in such inputs due to perspective changes. To explain this limitation, there is a need to introduce some algorithmic aspects first. A correspondence

between two inputs is usually estimated by detecting some interesting or outstanding points, i.e. *features*, from both inputs and matching them using their intrinsic descriptions, e.g. color gradient or neighborhood curvature. However, feature detection and description algorithms lack enough repeatability and distinctiveness under large viewpoint difference due to the inconsistencies observed around the same physical point within its observed very-different perspectives. For this, in the first contribution, a wrapper is proposed around existing algorithms, i.e. *wrapper*, for improving the invariance of feature-based algorithms against viewpoint difference. The idea is to rotate and translate surfaces to an invariant perspective, thus improving the feature detection and description. Another important benefit from improving an algorithm’s maximal viewpoint difference, i.e. the viewpoint invariance, is to decrease the number of required frames for covering the whole object, thus the consumed time on capture and reconstruction, thus addresses the previously deferred time-accuracy trade off. Secondly, as for the 1-M correspondences, even with highly repeatable and distinctive algorithms for feature detection and description, incorrect correspondences, i.e. outliers, are mostly contaminating the correspondences set due to noise, similar surfaces, occlusions ...etc. Filtering such contaminated, i.e. putative, correspondences by rejecting the outliers is a very distinct and well-known problem. Accordingly, in the second contribution, a rapid *voting*-based pattern recognition scheme is proposed to rank the single-structure putative correspondences according to their likelihood being inliers, thus enabling application-specific outlier rejection, which contributes in addressing the time-accuracy trade off as well. Thirdly, the segmentation problem embedded with the M-M correspondence problem gets more intractable due to the scarce information on the spatial shapes of each object, where the aim is to actually solve such problem for 3D reconstruction to begin with. The multi-structure hypotheses modeling approach in the third contribution extends that of the previous contribution, to enable modeling the motion of several objects within their environment, even under high outliers rate.

As for results, firstly, the detectors and descriptors wrapper improved their viewpoint invariance to different levels, depending on the scene geometric curvatures, while unwrapped existing, i.e. *2D*, feature algorithms fail to accommodate average viewpoint difference beyond 33.3° . Objects with distinct surface discontinuities were on average matchable up to 52.8° , and the overall average for all evaluated datasets was 45.4° . Similarly, out of a total of 140 combinations involving 20 feature algorithms and various objects with distinct surface discontinuities, only a single 2D feature algorithm exceeded the goal of 60° viewpoint difference in just two combinations, as compared with 19 different feature algorithms succeeding in 73 combinations when wrapped in the proposed wrapper. Furthermore, the proposed

approach operates robustly in the presence of input depth noise, even that of low-cost commodity depth sensors, and well beyond. Secondly, the proposed correspondences voting scheme scored $97.0\% \pm 12.9\%$ on the PR (precision-recall)-AUC (area under curve) metric measuring the correctly scored correspondences in average of all the experiments, while the two state-of-the-art schemes scored $74.2\% \pm 22.2\%$ and $78.3\% \pm 26.4\%$ respectively. Furthermore, the proposed scheme did not require more than $41.5\% \pm 12.5\%$ of the time consumed by the fastest state-of-the-art scheme. Likewise, the proposed voting scheme also demonstrated high robustness against occlusions and scarce inliers. Thirdly, the multi-structure hypothesis generation method outperformed famous multi-structure hypothesis generation methods in hypotheses precision, points recall, and speed aspects. Quantitatively, with 750 generated hypotheses and inliers rate as low as 3.5% on a total of 50 experiments, the proposed method scored $69.6\% \pm 11.5\%$ precision and $69.1\% \pm 5.4\%$ recall, while the remaining methods scored no higher than 12% on both metrics. Furthermore, the proposed method did not require more than 6.45 ms per each hypotheses, which is about 1.16% of the time required by a sophisticated method, and about 268% of the time required by random hypothesis generation, the most simple method. Importantly, the balance between precision and recall, as well as the execution time is totally controllable in the proposed method, as it performs in a progressive manner, with the first generated hypotheses having high precision probability. By limiting the generated hypotheses to 200 in the above mentioned experiments, the execution time reduces to one fourth, while the precision and recall of proposed method becomes $93.9\% \pm 4.2\%$ and $32.4\% \pm 8.0\%$, respectively, in compare to $6.6\% \pm 2.1\%$ and $3.8\% \pm 1.0\%$ for the best compared method.

List of Publications

Publications by the Candidate Relevant to the Thesis

1. H. Sahloul, S. Shirafuji, and J. Ota, “3D affine: An embedding of local image features for viewpoint invariance using RGB-D sensor data,” *Sensors*, vol. 19, no. 2, pp. 291.1–191.32, 2019 Jan., ISSN: 1424-8220. DOI: 10.3390/s19020291
2. H. Sahloul, S. Shirafuji, and J. Ota, “An accurate and efficient voting scheme for maximally all-inlier correspondences set,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (in revision)*, 2019 Apr.
3. H. Sahloul, S. Shirafuji, and J. Ota, “Multi-structure rigid-body geometric fitting in the presence of high outliers rate,” (*draft*), 2019 Jul.

Chapter 1

Introduction

3D reconstruction aims at creating a three dimensional representation of observed object(s) and possibly its surrounding environment (see Figure 1-1). It is a reversed process in compare to the standard CAD (computer aided design), for which in CAD, a design is created in the computer, and then a physical object is realized accordingly. While in 3D reconstruction, an existing physical object/scene is measured and its 3D shape and possibly the texture are imported to the computer. A 3D reconstructed model of some object usually refers to its shape profile, but can also refers to the object's appearance and texture as well.

1.1 Motivation

3D reconstruction has numerous applications in various fields including medical imaging, computational science, digital media, computer graphics, computer animation, computer vision, and importantly, the robotics field. 3D reconstruction provides both dense maps for robots navigation, as well as object models for manipulation.

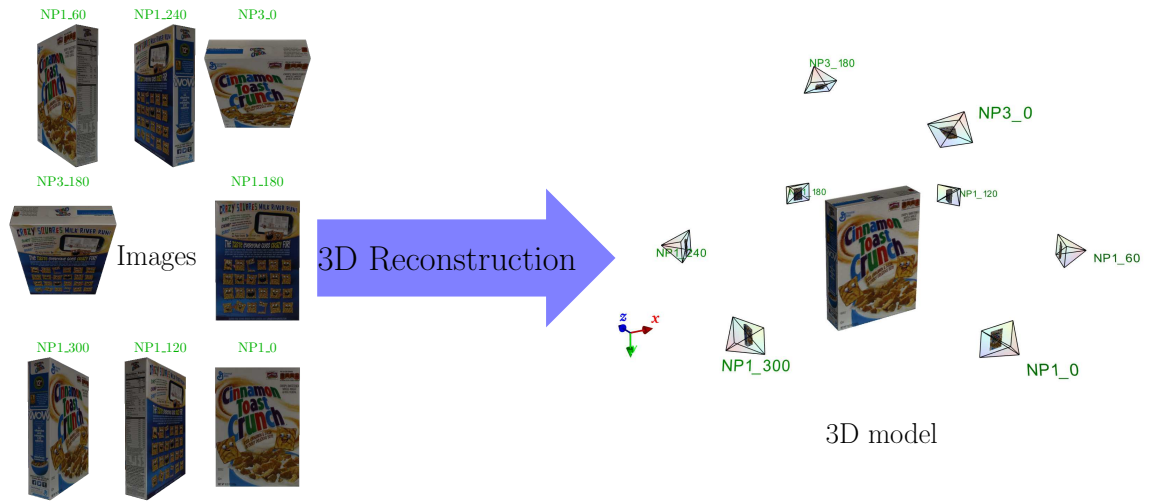


Figure 1-1: What is 3D reconstruction? It is the process of creating 3D models out of some less dimensional data, e.g. 2D or 2.5D images, by recovering their correspondence to estimate their relative poses. The leading label parts (NP1 and NP3) denotes two cameras utilized to capture the ‘C. T. Crunch’ instance of the BigBIRD datasets [5] from two different elevation angles, while the trailing numbers indicate the azimuth angle of the corresponding camera.

Without 3D reconstruction, both autonomous navigation and object manipulation are hardly attainable, especially in dynamic environments.

Nowadays, RFIDs (radio-frequency identifiers) as well as 1D/2D barcodes are widely used to identify objects and enable robotic manipulation, however, these approaches cannot provide sufficient information for the grasping task, for which object's geometry is essential. It seems that the only reason 3D models are not utilized widely these days is that constructing accurate 3D models consumes both time and efforts. What makes the bad worse, in many environments, e.g. online stores' repositories, hundreds of new products may enter or leave the store in daily basis, making it almost impossible to reconstruct such tremendous amount of products within small margin of their cost, or before they even run out-of-stock. Accordingly, the manual labor is the only option for these cases so far. On the other hand, a new paradigm shift of automation might emanate once 3D models are rapidly and cheaply reconstructed with sufficient accuracy.

1.2 Reconstruction from 2D images or RGB-D images?

In the last decade, parallel computations even inside the graphic cards became feasible, which enabled the emergence of a special SfM (structure from motion)-based technique called the VSLAM (visual SLAM (simultaneous localization and mapping)), in which the camera takes numerous shots per each second while moving, where the view changes marginally between any two sequential RGB-D images.

Thanks to this marginal camera pose change, a lot of algorithmic complexities are reduced and the computation became faster, enabling real-time 3D reconstruction. KinectFusion [1] is one of the early RGB-D based VSLAM methods, which gained a lot of attention, and motivated a lot of recent researches. Nonetheless, because of the marginal camera motion, a complete 3D model is impossible, since the bottom parts of objects are not visible. Moreover, without a careful camera motion, environment tracking might get lost, resulting in inaccurate reconstruction. On the other hand, employing MVS (multi-view stereo) techniques with RGB-D images such as [5] can achieve accurate 3D reconstruction since the cameras are intrinsically and extrinsically calibrated. However, 3D model completeness is still a persisting issue for such setups, even when introducing SfM techniques such as turn-tables. One way to achieve complete 3D model is by employing SfM techniques for RGB-D images, as in [6], for which no motion constraints are made on the camera(s) or object(s). Nonetheless, such approaches are generally very slow, and can take long times as 20 min or beyond.

This thesis, however, is focused on a more complicated 3D reconstruction problem, in which the input consists of several RGB-D images involving various same- or different-type objects observed from numerous viewpoints without marginal motion assumption. It can be regarded as a SfM problem only if the input RGB-D images are segmented according to their corresponding captured objects and viewpoints. In such case, virtual cameras can be assumed capturing each segment independently, for which the problem is to estimate their relative camera poses. Although RGB-D data are being utilized, SfM techniques are indispensable to tackle this problem. For that, a lot of techniques are borrowed from older 3D reconstruction techniques, which

were originally developed for 2D images, but can be adapted to RGB-D images as well.

1.3 Methods of 3D Reconstruction

3D reconstruction can be achieved from various image and non-image based techniques, active and passive methods. For instance, the ultrasonic devices of many medical facilities and underwater vehicles are utilized to reconstruct the surfaces of some targeted object(s) to diagnose internal origins, or to observe surrounding environment and to avoid possible collisions, respectively. This is achieved by transmitting mechanical waves that gets partially reflected on some surface, e.g. sea floor or internal organs, and back to a sensing probe, and through utilizing the Doppler effect, the depth of the objects are calculated. Another example of non-image techniques is a depth gauge that maintains contact with a physical object rotating over a roundtable.

Examples of image-based methods include the radiometric methods with laser beams or infrared pattern, which are very common in robotic fields. For instance, an infrared projector can be used to draw some patterns on the surface of the object(s) with an infrared camera to capture its reflection. The scale of the patterns and the curvatures of them give some information about how far the object from the camera plane is, and how its surface look like. This technique is in fact the principle of depth sensors that got widespread recently. These sensors provide depth maps of the observed scene, with intensity images that can additionally provide texture information. Monocular camera is another example that passively observe a given

object in order to recover its shape. This can be based on shedding pattern, camera motion around an object, and much more techniques. A good example is also the use of binocular stereo cameras to construct a disparity map of the scene, which can be used to describe the depth of the objects, and hence reconstruct them.

Generally, there are three main reconstruction techniques: MVS, SfM, or their hybrid approaches. In MVS, the cameras are fixed in the environment, and their intrinsic and extrinsic parameters are predetermined. Therefore, it can be thought of as reconstruction from two or more images of known viewpoints. On the other hand, the camera is free to move in SfM approach, in which camera relative poses between different shots need to be inferred from the observations. More generally, images of different viewpoints can also be taken from different cameras, with unknown intrinsic parameters. In hybrid approaches, the relative poses are estimated using SfM, and then the problem is handled as a MVS. All these approaches were not motivated by real-time applications, until the emergence of VSLAM.

1.3.1 MVS

MVS is a generalization of stereo vision, which is motivated by the human stereopsis system that enable us to perceive 3D object and scenes. It utilizes stereo correspondence as its visual cue, within a system of two or more cameras with predetermined relative poses to each camera. Accordingly, MVS is more generic than standard stereo cameras, in which an object can be even enclosed between its cameras, with an arbitrary number of 2D or 2.5D images from different views.

Reconstruction from MVS is the problem of recovering the shape from two or more different cameras which are intrinsically and extrinsically calibrated. Some

MVS are dedicated to a special stereo setup, such as a multi-baseline [7] stereo cameras, while many are for any arbitrary setup. In MVS, stereo correspondence is used to recover the 3D position of the real world points by the epipolar constraint and triangulation. Although some researchers categorize MVS into four classes [8], two major methods are recognized widely: global MVS and local MVS.

Global MVS

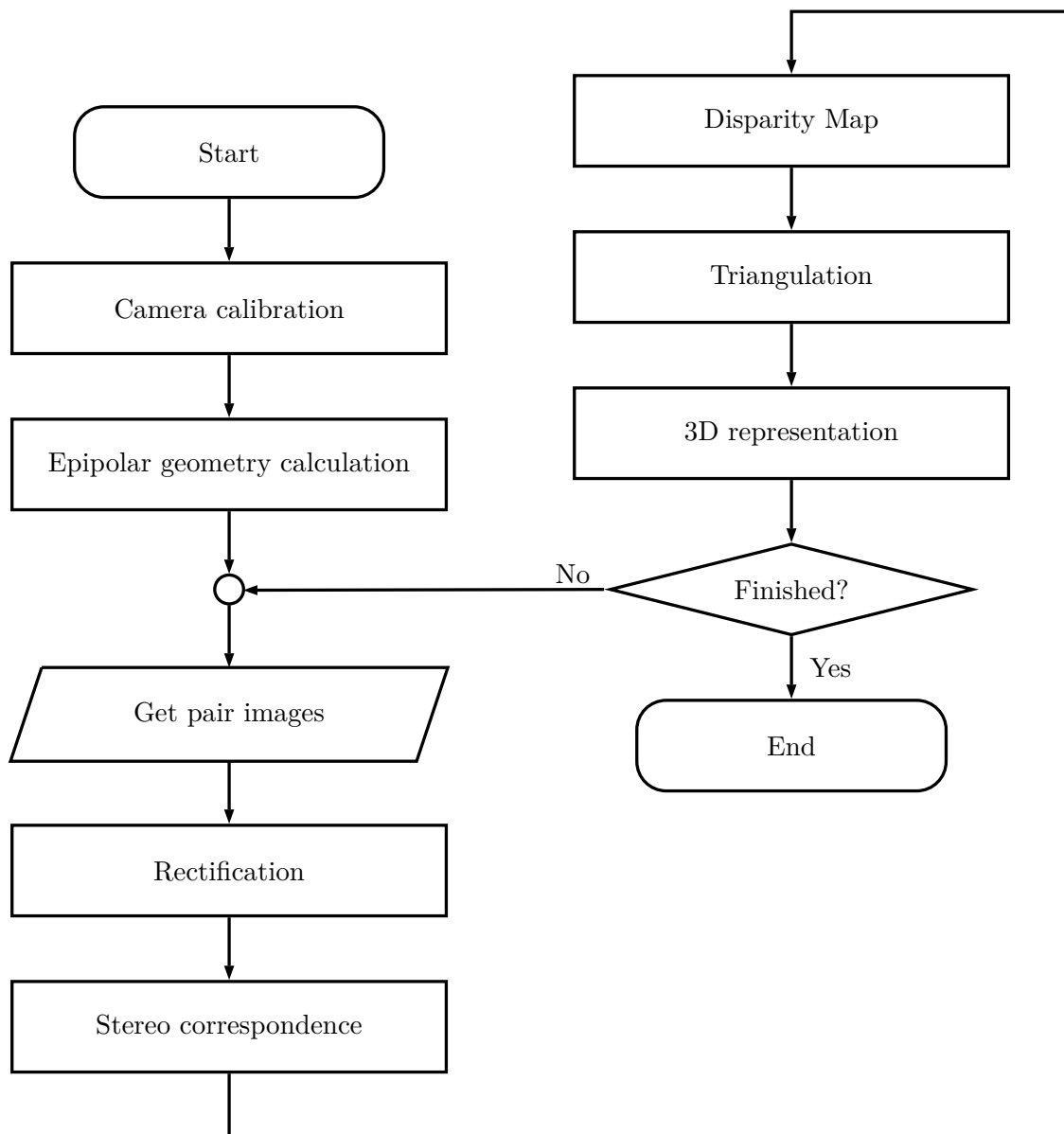
Global MVS usually utilizes a volumetric representation and iteratively recover the shape using photo-consistency and space carving. This class of methods is robust against lacking surfaces and texture, and it can generate complete (watertight) meshes easily.

Local MVS

Local MVS constructs disparity maps from the stereo correspondences then triangulating for the 3D points is a common technique [9] of 3D reconstruction. This class of methods is robust against surface curvatures and achieves accurate results on ideal datasets. The overall flowchart of the framework is shown in Figure 1-2.

Hybrid MVS and SfM approaches

When the extrinsic calibration parameters are unknown, SfM is a handy tool. However, it only makes sense when SfM is utilized to obtain the relative pose between two scenes, and then MVS techniques (such as stereo matching) are exploited. A good example that follows the above logic is the MVS for community photo collections [10]. They collected 2D images via Flickr images web service, and then

**Figure 1-2:** The local MVS framework.

calibrated the images intrinsically and extrinsically using EXIF (exchangeable image format) tags and SfM respectively. Finally, they recovered the shapes utilizing MVS reconstruction.

In fact, numerous and various MVS researches have been proposed, however, they are not strictly MVS approaches. That is, although MVS is extrinsically calibrated by definition, some researches adapt SfM techniques under the MVS titles. For example, stratified reconstruction is considered a MVS approach([11], section 10.4), although it does not assume the calibration of the cameras, and performs projective, affine, and/or similarity reconstructions. Another example is the patch-based MVS [12] which utilizes 2D image features in order to establish correspondence between their cameras.

1.3.2 SfM

Structure-from- X

Structure-from- X or Shape-from- X , where X stands for “visual cues” in which 3D shape can be reconstructed from, are very well studied techniques. Examples of visual cues can be motion [11, 13–15], shading [16, 17], texture [18, 19], focus [20], specular highlights [21], shadows [22, 23], silhouettes [24], symmetry [25] ...etc.

The focus is made here on the SfM due to its fame and relation to stereo reconstruction and SLAM (Section 1.3.3) in general. SfM is one of the photometric range imaging techniques for 3D reconstruction from unordered set of 2D images by utilizing motion cues. These 2D images are from different perspectives around an object and can be from different cameras as well. SfM can be seen as being more

generalized than MVS problem, in which calibration is not required. To estimate camera motion and parameters accurately, SfM establishes correspondence between input 2D images by computing their features and matching them. In the last stage of SfM, BA (bundle adjustment) is performed to optimize for the motion poses and camera parameters. After that, the problem can be reduced into MVS.

SfM has been scaled to the planet-level, and some researchers have demonstrated that by reconstructing big portions of the world from Google street photos and Yahoo webscope respectively [26, 27]. Although, global SfM has low or no-time constraints, incremental SfM can achieve real-time execution.

SfM either processes the whole offline input in a global optimization, or incrementally fuse newly online input as it is received. Incremental SfM differs from the standard SfM in which its input set of 2D images is an ordered set captured by the same camera.

Global SfM

The global SfM computes the 2D features of all input 2D images, and then matches them to find the nearest similar pairs. Starting from a reference 2D image, pairs of 2D features are used in estimating coarse relative viewpoint of the rest of 2D images. The shape is constructed by jointly optimizing the coarse relative viewpoints of the 2D images and 3D points of the detected 2D features. The flowchart is shown in Figure 1-3.

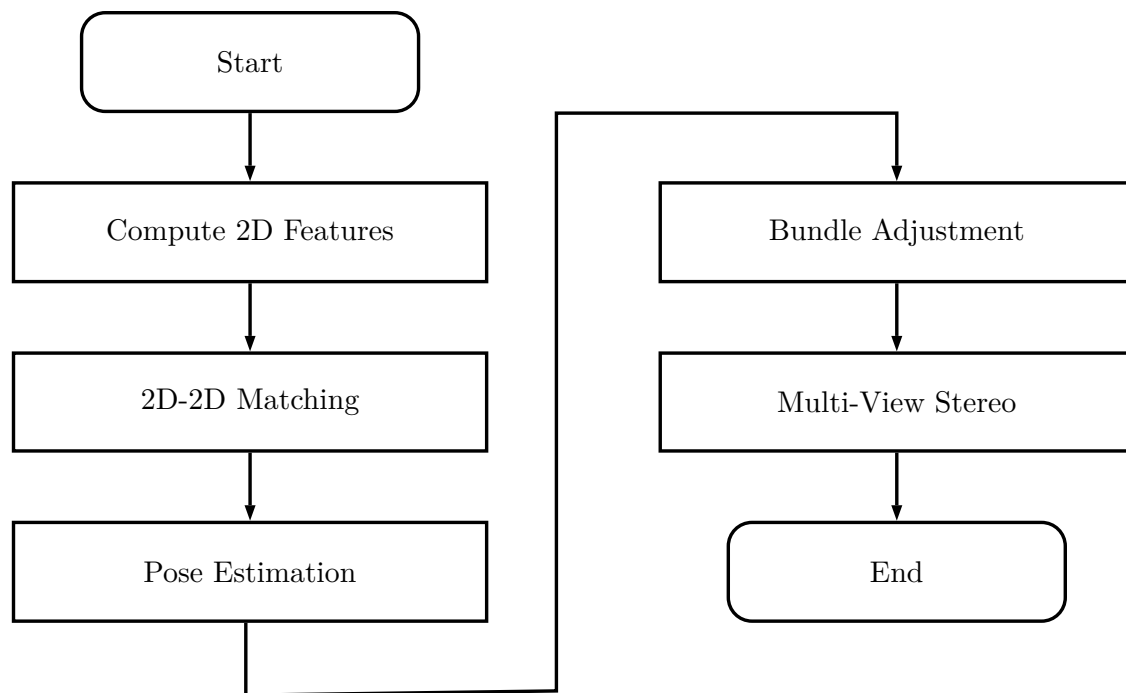


Figure 1-3: The global SfM framework.

Incremental SfM

Sequential/Incremental SfM is handy replacement to global SfM by sacrificing some accuracy if time budget is limited. Since the stream of the 2D images captured by a camera are taken in a fixed frequency, many features are assumed intersecting between any consecutive pair of 2D images. The idea is based on 2D tracking of the being reconstructed 3D structure. That is, in addition to the recovered pose, the algorithm returns a set of “track” points visible in the current input 2D image. Any newly input 2D image is matched against these 2D tracks to localize its neighboring 2D images and match it against them. This contributes dramatically in reducing the search requirements for nearby 2D images in the standard SfM. Further, motion-only bundle adjustment is needed. While it is not part of the sequential/incremental SfM, bundle adjustment can be performed at k^{th} 2D image to prevent drift (Figure 1-4). Hence, incremental SfM gained more speedup, however, it remained far from real-time constraint.

1.3.3 VSLAM

SLAM addresses both problems of reconstructing the environment map as well as localizing the current camera pose in that map [28–30]. In the mapping phase, relations between collected visual data about targeted scene or object are established, while in the localization phase the sensor pose related to the obtained data is estimated. Knowing sensor pose is crucial to properly estimate map relations, and building correct map is the key for localization. This was also referred to as CLM (concurrent localization and mapping) [31]. Although SLAM dose not perform be-

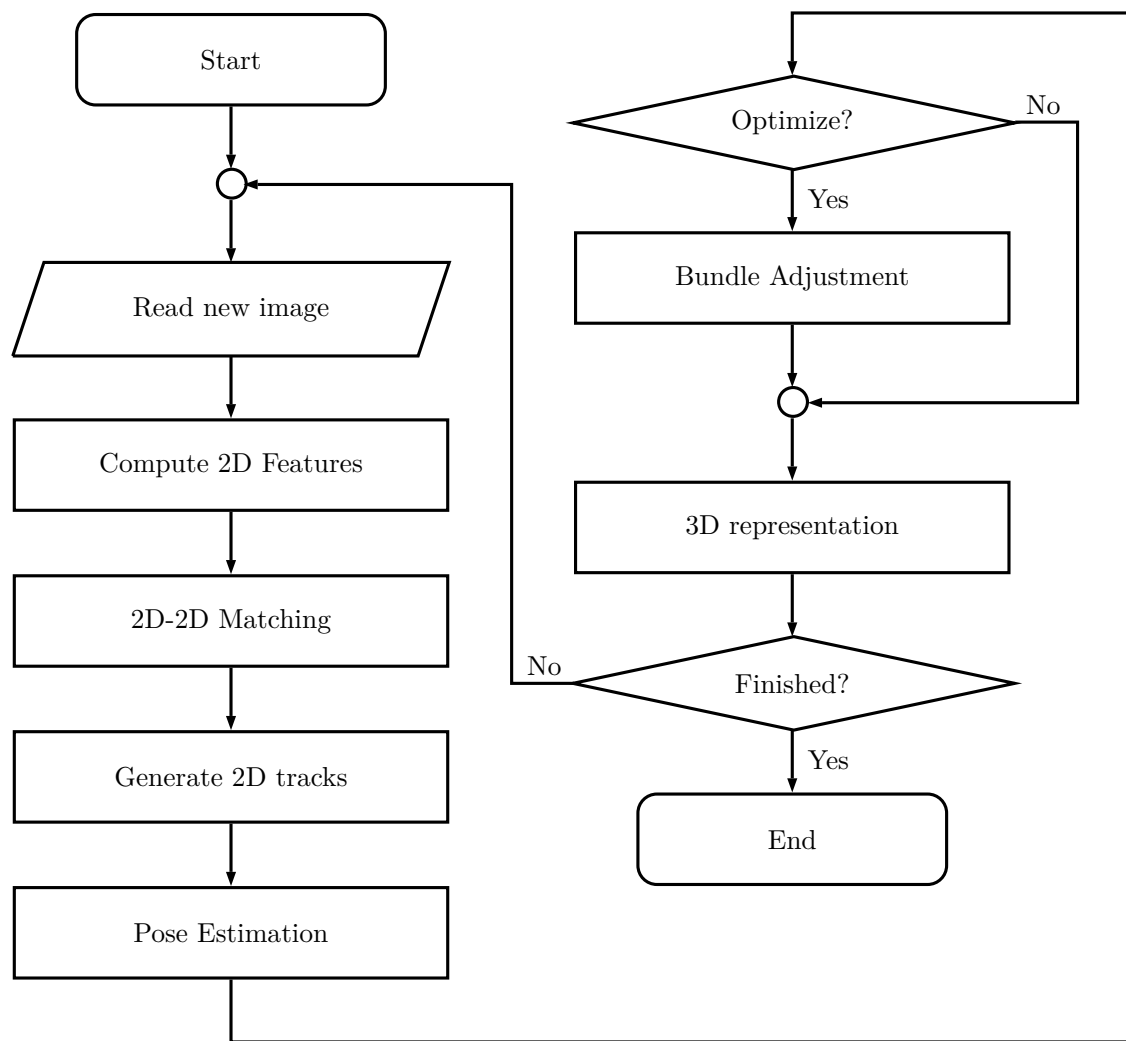


Figure 1-4: The sequential/incremental SfM framework.

yond the scale of few kilometers, it exhibits higher framerate approaching or arriving to real-time execution. VSLAM utilizes radiometry for its input sensors, and can be classified as either sparse or dense algorithms. Sparse VSLAM uses only a small subset of input pixels, hence the map is a coarse representation of the environment. In contrast, dense VSLAM uses the whole input pixels, therefore it provides more complete representation of the environment. Accordingly, when SLAM is mentioned in 3D reconstruction context, usually the dense type is implied.

SLAM can be further classified as three binary categories: direct and indirect, dimensionless and dimensional mapping, or, more importantly, filtering and smoothing methods. Direct methods use pixels directly to register an RGB-D image to the map, while indirect methods use features instead for the registration. Dimensionless mapping is performed when the depths of the points are not observed directly such as in the case of monocular SLAM. This is due to the fact that all the measurements are done in 2D, hence they are related to the real-world measurements up to an unknown scale factor. On the other hand, dimensional methods are able to obtain the depths of observed points, thanks to binocular stereo cameras and depth sensors (commonly known as RGB-D (RGB trichromatic color image and per-pixel depth map) cameras). In the filtering type, the estimation of current sensor pose depends only on current input, the map and the last sensor pose. Due to its incremental behavior, methods belonging to this type are alternatively categorized as online SLAM methods. On the other hand, methods that requires numerous previous sensor poses in order to estimate current pose belong to the smoothing type. It usually formulates the problem as a graph-based optimization when loops are detected to compensate for drift and backtrack this compensation to previous poses. While this type was

considered offline SLAM, advancement in computation techniques and power places it among the realtime methods with good balance between speed and accuracy.

1.4 Types of 3D Reconstruction

3D reconstruction problem can be studied in terms of its inputs relations, in which three types emerge: 1-1 (one-to-one), 1-M (one-to-many), and M-M (many-to-many). To digest these relations, the M-M correspondence is considered first, which is the most complex one, and reduce it down to the remaining two types. Although this categorization has no relation to the 3D reconstruction methods themselves, a SfM method is assumed while discussing methodological technical details.

Consider the two laser scans shown in Figure 1-5a, which represent the inputs to a 3D reconstruction method. Since these scenes have no texture, usually they are uniformly down-sampled to represent their interesting geometric keypoints (Figure 1-5b), which are described using their intrinsic attributes, e.g. via the FPFH (fast point feature histograms) features [32], and their descriptions are matched to form the putative correspondences (Figure 1-5c). While these procedures are generally followed in all the three types of correspondence, the problem is more involved in the case of M-M or scene-scene correspondence. First, since each scene has several objects in it, estimating the pairwise correspondences between such scenes requires multi-structure geometric fitting. That is, each scene is consisting of multiple rigid-bodies, and each of which has its own rigid-body transformation to superimpose its view in the other scene. Second, even when some multi-structure hypothesis is developed (Figure 1-5d), finding the appropriate boundaries of each object to segment

it automatically from its scene is very challenging. That is, without segmentation, superimposition of any two scenes would not only result in just the object model registration, but also its surroundings, and in the case of M-M correspondences, it is unfeasible to reconstruct all the scene objects without segmentation. In fact, as far as the knowledge goes, it seems impossible to automatically segment some static scenes by depending solely on M-M correspondences, without geometry or texture assumptions.

If by some mean, one of the scenes in a M-M relationship is properly segmented, then each segment will have a 1-M (also called view-scene, or model-scene) relationship with the other unsegmented scene, as shown in Figure 1-6. By this, the problem gets more relaxed as further segmentation is not mandatory if the object surroundings are allowed to exist in the reconstructed model. Instead, the concern is the stability of each superimposition hypotheses to register each model to its related scene. While in some cases this might not be so challenging, e.g. as shown in Figures 1-6a and 1-6d, most of the cases suffer from scarce inliers, mainly due to clutter and occlusions, such as the shown cases in Figures 1-6b and 1-6c.

Similarly, the third type of these relationships is the 1-1 correspondence, which is obtained by properly segmenting the object view from the remaining scene in the 1-M correspondence, as show in Figure 1-7. In such cases, model-only 3D reconstruction is straightforward once a superimposition hypothesis is developed. However, the quality of correspondences in these relations are mostly affected by the indistinct feature descriptions due to similarities in geometry or texture, or more importantly, due to large viewpoint differences.

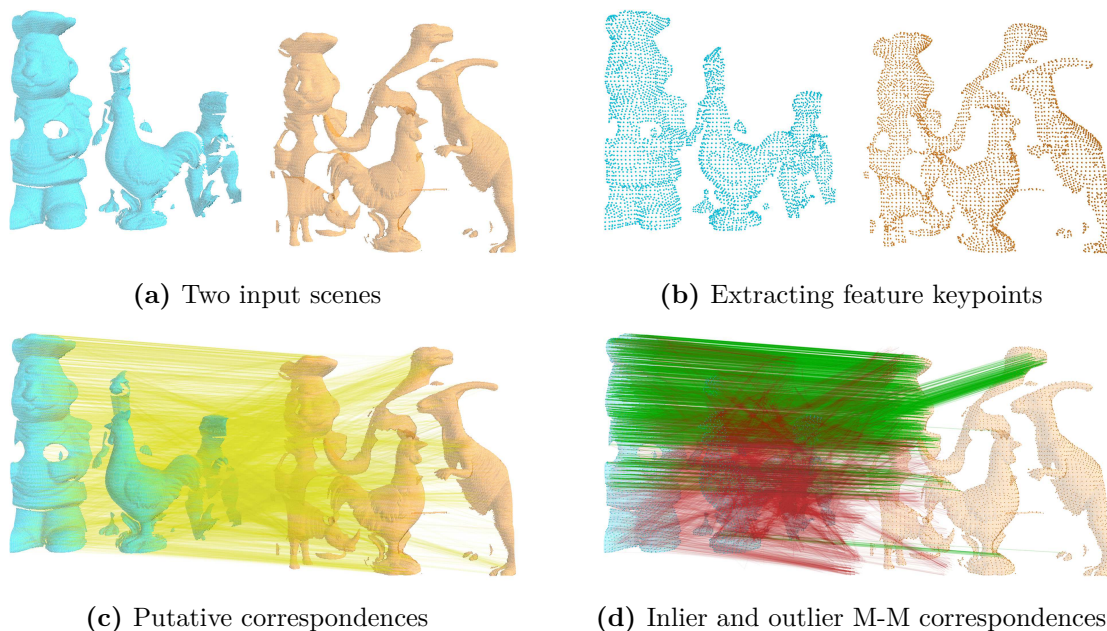


Figure 1-5: Computing putative correspondences and sample M-M correspondences. In order to estimate the correspondences, features of each input are usually computed by detecting some interesting keypoints on the texture representation, or by sampling the geometric representation uniformly, as in this shown case. After that, matching these features pairwise results in the putative correspondences (the yellow lines). Each of the shown scenes contains several objects, and the task is to match each object's views together, and hence the problem is a M-M correspondences estimation, for which the expected outcome is the inlier correspondences (the green lines) that form a multi-structure geometric fitting hypothesis. In 3D reconstruction of M-M correspondences, however, matching alone is not sufficient, as superimposition of the matching views is also required, for which segmentation is a necessity to obtain consistent representation of each individual object. The shown scenes are the RS10 and RS35 of the U3OR (UWA 3D object recognition) dataset [33, 34], which are assigned different colors for distinction purposes.

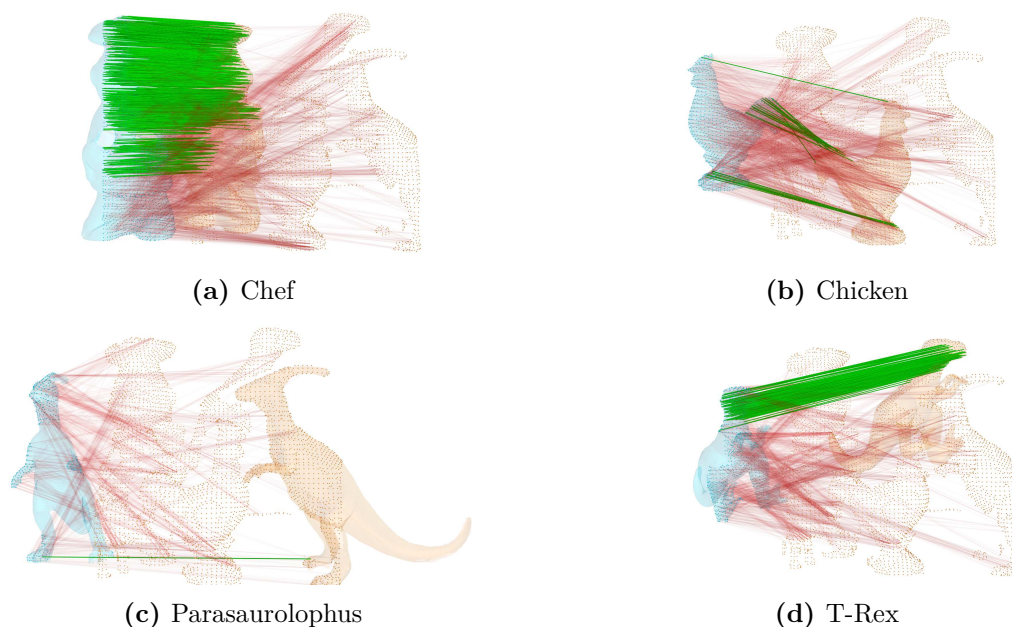


Figure 1-6: Sample 1-M correspondences, obtained by segmenting the left scene of Figure 1-5. In this case, each 1-M relation requires estimating its own single-structure rigid-body transformation from the inlier correspondences (shown in green), to superimpose the object view in the other scene. Segmentation here is not the main concern, as resulting models without it would not suffer inconsistency issues, but they would include their surrounding environment as well. The biggest concern of 1-M relations is the appropriate superimposition hypothesis estimation, obtained by rejecting outliers and recovering an all-inlier correspondences set.

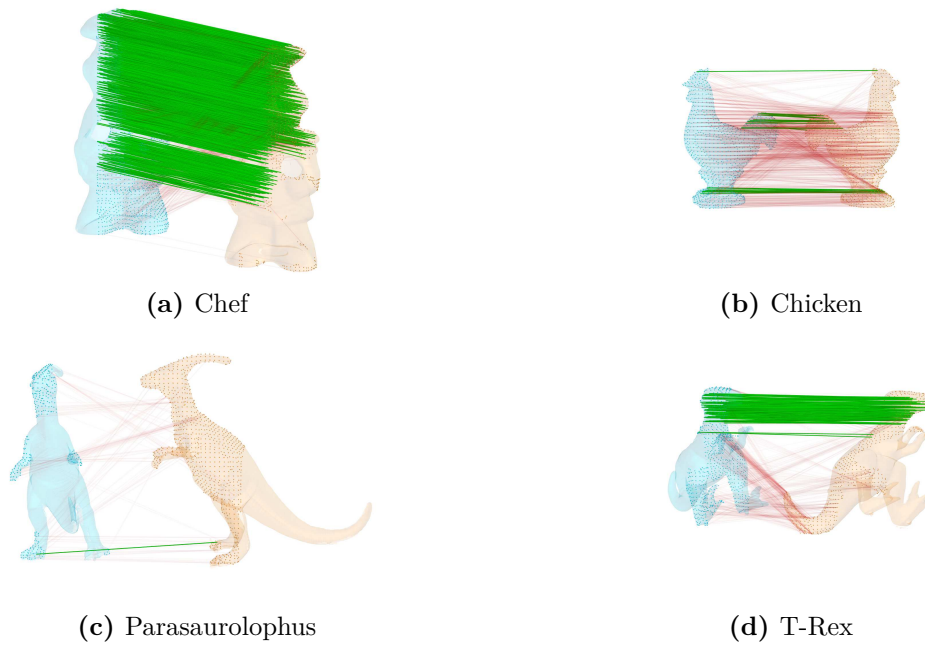


Figure 1-7: Sample 1-1 correspondences, obtained by segmenting the right scene of Figure 1-6. In this case, inlier correspondences (shown in green) are mostly affected by the viewpoint difference and the distinctiveness of the feature descriptions.

1.5 Purpose and Problem Statement

The balance between the reconstruction quality and the consumed time is important. MVS and VSLAM reconstruction algorithms constrain camera motion by some assumptions, preventing it from capturing object whole surfaces, e.g. the bottom parts of the objects, thus affects the overall model quality. On the other hand, by seeking coverage of more surfaces, the consumed time is dramatically increases. Even by stitching multiple reconstructions generated by MVS or VSLAM, the time consumed triples easily, due to the additional stitching overhead as well as the increase in the capture time. To reconstruct a model covering all surfaces, the most prevalent approach is to perform SfM on a collection of 2D or RGB-D images capturing the object from different sides. Nonetheless, this approach has limited applicability due to its computation time that correlates exponentially with the number of utilized images. Although the execution time can be reduced by decreasing the number of images, the problem is not always as easy as carefully selecting the viewpoints to cover the object whole surfaces, since the overlapping regions between captured images shrinks due to the increase in the viewpoint difference between camera poses (Figure 1-8), which affects the correspondence estimation (Figure 1-9).

Another issue is the uncertainties of the pairwise relationships between the inputs, i.e. the putative *correspondences*, that arise due to locality of measurements, similarities in geometry and texture, or ambiguities steaming from clutter and occlusions. All of these issues increase the correspondence outliers rate, causing a combinatorial search explosion while seeking a hypothesis that is maximally-supported by consistent correspondence inliers. Currently, there are either slow and complex

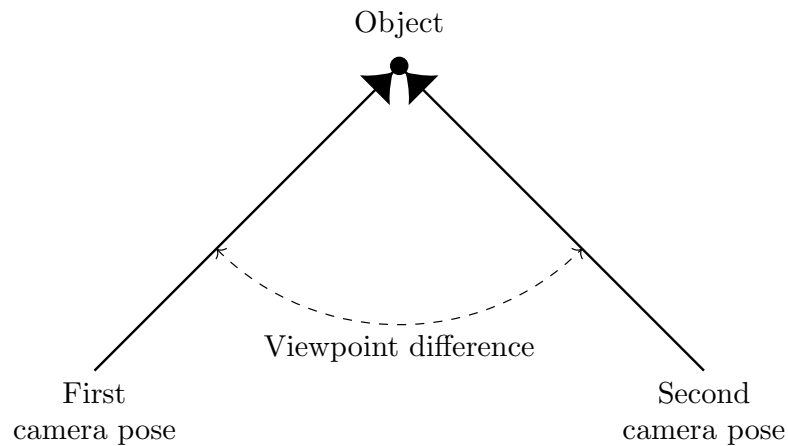


Figure 1-8: The concept of viewpoint difference. The more the difference between two viewpoints, the less images required to cover all the object sides, but the less robust the reconstruction since overlapping regions within corresponding images decrease as well.

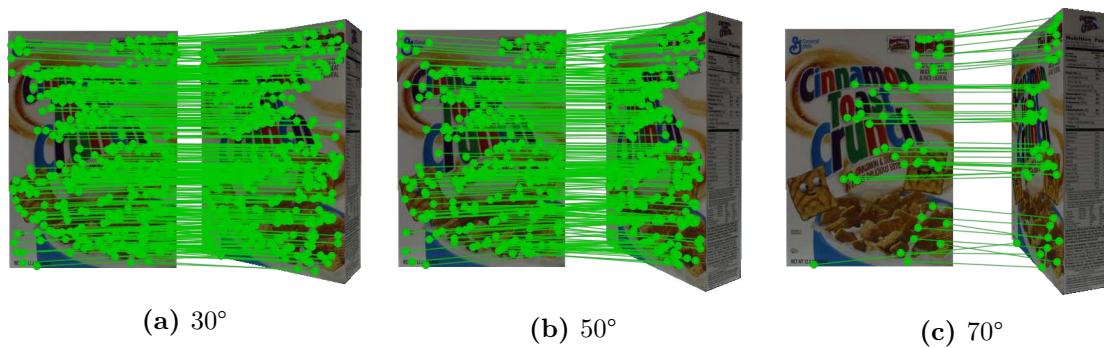


Figure 1-9: The relation between viewpoint difference and correspondence estimation. Generally, the larger the viewpoint difference (measured in degrees) between two inputs (left and right shapes of each sub-figure showing a rotating cereal box), the harder it gets to estimate their correspondences (the green lines). However, the maximal viewpoint difference, i.e. the viewpoint invariance, is an algorithm-specific limit, which depends on the repeatability and distinctiveness of its features (the green dots).

techniques to recover abundant inliers, or fast but low-recall techniques to recover a minimal inliers set [35]. Some putative, outlier, and inlier correspondences are shown in Figure 1-10 for graphical demonstration.

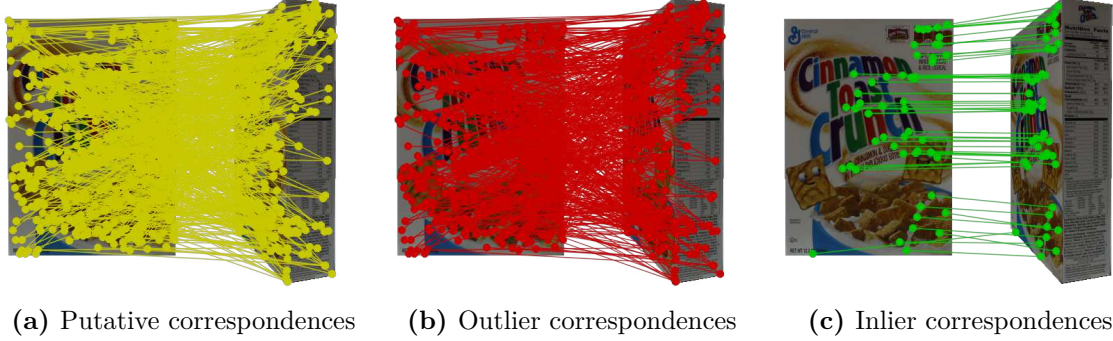


Figure 1-10: Putative, outlier, and inlier correspondences. Putative correspondences are the raw outcomes of features matching, which need to be filtered by rejecting outliers subset that might constitutes most of its elements, in order to obtain an all-inlier correspondences set.

A third issue is the segmentation of the object model from its surrounding environment, where it is conventionally performed either manually, or based on motion, geometric, or texture assumptions. An example of automatic segmentation, a motion-based technique is shown in Figure 1-11.

In summary, previous studies have not reconstructed a model covering all the object surfaces and segmented it within reasonable time. They either cannot capture the whole shape, e.g. the bottom parts, or consume a lot of time setting-up the environment, capturing the RGB-D images, or performing the reconstruction. Another issue is that, a lot of information is sacrificed while searching for minimal inliers set, and on the other hand, substantial time is required to seek a maximal inliers set, while balance is hard to achieve. Furthermore, the segmentation process

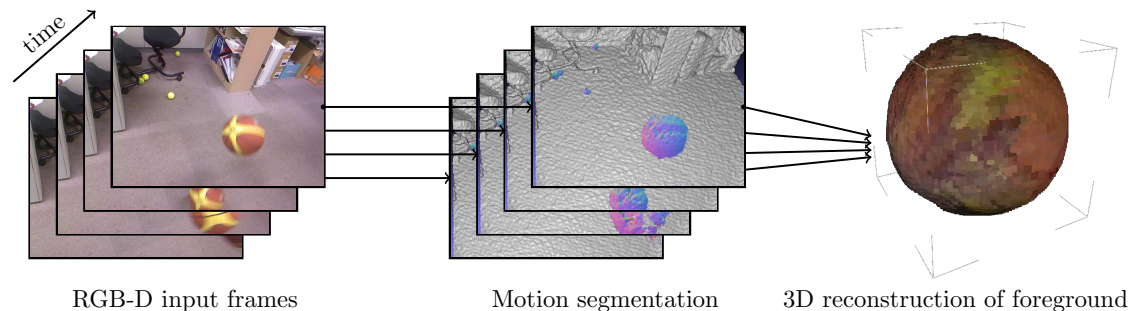


Figure 1-11: An example of model segmentation. In several occasions, the interest is set only on constructing a 3D model for a specific object in the environment. To automatically segment an object, some assumptions are usually made about its motion, its texture, background color or geometry ...etc. The shown example is a motion-based segmentation of a bouncing basketball, obtained from [36] (author’s master thesis).

usually involves some assumption about the environment geometry or texture, object motion, or it is just performed manually.

Accordingly, the goal is set to simultaneously and automatically reconstruct objects’ models and segment them from their environment within reasonable time. It seems model reconstruction for industrial purposes can even approach an hour in some cases, which renders the whole process infeasible for production purposes. For this reason, ten minutes are set as a reasonable time limit in this research. See Figure 1-12 for the system block diagram.

1.6 Challenges and Approaches

As explained previously in Section 1.5, in order to simultaneously and automatically reconstruct objects’ models and segment them from their environments within reasonable time, there are three issues that needs to be addressed. First, there is

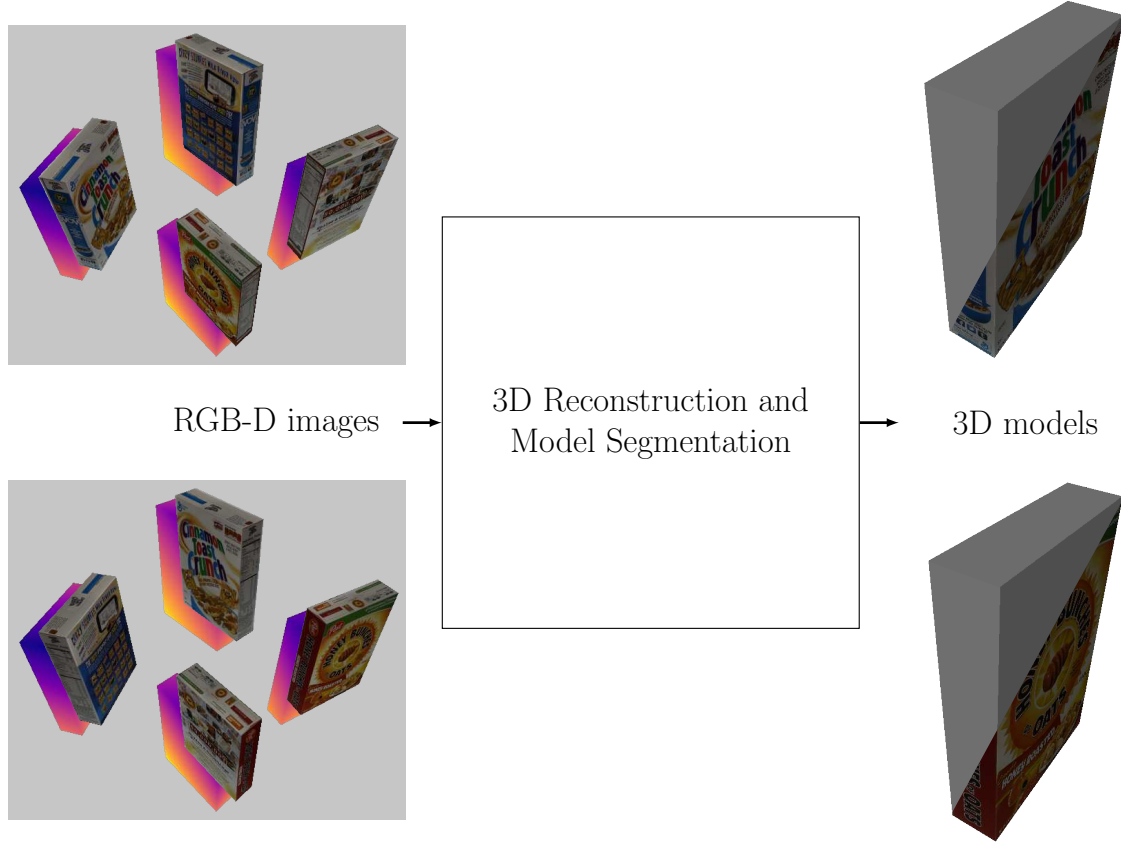


Figure 1-12: 3D reconstruction system block diagram. The input RGB-D images (depth is depicted as a heat map shown behind each RGB image) include different objects with different viewpoints. These inputs are matched together, and their object-wise relative poses are estimated from their correspondences. Furthermore, the objects' models are segmented by clustering the multi-structure hypotheses, which acts as feature-space for the corresponding points. As an outcome, 3D models of the input objects are simultaneously reconstructed (geometry and texture are shown in upper and lower halves of each model, respectively).

a need to ensure a good balance between the constructed model quality and the time consumed while capturing its RGB-D images and constructing them. Previous methods cannot cover all object sides, thus affect the model quality, or otherwise exponential time is consumed. Second, the problem of correspondence uncertainties causes another accuracy-time trade off, for which previous studies either sacrifice valuable information by returning minimal inlier correspondences set, or consumed a large portion of time while seeking a maximal all-inlier correspondences set. Third, the automatic segmentation problem needs to be solved without geometry, texture, or motion assumptions in order to decouple the scenes observed in the RGB-D input images. In this section, the approaches to these three challenges are presented.

In regards to balancing the quality and the computation time, in Chapter 2, the execution time is decreased by minimizing the number of RGB-D images, while improving the underlying algorithms robustness against small overlapping regions. Such algorithms detect interesting keypoints within the inputs, called features, and compute their intrinsic attributes, called description, in order to match two pairs of them together, making a correspondence (see Figure 1-9). However, these algorithms suffer from a relatively small tolerance against viewpoint difference, i.e. *viewpoint invariance*, (see Figure 1-13). By increasing these algorithms' viewpoint invariance, even with small overlapping regions, a robust estimation is possible. The idea is to warp surfaces of similar properties (i.e., *smooth surfaces*) to a viewpoint invariant representation, where the features are extracted. Accordingly, the challenges are to achieve a stable viewpoint invariant representation and to provide a general wrapper that is applicable to any local image detector/descriptor. The approach to viewpoint invariant representation involves labeling stable smooth surfaces by back-

projecting surface-normal clusters that are aggregated using nonparametric spherical k -means, estimating their warp transforms, and then warping the labeled surfaces to a reference local plane using a hybrid rigid-homography method. The generality of the wrapper is guaranteed by introducing preprocessing and post-processing stages, before and after the detection/description processes, to match their standard interfaces. The aforementioned approach forms the preprocessing, and post-processing is to put together all obtained features from the per-surface independent computations by mapping the extracted features to the local frame. See Figure 1-14 for a graphical summary and sample qualitative result of the proposal.

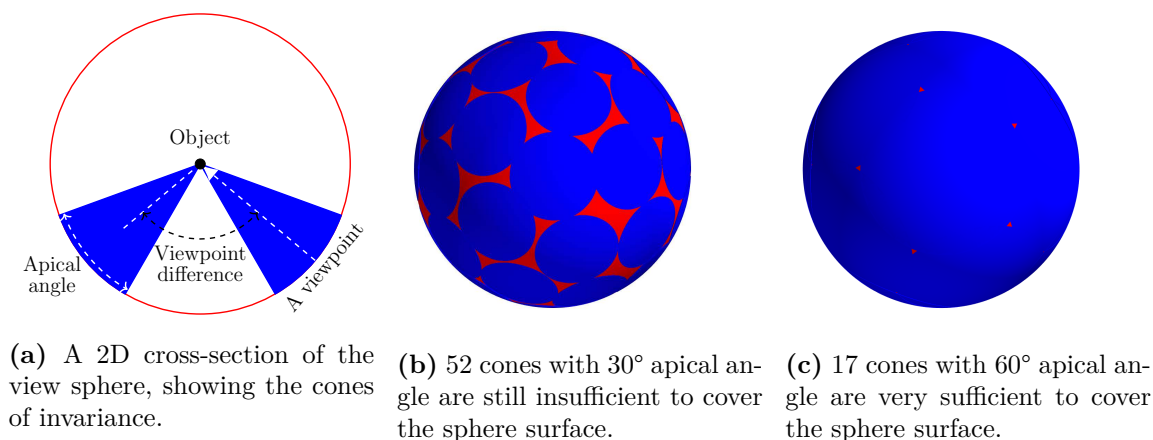


Figure 1-13: The relation between the viewpoint difference, number of inputs, and the reconstruction robustness. Let the viewpoint invariance be represented as the cones' apical angles, and their centers to represent the viewpoint. In that case, a reconstruction from n inputs is analogous to the n cones (shown in blue), after distributing them equally on the sphere surface (shown in red). If some sphere regions are not intersected by the invariance cones, then its viewpoints are distant-apart beyond the viewpoint invariance of the matching algorithms, thus reconstruction robustness is affected.

In regards to the correspondence uncertainties, in Chapter 3, a correspondences

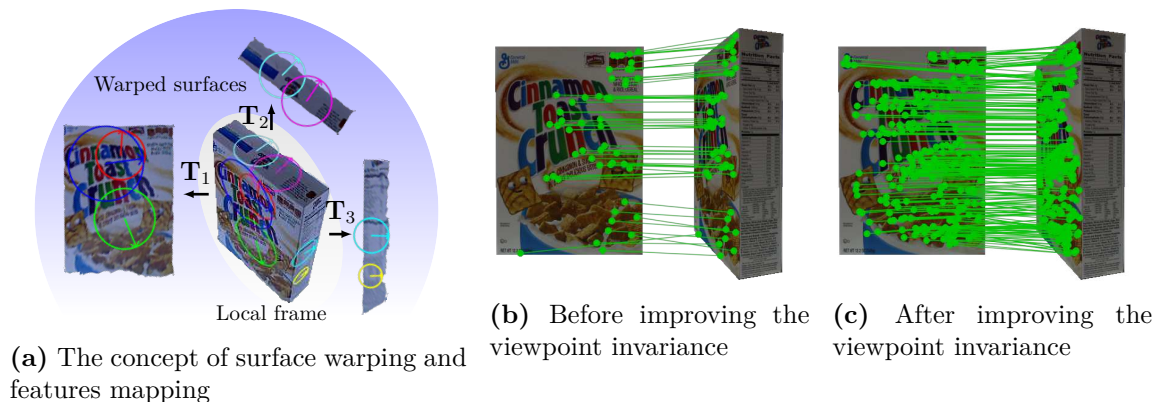


Figure 1-14: A glance into first contribution to increase the viewpoint invariance. Increasing the viewpoint invariance improves the correspondences quality, thus the 3D reconstruction robustness. The shown example is for a cereal box observed from a viewpoint difference of 70° , while the correspondences are estimated using SIFT (scale-invariant feature transform) [37] features before and after improving its viewpoint invariance. Refer to Chapter 2 for details of the proposed method.

voting scheme is proposed in two stages to exploit the local neighborhood rigidity constraint in order to elect a voting set, which is utilized to rank the correspondences according to their likelihood being inliers. Nonetheless, it is challenging to come up with criteria for each stage that maximize the accuracy without affecting the efficiency. The approach for the first voting stage involves utilizing the LRC (local rigidity constraint) to obtain crude inliers ranking scores. Importantly, the strength of the proposal stems from the second voting stage, in which contaminations are minimized in both the global transforms and voting set. To minimize outlier effects without causing high computation footprint, an ambiguity-free and computationally cheap variant of the 1PSTs (single-point superimposition transforms) transforms is computed solely from the voting set, utilized them to revalidate the voting set, and

then ranked the putative correspondences. See Figure 1-15 for a graphical summary and Figure 1-16 for an example qualitative comparison.

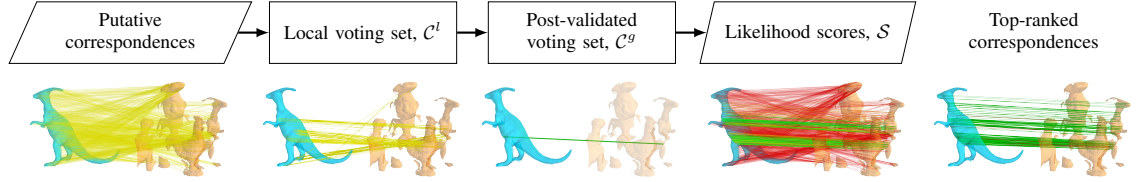


Figure 1-15: A glance into the second voting-based contribution for correspondences filtration. Refer to Chapter 3 for details of the proposed method.

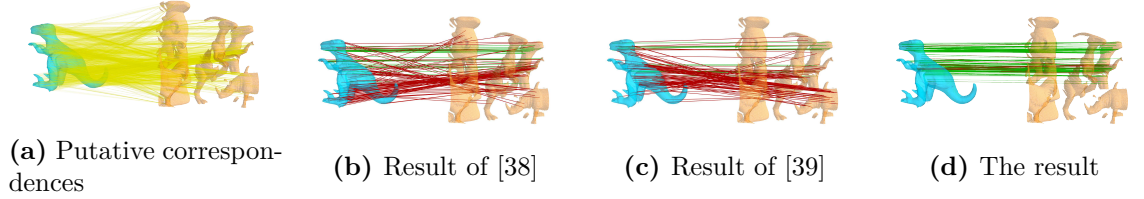


Figure 1-16: A sample result of the proposed voting-base correspondences filtration scheme. Inlier correspondences are shown in green, in which the proposal outperformed both state-of-the-art methods. The putative correspondences have been generated by matching the FPFH features [32].

As for the segmentation issue, in Chapter 4, the clustering of multi-structure hypothesis is considered as a feature-space descriptor for corresponding points. Accordingly, several observed objects in one scene would get geometrically fitted by different hypotheses, and the clusters formed by these hypotheses would result in the spatial-domain segmentation.

1.7 Overall Overview

A global SfM framework is followed, where 2D-related techniques are replaced with their RGB-D, so called 2.5D, or 3D techniques, as shown in Figure 1-17. While the whole framework is shown in the figure, this thesis focuses only on three processes within the overall framework (namely, the features computation, single-, and multi-structure modeling). Initially, the RGB-D images are read, and their keypoints and features are computed. After that, the features are matched, which results in the putative correspondences set. This set is assumed contaminated, i.e. containing outliers, and it cannot be utilized directly in geometric modeling, i.e. pose estimation, until outlier correspondences are excluded. Within the single-structure modeling process, outliers rejection is utilized to filter the correspondences to minimize its contamination. After that, geometric modeling takes place. In case multi-structures exists in the input data, several hypotheses are generated with the aim to maximize their precision, so that they can be clustered in order to segment the underlying multi-structures within the input data. In this case, the hypotheses clusters forms the multi-structure geometric model. After that, global optimization is needed, since estimated poses in the geometric modeling phase are usually inconsistent in the global scale. Finally, the 3D model is generated in the 3D representation process. The contributions are in the features computation phase (Chapter 2), single-structure modeling phase (Chapter 3), and multi-structure modeling phase (Chapter 4).

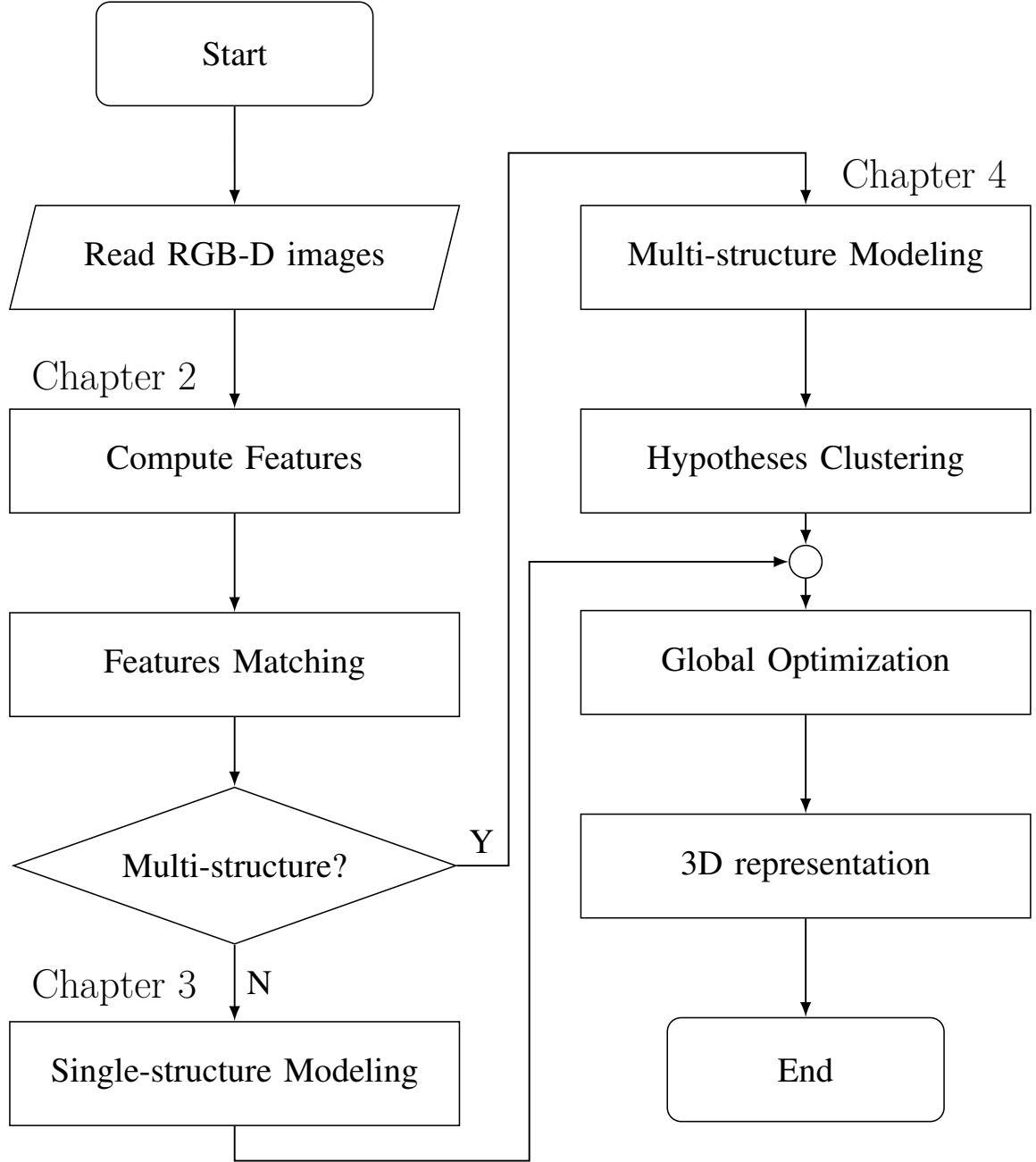


Figure 1-17: Overall overview of the 3D reconstruction framework followed in this thesis. The focus of this thesis is on the features computation, single-structure modeling, and multi-structure modeling. The remaining processes are shown for completeness purposes only.

1.8 Contributions

In the features computation phase, the contribution is the development of a general wrapper that:

- improves the viewpoint invariance of local image features,
- wraps virtually any local image detector/descriptor algorithm, and
- requires no additional interfacing or modifications.

As for correspondences filtration phase, a contribution of correspondence voting scheme is proposed for outliers rejection that is:

- highly accurate and extremely efficient,
- deterministic and rigorously repeatable,
- simple to implement.

In case of multi-structure geometry, a multi-structure rigid-body geometric fitting method is proposed, which is:

- progressively precise and extremely efficient,
- deterministic and rigorously repeatable,
- simple to implement.

1.9 Assumptions and Constraints

In this research, the rigidity of the observed objects is assumed. It is worth noting that no constraints on the camera motion are assumed, to enable covering all object surfaces, including their bottom parts.

1.10 Thesis Organization

Chapter two describes the proposed 3D Affine wrapper to improve viewpoint invariance of local-image features, and presents its quantitative results in improving viewpoint invariance of numerous local-image features.

Chapter three describes the proposed voting scheme and presents its comparative results to the recent state-of-the-art algorithms, while chapter four provides a framework for multi-structure modeling with applications to scene-scene segmentation.

Finally, chapter five concludes the thesis and discusses possible future works.

Chapter 2

A 3D Affine Framework

2.1 Introduction

Local image features are major low-level building blocks in various computer vision and image processing algorithms; however, they have a certain degree of sensitivity to viewpoint difference. In light of the recent growth in the applications of RGB-D sensors, the viewpoint invariance of local image features has been gradually improving [40–43] by utilizing input depth maps. Despite the current trend of using machine learning for low-level local image features [44–50], or even for high-level tasks, such as 6D pose estimation, [51–55], several hand-crafted features are still actively employed in various algorithms. This trend is due to the on-par or better performance of hand-crafted features [56] and their well-established maturity gained from 10–20 years of research. Examples of hand-crafted features include GFTT (good features to track) [57], SIFT (scale-invariant feature transform) [37], and SURF (speeded up robust features) [58]. Nevertheless, these are local intensity-

image approaches, which typically *detect* keypoints and then *describe* their features from 2D image patches (hence the name *local*) with gray or trichromatic intensities *under some geometric assumptions*. Thus, they tend to lack robustness to geometric transformations involving mainly viewpoint difference, being robust up to 25° – 30° [59]. Consequently, issues such as *unrepeatable detection*, *indistinct description*, or *non-covariant keypoints* can arise. These issues essentially originate from the neighborhood perspective changes that accompany out-of-plane rotations, sampling window overlap of a keypoint located near the edges of a surface, or when the surface has different orientation from the viewpoint, respectively. Evidently, Vedaldi *et al.* demonstrated that local image features can achieve viewpoint invariance for generic non-planar scenes without assuming a locally planar scene [60]. Achieving viewpoint invariance will improve the efficiency and robustness of various computer vision *ill-posed problems*, including wide baseline matching, 6D pose estimation (i.e., *rigid body transformation*), 3D reconstruction, recognition by reconstruction, and visual SLAM (simultaneous localization and mapping).

Little research has focused on using geometric information to improve viewpoint invariance. For example, Wu *et al.* proposed a SIFT-based descriptor that utilizes depth maps to construct VIPs (viewpoint invariant patches) [40]. Similar proposed solutions include a SIFT-based detector with a region-affine sampling window [41], and BRISK (binary robust invariant scalable keypoints)-based features with adaptive sampling pattern orientations and scale factors [42, 43]. However, many such local image detectors and descriptors are intended for a particular application [61], and the cited studies targeted specific local image features, thus only covering a small subset of the numerous applications presented by each of the vastly available features.

Several studies have presented general *wrappers*. A wrapper wraps around local image features, independent of their implementation, for the purpose of enhancing certain aspects of the wrapped features, such as distinctiveness [62, 63] or invariance. Specific examples include ASIFT (2D affine SIFT) [64] and ASURF (2D affine SURF) [65], which are local image feature wrappers for 2D affinity invariance and are capable of wrapping features in addition to those described in their original SIFT- and SURF-based proposals. While these wrappers do not succumb to the pitfall of targeting specific applications, existing wrappers depend solely on intensity images and are thus limited to in-plane motion invariance, i.e., the *2D-affine transformations*.

Currently, it seems there is no available 3D-affine wrapper, for which there is a growing need, as the list of local image features continues to grow [37, 44, 45, 57, 58, 62–78], each being tailored for a particular application and having inadequate viewpoint invariance [59]. To this end, a general 3D-affine wrapper is proposed to wrap virtually any chosen *2D* detector/descriptor pair to improve its viewpoint invariance without changing the *invariance formula(s)* or implementation. The aim is not to propose another detector/descriptor, but rather to fill in the gaps and improve existing local image features by enhancing their viewpoint invariance. Briefly, the proposed method aims to achieve viewpoint invariance by: (1) extracting surfaces of similar properties (i.e., *smooth surfaces*) from the scene; (2) representing these surfaces in a *reference local plane* with a viewpoint invariant representation; (3) applying the wrapped detector/descriptor to the viewpoint invariant representation; and (4) mapping the extracted features back to their original input frame, i.e., the *local frame*. This scheme is supposed to address the robustness issues to ge-

ometric transformations, namely unrepeatable detection, indistinct description, and non-covariant keypoints. Ideally, the scheme will result in repeatable detection of near or faraway regions, since each smooth surface is brought to the same reference local plane. Similarly, the description overlap between one surface and another is mitigated because feature detection and description are performed independently per each smooth surface. Finally, features will have an affine-region sampling window that will covariantly change with plane orientation, since they are first computed in the reference local plane and then mapped back to their original frame.

Accordingly, the challenges are to achieve a stable viewpoint invariant representation and to provide a general wrapper that is applicable to any local image detector/descriptor. The approach to viewpoint invariant representation involves labeling stable smooth surfaces by back-projecting surface-normal clusters that are aggregated using nonparametric spherical k -means (Section 2.2.2), estimating their warp transforms (Section 2.2.3), and then warping the labeled surfaces to a reference local plane using a hybrid rigid-homography method (Section 2.2.4). The generality of the wrapper is guaranteed by introducing preprocessing and post-processing stages, before and after the detection/description processes (Section 2.2.5), to match their standard interfaces. The aforementioned approach forms the preprocessing, and post-processing is to put together all obtained features from the per-surface independent computations by mapping the extracted features to the local frame (Section 2.2.6). The main contribution is the development of a general wrapper that:

- improves the viewpoint invariance of local image features,
- wraps virtually any local image detector/descriptor algorithm, and

- requires no additional interfacing or modifications.

The remainder of this chapter is organized as follows. Section 2.2 describes the proposed wrapper, Section 2.3 demonstrates an application of 6D pose estimation, and Section 2.4 explains the experimental setup, the datasets, and the performance metrics that were utilized. The results are discussed in Section 2.5, and Section 2.6 presents conclusions and future work.

2.2 Methodology

The proposed method follows the standard *two-steps scheme*: keypoint detection and feature description from an intensity-image, with pre- and post- processing stages. The preprocessing stage comprises smooth-surface annotation (Section 2.2.2), warp transform estimation (Section 2.2.3), and smooth-surface viewpoint invariant representation (Section 2.2.4). After preprocessing, the features are computed using the local image detector/descriptor pair of choice (Section 2.2.5). Finally, the post-processing stage puts together all the scene features, which are independently extracted from different surfaces, by mapping their corresponding keypoints back to the local frame of the original input (Section 2.2.6), which also provides the appropriate affine-region keypoint representation. See Figure 2-1 for an overview of the approach, and refer to Table 2.1 for a summary of the notation used throughout this chapter.

The proposed method offers both 3D-affine invariance, similar to VIPs [40], and wrapper flexibility (e.g., [64, 65]) that enables wrapping any local image detector/descriptor approach that follows the two-steps scheme. For VIPs, detection is

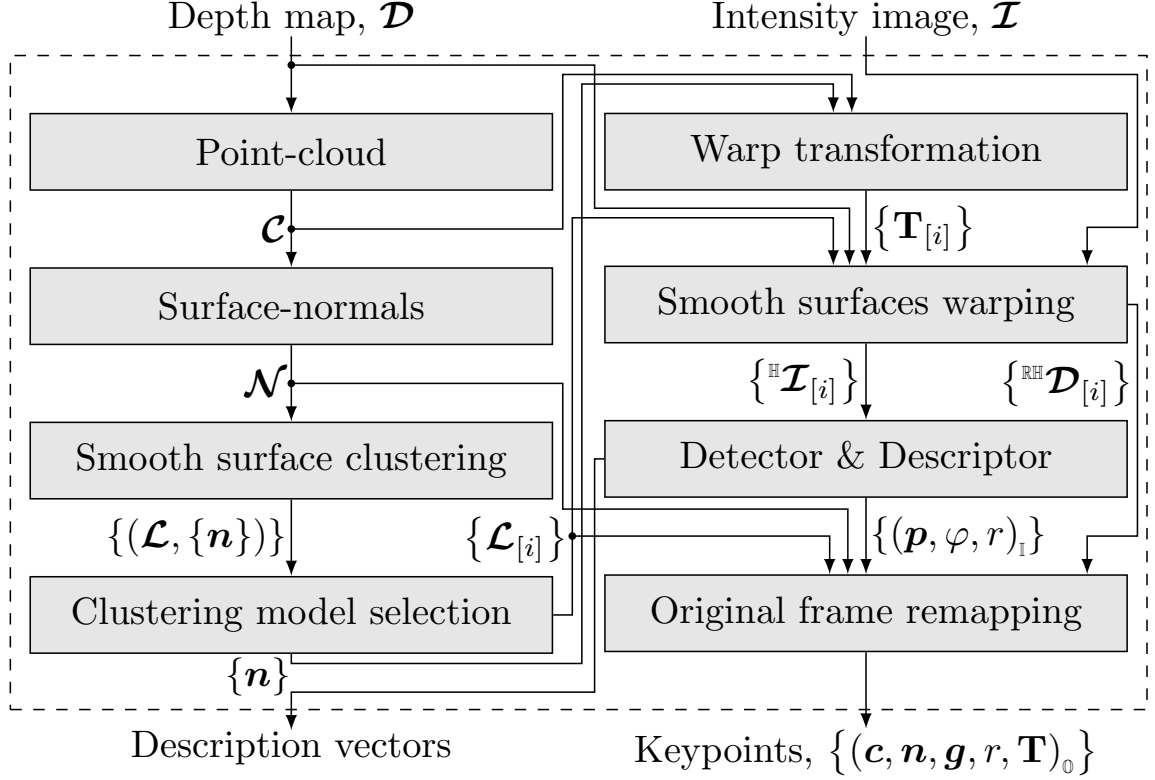


Figure 2-1: An overview of the proposed 3D-affine wrapper pipeline. First, a point-cloud, \mathcal{C} , and the surface normals, \mathcal{N} , are computed from the input depth map, \mathcal{D} (Section 2.2.1). Then, the surface normals are used to form several spherical k -mean modules in parallel, and the best model, $(\mathcal{L}, \{\mathbf{n}_i\})$, is selected to annotate the smooth surfaces in the scene (Section 2.2.2). After that, all smooth surfaces are parameterized by their point-cloud and surface-normal centroids, $\{(\mathbf{c}, \mathbf{n})_i\}$, and their individual warp transforms, $\{\mathbf{T}_i\}$, are computed (Section 2.2.3). Using their corresponding warp transforms, the smooth surfaces are rigidly warped and homographically morphed in parallel into a viewpoint invariant representation, $\{(\mathcal{I}_i^{\mathbb{H}}, \mathcal{D}_i^{\mathbb{RH}})\}$ (Section 2.2.4). Using the proposed 3D Affine detector/descriptor, keypoints are detected and then described in parallel from the intensity components of the invariant surfaces, $\{\mathcal{I}_i^{\mathbb{H}}\}$ (Section 2.2.5). Finally, the depth components of the smooth surfaces, $\{\mathcal{D}_i^{\mathbb{RH}}\}$, are used to remap the coordinates back to the RGB-D input frame in parallel, by inverting the warp transforms on the extracted keypoints, $(\mathbf{p}, \varphi, r)_j$, and augmenting them with more information about the 3D space, $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(o)}$ (Section 2.2.6). For a summary of notations, see Table 2.1.

performed before warping, while description comes after it, thus might not address repeatability issues. Per contra, in this proposal, both detection and description are allowed from the warped surfaces, by applying smooth-surface labeling using a non-parametric method based on spherical k -means [79], as described in Section 2.2.2. Section 2.2.4 describes a similar approach to VIPs warping, which is further enhanced to generate flawless and accurate representations, by proposing a hybrid rigid-homography warping (Section 2.2.4). Sections 2.2.5 and 2.2.6 describe detecting, extracting, and correctly mapping the features to the local frame of the original input; the steps are similar to those of 2D-affine wrappers [64, 65], but the proposed is in three dimensions.

Table 2.1: Table of notations used throughout the chapter, where \mathbb{N} is the natural numbers set, \mathbb{R} is the real numbers set, \mathbb{SO} is the special orthogonal group, and \mathbb{SE} is the special Euclidean group.

Scalar	Definition
$k \in \mathbb{N}$	The k in k -means, # of smooth surfaces
$n \in \mathbb{N}$	Number of elements
$s \in \mathbb{R}$	A scale factor
$\alpha \in \mathbb{R}$	Azimuth rotation angle
$\theta \in \mathbb{R}$	Rotation angle in the axis-angle formalism
$\varphi \in \mathbb{R}$	Dominant-gradient orientation (2D)
$\psi \in \mathbb{R}$	Out-of-plane rotation angle

Continued on next page

Table 2.1 – continued from previous page

Vector	Definition
$\boldsymbol{\omega} \in \mathbb{R}^3$	Rotation axis in the axis-angle formalism
$\boldsymbol{c} \in \mathbb{R}^3$	A point
$\boldsymbol{n} \in \mathbb{R}^3$	A surface-tangent vector
$\boldsymbol{g} \in \mathbb{R}^3$	Dominant-gradient orientation
$\boldsymbol{t} \in \mathbb{R}^3$	A translation vector
$\boldsymbol{u} \in \mathbb{N}^2$	Camera basis in the horizontal direction
$\boldsymbol{v} \in \mathbb{N}^2$	Camera basis in the vertical direction
$\boldsymbol{p} \in \mathbb{N}^2$	A pixel in $(\boldsymbol{u}, \boldsymbol{v})$ coordinates
Operator	Definition
$ \cdot $	Absolute value
$\cdot \times \cdot$	Cross product
$\ \cdot\ _2$	Euclidean, L^2 , norm
$\exp(\cdot)$	Matrix exponential (Lie algebra)
\cdot^{-1}	Matrix inverse
\cdot^T	Matrix transpose
$\cdot(\boldsymbol{p})$	Map element at pixel \boldsymbol{p}
$[\cdot]_{\times}$	Skew-symmetric matrix
Matrix	Definition
\mathbf{I}	Identity matrix

Continued on next page

Table 2.1 – continued from previous page

$\mathbf{K} \in \mathbb{R}^{3 \times 3}$	Camera intrinsic parameters
$\mathbf{P} \in \mathbb{R}^{3 \times 4}$	Camera projection matrix
$\mathbf{R} \in \mathbb{SO}_3$	Rotation matrix
$\mathbf{T} \in \mathbb{SE}_3$	Rigid-body transform (6D pose)

Map	Definition
$\mathcal{I}(\mathbf{p}) \in \mathbb{N}^{1 \text{ or } 3}$	Gray or trichromatic intensity image
$\mathcal{D}(\mathbf{p}) \in \mathbb{R}$	Depth map
$\mathcal{C}(\mathbf{p}) \in \mathbb{R}^3$	3D-space point-cloud
$\mathcal{N}(\mathbf{p}) \in \mathbb{R}^3$	Surface normals map
$\mathcal{L}(\mathbf{p}) \in \mathbb{N}$	Smooth-surfaces annotation map

Script	Definition
$\cdot^{[\mathbf{T}]}$	Rigidly warped frame
$\cdot^{[\mathbf{H}]}$	Homographically morphed frame
$\cdot^{[\tilde{\mathbf{T}}]}$	Unisomorphic-rigidly warped frame
$\cdot^{[\mathbf{H}\tilde{\mathbf{T}}]}$	Equivalent to $\cdot^{[\tilde{\mathbf{T}}]}$ followed by $\cdot^{[\mathbf{H}]}$

$\cdot^{(\mathbf{T}_i)}$	A warped frame using $\mathbf{T}_i \in \mathbb{SE}_3$
$\cdot^{(o)}$	Original input local frame
$\cdot^{(\tilde{o})}$	$\cdot^{(o)}$ with coarse keypoint orientation

\cdot_*	Virtual image plane, located at $z = 1$
\cdot_i	The i th smooth surface

Continued on next page

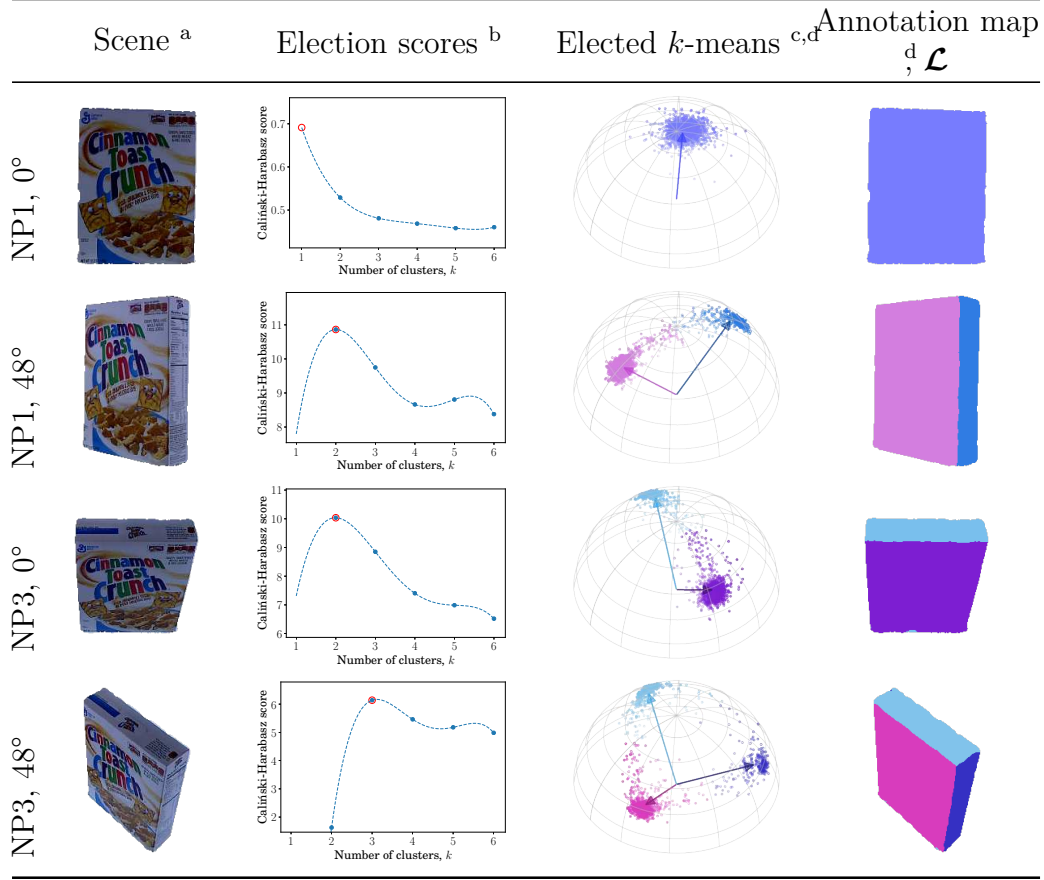
Table 2.1 – continued from previous page

\cdot_j	The j th keypoint
\cdot_s	Source frame
\cdot_d	Destination frame

2.2.1 Point-Cloud and Surface Normals

For a given intensity and depth RGB-D input pair, $(\mathcal{I}, \mathcal{D})$, which are registered in a 1:1 spatial relationship, and the noisy depth effects are initially minimized by applying a bilateral filter [80] (depth deviation of 5 mm within 3 pixel spatial Gaussian neighborhood). Then, the point-cloud and surface normals are computed, as both are frequently utilized throughout the remainder of the chapter.

The point-cloud is given by $\mathcal{C}(\mathbf{p}) = \mathcal{D}(\mathbf{p}) \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p}^\top & 1 \end{bmatrix}^\top$, where \mathbf{K} is the camera-intrinsic parameters matrix, and \mathbf{p} is a pixel. The unit surface normals, $\mathcal{N}(\mathbf{p}) = \frac{\nabla \mathcal{C}(\mathbf{p})}{\|\nabla \mathcal{C}(\mathbf{p})\|_2}$, are given by the normalized gradient of the point-cloud. Since the point-cloud is parameterized by the pixel curvilinear coordinates, $\mathbf{p} \stackrel{\text{def}}{=} (\mathbf{u}, \mathbf{v})$ denoting the horizontal and vertical directions respectively, its gradient is given by and further approximate with [81] $\nabla \mathcal{C}(\mathbf{p}) = \frac{\partial \mathcal{C}(\mathbf{p})}{\partial \mathbf{u}} \times \frac{\partial \mathcal{C}(\mathbf{p})}{\partial \mathbf{v}} \approx \begin{bmatrix} \frac{\partial \mathcal{D}(\mathbf{p})}{\partial \mathbf{u}} & \frac{\partial \mathcal{D}(\mathbf{p})}{\partial \mathbf{v}} & -1 \end{bmatrix}^\top$. It is worth noting that the normal direction is reversed to allow a more natural, out-of-surface orientation.



^a Surface-normals maps are utilized as input, while RGB images are shown for demonstration purposes. ^b The solid blue dots represent the measured scores. The dashed blue curves are polynomials fit over the measurements and extrapolated to the single-cluster case. The red circles denote the highest score. The non-integer k values of the extrapolation curves are for demonstration only, as they are not actually utilized nor have any meaningful interpretation. ^c The k -means model corresponding to the highest score. Cluster centroids, $\{\mathbf{n}\}$, are depicted by the arrows. Perspective is observed from a -45° elevation and 120° azimuth. ^d Up to a rotation, RGB colors in the elected k -means and the annotation map columns encode the orientations in the 3D space.

Figure 2-2: Clustering on the unit hemisphere. For each input, several spherical k -means models were constructed in parallel from the surface-normal map. Their Calinski-Harabasz scores were determined, where the score for the single-cluster was extrapolated. The highest-scoring k -means model was selected, and its clusters were back-projected to the pixel space to form the annotation map. NP1 and NP3 are two cameras fixed at 88.4° and 38.1° elevation angles, respectively.

2.2.2 Enumerating and Labeling Smooth Surfaces

This section describes the process for constructing an annotation map, \mathcal{L} , to uniquely identify k -smooth surfaces in the RGB-D input image. The approach involves clustering the scene surface normals using nonparametric spherical k -means and then identifying the smooth surfaces by projecting the clusters back to their corresponding pixel space. The latter is performed by swapping the *domain* and *range* of the surface normals map, thus constructs an inverse map to the image space pixels, $\mathbf{n} \rightarrow \{\mathbf{p}\}$.

Banerjee *et al.* proposed a spherical k -means [79] method for clustering on a unit hypersphere. The concept of spherical k -means is very similar to that of the standard k -means [82], except that the centroids are normalized after each expectation–maximization step. With the input being the scene surface tangent vectors (i.e., the surface normals computed in Section 2.2.1), clusters corresponding to the smooth surfaces of the scene are obtained. However, k -means is a parametric algorithm, which presents another distinct problem: determining the appropriate number of clusters, k .

To determine the number of smooth surfaces corresponding to k in the k -means approach, the Caliński–Harabasz score [83] is utilized, which measures the within-cluster to between-cluster dispersion ratio. Different k -means models are constructed in parallel over a range of k so that the best model can be selected according to the highest Caliński–Harabasz score. Since the Caliński–Harabasz score requires at least two clusters, an exact-fit polynomial is formed over the first few obtained scores and is extrapolated for the single-cluster case. Following this nonparametric approach, an appropriate spherical k -means clustering model is accordingly chosen. Back-

projection of the chosen model’s clusters forms the annotation map, \mathcal{L} , while its centroids, $\{\mathbf{n}_i\}$, are utilized in the warping transform estimation (Section 2.2.3). See Figure 2-2 for a visualized demonstration.

2.2.3 Per-Surface Warp Transforms

For the i th smooth-surface, \mathcal{L}_i , the aim is to transform the smooth surface to produce a viewpoint invariant representation, thus facilitating the computation of invariant features. In this section, the surface-warping 6D pose, \mathbf{T}_i , is computed as a preliminary step to warp the i th smooth surface into a viewpoint invariant representation (Section 2.2.4) by aligning a smooth surface parallel to and centered at a reference local plane (e.g., the virtual image plane). The exact same process is performed in parallel for all surfaces.

First, the smooth surface is represented by its mass and orientation centroids (Figure 2-3). Then, a standard plane–plane alignment method is applied. The mass centroid \mathbf{c}_i is expressed as the average of the surface cloud points, \mathcal{C}_i , whereas the orientation centroid \mathbf{n}_i is given by the i th centroid of the spherical k -means estimation (Section 2.2.2).

Since $\mathbf{T}_i \stackrel{\text{def}}{=} (\mathbf{R}_i \ \mathbf{t}_i)$ is a composition of both a rotation matrix and a translation vector, they are derived separately. The rotation required to align the smooth surface to the virtual image plane is expressed first in the axis–angle $(\boldsymbol{\omega}, \theta)_i$ representation, and it is then expanded into a rotation matrix. The unit rotation axis is given by $\boldsymbol{\omega}_i = \mathbf{n}_* \times \mathbf{n}_i$, while the rotation angle is $\theta_i = \arccos(\mathbf{n}_*^\top \mathbf{n}_i)$, where \mathbf{n}_* is the orientation of the virtual image plane. The rotation matrix is an exponential map of the rotation axis–angle formalism, $\mathbf{R}_i = \exp(\theta_i [\boldsymbol{\omega}_i]_\times)$, which is efficiently expanded

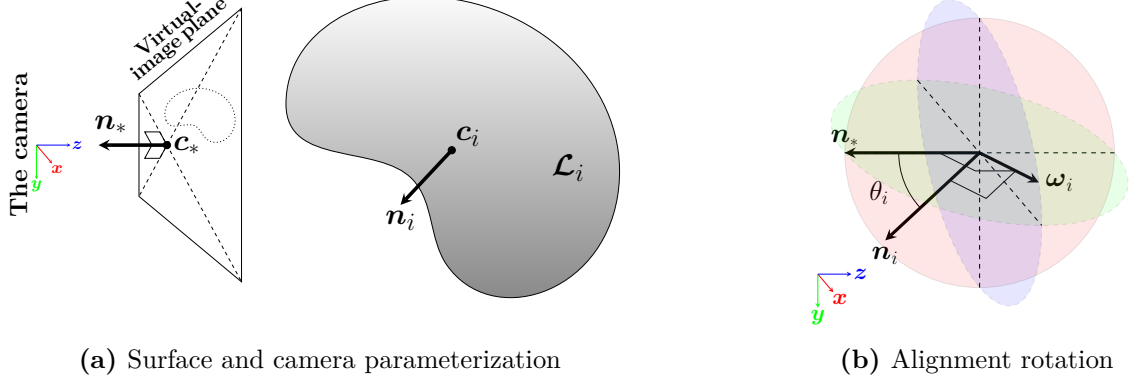


Figure 2-3: A smooth surface, \mathcal{L}_i , is parameterized by its mass and orientation centroids, \mathbf{c}_i and \mathbf{n}_i , respectively. The surface is aligned to and centered on a the virtual image plane (similarly, parameterized by \mathbf{c}_* and \mathbf{n}_*) by performing: (1) a rotation of $\theta_i = \arccos(\mathbf{n}_*^\top \mathbf{n}_i)$ degrees around the $\boldsymbol{\omega}_i = \mathbf{n}_* \times \mathbf{n}_i$ axis and (2) a translation of $\mathbf{t}_i = \mathbf{c}_* - \exp(\theta_i [\boldsymbol{\omega}_i]_\times) \mathbf{c}_i$.

using Rodrigues' rotation formula [84], $\mathbf{R}_i = \mathbf{I} + [\boldsymbol{\omega}_i]_\times \sin(\theta_i) + [\boldsymbol{\omega}_i]_\times^2 (1 - \cos(\theta_i))$. The translation vector, $\mathbf{t}_i = \mathbf{c}_* - \mathbf{R}_i \mathbf{c}_i$, is formulated to center the smooth surface at the center point of virtual image plane, \mathbf{c}_* . If a scale change is desired, the smooth surface can instead be centered at a non-unit \mathbf{z} -vector.

2.2.4 Smooth-Surface Warping

In the previous section, the warp transform, \mathbf{T}_i , needed to bring the smooth surface into a viewpoint invariant representation was computed. In order to apply that transform to the surface depth and intensity maps, this section investigates two major warping methods (Section 2.2.4). However, each of which has its own limitations, including outlier extrapolation, non-uniform pixel grids, and invalid depth mapping. Thus, a hybrid method (Section 2.2.4) between the two is proposed to overcome their

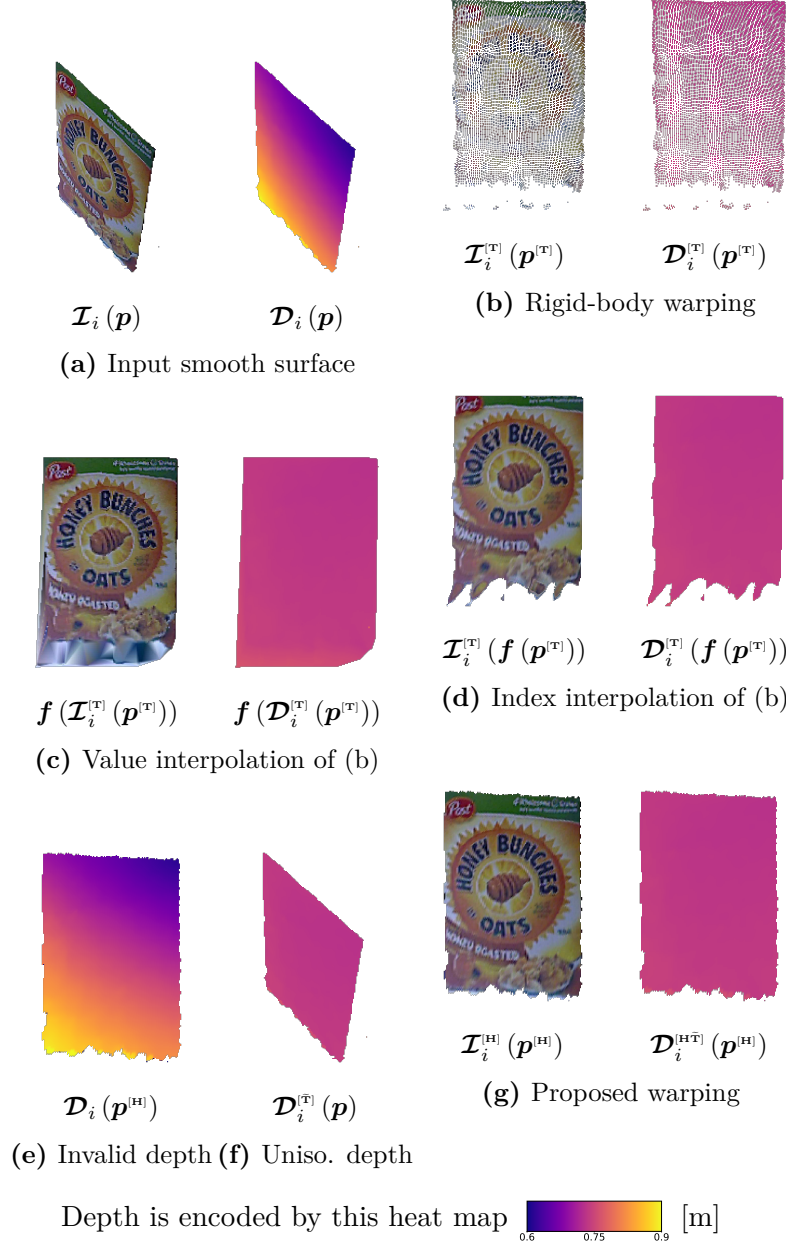


Figure 2-4: RGB-D warping techniques and the proposed improvement. (a) a smooth-surface tuple, $(\mathcal{I}, \mathcal{D})_i$, that is not parallel to the virtual image plane, which can be (b) rigidly warped parallel to the virtual image plane; however, this produces many missing-value pixels, as well as some outliers (Section 2.2.4). Common approaches to fill the missing pixels include (c) value interpolation or (d) index interpolation, where $f(\cdot)$ denotes the interpolation operator. Although interpolation can fill intra-point spaces, it also extrapolates spaces between the smooth surface and the boundary outliers, thereby creating artifacts. However, because the warped points represent a smooth surface, homography is investigated; (e) planar homography results in invalid depth warping. Therefore, a hybrid method is proposed, in which (f) an unisomorphic rigid warping to the depth map is computed, followed by a homography transformation; resulting in (g) the proposed hybrid rigid-homography warping (Section 2.2.4).

individual limitations. The proposed warping method is performed in parallel on all surfaces.

Rigid-Body Warping

The rigid-body transform can easily warp the depth and intensity maps to the desired orientation and position, as expressed in the following equations:

$$\begin{aligned} \mathcal{D}_i^{[\mathbf{T}]}(\mathbf{p}^{[\mathbf{T}]}) \begin{bmatrix} (\mathbf{p}^{[\mathbf{T}]})^\top & 1 \end{bmatrix}^\top &= \mathbf{P}\mathbf{T}_i \begin{bmatrix} \mathcal{C}_i(\mathbf{p})^\top & 1 \end{bmatrix}^\top, \\ \mathcal{I}_i^{[\mathbf{T}]}(\mathbf{p}^{[\mathbf{T}]}) &= \mathcal{I}_i(\mathbf{p}), \end{aligned} \quad (2.1)$$

where $\mathcal{D}_i^{[\mathbf{T}]}(\mathbf{p}^{[\mathbf{T}]})$, $\mathbf{p}^{[\mathbf{T}]}$ are the new depth and pixel position after applying the rigid-body transform, respectively, and $\mathbf{P} \stackrel{\text{def}}{=} \mathbf{K}(\mathbf{I}_3 \quad \mathbf{0}_3)$ is the camera projection matrix.

After rigid-body warping, the warped surface may have several defects which affect the quality of the resulting RGB-D image (Figure 2-4b). These defects include missing-value pixels due to surface out-of-plane rotation, non-uniform pixel grids from noisy depth measurements, and boundary outliers due to inaccurate surface labeling. Although linear-value interpolation or even cubic spline index interpolation can overcome missing-value issues to some extent, these methods also extrapolate to boundary outliers creating artifacts (Figures 2-4c and 2-4d). Fortunately, planar homography warping circumvents the non-uniform pixel grid and boundary outlier issues, as discussed next.

Planar Homography Warping

The planar homography transform, in terms of rotation and translation, is expressed as [11]:

$$\mathbf{H}_i = \mathbf{K} \left(\mathbf{R}_i - \frac{\mathbf{t}_i \mathbf{n}_i^\top}{|\mathbf{c}_i^\top \mathbf{n}_i|} \right) \mathbf{K}^{-1}, \quad (2.2)$$

where $(\mathbf{R}, \mathbf{t}, \mathbf{n}, \mathbf{c})_i$ are as in Section 2.2.3. After this transform, the warped pixel position $\mathbf{p}^{[\text{H}]}$ is obtained [11], and the RGB-D frame is morphed accordingly:

$$\begin{aligned} s \begin{bmatrix} (\mathbf{p}^{[\text{H}]})^\top & 1 \end{bmatrix}^\top &= \mathbf{H}_i \begin{bmatrix} \mathbf{p}^\top & 1 \end{bmatrix}^\top, \\ \mathcal{D}_i^{[\text{H}]}(\mathbf{p}^{[\text{H}]}) &= \mathcal{D}_i(\mathbf{p}), \\ \mathcal{I}_i^{[\text{H}]}(\mathbf{p}^{[\text{H}]}) &= \mathcal{I}_i(\mathbf{p}), \end{aligned} \quad (2.3)$$

where s is a scale factor introduced by unnormalized homography transforms, and $\mathbf{p}^{[\text{H}]}$ denotes the newly obtained pixel coordinates after the homography warping. Despite homography warping being more straightforward compared with rigid-body warping (Section 2.2.4), it fails to properly map the depth values $\mathcal{D}_i^{[\text{H}]}(\mathbf{p}^{[\text{H}]})$ because it operates in two dimensions, and therefore it is not directly feasible to obtain a warped depth map in the aligned plane (Figure 2-4e). Therefore, a rigid warping of the depth map is indispensable. On the other hand, boundary outliers, such as those at the bottom of Figure 2-4b, will be kept adjacent to the homographically morphed surface, with no intra-points between them. Thus, unlike in rigid warping extrapolation (Figures 2-4c and 2-4d), interpolation of planar homography warping will not locate any outlier intra-points to extrapolate. Similarly, homography implies a plate constraint on the points; thus, a uniform pixel grid is achieved despite depth noise. In light of these

strengths and weaknesses of the individual approaches, a hybrid approach that takes advantage of both methods is proposed in Section 2.2.4.

Hybrid Rigid–Homography Warping

Knowing that rigid warping does not affect the distances or angles between points, the surface can be approximated using homography after a rigid transform. Accordingly, a hybrid rigid–homography method is proposed to achieve accurate and high-quality surface warping, which is summarized by the following:

$$\mathcal{D}_i^{[\text{H}\tilde{\text{T}}]}(\mathbf{p}^{[\text{H}]}) = \mathcal{D}_i^{[\tilde{\text{T}}]}(\mathbf{p}) = \mathcal{D}_i^{[\text{T}]}(\mathbf{p}^{[\text{T}]}) , \quad (2.4)$$

where $\mathcal{D}_i^{[\tilde{\text{T}}]}(\mathbf{p})$ is an intermediate depth map with rigidly and unisomorphically transformed depth values. Its pixel positions are unchanged (Figure 2-4f), so it does not suffer from missing-value pixels as occurs with rigid-body warping (Section 2.2.4). Then, the homography transform (Section 2.2.4) is applied to the unisomorphic map, thereby properly warping the RGB-D frame (Figure 2-4g), with a uniform pixel grid even in presence of some noise, since homography imposes a flat-surface constraint.

Subsequently, robust viewpoint invariant smooth surfaces are obtained for different viewpoints of the same surface (Figure 2-5). Although only $\mathcal{I}_i^{[\text{H}]}$ is required for feature detection (Section 2.2.5), $\mathcal{D}_i^{[\text{H}\tilde{\text{T}}]}$ will be utilized later for keypoint remapping (Section 2.2.6). Finally, it is worth noting that surfaces with large out-of-plane rotations may not have ideal depth maps (e.g., the camera denoted by NP3 with $\alpha = 63^\circ$ shown in Figure 2-5 has a large missing-depth region). In that case, the missing depth is interpolated in small regions and the computation of keypoints in largely

missing-depth regions is forwent.

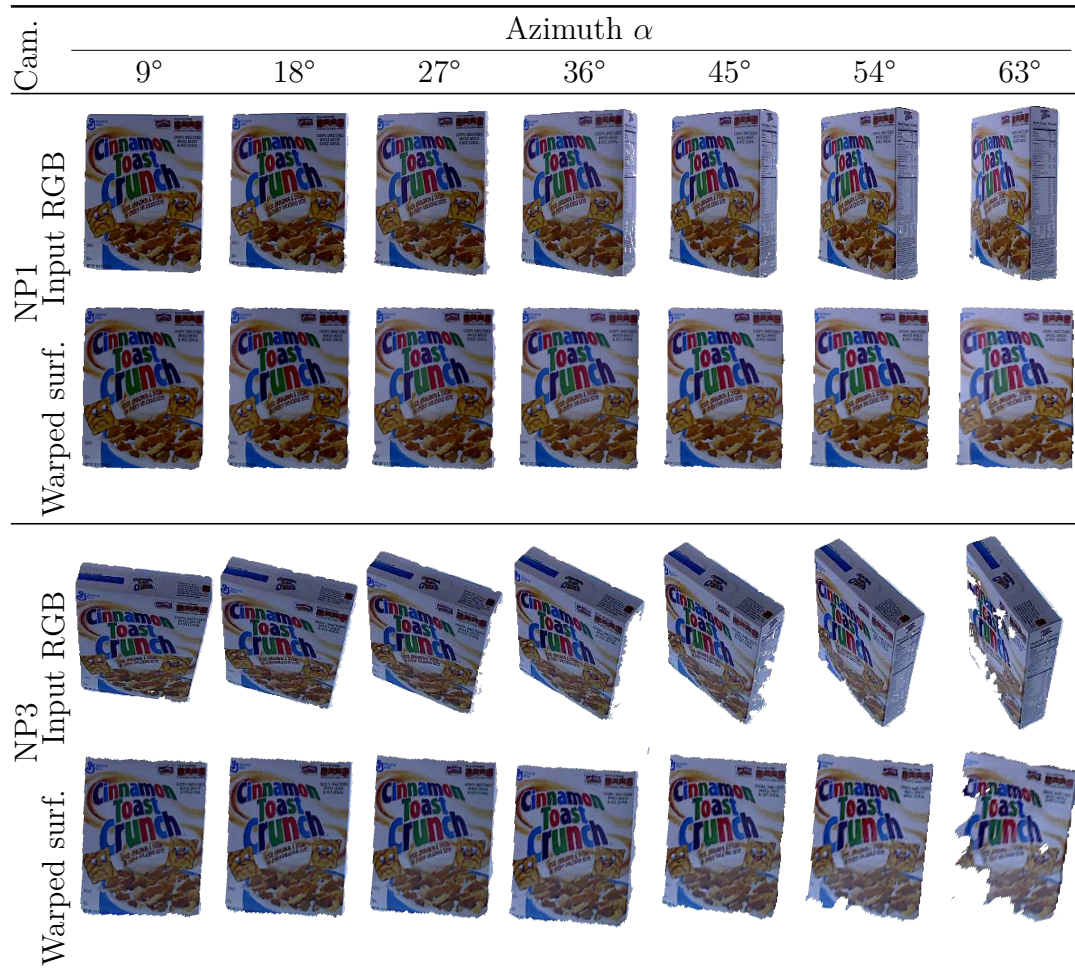


Figure 2-5: Warped smooth surfaces representing the frontal surface of a given sample. Despite various elevation and azimuth angles of the RGB-D input, smooth surfaces remain aligned to the virtual image plane. NP1 and NP3 are two cameras fixed at 88.4° and 38.1° elevation angles, respectively. Although depth maps are also warped, only RGB images of the frontal surface are shown here.

2.2.5 Feature Detection and Description

At this stage, the smooth surfaces are in a viewpoint invariant representation; they are independent from the viewpoint. Therefore, fully affine keypoints and feature description vectors can, in parallel, be extracted and computed from the smooth surfaces by applying standard detection and description processes for each smooth surface, \mathcal{I}_i^{H} . Accordingly, virtually any detector and descriptor pair of choice can be utilized without regard to its invariance formula(s) or implementation.

Detectors of local image features express keypoints in 2D image space parameters, which are passed to the feature descriptors to compute their corresponding description vectors. The feature description of each detected keypoint, despite being in arbitrary dimensions (implementation dependent), is directly utilized in correspondence estimation (Section 2.3.1) without requiring any adaptations. However, keypoints from different surfaces have different transforms despite representing the same scene, so combining them remains an issue, which is tackled in the next section.

2.2.6 Mapping Keypoints to Their Original Local Frame

This step is intended to simplify computations by eliminating the dependency of each keypoint on its per-frame transform, \mathbf{T}_i (Section 2.2.3), based on the fact they all belonged to the same scene before mapping (Figure 2-6a). Putting together all keypoints in the local frame of the RGB-D input prepares them for correspondence estimation with other RGB-D images and also results in elliptical affine region neighborhoods that are compatible with affine region detectors [85].

Each j th detected keypoint $(\mathbf{p}, \varphi, r)_j$ is initially expressed using image space pa-

rameters $(\cdot)_j$, where $\mathbf{p}_j \in \mathbb{R}^2$ is the keypoint center, $\varphi_j \in \mathbb{R}$ is the dominant gradient orientation [37], and $r_j \in \mathbb{N}$ is the neighborhood radius in pixels. To map the keypoints from the image frame $(\cdot)_j$, to the local frame $(\cdot)_j^{(o)}$, two intermediate parameterizations are utilized: the warped-frame $(\cdot)_j^{(T_i)}$, and the coarse orientation of keypoints in the local frame $(\cdot)_j^{(o)}$ (see Figure 2-6b). these three steps are described in the following sections. The exact same process is performed in parallel for all keypoints.

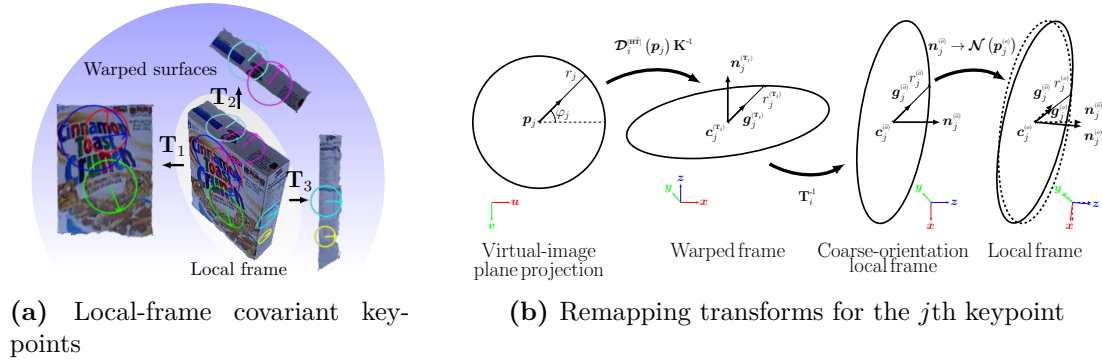


Figure 2-6: (a) keypoints detected on the warped surfaces are remapped back to the original input local frame to eliminate their per-surface parameterization. The resulting elliptical keypoints not only resemble those of affine region detectors but also vary covariantly with the object geometry and are invariant to viewpoint difference; (b) each keypoint is remapped from the virtual image plane of the smooth surface viewpoint invariant representation to the local frame of the original input by applying several transforms. The keypoint is first unprojected to the 3D space (Section 2.2.6), then its basis is changed to the local frame (Section 2.2.6). Finally, its orientation is corrected (Section 2.2.6). Refer to Section 2.2.6 for annotation and technical details.

Image to Warped Frame Transformation

This section re-parameterizes each detected 2D keypoint $(\mathbf{p}, \varphi, r)_j$ on the warped surfaces into 3D-space parameters $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(T_i)}$ of the corresponding i th warped frame,

where \mathbf{T}_i is the per-surface warp transform (Section 2.2.3), $\mathbf{c}_j^{(\mathbf{T}_i)} \in \mathbb{R}^3$ is the keypoint center, $\mathbf{n}_j^{(\mathbf{T}_i)}$ is the plane tangent orientation, $\mathbf{g}_j^{(\mathbf{T}_i)} \in \mathbb{R}^3$ is the unit gradient orientation, and $r_j^{(\mathbf{T}_i)} \in \mathbb{R}$ is the radius. Without loss of generality, at this stage, the keypoints are assumed parallel to the virtual image plane, $\mathbf{n}_j^{(\mathbf{T}_i)} = \mathbf{n}_*$ (Section 2.2.3), to ensure their orthogonality with the gradient orientation, $\mathbf{g}_j^{(\mathbf{T}_i)}$. This orientation assumption is compensated for by Equation (2.7) in a subsequent section. The rest of the parameters, $(\mathbf{c}, \mathbf{g}, r)_j^{(\mathbf{T}_i)}$, are given by:

$$\begin{aligned} \mathbf{c}_j^{(\mathbf{T}_i)} &= \mathcal{D}_i^{[\mathbf{H}\tilde{\mathbf{T}}]}(\mathbf{p}_j) \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p}_j^\top & 1 \end{bmatrix}^\top, \\ r_j^{(\mathbf{T}_i)} \mathbf{g}_j^{(\mathbf{T}_i)} &= \mathcal{D}_i^{[\mathbf{H}\tilde{\mathbf{T}}]}(\mathbf{p}_j) \mathbf{K}^{-1} \mathbf{T}_j \begin{bmatrix} \mathbf{p}_j^\top & 1 \end{bmatrix}^\top - \mathbf{c}_j^{(\mathbf{T}_i)}, \end{aligned} \quad (2.5)$$

where $r_j^{(\mathbf{T}_i)}$ and $\mathbf{g}_j^{(\mathbf{T}_i)}$ are separable due to $\|\mathbf{g}_j^{(\mathbf{T}_i)}\|_2 = 1$, and

$\mathbf{T}_j = \left(\mathbf{I}_{3 \times 2} \begin{bmatrix} r_j \cos \varphi_j & r_j \sin \varphi_j & 1 \end{bmatrix}^\top \right) \in \mathbb{SE}_2$ is an image transform that shifts \mathbf{p}_j by r_j in the φ_j direction. Note that \mathbf{p}_j is a tuple of real numbers, in which the depth value $\mathcal{D}_i^{[\mathbf{H}\tilde{\mathbf{T}}]}(\mathbf{p}_j)$ is interpolated over the neighborhood of \mathbf{p}_j .

Warped to Coarse-Orientation Local Frame Transformation

By applying the inverse of the smooth-surface transform, $\mathbf{T}_i \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{R}_i & \mathbf{t}_i \end{pmatrix}$, to the warped frame parameters $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(\mathbf{T}_i)}$, the coarse-orientation local parameters $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(\odot)}$ are obtained, where radius, $r_j^{(\odot)} = r_j^{(\mathbf{T}_i)}$, remains unchanged, and $(\mathbf{c}, \mathbf{n}, \mathbf{g})_j^{(\odot)}$ are given by:

$$\begin{aligned} \begin{bmatrix} (\mathbf{c}_j^{(\odot)})^\top & 1 \end{bmatrix}^\top &= \mathbf{T}_i^{-1} \begin{bmatrix} (\mathbf{c}_j^{(\mathbf{T}_i)})^\top & 1 \end{bmatrix}^\top, \\ \begin{bmatrix} \mathbf{n}_j^{(\odot)} & \mathbf{g}_j^{(\odot)} \end{bmatrix} &= \mathbf{R}_i^\top \begin{bmatrix} \mathbf{n}_j^{(\mathbf{T}_i)} & \mathbf{g}_j^{(\mathbf{T}_i)} \end{bmatrix}. \end{aligned} \quad (2.6)$$

Coarse-to-Fine Local Frame Transformation

Among the local frame parameters $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(\circ)}$, center and radius, $(\mathbf{c}, r)_j^{(\circ)} = (\mathbf{c}, r)_j^{(\odot)}$ are exactly the same as the corresponding coarse-orientation local parameters, and only tangent and gradient orientations $(\mathbf{n}, \mathbf{g})_j^{(\circ)}$ needs to be corrected due to the orientation assumption made earlier. It is worth noting that (1) any rigid transform preserves orthogonality, (2) by the construction of Section 2.2.6, $(\mathbf{n}_j^{(\mathbf{T}_i)})^\top \mathbf{g}_j^{(\mathbf{T}_i)} = 0$; therefore, the unit vectors $(\mathbf{n}_j^{(\mathbf{T}_i)}, \mathbf{g}_j^{(\mathbf{T}_i)}, \mathbf{n}_j^{(\mathbf{T}_i)} \times \mathbf{g}_j^{(\mathbf{T}_i)})$ form an orthogonal basis. Accordingly, the flat-plane assumption is compensated for by replacing $\mathbf{n}_j^{(\circ)}$ with the correct surface normal (as computed in Section 2.2.1) and thereupon correcting $\mathbf{g}_j^{(\circ)}$. Consequently:

$$\begin{aligned} \mathbf{n}_j^{(\circ)} &= \mathcal{N}(\mathbf{p}_j^{(\circ)}), \\ \mathbf{g}_j^{(\circ)} &= \mathbf{n}_j^{(\circ)} \times (\mathbf{g}_j^{(\circ)} \times \mathbf{n}_j^{(\circ)}), \end{aligned} \tag{2.7}$$

where $\mathbf{p}_j^{(\circ)} \in \mathbb{R}^2$ is the projection of $\mathbf{c}_j^{(\circ)}$ on the virtual image plane, i.e., $\mathcal{D}(\mathbf{p}_j^{(\circ)})[(\mathbf{p}_j^{(\circ)})^\top \quad 1]^\top = \mathbf{K}\mathbf{c}_j^{(\circ)}$, by which the surface-normal map is interpolated.

Thus far, invariant detection and description is achieved, in which the wrapper returns the feature description vector (depending on the descriptor’s own implementation). Furthermore, it returns a 3D keypoint tuple $(\mathbf{c}, \mathbf{n}, \mathbf{g}, r)_j^{(\circ)}$ for each detected and described 2D keypoint $(\mathbf{p}, \varphi, r)_j$, given the smooth-surface transform \mathbf{T}_i (Section 2.2.3) and its depth map $\mathcal{D}_i^{(\text{HT})}$ (Section 2.2.4). The proposed keypoint parametric representation is useful for enriching additional tasks, such as 3D matching (Section 2.3) and elliptical neighborhood projection (Figure 2-6a), commonly used by affine region detectors [85].

2.3 An Application: 6D Pose Estimation

The challenging 6D pose estimation problem is chosen as a high-level algorithm in which the proposed wrapper acts as a building block. See Figure 2-7 for an abstraction. Given two tuples of source and destination RGB-D images, $(\mathcal{I}, \mathcal{D})_s$ and $(\mathcal{I}, \mathcal{D})_d$, the aim is to find the relative position and orientation, i.e., the 6D pose $\hat{\mathbf{T}}_s^d$ [84], that properly registers these partial, so-called 2.5D, scenes together. The 6D pose correspondences, $\hat{\mathbf{T}}_s^d$, between the source and destination feature vectors are established using the standard k -NN (k -nearest neighbor) algorithm [86]. A 3D space geometric verification method [87] is employed with a RANSAC (random sample consensus) [88] variant called Optimal RANSAC [89] in order to achieve robustness to correspondence outliers between the keypoint centers. Further details are provided in the following subsections.

2.3.1 Correspondence Estimation

Given two sets of description vectors—namely, the source and destination—and using the k -NN algorithm [86], a tree-based index is built from one set and then all of the members in the other set are queried for the k -NN in the index. The k -NN is computed, where $k = 1$, from one set to another and vice versa, and then limited correspondence to mutual k -NNs. The correspondence can be filtered further by thresholding the ratio of first-to-second nearest-neighbor distances [37]. After obtaining the correspondence map between the two sets, the keypoints from both sets are grouped into corresponding pairs in order to estimate the 6D pose (Section 2.3.2).

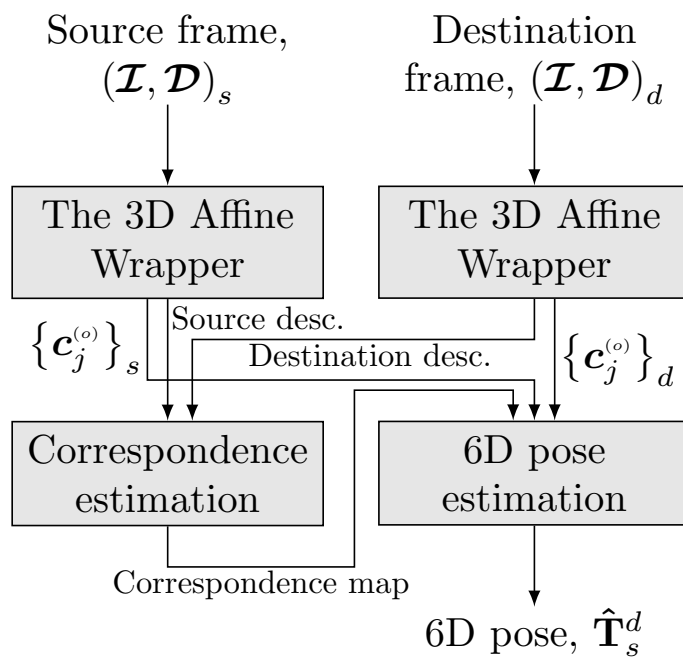


Figure 2-7: 6D pose estimation is one high-level application, among many, that can utilize the proposed wrapper as a building block. First, feature description vectors are utilized to compute the correspondence (Section 2.3.1). The 6D pose is then computed from the correspondence map (corr. map) and keypoint centers via geometric verification (Section 2.3.2).

2.3.2 Geometric Verification and 6D Pose Estimation

Utilizing the center point, and the normal and gradient orientations of the corresponding keypoints, $\{(\mathbf{c}, \mathbf{n}, \mathbf{g})_j^{(o)}\}$ (Section 2.2.6), the 6D pose can be estimated from only a single correspondence [40]. However, not all detectors produce sufficiently large keypoints; the transformation robustness is highly dependent upon the keypoint gradient, and a large enough neighborhood is essential for stable estimation. Furthermore, the 2D local image features, the points of comparison, do not provide normal and gradient orientations that are viewpoint invariant. Therefore, the utilization of normal and gradient orientations is sacrificed and only the keypoint centers are utilized, $\{\mathbf{c}_j^{(o)}\}$, in order to enable a fair comparison. For the aforementioned reason, the point-cloud (Section 2.2.1) is computed for 2D cases to interpolate their 3D keypoint centers.

Verification was performed by forming a rigidity constraint out of two corresponding sets of 3D keypoint centers, each containing three points at minimum. A SVD (singular value decomposition)-based method [87] is employed to compute the 6D pose $\hat{\mathbf{T}}_s^d$, within the Optimal RANSAC iterations, as it acts as the fitting model for the hypothetical inliers.

2.4 Experimental Setup

In Section 2.2, a wrapper is proposed to improve the viewpoint invariance of virtually any local image detector/descriptor without requiring any adaptations to their interfaces or internal invariance formulas. This section is dedicated to the experimental setup to demonstrate the proposed method’s effectiveness, generality, and

robustness by performing two types of experiments: performance comparison and sensitivity analysis.

The performance in all experiments was assessed using viewpoint invariance scores (Section 2.4.3). Viewpoint invariance measures the stability against viewpoint difference by counting the number of out-of-plane samples for which a correct estimation of the relative pose is achieved. The samples consists of several pairs of RGB-D images, a source and a destination at each iteration, taken from a wide interval of different relative viewpoints.

Additionally, because there are randomized factors that stem from k -means initialization (proposed wrapper only) and Optimal RANSAC [89] (2D and 3D Affine), each experiment, for both 2D and 3D Affine approaches, was performed in triplicate, and the mean and standard deviation of the performance metric are reported.

The proposed wrapper was implemented in PythonTM [90]. The implementation widely depended on the NumPy [91], OpenCV [92], Scikit-learn [93], and SciPy [94] packages for linear algebra, image processing, modeling, and signal processing, respectively. Furthermore, the Matplotlib [95] and MayaVi [96] packages were used for 2D and 3D graphics, respectively.

2.4.1 Performance Comparison Experiments

A brief comparison is first performed in an ideal synthetic setup in order to show the extended level, in terms of the viewpoint invariance range, of the proposed wrapper compared with the well-known SURF detector and SIFT descriptor. The synthetic dataset consisted of a cuboid object representing a cereal box, where its rendered RGB-D frames resembles both texture and viewpoints of the real-world ‘C.

DS.		Relative viewpoint angle ψ												
Cam.	α_s	ψ_{\min}	ψ_{\max}	$\psi_{\delta, \max}$	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°	
C. T. Crunch	NP3	48°	-75°	75°	75°									
	NP1 ^a	48°	-138°	132°	135°									
	NP1	0°	-90°	90°	90°									
H. B. Oats	NP3	0°	-75°	75°	75°									
	NP1	180°	-90°	90°	90°									
	NP1	180°	-90°	90°	90°									
Cheez It	NP3	90°	-75°	75°	75°									
	NP1	180°	-90°	90°	90°									
	NP1	180°	-90°	90°	90°									
Pringles	NP3	192°	-75°	75°	75°									
	NP1	192°	-81°	81°	81°									
	NP1	192°	-81°	81°	81°									

^a Labeled as ‘NP1 - v2’ in Section 2.5 to avoid ambiguity.

Figure 2-8: RGB images of the evaluation datasets (DS.) from different cameras (Cam.) sampled along different relative viewpoint angles, ψ . The source RGB-D image is fixed to α_s (equivalent to $\psi = 0^\circ$), while the destination RGB-D image is taken from the interval $[\psi_{\min}, \psi_{\max}]$ at every 3° step. Based on the source frame and camera elevation, each dataset limits the viewpoint invariance score, ψ_δ , to a specific maximum, $\psi_{\delta, \max}$. Only sample images are shown.

T. Crunch’ object observed from the NP1 camera (Figure 2-8). After the synthetic experiment, several state-of-the-art feature detectors and descriptors were embedded within the proposed method to establish its generality. At the same time, to demonstrate the effectiveness of the proposed wrapper, each proposed 3D Affine detector/descriptor was compared with its 2D version. A list of the detectors and descriptors studied, at the 2D and 3D Affine levels, is provided in Table 2.2. Because it is impractical to evaluate all intermixed combinations, the performance of all detectors are evaluated with SIFT as the representative descriptor and all descriptors are evaluated with SURF as the representative detector. These are chosen since both SURF and SIFT are common feature detectors/descriptors and because SURF has reasonably large keypoints.

To ensure realistic results with different settings, this comparison was performed using real-world objects with planar and curved surfaces, various texture patterns, and different illumination conditions. Four objects were chosen, each of which has a different texture, color intensities, and contrast. The first three have box-like polygonal shapes to represent planar surfaces, and the fourth is cylindrical to represent curved surfaces. All objects were rotated in wide-azimuth rotations and were observed by cameras at two different elevations, from which viewpoint invariance was evaluated. At the same time, illumination conditions change on the objects’ sides while they rotate. The effects of the source frame selection is also investigated by repeating an experiment with a different source frame. The 960 RGB-D frames representing the four objects (Figure 2-8) are instances of the BigBIRD datasets [5] and were captured using intrinsically and extrinsically calibrated RGB-D cameras fixed at different elevation angles. Over the entire 360° rotation window for the motorized

turntable below the target object, RGB-D images were captured for each 3° azimuth step by all cameras. Furthermore, the datasets provide ground-truth data which define the object region in the image, i.e., binary annotation maps, $\mathcal{L}_k(\mathbf{p}) \in \{0, 1\}$. Refer to Reference [5] for further details.

Table 2.2: List of 10 detectors and 11 descriptors against which the proposed wrapper was evaluated. For practicality, all detectors were intermixed with a SIFT [37] descriptor, and all descriptors were intermixed with a SURF [58] detector.

Local Image Feature	Detector Descriptor	
AGAST (adaptive & generic detection based on accelerated segment test) [71]	✓	—
AKAZE (accelerated KAZE) [76]	✓	—
BOOST [45]	—	✓
BRIEF (binary robust independent elementary features) [70]	—	✓
BRISK (binary robust invariant scalable keypoints) [73]	✓	✓
CenSurE (center surround extremas detector) [66]	✓	—
DAISY [69]	—	✓
DLCO (descriptor learning using convex optimization) [44]	—	✓
FAST (features from accelerated segment test) [68]	✓	—
FREAK (fast retina keypoint) [75]	—	✓
GFTT (good features to track) [57]	✓	—
LATCH (learned arrangements of three patch codes) [77]	—	✓
MSER (maximally stable extremal regions) [67]	✓	—
ORB (oriented FAST and rotated BRIEF) [72]	✓	✓
RootSIFT (root-normalized SIFT descriptor) [74]	—	✓
SIFT (scale-invariant feature transform) [37]	✓	✓
SURF (speeded up robust features) [58]	✓	✓

The datasets utilized in the evaluation are ‘C. T. Crunch’, ‘H. B. Oats’, ‘Cheez It’,

and ‘Pringles’, which represent some polygonal and cylindrical objects. The cameras used for evaluation are the NP1 and NP3 cameras, which were fixed at 88.4° and 38.1° elevation angles, respectively. Dataset intrinsic and extrinsic parameters were utilized for 1:1 depth-to-color registration. These parameters were also used with the turntable azimuth angle to compute the ground-truth of relative pose transform $\mathbf{T}_s^d \stackrel{\text{def}}{=} (\mathbf{R}_s^d \ \mathbf{t}_s^d)$ between any two RGB-D pairs, $(\mathcal{I}, \mathcal{D})_s$ and $(\mathcal{I}, \mathcal{D})_d$. The background was masked using the dataset annotation maps, \mathcal{L}_k .

Sample dataset frames, along with their relative viewpoint angles, are shown in Figure 2-8. Note that the relative viewpoint angles are well beyond $\pm 60^\circ$, as denoted in Figure 2-8 by the $[\psi_{\min}, \psi_{\max}]$ interval. It is also worth noting that the second and third rows in Figure 2-8, which differ in source frame, are intended to investigate the effects of the source frame selection. Furthermore, note the realistic setup, in which the missing-depth instances and different illumination effects are quite apparent.

2.4.2 Sensitivity Analysis Experiments

A depth-noise sensitivity analysis was performed during viewpoint difference in the following manner. A synthetically ideal RGB-D frame was matched against several other frames, each of which has a noisy depth and a different viewpoint angle. In this experiment, the response to numerous SNRs (signal-to-noise ratios) was studied, with SURF and SIFT as the feature detector and descriptor, respectively. To obtain a noisy depth map with a specific SNR, the standard definition was utilized, $SNR = 20 \log_{10} \left(\frac{\mu}{\sigma} \right)$, where μ is the signal mean, which ranges in $0.70 \text{ m} \pm 0.07 \text{ m}$, and σ is the noise deviation. More specifically, a synthetic ideal depth map of the scene geometry was elementwise multiplied by samples drawn at random from a normal

distribution, $\mathcal{N}(\mu, \sigma^2)$, with mean $\mu = 1$ and variance $\sigma^2 = 10^{(-SNR/10)}$. A sufficiently large range of SNRs was considered, namely, from the range 20 dB–65 dB in steps of 5 dB.

2.4.3 Performance Metric

Although local image features are usually evaluated in terms of detection repeatability [85] and description distinctiveness [97], these indices do not seem to be very accurate indicators of the performance of a local image feature when it is utilized in other high-level algorithms, which is often the case. That is, in high-level problems, there is no guarantee that a pair comprising a highly repeatable detector and a highly distinct descriptor can perform better than other combinations of detectors and descriptors. Admittedly, repeatability is measured as the sampling window intersection-to-union ratio of corresponding keypoints, which is misleading for applications that depend only on the center point, such as correspondence estimation. For instance, two concentric keypoints with different radii are deemed to have a low repeatability despite being a perfect correspondence pair. Similarly, two elliptical keypoints with high aspect ratios and with their centers located along each other’s major axis would be designated with moderately high repeatability, even if their centers are far apart. Therefore, the performance of each detector/descriptor pair was evaluated at the application level using a high-level performance indicator.

Accordingly, the stability against different viewpoint difference is utilized as a performance metric. Briefly, it measures the accumulative-interval length along the axis of a relative viewpoint angle, for which the alignment error is tolerated. In this section, as a preliminary step to introduce the performance metric of viewpoint

invariance, there is a need to first define the relative viewpoint angle, the transform estimation error, and the alignment error.

The relative viewpoint angle, ψ , is defined as the source–destination out-of-plane rotation angle:

$$\psi = \arccos \left(\mathbf{n}_*^\top \mathbf{R}_s^d \mathbf{n}_* \right), \quad (2.8)$$

where \mathbf{n}_* is the virtual image plane orientation (Section 2.2.3). Note that ψ is also assigned the azimuth sign to indicate the rotation direction.

The transform estimation error, $\mathbf{T}_e \in \mathbb{SE}_3$, measures the difference between the estimated 6D pose, $\hat{\mathbf{T}}_s^d$, and that of the ground truth, \mathbf{T}_s^d , while the alignment error, $\ell(\psi) \in \mathbb{R}$, is the RMS (root-mean-square) of the point-to-point misalignment distance resulting from \mathbf{T}_e . These quantities are given by:

$$\begin{aligned} \mathbf{T}_e(\psi) &= \mathbf{T}_s^d(\psi)^{-1} \hat{\mathbf{T}}_s^d(\psi), \\ \ell(\psi) &= \left(\frac{1}{n} \sum_{\mathbf{p}} \left\| \mathbf{T}_e(\psi) \begin{bmatrix} \mathbf{c}_s(\mathbf{p}) \\ 1 \end{bmatrix} - \begin{bmatrix} \mathbf{c}_s(\mathbf{p}) \\ 1 \end{bmatrix} \right\|_2^2 \right)^{\frac{1}{2}}, \end{aligned} \quad (2.9)$$

where \mathbf{C}_s and n are the point-cloud of the source frame (Section 2.2.1) and the number of points in it, respectively.

The performance metric of the viewpoint invariance, ψ_δ , can be defined as the ψ -axis length where the alignment error, $\ell(\psi)$, is less than or equal to a maximally tolerated alignment error, ℓ_ϵ , up to a normalizing constant. Mathematically, let $\{\psi_0, \dots, \psi_{n-1}\}$ be an n -element set in ascending order, containing both the dataset bounds $\{\psi_{\min}, \psi_{\max}\}$

and the roots of the polynomial $\ell(\psi) = \ell_\epsilon$, then ψ_δ is given by:

$$\psi_\delta = \frac{1}{s} \sum_{m=0}^{n-2} \begin{cases} \psi_{m+1} - \psi_m, & \text{if } \ell\left(\frac{\psi_{m+1} + \psi_m}{2}\right) \leq \ell_\epsilon, \\ 0, & \text{otherwise,} \end{cases} \quad (2.10)$$

where s is a normalizing constant set to $s = 2$ (assuming symmetric alignment error in both relative viewpoint directions). A detector/descriptor pair is considered viewpoint invariant if the length of its viewpoint invariance score, ψ_δ , is greater than or equal to a given threshold, ψ_Δ .

The maximally tolerated alignment error is set to $\ell_\epsilon = \sqrt{2}$ cm and the desired degree of viewpoint invariance to $\psi_\Delta = 60^\circ$. The maximally tolerated alignment error is constrained by factors that originate from the motivation; to perform 3D object reconstruction and grasp, the said object using a robotic manipulator, in which ℓ_ϵ corresponds to the manipulator tolerance. Furthermore, based on images corresponding to equal cones of vision on a systematically sampled sphere, only 13 cones, thus images, are sufficient to cover the view sphere with $\psi_\Delta = 60^\circ$, as compared with 53 cones in the case of the 2D features viewpoint limit, 25° – 30° .

The performance metric, ψ_δ , measures the viewpoint invariance by considering a wide range of viewpoint difference and quantifying their corresponding errors. It is utilized in the next section to compare the viewpoint invariance of both 2D and 3D Affine local image features in order to show the proposed method's effectiveness.

2.5 Results and Discussion

2.5.1 Performance Comparison Experiments

This section aims to provide an in-depth understanding of the viewpoint limitations in previous developments and the extended viewpoint range of the proposed method. First, a synthetic dataset is used. Then, the results are presented from numerous experiments that were performed using various local image features, both with and without the proposed wrapper and on several real-world objects. The results are quantitatively compared, and the proposed wrapper demonstrates viewpoint invariance gains for almost any local image detector/descriptor without requiring any interfacing or internal adaptations. On the basis of further qualitative results for some challenging viewpoints, the proposed wrapper shows great stability in correspondence and pose estimation against viewpoint difference.

Ideal Synthetic Data

After a brief comparison of both proposed and existing approaches, this section answers the question of why the proposed method behaves differently and favorably despite its dependence on the same detector and descriptor used by the 2D approach. A synthetically ideal dataset resembling the real-world ‘C. T. Crunch’ dataset (third row in Figure 2-8) was constructed, in which one RGB-D frame is fixed and matched against the rest of the frames using the SURF detector and SIFT descriptor for both the 2D and proposed methods. The 3D Affine variant of the SURF/SIFT approach (*3D affine* for short) outperformed the 2D version as shown in Figure 2-9.

In Figure 2-9, the horizontal axis, ψ , represents a total of 90 matching exper-

iments between a fixed source frame and various destination frames with different out-of-plane rotations, sampled at every 3° . The vertical axis represents the alignment error, $\ell(\psi)$, Equation (2.9), which is desirably kept lower than the maximally tolerated alignment error, ℓ_ϵ (the black dotted horizontal line). The viewpoint invariance score, ψ_δ , Equation (2.10), represents the length of the relative viewpoint angle range for which the alignment error curve of each approach remains below the maximally tolerated alignment error, ℓ_ϵ . Since the curve is generally symmetric around $\psi = 0^\circ$ (i.e., the source RGB-D image corresponding to the α_s azimuth value indicated in Figure 2-8), a normalizing constant, $s = 2$, is utilized to consider only half of the accumulative interval as the actual viewpoint invariance score. When using 2D features, a viewpoint invariance range of $\psi_\delta = 75.89^\circ$ is observed, while the proposed method achieved a higher viewpoint invariance range, $\psi_\delta = 113.53^\circ$. It is worth noting that both the 2D and 3D Affine features exhibit similar U-shaped curves, in which the minimum value is observed around $\psi = 0^\circ$, and they approach saturation values on ψ -axis edges related to the object dimensions. Furthermore, the maximally tolerated alignment error, ℓ_ϵ , is observed to reside mostly near the elbow of the $\ell(\psi)$ curves, thereby enabling an effective and low deviation assessment of viewpoint invariance. Overall, the 2D approach was able to more-or-less keep up with the proposed method, except in the ranges of -126° to -108° and 45° to 102° . Thus, a sample matching experiment from these ranges is inspected.

Next, to investigate the reasons behind the weakness of the 2D approach compared with the proposed method, a poorly matched 2D case, as previously reported in Figure 2-9, is considered as a case study here. As a sample, $\psi = 60^\circ$ matching case is taken and its inlier and outlier correspondences are visualized in the blue-to-

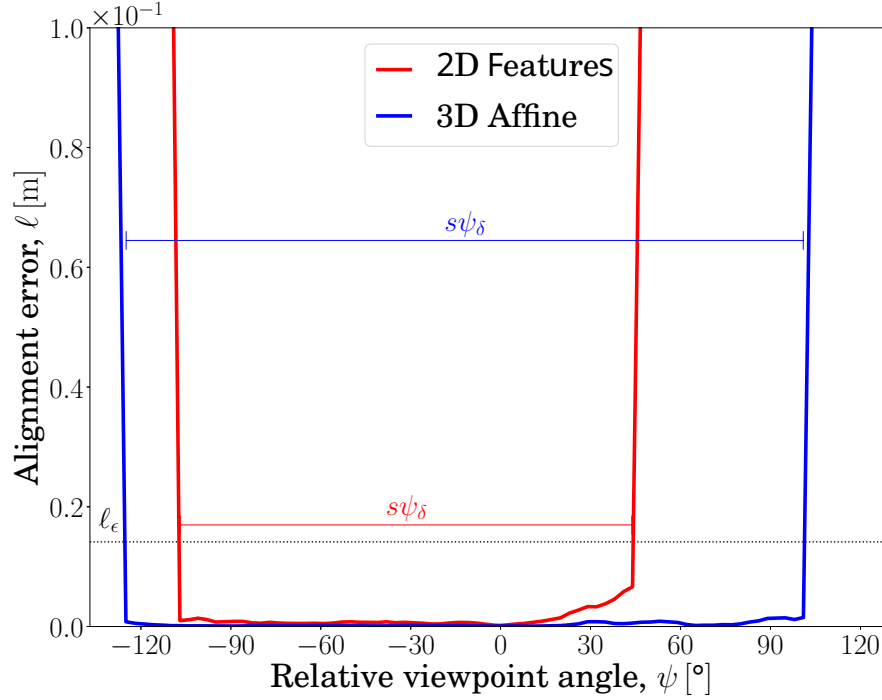


Figure 2-9: The proposed wrapper, denoted by the blue curve, is compared with the 2D feature approach, represented by the red curve. The proposed wrapper remained viewpoint invariant (i.e., $\ell(\psi) \leq \ell_\epsilon$) over a large interval of ψ -axis values. Thus, it outperformed the 2D intensity-based approach. The dotted horizontal line denotes the maximally tolerated alignment error, ℓ_ϵ ; the longer a curve, $\ell(\psi)$, remains below this line, the larger its invariance score, ψ_δ , where s is a normalizing constant. In this setting, the SURF [58] detector and the SIFT [37] descriptor were evaluated on a synthetic dataset resembling the ‘C. T. Crunch’ object observed from the NP1 camera.

red color range depending on how far they are from the ground truth. As shown in Figure 2-10a, it is apparent that the 2D detector/descriptor pair has a huge outlier ratio at the correspondence stage (Section 2.3.1). The low inlier ratio leads to RANSAC failing to recover a proper set of inliers, as shown in Figure 2-10b. On the other hand, despite the proposed wrapper utilizing the exact same detector and descriptor, it is able to produce better correspondences with a higher inlier ratio, thus properly filtering out the outliers at the geometric verification stage (Section 2.3.2).

Now, to delve further into answering why the 2D feature has a low inlier ratio, the detection and description stages are investigated independently. By constructing an ideal descriptor based on the ground truth, as shown in Figure 2-10c, ideal correspondences can be obtained for any set of detected keypoints. The 2D detected keypoints in the corresponding region have a lower density than those of the 3D-affine methods. With an almighty descriptor and without synthesized factors (e.g., scale, illumination, blur, or contrast) apart from the viewpoint change, results suggest that the detection repeatability decreases with large viewpoint difference. On the other hand, despite using the same detector, the proposed 3D-affine wrapper was able to bypass such a scenario by warping the surfaces (Section 2.2.4) into viewpoint-independent RGB-D images, enabling a more repeatable detection from both source and destination frames.

Similarly, the descriptor can be studied independently by limiting the set of k -NN correspondences (Figure 2-10a) to those sharing the same domain set with the ideal ones (Figure 2-10c), as shown in Figure 2-10d. In this case, most of the 2D correspondences, if not all, paired with the wrong keypoint, despite the existence of an ideal corresponding keypoint in the range set. This indicates that, due to the

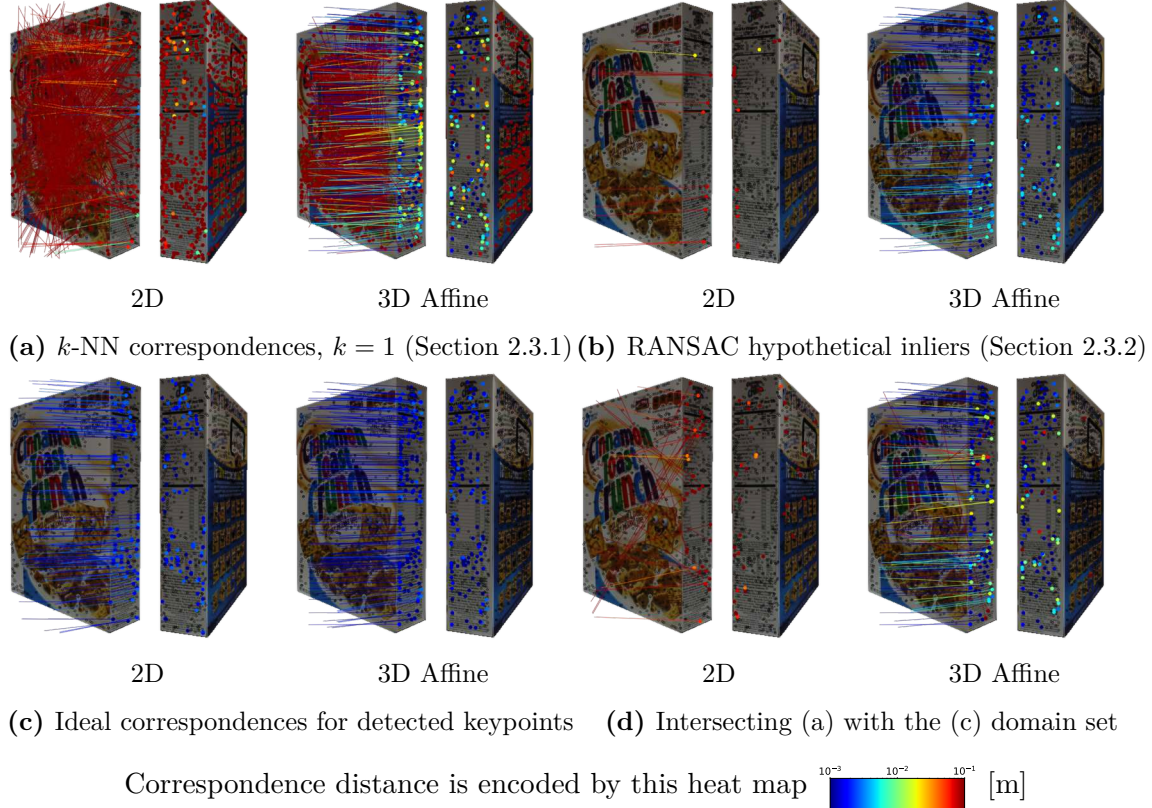


Figure 2-10: The 2D SURF [58] detector and SIFT [37] descriptor had difficulty detecting sufficient repeatable keypoints (left sides of **a**, **b**, and **c**) and computing distinct descriptions (left sides of **a**, **b**, and **d**), respectively, under relatively-large viewpoint change. The proposed wrapper (3D affine) alleviated the problem by leveraging the geometric information from the depth maps to undo the viewpoint change effects (right side of **b**, refer to Section 2.5.1 for details). Correspondences are visualized as a sparse optical-flow to improve visibility.

viewpoint change, indistinct descriptions were computed for these ideally repeatable keypoints, resulting in higher matching distances than those measured with other invalid pairs. Failing to generate proper descriptions suggests that the sampling window cannot invariantly describe large skew-like neighborhood changes to handle the viewpoint difference. Even small sampling windows cannot alleviate the situation, which implies the need for the elliptical keypoints (Figure 2-6a). Contrarily, the proposed method renders the surfaces as viewpoint independent representations. Thus, it is bestowed with more distinctiveness (not perfect though), albeit using the same descriptor.

Real-World Data

After the in-depth insights obtained from the previously discussed experiment, the focus is now on demonstrating the proposed method’s effectiveness and generality through extensive experiments. The discussion is grouped into three segments based on the utilized datasets, detectors, and descriptors. After that, more comprehensive results are reported, and the section is concluded with a discussion of the proposed method’s limitations. For conciseness, it is deemed sufficient to state the viewpoint invariance scores, ψ_δ , given that most of the detector/descriptor pairs feature similar, yet noisy, tendencies to the curves in Figure 2-9, and their viewpoint invariance is well captured by the ψ_δ metric from Equation (2.10).

Although the viewpoint invariance depends on the detectors/descriptors invariance formula(s), it is also affected by the evaluated scene properties (e.g., texture, blur, contrast, object geometry, and the number of distinct surfaces observed on it). Furthermore, the score is affected by source frame selection, especially if repeated

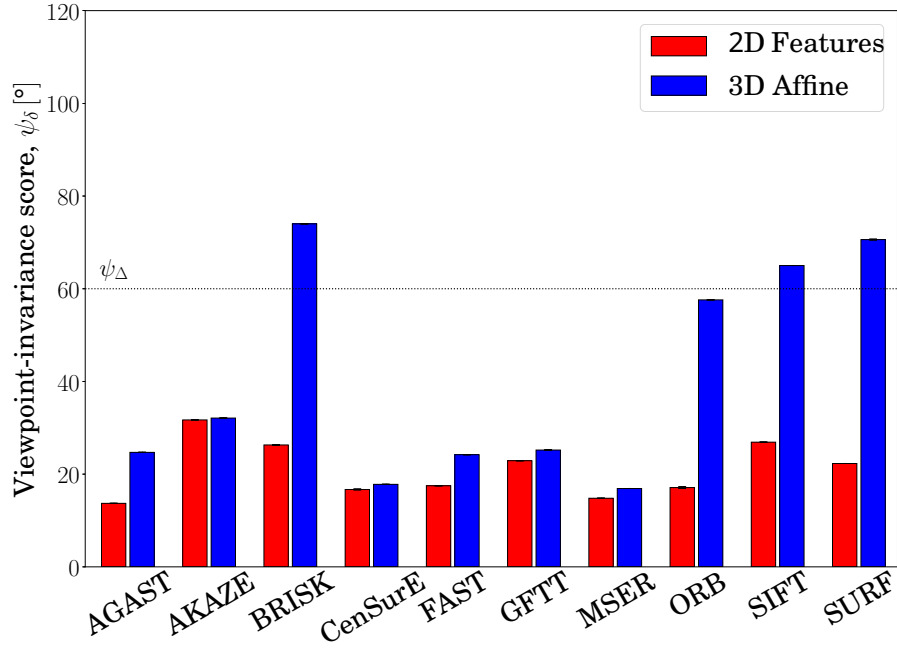


Figure 2-11: Viewpoint invariance for each detector intermixed with the SIFT [37] descriptor. Ten different detectors were evaluated. The three that exceed the targeted viewpoint invariance, ψ_Δ , as denoted by the dotted horizontal line, were wrapped in the proposed wrapper, represented by the blue bars. On the other hand, none of the 2D features approached the goal, as their red bars indicate. This figure utilizes the data from the ‘C. T. Crunch’ dataset observed from the NP3 camera. More extensive results are presented in later parts of Section 2.5.1.

texture, blurry regions, or few surfaces are observed on it. To demonstrate this, a total of nine different combinations—composed of four different objects, two different elevations, and several azimuth angles (more than 120 samples each)—were utilized, as shown in Figure 2-8. Although the next discussion of the different datasets is limited to the representative SURF detector and SIFT descriptor, similar observations can be made for the 18 remaining detectors/descriptors pairs, as reported later in this section.

In Figure 2-12, the horizontal axis denotes different datasets and cameras, the vertical axis denotes the viewpoint invariance score, ψ_δ , and the black dotted horizontal line denotes the goal viewpoint invariance threshold, ψ_Δ . The 2D and 3D Affine local image feature are denoted by the red and blue bars, respectively. The proposed wrapper achieved the targeted viewpoint invariance, ψ_Δ , for most datasets featuring polygonal objects and distinct surface discontinuities; however, the 2D feature failed to approach the target for any of the datasets. It is worth noting that the scores reported in this realistic setup are less than the equivalent scores reported in Section 2.5.1, which is a natural outcome, given that the previous experiment was in a synthetic setting. In fact, the differences in the score levels correspond to the dataset difficulty in terms of geometry, texture, blur, contrast, and the number of observed surfaces. For example, the ‘Cheez It’ dataset observed from the NP3 camera was very challenging for the 2D feature approach (only 6.7°), because it included small and repeated texture patterns; however, the proposed wrapper almost succeeded (55.1°) by exploiting geometric information resident in the scene. The most challenging case for the proposed wrapper involved indistinct surface discontinuity (the ‘Pringles’ dataset; see the last paragraph in this section for details).

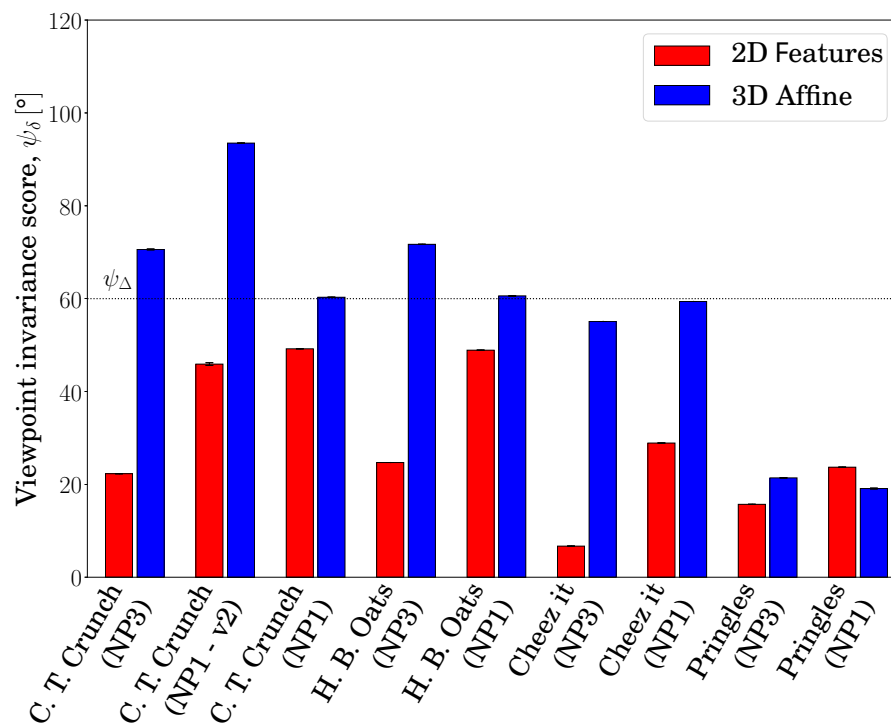


Figure 2-12: Viewpoint invariance depends on the object geometry, texture, and the number of observed distinct surfaces. The vertical axis represents the viewpoint invariance score, ψ_δ , while the horizontal axis represents different datasets, observed from the NP1 and NP3 cameras (Figure 2-8). The red bars and the blue bars represent the 2D and the 3D Affine local image features, respectively. Generally, the proposed method outperformed the 2D one for the vast majority of the datasets, where none of the 2D features approached the targeted viewpoint invariance, ψ_Δ . The data in this figure are from experiments with the SURF [58] detector and the SIFT [37] descriptor. Refer to later parts of Section 2.5.1 for the results of other detectors/descriptors.

Importantly, the more surfaces observed, the better the achieved viewpoint invariance of the proposed 3D Affine features, and the more deteriorated the viewpoint invariance obtained by the 2D features. This is captured in Figure 2-12 by the bars denoting the ‘C. T. Crunch’ dataset with different source RGB-D image surfaces (see the $\psi = 0^\circ$ column in Figure 2-8). Similarly, relative to the maximum possible viewpoint invariance score, $\psi_{\delta, \max} = \frac{1}{s}(\psi_{\max} - \psi_{\min})$, has its own column in Figure 2-8, the proposed wrapper generally scored a higher viewpoint invariance in scenes containing more surfaces, e.g., datasets observed from the NP3 camera. The highest viewpoint invariances, 74.8° , 66.15° , and 57.25° , were achieved for the ‘C. T. Crunch’, ‘H. B. Oats’, and ‘Cheez It’ datasets, respectively, which contained polygonal objects arranged from highly distinct patterns to more blurred and repeated texture patterns. Overall, the proposed wrapper improved the viewpoint invariance of the representative local image features in all datasets and cameras; the proposed approach had an average viewpoint invariance of 56.9° , compared with 29.5° in the 2D case.

Second, detectors generally benefited from the proposed wrapper, with those featuring abundant and large area keypoints gaining the most benefit. To illustrate this, Figure 2-11 summarizes the viewpoint invariance of the detectors with the ‘C. T. Crunch’ dataset and the NP3 camera (more extensive results are reported later in this section). Although none of the 2D detectors achieved the targeted viewpoint invariance, ψ_{Δ} , denoted by the dotted horizontal line, three of them (BRISK, SURF, and SIFT) achieved the target when wrapped in the proposed wrapper. Furthermore, the ORB detector was only 2.4° off the target. Although MSER produces large keypoint radii, both MSER and CenSurE only had a few keypoints, which affected

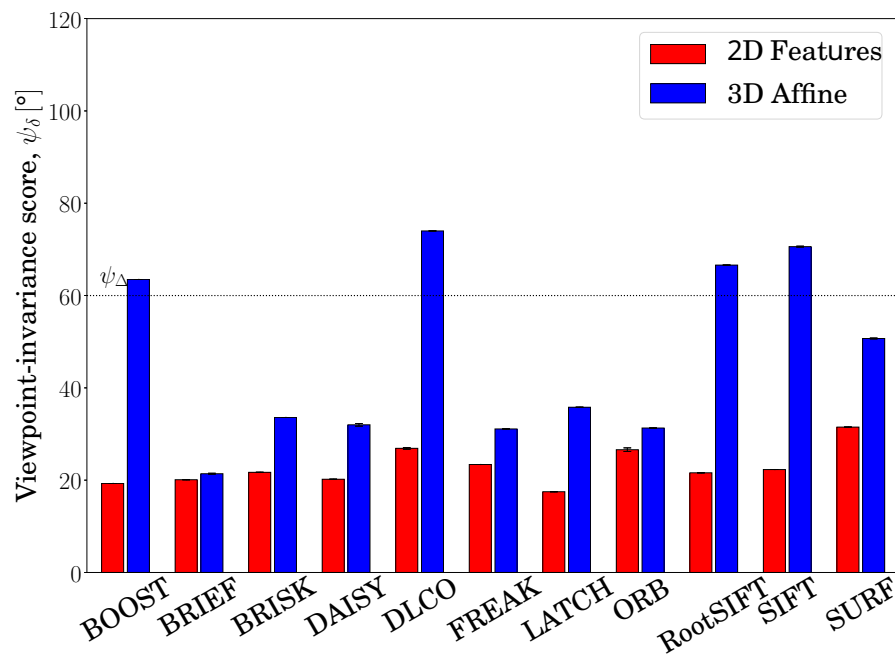


Figure 2-13: Viewpoint invariance for each descriptor intermixed with the SURF [58] detector. Eleven different descriptors were evaluated, with four of them achieving and one closely approaching the targeted viewpoint invariance threshold, ψ_Δ , after being wrapped in the proposed wrapper (blue bars). On the other hand, all 2D descriptors (red bars) missed the target. This figure utilizes the data from the ‘C. T. Crunch’ dataset and the NP3 camera. The rest of the data is presented in later parts of Section 2.5.1.

the correspondence establishment in both the 2D and proposed wrapper results. Accordingly, MSER and CenSurE remained below the targeted viewpoint invariance, but their proposed 3D Affine features still outsourced the 2D ones.

The highest viewpoint invariances, 74.0° , 70.6° , 65.0° , and 57.6° , were obtained with the BRISK, SURF, SIFT, and ORB detectors, which are characterized by a relatively high number of keypoints with medium-to-large radii. The detectors benefiting the most from the proposed wrapper were ORB and SURF, which have large keypoint radii. Overall, as detailed later in this section, the 2D detectors intermixed with the SIFT descriptor scored on all datasets an average viewpoint invariance of 33.6° , which increased to 46.3° when wrapped in the proposed wrapper. Detectors with abundant and considerably large keypoints benefited the most from being embedded in the method; however, for detectors with a small number of keypoints, the wrapper was inadequate.

For a third study, individual descriptors evaluated on keypoints detected by the SURF detector are analyzed, as reported in Figure 2-13. In this experiment, all descriptors benefited from the wrapper wrapping, where 4 of 11 achieved the targeted viewpoint invariance, ψ_Δ . The highest viewpoint invariances, 70.4° , 70.6° , 66.6° , and 63.5° , were attained by DLCO, SIFT, RootSIFT, and BOOST descriptors, respectively. The descriptors benefiting the most from the proposed wrapper were BOOST, SIFT, and RootSIFT. Although studying each detector invariance formula is out of scope of this chapter, these results are most likely affected by such formulas. Overall, the average viewpoint invariance of the 2D descriptors on all datasets was 32.7° , which increased to 45.7° when wrapped in the proposed wrapper.

Table 2.3: Overall viewpoint invariance, ψ_δ , for various features and datasets. The scores exceeding the targeted threshold, ψ_Δ , are shown in bold. Detectors and descriptors are detailed in Table 2.2.

Detect./Desc.	C. T. Crunch			H. B. Oats		Cheez It		Pringles		Average
	NP3	NP1 - v2	NP1	NP3	NP1	NP3	NP1	NP3	NP1	
AGAST/SIFT	13.7° ± 0.0°	55.2° ± 0.0°	56.5° ± 0.0°	20.3° ± 0.0°	56.1° ± 0.0°	12.7° ± 0.0°	54.9° ± 0.0°	15.7° ± 0.0°	24.7° ± 0.0°	34.4° ± 0.0°
AKAZE/SIFT	31.7° ± 0.0°	52.7° ± 0.0°	55.1° ± 0.0°	29.2° ± 0.0°	53.2° ± 0.0°	6.2° ± 0.0°	47.5° ± 0.0°	15.4° ± 0.0°	25.0° ± 0.0°	35.1° ± 0.0°
BRISK/SIFT	26.3° ± 0.1°	51.9° ± 0.0°	53.9° ± 0.1°	39.1° ± 0.3°	54.1° ± 0.0°	15.0° ± 0.0°	50.5° ± 0.1°	17.1° ± 0.0°	24.2° ± 0.0°	36.9° ± 0.1°
CenSurE/SIFT	16.7° ± 0.1°	53.3° ± 0.0°	56.7° ± 0.0°	14.1° ± 0.0°	54.7° ± 0.0°	13.7° ± 0.0°	54.6° ± 0.0°	1.9° ± 0.0°	18.2° ± 0.1°	31.5° ± 0.0°
FAST/SIFT	17.5° ± 0.0°	55.8° ± 0.0°	55.6° ± 0.0°	20.7° ± 0.0°	55.2° ± 0.0°	18.8° ± 0.0°	55.9° ± 0.0°	15.8° ± 0.0°	24.3° ± 0.0°	35.5° ± 0.0°
GFTT/SIFT	22.9° ± 0.0°	77.7° ± 0.1°	59.2° ± 0.1°	25.6° ± 0.9°	60.5° ± 0.0°	19.6° ± 0.0°	57.9° ± 0.0°	19.1° ± 0.0°	40.3° ± 0.0°	42.5° ± 0.3°
MSER/SIFT	14.8° ± 0.0°	40.7° ± 0.3°	50.2° ± 0.0°	18.2° ± 0.0°	43.6° ± 0.0°	0.0° ± 0.0°	31.3° ± 0.0°	8.8° ± 0.0°	21.7° ± 0.3°	25.5° ± 0.2°
ORB/SIFT	17.1° ± 0.2°	37.9° ± 0.0°	52.0° ± 0.0°	23.1° ± 0.0°	45.8° ± 0.0°	12.3° ± 0.0°	44.4° ± 0.0°	16.5° ± 0.0°	23.8° ± 0.0°	30.3° ± 0.1°
SIFT	26.9° ± 0.0°	53.4° ± 0.0°	55.3° ± 0.0°	31.0° ± 0.0°	53.1° ± 0.1°	0.0° ± 0.0°	52.2° ± 0.0°	21.2° ± 0.1°	18.4° ± 0.0°	34.6° ± 0.0°
SURF	31.5° ± 0.0°	52.9° ± 0.4°	51.9° ± 0.0°	28.3° ± 0.0°	56.6° ± 0.0°	19.0° ± 0.0°	49.3° ± 0.0°	17.7° ± 0.0°	27.9° ± 0.0°	37.2° ± 0.1°
SURF/BOOST	19.3° ± 0.0°	49.2° ± 0.1°	48.1° ± 0.1°	20.5° ± 0.0°	47.4° ± 0.0°	6.3° ± 0.0°	39.6° ± 0.0°	15.7° ± 0.0°	24.3° ± 0.0°	30.0° ± 0.0°
SURF/BRIEF	20.1° ± 0.0°	52.2° ± 0.0°	55.5° ± 0.2°	23.3° ± 0.0°	54.4° ± 0.2°	7.8° ± 0.0°	55.8° ± 0.0°	16.4° ± 0.0°	24.3° ± 0.0°	34.4° ± 0.1°
SURF/BRISK	21.7° ± 0.0°	50.4° ± 0.0°	48.7° ± 0.0°	23.3° ± 0.1°	50.9° ± 0.1°	3.5° ± 0.0°	49.3° ± 0.0°	17.0° ± 0.0°	23.7° ± 0.0°	32.1° ± 0.0°

Continued on next page

Table 2.3 – continued from previous page

SURF/DAISY	$20.2^\circ \pm 0.0^\circ$	$56.3^\circ \pm 0.0^\circ$	$56.1^\circ \pm 0.0^\circ$	$26.0^\circ \pm 0.0^\circ$	$57.1^\circ \pm 0.0^\circ$	$22.8^\circ \pm 0.0^\circ$	$50.1^\circ \pm 0.0^\circ$	$17.3^\circ \pm 0.0^\circ$	$24.5^\circ \pm 0.0^\circ$	$36.7^\circ \pm 0.0^\circ$
SURF/DLCO	$26.9^\circ \pm 0.2^\circ$	$52.8^\circ \pm 0.0^\circ$	$50.9^\circ \pm 0.0^\circ$	$28.3^\circ \pm 0.0^\circ$	$52.8^\circ \pm 0.0^\circ$	$8.8^\circ \pm 0.0^\circ$	$40.5^\circ \pm 0.0^\circ$	$16.4^\circ \pm 0.0^\circ$	$23.9^\circ \pm 0.0^\circ$	$33.5^\circ \pm 0.1^\circ$
SURF/FREAK	$23.4^\circ \pm 0.0^\circ$	$46.7^\circ \pm 0.2^\circ$	$49.8^\circ \pm 0.1^\circ$	$23.7^\circ \pm 0.0^\circ$	$45.1^\circ \pm 0.0^\circ$	$4.4^\circ \pm 0.0^\circ$	$47.6^\circ \pm 0.0^\circ$	$18.1^\circ \pm 0.0^\circ$	$24.3^\circ \pm 0.0^\circ$	$31.5^\circ \pm 0.1^\circ$
SURF/LATCH	$17.5^\circ \pm 0.0^\circ$	$48.7^\circ \pm 0.0^\circ$	$48.5^\circ \pm 0.0^\circ$	$16.8^\circ \pm 0.0^\circ$	$47.9^\circ \pm 0.1^\circ$	$12.8^\circ \pm 0.0^\circ$	$42.0^\circ \pm 0.0^\circ$	$16.5^\circ \pm 0.0^\circ$	$24.1^\circ \pm 0.0^\circ$	$30.5^\circ \pm 0.0^\circ$
SURF/ORB	$26.6^\circ \pm 0.4^\circ$	$50.9^\circ \pm 0.0^\circ$	$49.4^\circ \pm 0.0^\circ$	$24.5^\circ \pm 0.2^\circ$	$51.7^\circ \pm 0.0^\circ$	$21.4^\circ \pm 0.0^\circ$	$46.0^\circ \pm 0.0^\circ$	$18.0^\circ \pm 0.0^\circ$	$29.6^\circ \pm 0.0^\circ$	$35.3^\circ \pm 0.1^\circ$
SURF/RootSIFT	$21.6^\circ \pm 0.0^\circ$	$44.2^\circ \pm 0.0^\circ$	$46.7^\circ \pm 0.0^\circ$	$24.0^\circ \pm 0.0^\circ$	$46.1^\circ \pm 0.0^\circ$	$6.0^\circ \pm 0.0^\circ$	$28.9^\circ \pm 0.0^\circ$	$15.7^\circ \pm 0.0^\circ$	$24.2^\circ \pm 0.0^\circ$	$28.6^\circ \pm 0.0^\circ$
SURF/SIFT	$22.3^\circ \pm 0.0^\circ$	$45.9^\circ \pm 0.3^\circ$	$49.2^\circ \pm 0.0^\circ$	$24.7^\circ \pm 0.0^\circ$	$48.9^\circ \pm 0.0^\circ$	$6.7^\circ \pm 0.0^\circ$	$28.9^\circ \pm 0.0^\circ$	$15.7^\circ \pm 0.0^\circ$	$23.7^\circ \pm 0.0^\circ$	$29.5^\circ \pm 0.1^\circ$
Average	$21.9^\circ \pm 0.1^\circ$	$51.4^\circ \pm 0.1^\circ$	$52.5^\circ \pm 0.1^\circ$	$24.2^\circ \pm 0.2^\circ$	$51.8^\circ \pm 0.0^\circ$	$10.9^\circ \pm 0.0^\circ$	$46.4^\circ \pm 0.0^\circ$	$15.8^\circ \pm 0.0^\circ$	$24.8^\circ \pm 0.1^\circ$	$33.3^\circ \pm 0.1^\circ$
$\psi_\delta/\psi_{\delta,\max}$ ratio	29.2 %	38.07 %	58.33 %	32.27 %	57.56 %	14.5 %	51.56 %	21.07 %	30.61 %	37.02 %
AGAST/SIFT	$24.7^\circ \pm 0.0^\circ$	$96.8^\circ \pm 0.0^\circ$	$61.9^\circ \pm 0.0^\circ$	$27.0^\circ \pm 0.0^\circ$	$64.0^\circ \pm 0.0^\circ$	$22.9^\circ \pm 0.0^\circ$	$62.8^\circ \pm 0.0^\circ$	$26.6^\circ \pm 0.0^\circ$	$56.3^\circ \pm 0.5^\circ$	$49.2^\circ \pm 0.2^\circ$
AKAZE/SIFT	$32.1^\circ \pm 0.0^\circ$	$57.9^\circ \pm 0.0^\circ$	$63.1^\circ \pm 0.0^\circ$	$31.3^\circ \pm 0.2^\circ$	$61.8^\circ \pm 0.0^\circ$	$14.2^\circ \pm 0.2^\circ$	$62.5^\circ \pm 0.0^\circ$	$11.0^\circ \pm 0.0^\circ$	$0.0^\circ \pm 0.0^\circ$	$37.1^\circ \pm 0.1^\circ$
BRISK/SIFT	$74.0^\circ \pm 0.0^\circ$	$56.0^\circ \pm 0.0^\circ$	$60.1^\circ \pm 0.0^\circ$	$49.4^\circ \pm 0.0^\circ$	$60.3^\circ \pm 0.0^\circ$	$55.6^\circ \pm 0.0^\circ$	$59.3^\circ \pm 0.0^\circ$	$16.0^\circ \pm 0.0^\circ$	$13.5^\circ \pm 0.2^\circ$	$49.4^\circ \pm 0.1^\circ$
CenSurE/SIFT	$17.8^\circ \pm 0.0^\circ$	$63.8^\circ \pm 0.0^\circ$	$66.9^\circ \pm 0.0^\circ$	$28.1^\circ \pm 0.0^\circ$	$64.3^\circ \pm 0.0^\circ$	$0.0^\circ \pm 0.0^\circ$	$67.0^\circ \pm 0.0^\circ$	$0.0^\circ \pm 0.0^\circ$	$0.0^\circ \pm 0.0^\circ$	$34.2^\circ \pm 0.0^\circ$
FAST/SIFT	$24.2^\circ \pm 0.0^\circ$	$97.1^\circ \pm 0.0^\circ$	$61.7^\circ \pm 0.0^\circ$	$26.7^\circ \pm 0.0^\circ$	$63.7^\circ \pm 0.0^\circ$	$21.0^\circ \pm 0.0^\circ$	$62.7^\circ \pm 0.0^\circ$	$27.0^\circ \pm 0.0^\circ$	$57.4^\circ \pm 0.0^\circ$	$49.1^\circ \pm 0.0^\circ$
GFTT/SIFT	$25.2^\circ \pm 0.1^\circ$	$95.9^\circ \pm 0.0^\circ$	$61.1^\circ \pm 0.1^\circ$	$30.4^\circ \pm 0.0^\circ$	$63.3^\circ \pm 0.0^\circ$	$25.7^\circ \pm 0.0^\circ$	$61.9^\circ \pm 0.0^\circ$	$26.9^\circ \pm 0.0^\circ$	$54.4^\circ \pm 0.0^\circ$	$49.4^\circ \pm 0.1^\circ$
MSER/SIFT	$16.9^\circ \pm 0.0^\circ$	$55.3^\circ \pm 0.0^\circ$	$57.4^\circ \pm 0.0^\circ$	$25.6^\circ \pm 0.0^\circ$	$54.4^\circ \pm 0.0^\circ$	$0.0^\circ \pm 0.0^\circ$	$59.9^\circ \pm 0.0^\circ$	$5.2^\circ \pm 0.0^\circ$	$11.3^\circ \pm 0.0^\circ$	$31.8^\circ \pm 0.0^\circ$
ORB/SIFT	$57.6^\circ \pm 0.0^\circ$	$57.7^\circ \pm 0.0^\circ$	$61.0^\circ \pm 0.0^\circ$	$41.7^\circ \pm 0.0^\circ$	$61.6^\circ \pm 0.0^\circ$	$25.0^\circ \pm 0.0^\circ$	$60.8^\circ \pm 0.0^\circ$	$13.6^\circ \pm 0.0^\circ$	$30.0^\circ \pm 0.0^\circ$	$45.4^\circ \pm 0.0^\circ$
SIFT	$65.0^\circ \pm 0.0^\circ$	$89.8^\circ \pm 0.0^\circ$	$60.3^\circ \pm 0.0^\circ$	$75.0^\circ \pm 0.0^\circ$	$61.2^\circ \pm 0.0^\circ$	$61.2^\circ \pm 0.1^\circ$	$59.4^\circ \pm 0.0^\circ$	$21.5^\circ \pm 0.1^\circ$	$46.9^\circ \pm 0.5^\circ$	$60.0^\circ \pm 0.2^\circ$
SURF	$50.7^\circ \pm 0.1^\circ$	$91.5^\circ \pm 0.0^\circ$	$61.1^\circ \pm 0.0^\circ$	$48.1^\circ \pm 0.2^\circ$	$63.0^\circ \pm 0.0^\circ$	$43.7^\circ \pm 0.7^\circ$	$60.3^\circ \pm 0.0^\circ$	$21.7^\circ \pm 0.0^\circ$	$25.8^\circ \pm 0.0^\circ$	$51.8^\circ \pm 0.2^\circ$

Continued on next page

Table 2.3 – continued from previous page

SURF/BOOST	63.5° ± 0.0°	72.8° ± 0.1°	61.5° ± 0.0°	63.8° ± 0.0°	60.9° ± 0.1°	36.1° ± 0.8°	60.8° ± 0.0°	20.0° ± 0.0°	13.2° ± 0.8°	50.3° ± 0.4°
SURF/BRIEF	21.4° ± 0.1°	71.4° ± 0.0°	60.7° ± 0.0°	23.6° ± 0.0°	63.3° ± 0.0°	14.6° ± 0.0°	60.2° ± 0.0°	15.2° ± 0.0°	9.9° ± 0.1°	37.8° ± 0.1°
SURF/BRISK	33.6° ± 0.0°	55.5° ± 0.0°	60.4° ± 0.0°	32.7° ± 0.0°	60.0° ± 0.0°	0.0° ± 0.0°	60.9° ± 0.0°	16.0° ± 0.0°	0.0° ± 0.0°	35.5° ± 0.0°
SURF/DAISY	32.0° ± 0.3°	94.2° ± 0.1°	60.5° ± 0.0°	30.2° ± 0.0°	61.6° ± 0.0°	24.4° ± 0.0°	58.9° ± 0.0°	27.9° ± 0.0°	53.6° ± 0.1°	49.3° ± 0.1°
SURF/DLCO	74.0° ± 0.0°	94.0° ± 0.0°	61.5° ± 0.0°	64.5° ± 0.0°	62.8° ± 0.0°	46.0° ± 0.0°	59.3° ± 0.1°	20.5° ± 0.0°	16.4° ± 0.0°	55.4° ± 0.0°
SURF/FREAK	31.1° ± 0.0°	62.7° ± 0.0°	68.5° ± 0.0°	30.0° ± 0.0°	68.0° ± 0.0°	0.0° ± 0.0°	63.5° ± 0.0°	1.5° ± 0.0°	0.0° ± 0.0°	36.1° ± 0.0°
SURF/LATCH	35.8° ± 0.0°	56.6° ± 0.0°	60.1° ± 0.0°	38.7° ± 0.1°	60.0° ± 0.0°	18.2° ± 0.0°	60.0° ± 0.0°	19.1° ± 0.0°	1.8° ± 0.0°	38.9° ± 0.0°
SURF/ORB	31.3° ± 0.0°	55.6° ± 0.0°	61.8° ± 0.0°	36.4° ± 0.0°	61.7° ± 0.0°	5.6° ± 0.2°	60.3° ± 0.0°	17.9° ± 0.1°	5.5° ± 0.0°	37.3° ± 0.1°
SURF/RootSIFT	66.6° ± 0.0°	81.7° ± 0.0°	60.3° ± 0.0°	55.7° ± 0.0°	59.8° ± 0.0°	59.4° ± 0.0°	59.7° ± 0.0°	20.9° ± 0.0°	12.5° ± 0.2°	53.0° ± 0.1°
SURF/SIFT	70.6° ± 0.1°	93.5° ± 0.0°	60.3° ± 0.0°	71.7° ± 0.0°	60.6° ± 0.1°	55.1° ± 0.0°	59.4° ± 0.0°	21.4° ± 0.0°	19.1° ± 0.1°	56.9° ± 0.0°
Average	42.4° ± 0.1°	75.0° ± 0.0°	61.5° ± 0.0°	41.5° ± 0.1°	61.8° ± 0.0°	26.4° ± 0.2°	61.0° ± 0.0°	17.5° ± 0.0°	21.4° ± 0.2°	45.4° ± 0.1°
$\psi_\delta/\psi_{\delta,\max}$ ratio	56.53 %	55.56 %	68.33 %	55.33 %	68.67 %	35.20 %	67.78 %	23.33 %	26.42 %	50.79 %

More comprehensively, Table 2.3 reports the viewpoint invariance scores, ψ_δ , for 20 pairs of various detector/descriptor with nine different setups of the four evaluation datasets and the two cameras. Out of 140 combinations with distinct surface discontinuities, the proposed wrapper achieved the targeted viewpoint invariance, ψ_Δ , in 73 combinations using 19 3D Affine feature pairs out of the total 20 pairs. On the other hand, the 2D features met the target score in only two combinations using a single pair consisting of the GFTT detector and the SIFT descriptor.

These results confirm the generality of the proposed approach, in which almost all the evaluated detector/descriptor pairs gained a performance boost and achieved the targeted viewpoint invariance. Despite the first three datasets belonging to the same category of polygonal objects, the illumination, texture patterns, and contrast played an important role in establishing the difficulty of each dataset. Furthermore, with more distinct surfaces in the source RGB-D image, a higher score was achieved. In light of this, ‘Pringles’ was the most challenging dataset (see the last paragraph in this section) because of its indistinct surface discontinuities, thus no features were able to achieve the target score, in either the 2D or 3D Affine case. Averaging over all of the datasets, the local image features scored 45.4° when embedded in the proposed framework compared with 33.3° in the 2D case. By limiting the scope to distinct surface discontinuities, the average viewpoint invariance of 37° for 2D approaches is boosted to 52.8° when wrapped in the proposed wrapper .

Additionally, Figure 2-14 shows four sample qualitative cases. The first case has a relatively small common surface whose colors are similar to the other irrelevant surfaces, thereby causing the 2D SURF/SIFT to fail miserably. On the other hand, after wrapping the same detector/descriptor pair into the proposed wrapper , the alignment error, $\ell(60^\circ)$, reduced from 319.6 mm to 4.5 mm.

The second case involved a viewpoint change in both azimuth and elevation direc-

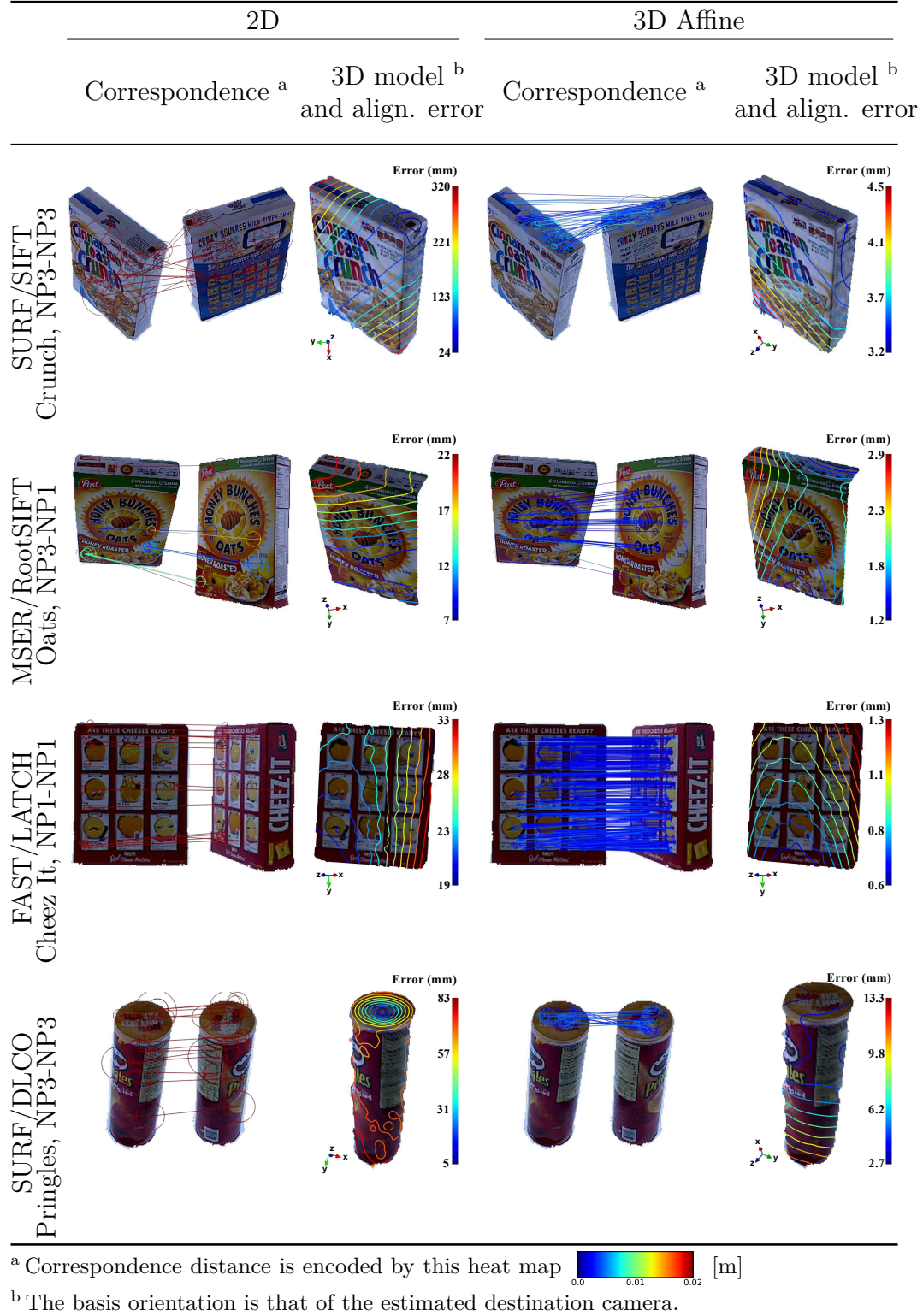


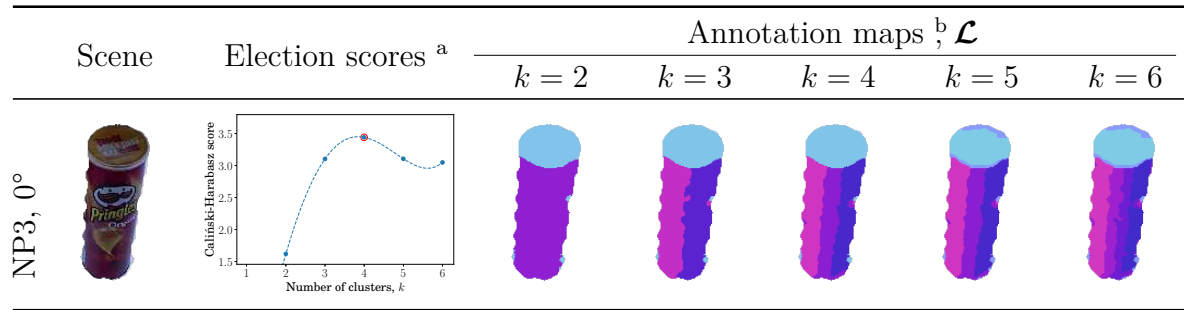
Figure 2-14: Sample qualitative cases of correspondences and 6D pose estimation using local image features, for the 2D and 3D Affine methods, under a wide viewpoint change. The alignment error, $\ell(\psi \approx 60^\circ)$, is visualized over the 3D model of the source RGB-D image.

tions, with less colorful surfaces and different illumination conditions, which reduced the inlier ratio. Despite this, the proposed wrapper handled the case properly.

In the third case, similar repeated patterns resulted in many outliers; however, again, the proposed wrapper recovered the correct correspondence with abundant inliers and an alignment error of only $\ell(60^\circ) = 1.4\text{ mm}$. The fourth case has a cylindrical surface, which is challenging for the labeling process (Section 2.2.2) and has a high outlier ratio. The proposed approach using a 3D Affine local image feature still outperformed the 2D one by using the top surfaces to recover a good correspondence.

Although the proposed wrapper had the lowest performance on the ‘Pringles’ dataset (Table 2.3), such a bottleneck is only due to the smooth-surface labeling phase (Section 2.2.2). As illustrated in Figure 2-15, the indistinct surface discontinuities cause the cylindrical surface to be segmented into thin longitudinal strips ($k \geq 3$ columns) that split some local regions, thus reducing their describability. Furthermore, connectivity issues are observed, in which non-uniform longitudinal strips (e.g., $k = \{3, 6\}$ columns) are formed, and some tiny fragments at the boundaries are scattered away from their corresponding clusters. Based on these observations, constraining the clusters by smoothness [98], incorporating k -NN connectivity [99], and performing non-rigid registration [100] of the curved surface into the virtual image plane, will all improve the overall performance, individually or collectively. Moreover, since only surface normals are employed in the label computation, it is also expected that employing depth can improve the results.

Regarding the execution time, it is unfair to directly compare the proposed method execution time to that of 2D local image features, since the proposal is implemented in Python while the compared methods are implemented in C++. However, for completeness purposes, Table 2.4 reports the execution time of both 2D and pro-



^a The non-integer k values of the extrapolation curve are for demonstration only, as they are not actually utilized nor have any meaningful interpretation.

^b Up to a rotation, RGB colors in the annotation map column encode the orientations in the 3D space.

Figure 2-15: Although approximating curved surfaces as planar segments improves the viewpoint representation, some interesting points are split apart, which drastically affects their descriptability.

posed 3D Affine variants of local image features. Knowing that Python is generally slower than C++ by an order of magnitude, it can be inferred to some extent that the proposed method would still be comparable to the compared methods in execution time as well. This is due to the fact, that the time complexity of the proposed algorithm is no different from that of the compared ones, except for the slight overhead of surfaces segmentation and warping.

2.5.2 Sensitivity Analysis Experiments

So far, several experiments have been conducted in ideal and realistic situations (Section 2.5.1), and the results demonstrated both effectiveness and generality of the proposal. In this section, the robustness of the proposed method to depth noise is demonstrated. Different SNRs were considered in a synthetic setup. Additionally, while it is sufficient to utilize one dataset, two datasets (resembling the second and third columns in Figure 2-8) were evaluated to avoid any possible bias.

The total of 90 matching experiments, as reported in Section 2.5.1, constitute

Table 2.4: The average execution time of both proposed and compared methods per each dataset. It might seems the proposed methods is significantly slower than the compared ones, but that is just because of implementation differences, in which proposed methods are written in Python, while the compared ones are C++ ones.

Meth.	Dataset	Features		
		Source	Destination	Matching
2D (C++ implementation)	C. T. Crunch (NP3)	128.0 ms \pm 137.6 ms	565.9 ms \pm 1763.2 ms	54.2 ms \pm 469.4 ms
	Cheez it (NP1 - v2)	416.4 ms \pm 317.6 ms	553.4 ms \pm 1708.4 ms	47.9 ms \pm 70.0 ms
	Cheez it (NP1)	357.1 ms \pm 259.4 ms	1165.9 ms \pm 3495.6 ms	57.6 ms \pm 148.7 ms
	H. B. Oats (NP3)	1409.0 ms \pm 5037.6 ms	1810.4 ms \pm 6547.0 ms	75.0 ms \pm 236.7 ms
	Cheez it (NP1)	251.9 ms \pm 185.6 ms	1199.2 ms \pm 4819.2 ms	73.5 ms \pm 124.6 ms
	Cheez it (NP3)	366.0 ms \pm 439.3 ms	4230.5 ms \pm 14595.1 ms	126.2 ms \pm 335.7 ms
	Cheez it (NP1)	194.3 ms \pm 198.6 ms	3879.6 ms \pm 13916.3 ms	199.0 ms \pm 426.1 ms
	Pringles (NP3)	95.8 ms \pm 85.9 ms	713.1 ms \pm 2026.2 ms	18.8 ms \pm 64.7 ms
	Cheez it (NP1)	135.9 ms \pm 143.3 ms	84.9 ms \pm 56.2 ms	6.5 ms \pm 6.9 ms
	Average	368.3 ms \pm 1616.2 ms	1518.3 ms \pm 7334.1 ms	72.9 ms \pm 258.0 ms
3D Affine (Python implementation)	C. T. Crunch (NP3)	5749.6 ms \pm 791.7 ms	5201.7 ms \pm 7479.9 ms	283.5 ms \pm 819.0 ms
	Cheez it (NP1 - v2)	9856.4 ms \pm 1019.3 ms	5013.5 ms \pm 1785.2 ms	142.2 ms \pm 245.9 ms
	Cheez it (NP1)	8886.9 ms \pm 1359.1 ms	5628.8 ms \pm 3919.0 ms	209.4 ms \pm 729.1 ms
	H. B. Oats (NP3)	5610.5 ms \pm 442.7 ms	5273.1 ms \pm 9952.6 ms	185.3 ms \pm 381.6 ms
	Cheez it (NP1)	8115.5 ms \pm 804.6 ms	5270.3 ms \pm 4671.0 ms	171.6 ms \pm 436.7 ms
	Cheez it (NP3)	8017.5 ms \pm 18182.1 ms	7967.9 ms \pm 22582.4 ms	182.7 ms \pm 421.4 ms
	Cheez it (NP1)	6894.7 ms \pm 1903.7 ms	7108.6 ms \pm 17240.6 ms	331.8 ms \pm 678.2 ms
	Pringles (NP3)	3175.8 ms \pm 813.1 ms	3736.1 ms \pm 3933.1 ms	87.3 ms \pm 217.3 ms
	Cheez it (NP1)	3325.0 ms \pm 221.2 ms	2154.6 ms \pm 337.0 ms	6.1 ms \pm 11.1 ms
	Average	6918.9 ms \pm 6149.9 ms	5261.4 ms \pm 10308.8 ms	177.2 ms \pm 503.2 ms

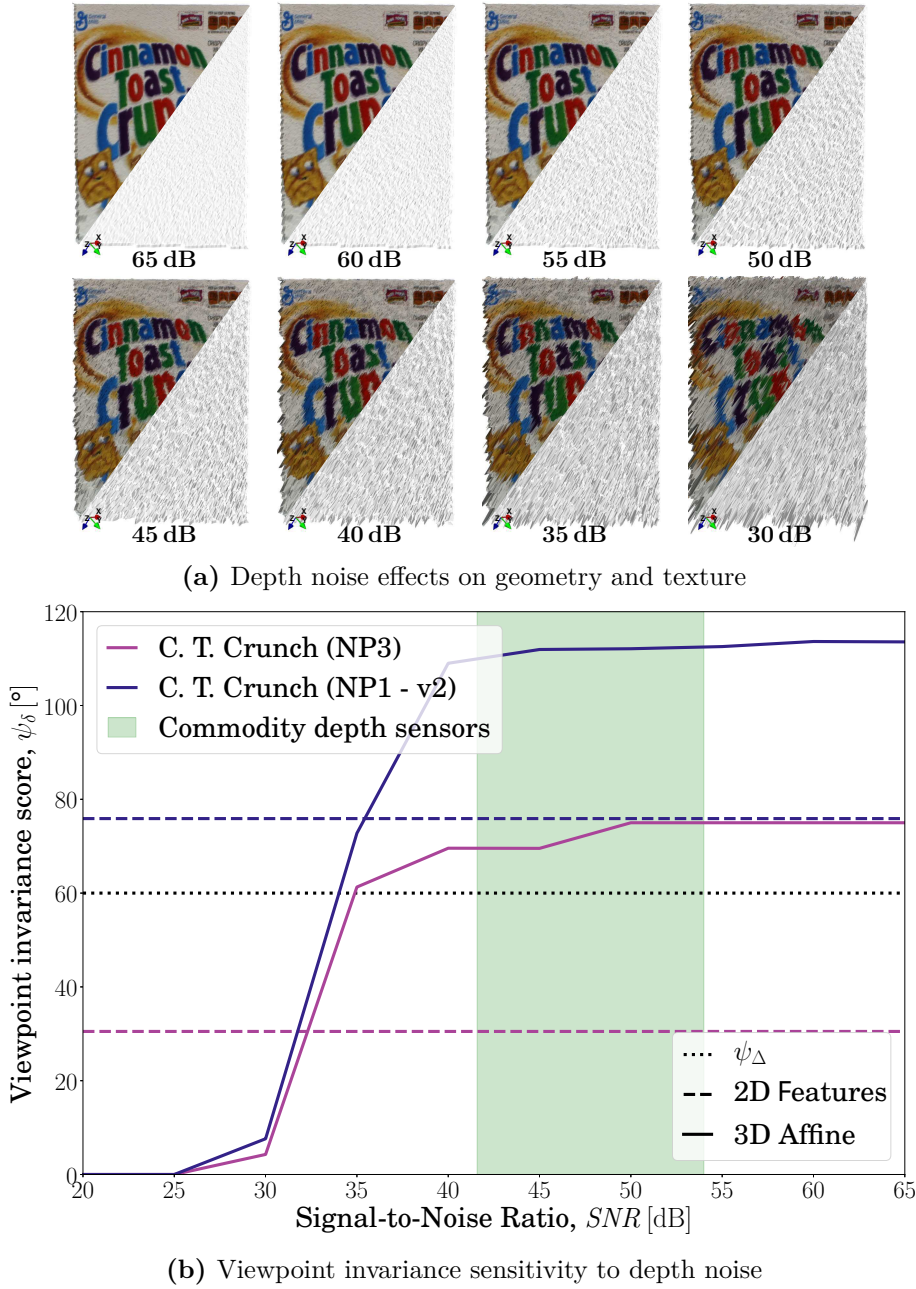


Figure 2-16: (a) a demonstration of different tested noise levels. The frame shows one of the 140 synthetic frames utilized, which simulated the ‘C. T. Crunch’ dataset with different viewpoints and cameras (the first two rows of Figure 2-8); (b) the proposed wrapper achieved the targeted viewpoint invariance, ψ_Δ , for a wide range of depth noise, $SNR \geq 35$ dB, in both datasets, and it outperformed the 2D intensity-based approach in estimating viewpoint rotations despite the ideal intensity setting, in which no illumination or blur effects were synthesized.

just 5% of this experiment. The ideal setup experiments were performed on the synthetic ‘C. T. Crunch (NP1 - v2)’ dataset. The results of the noise-free experiment are viewpoint invariances of 75.89° and 113.53° for the 2D and proposed approaches, respectively, which corresponds to $SNR = 65$ dB. As a reference, the scores obtained at $SNR = 65$ dB of the 2D approach are kept constant throughout the entire range, even though their scores actually degraded with the depth noise since one needs to compute the keypoint 3D coordinates (refer to Section 2.3.2 for details).

Figure 2-16 shows the proposed approach response to different SNR levels. To clearly demonstrate the evaluated noise levels, Figure 2-16a shows a sample RGB-D frame projected into 3D with different noise levels. To better exhibit the noise effects, the frame is observed from a different viewpoint than that of its camera. It is worth noting that, even under ideal situations, some tiny protuberances can be seen on the geometry (the lower-triangle part), but it is only at high noise levels that the eyes notice any texture defects.

The proposed method generates a proportional response to different SNRs and demonstrates robustness to noise starting from 35 dB and higher, as shown in Figure 2-16b. At around the same $SNR = 35$ dB value, the proposed wrapper outperformed the 2D approach in both datasets while also achieving the targeted viewpoint invariance, ψ_Δ . Starting around 40 dB–50 dB, the curves approach a saturation value, which is related to the dataset-dependent maximum viewpoint invariance score (denoted by the $\psi_{\delta, \max}$ in Figure 2-8). Knowing that low-cost commodity depth sensors exhibit 50 dB–54 dB within the first meter and subsequently deteriorates quadratically with depth to 41.58 dB–46.62 dB at a 3 m depth [101, 102], it is safe to conclude that the proposed approach can operate robustly in the presence of input noise, and well beyond. Accordingly, the proposed wrapper might even operate without requiring an external depth-map input; instead, it may compute the depth maps from

the RGB image sequences using approaches like DTAM (dense tracking and mapping) [103], despite any potential high noise levels. This provides an interesting point worthy of further investigation.

2.6 Conclusions and Future Work

Despite their variety, local image features suffer from a lack of robustness to viewpoint difference. Improving the viewpoint invariance has compelling applications in different fields and is gaining increasing attention, especially with the availability of geometric information attained through depth sensors. In this chapter, a general wrapper framework is proposed to empower 2D local image features with viewpoint invariance in a generic manner, without depending on a specific local image feature type or requiring interface or internal adaptations. The proposed wrapper utilizes depth information to annotate smooth surfaces and then warps the annotated surfaces to a viewpoint invariant representation, wherein all the feature-related computations take place. A nonparametric labeling of smooth surfaces is proposed to achieve robust annotation of distinct surfaces by clustering surface normals using the spherical k -means algorithm, where k is determined by the Caliński–Harabasz score. Invariant warping is achieved using a hybrid rigid–homography method, where a good balance between quality and accuracy is maintained, thereby improving the repeatability and distinctiveness of the detected and extracted features. Keypoints are then detected, and descriptors are computed from the viewpoint invariant representation, followed by keypoint remapping to the input space. The proposed wrapper increased the stability against viewpoint difference (i.e., *viewpoint invariance*) by leveraging the geometric information of a given scene. The performance was evaluated quantitatively by measuring the accumulative length of the relative viewpoint axis having

tolerated error.

Initially, a brief comparison in an ideal setup was conducted between the proposed wrapper and the highly respected SIFT/SURF detector and descriptor. The 2D density-based approach could not keep up with large out-of-plane rotations, and it turned out that both the detector repeatability and the descriptor distinctiveness are affected by viewpoint difference. While the detector repeatability was not severely affected, the descriptors face much greater difficulty in computing distinct descriptions for physically identical keypoints observed from different viewpoints. Most of the issue lies in the sampling-window size, which has no information about the underlying geometry. In this case, the best strategy, i.e., decreasing the keypoint area, also fails. On the contrary, the proposed method facilitated more repeatable detection and more distinctive description with its viewpoint-independent surface representation.

Furthermore, empirical results on real-world datasets showed that various detectors and descriptors, regardless of their invariance formula(s) or implementation, benefited from the proposed wrapper. This was tested on several datasets to varying extents, and depended on the keypoint size and object geometry. For keypoint size, the proposed wrapper best contributes to the increased viewpoint invariance of detectors that have abundant and considerably large keypoints. Ten different local image detectors intermixed with the SIFT descriptor are evaluated, and they obtained an average viewpoint invariance of 46.3° when wrapped in the proposed wrapper, compared with 33.6° in the 2D case. Similarly, 11 different local image descriptors were intermixed with the SURF detector, and they achieved an overall average viewpoint invariance of 45.7° when embedded in the proposed framework, as compared with 32.7° in the 2D case.

For object geometry, the best results achieved were with polygonal objects, whose

surface discontinuities are more distinct. Furthermore, objects with bright colors and distinct patterns were associated with higher scores in the proposed wrapper than those obtained with blurrier and repeated-pattern objects. Out of 140 test cases involving 20 detector/descriptor pairs and distinct-surface discontinuities, the proposed wrapper scored an average viewpoint invariance of 52.8° and achieved the targeted viewpoint invariance, ψ_Δ , in 73 cases belonging to 19 pairs as compared with an average of 37° and only two cases reaching the target, ψ_Δ , both belonging to a single 2D feature pair. The wrapper demonstrated some generality, based on the fact that 19 out of 20 local image features benefited from it, and scored the target invariance in several cases, while only one 2D local image feature managed to score in two test cases. Overall, 2D features had an average viewpoint invariance of 33.3° on all evaluated datasets, which improved to 45.4° with the proposed wrapper. In terms of noise sensitivity, the proposed method proved tolerant to noise beyond the requirements of 41.58 dB imposed by low-cost commodity depth sensors within the first 3 m of depth, as it can withstand SNRs as low as 35 dB.

This study addresses keypoint detection and feature computation under a wide baseline and demonstrates a large viewpoint invariance gain in cases of distinct surface discontinuities. On the other hand, the proposed wrapper was unable to increase the viewpoint invariance for detectors of few keypoints or objects of complex geometries and indistinct surface-normal discontinuities. This limitation stems from the smooth-surface labeling phase, in which the spherical k -means utilizes only the surface normals to label different surfaces. To overcome this limitation, several possibilities are considered, such as including depth discontinuities, constraining the clusters by k -NN pixel-based connectivity [99], imposing smoothness [98] constraints, or performing non-rigid registration [100] of the surface after the rigid alignment to the virtual image plane. Furthermore, there remains room for improvement in regard

to the current hybrid rigid-homography method, which depends on the homography transforms, where index interpolation with iterative back-projection [104] represents another alternative and constitutes a point of future work. Currently, depth input is needed, which is a reasonable requirement given the prevalence of low-cost commodity depth sensors nowadays. However, since the proposed method can handle low SNRs, computing depth maps from RGB image sequences using approaches like DTAM [103] might be an interesting extension of this research. Similarly, depth maps are currently interpolated for missing-depth small regions while neglecting larger ones; incorporating gradient-aware depth painting/interpolation constitutes a potential future direction. Similarly, it seems beneficial to utilize a trilateral filter [105] to smooth the depth map, as it incorporates gradients to achieve commendable results.

Various applications, including 6D pose estimation, will be able to benefit from the proposed wrapper beyond what was demonstrated in this chapter, by using normal and gradient orientations of the region-affine keypoints along with their center points (refer to Section 2.3.2 for details). It is also believed that the current work can be utilized in non-rigid environments by replacing RANSAC-based matching with a global-motion modeling technique, such as factorized graph matching [106]. Additionally, fine alignment can be implemented by using ICP (iterative closest/corresponding point) variants (color [107] or point-to-plane [108]). Global alignment with sensor-uncertainty modeling using sparse-surface adjustment [109] appears to improve high-level performance, especially when considering multi-way alignment. Ultimately, the plan is to adapt this work to a 3D-reconstruction application.

Chapter 3

A Single-structure Voting Scheme

This section is scheduled to be published as part of a journal, thus it is not available here.

3.1 Introduction

This section is scheduled to be published as part of a journal, thus it is not available here.

3.2 Methodology

This section is scheduled to be published as part of a journal, thus it is not available here.

3.3 Experimental Setup

This section is scheduled to be published as part of a journal, thus it is not available here.

3.4 Results and Discussion

This section is scheduled to be published as part of a journal, thus it is not available here.

3.5 Conclusions and Future Work

This section is scheduled to be published as part of a journal, thus it is not available here.

Chapter 4

Multi-structure Hypotheses Generation

This section is scheduled to be published as part of a journal, thus it is not available here.

4.1 Introduction

This section is scheduled to be published as part of a journal, thus it is not available here.

4.2 Methodology

This section is scheduled to be published as part of a journal, thus it is not available here.

4.3 Experimental Setup

This section is scheduled to be published as part of a journal, thus it is not available here.

4.4 Results and Discussion

This section is scheduled to be published as part of a journal, thus it is not available here.

4.5 Conclusions and Future Work

This section is scheduled to be published as part of a journal, thus it is not available here.

Chapter 5

Conclusion and Future Work

5.1 Contributions Summary

Regarding the study involving local features invariance, a general wrapper was presented, which wraps a detector/descriptor pair in order to increase viewpoint invariance by exploiting input depth maps. The proposed wrapper locates smooth surfaces within the input RGB-D images and projects them into a viewpoint invariant representation, enabling the detection and description of more viewpoint invariant features. The wrapper can be utilized with different combinations of descriptor/detector pairs, according to the desired application. Using synthetic and real-world objects, the viewpoint invariance of various detectors and descriptors was evaluated, for both *2D* and proposed *3D Affine* approaches. While 2D local image features fail to accommodate average viewpoint difference beyond 33.3° , the proposed wrapper boosted the viewpoint invariance to different levels, depending on the scene geometry. Objects with distinct surface discontinuities were on average invariant up to 52.8° , and the overall average for all evaluated datasets was 45.4° . Similarly, out

of a total of 140 combinations involving 20 local image features and various objects with distinct surface discontinuities, only a single 2D local image feature exceeded the goal of 60° viewpoint difference in just two combinations, as compared with 19 different local image features succeeding in 73 combinations when wrapped in the proposed wrapper . Furthermore, the proposed approach operates robustly in the presence of input depth noise, even that of low-cost commodity depth sensors, and well beyond.

As for rejection of outliers contaminating a putative-correspondences set, a highly accurate and efficient, yet simple, two-stage voting scheme was presented for distinguishing inlier correspondences by densely assessing and ranking their local and global geometric consistencies. At the local verification phase, a voting set is elected based on their spatial neighborhood rigidity invariance. In the global stage, the voting set is post-validated based on the global fitness of its geometric model hypotheses, and then, the putative-correspondences set is scored based on its elementwise covariance with the validated geometric hypotheses. The proposal strength stems from both the novel idea of post-validated voting set, as well as the single-point superimposition transforms that are ambiguity-free and computationally cheap. Using a well-known dataset consisting of various 3D models and numerous scenes that include different occlusion rates, the proposed scheme is evaluated against the state-of-the-art 3D voting schemes in terms of both the correspondence PR (precision-recall)-AUC (area under curve) and the execution time. Namely, a total of 374 experiments were conducted per each method, which involved the combination of 4 models, 50 scenes, and two down-samplings. The proposed scheme outperformed the state-of-the-art 3D voting schemes in both accuracy and speed aspects. Quantitatively, the proposed scheme scored $97.0\% \pm 12.9\%$ on the PR-AUC metric in average of all the experiments, while the two state-of-the-art schemes scored $74.2\% \pm 22.2\%$ and

78.3 % \pm 26.4 % respectively. Furthermore, the proposed scheme did not require more than 41.5 % \pm 12.5 % of the time consumed by the fastest state-of-the-art scheme. Likewise, the proposed voting scheme also demonstrated high robustness against occlusions and scarce inliers.

In terms of the multi-structure hypothesis generation, the rigid-body single-structure proposal was extended for multi-structure geometric fitting. It demonstrated a high level of accuracy, with around 69 % precision and recall for a total of 50 experiments with 750 generated hypotheses and inliers rate as low as 3.5 %. On the other hand, the remaining methods scored no higher than 12 % on both metrics, thus they were outperformed by the proposal. The proposed method also demonstrated high robustness against scarce inliers, as well as high effectiveness. The local rigidity constraint and the weighted superimposition transformations are the major players in robustness against scarce inliers, while the effectiveness of the proposal comes from the possibility to limit generated hypotheses to the first few ones, since they are generated in a progressive manner. Even without imposing such limits, the proposed method did not require more than 6.45 ms per each hypotheses, which is about 1.16 % of the time required by a sophisticated method, and about 268 % of the time required by random hypothesis generation, the most simple method. Overall, the proposed method exceeded all the compared methods in the balance between precision and execution time.

5.2 Future Work

5.2.1 3D Affine Framework

The proposed wrapper was unable to increase the viewpoint invariance for detectors of few keypoints or objects of complex geometries and indistinct surface-normal discontinuities. This limitation stems from the smooth-surface labeling phase, in which the spherical k -means utilizes only the surface normals to label different surfaces. To overcome this limitation, various possibilities are considered, such as including depth discontinuities, constraining the clusters by k -NN (k -nearest neighbor) pixel-based connectivity [99], imposing smoothness [98] constraints, or performing non-rigid registration [100] of the surface after the rigid alignment to the virtual image plane. Furthermore, there remains room for improvement in regard to the current hybrid rigid-homography method, which depends on the homography transforms, where index interpolation with iterative back-projection [104] represents another alternative and constitutes a point of future work. Furthermore, depth maps are currently interpolated for missing-depth small regions while neglecting larger ones; incorporating gradient-aware depth painting/interpolation constitutes a potential future direction.

Various applications, including 6D pose estimation, will be able to benefit from the proposed wrapper beyond what was demonstrated in Chapter 2, by using normal and gradient orientations of the region-affine keypoints along with their center points (refer to Section 2.3.2 for details). It is also believed that the current work can be utilized in non-rigid environments by replacing RANSAC (random sample consensus)-based matching with a global-motion modeling technique, such as factorized graph matching [106].

5.2.2 Correspondence Voting Scheme

The proposed voting scheme is currently limited to single rigid-body geometric fitting, for which addressing multi-structure geometric modeling would be an interesting extension of this proposal. Moreover, the proposed scheme picks the highest supported hypothesis without resampling it, for which resampling seems to robustify the estimation [89, 110, 111], and constitutes one of the future directions. Borrowing the concept of higher-than-minimal subset sampling [112, 113] into the voting schemes is also considered, perhaps by clustering correspondences [114] or poses [115]. Additionally, only the down-sampled point clouds are employed, for which involving the complete ones in the hypothesis election or fine-tuning is expected to bring about astonishing results.

5.2.3 Multi-structure Hypotheses Generation

As for the multi-structure geometric fitting proposed method, it is tailored specifically for multi-structure rigid-body geometric fitting, while the other compared methods are designed to address wider range of applications, for which generalizing the proposed method to 2D or higher-than-3D dimensions would be an interesting extension of this proposal. Moreover, the proposed method weights the neighborhood according to the estimated score, but does not resample it, for which resampling seems to robustify the estimation [89, 110, 111], and constitutes one of the future directions. Extending the method to perform hypothesis clustering and points segmentation is also considered to split the underlying multi-structure geometries.

5.3 Conclusion

3D reconstruction have been heavily investigated in the recent decades, and a lot of progress have been made in various aspects. A prevalent approach for 3D reconstruction is to compute some features for the scene RGB-D images, establish correspondence between them, and then estimate the relative poses between these RGB-D images to fit them together in a single 3D model. This research contributed in improving the viewpoint invariance of the features, the filtration of the correspondences, and the multi-structure modeling of the relative poses as well as the model segmentation from its environment. Each of these contributions has its own chapter, and proposed approaches are explained and dicussed in details in each corresponding chapter. Some contributions are believed to be novel since similar methods were not found in the literature. Whenever possible, proposed and existing methods were compared and evaluated to demonstrate proposed methods feasibility. Briefly, the local-features wrapper demonstrated high gains in viewpoint invariance, and the correspondence voting scheme produced very high-quality rankings for the likelihood of any particular correspondence being an inlier. Some preliminary results of multi-structure modeling and segmentation were also presented. This research was conducted in the hope it would contribute to the progress of science and/or technology advancement as well as beneficial outcomes for the society development.

Bibliography

- [1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *ISMAR*, (Basel, 2011 Oct. 26–29), vol. 11, IEEE, 2011 Oct., pp. 127–136, ISBN: 9781457721830. DOI: 10.1109/ismar.2011.6092378.
- [2] H. Sahloul, S. Shirafuji, and J. Ota, “3D affine: An embedding of local image features for viewpoint invariance using RGB-D sensor data,” *Sensors*, vol. 19, no. 2, pp. 291.1–191.32, 2019 Jan., ISSN: 1424-8220. DOI: 10.3390/s19020291.
- [3] —, “An accurate and efficient voting scheme for maximally all-inlier correspondences set,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (in revision)*, 2019 Apr.
- [4] —, “Multi-structure rigid-body geometric fitting in the presence of high outliers rate,” (*draft*), 2019 Jul.
- [5] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, “BigBIRD: A large-scale 3D database of object instances,” in *International Conference on Robotics and Automation*, (Hong Kong, China, 2014 May 31–Jun. 7), IEEE, 2014 May, pp. 509–516, ISBN: 9781479936854. DOI: 10.1109/icra.2014.6906903, Dataset available at <http://rll.berkeley.edu/bigbird/>.
- [6] Z. Teng and J. Xiao, “Surface-based detection and 6-dof pose estimation of 3-d objects in cluttered scenes,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1347–1361, 2016 Dec., ISSN: 1552-3098. DOI: 10.1109/tro.2016.2596799.
- [7] M. Okutomi and T. Kanade, “A multiple-baseline stereo,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 4, pp. 353–363, 1993 Apr., ISSN: 0162-8828. DOI: 10.1109/34.206955.

- [8] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Conference on Computer vision and pattern recognition*, (NY, USA), vol. 1, IEEE, 2006 Jul., pp. 519–528, ISBN: 0769525970. DOI: 10.1109/cvpr.2006.19.
- [9] C. Ttofis, S. Hadjitheophanous, A. S. Georgiades, and T. Theocharides, "Edge-directed hardware architecture for real-time disparity map computation," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 690–704, 2013 Apr., ISSN: 0018-9340. DOI: 10.1109/tc.2012.32.
- [10] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *International Conference on Computer Vision*, (Rio de Janeiro, Brazil, 2007 Oct. 14–21), IEEE, 2007, pp. 1–8, ISBN: 9781424416301. DOI: 10.1109/iccv.2007.4408933.
- [11] R. I. Hartley and A. Zisserman, *Multiple View Geometry In Computer Vision*, 2nd. Cambridge, UK: Cambridge University Press, 2004, ISBN: 9780511811685. DOI: 10.1017/cbo9780511811685.
- [12] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010 Aug., ISSN: 0162-8828. DOI: 10.1109/tpami.2009.161.
- [13] P. Beardsley, P. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," *European Conference on Computer Vision*, pp. 683–695, 1996, ISSN: 0302-9743. DOI: 10.1007/3-540-61123-1_181.
- [14] A. Fitzgibbon and A. Zisserman, "Automatic 3D model acquisition and generation of new images from video sequences," in *European Signal Processing Conference*, IEEE, 1998, pp. 1–8.
- [15] R. Koch, M. Pollefeys, and L. Van Gool, "Realistic surface reconstruction of 3D scenes from uncalibrated image sequences," *The Journal of Visualization and Computer Animation*, vol. 11, no. 3, pp. 115–127, 2000 Jul., ISSN: 1049-8907. DOI: 10.1002/1099-1778(200007)11:3<115::aid-vis228>3.0.co;2-2.
- [16] B. K. Horn and M. J. Brooks, *Shape From Shading*. Elsevier BV, 1990 May. DOI: 10.1016/0734-189x(90)90043-u.
- [17] E. Mingolla and J. T. Todd, "Perception of solid shape from shading," *Biological Cybernetics*, vol. 53, no. 3, pp. 137–151, 1986 Jan., ISSN: 0340-1200. DOI: 10.1007/bf00342882.

- [18] J. J. Gibson, "The perception of the visual world.," 1951 Oct., ISSN: 0031-8108. DOI: 10.2307/2181436.
- [19] J. T. Todd and R. A. Akerstrom, "Perception of three-dimensional form from patterns of optical texture.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 2, pp. 242–255, 1987, ISSN: 0096-1523. DOI: 10.1037//0096-1523.13.2.242.
- [20] S. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 824–831, 1994, ISSN: 0162-8828. DOI: 10.1109/34.308479.
- [21] G. Healey and T. O. Binford, "Local shape from specularity," *Computer Vision, Graphics, and Image Processing*, vol. 42, no. 1, pp. 62–86, 1988 Feb., ISSN: 0734-189X. DOI: 10.1016/0734-189x(88)90143-0.
- [22] P. Cavanagh and Y. G. Leclerc, "Shape from shadows.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 1, pp. 3–27, 1989, ISSN: 0096-1523. DOI: 10.1037//0096-1523.15.1.3.
- [23] M. Daum and G. Dudek, "On 3-d surface reconstruction using shape from shadows," in *Conference on Computer Vision and Pattern Recognition*, (Santa Barbara, CA, USA), IEEE Computer Society Press, 2002 Nov., ISBN: 0818684976. DOI: 10.1109/cvpr.1998.698646.
- [24] M. Potmesil, "Generating octree models of 3D objects from their silhouettes in a sequence of images," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 1, pp. 1–29, 1987 Mar., ISSN: 0734-189X. DOI: 10.1016/0734-189x(87)90053-3.
- [25] G. G. Gordon, "Shape from symmetry," in *Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, (Philadelphia, PA, USA), SPIE, 1990 Mar. DOI: 10.1117/12.969742.
- [26] B. Klingner, D. Martin, and J. Roseborough, "Street view motion-from-structure-from-motion," in *International Conference on Computer Vision*, (Sydney, Australia, 2013 Dec. 1–8), IEEE, 2013 Dec., ISBN: 9781479928408. DOI: 10.1109/iccv.2013.122.
- [27] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm, "Reconstructing the world* in six days," in *Conference on Computer Vision and Pattern Recognition*, (Boston, MA, USA, 2015 Jun. 7–12), IEEE, 2015 Jun., ISBN: 9781467369640. DOI: 10.1109/cvpr.2015.7298949.

- [28] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Autonomous Robot Vehicles*, Institute of Electrical and Electronics Engineers, 2005 Mar., pp. 167–193. DOI: 10.1007/978-1-4613-8997-2_14.
- [29] P. Moutarlier and R. Chatila, “Stochastic multisensory data fusion for mobile robot location and environment modeling,” in *International Symposium on Robotics Research*, (Tokyo, Japan), vol. 1, 1989.
- [30] H. J. S. Feder, J. J. Leonard, and C. M. Smith, “Adaptive mobile robot navigation and mapping,” *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999 Jul., ISSN: 0278-3649. DOI: 10.1177/02783649922066484.
- [31] M. Deans and M. Hebert, “Invariant filtering for simultaneous localization and mapping,” in *International Conference on Robotics and Automation*, (San Francisco, CA, USA), IEEE, 2002 Nov., ISBN: 0780358864. DOI: 10.1109/robot.2000.844737.
- [32] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *International Conference on Robotics and Automation*, (Kobe, Japan, 2009 May 12–17), IEEE, 2009 May, pp. 3212–3217, ISBN: 9781424427888. DOI: 10.1109/robot.2009.5152473.
- [33] A. S. Mian, M. Bennamoun, and R. Owens, “Three-dimensional model-based object recognition and segmentation in cluttered scenes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1584–1601, 2006 Oct., ISSN: 0162-8828. DOI: 10.1109/tpami.2006.213.
- [34] A. Mian, M. Bennamoun, and R. Owens, “On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes,” *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 348–361, 2010 Sep., ISSN: 0920-5691. DOI: 10.1007/s11263-009-0296-z.
- [35] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in *Conference on Computer Vision and Pattern Recognition*, (Honolulu, HI, 2017 Jul. 21–26), IEEE, 2017 Jul., pp. 4181–4190, ISBN: 9781538604571. DOI: 10.1109/cvpr.2017.302.
- [36] H. M. S. Sahloul, “Real-time slight 3D motion segmentation using an RGB-D camera,” Master’s thesis, Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-8656, Japan, 2016 Sep.

- [37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004 Nov., ISSN: 0920-5691. DOI: 10.1023/b:visi.0000029664.99615.94.
- [38] A. G. Buch, Y. Yang, N. Kruger, and H. G. Petersen, “In search of inliers: 3D correspondence by local and global voting,” in *Conference on Computer Vision and Pattern Recognition*, (Columbus, OH, USA, 2014 Jun. 23–28), IEEE, 2014 Jun., pp. 2067–2074, ISBN: 9781479951185. DOI: 10.1109/cvpr.2014.266.
- [39] J. Yang, Y. Xiao, Z. Cao, and W. Yang, “Ranking 3D feature correspondences via consistency voting,” *Pattern Recognition Letters*, vol. 117, pp. 1–8, 2019 Jan., ISSN: 0167-8655. DOI: 10.1016/j.patrec.2018.11.018.
- [40] C. Wu, B. Clipp, X. Li, J. Frahm, and M. Pollefeys, “3D model matching with viewpoint-invariant patches (VIP),” in *Conference on Computer Vision and Pattern Recognition*, (Anchorage, AK, USA, 2008 Jun. 23–28), IEEE, 2008 Jun., pp. 1–8, ISBN: 9781424422425. DOI: 10.1109/cvpr.2008.4587501.
- [41] M. Karpushin, G. Valenzise, and F. Dufaux, “Local visual features extraction from texture+depth content based on depth image analysis,” in *International Conference on Image Processing*, (Paris, France, 2014 Oct. 27–30), IEEE, 2014 Oct., pp. 2809–2813, ISBN: 9781479957514. DOI: 10.1109/icip.2014.7025568.
- [42] —, “Improving distinctiveness of BRISK features using depth maps,” in *International Conference on Image Processing*, (Quebec City, QC, Canada, 2015 Sep. 27–30), IEEE, 2015 Sep., pp. 2399–2403, ISBN: 9781479983391. DOI: 10.1109/icip.2015.7351232.
- [43] Y. Liu, H. Zhang, H. Guo, and N. Xiong, “A fast-brisk feature detector with depth information,” *Sensors*, vol. 18, no. 11, pp. 3908.1–3908.19, 2018 Nov., ISSN: 1424-8220. DOI: 10.3390/s18113908.
- [44] K. Simonyan, A. Vedaldi, and A. Zisserman, “Learning local feature descriptors using convex optimisation,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1573–1585, 2014 Aug., ISSN: 0162-8828. DOI: 10.1109/tpami.2014.2301163.
- [45] T. Trzcinski, M. Christoudias, and V. Lepetit, “Learning image descriptors with boosting,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 597–610, 2015 Mar., ISSN: 0162-8828. DOI: 10.1109/tpami.2014.2343961.

- [46] K. M. Yi, Y. Verdie, P. Fua, and V. Lepetit, “Learning to assign orientations to feature points,” in *Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA, 2016 Jun. 27–30), IEEE, 2016 Jun., pp. 107–116, ISBN: 9781467388511. DOI: 10.1109/cvpr.2016.19.
- [47] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, “LIFT: Learned invariant feature transform,” in *European Conference on Computer Vision*, (Amsterdam, The Netherlands, 2016 Oct. 8–16), Cham, Switzerland: Springer International Publishing, 2016, pp. 467–483, ISBN: 9783319464657. DOI: 10.1007/978-3-319-46466-4_28.
- [48] B. Kumar, G. Carneiro, and I. Reid, “Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions,” in *Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA, 2016 Jun. 27–30), IEEE, 2016 Jun., pp. 5385–5394, ISBN: 9781467388511. DOI: 10.1109/cvpr.2016.581.
- [49] H. Altwaijry, E. Trulls, J. Hays, P. Fua, and S. Belongie, “Learning to match aerial images with deep attentive architectures,” in *Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA, 2016 Jun. 27–30), IEEE, 2016 Jun., pp. 3539–3547, ISBN: 9781467388511. DOI: 10.1109/cvpr.2016.385.
- [50] Y. Tian, B. Fan, and F. Wu, “L2-net: Deep learning of discriminative patch descriptor in euclidean space,” in *Conference on Computer Vision and Pattern Recognition*, (Honolulu, HI, USA, 2017 Jul. 21–26), vol. 1, IEEE, 2017 Jul., pp. 661–669, ISBN: 9781538604571. DOI: 10.1109/cvpr.2017.649.
- [51] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, “SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again,” in *International Conference on Computer Vision*, (Venice, Italy, 2017 Oct. 22–29), IEEE, 2017 Oct., pp. 22–29, ISBN: 9781538610329. DOI: 10.1109/iccv.2017.169.
- [52] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes,” in *Robotics: Science and Systems*, (Pittsburgh, PA, USA, 2018 Jun. 26–30), Robotics: Science and Systems Foundation, 2018 Jun., ISBN: 9780992374747. DOI: 10.15607/rss.2018.xiv.019.

- [53] B. Tekin, S. N. Sinha, and P. Fua, “Real-time seamless single shot 6D object pose prediction,” in *Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT, USA, 2018 Jun. 18–23), IEEE, 2018 Jun., pp. 292–301, ISBN: 9781538664209. DOI: 10.1109/cvpr.2018.00038.
- [54] T.-T. Do, M. Cai, T. Pham, and I. Reid, “Deep-6DPose: Recovering 6D object pose from a single RGB image,” *CoRR*, vol. abs/1802.10367, 2018. arXiv: 1802.10367.
- [55] C. Li, J. Bai, and G. D. Hager, “A unified framework for multi-view multi-class object pose estimation,” in *European Conference on Computer Vision*, (Munich, Germany, 2018 Sep. 8–14), Cham, Switzerland: Springer International Publishing, 2018, pp. 263–281, ISBN: 9783030012694. DOI: 10.1007/978-3-030-01270-0_16.
- [56] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, “Comparative evaluation of hand-crafted and learned local features,” in *Conference on Computer Vision and Pattern Recognition*, (Honolulu, HI, USA, 2017 Jul. 21–26), IEEE, 2017 Jul., pp. 6959–6968, ISBN: 9781538604571. DOI: 10.1109/cvpr.2017.736.
- [57] J. Shi and C. Tomasi, “Good features to track,” in *Conference on Computer Vision and Pattern Recognition*, (Seattle, WA, USA, 1993 Jun. 15–17), IEEE Computer Society Press, 1994, pp. 593–600, ISBN: 0818658258. DOI: 10.1109/cvpr.1994.323794.
- [58] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008 Jun., ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014.
- [59] P. Moreels and P. Perona, “Evaluation of features detectors and descriptors based on 3D objects,” *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007 Jul., ISSN: 0920-5691. DOI: 10.1007/s11263-006-9967-1.
- [60] A. Vedaldi and S. Soatto, “Features for recognition: Viewpoint invariance for non-planar scenes,” in *International Conference on Computer Vision*, (Beijing, China, 2005 Oct. 17–21), vol. 2, IEEE, 2005, pp. 1474–1481, ISBN: 076952334X. DOI: 10.1109/iccv.2005.99.
- [61] N. Khan, B. McCane, and S. Mills, “Better than SIFT?” *Machine Vision and Applications*, vol. 26, no. 6, pp. 819–836, 2015 Aug., ISSN: 0932-8092. DOI: 10.1007/s00138-015-0689-7.

- [62] G. Hua, M. Brown, and S. Winder, “Discriminant embedding for local image descriptors,” in *International Conference on Computer Vision*, (Rio de Janeiro, Brazil, 2007 Oct. 14–21), IEEE, 2007, pp. 1–8, ISBN: 9781424416301. DOI: 10.1109/iccv.2007.4408857.
- [63] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” in *Conference on Computer Vision and Pattern Recognition*, (Washington, DC, USA, 2004 Jun. 27–Jul. 2), vol. 2, IEEE, 2004 Nov., pp. 506–513, ISBN: 0769521584. DOI: 10.1109/cvpr.2004.1315206.
- [64] J.-M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009 Jan., ISSN: 1936-4954. DOI: 10.1137/080732730.
- [65] Y. Pang, W. Li, Y. Yuan, and J. Pan, “Fully affine invariant SURF for image matching,” *Neurocomputing*, vol. 85, pp. 6–10, 2012 May, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2011.12.006.
- [66] M. Agrawal, K. Konolige, and M. R. Blas, “CenSurE: Center surround extremas for realtime feature detection and matching,” in *European Conference on Computer Vision*, (Marseille, France, 2008 Oct. 12–18), Berlin, Germany: Springer, 2008 Oct., pp. 102–115, ISBN: 9783540886921. DOI: 10.1007/978-3-540-88693-8_8.
- [67] D. Nistér and H. Stewénus, “Linear time maximally stable extremal regions,” in *European Conference on Computer Vision*, (Marseille, France, 2008 Oct. 12–18), Berlin, Germany: Springer, 2008, pp. 183–196, ISBN: 9783540886853. DOI: 10.1007/978-3-540-88688-4_14.
- [68] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010 Jan., ISSN: 0162-8828. DOI: 10.1109/tpami.2008.275.
- [69] E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide-baseline stereo,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010 May, ISSN: 0162-8828. DOI: 10.1109/tpami.2009.77.
- [70] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *European Conference on Computer Vision*, (Heraklion, Crete, Greece, 2010 Sep. 5–11), Berlin, Germany: Springer, 2010, pp. 778–792, ISBN: 9783642155604. DOI: 10.1007/978-3-642-15561-1_56.

- [71] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in *European Conference on Computer Vision*, (Heraklion, Crete, Greece, 2010 Sep. 5–11), Berlin, Germany: Springer, 2010, pp. 183–196, ISBN: 9783642155512. DOI: 10.1007/978-3-642-15552-9_14.
- [72] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *International Conference on Computer Vision*, (Barcelona, Spain, 2011 Nov. 6–13), IEEE, 2011 Nov., pp. 2564–2571, ISBN: 9781457711015. DOI: 10.1109/iccv.2011.6126544.
- [73] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *International Conference on Computer Vision*, (Barcelona, Spain, 2011 Nov. 6–13), IEEE, 2011 Nov., pp. 2548–2555, ISBN: 9781457711015. DOI: 10.1109/iccv.2011.6126542.
- [74] R. Arandjelovi and A. Zisserman, “Three things everyone should know to improve object retrieval,” in *Conference on Computer Vision and Pattern Recognition*, (Providence, RI, USA, 2012 Jun. 16–21), IEEE, 2012 Jun., pp. 2911–2918, ISBN: 9781467312264. DOI: 10.1109/cvpr.2012.6248018.
- [75] A. Alahi, R. Ortiz, and P. Vanderghenst, “FREAK: Fast retina keypoint,” in *Conference on Computer Vision and Pattern Recognition*, (Providence, RI, USA, 2012 Jun. 16–21), IEEE, 2012 Jun., pp. 510–517, ISBN: 9781467312264. DOI: 10.1109/cvpr.2012.6247715.
- [76] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” in *British Machine Vision Conference*, (Bristol, UK, 2013 Oct. 9–13), Surrey, UK: British Machine Vision Association, 2013, pp. 13.1–13.11, ISBN: 1901725499. DOI: 10.5244/c.27.13.
- [77] G. Levi and T. Hassner, “LATCH: Learned arrangements of three patch codes,” in *Winter Conference on Applications of Computer Vision*, (Lake Placid, NY, USA, 2016 Mar. 7–10), IEEE, 2016 Mar., pp. 1–9, ISBN: 9781509006410. DOI: 10.1109/wacv.2016.7477723.
- [78] Z. Fu, Q. Qin, B. Luo, C. Wu, and H. Sun, “A local feature descriptor based on combination of structure and texture information for multispectral image matching,” *Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 100–104, 2019 Jan., ISSN: 1545-598X. DOI: 10.1109/lgrs.2018.2867635.
- [79] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von mises-fisher distributions,” *Journal of Machine Learning Research*, vol. 6, pp. 1345–1382, 2005.

- [80] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *International Conference on Computer Vision*, (Bombay, India, 1998 Jan. 4–7), Narosa Publishing House, 2002 Nov., pp. 839–846, ISBN: 8173192219. DOI: 10.1109/iccv.1998.710815.
- [81] T. Kurita and P. Boulanger, “Computation of surface curvature from range images using geometrically intrinsic weights,” in *Workshop on Machine Vision Applications*, (Tokyo, Japan, 1992 Dec. 7–9), IAPR, 1992, pp. 389–392.
- [82] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Berkeley Symposium on Mathematical Statistics and Probability*, (Oakland, CA, USA, 1965 Dec. 27–1966 Jan. 7), vol. 1, Berkeley, CA, USA: University of California Press, 1967, pp. 281–297.
- [83] T. Caliski and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics - Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974, ISSN: 0361-0926. DOI: 10.1080/03610927408827101.
- [84] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction To Robotic Manipulation*, 1st. Boca Raton, FL, USA: CRC Press, 2017 Dec., ISBN: 9781315136370. DOI: 10.1201/9781315136370.
- [85] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaf-falitzky, T. Kadir, and L. J. V. Gool, “A comparison of affine region detec-tors,” *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005 Nov., ISSN: 0920-5691. DOI: 10.1007/s11263-005-3848-x.
- [86] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with auto-matic algorithm configuration,” in *International Conference on Computer Vi-sion Theory and Applications*, (Lisboa, Portugal, 2009 Feb. 5–8), SciTePress - Science, 2009, pp. 331–340, ISBN: 9789898111692. DOI: 10.5220/0001787803310340.
- [87] W. Kabsch, “A discussion of the solution for the best rotation to relate two sets of vectors,” *Acta Crystallographica Section A*, vol. 34, no. 5, pp. 827–828, 1978 Sep., ISSN: 0567-7394. DOI: 10.1107/s0567739478001680.
- [88] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartogra-phy,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981 Jun., ISSN: 0001-0782. DOI: 10.1145/358669.358692.
- [89] A. Hast, J. Nysjö, and A. Marchetti, “Optimal RANSAC—towards a repeat-able algorithm for finding the optimal set,” *Journal of World Society for Com-puter Graphics*, vol. 21, no. 1, pp. 21–30, 2013.

- [90] Python Software Foundation, *Python language reference, version 3.6*. [Online]. Available: <https://www.python.org/> (visited on 2018 Dec. 17).
- [91] T. E. Oliphant, *Guide To NumPy*, 2nd. CreateSpace Independent Publishing Platform: North Charleston, SC, USA, 2015, ISBN: 9781517300074.
- [92] G. R. Bradski, “The OpenCV library,” *Dr. Dobb’s Journal of Software Tools*, vol. 25, no. 11, pp. 120, 122–125, 2000 Nov.
- [93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [94] E. Jones, T. E. Oliphant, P. Peterson, *et al.*, *SciPy: Open source scientific tools for Python*. [Online]. Available: <https://www.scipy.org/> (visited on 2018 Dec. 17).
- [95] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007 May, ISSN: 1521-9615. DOI: 10.1109/mcse.2007.55.
- [96] P. Ramachandran and G. Varoquaux, “MayaVI: 3D visualization of scientific data,” *Computing in Science & Engineering*, vol. 13, no. 2, pp. 40–51, 2011 Mar., ISSN: 1521-9615. DOI: 10.1109/mcse.2011.35.
- [97] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005 Oct., ISSN: 0162-8828. DOI: 10.1109/tpami.2005.188.
- [98] T. Rabbani, F. V. D. Heuvel, and G. Vosselmann, “Segmentation of point clouds using smoothness constraint,” in *ISPRS Commission V Symposium Volume 36, Part 5: Image Engineering and Vision Metrology*, vol. 36, Dresden, Germany: International Society for Photogrammetry and Remote Sensing, 2006 Sep., pp. 248–253.
- [99] M. Brito, E. Chavez, A. Quiroz, and J. Yukich, “Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection,” *Statistics & Probability Letters*, vol. 35, no. 1, pp. 33–42, 1997 Aug., ISSN: 0167-7152. DOI: 10.1016/s0167-7152(96)00213-1.

- [100] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010 Dec., ISSN: 0162-8828. DOI: 10.1109/tpami.2010.46.
- [101] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012 Feb., ISSN: 1424-8220. DOI: 10.3390/s120201437.
- [102] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, (Zurich, Switzerland, 2012 Oct. 13–15), IEEE, 2012 Oct., pp. 524–530, ISBN: 9781467344708. DOI: 10.1109/3dimpvt.2012.84.
- [103] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *International Conference on Computer Vision*, (Barcelona, Spain, 2011 Nov. 6–13), IEEE, 2011 Nov., pp. 2320–2327, ISBN: 9781457711015. DOI: 10.1109/iccv.2011.6126513.
- [104] G. Chaurasia, S. Duchêne, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Transactions on Graphics*, vol. 32, no. 3, 30:1–30:12, 2013 Jun., ISSN: 0730-0301. DOI: 10.1145/2487228.2487238.
- [105] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *ACM SIGGRAPH 2005 Courses*, (Los Angeles, California, 2005 Jul. 31–Aug. 4), New York, New York, USA: ACM Press, 2005, pp. 1–11. DOI: 10.1145/1198555.1198565.
- [106] F. Zhou and F. D. la Torre, "Factorized graph matching," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774–1789, 2016 Sep., ISSN: 0162-8828. DOI: 10.1109/tpami.2015.2501802.
- [107] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3D data," *Image and Vision Computing*, vol. 17, no. 2, pp. 135–147, 1999 Feb., ISSN: 0262-8856. DOI: 10.1016/s0262-8856(98)00117-6.
- [108] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR04-004, 2004 Feb.
- [109] M. Ruhnke, R. Kümmerle, G. Grisetti, and W. Burgard, "Highly accurate 3D surface models by sparse surface adjustment," in *International Conference on Robotics and Automation*, (St Paul, MN, USA, 2012 May 14–18), IEEE, 2012 May, pp. 751–757, ISBN: 9781467314039. DOI: 10.1109/icra.2012.6225077.

- [110] O. Chum, J. Matas, and J. Kittler, “Locally optimized RANSAC,” in *Joint Pattern Recognition Symposium*, Berlin, Heidelberg: Springer, 2003, pp. 236–243, ISBN: 9783540408611. DOI: 10.1007/978-3-540-45243-0_31.
- [111] O. Chum, J. Matas, and S. Obdrzalek, “Enhancing RANSAC by generalized model optimization,” in *Asian Conference on Computer Vision*, vol. 2, 2004, pp. 812–817.
- [112] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, “The random cluster model for robust geometric fitting,” *transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1658–1671, 2014 Aug., ISSN: 0162-8828. DOI: 10.1109/tpami.2013.2296310.
- [113] R. B. Tennakoon, A. Bab-Hadiashar, Z. Cao, R. Hoseinnezhad, and D. Suter, “Robust model fitting using higher than minimal subset sampling,” *transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 350–362, 2016 Feb., ISSN: 0162-8828. DOI: 10.1109/tpami.2015.2448103.
- [114] A. E. Johnson and M. Hebert, “Surface matching for object recognition in complex three-dimensional scenes,” *Image and Vision Computing*, vol. 16, no. 9-10, pp. 635–651, 1998 Jul., ISSN: 0262-8856. DOI: 10.1016/s0262-8856(98)00074-2.
- [115] B. Drost and S. Ilic, “3D object detection and localization using multimodal point pair features,” in *International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, (Zurich, Switzerland, 2012 Oct. 13–15), IEEE, 2012 Oct., pp. 9–16, ISBN: 9781467344708. DOI: 10.1109/3dimpvt.2012.53.

Glossary

Notation	Description	Symbol	Page List
6D	a short form of 6 DoFs (degrees of freedom)	\mathbb{SE}_3	33, 34, 37, 41, 45, 56–58, 65, 83, 92, 100
ambiguity-free	does not suffer from the eigenvectors ambiguity of sign		27, 98
BOOST	named after a utilized machine learning concept; the boosting		62, 78, 79, 81
contaminated	impure; contains outliers		xv, 29
DAISY	named after a flower resembling its sampling window		62, 80, 81
inlier	valid; correct; the opposite of outlier		xv, xvi, 20, 27, 58, 68, 70, 71, 84, 98, 99, 102, 120
KAZE	named after the “wind” in Japanese; an analogy to nonlinear diffusion		62, 121

Notation	Description	Symbol	Page List
outlier	spurious; incorrect		xv, 20, 27, 29, 31, 46–49, 56, 68, 70, 84, 98, 119, 120
outlier rejection	the process of excluding outlier elements from a set		xv, 29, 31
post-validated	validated at a later stage to minimize its outliers elements		98
putative	unverified; mostly a mixture of inliers and outliers		xv, 28, 29, 98
viewpoint difference	the rotation angle made by the camera when moving from one viewpoint to the other		xiv, xv, 16, 19–21, 25–27, 33, 34, 53, 59, 63, 64, 66, 67, 70, 72, 89, 90, 97, 98, 120
viewpoint invariance	the maximal tolerated viewpoint difference by features detection and description algorithms		xv, 21, 25–27, 31–36, 58–61, 64–68, 72–79, 82, 87–91, 97, 100, 102
voting scheme	a system that collects support for some hypotheses set based on its elementwise invariant or covariant consistency with another set		xvi, 27, 31, 32, 98, 101, 102

Acronyms

Notation	Description	Symbol	Page List
1-1	one-to-one		xiv, 15, 16, 19
1-M	one-to-many		xiv, xv, 15, 16, 18
1PST	single-point superimposition transform		27, 98
AGAST	adaptive & generic detection based on accelerated segment test		62, 79, 80
AKAZE	accelerated KAZE		62, 79, 80
ASIFT	2D affine SIFT (scale-invariant feature transform)		35
ASURF	2D affine SURF (speeded up robust features)		35
AUC	area under curve		xvi, 98
BA	bundle adjustment		10, 122
BRIEF	binary robust independent elementary features		62, 79, 81, 123
BRISK	binary robust invariant scalable keypoints		34, 62, 76, 78–81
CAD	computer aided design		1
CenSurE	center surround extremas detector		62, 76, 78–80
CLM	concurrent localization and mapping		12

Notation	Description	Symbol	Page List
DLCO	descriptor learning using convex optimization		62, 78, 80, 81, 83
DoF	degree of freedom		119
DTAM	dense tracking and mapping		89, 92
EXIF	exchangeable image format		9
FAST	features from accelerated segment test		62, 79, 80, 83, 123
FPFH	fast point feature histograms		15, 28
FREAK	fast retina keypoint		62, 80, 81
GFTT	good features to track		33, 62, 79, 80, 82
ICP	iterative closest/corresponding point		92
k -NN	k -nearest neighbor		56, 70, 71, 84, 91, 100
LATCH	learned arrangements of three patch codes		62, 80, 81, 83
LRC	local rigidity constraint		27, 99
M-M	many-to-many		xiv, xv, 15–17
MSER	maximally stable extremal regions		62, 76, 78–80, 83
MVS	multi-view stereo		xiii, xiv, 4, 6–10, 20
ORB	oriented FAST (features from accelerated segment test) and rotated BRIEF (binary robust independent elementary features)		62, 76, 78–81
PR	precision-recall		xvi, 98

Notation	Description	Symbol	Page List
RANSAC	random sample consensus		56, 58, 59, 70, 71, 92, 100
RFID	radio-frequency identifier		3
RGB-D	RGB trichromatic color image and per-pixel depth map		14, 24, 33, 38, 42, 44, 47, 49–52, 56, 59, 61, 63, 67, 88, 97
RMS	root-mean-square		65
RootSIFT	root-normalized SIFT descriptor		62, 78, 80, 81, 83
SDF	signed distance function		124
SfM	structure from motion		xiii, xiv, 3, 4, 6, 7, 9–13, 15, 20, 29
SIFT	scale-invariant feature transform		27, 33–35, 59, 61–63, 67, 69, 71, 73–76, 78–83, 90, 121, 123
SLAM	simultaneous localization and mapping		xiii, 3, 9, 12, 14, 15, 34, 124
SNR	signal-to-noise ratio		63, 64, 85, 88, 91, 92
SURF	speeded up robust features		33, 35, 59, 61–63, 67, 69, 71, 74–83, 90, 121
SVD	singular value decomposition		58
TOF	time of flight		xiii
U3OR	UWA 3D object recognition		17
VIP	viewpoint invariant patch		34, 37, 39
VSLAM	visual SLAM (simultaneous localization and mapping)		xiii, xiv, 3, 4, 6, 12, 14, 20