DISSERTATION

A hypothesis verification framework for 3D object recognition in clutter

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
E376
Institut für Automatisierungs- und Regelungstechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Dipl.-Ing. Aitor Aldoma
geb. am 19.03.1985
Matr. Nr.: 0926483

Wien, im Juni 2014

Aitor Aldoma

Abstract

The ability to autonomously recognize objects and estimate their pose is a key component of robotic agents. It enables autonomous systems to interact and understand their environment and, because of the important role of objects within our society and industries, it increases their usefulness. The necessity of such components combined with the difficult and interesting challenges posed by this problem, has caused object recognition to be an important field of research in recent years.

Despite of recent advances – in terms of sensing technologies and algorithms – the object recognition problem is considered in general unsolved. In particular, being able to recognize a larger number of partially occluded objects in complex scenes populated with clutter has happened to be specially challenging. Moreover, the different recognition relevant properties of objects (i.e., texture or texture-less, geometrically unique or common) as well as several artefacts associated with current sensing capabilities (i.e, noisy or missing data) pose additional problems.

Aiming at increasing the recognition capabilities of autonomous systems, this thesis investigates and proposes improvements related to different object recognition paradigms. Because the different paradigms present unique characteristics that make them suitable for different scene configurations and/or different types of objects, we argue that the parallel deployment of multiple paradigms broadens the range of situations on which a recognition system can be successfully applied. While this results in more objects being correctly recognized (together with their pose in the scene), it inevitably incurs in the generation of wrong object hypotheses.

In order to maximize the positive effects of multiple recognition pipelines (correct recognitions) while minimizing their negative effects (wrong hypotheses), this thesis proposes a novel hypotheses verification stage. The goal of this stage is simple: reject wrong hypotheses while preserving correct ones, effectively increasing the operating point of the overall system. This is achieved by selecting a subset of object hypotheses which best represents the scene under consideration. A unique trait of the proposed verification stage is that all hypotheses are simultaneously considered, instead of one at a time, which results in a global model of the scene. We formalize this stage as the minimization (over the object hypotheses) of a global cost function enforcing geometrical and appearance cues as well as physical constraints.

We show how the proposed recognition system results in excellent performance on six different benchmark datasets presenting heterogeneous recognition scenarios (in terms of sensory data, scene configurations and type of objects). The proposed framework currently candidates itself as the first algorithm being able to outperform the state of the art on such a vast and diverse set of benchmark datasets.

This manuscript is best to read in colour.

Acknowledgement

This work would not have been possible without the continued support from my family, friends and colleagues and most notably my supervisor Markus Vincze for his encouragement and guidance during this process as well as for permitting me to pursue my own ideas. I also want to thank my external reviewer Luigi Di Stefano for his time and interesting collaborations during these years.

My special gratitude goes to my mentors, colleagues and friends Federico Tombari, Johann Prankl and Radu Bogdan Rusu for their time, patience and inspiring ideas. The endless discussions and collaborations during this period have been of great value for the development of this thesis. In addition, many thanks to Radu Bogdan Rusu, Markus Vincze and Willow Garage for making my stay there possible. An inspiring and enriching period which directed my research in the areas covered throughout this thesis.

Not least I have to thank the general public for funding this work via various EU and nationally funded projects. In addition, I would like to thank all the individuals developing excellent Open Source software that has greatly simplified the development of this thesis. In particular, many thanks to the community involved in the development of the Point Cloud Library.

Finally I want to express my love and gratitude to Nicole for her encouragement, patience and lovely companion during these years. I also want to thank my beloved parents, Ramon and Pepita, for their unconditional support and guidance which set the foundations for my scientific career.

Contents

Chapter 1

Introduction

Objects and their associated functionalities are essential components of human and industrial processes. We use them constantly in our daily activities when solving different tasks or simply consume them to increase our comfort. Objects are so important that many industrial processes are related to their production and distribution. As robotic agents make their way into our homes and industries to assist or replace humans, they will be required to understand the environment as well as the objects therein. In other words, robots have to be able to search, recognize, sort out and manipulate objects, among other things falling outside the scope of this thesis. Because robots will coexist with humans, we would like robots to perform such tasks with minor adaptations to the environment and the objects we already have. Such expectation explains why robots have already been successfully integrated into industrial processes (where the environment and objects can be adjusted to simplify the robot task) while their presence in domestic environments is rare as private users are not willing to drastically change their environment[1].

Fortunately, within the field of robotics, the complexity of object recognition and scene understanding might be reduced by exploiting the fact that the robotic instance is confined to a certain environment — home, office, building, etc. — at a given time and sometimes even constrained by a few tasks we would like the robot to help us with. Such fact allowing us to teach precisely the meaning of *objects* compared to the understanding of *object recognition* in other fields such as computer vision where the amount of objects is usually much larger and constraints such as locality and temporality are rarely exploited. For instance, a recognition system on a robotic agent is probably not interested in learning visual and shape properties of computer monitors produced during the late 80s when solving tasks in 2014; coming across such instances renders improbable. Additionally, robots are usually equipped with recent sensors able to provide color images together with depth information of the scene which has been proved to enhance object recognition [26, 60, 5]. This results in the robotic agent being able to create models (i.e, in form of 3D models, see Chapter 2)

[1]Without considering other factors such as economical costs and current usefulness of autonomous systems.

of the specific object instances which are to be recognized.

Even after such simplifications and the adoption of recent sensing devices, the problem of object recognition remains challenging and is in general considered unsolved. We still expect such systems to be able to handle dozens of different objects in a reasonable amount of time. Additionally, objects might present different properties:

- **Visual appearance**: Textured or texture-less objects
- **Size**: Small or big objects
- Different **materials**
- Distinctive or uniform **shapes**
- **Rigid**, **deformable** or **articulated** objects[2]

and might be **occluded**, **cluttered**, poorly or too **illuminated** and present several nuisances coming from the sensing capabilities of the agent (**noise**, **missing data**, etc.). Other factors to consider are **similarities between objects** that might be difficult to distinguish, the requirement of **accurate poses** for autonomous object manipulation or **processing time** constraints.

These factors pose several challenges for the design of a recognition system to be deployed in situations where the object properties and environmental conditions cannot be fully controlled. Unfortunately, such issues have been largely ignored in the literature which presents several methods that successfully address a few of these factors but are unable to handle the rest. Indeed, methods such as [24] exploiting local textured patches are able to handle textured objects and occlusions but fail under the absence of texture [19]. Other methods devised to detect untextured objects [26, 6] struggle to handle occlusions and require large amounts of training data which poses scalability problems. Several recognition paradigms as well as specific methods are discussed later on in Chapter 3 focusing on their strengths and weaknesses. To some extend, proposed methods are mutually exclusive when considering the aforementioned factors being addressed by each method. Taking into account the vast amount of literature concerned with object recognition, addressing the problem in a generic way seems to be very difficult, specially when aiming at high and accurate recognition rates.

Nevertheless, the aforementioned mutual exclusivity and different strengths provided by particular recognition methods can be exploited by letting different recognition instances generate object hypotheses (i.e., the object id and its pose in the scene) in parallel. Even though this parallel deployment of individual recognition methods effectively broadens the range of situations on which the recognition system can be successfully applied, it inevitably increases the amount of object hypotheses being generated. In addition, some of these hypotheses might represent incorrect

[2]Only the recognition of rigid objects is considered within this work.

(i.e., false positives) or duplicate detections[3] that need to be posteriorly verified in order to provide a solution as consistent as possible with the scene being analysed.

The problems associated with the presence of wrong and duplicate detections becomes specially evident when individual recognition methods are configured in a *loose* way. By *loose* we understand that the parameters do not restrict the generation of object hypotheses if the evidence or consensus gathered by the recognition method about the object being present in the scene is weak or low. In order to exemplify this, consider methods based on point-to-point correspondences between the scene and a particular model of an object we would like to recognize. Theoretically, only three such correspondences are required to estimate the pose of an object in a scene, however, it is usually possible to configure methods based on point-to-point correspondences to require a higher consensus in order to hypothesize about objects. While this reduces the amount of hypotheses being generated, it also hinders the generation of hypotheses associated with occluded objects and thus prevents their detection.

Aiming at mitigating the undesired effect caused by the presence of false responses while maintaining the benefits of multiple recognition pipelines (each of them potentially being loosely configured to handle borderline situations, i.e., highly occluded or cluttered objects), recognition systems are equipped with a final *hypothesis verification* stage. The goal of this stage is to analyse the generated hypotheses in order to discard those that are wrong while maintaining correct ones. A successful verification stage has an important repercussion in the overall performance of the system. In particular, being able to reject false positives causes a positive increase in the precision of the system while the acceptance of correct hypotheses maintains recall high, all in all improving the operation point of the system.

1.1 Proposed recognition framework

With a long-term goal of enabling robotic agents to robustly recognize objects in environments such as offices, homes or industries, this thesis investigates the problem of object instance recognition and 6DoF (6 Degrees of Freedom, i.e., 3D rotation and translation) pose estimation. Such environments might be populated by objects presenting different recognition relevant traits (textured or texture-less, distinctive or uniform shapes), might be partially occluded from the current vantage point of the robot and might be surrounded by elements unknown to the recognition system (i.e. clutter). Because these factors cannot be always controlled, the proposed recognition framework needs to be general enough to handle these situations. In addition, the robot needs to be equipped with certain mechanisms enabling it to learn the properties of the objects to be recognized.

Taking the aforementioned requirements into account, the deployment of the proposed recognition framework is divided into two phases. The first one is responsible for the acquisition of knowledge about the objects of interest (i.e., those objects that

[3]Please note that is still true when a single recognition method is used.

the system needs to recognize) and is commonly executed in an off-line manner prior to the deployment of the robot. In particular and within the scope of this thesis, a human tutor places the objects to be learned in front of the sensor of the robot. By presenting it from different perspectives, the robot is able to reconstruct a 3D model of the object and in addition, learns different properties of the object that can be used to recognize it later on. If models of the objects are already available (i.e., provided by their manufacturer) this stage can be skipped, easing the deployment of the robot.

After the robot has gathered enough knowledge about the objects that it needs to recognize, the robot is ready for deployment. This second part of the recognition framework is executed on-line (i.e., the robot moving around the environment) and involves (i) sensing the current environment of the robot, (ii) generation of object hypotheses in the provided scene and (iii) verification of object hypotheses yielding a consistent solution. Depending on the sensing capabilities of the agent, different data modalities are available (e.g., 3D or RGB-D data).

In order to exploit the different strengths of different recognition methods as well as the multi-modal representation of the scene, the recognition system deploys in parallel multiple recognition pipelines. The hypotheses generated by the different methods are merged into a hypothesis verification module as depicted in Figure 1.1, responsible for the rejection of wrong or duplicate hypotheses in order to provide a consistent recognition result.

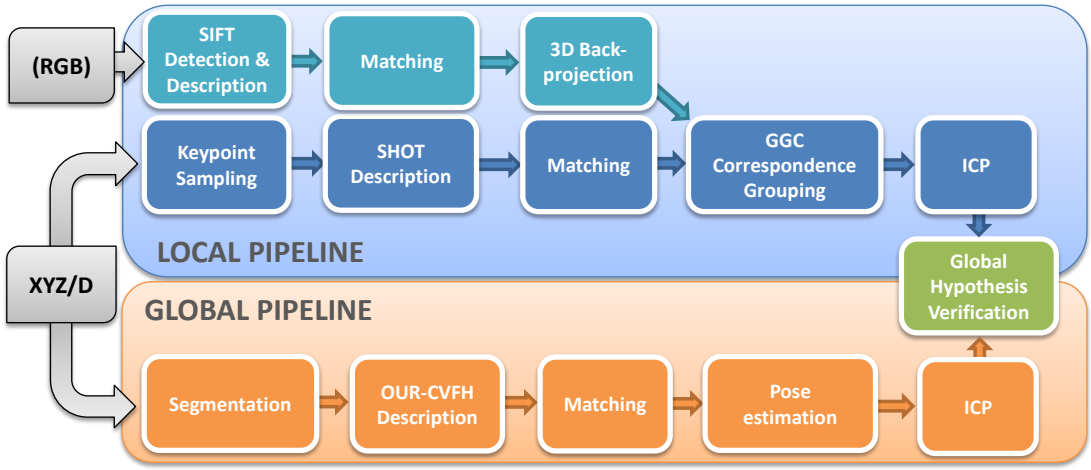


Figure 1.1: Proposed recognition system exploiting different data modalities provided by current RGB-D sensor as well as the different strengths characterizing different recognition paradigms (local, global). The hypotheses generated by the multiple pipelines are finally merged into a hypothesis verification stage aiming at providing a solution (in terms of available hypotheses) consistent with the scene under analysis.

In more detail, the generation of object hypotheses is based on three different recognition pipelines belonging to the *local* and *global* recognition paradigms. In a nutshell, the local paradigm establishes point-to-point correspondences between the

object models and the scene by matching common *local features*. As Figure 1.1 depicts, the local pipeline exploits the multi-modality of the data by using SIFT [39] (2D features focusing on textured areas) as well as SHOT [62] (3D features focusing on geometrically distinct regions) to establish point-to-point correspondences. These correspondences are posteriorly used to estimate the pose of the objects, effectively generating object hypotheses. The exploitation of these two modalities enables the recognition of textured and texture-less objects and thus meets the aforementioned requirements. Because the region described by local features is restricted to a specific support around *keypoints*, local pipelines perform well even in situations where objects are partially occluded or cluttered. However, their performance decreases when objects do not present enough or distinctive local features.

Conversely to the local paradigm which is based on point-to-point correspondences, the global paradigm directly establishes correspondences between views of the models and segments in the scene. To this end, the scene needs to be preprocessed by a suitable segmentation method aiming at clustering points in the scene that are likely to belong to single objects, effectively separating objects from clutter. Because the extension of the objects is provided by the segmentation stage, the support of *global features* includes all object points. This global support provides an enhanced discriminative power compared to local features, specially for objects that do no present distinctive local features. In addition, the representation of an object (as seen from a specific vantage point) by means of global features is usually more compact (a single or a few features) compared to the representation obtained with local features. However, because of their global nature, the performance of global pipelines decreases when objects undergo partial occlusions.

After the multiple recognition pipelines have hypothesized about the objects in the scene, the hypothesis are merged into the final hypothesis verification stage. At this point, a hypothesis is represented by the corresponding object model and by the transformation aligning the object model to the current scene. The availability of object poses enables this stage to directly compare the overlapping regions between scene and hypotheses, hence facilitating the definition of powerful cues. These cues are used to decide if the object hypotheses should be accepted or rejected. In addition, because multiple hypotheses are provided, it is possible to not solely compare the specific hypotheses to the scene but to consider the interaction between object hypotheses during the decision process. To this end, the proposed recognition framework incorporates a powerful hypothesis verification stage, dubbed *Global Hypothesis Verification* (GHV).

Instead of analysing one hypothesis at a time and deciding whether it should be accepted or rejected, GHV simultaneously considers all available hypotheses and selects a specific subset that globally represent a consistent interpretation of the scene. In particular, GHV formalizes the verification problem as the minimization of a global cost function defined over the set of available hypotheses. The minimization is guided by geometrical and appearance cues (computed both on the scene as well as on the specific hypotheses) enforcing a solution (accepted hypotheses) that best represent the scene and is physically plausible.

1.2 Contributions and outline

The applications, requirements and challenges associated with object instance recognition as well as our general approach have been outlined in the previous sections. We have seen that the proposed recognition system can be divided into three main stages (i) object modelling - Chapter 2, (ii) generation of object hypotheses - Chapter 3 and (iii) hypotheses verification - Chapter 4.

The recognition capabilities and advantages of the proposed recognition framework are demonstrated in Chapter 5 by a thorough evaluation on six public benchmark datasets for object instance recognition. In particular, the outstanding performance on heterogeneous scene configurations showcasing objects with different recognition relevant properties validates our design choices and contributions.

Finally, Chapter 6 demonstrates how the proposed recognition system can be used to effectively provide automatic ground-truth annotations of multi-view recognition datasets by taking advantage of the additional information provided by multiple vantage points. The rest of this section discusses in more detail the problems associated with each stage and outlines the contributions of this thesis.

1.2.1 Object modelling - Chapter 2

The availability of 3D models (eventually including color information) of objects is a key element for a recognition system like the one proposed in this thesis. It allows to uniformly and extensively sample different viewpoints around the object to train the different object recognizers as well as to relate the pose of an object in the scene to a 3D model of the object.

To this end, this chapter addresses the problem of reconstructing 3D models of objects from a set of partial views that can be acquired by the sensing machinery deployed on the robot. In the context of this thesis and in order to be useful during the on-line stages of the recognition system, we focus on accuracy and completeness of the reconstructed models. Because objects can be observed from arbitrary viewpoints during recognition, the availability of complete models (fully covering the viewing sphere of the object) facilitates this. Furthermore and as already outlined before, we would like to recognize objects with different properties (e.g. textured or texture-less, geometrically rich or uniform, big or small, etc.). Since models are used throughout the recognition system, it is important that the methods deployed during reconstruction are also able to handle the same variety of objects.

While reducing the assumptions on the type of objects to reconstruct is beneficial, it also increases the complexity of the problem. In order to reduce this complexity, the reconstruction pipeline in this thesis introduces the assumption that during the acquisition of the partial views, the object is placed on a textured planar surface like the one depicted in Figure 1.2. With this setup, the robot can move around the object or remain static while the planar surface rotates (i.e., turn table). The presence of a textured planar surface eases (i) the segmentation of the object from the background and (ii) provides enough features to lock the registration between consecutive scans –

pair-wise registration – when modelling feature-less objects. Aiming at increasing the convergence basin of the *pair-wise registration* stage, this chapter proposes a variant of the well-known Iterative Closest Point (ICP [12]) algorithm. The main contribution in this aspect is the use of a correspondence grouping algorithm to expand a tree of promising transformations during the iterative registration process. The desired accuracy in the reconstructed models is attained by means of a *multi-view refinement* stage that simultaneously optimizes the camera poses associated with different scans.

Figure 1.2: Pairwise registration and multi-view refinement. From left to right: (i) input scans, (ii) initial alignment obtained by concatenating transformations between consecutive scans and (iii) alignment after multi-view refinement. After segmenting the object from the background, the resulting point cloud represents a partial 3D model of the object.

In order to obtain complete 3D models of the object, the object is placed in a different configuration on the planar surface in order to reveal parts that were occluded in the previous configuration and the process outlined above is repeated. This process results in a set of partial 3D models (coming from different sequences) that need to be brought in alignment as depicted in Figure 1.3. With the assumption that objects lie stably on a planar surface during the data acquisition, the stable planes of the different partial models can be used to constraint and initialize the registration of two partial models. Please note that since the object configuration changes between two sequences, the surroundings cannot be used in this case to aid during registration and thus, the exploitation of stables planes facilitates the registration of partial models associated with feature-less objects.

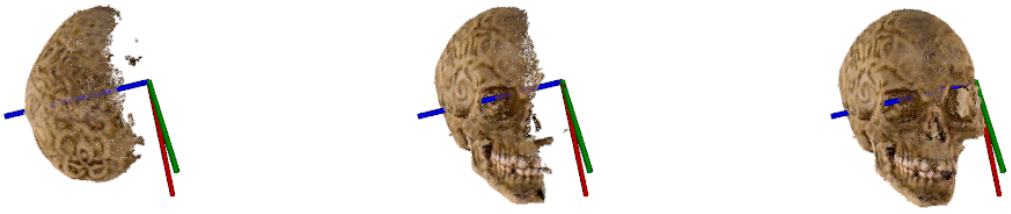


Figure 1.3: The first two point clouds represent 3D partial models reconstructed from two different sequences. The one on the right depicts the final reconstructed model obtained by aligning the partial models into a common coordinate system.

The proposed pipeline has been used to reconstruct 3D models of more than 50 objects with different properties which have been posteriorly used during the recognition and verification stages, hence validating the reconstruction pipeline.

1.2.2 Generation of object hypotheses - Chapter 3

During the previous sections, we have outlined the challenges associated with the recognition of objects that are partially occluded, surrounded by clutter or difficult to recognize because of their properties. We have also seen that the parallel deployment of different recognition methods with complementary strengths presents interesting properties when aiming at broadening the range of scene configurations and type of objects that the recognition system is able to handle.

Aiming at providing a further understanding of the different recognition paradigms available in the literature, this chapter first analyses the main stages involved in each paradigm. Based on this analysis, we present alternatives for the *correspondence grouping* stage, a key component of local recognition pipelines aiming at grouping point-to-point correspondences in order to generate enough consensus to hypothesize about the presence of objects in the scene. Specifically, we address the problem of grouping correspondences between an object model and the scene based on geometric constraints and propose a novel graph-based formulation that allows to solve the problem optimally. This formulation is specially useful for the detection of challenging objects (e.g, highly occluded or cluttered) which usually present just a few noisy correspondences.

Regarding the global paradigm, this chapter presents two global descriptors aiming at mitigating some of the caveats associated with this paradigm. In particular, the proposed global descriptors increase robustness to partial occlusions as well as the discriminative power of global features. A major contribution in this aspect is the definition of a repeatable coordinate system based on the surface properties of the object being described. This allows on one hand to effectively estimate the 6DoF pose of the objects in the scene and on the other hand increases the descriptiveness of global features. Finally, this chapter presents the details of the different components within the recognition system involved in the generation of object hypotheses.

1.2.3 Global Hypothesis Verification (GHV) - Chapter 4

The deployment of the recognition pipelines previously presented results in a set of object hypotheses, some of them representing correct detections while others being incorrect or duplicate. Aiming at reducing the amount of false responses while maintaining correct ones, recognition systems usually integrate an hypothesis verification stage aimed at improving the final result.

The common paradigm for this stage is represented by sequentially analysing each hypothesis and deciding whether it should be accepted or not (i.e., based on the overlapping quality between the specific hypothesis and the scene). While this paradigm has been shown to perform reasonably well in different scenarios, it usually involves

the definition of several hard thresholds to decide if the hypothesis is accepted or rejected. Moreover, this paradigm ignores the interaction between object hypotheses which in some cases provides additional information indicating whether a hypothesis is plausible or not. This is easily exemplified in the case of duplicate hypotheses, each of them overlapping on common parts of the scene. In this situation, only one (or none of them) of the overlapping hypotheses is correct. However, all these hypotheses might present a score above the acceptance threshold and therefore, become accepted if the decision is taken considering one hypothesis as a time. Please note that the sequential verification paradigm has been extended to handle this situation (by means of Non Maxima Suppression techniques) used here to exemplify the importance of simultaneously considering multiple hypotheses.

Motivated by this, the global hypothesis verification stage (GHV) proposed within this chapter simultaneously consider all available hypotheses and formalizes the verification problem as the minimization of a global cost function aiming at finding a subset of hypotheses that as a whole best represent the scene under consideration. In a nutshell, this stage aims at maximizing the amount of scene parts being explained by the selected hypotheses while minimizing several cues indicating implausible hypotheses (i.e., scene parts being explained by multiple hypotheses or visible model points without a correspondent region in the scene). In other words, GHV aims at finding a global model of the scene in terms of object hypotheses as depicted in Figure 1.4.

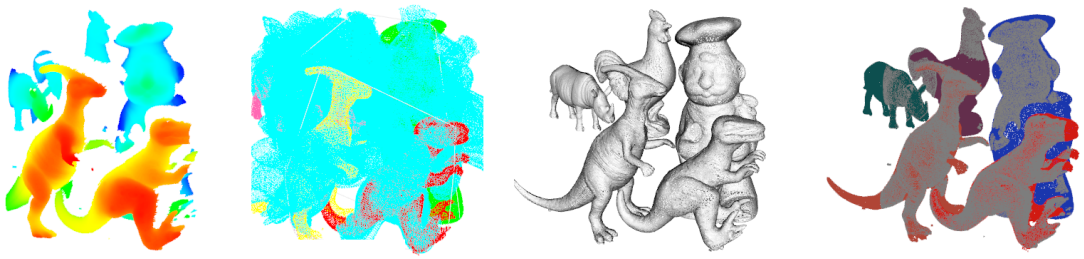


Figure 1.4: From the 162 hypotheses generated by the local recognition pipeline (middle left), only the correct 5 are selected (middle right) by the GHV framework. It can be observed that the accepted hypotheses accurately resemble the ground truth data (right) associated to the scene (left; color-coded according to distance to sensor). The 162 object hypotheses were generated using a 3D local recognition pipeline (SHOT) with Graph-based Geometric Consistency Grouping, $\tau = 3$.

A nice characteristic of the GHV framework is that it is agnostic on the underlying recognition methods. Therefore, better (and faster) recognition methods devised in the near future should be able to benefit from it. Related to this is the fact that the set of hypotheses being verified by GHV is not required to contain solely object hypotheses and can be extended with other hypotheses representing additional scene elements. For instance, this chapters shows how the hypotheses set can be extended with planar hypotheses in order to represent a recurrent element in human environments (see Figure 1.5). The inclusion of planar hypotheses allows on one hand to provide a more complete interpretation of the scene (in terms of objects and

planes) as well as aid in the rejection of false object hypotheses (e.g. hypotheses spanning on both sides of planes are penalized since they are physically unsound). Please, keep in mind that the more we know about the scene under analysis, the more accurate our decision will be. In particular, if we have models for all elements in the scene, the challenges posed by clutter are greatly simplified.

Figure 1.5: A sample scene extracted from the Challenge dataset (left) to demonstrate how planar and object hypotheses (middle) can be simultaneously provided to GHV. Their verification results in a complete model of the scene in terms of recognized objects and planar surfaces.

A challenge introduced by this new formulation involving the minimization of a global cost function is its high computational complexity. An hypotheses set of size n provides 2^n possible scene interpretations which rapidly renders exhaustive enumeration prohibitive. To this end, this chapter provides a careful analysis and evaluation of different meta-heuristic techniques in order to provide an approximate solution in polynomial time.

1.2.4 Automating "Ground Truth" annotations - Chapter 6

Motivated by the large amount of manual intervention and limitations of current automatisms related to the acquisition of object recognition datasets with ground truth data (objects in the scene with their associated poses), this chapter presents an accurate multi-view recognition method to automate the annotation process.

Under the assumption that the dataset provides multiple vantage points of particular static scene configurations, we argue that the additional information provided by multiple viewpoints eases the recognition problem to an extend that the results obtained with the proposed method closely resemble the ground truth of the data and can thus be used to accurately annotate the individual frames of the scene.

Specifically, this chapter proposes an extension to the hypothesis verification stage proposed in Chapter 4 in order to take advantage of multiple vantage points. In addition, we show how single-view recognition results for each particular frame in the sequence, together with visual features, can be used to provide an accurate estimate of the camera poses associated with the individual frames. This provides a 3D reconstruction of the scene that is used within the multi-view hypothesis verification stage to select object hypotheses obtained at individual frames that best represent the reconstructed 3D scene. The hypotheses verified during this last stage

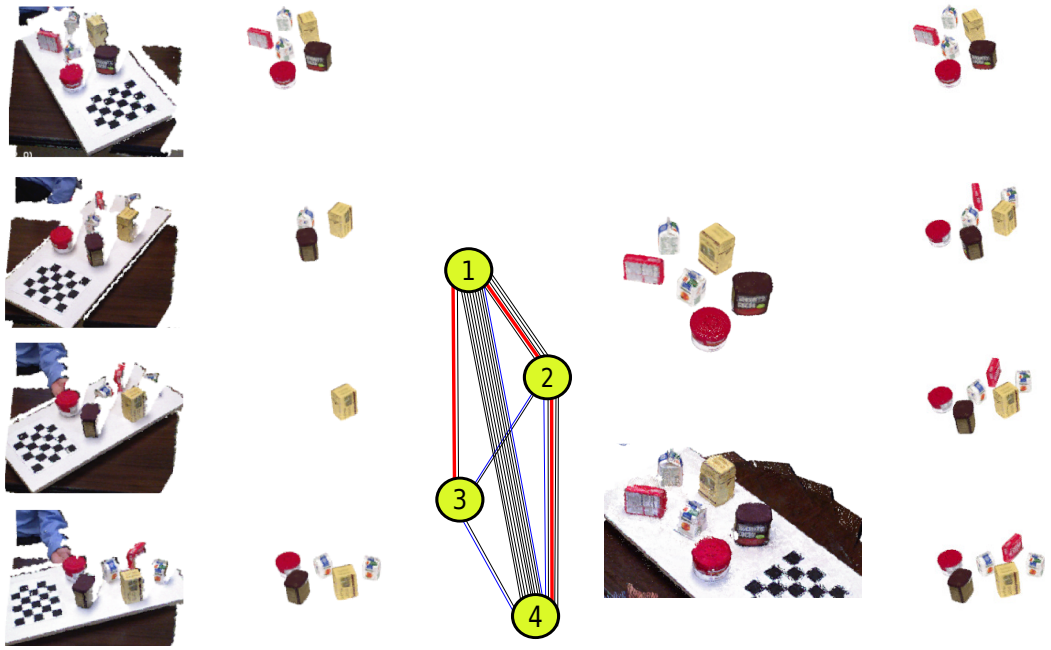


Figure 1.6: Workflow of proposed multi-view recognition method to generate "ground truth" annotations in a sequence consisting of 4 RGB-D frames: a) input RGB-D frames; b) single-view recognition results; c) graph representation of multiple views. If the same object was recognized in two views or the views can be registered by visual features (blue edges), an edge is added to the graph connecting the views with an associated transformation and an appropriate weight. The subsequently calculated Minimum Spanning Tree is shown by red edges; d) reconstructed scene and projected hypotheses remaining after the 3D verification stage; e) verified hypotheses are back-projected to the original frames, generating "ground truth" annotations.

are back-projected to the original frames, effectively generating their "ground truth" annotations. This process is visually depicted in Figure 1.6.

This scheme was used to automatically annotate more than 95% of the 3500 object instances in two large datasets totalling 516 RGB-D frames, including many frames where some objects were largely occluded. Thus, in combination with a final manual stage to verify and extend automatic annotations, the method is useful to annotate large amounts of data with a significant reduction in the amount of manual intervention.

1.2.5 List of publications

Parts of the content presented in this thesis have been previously published in the following manuscripts:

> *Aitor Aldoma and Markus Vincze.* **Pose alignment for 3D models and single view stereo point clouds based on stable planes**. In International

Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011. [7]

Aitor Aldoma, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, Markus Vincze, and Gary Bradski. **CAD-Model Recognition and 6DoF Pose Estimation Using 3D Cues**. In 3DRR Workshop (ICCV), 2011. [1]

Aitor Aldoma, Z-C Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. **Tutorial: Point Cloud Library: Three-dimensional object recognition and 6DoF pose estimation**. Robotics & Automation Magazine 2012. [3]

Aitor Aldoma, Federico Tombari, Luigi Di Stefano, and Markus Vincze. **A global hypothesis verification method for 3D object recognition**. In European Conference on Computer Vision (ECCV), 2012. [4]

Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. **OUR-CVFH: Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for object recognition and 6DoF pose estimation**. In Joint DAGM-OAGM Pattern Recognition Symposium, 2012. [6]

Walter Wohlkinger, Aitor Aldoma, Radu Bogdan Rusu, and Markus Vincze. **3D-NET: Large-scale object class recognition from CAD models**. In International Conference on Robotics and Automation (ICRA), 2012. [66]

Aitor Aldoma, Federico Tombari, Johann Prankl, Andreas Richtsfeld, Luigi Di Stefano, and Markus Vincze. **Multimodal cue integration through hypotheses verification for RGB-D object recognition and 6DoF pose estimation**. In International Conference on Robotics and Automation (ICRA), 2013. [5]

Aitor Aldoma, Thomas Faeulhammer, and Markus Vincze. **Automation of "Ground Truth" Annotation for Multi-View RGB-D Object Instance Recognition Datasets** (accepted for publication). In International Conference on Intelligent Robots and Systems (IROS), 2014. [2]

Chapter 2

Object modelling

The availability of 3D models (eventually including color information) of objects is a key element for a recognition system. They allow to uniformly and extensively sample different viewpoints around the object to train different object recognizers as well as to relate the pose of an object in the scene to a 3D model of the object. Registering partially overlapping scans of an object is a well-studied problem in 3D computer vision [29, 45, 65, 48, 20]. The main challenge remains to robustly assemble the different pieces to solve the task efficiently without many assumptions on the type of objects to be reconstructed. Huber and Hebert present in [29] a taxonomy to classify the reconstruction problem to be solved.

Following their taxonomy, this chapters addresses the problem of aligning n-views of an object with unknown initial pose estimates. To constrain the combinatorial explosion of the pairwise registration problem, the scans are assumed to belong to one or more ordered sequences. In this way, pairwise registration is carried on consecutive views. The original scans might come from a static sensor while the object moves (i.e. on a turn-table) or from a hand-held like sensor moving around a stationary object. If more than one sequence is available, the relation between sequences is unknown and needs to be estimated. Please note that if the object being reconstructed does not present salient geometrical or visual features, the registration problem is ill-posed and needs to be solved by using additional cues coming from the stationary — with respect to the object — surroundings of the object. In practice, in order to model all kind of objects accurately (even small, uniform and texture-less objects), we propose to always use the surrounding areas to increase the registration accuracy.

The registration process is divided into two parts. First, a pairwise registration stage (Section 2.2) is deployed to estimate the transformation between consecutive scans. This results in a chain of $N-1$ rigid transformations $\mathbb{T}$, N being the number of scans to be aligned, where T_i aligns the $(i+1)^{th}$ scan with its predecessor. Concatenating the different transformations, is it possible to bring all scans into the same coordinate system, effectively providing an *initial alignment*. Unfortunately, such initial alignment is not always accurate enough, as the local transformations — even if locally accurate — will result in a certain drift after several concatenations.

The second part, a multi-view registration stage (Section 2.3) aims at reducing

the registration error accumulated during the first stage. It considers all overlapping views to reduce the overall registration error by simultaneously optimizing the different camera poses. This process results in a significant error reduction providing accurate 3D models.

Moreover, the problem of merging two or more sequences of the same object (Section 2.4) to reconstruct a full 3D model is addressed. This poses an additional challenge because only information of the object can be used. It is clear that in order to reveal occluded parts of the object within the acquisition of a new sequence, the object's position relative to its environment needs to be changed relative to previous sequences, thus making the surrounding of the objects unusable in the registration of multiple sequences. For instance, when modelling a mug standing up-right on a surface, the base of the mug is not visible and the object needs to be turned upside-down to reveal the base.

Finally, in Section 2.5 we demonstrate how a noise model for RGB-D sensors as well as the availability of redundant data (i.e. surface parts being observed in multiple scans) can be exploited to increase the quality of the reconstructed models.

2.1 Related work

Within this section, methods addressing the problem of reconstructing a 3D representation of an object based on partial views acquired with 3D or RGB-D sensors are reviewed.

Huber and Hebert present in [29] a fully automatic in-hand object scanner. To this end, the process is divided into three stages: (i) data acquisition, (ii) pair-wise registration and (iii) global registration and refinement. During the data acquisition stage, the object is held before the sensor in different orientations. The object of interest is segmented out from the black background (the object is held using a black glove) by thresholding the intensity image. During the pair-wise registration, the different views are matched to one another by means of local features aiming at providing an initial registration for all view pairs, effectively constructing a complete graph. The graph nodes are represented by the different views and an edge between two nodes is represented by the transformations aligning two views together with an associated weight. The weight represents the alignment quality between two views and is computed by means of several surface consistency measures. Based on the aforementioned weights, edges are removed from the graph to remove inconsistent pair-wise alignments. Finally, an iterative algorithm is deployed to find a spanning tree of the model graph which can be used to bring all views in alignment and provides a globally consistent registration.

Because Huber and Hebert do not assume the order of views to be known, the algorithm incurs high computational cost as it requires all views to be registered to one another. A simplification is possible by assuming the order to be known which effectively reduces the complexity of the pair-wise registration stage from quadratic to linear. Indeed, other methods such as those presented in [65, 48] make use of

this assumption. In particular, they take advantage of real-time data acquisition to simplify the pair-wise registration by assuming that the object (or the camera) in the new frame present a small transformation with respect to the previous frame. Based on this assumption, two consecutive frames can be efficiently registered by a fast ICP variant, where the corresponding problem is solved by means of projective geometry. In addition to a great simplification of the pair-wise registration process, the redundant data can be effectively used to smooth the noise present in individual scans which results in a visually appealing 3D model of the object. [65] proposes as well an online loop-closure stage to reduce the error accumulation arising from small pair-wise transformation errors.

While the aforementioned methods are able to generate accurate 3D models under some assumptions (i.e., consecutive frames at high frame rate [65, 48] or distinctive shape properties on the object [29, 45]), they cannot be generically applied to obtain models for object recognition. In particular, high density of frames is not always available (i.e, some recognition datasets where the training data is provided by third parties) and/or the objects do not present enough geometrical features to solve the registration problem without ambiguities (texture-less objects with uniform shapes, i.e. bowl). In addition, if the data acquisition occurs with a stationary object and a moving camera, there are parts of the objects which are not visible in the specific configuration. While in-hand scanners do not present this issue, they require the object to be segmented from the background by means of an additional setup such as uniform coloured backgrounds and gloves.

In order to address the aforementioned issues and as anticipated during the introduction, a simple modelling setup is proposed within this thesis. In particular, we assume the object to be stationary (standing on top of a textured planar surface) while the camera is moving[1]. This allows to use the environment of the object to aid in the registration of featureless objects. In order to obtain a full 3D model, we propose the acquisition of multiple sequences of the object (each sequence revealing parts of the object occluded in other sequences) and a merging procedure based on the fact that the object to be modelled lies on a planar surface. Note that the assumption of the object being on a stable plane also simplifies the segmentation of the object from the background.

2.2 Pairwise registration

Let $\{\mathcal{S}_k\}_{k=1}^{N_\mathcal{S}}$ be the *source* scan and $\{\mathcal{T}_k\}_{k=1}^{N_\mathcal{T}}$ the *target* scan: pairwise registration aims at estimating the rigid transformation T that aligns (registers) the source to the target scan. There exist in the literature, several ways to define optimal alignment. Commonly, alignment is obtained by minimizing the error function

$$E(T) = \sum_{i=1}^{N_\mathcal{S}} e_i(T)^2 \tag{2.1}$$

[1] A turn table setup with static camera is also possible and presents similar characteristics

where

$$e_i(T) = \min_j \|\mathcal{T}_j - T(\mathcal{S}_i)\| \tag{2.2}$$

is the closest point distance. Such error definition is known as the point-to-point distance. Another common way to obtain the optimal alignment is given by minimizing the so called point-to-plane distance between source and target points [56]. Having defined the error function, the next step consist of finding an appropriate solver for the minimization problem. Because $\mathcal{S}_i$ depends on the parameters being optimized (T), the problem is solved iteratively by means of general-purpose optimization techniques (such as Levenberg-Marquardt [16]) or specialized procedures; *Iterative Closest Point* (henceforth ICP ([12])) being the standard one and several of its variants ([56]) designed to increase robustness during registration. In general, ICP and its variants are not guaranteed to converge to an optimal alignment and require a good initialization in order to avoid being trapped in a local minimum. Recently, a registration method based on Branch & Bounds has been suggested which guarantees convergence to a global minimum regardless of the initial estimate [69]. However, its computational complexity is in general too high.

The algorithm proposed within this chapter for the pair-wise registration problem is based on ICP. To ease explanation as well as to better motivate our proposal, let us first review the stages involved in the general ICP framework:

1. *Selection of points*: Select points from $\mathcal{S}$ and $\mathcal{T}$ to be considered during the minimization. Several strategies have been used in the literature, the most common being: no sampling, uniform sampling, random sampling, points exhibiting high image gradients (if color information is available) and 3D keypoints. A good strategy in the selection of points eases the following stages by reducing ambiguities.

2. *Correspondence estimation*: After points have been selected on both scans, this stage aims at finding corresponding points between them. Correspondences are then used to estimate the rigid transformation between the scans. Classical choices are: closest point (using specialized structures like kd-trees and octrees for increased performance), normal shooting[14] (closest point along the direction of the normal), reverse calibration[47], etc.

3. *Correspondence rejection*: This stage aims at rejecting spurious correspondences found by the previous stage. For instance, correspondences whose distance is higher than a certain predefined threshold can be rejected, rejection based on the standard deviation of the distances between corresponding points [42], RANSAC outlier-rejection strategies or a combination of them.

4. *Error minimization and transform estimation*: For point-to-point metrics, closed form solutions for the rigid-body transformation exist [18]. On the other hand, for the point-to-plane case, the least-squares solution is solved using generic non-linear methods (e.g. Levenberg-Marquardt) or through linearisation [38].

5. *Repeat until convergence*: Convergence is defined either by (i) incremental transformations difference falling below a user-defined threshold, (ii) overall error below a certain threshold or (iii) maximum number of iterations reached.

Adopting these stages, in this section a new variant of ICP is proposed, dubbed GGC-ICP (Graph-based Geometric Consistency ICP), which aims at increasing the convergence basis of the pairwise registration problem as well as reducing the number of iterations required for convergence. Moreover, an alternative registration error metric is proposed assessing the registration quality by means of the common point-to-point distance in combination with surface consistency measures.

2.2.1 Motivation

The main idea behind GGC-ICP is to include a correspondence grouping stage (GGC — see Section 3.2.2) within the classical correspondence rejection stage. Each group of consistent correspondences (according to the geometric consistency (GC) constraint and a RANSAC outlier rejection strategy) is then used to estimate a transformation minimizing the error function. At each iteration, all transformations are evaluated keeping the most "promising" ones which are consecutively fed into the next iteration. Such process results into a tree of transformations as illustrated in Figure 2.1.

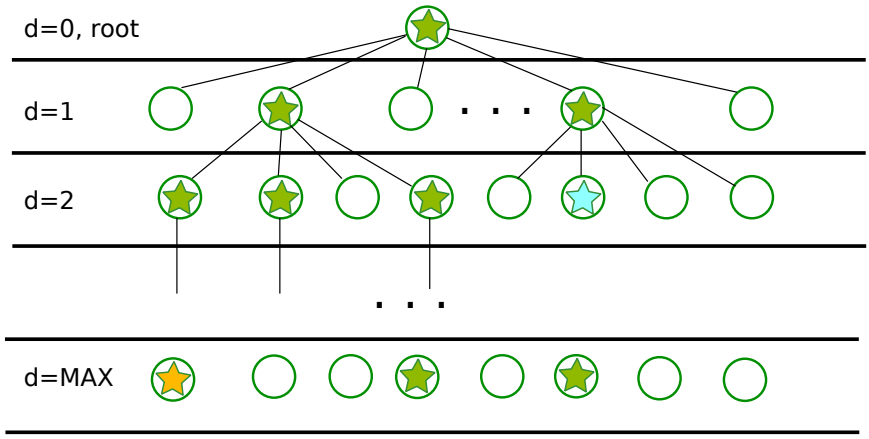


Figure 2.1: Tree of transformations generated during GGC-ICP. At each depth level, a certain number of nodes are being selected (marked with a green star) and fed to the next iteration, other nodes are discarded. Nodes marked with a blue hypotheses have converged and their branch is not expanded. The node marked with a yellow star (and its associated transformation) represents the best alignment.

Intuitively, there are several reasons to adopt such an strategy:

- After the second stage of ICP, consider the correspondence set $\mathcal{C}$ for a certain node $\mathcal{N}$ at depth d. When $\mathcal{C}$ is fed into the GGC algorithm, a set of geometrically consistent groups $\mathcal{G} = \{\mathcal{G}_i\}$ is obtained and for each group, an optimal

transformation T_i is estimated. Considering $|\{\mathcal{G}\}|$ is usually orders of magnitude smaller than the amount of RANSAC tests, it is possible to assess the registration quality for the transformations associated with each group, $E(T_i)$. Because $E(T_i)$ is used to prune the transformation tree at level d, branches might be explored arising from small consensus sets $\mathcal{G}_i$ instead of being discarded in favour of other transformations associated with a higher consensus based on the original correspondence set $\mathcal{C}$.

- The multiple-instance nature of the correspondence grouping stage allows to keep several *promising* hypotheses, thus exploring the space of possible transformations more exhaustively. One of them, eventually resulting, a few iterations later, in a better alignment than that yielded by the best hypothesis at current depth.

- Compared to rejecting correspondences based on their distance in combination with RANSAC outlier rejection, the *CG* stage allows to keep correspondences that are farther away and thus eventually accelerate convergence when a minimum consensus is reached. The maximum correspondence distance needs to be carefully selected based on the specific registration problem for the former.

2.2.2 GGC-ICP

As stated before, the GGC-CIP algorithm follows the general ICP framework and is thus composed of several stages. In addition to the modifications related to the correspondence grouping algorithm within the correspondence rejection stage, this section outlines the design choices in other stages to increase robustness. Figure 2.2 depicts the different stages involved in the proposed ICP pipeline.

Registration quality

We seek a transformation T^* bringing $\{\mathcal{S}_k\}_{k=1}^{N_\mathcal{S}}$ (the *source* scan) in alignment with $\{\mathcal{T}_k\}_{k=1}^{N_\mathcal{T}}$ (the *target* scan). Such T^* should maximize the following functional:

$$E(\mathcal{S}, \mathcal{T}, T) = \frac{\sum_{i=1}^{N_\mathcal{S}} e_i(T)}{\sum_{i=1}^{N_\mathcal{S}} \mathfrak{D}(\mathcal{S}_i, \mathcal{T}, T)} \cdot ov \cdot (1 - fsv) \tag{2.3}$$

where

$$e_i(T) = \begin{cases} 1 - \frac{\min_j \|\mathcal{T}_j - T(\mathcal{S}_i)\|^2}{\sigma^2} & \min_j \|\mathcal{T}_j - T(\mathcal{S}_i)\| \leq \sigma \\ \text{ignore} & \text{otherwise} \end{cases}, \tag{2.4}$$

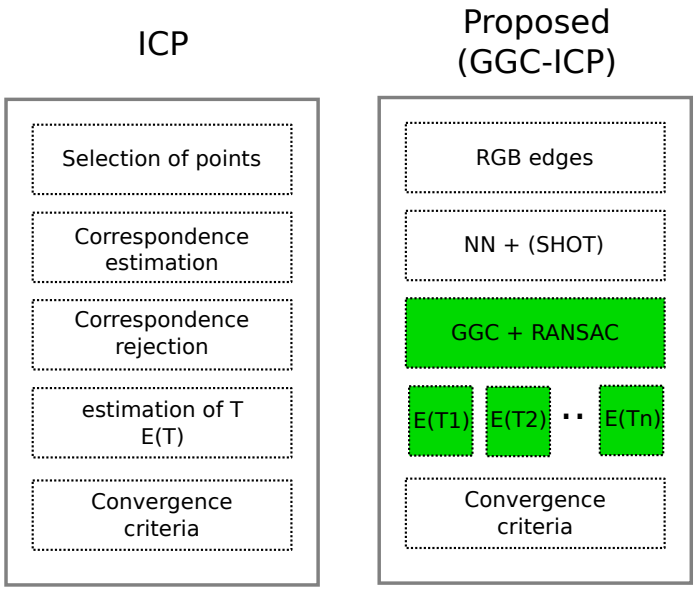


Figure 2.2: Stages within the general ICP framework (left). Proposed pipeline (right).

and σ indicates the maximum distance between two points ($\mathcal{S}_i$ and $\mathcal{T}_j$) in order to consider them to overlap. In particular, $\sum_{i=1}^{N_\mathcal{S}} \mathfrak{O}\left(\mathcal{S}_i, \mathcal{T}, T\right)$ counts the amount of overlapping points:

$$\mathfrak{O}\left(\mathcal{S}_i, \mathcal{T}, T\right) = \begin{cases} 1 & \min_j \|\mathcal{T}_j - T(\mathcal{S}_i)\| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The ov term indicates the overlap percentage between both scans:

$$ov = \frac{\min\left(\mathfrak{O}\left(\mathcal{S}, \mathcal{T}, T\right), \rho \cdot max_p\right)}{max_p}. \tag{2.6}$$

where $max_p = \min(N_\mathcal{S}, N_\mathcal{T})$ represents the maximum possible overlap between both scans (in points) and $\rho \in (0, 1]$ is a user-defined parameter indicating the desired maximum overlap percentage between both scans. Because consecutive scans will originate from different viewpoints, it is unrealistic to expect a full overlap between scans and thus the deployment of ρ suits the object modelling scenario.

For the fsv term, we follow the formulation of the FSV fraction proposed in [29] in order to penalize alignments infringing visibility constraints. Without going into detail, a point $T\left(\mathcal{S}_i\right)$ infringes the visibility constraint if it lies between the camera origin of $\mathcal{T}$ and an observed point $\mathcal{T}_k$. fsv is guaranteed to be in the interval $[0, 1]$ as it represents the ratio of points in $\mathcal{S}$ infringing the visibility constraint and the total number of points ($N_\mathcal{S}$). Note that in order to deploy the fsv term, the data needs to originate from a sensor following the pin-hole camera model (i.e. RGB-D sensors as

well as some laser scanners). Observe that $E(T)$ is bounded within $[0, \rho]$ and thus provides an easily interpretable error metric.

GGC-ICP algorithm stages

Let $\mathbb{A}$ be a list of alive nodes and $\mathbb{C}$ a list of nodes that already converged. Let us look at the stages depicted in Figure 2.2 initializing $\mathbb{A} = \{\mathcal{N}(T_0, E(T_0))\}$ and $\mathbb{C} = \emptyset$:

1. *Selection of points*: If RGB data is available, a quite robust and simple choice consists in selecting points in $\mathcal{S}$ and $\mathcal{T}$ with strong image gradients or those resulting from an edge detector (i.e. Canny [13]). Let $\mathcal{S}'$ and $\mathcal{T}'$ respectively represent the selected points on $\mathcal{S}$ and $\mathcal{T}$. $\mathcal{S}'$ and $\mathcal{T}'$ are computed at the beginning and do not change during registration.

For each node $\mathcal{N}_i$ in $\mathbb{A}$:

2. *Correspondence estimation*: To solve the correspondence problem, a simple closest point strategy is selected. The process is speed up by means of a kdtree. Optionally, a high-dimensional feature (i.e. SHOT) can be deployed to find good correspondences in order to bootstrap a good initial alignment or increase consensus when combined with the nearest neighbour correspondences. SHOT correspondences are computed at the beginning and remain unchanged throughout the iterations. For both types, correspondences are selected in both directions ($\mathcal{S}' \leftrightarrow \mathcal{T}'$). If the correspondences are equal in both directions, only one is kept. In the other case, both are fed into the next stage. GGC-ICP takes care of correspondences with same source or target, ensuring they won't be contained in the same correspondence group.

3. *Correspondence rejection*: The output of the previous stage is a set of correspondences $\{\mathcal{C}\}$ between $\mathcal{S}'$ and $\mathcal{T}'$. Within this stage:

   (a) Correspondences coming from closest point strategy are rejected based on their euclidean distance (similar to trimmed-ICP but with a looser threshold) [2].

   (b) The remaining correspondences are clustered by the GGC algorithm which includes a $RANSAC$ outlier rejection step. The output of GGC is a list of correspondences clusters, $\{\{\mathcal{C}_1, \}, \{\mathcal{C}_2, \}, ..., \{\mathcal{C}_n\}\}$, each correspondence in a certain cluster having passed the outlier-rejection test.

4. *Error minimization and transform estimation*: For each $\{\mathcal{C}_k^i\}$ in the clusters list, a transformation T_k^i is estimated and $E(\mathcal{S}, \mathcal{T}, T_k^i)$ evaluated.

Once all nodes in $\mathbb{A}$ have been processed and the resulting transformations T_k^i and $E(\mathcal{S}, \mathcal{T}, T_k^i)$ estimated,

[2] Is it possible to completely remove distance-based rejection. However, such a rejection might reduce the complexity for correspondence grouping.

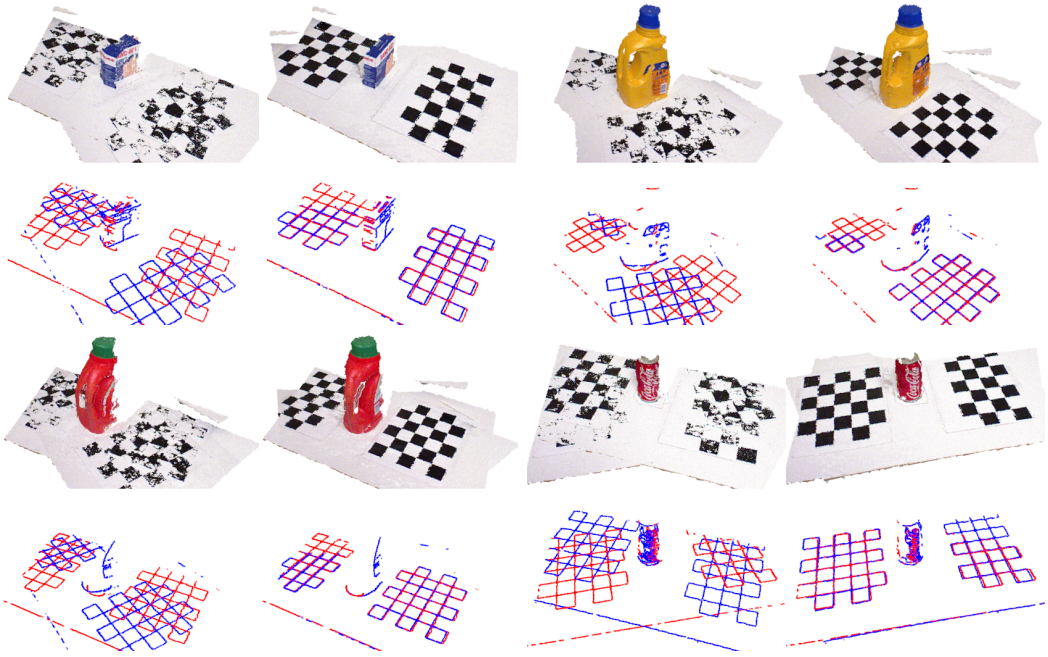


Figure 2.3: Four example registrations obtained using GGC-ICP when registering two views of an object on a turn-table (30 degrees apart from each other). The bottom row of each example shows the RGB edges used for registration (*source* shown in blue, *target* in red) the initial pose (left) and the resulting alignment (right). In all four examples, GGC-ICP was allowed to run for 10 iterations.

4. The transformations are then sorted in decreasing order according to $E(\mathcal{S}, \mathcal{T}, T_k^i)$. Similar transformations are filtered by means of non-maxima suppression. The best N transformations are passed to the next stage.

5. *Repeat until convergence*: At this stage, the newly created nodes are analysed for convergence. If the maximum number of iterations has been reached, all newly created nodes are considered to have converged and are added to $\mathbb{C}$. In the other case and for each *new* node independently, we analyse for convergence:

   (a) The incremental transformation from T_i and T_k^i is smaller than a certain threshold and $E(\mathcal{S}, \mathcal{T}, T_k^i) > E(\mathcal{S}, \mathcal{T}, T_i)$, then $\mathcal{N}_k^i$ is added to $\mathbb{C}$.

   (b) $E(\mathcal{S}, \mathcal{T}, T_k^i) \leq E(\mathcal{S}, \mathcal{T}, T_i)$, then $\mathcal{N}^i$ is added to $\mathbb{C}$

   In all other cases, the nodes are added to $\mathbb{A}$ and will be further explored in the next iteration.

Finally, considering all nodes in $\mathbb{C}$, the optimal transformation T^* is represented by the transformation associated to the node with the best cost. Figure 2.3 shows four examples registrations obtained using GGC-ICP on the training data provided for the Willow and Challenge datasets.

2.3 Multi-view refinement

Provided with a set of $N-1$ rigid transformations $\mathbb{T} = T_0, T_1, ..., T_i, ..., T_{N-1}$, N being the number of scans to be aligned and T_i bringing the $(i+1)^{th}$ scan into alignment with its predecessor, all N views might be transformed to the same coordinate system (i.e. the one defined by the first view) by concatenating the appropriate transformations $(T_0, ..., T_{N-1})$ in the correct order. As briefly mentioned before, through the concatenation of different transformations, small errors in each transformation get accumulated, resulting in a certain drift. Loop closure techniques might be used at this point to equally distribute the accumulated error among all pairwise transformations. The accumulated error can be computed by aligning the N^{th} scan with the first one. In practice, it has been seen that the pairwise error resulting from the GGC-ICP alignment is very small which in turn results into a small accumulated error. Therefore, the loop closure stage is unnecessary and the accumulated error can be directly reduced during the multi-view refinement stage studied in this section.

Within this section and without loss of generality, assume that all N scans to be registered have been brought to the same coordinate system by means of the aforementioned procedure and are effectively *initially* aligned. Now, the goal is to use all *overlapping* scans simultaneously to reduce the overall registration error. The basic idea behind multi-view refinement is to efficiently use constraints imposed by multiple views to optimize the absolute orientation of the scans in such a way that the registration errors between overlapping views are reduced. Recently, Fantoni et al. [20] provided a novel algorithm, extending a pairwise registration algorithm, the Levenberg-Marquardt Iterative Closest Point introduced by Fitzgibbon [16] (henceforth LM-ICP), to simultaneously cope with multiple views. A few minor modifications to [20] are proposed herein to increase robustness and accuracy by exploiting color and surface normals. For completeness and to ease the comprehension of the aforementioned modifications, let us first review [16] and [20].

2.3.1 LM-ICP

Fitzgibbon proposed to estimate the transformation aligning two scans by means of a general optimization technique (Levenberg-Marquardt) instead of specialized minimization algorithms like ICP and its variants. In [16], several nice properties of this choice are presented together with the derivation of the error function Jacobian required by the optimization method. Formally, let $\{\mathcal{S}_k\}_{k=1}^{N_\mathcal{S}}$ be the *source* scan and $\{\mathcal{T}_k\}_{k=1}^{N_\mathcal{T}}$ the *target* scan, the optimization aims at estimating the rigid transformation T, depending on certain parameters $\mathbf{a}$, that aligns (registers) the source to the target scan by minimizing the following error function:

$$E\left(T^{\mathbf{a}}\right) = \sum_{i=1}^{N_\mathcal{S}} e_i \left(T^{\mathbf{a}}\right)^2 \tag{2.7}$$

where

$$e_i\left(T^{\mathbf{a}}\right) = \min_j \|\mathcal{T}_j - T^{\mathbf{a}}(\mathcal{S}_i)\| \tag{2.8}$$

is the distance from a transformed point in the source scan $T^{\mathbf{a}}(\mathcal{S}_i)$ to the closest point in the target scan $\mathcal{T}$. For the 3D rigid-body transformation case, $\mathbf{a}$ is represented by seven parameters (3 for the translation component and 4 for the rotational part due to the use of unit quaternions). Hence, the goal of each LM iteration is to update the transformation parameters $\mathbf{a}$ such that the error is reduced. To this end, the Jacobian matrix $[J_{i,j}] = [\frac{\partial e_i}{\partial a_j}]$ of the error function is required. The derivatives can be efficiently computed by means of a distance transform, D, of the target scan:

$$D\left(x\right) = \min_j \|\mathcal{T}_j - x\| \tag{2.9}$$

where $x \in X$, X being a discrete volume enclosing the target scan. Applying the chain rule and combining Equation (2.9) and (2.8), the Jacobian is:

$$Ji, \cdot = \frac{\partial e_i}{\partial \mathbf{a}} = \bigtriangledown_x D(T^{\mathbf{a}}(\mathcal{S}_i)) \bigtriangledown_{T^{\mathbf{a}}} T^{\mathbf{a}}(\mathcal{S}_i) \tag{2.10}$$

$\bigtriangledown_{T^{\mathbf{a}}}$ can be computed analytically whereas $\bigtriangledown_x$ can be computed by finite differencing and remains constant throughout the optimization. It is possible to robustify the error function by means of an appropriate loss function (i.e. Huber):

$$\epsilon^2\left(d\right) = \begin{cases} d^2/2 & d \leq k \\ k \cdot d - d^2/2 & \text{otherwise} \end{cases}, \tag{2.11}$$

k representing the tuning parameter of the Huber loss function and d the distance between two closest points. The loss function can be easily integrated into the distance transform by defining the ϵ-distance transform:

$$D_\epsilon\left(x\right) = \epsilon(\min_j \|\mathcal{T}_j - x\|) \tag{2.12}$$

It is then possible to rewrite Equation (2.7) as follows:

$$E\left(T^{\mathbf{a}}\right) = \sum_{i=1}^{N_{\mathcal{S}}} \Big(D_\epsilon\big(T^{\mathbf{a}}\left(\mathcal{S}_i\right)\big) \Big)^2 \tag{2.13}$$

2.3.2 Multi-view LM-ICP

Using notions and notations from the previous section, [20] extends LM-ICP to handle multiple views. Let $V^1, ..., V^n$ be the set of views that are already initially aligned, in a common reference frame, and its absolute orientation needs to be refined. Because of the initial alignment of the views, it is possible to compute an overlap matrix, A, such that $A(h,k) = 1$ if the views h and k present enough overlap (i.e, the overlap is at least 30%). $A(h,k)$ is 0 otherwise. $\mathbf{a}_1, ..., \mathbf{a}_n$ represent the parameter vectors of the rigid transformation applied to the corresponding scan. The registration error between two views, V^h and V^k, can be written as:

$$E\left(\mathbf{a}_h, \mathbf{a}_k\right) = \sum_{i=1}^{N_V^h} A(h,k)\Big(D_\epsilon^k\big(T^{\mathbf{a}_h\mathbf{a}_k^{-1}}\left(V_i^h\right)\big)\Big)^2 \tag{2.14}$$

where D_ϵ^k is the ϵ-distance transform of V^k and $T^{\mathbf{a}_h\mathbf{a}_k^{-1}} = T^{\mathbf{a}_k^{-1}}T^{\mathbf{a}_h}$ aligns V^h to V^k. The overall error is then simply written by adding the contribution of all overlapping view pairs (i.e, those with $A(h,k)=1$), $S = \{(h,k) : A(h,k) = 1\}$:

$$\begin{aligned} E\left(\mathbf{a}_1, ..., \mathbf{a}_n\right) &= \sum_{(h,k)\in S}\sum_{i=1}^{N_V^h}\Big(D_\epsilon^k\big(T^{\mathbf{a}_h\mathbf{a}_k^{-1}}\left(V_i^h\right)\big)\Big)^2 \\ &= \sum_{(h,k)\in S}\sum_{i=1}^{N_V^h} e_{k,h,i(\mathbf{a}_h,\mathbf{a}_k)^2} \end{aligned} \tag{2.15}$$

The derivatives are computed similar to the LM-ICP case, considering both $\mathbf{a}_h, \mathbf{a}_k$:

$$e_{k,h,i(\mathbf{a}_h,\mathbf{a}_k)}/\partial_{\mathbf{a}_h,\mathbf{a}_k} = \bigtriangledown_x D_\epsilon^k(T^{\mathbf{a}_h\mathbf{a}_k^{-1}}\left(V_i^h\right))\bigtriangledown_{T^{\mathbf{a}_k,\mathbf{a}_h}}(T^{\mathbf{a}_h\mathbf{a}_k^{-1}}\left(V_i^h\right) \tag{2.16}$$

The resulting Jacobian J is a matrix composed by $q \times n$ blocks, q being the cardinality of S and n the total number of views. The block $J^{s,r}$ is associated with the s^{th} view pair in S. In such block-row s, only blocks related to h and k are non-zero. Because of this structure, J is a sparse matrix, which allows the use of sparse solvers. The total size of the Jacobian matrix is $N \times M$. M is $7 \times q$ (7 being the size of each parameter vector $\mathbf{a}_i$) while $N = \sum_{(h,k)\in S} N_V^h$.

For further details, the reader is encouraged to check the original papers [20, 16]. Please, note that the use of distance transforms to speed-up the Jacobian computation presents some practical limitations. Indeed, distance transforms will result in a discretization of the space and depending on the chosen resolution, result in inaccurate registration. Another limitation is presented by the memory consumption of distance transforms for large point clouds (large in the sense that their extension is large in respect to the chosen resolution). Therefore, in some situations, if speed is not a critical factor, it might be needed to use a slower version where finite differences are computed by means of octrees or kdtrees whenever the Jacobian is required. In such cases, computation of repeatable and sparse keypoints might be required to avoid large computational costs.

Adding color and normal information

This section considers a few modifications to the multi-view LM-ICP framework presented above. The motivation being that of improving registration in special situations as well as the addition of color information into the refinement process. Remember, that our goal was to register multiple scans of an object captured with RGB-D sensors.

Surface normals can be used to improve registration for objects presenting thin surfaces or double "walls". For instance, a mug presents an inner and outer surface, however, from a certain viewpoint, only one of them is visible. Because surface normals are initially computed on the partial scans, their sign is disambiguated by means of the viewpoint location and therefore, they are correctly oriented. The orientation of surface normals of close-by points can be used as an additional cue during registration to avoid absolute orientations resulting in such points to be registered. Additionally, color information might provide valuable information in order to lock registration on areas presenting uninformative geometrical traits. For instance and similar to the pairwise registration case, edges extracted on the color image, can be used during this refinement stage.

For the integration of color information, let $\mathcal{E}_{rgb}^{V^h}$ represent the color edges of a certain view V^h and $\mathcal{E}_{rgb}^{V^h} \subseteq V^h$. Then, each point $p_i \in V^h$ is assigned a certain weight w_i in such a way that if $p_i \in \mathcal{E}_{rgb}^{V^h}$, w_i is two times the weight assigned to points not in $\mathcal{E}_{rgb}^{V^h}$. Repeating that for all views, we obtain $W^1, ..., W^n$ representing the weights for all views. Equation (2.14) can be then rewritten as:

$$E\left(\mathbf{a}_h, \mathbf{a}_k\right) = \sum_{i=1}^{N_V^h} A(h,k) \cdot W_i^{(h,k)} \Big(D_\epsilon^k \big(T^{\mathbf{a}_h \mathbf{a}_k^{-1}} \left(V_i^h \right) \big) \Big)^2 \tag{2.17}$$

where $W_i^{(h,k)} = W_i^h \cdot W_{\phi(k,V_i^h)}^k$ and $\phi(k, V_i^h)$ is the closest point of V_i^h in V^k. Similarly, the weights can be integrated in the Jacobian. This effectively favours points on color edges to be properly registered and modifies the gradient accordingly.

Normal information is integrated in a similar fashion. Let $N^1, ..., N^n$ be the normals of the different views, for a certain point using V_i^h, its normal is N_i^h and the normal for its nearest neighbour in the k^{th} view is given by $N_{\phi(k,V_i^h)}^k$. To eradicate the influence of such situations during the optimization, we consider V_i^h to be an *outlier* within that iteration if $N_{\phi(k,V_i^h)}^k \cdot N_i^h < 0$. The corresponding row in the Jacobian matrix is than set to 0 to avoid the point of influencing the gradient. Figure 2.4 shows an example of the effect of including normals during the registration refinement.

2.4 Merging multiple sequences

In order to obtain complete 3D models of objects, multiple sequences of the object in different configurations are usually required. Since we cannot guarantee that the objects will always be in an up-right configuration, we will in general be required to learn the appearance and geometry of objects from all possible viewpoints. Unfortunately, in this stage, it is not possible to use cues that do not belong to the object. This poses additional challenges when modelling objects that do not present unique geometrical or visual features. However, the modelling process is performed under controlled situations (i.e, the object standing on a clutter free turn-table).

Figure 2.4: From left to right: initial alignment (pairwise), refined registration (multi-view without normals), refined registration (multi-view with normals). Observe how using normals avoids the inner and outer surface of the mug to be "merged" which results in a more reliable reconstruction. The sequence consists of 22 views, 18 taken while rotating the object on a turn table and the rest with hand-held camera to change the angle in order to see the concave part of the mug.

In [7], we proposed to use the *stable planes* of objects in order to reduce the degrees of freedom for the alignment of two objects belonging to the same class. The assumption there was that objects within the same class, share at least a stable plane. The problem is then reduced to find the stable plane shared among the two objects and solve the registration problem for the remaining degrees of freedom. It is clear that the same assumption holds when modelling a specific object and that the stable planes of the object of interest can effectively be used to provide initial estimates bringing the different partial models obtained from different sequences into alignment. In [7], stable planes of object were computed using the fact that such planes lie on the convex hull of an object, which remains true for partial models of an object. The problem of merging two partially overlapping sequences of an object is first addressed followed by the procedure to merge multiple sequences.

2.4.1 Stable planes based alignment (2 sequences)

Let $\mathcal{C}^1$ and $\mathcal{C}^2$ represent two partial models of the same object and $\mathcal{H}^1$,$\mathcal{H}^2$ their respective convex hulls represented as a triangle mesh. From $\mathcal{H}$, planes can be easily obtained by merging triangles sharing edges with similar normal information. During the merging procedure; total areas, vertices and average normals get computed. Let then $\mathcal{P}^1$ and $\mathcal{P}^2$ represent potential stable planes of the object on which the object could stand during modelling. Finally, the planes list are sorted in decreasing order based on their accumulated area. When partial models (coming from a sequence) are obtained with the object standing on a planar surface, it is possible to include the surface plane to the respective stable plane list.

The planes list, $\mathcal{P}^1$ and $\mathcal{P}^2$, of both partial models can be then used to estimate initial poses that bring into alignment both sequences. In particular, let $p_i \in \mathcal{P}^1$ and $p_j \in \mathcal{P}^2$ and suppose that p_i and p_j represent indeed the same stable plane in both partial models. By aligning the normals of p_i and p_j as well as their location along the aligned direction, $\mathcal{P}^1$ and $\mathcal{P}^2$ should be brought into an initial alignment. Still,

the position on the stable plane as well as the rotation about the plane's normal are unknown. From the initial 6 degrees of freedom, 3 have been approximated by means of the correct stable plane. The position on the stable plane is easily approximated by centring the two clouds along the plane according to their center of mass. Figure 2.5 depicts such a situation. Approximation of the final degree of freedom is quite more challenging. To overcome this, the input cloud $\mathcal{C}^1$ is spun around the stable plane normal at a certain angular step (i.e, 30°) in order to generate several approximations that need to be posteriorly refined.

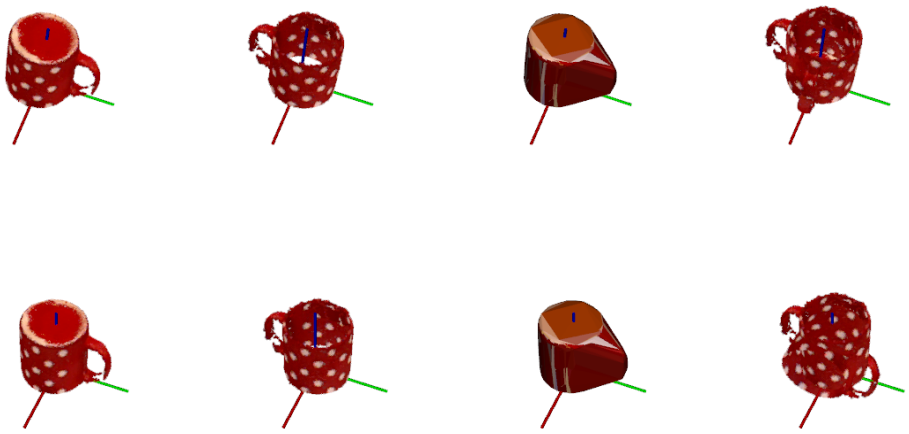


Figure 2.5: From left to right: input point cloud $\mathcal{C}^1$, target point cloud $\mathcal{C}^2$, convex hull of input cloud $\mathcal{H}^1$, $\mathcal{C}^1$ and $\mathcal{C}^2$ brought into initial alignment by means two stable planes from $\mathcal{H}^1$. The top row depicts the situation when the correct plane is selected (observe how the two partial models are correctly aligned up to the rotation about the z-axis; depicted in blue) while bottom row depicts the initial alignment after choosing an erroneous stable plane. Input and target point clouds obtained by aligning a sequence of scans.

For each stable plane pair (p_i, p_j), such that $p_i \in \mathcal{P}^1$ and $p_j \in \mathcal{P}^2$, the aforementioned scheme generates 12 (at an angular resolution of 30°) initial alignments for $\mathcal{C}^1$ and $\mathcal{C}^2$. The total amount of pairs (p_i, p_j) is easily reduced by considering the most probable planes of $\mathcal{P}^1$ and $\mathcal{P}^2$ according to their accumulated area (in practice, the 6 largest stable planes are included). Still, a certain large amount of initial alignments need to be refined; to this end, the previously proposed GGC-ICP is considered. The main idea being that of introducing all initial transformations obtained through stable planes at the *root* level of the GGC-ICP tree.

During refinement, the algorithm will automatically decide which hypotheses need to be further explored or pruned. Because GGC-ICP is in this case dealing with point clouds obtained by merging multiple viewpoints, the use of visibility constraints to evaluate registration accuracy is more complex. However, with the additional information obtained through multiple viewpoints, GGC-ICP seems in practice to

perform good without the visibility constraint in the cost function. Hence, Equation (2.3) can be rewritten for this case as:

$$E(\mathcal{S}, \mathcal{T}, T) = \frac{\sum_{i=1}^{N_{\mathcal{S}}} e_i(T)}{\sum_{i=1}^{N_{\mathcal{S}}} \mathfrak{O}(\mathcal{S}_i, \mathcal{T}, T)} \cdot ov \tag{2.18}$$

where $\mathcal{S}$ and $\mathcal{T}$ are two sequences of the same object and T will represent the transformation aligning both. The other terms of the function remain unchanged (see Equations (2.4), (2.5) and (2.6)).

2.4.2 Multiple sequences

If more than two sequences are to be merged, $\mathcal{C}^1, ..., \mathcal{C}^n$, the previous process is repeated for all sequences pairs. This results in an undirected weighted graph, G, its nodes being $\mathcal{C}^1, ..., \mathcal{C}^n$ and its edges representing the transformation resulting from the alignment process. The weight associated with each edge is ρ (user-defined maximal overlap in GGC-ICP) minus the confidence measure returned by GGC-ICP. The minimum spanning tree of G results in a chain of transformations that can be used to bring all sequences into the same coordinate system, effectively merging the provided sequences. Figure 2.6 shows a few examples of full 3D models obtained by merging different sequences.

2.5 Post-processing

The methods presented so far have been designed to be robust to noise and sensor nuisances. However, such artefacts are present in the data and a post-processing stage is required to remove them in order to obtain a visually appealing and accurate model. The techniques within this section provide a pleasant reconstruction by removing these artefacts from the underlying data. Figure 2.7 visualizes the improvement on the final reconstruction after the post-processing stage. Please note, that the methods herein, do not change the alignment results obtained during the registration process.

Two basic ideas are behind this stage: (i) the use of a noise model derived for RGB-D sensors [49] and (ii) the exploitation of data redundancy obtained by observing the object from different but overlapping viewpoints. On one side, the noise model provides an empirical foundation to remove or down-weight points that are likely to be wrong. On the other side, *good* points can be averaged together to provide a better estimate of the underlying surface based on the assumption that the average of multiple observations is more accurate than single observations. Data redundancy has been successfully employed to smooth noisy observations during real-time 3D reconstruction [30].

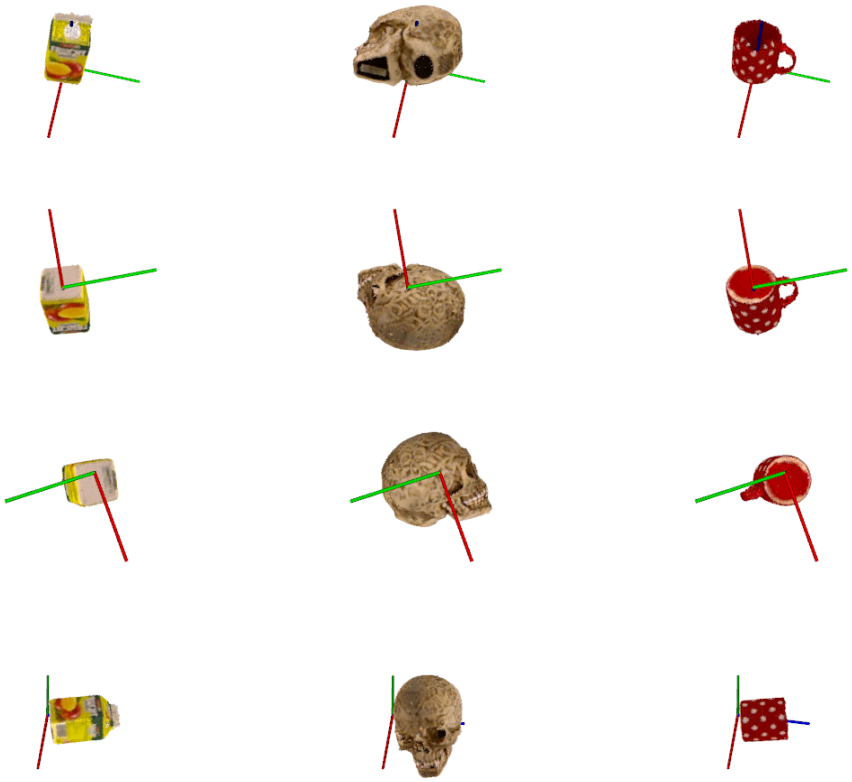


Figure 2.6: Four different viewpoints of three full models reconstructed using the proposed pipeline. From left to right: the models were obtained by merging 3, 2 and 4 sequences and are composed of a total of 50, 40 and 80 views. After merging sequences, the alignment was refined using all views by means of the multi-view refinement from Section 2.3.2 and processed afterwards using the methods in Section 2.5 to eliminate outliers and reduce noise.

2.5.1 Noise model

In [49], the authors study the effect of surface-sensor *distance* and *angle* on the sensor data (i.e., a Kinect sensor). They obtain axial and lateral noise distributions by varying the aforementioned two variables and show how to include the derived noise model into Kinect Fusion [30] to better accommodate noisy observations in order to reconstruct thin and challenging areas.

In particular, for object modelling, *surface-sensor angle* is more important than distance, since the later can be controlled and kept at an optimal range i.e., one meter or closer. For instance and following [30], one can observe that:

- Data quickly deteriorates when the angle between the sensor and the surface gets above 60 degrees.
- Lateral noise increases linearly with distance to the sensor. It results in jagged edges close to depth discontinuities causing the measured point to jump be-

Figure 2.7: Effects of the post-processing stage on the reconstruction results.

tween foreground and background. Combining depth with color information (even after accurate extrinsic calibration [25]) makes this effect clearly visible as color information from the background appears on the foreground object and viceversa. Observe the white points on the left instances of reconstructed models in Figure 2.7 coming from the plane on the background where the objects are standing.

From the previous two observations, a simple noise model suited for object modelling, can be derived that will result in a significant improvement on the visual quality of the reconstruction. To this end, the noise model is desired to result in a specific weight w_i for each of the points in the sensor scans.

Let $\mathcal{C} = \{p_i\}$ represent a point cloud in the sensor reference frame, $\mathcal{N} = \{n_i\}$ the associated normal information and $\mathcal{E} = \{e_i\}$, e_i being a boolean variable indicating whether p_i is located at a depth discontinuity or not. w_i is readily computed as follows:

$$w_i = \left(1 - \frac{\theta - \theta_{max}}{90 - \theta_{max}}\right) \cdot \left(1 - \frac{1}{2} exp^{\frac{d_i^2}{\sigma_L^2}}\right) \tag{2.19}$$

where θ represents the angle between n_i and the sensor, $\theta_{max} = 60°$, $d_i = ||p_i - p_j||_2$ (p_j being the closest point with $e_j = true$) and $\sigma_L = 0.002$ represents the lateral noise sigma. Because lateral noise is almost constant up to a certain angle, $\frac{\theta - \theta_{max}}{90 - \theta_{max}} = 0$ if $\theta < \theta_{max}$. The resulting weight map is used to filter points whose weight is below a certain w_t.

2.5.2 Exploiting data redundancy

In this final step, the data coming from multiple views is improved by averaging overlapping data. Assume that the different views are already brought into alignment with the resulting transformations. In particular, "double walls" originating from axial noise are removed by applying Moving Least Squares [8] with a small radius (i.e., 1 to 2mm) and taking into consideration the normal orientation (similar to previously done during the multi-view refinement). Finally, data is averaged by putting all points

into an octree structure with a certain leaf resolution. A representative is computed from all points falling in the same leaf by means of a weighted average (weights coming from the previous section). The resolution of the final model is equal to the selected leaf size.

2.6 Results

Using the proposed modelling pipeline, we were able to reconstruct accurate and visually appealing 3D models for more than 50 objects (some of them composed of multiple sequences) acquired with RGB-D sensors (Kinect and Asus Xtion). Figures 2.8 and 2.9 show several examples of 3D models reconstructed with the proposed pipeline. By taking advantage of the stationary surroundings of the object, the method is able to reconstruct uniform objects (both in terms of color and shape). Even though no quantitative results regarding the accuracy of the models are reported within this thesis, we show later on how these models can be effectively used during object recognition (the main goal of this thesis) which indirectly validates the accuracy of the reconstructed models obtained with the proposed pipeline. Note that the modifications introduced in the multi-view refinement stage regarding color information as well as the application of noise models exploiting data redundancy, provide a visually pleasant appearance in terms of texture registration.

Figure 2.8: Twelve 3D models from the Willow and Challenge dataset reconstructed using the proposed pipeline (a single sequence with 36 views per object).

Figure 2.9: Fifteen 3D models obtained with the proposed reconstruction pipeline. Some of them composed of multiple sequences. Observe the ability of the method to reconstruct texture-less objects as well as objects with uniform shapes.

Chapter 3

Generation of object hypotheses

This chapter addresses the problem of generation objects hypotheses in a given scene. In the context of this thesis, an object hypothesis, h_i, in a scene $\mathcal{S}$ is identified by a tuple $\{\mathcal{M}_i, \mathcal{T}_i\}$ with $\mathcal{M}_i$ being an object in $\mathbf{M}$ (model library) and $\mathcal{T}_i$ being the pose — 6DoF rigid body transformation (i.e. a 3D rotation and translation) — of $\mathcal{M}_i$ in $\mathcal{S}$. The general case of $\mathcal{S}$ containing any number of instances from $\mathbf{M}$ (as well as no instance at all), including the case of multiple instances of the same model is addressed. The output of the recognition methods detailed in this chapter is a set of n object hypotheses $\mathcal{H} = \{h_1, \cdots, h_n\}$, each hypothesis h_i will be finally verified or rejected during the hypothesis verification stage (see Chapter 4).

Aiming at broadening the range of object types and scene configurations handled by the proposed system, the deployment of multiple pipelines with complementary strengths has been thoroughly motivated throughout the introduction of this thesis. In order to provide a better understanding of the different object recognition and pose estimation techniques for 3D and RGB-D data, Section 3.1 provides an in-depth review of the most common recognition paradigms (e.g. *local*, *global* and *area-based*).

Based on this analysis, Section 3.2 present alternatives for the correspondence grouping stage, a key component of local recognition pipelines aiming at grouping point-to-point correspondences in order to generate enough consensus to hypothesize about the presence of objects in the scene. The deployment of correspondence grouping methods is of special interest in situations where the scene under analysis might be populated with several instances of the same model. Specifically, we further investigate the problem of grouping correspondences between an object model and the scene based on geometric constraints and propose a novel graph-based formulation that allows to solve the problem optimally. This formulation is specially useful for the detection of highly occluded or cluttered objects which usually present just a few noisy correspondences. Different methods to group correspondences in meaningful clusters are evaluated later on in Section 3.2.3.

Regarding the global paradigm, Section 3.3 presents two global descriptors aiming at mitigating some of the caveats associated with this paradigm. In particular, the proposed global descriptors increase (i) robustness to partial occlusions and (ii) the discriminative power of global features. A major contribution in this aspect is the

definition of a repeatable coordinate system based on the surface properties of the object being described. This allows on one hand to effectively estimate the 6DoF pose of the objects in the scene and on the other hand increases the descriptiveness of the associated global feature by enabling a richer spatial description of the object surface. The performance of different global descriptors is posteriorly evaluated in Section 3.3.5. Finally, the proposed recognition system is detailed in Section 3.4.

3.1 Related work

The great majority of methods for object recognition are based on the paradigm of feature extraction and matching, where each feature is grounded on specific characteristics of the processed data (e.g., 3D shape, topology, color, texture). Feature-based approaches can be further organized in *local* and *global*. The main difference being the supporting region that the underlying feature describes. An alternative to feature-based approaches is represented by area-based approaches which rely on template matching to detect model patterns in the current scene. In the following, the most common stages of the different paradigms are reviewed together with some of their representative methods. Special emphasis is given to the strength and weaknesses of the different paradigms.

3.1.1 Local recognition paradigm

A key component within the local paradigm is that of establishing point-to-point correspondences between model and scene points. To reduce the number of possible correspondences in a sensible way, (key)points [70, 39] in the model and the scene are associated with high-dimensional representations (*features* [39, 32, 58, 62]) which describe, in form of histograms, the associated point and its neighbourhood. Such representations allow to efficiently create correspondences among *similar* points by performing closest point searches in the high-dimensional feature space. The similarity between two points is implicitly defined by the feature itself as well as the metric used to compare them.

The matching stage is performed either by (i) sequentially matching each model (more precisely, the features associated with it) to the scene or alternatively, (ii) each descriptor extracted in the scene is matched, via fast indexing [46], against all descriptors associated with the objects in the model library. The second scheme provides better scalability when the model library grows at the cost of decreasing the probability of finding correct correspondences within the nearest neighbour (NN) search due to local inter-model similarities. A k-NN (with $k > 1$) matching stage can be deployed to counter-attack this, allowing each feature extracted in the scene to be paired with k features in the training set. Alternatively and pursuing the same goal, it is possible to use unsupervised clustering techniques (i.e., RNN [36] or k-means [33]) aiming at grouping similar model features; each feature in the scene is then matched against the representatives obtained during the clustering stage and finally paired with all training features associated with the closest representative feature.

Additionally, the distance between two points in the high-dimensional feature might be used in order to early discard spurious correspondences by means of hard thresholds or ratio thresholds between 1st and 2nd nearest neighbour as proposed by [24]. Nevertheless, such early rejection involves the selection of parameters (i.e. the definition of a L2-distance threshold in a high-dimensional space) and might result in correct correspondences being rejected. In general, it is advantageous to feed all correspondences to the next stages and decide later on if they are correct or not.

As a result of the matching stage, point-to-point correspondences are determined by associating pairs of model-scene descriptors. In order to hypothesize about objects in the scene, the set of model-scene correspondences requires to be appropriately processed in order to estimate one or more transformations (in the general case where multiple instances of the same object are to be found in the scene) aligning the object model with the scene (6DoF pose estimation). In addition, since some of the correspondences might represent erroneous point-to-point correspondences, the selected method is required to be robust to the presence of outliers.

To this end, several methods have been devised in the literature aiming at grouping the set of correspondences into outlier free subsets from which a rigid transformation can be directly estimated [9]. Within this thesis, we will refer to this stage as *correspondence grouping*. A popular choice is represented by iteratively applying RANSAC on the set of correspondences in order to find the best subset of correspondences providing consensus for a valid rigid transformation. Once such a group is found, the associated correspondences are removed from the original set and the algorithm is repeated until no more transformations can be estimated or no correspondences are left.

Aiming at reducing the computational cost associated with this approach, alternative methods have been proposed. One of them, which is further analysed and refined within this thesis in Section 3.2, exploits geometric constraints between pairs of correspondences in order to efficiently discard subsets that cannot represent a valid transformation. As a result of this algorithm, the original set of model-scene correspondences is clustered into geometrically consistent subsets that are further analysed and used to estimate object poses in the current scene. Another alternative is represented by the method proposed by Tombari and Di Stefano [61]. In their approach, each point is associated with an oriented local reference frame which is used to vote in a 3D Hough space, each cell representing a specific rigid transformation. Even though their approach is very efficient and theoretically sound, we will show in Section 3.2.3 that the stability of the local reference frames associated with each point is compromised by high levels of clutter and occlusion.

Representative methods

Because of its ability to handle clutter and occlusions, the local recognition paradigm has been extensively explored in recent years. Without aiming at a full review, some of the most recent and successful methods within this category are reviewed hereinafter. Even though local features have been extensively used in 2D computer

vision for the task of object recognition [39], these methods are not reviewed in this thesis due to their inability to estimate a 6DoF pose unless multiple views of the scene are provided [24, 15]. In particular, we focus on methods deployed on 3D data as well as RGB-D frames.

Concerning local features exploiting the color (and texture) modality of RGB-D frames, SIFT features [39] are used in [60] to stablish point-to-point correspondences between the scene (segmented clusters in this particular case) and a candidate list of models obtained using color histogram comparison. The pose of the models is posteriorly obtained by means of a RANSAC stage aiming at finding the object hypothesis with highest consensus. A similar approach is exploited in [68]. However, in this case, SIFT features are densely extracted in the models and the scene which results in a significant improvement over keypoint-based SIFT extraction.

Regarding methods exploiting solely the shape information available in range images, 3D scans or RGB-D frames, the method in [31, 32] uses the *Spin Images* feature to solve the correspondence problem. After correspondences have been established between the models and the scene, they are grouped in geometrically consistent clusters from which a rigid transformation can be estimated. The authors propose for the grouping stage a grouping criterion that promotes correspondences that are far apart from each other (which usually results in more stable transformations).

Mian et al. [44] propose to match scene points to the model library by means of *tensors*. Contrary to [31, 32] where each object is sequentially matched with the scene, this methods matches scene points to the model library and thus incurs in a lower computational cost. Because tensors provide a fully oriented coordinate basis, the pose of the objects in the scene is estimated based on a single correspondence and thus does not require a correspondence grouping stage. It is however important to note that pose estimates based on a single correspondence are sensitive to small differences in the respective coordinate bases (scene and model).

More recently, methods based on point-pairs matching [17, 50] have been successfully deployed for object recognition and pose estimation. The idea behind these methods is to sample pairs of points in the scene from which a descriptor is extracted and used to efficiently find correspondences on the model database. Because point-pairs (together with their normals) can be used to estimate a unique reference frame, single point-pair correspondences can be used to generate model pose estimates. To increase the reliability of the pose estimates, point-pairs are selected only if the distance between both points lies in a specific range: chosen to be small enough to handle clutter and occlusions while being large enough to provide accurate pose estimates.

Finally, another common alternative is represented by methods based on point histograms [62, 58]. In particular, these methods compute a histogram that describes geometrical traits in a region around the selected point. These histograms are posteriorly used to stablish point-to-point correspondences between model and scene. In [4] we used the SHOT descriptor[62] in combination with a correspondence grouping stage for 3D object recognition in clutter.

Strengths and weaknesses

By constraining the supporting region that local features describe (both during training and recognition stages), the deployment of methods based on local features is suited for the recognition of objects under occlusion and clutter. In addition, because the supporting region is predetermined (by a parameter choice or through an analysis of the processed data, i.e, keypoint scale), they do not require objects to be segmented in advance.

However, because of their locality, they struggle to recognize objects presenting repetitive patterns (within the same object or among different objects) as well as objects presenting featureless regions (in terms of the specific characteristic described by the underlying feature). Both factors inevitably increasing ambiguity during the matching stage. While increasing the extent of the supporting region associated with local features increases their discriminative power and thus reduces ambiguities, it has also an undesired effect on their ability to handle clutter and occlusions.

3.1.2 Global recognition paradigm

Within the global paradigm and differently to the local one, the support being described by a global feature includes the whole surface of the object. This requires the scene to be processed by a suitable segmentation stage, aiming at grouping points or pixels belonging to the same object. After the extraction of appropriate object segments, they are described by global features [6, 1, 67, 57] yielding a compact representation of the segment's surface, in the form of an histogram. Contrary to local recognition pipelines, object hypotheses are in this case directly obtained during the matching stage which associates each segment in the scene with the k most similar model views. The association is again provided by comparing the descriptors extracted from the scene with the histograms obtained during a training stage. To recover a 6DoF pose, an additional stage is required, aiming at solving the ambiguity along the camera roll. Within this thesis, two global features are proposed together with two methods for 6DoF pose recovery.

Representative methods

In addition to the global features presented within this thesis, other methods have been proposed in the literature. Aiming at object classification rather than object instance recognition, the ESF (Ensemble of Shape Functions) descriptor was introduced in [67]. It is an ensemble of ten 64-bin sized histograms (resulting in a total descriptor size of 640 bins) of shape functions describing characteristic properties of the point cloud. The shape functions consist of angle (point triplets), point distance (point pairs) and area shape (point triplets) distributions. A voxel-grid ($64 \times 64 \times 64$) serves as an approximation of the real surface and is used to efficiently trace the line joining a point-pair sample. By tracing a line within the voxel-grid, the statistics related to the different shape functions can be classified to be either "on the surface", "off the surface" or a combination of both. In [66], the ESF descriptor is evaluated for

the task of object classification. Please note that because ESF was designed targeting object classification instead of object recognition, the descriptor is desired to capture rough geometrical traits of objects within a certain category instead of specific details associated with a particular object instance.

Another representative of global features is the VFH (Viewpoint Feature Histogram) descriptor, originally proposed in [57]. The VFH descriptor is composed of four angular distributions extracted from the surface normals. Particular to VFH is the encoding of statistics related to the viewpoint from which the object is sensed. VFH serves as a basis for the CVFH and OUR-CVFH descriptors presented within this thesis and is therefore reviewed in more detail in Section 3.3.

Strengths and weaknesses

The major weakness of methods based on global features is their poor performance when objects undergo occlusions. In addition, because of the required segmentation stage, their performance is strongly affected by typical segmentation issues such as under- and over-segmentation. Since accurate segmentation in cluttered scenes remains unsolved, the successful deployment of these methods is limited.

On the other hand, thanks to their global nature, they are highly discriminative and show good performance in recognition scenarios characterized by objects presenting similar shapes. Moreover, the representation of the model library by means of global features is very compact (only a few dozens of global descriptors are required for a reliable representation of an object model) and they present in general low computational costs (both in terms of feature computation and matching).

3.1.3 Area-based recognition paradigm

An alternative to feature-based approaches is represented by area-based approaches [26, 64] which rely on template matching to detect model patterns in the current RGB-D scene, i.e. by comparing a high number of model templates related to different vantage points and distances at different positions and scales in the scene.

Representative methods

One of the most successful area-based methods in the literature is LINEMOD 3D [26]. During a training stage, each object to be recognized is represented by multi-modal templates which compactly describe color and normal gradients on the object's surface. Because templates are not invariant to scale or in-plane rotations, in addition to the different vantage points, templates must cover different scales as well as in-plane rotations. This results in a few thousand of model templates to cover the upper hemisphere of the object of interest. Note that templates can be annotated with pose information in order to provide a coarse estimate of the pose of the object.

Due to this large number of model templates, the authors propose several optimizations based on the pre-computation of response maps and take into account the architecture of modern processors to render the recognition stage computationally

efficient. This allows to match each template at densely sampled locations in the scene, effectively providing a template matching score for each evaluated location, together with a coarse pose estimate. Furthermore, the detections are processed by means of Non Maxima Suppression stage in order to keep the best response at similar image locations. Finally, the authors propose a post-processing stage to remove false detections that will be analysed in more detail in Chapter 4.

Strengths and weaknesses

Similar to global feature-based approaches, the performance of area-based approaches decreases as objects undergo occlusion because of the global extent of the model templates. However, they do not require the objects in the scene to be previously segmented which represents an advantage in scene configurations populated with clutter.

Even though, template-matching approaches have been shown to be able to perform in real-time [26] for model libraries containing a small amount of objects, the large amount of templates required to cover the full spectrum of vantage points and scales raises some concerns regarding their scalability with respect to the model library size. In addition, area-based methods require the input data to be organized in a grid structure (i.e., images or RGB-D frames).

3.2 Correspondence Grouping

Because of the important role of correspondence grouping within the local recognition paradigm, this section analyses two methods aimed at efficiently solve the grouping problem at hand. Because methods based on local features do not necessarily rely on a previous segmentation of the scene, model-scene correspondences (some of them possibly representing wrong associations) require to be processed in order to hypothesize about the objects in the scene.

In general, the grouping problem can be formulated as follows: Given a set of correspondences $\mathcal{C} = \{c_1, .., c_i, .., c_n\}$ between a specific model in the library and the scene under consideration, find subsets of correspondences $\mathcal{C}_k$ from which a rigid transformation can be estimated. Each correspondence, c_i, is represented by a triplet including the corresponding model and scene points together with the distance between model and scene features obtained during the matching stage:

$$c_i = \{p_i^m, p_i^s, d_i\} \tag{3.1}$$

In addition to other parameters specific to particular methods, correspondence grouping methods usually involve at least a parameter, τ_{CG}, representing the minimum consensus size desired to hypothesize about objects. Because at least three points are required to estimate a rigid transformation, the theoretical minimum for τ_{CG} is 3. In the following, two methods based on the geometric constraints are analysed in detail.

3.2.1 Iterative Geometric Consistency grouping (IGC)

The basic idea behind IGC is to find in an efficient way subsets of correspondences $\mathcal{C}_k$ in $\mathcal{C}$ such that all $c_i \in \mathcal{C}_k$ are geometrically consistent to one another. Intuitively, subsets of correspondences being geometrically consistent are more likely to represent a good set from which a rigid transformation can be estimated. Two correspondences, c_i and c_j, are said to be geometrically consistent if:

$$\left| ||p_i^m - p_j^m||_2 - ||p_i^s - p_j^s||_2 \right| < \varepsilon \tag{3.2}$$

with ε being a parameter of this method, intuitively representing the consensus set dimension. Figure 3.1 illustrates the geometric constraint in Equation (3.2). The choice of ε is usually related to the expected inaccuracy of the keypoint locations.

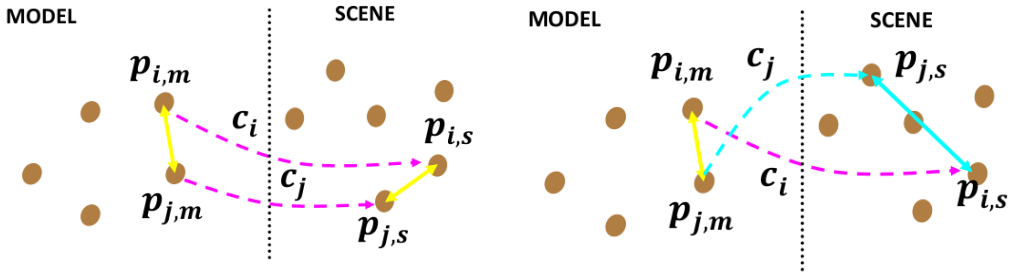


Figure 3.1: Illustration of the geometric consistency constraint. The correspondences on the left are said to be geometrically consistent because their distance in the model and the scene is below ε (see length of the yellow arrows). On the other hand, the correspondences on the right part of the image, are not geometrically consistent.

The IGC algorithm to find geometrically consistent subsets of correspondences is very simple. Let $b_i = 1$ indicate that the correspondence c_i has already been included in a consensus set and $b_i = 0$ otherwise. Starting from a seed correspondence c_i, such that $b_i = 0$ and iterating over all correspondences c_j, such that $b_j = 0$, the correspondence c_j is added to the subset seeded by c_i if Equation (3.2) holds between c_j and all correspondences within the group seeded by c_i. Because there is no guarantee that all correspondences within a group correspond to a real rigid transformation (see Figure 3.2), after the consensus set cannot be further expanded, outliers are discarded by means of RANSAC stage and their respective b_i are set back to 0 which allows these correspondences to participate in the consensus provided by subsequent groups. The process is repeated until all correspondences have been assigned to a subset or all correspondences have been used to seed a new consensus set. The correspondences are initially sorted based on their score obtained during the matching stage (d_i) in order to build consensus subsets seeded by *good* correspondences (i.e, close in the feature space).

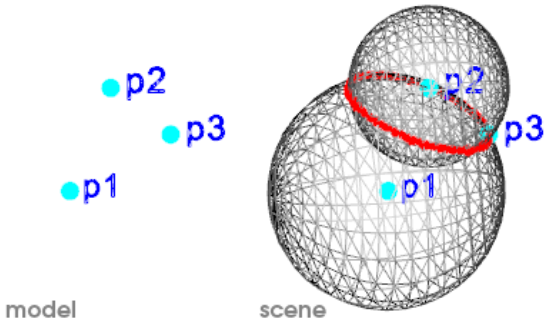


Figure 3.2: *IGC* constraint ambiguity: in this toy example, let the 3 points $\{p_1, p_2, p_3\}$ on the model (left-side) be associated with the respective ones on the scene (right-side), forming the 3 correspondences $\{c_1, c_2, c_3\}$. If the current consensus set only contains c_1, by evaluating p_2, all points belonging to the sphere centred in p_1 and of radius $p_2 - p_1$ will satisfy the IGC constraint. If the consensus set contains both c_1 and c_2, when evaluating p_3, all points lying on the intersection of the two spheres centred in p_1 and in p_2 of radius, respectively, $p_3 - p_1$ and $p_3 - p_2$, (depicted in red in the figure) will satisfy the constraint.

3.2.2 Graph-based Geometric Consistency grouping (GGC)

Despite of its speed, simplicity and good performance, IGC has a few drawbacks. In particular, because of its greedy nature, the final consensus sets depend on the initial sorting of correspondences. This is usually not a problem if $\mathcal{C}$ contains enough *good* correspondences and the grouping problem is *easy*. However, in more challenging scenarios with fewer and noisier correspondences, this might result in missing the unique hypothesis representing the actual object. Related to this is the fact that once a correspondence gets assigned to a consensus set, the correspondence cannot contribute to other sets.

Nevertheless, the formulation of the correspondence grouping problem based on geometrical consistency constraints has several nice properties and has been shown to work very well on practical situations [4, 3]. Therefore, it is worthy trying to improve the basic algorithm in order to solve or mitigate the aforementioned caveats.

A simple modification to the GC constraint (Equation (3.2)) aiming at decreasing ambiguities is attained by considering surface normals. Let $\{n_i^m, n_i^s\}$ respectively represent the surface normals at points $\{p_i^m, p_i^s\}$ associated with c_i and $\{n_j^m, n_j^s\}$ the normals at $\{p_j^m, p_j^s\}$ associated with c_j. Correspondences c_i, c_j are geometrically consistent if Equation (3.2) holds and

$$\left| n_i^m \cdot n_j^m - n_i^s \cdot n_j^s \right| < \varepsilon_n \tag{3.3}$$

holds as well. ε_n represents the maximum angle deviation between normals in the scene and the model so that c_i, c_j are geometrically consistent. With the addition of normals, an extra condition on the surface normals of the model is formulated,

$$n_i^m \cdot n_j^m < 0 \tag{3.4}$$

aiming at discarding correspondences where the points involved are on opposing surfaces on the model[1]. Such correspondences are not consistent due to the fact that the scene is sensed from a single viewpoint and the observation of opposing surfaces is physically impossible.

Slightly more subtle, one can observe that the GC problem can be formulated by constructing an appropriate graph $G_{GC} = (\mathcal{C}, E)$. The node set is composed of all correspondences between the scene and the model under consideration. The edge set, E, is composed of edges joining two nodes (correspondences), c_i, c_j, if c_i, c_j are consistent according to Equations (3.2), (3.3) and (3.4). With this new representation, it is possible to exploit several results in graph theory as well as optimal algorithms to solve the problem at hand. In particular:

- The correspondence grouping problem (based on the GC constraint) is optimally solved by finding all maximal cliques within G_{CG} such that their size is $\geq$ than a user defined consensus threshold τ_{CG}. By definition, the maximal cliques of a specific graph represent the largest complete sub-graphs in it. In this case, the largest possible subsets of correspondences being geometrically consistent to one another.

- For a correspondences c_i to possibly be part of a consensus set, its degree, $\delta(c_i)$, is at least $\tau_{CG} - 1$. Otherwise, it cannot possibly belong to a clique of size τ_{CG} or larger.

- The connected components of G_{GC} allow to split the grouping problem in smaller sub-problems. Ideally, each connected component should map to an actual instance of an object in the scene. Because of ambiguities associated with the GC constraint, there is however no guarantee that a specific connected component represents a single object instance. A stronger condition can be defined by means of biconnected components [28].

Unfortunately, finding the maximal cliques in a given graph is a hard problem and its computational complexity is high ($\mathcal{O}\left(3^{n/3}\right)$ as proved by [63], n being the number of nodes in a graph). However, as mentioned before, IGC is reliable when the number of correspondences is large enough which usually indicates that the object is not occluded in the scene. Because the algorithm in [63] is in practice very fast for *small* graphs, a mixed algorithm is proposed (see Algorithm 1). Instead of trying to define based on characteristics of a graph if it is small enough (i.e., number of nodes or edges), the maximal clique computation is allowed to run for a specific amount of time (i.e, 100ms). In case that it is not able to terminate within this amount of time, the algorithm is pre-empted and the correspondences belonging to the specific connected component are grouped with IGC.

Because the number of cliques can be large even for small graphs, the parameter max_{taken} controls the amount of hypotheses in which a single correspondence can participate. A value of 5 usually provides a good trade-off between accuracy and

[1]In practice, a small negative number can be used to accommodate for noisy normal estimations

Algorithm 1 Graph-Based Geometric Consistency Grouping. The biconnected components of the graph G_{GC} obtained by means of geometric constraints are analysed. For each component, the maximal cliques are extracted and sorted based on the correspondences properties included in each clique. The cliques are further analysed to remove spurious correspondences (RANSAC outlier rejection) and inliers are used to generate object hypotheses. Each time a correspondence is used to generate an object hypothesis, the algorithm increases the taken counter for the specific correspondence. Once this counter reaches the max_{taken} value, the correspondence cannot be posteriorly used to create consensus for object hypotheses originating from subsequent cliques. In case that the clique extraction does not succeed within the allocated time, i.e. 100ms, object hypotheses for the specific component are generated by means of IGC.

Require: $G_{GC} = (\mathcal{C}, E)$, τ_{CC}, $max_{taken} = 5$
  $\mathcal{H}_{GC} = \{\emptyset\}$
  $CC_{G_{GC}} = \{cc_1, ..., cc_n\} \leftarrow$ *biconnected_components*(G_{GC})
  for all $cc_k \in CC_{G_{GC}}$ **do**
    if $\left|cc_k\right| \geq \tau_{CC}$ **then**
      success, cliques $\leftarrow$ *maximal_cliques*(cc_k, 100ms)
      if success **then**
        sort(cliques)
        for all clique $\in$ cliques **do**
          clique $\leftarrow$ *preprocess*(clique, max_{taken})
          clique $\leftarrow$ *RANSAC*(clique)
          if $\mid$ clique $\mid \geq \tau_{CC}$ **then**
            $\mathcal{H} \Leftarrow$ *obj_inst*(clique)
            Update taken $\forall c_i \in$ clique.
          end if
        end for
      else
        $\mathcal{H}_{GC} \Leftarrow IGC(cc_k)$
      end if
    end if
  end for
  return $\mathcal{H}_{GC}$

number of generated hypotheses and is used throughout the thesis. max_{taken} takes implicitly a value of 1 for IGC. The advantages of allowing single correspondences to be part of multiple clusters has already been pointed out in [31] in order to handle object symmetries. Please note that the IGC version used in the GGC algorithm also takes advantage of the graph structure which includes the normal consistency checks.

An additional advantage of GGC over IGC is that the sorting stage takes places after all geometrically consistent groups (i.e, maximal cliques) have been generated. This allows to consider global clique properties (i.e, clique size, average correspondence distance in feature or Euclidean space) instead of single correspondence properties which are in general less stable to noise. In addition, splitting the problem by means of the graph's connected components allows to properly handle the case where multiple instances of the same object are present in the scene, some of them being occluded and cluttered (GGC) while others being easy to detect (IGC); under the assumption that they do fall into different connected components.

An alternative to the max_{taken} parameter would be to generate hypotheses for all cliques while discarding hypotheses whose poses are similar to those previously generated. However, this would require the definition of a similarity measure among different poses and would incur in an additional computational overhead to compare the current pose with those previously generated.

3.2.3 Evaluation

The aim of this section is to provide an initial evaluation of different correspondence grouping methods. In particular, we consider the following methods:

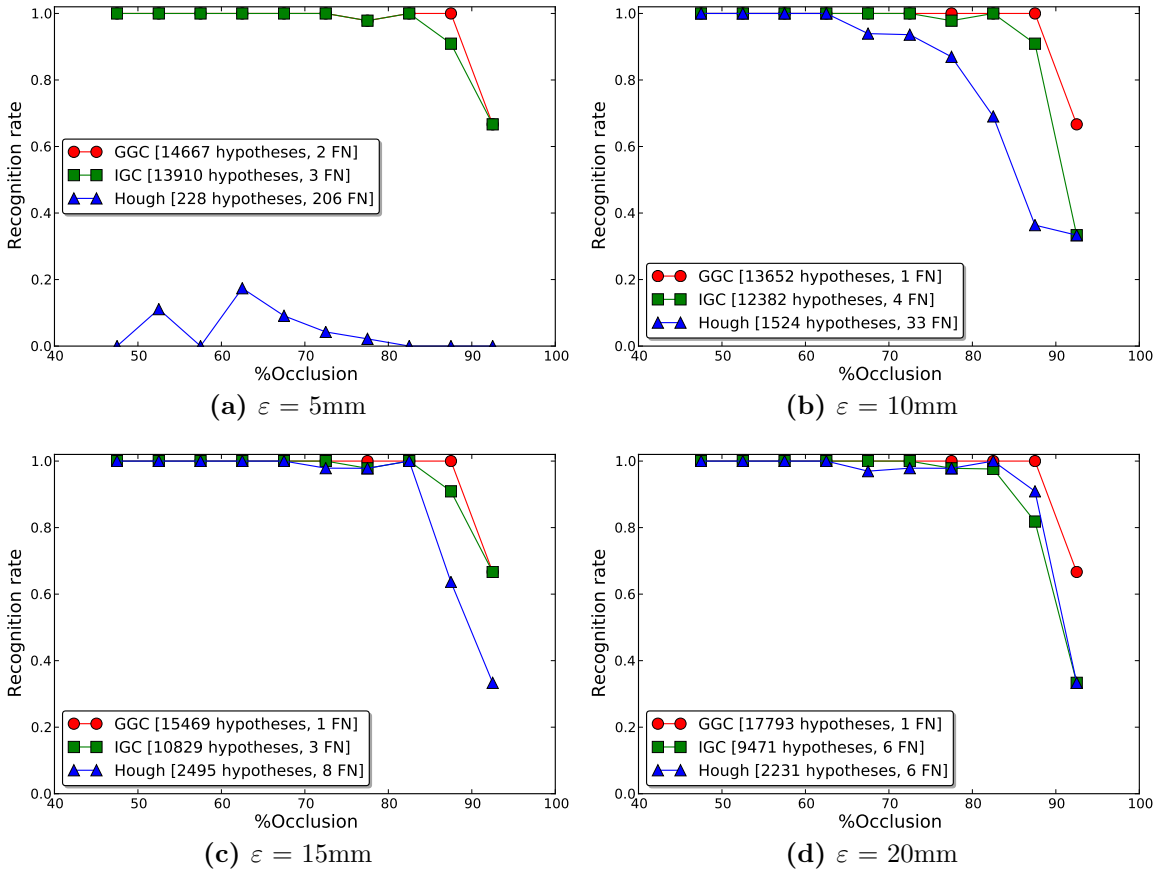
- Hough correspondence grouping [61]
- Iterative Geometric Consistency Grouping (IGC) [4]
- Graph-based Geometric Consistency Grouping (GGC)

and evaluate them in the context of object recognition. Given a set of correspondences, $\mathcal{C}$, between a scene and a set of models, the generated object hypotheses, $\mathcal{H}$, are evaluated against ground truth data. Specifically, for each correspondence grouping algorithm, the ability to generate the correct hypotheses is considered regardless of the amount of generated hypotheses. The assumption here is that in practice, the generated hypotheses are verified by an hypothesis verification stage able to keep correct hypotheses while rejecting those that are wrong. In other words, the maximum recall is evaluated regardless of precision. An hypothesis is considered to be correct if its translation and rotation error are smaller than $1cm$ and $10°$, respectively. These small ranges ensure that the pose refinement stage based on ICP will converge.

Two 3D object recognition benchmark datasets are used for the evaluation. The correspondence set $\mathcal{C}$ is generated by matching SHOT descriptors extracted respectively from the scene and rendered views of the models. SHOT descriptors are computed at keypoints obtained by means of uniform sampling. The experiment

is designed to evaluate the robustness of the methods to occlusion and clutter by fixing the minimum consensus size, τ_{CC}, to 3 (the theoretical minimum to estimate a transformation) while varying the threshold ε controlling the inaccuracy between two correspondences to form consensus.

Figure 3.3 shows the results for the Laser Scanner¶ Dataset for the different grouping methods while varying the ε parameter and keeping the minimum consensus size to 3. Figure 3.4 presents the results on two datasets (Laser Scanner¶ and Queen's) with the best performing ε for each of the methods. We can observe that *GGC* is the best performing method with a single false negative on the whole dataset (*rhino* instance with 93% occlusion). The datasets used within this section are presented in Section 5.1.



(a) $\varepsilon = 5$mm **(b)** $\varepsilon = 10$mm

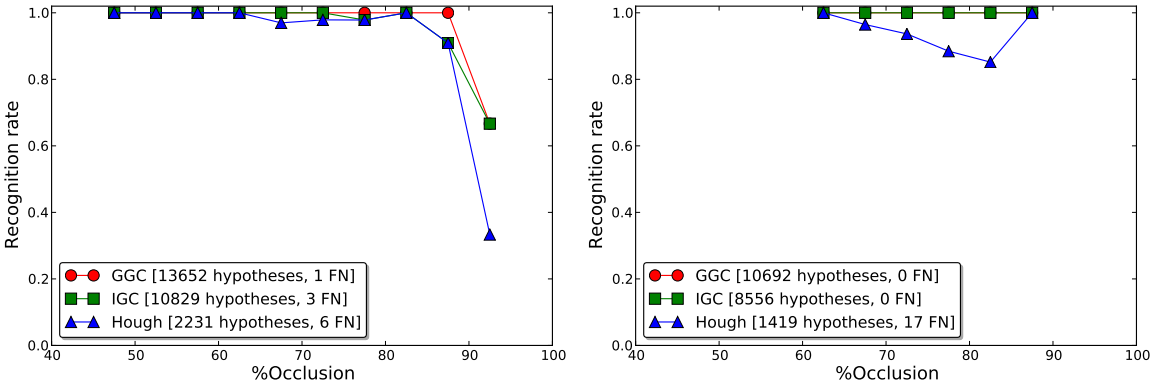
(c) $\varepsilon = 15$mm **(d)** $\varepsilon = 20$mm

Figure 3.3: Recognition versus occlusion results for different ε on the Laser Scanner¶ Dataset. Minimum consensus size fixed to 3.

While the performance of *GGC* is quite invariant to the value of ε, *IGC* and Hough are quite sensitive to it. In particular, Hough requires the accumulator bin size to be relatively big (compared to the other methods, see Figure 3.3-(a,b)) in order to accommodate for inaccuracies in the reference frame, in addition to inaccuracies related to the keypoint locations coming from the uniform sampling strategy. As instances undergo stronger occlusions, the performance on *IGC* and Hough drops.

This is explained for *IGC* due to the greedy selection process of correspondences to seed the consensus sets and for Hough, related to higher inaccuracies of the reference frames due to occluded parts in the reference frame support. Observe as well in Figure 3.3-(d) how the performance of IGC worsens if ε is configured too high while that from GGC remains stable. This is again related to the greedy consensus building as well as the fact that correspondences can only be part of a single consensus set. Please note that the RANSAC outlier rejection requires as well the deployment of a threshold $\varepsilon_{RANSAC} = \varepsilon$ to accommodate for inaccuracies in the keypoint location and therefore is in this case not able to reject wrong correspondence associations to clusters.

Based on these results, it is observable that the correspondence groups provided by Hough are sensitive to the quality and repeatability of the reference frames. For this particular experiment, we use BOARD reference frames [52] with a support radius of $4cm$ (i.e, the same support used for SHOT descriptors). On the other hand, thanks to the oriented reference frames that solve the ambiguities associated with the *GC* constraint, Hough is able to generate a good amount of correct hypotheses while keeping the total number of hypotheses low (about an order of magnitude lower than *GGC*). Please note that in terms of precision and recall Hough is the best performing method. However, when the generated hypotheses are post-processed by means of a suitable verification stage, the additional false hypotheses generated by *IGC* and *GGC* are correctly rejected while maintaining the additional correct hypotheses. This results in an increase of the operating point of the overall recognition system. Section 5.3 provides additional comparisons between GGC and IGC in combination with the hypotheses verification stage.



(a) Laser Scanner Dataset; ε = best **(b)** Queen's Dataset; ε = best

Figure 3.4: Recognition versus occlusion results for the Laser Scanner¶ Dataset and the Queen's dataset. Minimum consensus size fixed to 3. Using the best ε for each method; Laser Scanner¶ Dataset: ε = 10mm for *GGC*, ε = 15mm for *ICC* and ε = 20mm for Hough; Queen's Dataset: ε = 10mm for *GGC* and *ICC* and ε = 30mm for Hough.

Finally, Figure 3.5 shows the effect on the recognition rate when varying the minimum consensus size, τ_{CG}. It can observed that when τ_{CG} increases, the recognition of highly occluded objects becomes more challenging, since not enough consensus is

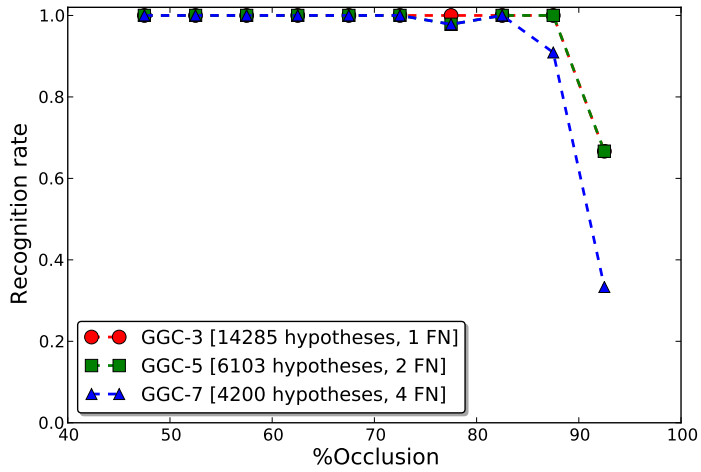


Figure 3.5: Recognition versus occlusion results for different minimum consensus sizes, τ_{CG}, on the Laser Scanner¶ Dataset. $\varepsilon = 15$mm

provided. However, increasing τ_{CG} also results in a dramatic reduction on the number of hypotheses being generated. As usual, this trade-off between number of hypotheses and recognition of objects in borderline situations needs to be accounted during the configuration of the recognition system.

3.3 Global Features

With the availability of object segments given by a suitable segmentation stage, global features aim at finding a distinctive compact representation of the segment. This representation is matched against the training set to yield correspondences between the segment under consideration and the views of the objects learned during the training stage. Within this section, the CVFH and OUR-CVFH descriptors, based on the Viewpoint Feature Histogram (VFH) [57], are presented aiming at solving some of the problems associated with it while maintaining a low computational burden. In particular, the proposed modifications increase (i) robustness to partial occlusions and missing data, (ii) discriminative power and (iii) 6DoF pose estimation capabilities.

3.3.1 VFH: Viewpoint Feature Histogram

For the sake of completeness and to ease the discussion of the proposed features, VFH is shortly reviewed in this section. The VFH descriptor is a compound histogram representing four different angular distributions of surface normals. Let p_c and n_c be the centroids of all surface points and their normals of a given object partial view in the camera coordinate system (with $||\mathsf{n}_c|| = 1$). Then $(\mathsf{u}_i, \mathsf{v}_i, \mathsf{w}_i)$ defines a Darboux

coordinate frame for each point $\mathsf{p_i}$:

$$\begin{aligned} \mathsf{u}_i &= \mathsf{n_c} \\ \mathsf{v}_i &= \frac{\mathsf{p_i} - \mathsf{p_c}}{||\mathsf{p_i} - \mathsf{p_c}||} \times \mathsf{u}_i \\ \mathsf{w}_i &= \mathsf{u}_i \times \mathsf{v}_i \end{aligned} \tag{3.5}$$

The normal angular deviations $\cos(\alpha_i)$, $\cos(\beta_i)$, $\cos(\phi_i)$ and θ_i for each point $\mathsf{p_i}$ and its normal $\mathsf{n_i}$ are given by:

$$\begin{aligned} \cos(\alpha_i) &= \mathsf{v}_i \cdot \mathsf{n_i} \\ \cos(\beta_i) &= \mathsf{n_i} \cdot \frac{\mathsf{p_c}}{||\mathsf{p_c}||} \\ \cos(\phi_i) &= \mathsf{u}_i \cdot \frac{\mathsf{p_i} - \mathsf{p_c}}{||\mathsf{p_i} - \mathsf{p_c}||} \\ \theta_i &= \text{atan2}(\mathsf{w}_i \cdot \mathsf{n_i}, \mathsf{u}_i \cdot \mathsf{n_i}) \end{aligned} \tag{3.6}$$

For $\cos(\alpha_i)$, $\cos(\phi_i)$ and θ_i histograms with 45 bins each are computed and a histogram of 128 bins for $\cos(\beta_i)$, thus the VFH descriptor has 263 dimensions. Using the centroid and average normals over the partial view ($\mathsf{p_c}$ and $\mathsf{n_c}$) to build the Darboux coordinate system, makes VFH sensitive to missing parts of the object caused by partial occlusions, segmentation or sensor artefacts.

These artefacts can result in unstable estimations of the object points and normals centroid (p_c and n_c from Equation (3.5)), thus affecting the resulting VFH histogram and making it unsuitable to match against the corresponding training view that will not present the aforementioned artefacts.

3.3.2 CVFH: Clustered Viewpoint Feature Histogram

The main goal behind CVFH's design is to improve the stability of the coordinate system used during the encoding of surface points within VFH by exploiting structural properties of the object. Specially, in situations where (i) the object undergoes partial occlusions or (ii) its geometry cannot be fully recovered due to sensor nuisances. Additionally, the absolute scale of the object is encoded within the descriptor as well as a distance histogram to increase descriptiveness. Furthermore, CVFH might provide a multivariate representation of the object viewpoint depending on the structural properties of the object.

To overcome the instability of $\mathsf{p_c}$ and $\mathsf{n_c}$, CVFH performs in a first stage a structural analysis of the object's surface $\mathcal{P}_O$ under consideration; aiming at finding *smooth* and *stable* surface parts, $\mathcal{S}_{smooth}$ in $\mathcal{P}_O$. To do so, a smooth region growing algorithm is applied on $\mathcal{S}_{object}$ after removing points with high curvature (caused by noise, object edges or highly non-smooth regions). Each new region is initialized with a random point. A point p_i with normal n_i is added to a region C_k if the region contains a point p_j with normal n_j in the direct neighbourhood of p_i with a similar normal, i.e.,

the following constraint is fulfilled:

$$\exists p_j \in C_k : ||p_i - p_j|| < t_d \wedge n_i \cdot n_j > t_n \tag{3.7}$$

Within this work, t_d is set to three times the resolution of $\mathcal{P}_O$ and t_n to $\cos(2°)$.

For each stable region $s_i \in \mathcal{S}_{smooth}$ and $\mathsf{s}_i \subseteq \mathcal{P}_O$, a CVFH descriptor is computed (thus providing a multivariate representation of the object in cases where multiple stable surfaces are found). It is possible to define a Darboux coordinate system $\mathcal{D} = (\mathsf{u}_i, \mathsf{v}_i, \mathsf{w}_i)$ like in Equation (3.5) but in this case p_c and n_c represent the euclidean centroid and normal centroid of s_i and not that of the whole partial view $\mathcal{P}_O$. Given $\mathcal{D}$ and using Equation (3.6), the normal angular deviations for all points in $\mathcal{P}_O$ can be computed.

Let then $(\alpha, \phi, \theta, \beta)$ represent the normal angular deviations already binned in (45,45,45,128) bins, the CVFH histogram $\mathsf{h}_i \in \mathcal{H}$ is defined as the following concatenation:

$$(\alpha, \phi, \theta, \mathcal{SDC}, \beta) \tag{3.8}$$

where $\mathcal{SDC}$ represents the Shape Distribution Component of CVFH computed as follows:

$$\mathcal{SDC} = \frac{(\mathsf{p}_c - \mathsf{p}_i)^2}{\max((\mathsf{p}_c - \mathsf{p}_i)^2)} \tag{3.9}$$

The number of bins used for this component is again 45 thus making a total size of 308 for CVFH. This component allows to differentiate surfaces that have very similar normal distributions and sizes but their points present a different spatial distribution. Note that the different CVFH histograms obtained from different stable regions are independent from each other and not complementary as they describe the same geometry but encode them differently.

To avoid scale invariance, each bin in CVFH counts the absolute number of points falling in that bin. To reduce ambiguities, the surface resolution (during the recognition stage as well as during the training stage) are normalized by means of a voxel-grid filter to a certain resolution. Because the actual size of the object is given by the 3D sensor, the amount of points for a given view will be the same no matter what the distance to the camera is and therefore is a good approximation of the object's size. [2] Avoiding the normalization step allows us to distinguish between objects of different size but identical shape. It also makes the descriptor more robust to missing parts because only certain bins of the histograms are influenced (those where the missing points would fall, see Figure 3.6). Normalizing the histogram by the total number of points would eventually increase individual bins under the presence of missing data.

A metric for the comparison of histograms in order to handle outliers arising from occlusions or missing data is proposed. Let A and B represent two CVFH histograms,

[2] This does not consider that at a certain distance and farther away, the resolution of the sensor might become too low and thus this approximation would provide an under estimate of the actual size of the object.

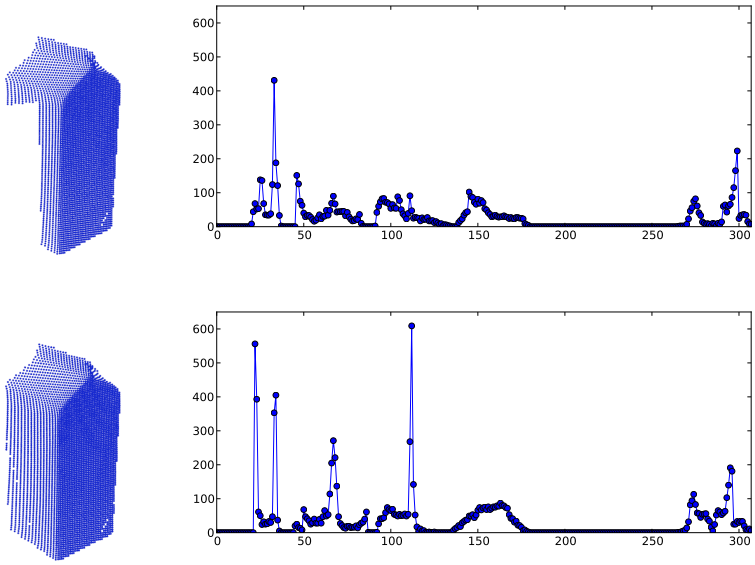


Figure 3.6: The CVFH histograms become additive when the centroids are consistent. *top:* Missing part on the view and the corresponding CVFH signature. *bottom:* Whole view and the corresponding CVFH signature.

their distance is defined as:

$$d(A,B) = 1 - \frac{1 + \sum_{i=1}^{308} \min(A_i, B_i)}{1 + \sum_{i=1}^{308} \max(A_i, B_i)}, \tag{3.10}$$

This metric is not element-wise additive, making it unsuitable for kd-tree search but suitable for hierarchical k-means indexing when the size of the training set increases. In practice, global descriptors can represent large datasets with a few thousands of features allowing to perform linear search in a small amount of time.

The advantages of CVFH are two-fold: (i) the coordinate system is more likely to resemble the one obtained from the training view making the descriptor more stable and (ii) because the set of CVFHs represent a multivariate description of the partial view, partial occlusions are better handled as long as at least one of the stable region is fully visible (ensuring a stable coordinate system). However, CVFH requires an additional stage to estimate a 6DoF pose due to its invariance to rotations on the image plane. In-plane rotation invariance is a common characteristic of most global features. Section 3.3.4 presents two algorithms to estimate the 6DoF pose of an object. Another limitation of CVFH and VFH is the lack of an oriented coordinate system which does not allow to define absolute spatial relations between points on the object's surface. For instance, it is not possible to say if $p_i \in \mathcal{P}_O$ is left or right, above or below, behind or in front of p_c and therefore, the descriptiveness of the shape distribution component, $\mathcal{SDC}$, is very limited.

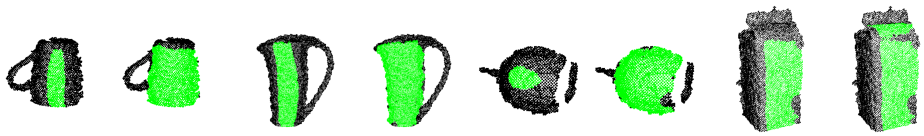


Figure 3.7: Smooth clusters for different surfaces after and before filtering, left and right respectively. Cloud resolution (r) is 3mm, t_n is 0.15, t_c is 0.015 and $t_d = 2.5 * r$

3.3.3 OUR-CVFH: Oriented, Unique and Repeatable CVFH

Aiming at increasing the spatial descriptiveness of the CVFH feature, OUR-CVFH defines an oriented, unique and repeatable reference frame for each stable region. To do that, several ideas from SHOT are borrowed and adapted to the global nature of CVFH. The definition of such reference frames allows to directly estimate 6DoF poses and therefore, removes the need of an additional stage to solve for in-plane rotation (see Section 3.3.4).

Reference frames: Let $\mathcal{S}$ be the surface of the object to be encoded, the first step consists in estimating smooth and continuous clusters $C_i \in \mathcal{S}$ similarly to what CVFH does. First, points whose curvature is higher than a certain t_c threshold are removed from $\mathcal{S}$, yielding $\mathcal{S}^f$. Afterwards, each new cluster is initialized with a random point in $\mathcal{S}^f$ which has not been yet assigned to any cluster. A point p_k with normal n_k is added to a cluster C_i if the cluster contains a point p_j with normal n_j in the direct neighbourhood of p_k with a similar normal, i.e. the following constraint is fulfilled:

$$\exists p_j \in C_i : ||p_h - p_j|| < t_d \wedge n_h \cdot n_j > t_n \tag{3.11}$$

In plain words, the surface $\mathcal{S}^f$ is clustered into smooth and continuous regions, smoothness being controlled by the dot product between the normals of neighbouring points while continuity by their Euclidean distance. Differently to CVFH, the points $p_k \in C_i$ are filtered once more by the angle between n_k and n_i (the average normal of the points in C_i). Figure 3.7 shows the clusters C_i of different surfaces before and after the filtering stage resulting in better shaped clusters for a more robust estimation of the reference frame directions. Each C_i is associated with a pair (c_i, n_i) representing its centroid and average normal. For a specific C_i, the computation of the associated reference frame is as follows:

(i) Compute the eigenvectors of the weighted scatter matrix of the points in C_i, similar to [62]:

$$\mathbf{M} = \frac{1}{\sum_{k \in C_i} (R - d_k)} \sum_{k \in C_i} (R - d_k)(\mathbf{p}_k - \mathbf{c}_i)(\mathbf{p}_k - \mathbf{c}_i)^T \tag{3.12}$$

where $d_k = \|\mathbf{p}_k - \mathbf{c}_i\|_2$ and R is the maximum euclidean distance between any point in C_i and c_i.

(ii) The sign of the eigenvector related to the smallest eigenvalues, $\mathbf{v_3}$, is disambiguated, differently from [62], by taking the direction yielding a positive dot product with n_i and will represent the z-axis of the reference frame. Because the normals of a surface are oriented towards the position of the camera and $\mathbf{v_3}$ is often nearly orthogonal to the surface, the sign disambiguation for this axis is robust.

(iii) At this point, the sign of one axis among the remaining eigenvectors ($\mathbf{v_1}$,$\mathbf{v_2}$) needs to be disambiguated. Let us recall as $\mathbf{v_1}^-$ and $\mathbf{v_2}^-$ as the opposite vectors to ($\mathbf{v_1}$,$\mathbf{v_2}$). Disambiguation is carried out by evaluating the difference of point density between the two hemispheres defined by each eigenvector as in [62]. Conversely to [62], though, the disambiguation deploys the whole surface S (and not just those points used for computing the eigenvectors – this characterizing the *global* aspects of the reference frame) and weights each point k according to their distance to c_i. For example, the sign of $\mathbf{v_1}$ is established as follows (analogously for $\mathbf{v_2}$):

$$S^+_{\mathbf{v_1}} = \sum_{k \in S} \|(\mathbf{p}_k - \mathbf{c}_i) \cdot \mathbf{v_1}\| \cdot ((\mathbf{p}_k - \mathbf{c}_i) \cdot \mathbf{v_1} \geq 0) \tag{3.13}$$

$$S^-_{\mathbf{v_1}} = \sum_{k \in S} \|(\mathbf{p}_k - \mathbf{c}_i) \cdot \mathbf{v_1}\| \cdot \big((\mathbf{p}_k - \mathbf{c}_i) \cdot \mathbf{v_1}^- > 0\big) \tag{3.14}$$

$$\mathbf{v_1} = \begin{cases} \mathbf{v_1}, & |S^+_{\mathbf{v_1}}| \geq |S^-_{\mathbf{v_1}}| \\ \mathbf{v_1}^-, & \text{otherwise} \end{cases} \tag{3.15}$$

For each of the two eigenvectors, we also compute a *disambiguation factor* f_1, f_2:

$$f_i = \frac{min(|S^-_{\mathbf{v_i}}|\ ,\ |S^+_{\mathbf{v_i}}|)}{max(|S^-_{\mathbf{v_i}}|\ ,\ |S^+_{\mathbf{v_i}}|)},\ i = 1, 2 \tag{3.16}$$

This factor ranges in $[0, 1]$, 0 representing perfect disambiguation while 1 representing complete ambiguity.

(iv) Among $\mathbf{v_1}$,$\mathbf{v_2}$, the one with lower disambiguation factor (f_1,f_2) is chosen as the x-axis of the reference frame, since the lower this factor, the less ambiguous the choice of the sign of the eigenvector.

(v) The final y-axis is obtained as $x \times z$.

Unfortunately, in some specific situations the disambiguation is not robust. For example, when both eigenvectors report a similar disambiguation factor, we need to generate two RFs, one using $\mathbf{v_1}$ as the x-axis and the other using $\mathbf{v_2}$. The most challenging case occurs when f_1 and f_2 are similar and both close to 1. In this case, four different reference frames ought to be generated, including both eigenvectors, each encompassing both signs.

Descriptor: So far, for a specific surface $\mathcal{S}$ we have computed N triplets (c_i, n_i, RF_i) obtained from the smooth clustering and the reference frame computation. For the surface description we extend CVFH in the following way: first, c_i and n_i are used to compute the first three components of CVFH and the viewpoint component as presented in [1]. The viewpoint component is however encoded using 64 bins instead of the original 128. Since normals are always pointing towards the sensor position,

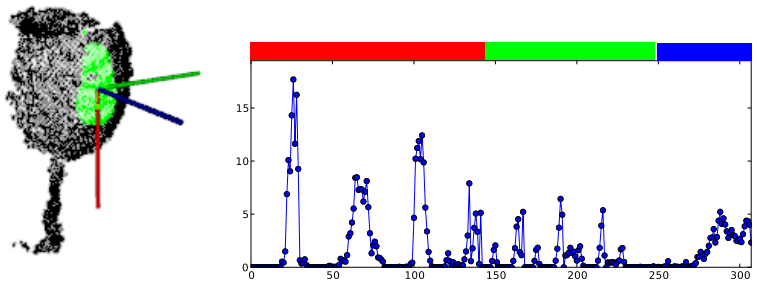


Figure 3.8: Left: Point cloud (black) of a wine glass with associated C_i (green) and the reference frame. Right: The resulting OUR-CVFH histogram. Red and blue bins represent the normal distributions (145 bins) and viewpoint component of CVFH (64 bins). Green bins are the 8 spatial distributions obtained from the points in each octant (104 bins) and the centroid of C_i.

their dot product with the central view direction is ensured to be in the range $[0, 1]$ and therefore there is no need to reserve histogram space for the rest of the range.

The fourth component of CVFH is completely removed and instead the surface $\mathcal{S}$ is spatially described by means of the computed RF_i. To perform this, $\mathcal{S}$ is rotated and translated so that the RF_i is aligned with the x, y, z axes of the original coordinate system of $\mathcal{S}$ and centred at c_i. After the transformation, the points in $\mathcal{S}$ can be easily divided into the 8 octants naturally defined by the signed axes (x^-, y^-, z^-) ... (x^+, y^-, z^-) ... (x^+, y^+, z^+). Additionally, in order to account for perturbations on RF_i due to noise or partially missing parts, interpolation is performed between neighboring octants by associating to each point p_k eight weights, each referred to one octant. The weights are computed by placing three 1-dimensional Gaussian functions over each axis centred at c_i and with σ = 1cm, which are combined by means of weight multiplication. Finally, the weights associated with p_k are added to all 8 histograms, its index in each histogram being selected as $\frac{c_i}{R}$, where R is the maximum distance between any point in $\mathcal{S}$ and c_i. The total size of the descriptor is $45 * 3 + 8 * 13 + 64 = 303$ bins. In Figure 3.8-(b) the OUR-CVFH histogram of a wine glass is reported.

OUR-CVFH: Multi-resolution and color extension

In [5], we proposed two modifications to the OUR-CVFH descriptor aiming at improving:

- The discriminative description of the object surface through the addition of color information provided by RGB-D sensors.
- The repeatability of the reference frames under sensor nuisances affecting the smooth clustering stage.

Color: Taking advantage of the reference frame computed by OUR-CVFH, eight color distributions are computed in addition to the aforementioned shape distributions. The points used to compute each color distribution are obtained by the natural

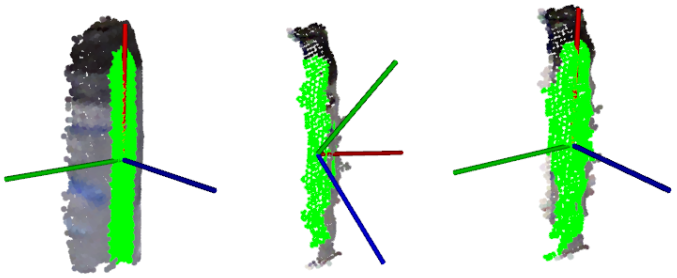


Figure 3.9: Left: training view of an object and its associated reference frame (RF). Middle and Right: scene segment relative to the same object with two associated RFs, yielded by two different clustering parametrizations. Despite the amount of noise and missing points, the RF on the right is repeatable enough to provide a correct match.

division defined by the octants of the reference frame. Each color distribution is obtained from the YUV values associated with each point and binned into a $2 \times 8 \times 8$ grid. A coarser binning for the Y channel with respect to U and V is desired in order to increase robustness with respect to illumination changes. Similar to the L1-shape distributions, a tri-linear interpolation is applied on the color distributions to account for small perturbations in the RF. The 8 color distributions are appended at the end of the OUR-CVFH histogram resulting in a feature dimensionality of $303 + 8 \times 128 = 1327$.

Multi-resolution: OUR-CVFH provides an accurate description and pose estimation thanks to the repeatable reference frames computed on both model views and scene segments. Unfortunately, the repeatability of the reference frame might be compromised due to noisy or missing parts that are often present in data acquired by RGB-D sensors. To overcome these difficulties, a multi-parametric smooth clustering stage is proposed, whereby different clustering instances are run on the same data, each with a different parameter set. Figure 3.9 shows the effect of different parametrization for the smooth clustering stage. Please note, that each clustering instance might yield a different set of smooth regions, this in turn resulting in a different set of RFs and descriptors. This results in a higher number of descriptors representing the same object surface but encoding it differently; e.g., a surface made up by 2 smooth patches might end up being associated with 16 descriptors due to different clustering parametrization as well as ambiguities in the disambiguation stage.

3.3.4 Pose estimation for global features

As briefly mentioned before, global descriptors can be used to directly yield correspondences between object segments in the scene and the *closest* view in the training set; *closest* being defined by the smallest distance between the associated global descriptors. Unfortunately, the closest view does not provide a full 6DoF pose of the object but a hint on the viewpoint from where the object is being observed. In fact, the translation of the object relative to the camera as well as the in-plane rotation

needs to be find to estimate the full pose. In general, the translation part is solved by aligning two stable points on the matching surfaces (i.e., the centroids of the smooth clusters computed by CVFH or OUR-CVFH). In this section, two methods are presented to estimate the pose of the object in the scene relative to the coordinate system of the corresponding model.

Let $\mathcal{C}$ and $\mathcal{S}$ respectively represent the closest view candidate and the object segment in the scene, both in their respective camera coordinate system. Let $\mathcal{T}_\mathcal{C}$ be the transformation aligning the view $\mathcal{C}$ to the 3D model of the corresponding object $\mathcal{O}$. The transformation $\mathcal{T}_\mathcal{O}$ aligning $\mathcal{O}$ to $\mathcal{S}$ is sought. In practice, because $\mathcal{T}_\mathcal{C}$ is known at training time, a transformation $\mathcal{T}$ is sought such that $\mathcal{T}_\mathcal{O} = \mathcal{T} \cdot {\mathcal{T}_\mathcal{C}}^{-1}$.

CRH: Camera Roll Histogram

The Camera Roll Histogram (CRH) was designed to capture in-plane rotations of the object and in contrast to the global features presented before, it is therefore not invariant to such rotations. The computation is based on the availability of surface normals and computed by taking the angle of the projected normal relative to the up-view vector of the camera on the plane. The normals at each point are projected onto a plane that is orthogonal to the vector given by the camera center and the centroid of the stable region used to compute CVFH. For the projection, we compute a rotation-axis v and a rotation angle θ using Equation (3.17) that transforms the CVFH centroid $\mathsf{p_c}$ to coincide with the camera's z-axis. Since we use an orthographic projection, the projected normals are given by the first two components of the transformed normals n_i.

$$\begin{aligned} v &= \frac{p_c \times z}{||p_c||} \\ \theta &= -\arcsin\left(||v||\right) \end{aligned} \tag{3.17}$$

The histogram contains 90 bins giving an angular resolution of 4 degrees. The number of bins for the CRH is selected from our empirical evaluations to provide a reasonable trade-off between efficiency and accuracy. Due to noise in the input data, we weight the projected normals by their magnitudes. This removes most of the equally distributed noise in the histogram, resulting from unstable projections of normals that are almost parallel to the roll axis of the camera. Figure 3.10 shows two histograms of the same object. The upper one is from the object in upright orientation, whereas the bottom histogram is computed from the object rotated around the roll axis by 44°.

In order to estimate the object's rotation around the roll axis, we need to find an orientation where the two roll histograms match best according to a metric. This can be considered a correlation maximization problem. Therefore, we apply a Discrete Fourier Transform for both histograms, and multiply the complex coefficients of the database view with the complex conjugate coefficients, and perform the inverse transform to compute the cross power spectrum R. The peaks of this spectrum appear at rotation angles that align the two histograms well.

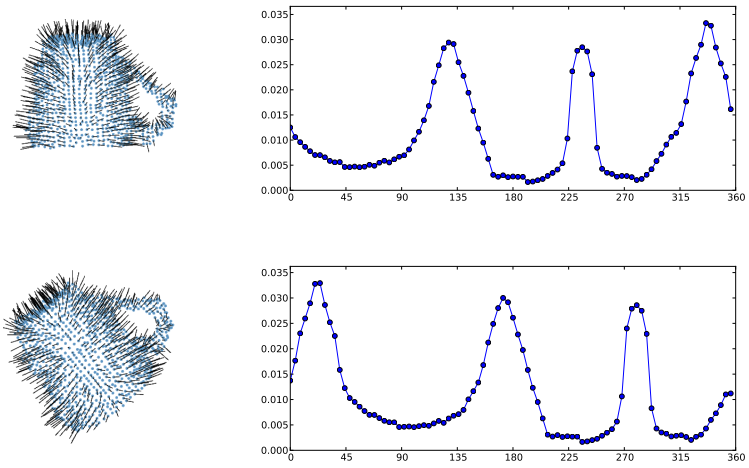


Figure 3.10: The camera roll histograms of the same object in different orientations.

There are cases where the power spectrum of two CRHs can have multiple high peaks due to different kinds of symmetries. Also, partial occlusions or sensor noise might deteriorate the CRH, so it is generally not sufficient to rely solely on the maximal peak in R. In order to select a set of orientations that can be pruned in a subsequent test, we select a minimum threshold t_p for peaks, and add peaks with higher magnitude to the set. We start with the highest peak, adding peaks if their corresponding rotation angles do not fall within a certain distance band t_b of any of the previously added peaks. This ensures that the set of orientations does not contain multiple entries for very similar alignments, but captures local maxima that are distributed over the whole set of rotations, if they indicate a good alignment. In our experiments, we set $t_b = 12°$ and chose a relatively high value for t_p in order to keep the size of the rotation set small. We found a value of $t_p = 0.9 * \max(R)$ to yield a low number of peaks - typically up to 5 peaks - while still capturing corner cases.

Being θ_{CRH} the angle obtained by maximizing the correlation between the corresponding CRH histograms (one from $\mathcal{S}$ and one from $\mathcal{C}$); $p_c^{\mathcal{C}}$ and $p_c^{\mathcal{S}}$ the corresponding centroids of the smooth surfaces, the pose $\mathcal{T}$ is obtained by the following steps:

1. Transform $\mathcal{C}$ applying Equation (3.17).
2. Then, rotate by θ_{CRH} around the roll axis of the camera.
3. Finally, translate such that $p_c^{\mathcal{C}}$ aligns with $p_c^{\mathcal{S}}$.

OUR-CVFH reference frame

By means of the reference frame obtained during the OUR-CVFH computation, the full 6DoF pose of the model in the scene is directly solved by aligning the reference frames of the matching descriptors. Let $\mathcal{T}_{\mathcal{S}}$ be the transformation computed during OUR-CVFH so that the RF_i is aligned with the x, y, z axes of the original coordinate

system of $\mathcal{S}$ and centred at c_i (the centroid of the smooth cluster). $\mathcal{T}_{\mathcal{O}}$ is symmetrically obtained for the object's surface. Then, the pose $\mathcal{T}$ of the object in the scene is $\mathcal{T}_S^{-1}\mathcal{T}_{\mathcal{O}}$.

Observe how the availability of an oriented and repeatable reference frame greatly simplifies the recovery of the pose associated with the object in the scene. In addition, the pose recovered by means of reference frame alignment is more accurate than that provided by CRH and reduces the computational complexity associated with this stage.

3.3.5 Evaluation

Aiming at evaluating the performance of the different global features as well as their pose estimation capabilities, a subset of the Kinect dataset (see Section 5.1.3) was selected by removing scenes showcasing highly occluded object instances (not suited for global features as previously mentioned).

First, we evaluate the performance of the different descriptors regarding object recognition and ignore pose estimation. This experiment allows us to evaluate the distinctiveness of each descriptor independently from the other pipeline stages. One single run is performed over the whole dataset retrieving the first 15 nearest neighbours in the descriptor space. An object is considered to be correctly recognized if the selected id matches that of the ground truth. The rank where the correct id is found is saved and results are presented in Figure 3.11 in form of accumulated recognition rate vs rank. For CVFH and OUR-CVFH variants, histograms were compared by means of the distance metric in Equation (3.10). For VFH [57] and ESF [67] we evaluated different metrics — $L1$, $L2$ and χ^2. VFH performed the best with χ^2 and ESF with $L2$ (both depicted in Figure 3.11). Figure 3.11-(a) highlights the importance of an oriented reference frame for a distinctive description of the objects as OUR-CVFH clearly outperforms the compared descriptors (especially when a low number of candidates is retrieved).

We experimentally observed (see Figure 3.12) that OUR-CVFH recognition capabilities decreased as the distance from the camera of the object to be recognized increased. Because OUR-CVFH relies on a common resolution between models and scene data to incorporate the object size in the descriptor, far away from the camera, the Kinect resolution is lower than 3mm (which is the models' resolution), this violating the aforementioned assumption regarding a common resolution between models and scene. To overcome this issue, we add a preprocessing step during recognition where the segmented object surface is up-sampled by means of uniformly sampling the Moving Least Squares (MLS) plane computed at each original point[8]. This increases the point density of the surface which afterwards is down-sampled to the desired 3mm resolution.

In a second experiment, we compare the 6DoF pose estimation capabilities of the reference frame associated with OUR-CVFH and CRH within the proposed object recognition pipeline. To this aim, we select the best performing descriptor from Figure 3.11, i.e, *OUR-CVFH MLS up-sampling*. The first 10 candidates are retrieved and

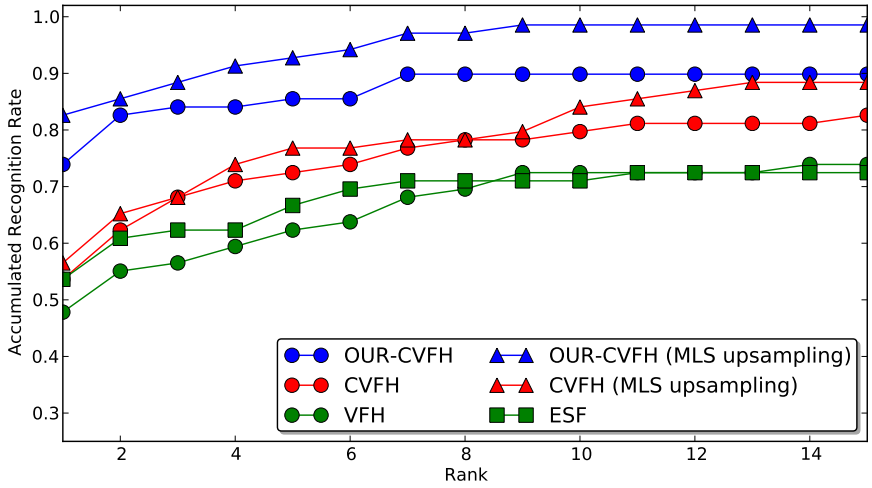


Figure 3.11: Accumulated Recognition Rate for all scenes in the dataset.

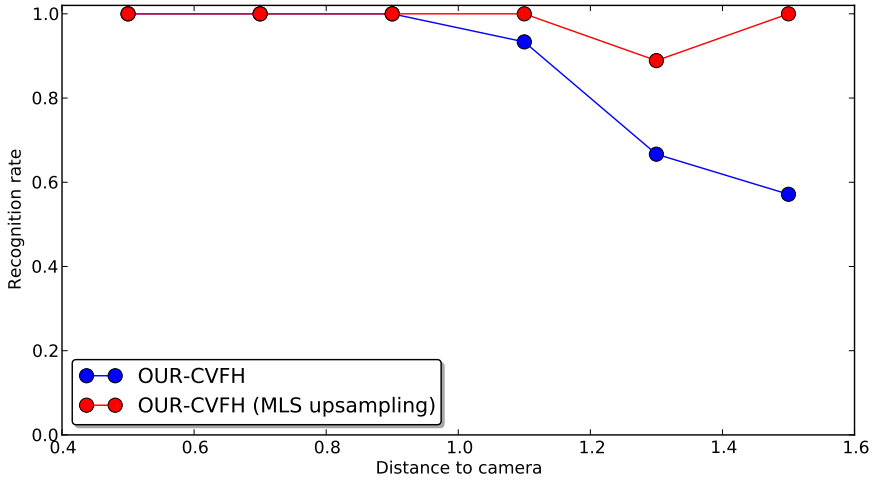


Figure 3.12: Recognition rate relative to sensor distance (computed as the distance from the camera to the centroid of the segmented object.

	#correct_id			#correct_pose			time (s)		
ICP iterations:	0	10	30	0	10	30	0	10	30
OUR-CVFH RF	62	64	66	57	61	63	28.1	48.5	79.2
CRH	49	61	61	35	53	57	42.0	61.3	129.0
Difference:	+13	+3	+5	+22	+8	+6	-13.9	-12.8	-49.8

Table 3.1: Results yielded by the proposed pipeline and OUR-CVFH with MLS upsampling at different ICP iterations (0,10,30), comparing pose estimation yielded by OUR-CVFH RF and by CRH.

their pose independently estimated with the OUR-CVFH reference frame alignment and CRH. Results are presented in Table 3.1 where the candidates pose is refined

with 0, 10 and 30 ICP iterations. Table 3.1 clearly shows the superiority of the OUR-CVFH reference frame alignment over CRH, this being even more notable when ICP refinement is not performed.

3.4 Proposed recognition pipeline

Aiming at exploiting the different strengths provided by different recognition paradigms, the proposed recognition system within this thesis is equipped with three different recognition pipelines (as depicted in Figure 3.13). In the most generic case (when color and 3D information are available), two local and one global recognition pipelines are deployed. The deployment of two local pipelines (one based on 3D shape and the other on texture information) aims at taking advantage of the different data modalities available on scenes captured with recent RGB-D sensors. The rest of this section reviews in detail some of the peculiarities of the proposed system.

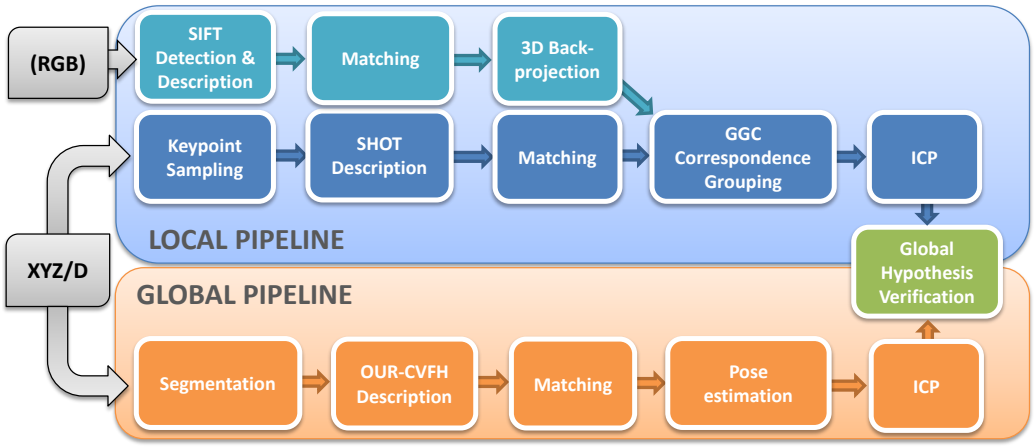


Figure 3.13: Example configuration of a 3D Object Recognition algorithm based on three different recognition pipelines which are then merged together at the Hypothesis Verification stage. In particular, local correspondences coming from the 2D (SIFT) and 3D (SHOT) local pipeline are merged together at the Correspondence Grouping stage trying to increase the desired consensus between scene and model.

3.4.1 Input data

By relying on generic point clouds, our approach does not make particular assumption concerning the characteristics of the input data being processed. Models can be provided either as 3D meshes, point clouds or range maps, either as a collection of views of the same 3D model or as a fully registered 3D model. In case models are provided as registered instead than as a collection of views, during a pre-processing step they are transformed into a set of rendered views by placing a virtual camera on each vertex of a tessellated sphere centred on the model centroid.

As typically available in most application scenarios, each scene is represented by a range map or a point cloud obtained from a single viewpoint. The object recognition pipeline thus processes point clouds associated to views for what concerns both the models as well as the scenes. The proposed approach is also able to handle data in the form of RGB-D frames.

3.4.2 Local pipeline

Regarding the local pipeline, point-to-point correspondences between the models and the scene are obtained by means of SIFT [40] and SHOT [62] features (briefly reviewed below). The SHOT descriptor is computed at each keypoint over a support size specified by radius σ_d, representing the surface around each keypoint that needs to be taken into account when describing the keypoint. As for SHOT parameter values, we have used those originally proposed in [62]. Keypoints are extracted at uniformly sampled positions on the surface of models and scene, parameter σ_s being the sampling distance.

In case RGB-D data is available, SIFT keypoints and descriptors are computed and back-projected on the 3D point cloud by means of the 3D information associated with each RGB pixel: obviously, keypoints detected at depth values being invalid on the range image are discarded. This yields an additional set of 3D keypoints with associated descriptors relying on a different cue with respect to that of 3D local descriptors, i.e. appearance texture as opposed to 3D shape.

After matching scene and models descriptors, SIFT and SHOT correspondences are merged into a unique set before the correspondence grouping stage, so that the clustering algorithm therein can determine correspondence subsets by seamlessly relying on both cues. This stage relies on the Graph-based Geometric Consistency (GGC) grouping algorithm proposed in Section 3.2.2.

To handle the case of multiple instances of the same model within the same scene, each scene descriptor is matched, via fast indexing (i.e., randomized kd-tree [46]), against all models descriptors (and not vice-versa). We explicitly avoid using a matching threshold to reject weak correspondences, given the ad-hoc choice of such thresholds and their strong dependency to the metric being used. Furthermore, notice that a single kd-tree is built including all model descriptors, instead than one kd-tree per model: although possibly increasing the match ambiguities, this approach allows to decrease the complexity of the algorithm with respect to the number of models from linear to sub-linear. This allows to easily scale up to a high number of models without losing computational efficiency. In the following, a brief description of the local features deployed within the proposed system is provided.

2D (SIFT)

The Scale Invariant Feature Transform (SIFT) was proposed by Lowe in [40] and has since been the most popular way of detecting and describing salient image regions. Because of its popularity, several efficient implementations of the method exist (some

of them taking advantages of GPUs), therefore, making SIFT a good candidate for the deployment of image features.

In a nutshell, the method detects keypoints on image locations defined by maxima and minima of the result of difference of Gaussian function applied in scale space to a series of smoothed and re-sampled versions of the image. Some of these keypoints are removed due to poor contrast or being poorly localized along an edge, the rest get assigned a dominant orientation. The location of the keypoint together with its scale and orientation, define a repeatable 2D local coordinate system which is used during a description stage to create an histogram providing invariance to translation, scale and rotation. The histogram is obtained by sampling the image gradients around the keypoint location at the appropriate scale which get summarized (after being transformed using the dominant orientation) into a $4x4$ grid of 8 orientations histograms (thus, the resulting 128 descriptor size). Finally, the descriptors undergo certain operations to increase robustness against illumination changes. For further details, the reader is referred to [39].

SIFT has been extensively applied for the task of object recognition. It is well known to perform well for textured objects, however, its performance decreases rapidly when texture-less objects ought to be detected (i.e., see [19]). Therefore, SIFT is deployed within this work aiming at the detection of textured objects whenever RGB data is available.

3D (SHOT)

Signature of Histograms of OrienTations (SHOT) was originally proposed by Tombari et al. in [62]. The SHOT descriptor encodes a signature of histograms representing topological traits, making it invariant to rotation and translation and robust to noise and clutter. Invariance is obtained by means of an oriented and unique local reference frame coming from the eigenvalue decomposition analysis of the surface falling within the support of the descriptor. The authors emphasize the importance of disambiguating the signs of the eigenvectors (based on the geometrical traits of the support) for an increased repeatability.

The descriptor for a given keypoint is formed by computing local histograms incorporating geometric information of point locations within a spherical support structure. For each spherical grid sector, a 1-dimensional histogram is constructed by accumulating point counts of the angle between the normal of the keypoint and the normal of each point belong to the spherical support structure. The final descriptor is formed by orderly juxtaposing all histograms together according to the local reference frame. Discrete quantization of the sphere introduces a boundary affect, when used in combination with histograms, resulting in abrupt changes from one histogram bin to another. Therefore, quadrilinear interpolation is applied to each accumulated element, resulting in an evenly distribution into adjacent histogram bins. Finally, for better robustness towards point density variations, the descriptor is L_∞-normalized. The dimensionality of the signature is 352.

3.4.3 Global pipeline

The global pipeline deployed within the proposed system is based on the OUR-CVFH descriptor. When color information is available, the OUR-CVFH descriptor is extended to encode color information as previously presented in Section 3.3.3. Features extracted in the scene are then matched against the training database by performing a k-NN linear search using the metric presented in Equation (3.10). After scene segments have been associated with the respective k closest training views, each correspondence is used to estimate the pose of the objects using the OUR-CVFH reference frame alignment technique outlined in Section 3.3.4. Regarding the segmentation of the scene required for the application of global features, the recognition framework is equipped with two different methods reviewed below.

Object segmentation

Object segmentation aims at the discovery of pixel/point clusters that represent single object instances. In general, the problem is hard to solve unless several assumptions are made or appropriate object models are available. A common assumption in robotics considers objects to be standing on planar surfaces (e.g. tables, floor, shelve, etc.) as well as their surroundings free of clutter [60, 6, 1, 67, 57, 5]. Thus allowing to define simple segmentation methods based on planar structures detection followed by a clustering stage on the remaining points aimed at grouping points that are close to each other in space. Because of their simplicity, such methods can be deployed on real-time applications, however, they usually suffer from under-segmentation in common situations where different objects are too close to each other or stacked on a pile.

A good representative of such methods was proposed in [27]. It is a simple but highly efficient two step strategy: (i) multi-plane segmentation of the scene and (ii) connected component clustering of points not detected by the first plane detection stage[3]. To efficiently compute planar regions in a scene, it uses a connected components strategy where neighbouring pixels are considered to be in the same component (planar region in this case) if the dot product of their normals and the euclidean distance between the points are within a certain range. The found planar regions are further analysed to merge regions that share the same planar model and were not detected during the first stage due to the constrained 4 neighbourhood search. The second step performs similarly to the first one, and groups points (without taking into consideration the points belonging to the detected planes) in the same component if their euclidean distance is smaller than τ. The resulting components form the object clusters provided to the recognition pipeline. Such a segmentation strategy assumes that the objects to be recognized will lie on a planar surface and that points belonging to different objects are at least two pixel away in a Manhattan world or farther away than τ. For future reference, we will refer to this method as *MPS*.

[3]Only planes with a certain amount of inliers are considered to provide enough support.

Recently, segmentation of RGB-D images has been extensively pursuit in robotics in order to overcome the limitations of the previous methods. For instance, Richtsfeld et al. [55] pre-segment RGB-D data using a recursive normal clustering approach to extract continuous surface patches before planes and B-spline surfaces are fitted, generating parametric models of the patches. Model selection with Minimum Description Length (MDL) chooses, in a merging procedure, whether a plane or a B-spline model fits better to the patches and delivers the best model representation for a given point cloud. Relations between parametric models can be found by taking into account the principles of perceptual organization. Support vector machines (SVM) are learning this principles during a training period that avoids the reduction of the segmentation framework to model matching. Finally a graph is built, consisting of surface models as nodes and predictions from the SVM's as edges, and a globally optimal segmentation solution can be found even if single predictions are wrong.

3.4.4 Pose refinement

If desired, the pose of the hypotheses generated by the recognition pipelines can be refined by means of ICP. Within our framework and following [26], a fast ICP based on model distance transforms for the nearest neighbour correspondence problem is executed, followed by a few standard ICP iterations yielding the final pose associated with the hypothesis.

3.5 Summary

Throughout this chapter, an overview of different recognition paradigms have been presented. Special emphasis has been given to highlight the strengths and weaknesses of the different paradigms, showing that to some extend, the different paradigms present mutually exclusive strengths and are therefore suited to be deployed in parallel, effectively increasing the range of scenarios on which a recognition system like the one proposed in thesis can be successfully applied.

To this end, the proposed recognition system deploys three different pipelines. The first two, belonging to the local paradigm, are deployed to enable recognition under clutter and occlusion. On one hand, the local pipeline based on SIFT features aims at the recognition of textured objects by exploiting the color image provided by RGB-D sensors. On the other hand, the pipeline based on SHOT features is suited to recognize objects presenting distinctive geometrical traits. Finally, the third pipeline based on the global OUR-CVFH feature is deployed with the goal of recognizing uniform objects (in terms of texture or shape) in situations where segmentation is feasible (to handle clutter) and the objects do not present strong occlusions.

Chapter 4

Global Hypothesis Verification (GHV)

In previous chapters, we have addressed the problems of (i) creating a model library $\mathbf{M}$ from partial views of objects of interest and (ii) the generation of object hypotheses $\mathcal{H} = \{h_1, \cdots, h_n\}$ in a scene $\mathcal{S}$ by means of object recognition methods. Unfortunately, the set of hypotheses $\mathcal{H}$ usually includes *wrong* hypotheses that need to be discarded as well as *good* hypotheses that we would like to be included in the final result. The presence of wrong hypotheses is specially problematic when the recognition pipelines are executed with loose parameters in order to be able to recognize objects in difficult situations (i.e, the object undergoing strong occlusions).

Aiming at reducing the amount of erroneous responses, recognition systems are usually equipped with an hypothesis verification (HV) stage responsible for the rejection of wrong or duplicated object hypotheses. In particular, the HV stage is understood as the process of selecting a subset of hypotheses in $\mathcal{H}$ that fulfil certain conditions. Conversely to other recognition stages, the availability of object poses at this stage enables the definition of powerful cues based on the direct comparison between the recognized models (hypotheses) and the scene under consideration. For example, a typical condition deployed during the HV stage is based on the amount of overlap between a specific hypotheses and the scene. A high overlap provides strong evidence that the hypotheses is correct and therefore, it should be selected. In addition to the simple overlap condition, additional cues (i.e, color comparison if available, hypotheses outliers, etc.) can be deployed aiming at increasing the discriminative (between *good* and *wrong* hypotheses) power of the HV stage.

Regardless of the underlying cues deployed during this stage, the most common verification paradigm is represented by sequentially analysing one hypothesis at a time and deciding whether it is correct or not. Unfortunately, this paradigm disregards the interaction between multiple object hypotheses and thus each decision is entirely based on the *quality* of a specific hypothesis. In addition, it requires the definition of several thresholds (alternatively, a binary classifier can be learned) on which the final rejection/acceptance decision is made upon. In order to handle (duplicate) hypotheses with common overlapping parts, sequential hypotheses verification methods perform a non-maxima suppression of conflicting hypotheses or remove the affected parts of the scene once an hypothesis is verified.

In order to reduce the amount of hard thresholds involved in the verification stage and to mitigate some of the problems related to the sequential verification paradigm, we proposed in [4] a novel hypothesis verification method. Particular to it is the fact that all hypotheses are simultaneously considered aiming at finding a subset of object hypotheses that provide the best globally consistent representation of the scene. To this end, the verification stage is formalized as a minimization problem over the hypotheses set. In particular, we denote a solution as a set of boolean variables $\mathcal{X} = \{x_0, \cdots, x_n\}$ having the same cardinality as $\mathcal{H}$, with each $x_i \in \mathbb{B} = \{0, 1\}$ indicating whether the corresponding hypothesis $h_i \in \mathcal{H}$ is dismissed/validated (i.e. $x_i = 0/1$). Hence, the *cost* function can be expressed as $\mathfrak{F}(\mathcal{X}) : \mathbb{B}^n \rightarrow \mathbb{R}$, $\mathbb{B}^n$ being the solution space, of cardinality 2^n.

As we might see within this chapter, this allows the definition of powerful geometrical and appearance cues (on the objects hypotheses as well as on the scene) to guide the minimization process (see Section 4.2). Because scenes being recognized are usually not solely composed of objects in the model library but other elements — usually referred to as *clutter* (i.e., planar surfaces or other elements), Section 4.4 shows how the proposed formulation allows the hypotheses set to be extended with planar hypotheses extracted from the scene under consideration. On one hand, this provides a richer understanding of the scene (in terms of planes and object hypotheses) and on the other hand, enables the verification stage to consider the interaction between object and planar hypotheses, effectively providing additional cues that can be seamlessly integrated in the proposed cost function (see Section 4.3). Because the solution space grows exponentially with the number of hypotheses to be verified, Section 4.5 provides an analysis and evaluation of different meta-heuristics to efficiently solve the optimization problem at hand.

4.1 Related work

In [32, 31] using the correspondences supporting a hypothesis as seeds, a set of scene points is grown by iteratively including neighbouring points which lie closer than a pre-defined distance to the transformed model points. If the final set of points is larger than a pre-defined fraction of the number of model points (from one fourth to one third of the number of model points), the hypothesis is selected and ICP is selectively run on the attained set of points in order to refine object's pose. Obviously, one disadvantage of such an approach is that it can not handle levels of occlusions higher than 75%.

The HV method proposed in [44] ranks hypotheses based on the quality of supporting correspondences, so that they are then verified sequentially starting from the highest rank. To verify each hypothesis, after ICP, two terms are evaluated: the former, similarly to [32], is the ratio between the number of model points having a correspondent in the scene and the total number of model points, the latter is the product between this ratio and the quality score of supporting correspondences. This step requires to set three different thresholds. Two additional checks are then

enforced, so as to prune hypotheses based on the number of *outliers* (model points without a correspondent in the scene) as well as on the amount of occlusion generated by the current hypothesis with respect to the remaining scene points. Again, these two additional checks require three thresholds. If an hypothesis gets through each of these steps, it is accepted and its associated scene points are eliminated from the scene, so that they will not be taken into account when the next hypothesis is verified.

In [10], for each model yielding correspondences, the set of hypotheses associated with the model is first pruned by thresholding the number of supporting correspondences. Then, the best hypothesis is chosen based on the overlap area $A(H_{best})$ between the model associated with that hypothesis and the scene, and the initial pose is refined by means of ICP. Finally, the accuracy of the selected hypothesis is given by the ratio $\frac{A(H_{best})}{M_a(H_{best})}$ where $M_a(H_{best})$ is the total visible surface of the model within the bounding box of the scene. The model is said to be present in the scene if its accuracy is above a certain threshold and, upon acceptance, the scene points associated with $A(H_{best})$ are removed.

Papazov and Burschka [50] evaluate how well a model hypothesis fits into the scene by means of an *acceptance function* which takes into account, as a bonus, the number of transformed model points falling within a certain distance from a scene point (*support*) and, as a penalty, the number of model points that would occlude other scene points (i.e. their distance from the closest scene point is above threshold but they lie on the line of sight of a scene point). A hypothesis is accepted by thresholding its support and occlusion sizes. Given the hypotheses fulfilling the requirements set forth by the acceptance function, a conflict graph is built, wherein forks are created every time two hypotheses share a percentage of scene points above a -third- threshold. Surviving hypotheses are then selected by means of a non-maxima suppression step carried over the graph and based on the acceptance function. This approach is the most similar to ours, as, thanks to the conflict graph, interaction between hypotheses is taken into account. Nevertheless, their method is only partially global, since the first stage of the verification still relies on pruning hypotheses one at a time and a *winner-take-all* strategy is used for conflicting hypotheses.

With the advent of RGB-D sensors, providing a range image with associated color information, additional verification stages have been proposed in the literature which take advantage of the multiple modalities available in the data. In particular, Hinterstoisser et al. propose in [26] a simple two step verification process. In more detail, the coarse pose provided by LINEMOD is used to perform an initial validation of the detections by means of the color information of the scene and the model. To this end, points of the object hypotheses with similar color to that of their projection in the scene are counted. The hue value is used to increase robustness against light changes (appropriately handling black and white objects). If more than 30% of the points do not have the expected color value, the hypotheses is rejected. The pose of the remaining hypotheses is then refined by means of ICP and a second verification is carried on by comparing the depth of the hypothesis with the depth of the corresponding points in the scene. Hypotheses that fulfil the depth test are then

accepted, otherwise, rejected.

Xie et al. propose in [68] a verification stage where each segmented cluster in the scene and the associated detections are validated by means of a multi-modal model; including shape, texture and color. Specifically, the authors propose to use a feature-weighted linear stacking (FWLS) approach to train a regression model based on the different modalities to score the specific hypotheses associated with a scene cluster. The feature vector used to train the model is computed as the product pairs between the different modalities and several meta-features; which provide information about the reliability of each modality for the hypothesis being scored (i.e., median color saturation of the image). The best scoring hypotheses is accepted if above a certain threshold. Other hypotheses are rejected.

Relevant to our work but aimed at piecewise surface segmentation on range images, Leonardis et al. proposed in [37] a model selection strategy based on the minimization of a cost function to produce a globally consistent solution. Even though the minimization is formalized in terms of a Quadratic Boolean Problem, the final solution is still attained taking into accounts hypotheses sequentially by means of a *winner-take-all* strategy.

4.2 Cues

This section presents in detail the cues enforced during the optimization of the cost function. In a nutshell, the main cues used within GHV guide the optimization process in order to:

- Maximize the number of *explained* scene points.
- Minimize the number of model outliers: Valid (visible and within the sensor range) model points that do not have a correspondent in the scene.
- Minimize the number of scene points that are simultaneously *explained* by two or more active hypotheses.
- Minimize the amount of clutter generated by active hypotheses. In particular, hypotheses that partially *explain* smooth parts of the scene are penalized.

4.2.1 Occlusion reasoning

Given an hypothesis h_i, model parts not visible in the scene due to self-occlusions or occlusions generated by other scene parts should be removed since they cannot have corresponding scene parts and thus cannot provide consensus for h_i. Therefore, for each hypothesis $h_i \in \mathcal{H}$, a modified version of $\mathcal{M}_{h_i}$ is computed by transforming the model according to $\mathcal{T}_{h_i}$ and removing all occluded points. Hereinafter this new point cloud will be denoted as $\mathcal{M}^v_{h_i}$.

Establishing whether a model point is visible or occluded can be done efficiently based on the range image associated to the scene point cloud, possibly generating

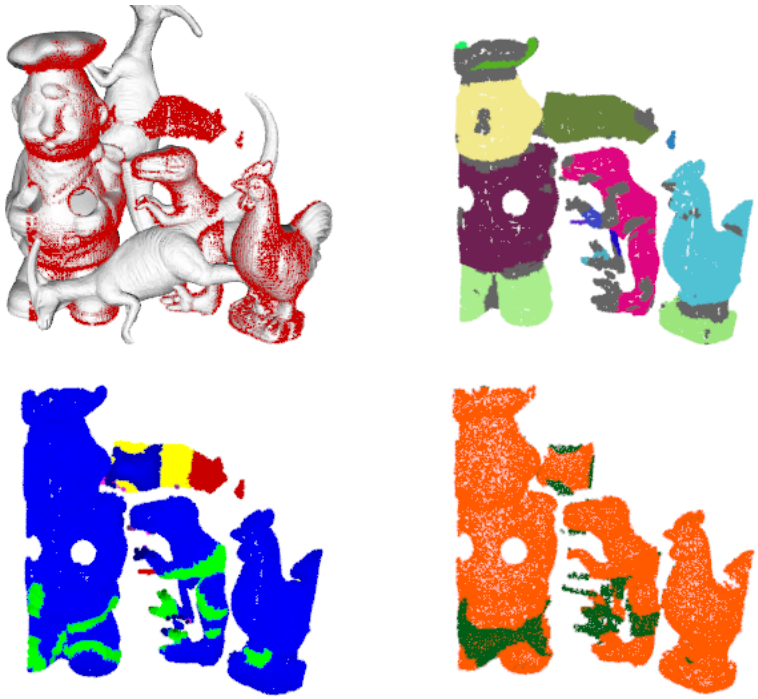


Figure 4.1: Proposed cues enforced during GHV. *Top left*: a solution consisting of a set of active model hypotheses super-imposed onto the scene. *Bottom left*: classification of scene points between explained by a single hypothesis (blue), by multiple hypotheses (green), unexplained (red), cluttered due to region growing (yellow) and to proximity (purple). *Bottom right*: classification of visible model points between inliers (orange) and outliers (green). *Top right*: scene labelling via smooth surface segmentation.

the range image from the point cloud whenever the former is not available directly. Thus, similarly to [50, 44], a point $p \in \mathcal{M}_{h_i}$ is considered visible if its back-projection into the range image falls on a valid pixel and its depth is smaller than that of the pixel. The same reasoning applies to self-occlusions.

4.2.2 Explained points and outliers

Once the set $\mathcal{M}_{h_i}^v$ has been obtained, we want to determine whether these points have a *good* correspondent on the scene, i.e. how well they *explain* scene points. A similar cue is exploited in [32, 44, 11, 50], where model and scene points are associated by hard thresholding the distance between each model point and its closest scene point. Here, we want to refine such approach, by measuring how well each visible model point locally fits the scene. To this aim, we introduce a term, $\delta(p, q)$, which measures how well two points p and q fit each other based on the relative distance as well as on their associated normal directions:

$$\delta(p,q) = \begin{cases} (n_p \circ n_q)\, exp\left(-\frac{\|p-q\|_2^2}{2\rho_e^2}\right), & \|p-q\|_2 \leq \rho_e \\ 0, & elsewhere \end{cases} \tag{4.1}$$

where n_p and n_q are respectively the normals at p and q, and $\circ$ denotes the dot product, which is rounded to 0 whenever negative to avoid negative weights (note that all normals have a consistent orientation based on the position of the sensor). For what concerns the relative distance between p and q, $\delta(p,q)$ is regulated by parameter ρ_e, which intuitively defines the *distance of interaction* between each pair of points (p,q) by taking into account noise and disturbances affecting data (further than ρ_e, p does not interact with q).

To apply Equation (4.1) to an object hypothesis, we define a scene point p as *explained* by an hypothesis h_i if model $\mathcal{M}_{h_i}^v$ has at least one point q for which $\delta(p,q) > 0$, *unexplained* otherwise:

$$\omega_{h_i}(p) = \begin{cases} 1, & \delta\left(p, \mathcal{N}\left(p, \mathcal{M}_{h_i}^v\right)\right) > 0 \\ 0, & elsewhere \end{cases} \tag{4.2}$$

where $\mathcal{N}(p,\mathcal{M})$ is the nearest neighbour of p to be found on model $\mathcal{M}$. The set of all scene points explained by hypothesis h_i according to (4.2) will be hereinafter denoted as $\mathcal{S}_{h_i}$.

Extending this definition from a single hypothesis to a whole solution $\mathcal{X}$, we define a scene point p to be explained by $\mathcal{X}$ if there is at least one model $\mathcal{M}_{h_i}^v$ activated in $\mathcal{X}$ that explains p; this is mathematically represented by term $\Omega_{\mathcal{X}}(p)$, which weights - proportionally to $\delta(p,q)$ - each scene point currently *explained* by solution $\mathcal{X}$:

$$\Omega_{\mathcal{X}}(p) = \max_{i=1..n} \left(x_i \cdot \delta\left(p, \mathcal{N}\left(p, \mathcal{M}_{h_i}^v\right)\right) \right) \tag{4.3}$$

For what concerns models, and analogously to the above definition of *explained* points associated to the scene, we term a model point $p \in \mathcal{M}_{h_i}^v$ an *outlier* for hypothesis h_i if it is not fitted by any scene point according to (4.1), an *inlier* otherwise. Hereinafter, we will denote as Φ_{h_i} the set of outliers for hypothesis h_i and as $|\Phi_{h_i}|$ the cardinality of each such a set.

The amount of explained scene points and outliers are powerful geometrical cues for evaluating the goodness of a solution $\mathcal{X}$ within the GHV framework. In particular, and referring to Equations (4.11) and (4.12): i) the number of explained scene points should be maximized; and ii) the number of outliers associated with all active hypotheses should be minimized.

4.2.3 Multiple assignment

An important cue highlighting the existence of incoherent hypotheses within a solution deals with a surface patch in the scene being fitted by multiple models. According to our definitions, this can be exploited by penalizing scene points explained by two or more hypotheses, as denoted by black points in the two bottom right figures of 4.1-a) and -b) (to be compared with blue points, denoting points explained by a single hypothesis). Thus, given a solution $\mathcal{X}$ and a scene point p, we define a function

$\Lambda_{\mathcal{X}}(p)$

$$\Lambda_{\mathcal{X}}(p) = \begin{cases} \sum_{i=1}^{n} x_i \delta\left(p, \mathcal{N}\left(p, \mathcal{M}_{h_i}^{v}\right)\right), & \sum_{i=1}^{n} x_i \omega_{h_i}(p) > 1 \\ 0, & elsewhere \end{cases} \tag{4.4}$$

which counts the number of conflicting hypotheses with respect to p and according to the definition given in (4.1) and (4.2). Hence, another cue being enforced by the GHV cost function through $\Lambda_{\mathcal{X}(p)}$ is that iii) the number of multiple hypothesis assignments to scene points should be minimized.

4.2.4 Clutter

In many application scenarios not all sensed shapes can be fitted with some known objects model. Exceptions might occur, for instance, in some controlled industrial environments where all the objects making up the scene are known *a priori*. More generally, though, several visible scene parts which do not correspond to any model in the library might locally - and erroneously - fit some model shapes, potentially leading to false detections. Maximizing the number of explained scene points (i.e. cue i)), although useful to increase the number of correct recognitions, nevertheless favours this circumstance. On the other hand, computing the outliers associated with these false positives (cue ii)) might not help, since the parts of the model which do not fit the scene might turn out occluded or outside the field of view of the 3D sensor.

To counterattack the effect of clutter, we devised an approach, inspired by surface-based segmentation [54], aimed at penalizing a hypothesis that locally explains some part of the scene but not nearby points belonging to the same smooth surface patch. This is also useful to penalize hypotheses featuring correct recognition but wrong alignment of the model in the scene. Surface-based segmentation methods are based on the assumption that object surfaces are continuous and smooth. Continuity is usually assessed by density of points in space and smoothness through surface normals. Following this idea, scene segmentation is performed by identifying smooth clusters of points. Each new cluster is initialized with a random point, then it is grown by iteratively including all points p_j lying in its neighbourhood which show a similar normal:

$$||p_i - p_j|| < t_d \wedge n_i \circ n_j > t_n \tag{4.5}$$

At the end of the process, each scene point is associated with a unique label $l(p)$. In top right of Fig. 4.1-a) and -b), we report two examples of scene segmentation.

Hence, given a solution $\mathcal{X}$, likewise in Equation (4.3), we compute a clutter term, $\Upsilon_{\mathcal{X}}(p)$, at each unexplained scene point p, so as to penalize those that are likely to belong to the same surface as nearby points being explained by hypotheses which are activated by solution $\mathcal{X}$:

$$\Upsilon_{\mathcal{X}}(p) = \sum_{i=1}^{n} x_i \cdot \gamma\left(p, \mathcal{N}\left(p, \mathcal{S}_{h_i}\right)\right) \tag{4.6}$$

Analogously to $\delta(p,q)$, we want $\gamma(p,q)$ to weight clutter based on the proximity of p to its nearest neighbour, $q \in \mathcal{E}_{h_i}$, as well as according to the alignment of their surface patches:

$$\gamma(p,q) = \begin{cases} \kappa, & \|p-q\|_2 \leq \rho_c \wedge l(p) = l(q) \\ (n_p \circ n_q) exp\left(\frac{\|p-q\|_2^2}{2\rho_c^2}\right), & \|p-q\|_2 \leq \rho_c \wedge l(p) \neq l(q) \\ 0, & elsewhere \end{cases} \tag{4.7}$$

Radius ρ_c defines the spatial support related to $\gamma(p,q)$ and should be chosen in relation to the expected level of clutter in the scene, while κ is a constant parameter used to penalize unexplained points that have been assigned to the same cluster by the smooth segmentation step. Thanks to the above formulation, incorrect or misaligned active hypotheses, such as the mug in Fig. 4.1-a), cause a significantly valued clutter term $\Upsilon_{\mathcal{X}}$ (e.g. the yellow regions caused by the handle), which will penalize their validation within the global cost function. Therefore, we have derived the last cue: iv) the amount of unexplained scene points close to an active hypothesis according to (4.7) should be minimized.

4.2.5 Extension to RGB-D data

As previously mentioned, several sensors have the capability of acquiring 3D data enhanced with color information, either in the form of point clouds with associated RGB triplets, or as RGB-D data. In either case, this color information is a valuable source for the HV stage to exploit in order to verify hypotheses whose shape accurately matches a surface patch in the scene but differs in color (see Figure 4.2).

Figure 4.2: Object instances with identical shape but different color. A hypothesis verification stage based solely on geometrical cues would be unable to distinguish between the different objects.

The proposed GHV framework is flexible enough to incorporate color information in its cost function: specifically, we propose to do so by extending the definition of *explained points* and *model outliers* to deal with color information, too. Notable

enough, this extension does not require explicitly an organized format of the data (i.e., RGB-D), as it only requires an associated color triplet to each 3D point. The generalization to color comes however at the price of being subject to varying illumination conditions and photometric distortions between model and scene. We propose to mitigate this phenomenon through the use of a color space robust to light changes, as well as by means of a specific tonal registration stage between each model and the scene.

Explained points and outlier cues with color

Given a point $p \in \mathcal{S}$ and its associated color value p_{Lab} (representing the 3D color vector associated to p and expressed in the *Lab* color space), p is considered to be explained by hypothesis h_i according to Equation (4.2) with the additional constraint that at least a model point, $q \in \mathcal{M}_{h_i}^v$ and within the ρ_e-neighbourhood of p, must exist fulfilling:

$$exp\left(-\frac{1}{2}\left(q_{Lab}-p_{Lab}\right)\Sigma_{Lab}^{-1}\left(q_{Lab}-p_{Lab}\right)\right) \geq \rho_l \tag{4.8}$$

where $\rho_l \in [0,1]$ is a user defined parameter indicating the desired color similarity between scene and model points, and the covariance matrix of *Lab* color channels, under the assumption of independence across color channels, is defined as:

$$\Sigma_{Lab} = \begin{bmatrix} \sigma_L & 0 & 0 \\ 0 & \sigma_a & 0 \\ 0 & 0 & \sigma_b \end{bmatrix} \tag{4.9}$$

To simplify the overall amount of parameters and without lack of generality we choose $\sigma_a = \sigma_b$. Symmetrically, any point $q \in \mathcal{M}_{h_i}^v$ that does not simultaneously fulfill Equations (4.8), (4.2) for any point $p \in \mathcal{S}$ is considered to be a model outlier.

These modified definitions of explained points and outliers have several effects in the cost function term associated to a solution $\mathcal{X}$:

- Less scene points will be explained, effectively decreasing $\Omega_{\mathcal{X}}$.
- Hypotheses will have a larger number of outliers due to color dissimilarities, increasing $\lambda \cdot f_{\mathcal{M}}(\mathcal{X})$.
- Hypotheses whose color partially matches part of the scene will result in a significant clutter value due to the reduction of explained scene points (remember that only unexplained scene points can contribute to the clutter term) and will increase $\Upsilon_{\mathcal{X}}$.

Generally speaking, the deployment of color within GHV results in a increase of its discriminative power and, consequently, a reduction of false positives (especially in presence of models characterized by similar shapes and different appearance/texture), although by relying also on color, notable photometric distortions between models and scene could bring in an unwanted reduction of true positives, too.

Tonal registration

While the addition of color in the verification stage results useful to distinguish between model hypotheses and scene points laying on geometrically similar neighbourhoods, the color properties of each point are strongly affected by illumination conditions (see Figure 4.3). Even though the use of color spaces robust to illumination changes mitigate this effect, it is possible to exploit the fact that model hypotheses are aligned with the scene so as to tonally re-map the model color distribution to match that of the corresponding *explained* points on the scene, so as to factor out possible illumination changes between the two. By assuming locally affine photometric distortions, the histogram of the L channel of the model is tonally registered to that of the scene points by means of the Histogram Specification technique[23]. A more refined re-mapping can be obtained by independently specifying the L-channel for each smooth face of the model, so as to take into account of different illumination conditions on different parts of the objects (see, i.e, the box in Figure 4.3).

To this end, during the off-line stage, the models are analysed in order to extract their smooth faces. In particular, the algorithm is based on the supervoxels extraction strategy proposed in [51]. Then, these supervoxels (with their associated normals) are merged together by creating a graph with edges linking a pair of supervoxels if they are adjacent and the angle between their normals is similar. The connected components of the graph yield the smooth faces of the model. During verification stage and prior to the computation of scene explained points and model outliers, the histogram of each smooth model face is specified against the histogram of the corresponding scene points. Finally, the mappings provided by the histogram specifications are used to adapt the luminance values of the model points.

4.3 Cost function

We have so far outlined four cues i)-iv). While i) is aimed at increasing as much as possible the number of recognized model instances (thus TPs and FPs), ii), iii) and iv) try to penalize unlikely hypotheses through geometrical constraints, so as to minimize false detections (FPs). The cost function $\mathfrak{F}$ we are looking for is obtained as the sum of the terms related to the cues that need to be enforced within our optimization framework:

$$\mathfrak{F}(\mathcal{X}) = f_{\mathcal{S}}(\mathcal{X}) + \lambda \cdot f_{\mathcal{M}}(\mathcal{X}) \tag{4.10}$$

where λ is a regularizer, and $f_{\mathcal{S}}$, $f_{\mathcal{M}}$ account, respectively, for cues defined on scene points and model points:

$$f_{\mathcal{S}}(\mathcal{X}) = \sum_{p \in \mathcal{S}} \left(\Lambda_{\mathcal{X}}(p) + \Upsilon_{\mathcal{X}}(p) - \Omega_{\mathcal{X}}(p) \right) \tag{4.11}$$

$$f_{\mathcal{M}}(\mathcal{X}) = \sum_{i=1}^{n} |\Phi_{h_i}| \cdot x_i \tag{4.12}$$

Figure 4.4 shows the evolution of the proposed cues as the optimization proceeds.

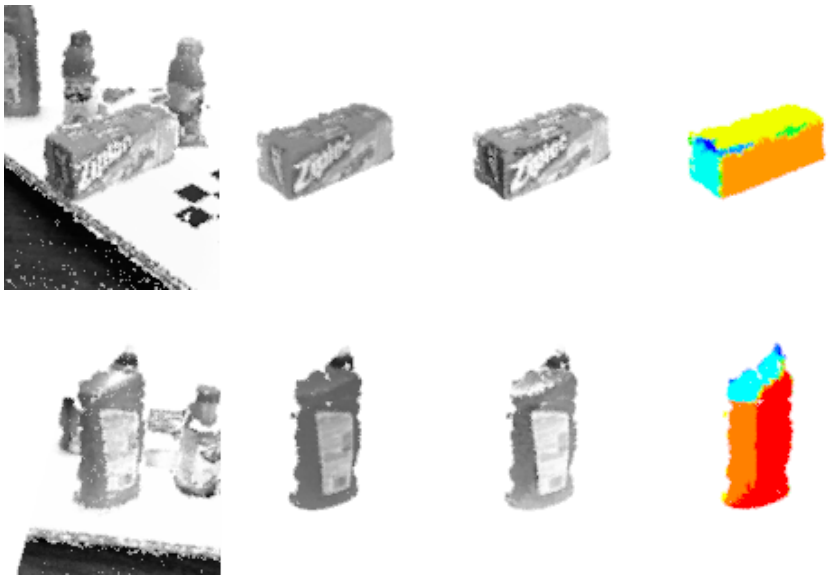


Figure 4.3: Effects of the tonal specification on the color information of two hypotheses. From left to right: relevant part of the scene, hypothesis, hypothesis after tonal specification and visible smooth areas. In all cases, color is represented as a grayscale image obtained from the L channel. Observe how specularities cannot be captured using this model (second row).

4.4 Planar hypotheses extension

A recurring trait of many object recognition scenarios is represented by the presence in the scene of planar structural elements such as tables, ground floor, walls, etc.. These elements, although not part of the database of models being analysed, when present, will easily comprise the majority of points in the scene. Being able to correctly recognize the planar surfaces in the scene, thus, would bring in a reduction in the number of false positives, since all hypotheses associated with such planes could be discarded. Furthermore, the interaction between estimated object and planar hypotheses can be exploited to remove those hypotheses (either associated to objects or planes) which are physically improbable in the current configuration (i.e, an object hypothesis intersecting a plane.)

To this aim, we want to extend the verification stage by explicitly allowing, in the evaluated solution, the presence of specific *planar hypotheses*. Moreover, although not exploited within the scope of this work, this improvement carried out by the GHV approach in terms of knowledge inferred from the analysed scene could lead, as a by-product, to a more complete scene understanding, useful for a variety of applications (e.g. path-planning, high-level human-user interaction, etc.).

4.4.1 Planar hypotheses generation

For the generation of planar hypotheses, two alternatives approaches are deployed depending on the underlying data representing the scene. If the data comes from recent

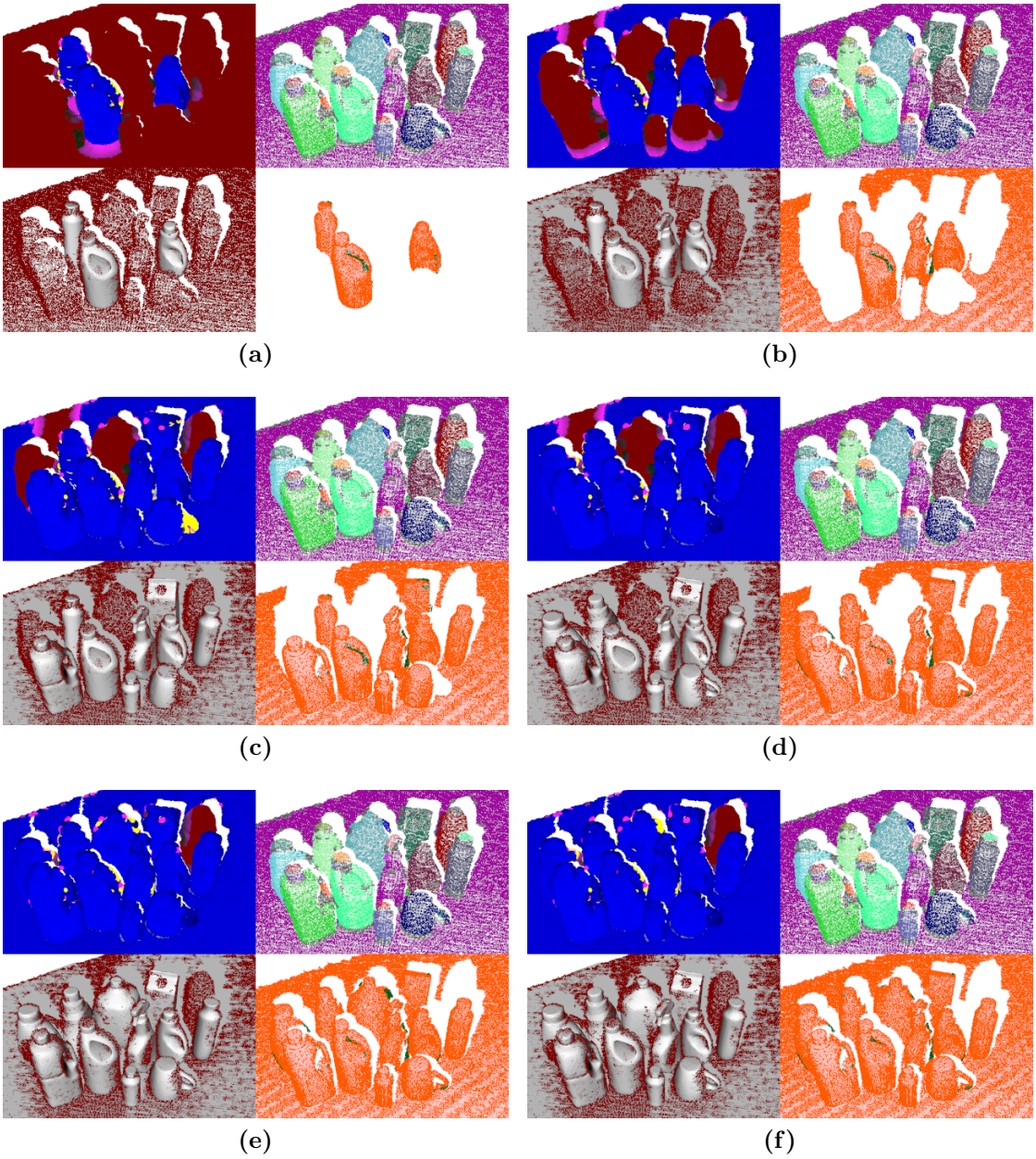


Figure 4.4: Evolution of the proposed cues for GHV at different times as the optimization proceeds. The optimization procedure used to generate this results was Local Search (Hill Climbing) with Replace Active Hypotheses moves activated (see Section 4.5).

RGB-D sensors (providing an *organized* structure of the point cloud), the multi-plane segmentation approach of [27] is used. If the point cloud data is *unorganized*, we follow a simple iterative plane fitting approach based on RANSAC where, after each iteration, the points associated to the dominant plane are removed from the scene. In both cases, each element of the extracted set of planar hypotheses $\mathcal{P} = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_{|\mathcal{P}|}\}$ is represented by the plane coefficients $\boldsymbol{p} = \{n_x, n_y, n_z, d\}$, with an associated set of

scene points $\mathcal{S}_{\boldsymbol{p}}$ as those that held the consensus for plane $\boldsymbol{p}$.

4.4.2 Effects on the hypothesis verification stage

By inserting the extracted planar hypotheses into the GHV framework, the solution space dimensionality of the cost function $\mathfrak{F}$ is increased from n to $n + |\mathcal{P}|$, and each solution $\mathcal{X} = \{x_0, \cdots, x_n, x_{n+1}, \cdots, x_{n+|\mathcal{P}|}\}$ will thus activate a specific configuration of both object and planar hypotheses. By means of the scene points $\mathcal{S}_{\boldsymbol{p}}$ associated to each plane, planar hypotheses can be seamlessly added to the verification framework, where inlier weights are computed from (4.1) by substituting the distance between p and q with the projection of each point in $\mathcal{S}_{\boldsymbol{p}}$ onto $\boldsymbol{p}$. If color information is available, the inlier weight for planar hypotheses is multiplied by ρ_l in order to favour object hypotheses with color over planar hypotheses. It is worth mentioning that, since planar hypotheses are generated via plane fitting and not from explicit models, color information cannot be exploited in this case as there would be no color model to compare with.

Hence, when planar hypotheses are activated, the global cost function $\mathfrak{F}$ is modified by summing to the right-hand term in Equation (4.10) an additional term $f_{\mathcal{P}}$ taking into account penalties derived from the interaction between planar and object hypotheses:

$$\mathfrak{F}(\mathcal{X}) = f_{\mathcal{S}}(\mathcal{X}) + \lambda \cdot f_{\mathcal{M}}(\mathcal{X}) + f_{\mathcal{P}}(\mathcal{X}) \tag{4.13}$$

where $f_{\mathcal{P}}$ is defined as:

$$f_{\mathcal{P}}(\mathcal{X}) = \sum_{i=1}^{n} \sum_{j=1}^{|\mathcal{P}|} \Pi(\boldsymbol{p}_j, h_i) \cdot x_i \cdot x_{n+j} \tag{4.14}$$

Term $\Pi(\boldsymbol{p}_j, h_i)$ in Equation (4.14) represents the penalty associated to the planar hypothesis $\boldsymbol{p}_j$ and the object hypothesis h_i:

$$\Pi(\boldsymbol{p}_j, h_i) = \begin{cases} 0, & \mathcal{S}_{\boldsymbol{p}_j} \cap \mathcal{S}_{h_i} = \oslash \\ min\left(\Pi^{+}\left(\boldsymbol{p}_j, h_i\right), \Pi^{-}\left(\boldsymbol{p}_j, h_i\right)\right), & otherwise \end{cases} \tag{4.15}$$

with

$$\Pi^{+}(\boldsymbol{p}_j, h_i) = \sum_{q \in \mathcal{M}_{h_i}} \left(\boldsymbol{p}_j \circ (q_x, q_y, q_z, 1) \geq \rho_e\right) \tag{4.16}$$

$$\Pi^{-}(\boldsymbol{p}_j, h_i) = \sum_{q \in \mathcal{M}_{h_i}} \left(\boldsymbol{p}_j \circ (q_x, q_y, q_z, 1) \leq -\rho_e\right) \tag{4.17}$$

In words, the penalty brought in by $\Pi(\boldsymbol{p}_j, h_i)$ is zero if the two hypotheses $\boldsymbol{p}_j$ and h_i do not share scene points, otherwise it equals the less populous subset of the points belonging to model $\mathcal{M}_{h_i}$ lying on either side of the plane $\boldsymbol{p}_j$. It is worth pointing out that, since $\Pi(\boldsymbol{p}_j, h_i)$ does not depend on the specific solution $\mathcal{X}$, it can be precomputed once at the beginning of the optimization stage, after planar hypotheses have been extracted from the current scene.

4.5 Optimization strategies

To solve the global cost function in Equation (4.10) or (4.13), a solution $\tilde{\mathcal{X}}$ minimizing the function $\mathfrak{F}(\mathcal{X})$ over the solution space $\mathbb{B}^n$ has to be determined:

$$\tilde{\mathcal{X}} = \underset{\mathcal{X} \in \mathbb{B}^n}{\operatorname{argmin}} \{ \mathfrak{F}(\mathcal{X}) \} \tag{4.18}$$

As the cardinality of the solution space is 2^n, even with a relatively small number of recognition hypotheses (e.g. in the order of tens) exhaustive enumeration becomes prohibitive. To reach an approximate solution within a feasible computational time, a suitable solver for the class of non-linear pseudo-boolean optimization problems has to be deployed. A common choice is represented by neighbourhood explorations based methods, also known as metaheuristics (e.g., Local Search (LS), Simulated Annealing (SA) [34] and Tabu Search (TS) [21]).

In general, the basic idea behind such techniques consists in iteratively exploring the neighbourhood of the current solution. The neighbourhood is characterized by solutions *close* to the current one which are reached by a set of domain dependant *moves*. Once a move is applied, the cost function is evaluated with the new solution and compared against the cost associated with the *incumbent* solution (the solution with the best cost so far). Once a new solution is found with a better cost than the incumbent, the incumbent is updated and the algorithm iteratively proceeds by exploring the neighbourhood of the updated solution. If a solution is reached such that its associated cost is better than any of the costs associated with the solutions in its local neighbourhood, the algorithm terminates.

The success of metaheuristic algorithms depends strongly on the initial solution as well as the set of available moves and allocated time to explore the solution space. In general, there is no guarantee that the algorithm will terminate in a global minimum. However, some techniques implement mechanisms allowing them to escape local minimums and in practice, their performance is satisfactory for the problem at hand. In the next sections, we present a set of moves and review three metaheuristic techniques to determine an appropriate solution for the hypothesis verification problem. The performance of the different techniques is evaluated in Section 4.5.3.

4.5.1 Local neighbourhood (Moves)

A key component for the determination of an appropriate solution by means of metaheuristic techniques is related to the definition of the *neighbourhood* of a specific solution, $\mathcal{N}(\mathcal{X})$. In particular, we would like to define efficient moves to transition between $\mathcal{X}$ and $\mathcal{X}' \in \mathcal{N}(\mathcal{X})$ in order to explore the solution space of the hypothesis verification problem. In general, desirable properties of moves are represented by:

- The cost associated with $\mathcal{X}'$ should be efficient to compute from $\mathcal{X}$. Because the costs associated with the solutions in $\mathcal{N}(\mathcal{X})$ are required to guide the optimization process, it is crucial that their computation is efficient to render the overall optimization computationally feasible. In particular, moves should be

designed in such a way that the cost $\mathfrak{F}(\mathcal{X}')$ can be incrementally computed from $\mathfrak{F}(\mathcal{X})$, i.e., allowing to reuse computations associated with common elements of $\mathcal{X}$ and $\mathcal{X}'$.

- The neighbourhood of $\mathcal{X}$, $\mathcal{N}(\mathcal{X})$, should be large. In other words, it is desirable that a single move changes more than a single element of the current solution. A larger neighbourhood allows the optimization process to escape local minimums and usually results in superior performance (compared to moves that yield smaller neighbourhoods).

With these properties in mind, two moves have been designed for the hypothesis verification problem: (i) *switch state* and (ii) *replace active hypothesis*.

Switch state

Given the current solution $\mathcal{X} = \{x_1, ..., x_n\}$ with $x_i = \{0, 1\}$, a *switch state* move applied on the i-th hypothesis will switch the boolean value associated with x_i. Let $x_i \in \mathcal{X}$ be the state of the i-th hypothesis in the current solution and $x'_i \in \mathcal{X}'$ its state after applying the move, x'_i can be then expressed as:

$$x'_i = \neg x_i \tag{4.19}$$

A *switch state* move allows to efficiently compute the associated cost $\mathfrak{F}(\mathcal{X}')$ based on the pre-transition cost $\mathfrak{F}(\mathcal{X})$ as well as scene points influenced by the i-th hypothesis together with its model outliers Φ_{h_i}. As an example, consider the model term $f_{\mathcal{M}}(\mathcal{X}) = \sum_{i=1}^{n} |\Phi_{h_i}| \cdot x_i$. Depending on the status of x'_i, $f_{\mathcal{M}}(\mathcal{X}')$ can be incrementally computed as follows:

$$f_{\mathcal{M}}(\mathcal{X}') = \begin{cases} f_{\mathcal{M}}(\mathcal{X}) - \Phi_{h_i}, & x'_i = 0 \\ f_{\mathcal{M}}(\mathcal{X}) + \Phi_{h_i}, & x'_i = 1 \end{cases} \tag{4.20}$$

Given the particular structure of terms included in $\mathfrak{F}$, they can all be incrementally computed from previous moves, this notably increasing the efficiency of exploring the solution space of $\mathfrak{F}$.

Replace active hypothesis

While a *switch state* move changes a single variable of the current solution, *replace active hypothesis* moves aims at enlarging the neighbourhood of a specific solution and result in two variable changes. In particular, a *replace active hypothesis* move is stated as an appropriate combination of two *switch state* moves. Given the current solution $\mathcal{X} = \{x_1, ..., x_n\}$ with $x_i = \{0, 1\}$, a *replace active hypothesis* move is considered between elements x_i and x_j such that $x_i = 1$ and $x_j = 0$. After the transition, $x'_i = 0$ and $x'_j = 1$.

In order to further reduce the amount of *replace active hypothesis* moves, a move is considered only for hypotheses h_i and h_j that interact with each other. In its simplest way, two hypotheses are considered to interact if their scene support intersects, i.e., both hypotheses share common scene points.

4.5.2 Metaheuristics

The deployment of metaheuristic algorithms in this work is based on the METSlib library [41]. The library provides a set of generic implementations of common metaheuristic techniques such as Local Search, Simulated Annealing and Tabu Search. The algorithms can be configured to use a set of user-defined moves such as those previously presented in order to solve a specific problem.

Local Search (LS)

Local Search arguably represents the simplest metaheuristic algorithm. It is a monotonic optimization method and thus transitions are only accepted when the cost associated with the new state represents an improvement over the current cost. Local Search can be configured in two ways: (i) the first improving move is used to transition to a new state (Hill Climbing, LS_hc) or (ii) the best improving move in the current neighbourhood is used (Gradient Descent, LS_gd). Local search exploration will terminate when the neighbourhood of the current solution does not contain improving moves. Because of its monotonic nature, it converges quickly but can easily get trapped in local minima.

Simulated Annealing (SA)

Simulated Annealing [34] is a metaheuristic technique inspired by the physical process of annealing in metallurgy. To simulate the annealing process, the minimization process is associated with an initial high temperature that decreases during the application of moves. The temperature of the system is used in combination with the costs associated to a particular transition, to define the probability of accepting that move. When the temperature is high, SA might transition to solutions with associated high costs, allowing the procedure to explore large regions of the solution space and thus, reducing the probability of being trapped in local minima. As the system cools down, the probability of accepting bad moves is reduced and the algorithm favours transitions that steadily decrease the global cost. In this work, an exponential cooling schedule has been deployed (see [34] for further details). SA will terminate if any of the following conditions are satisfied:

- The temperature of the system is lower than T_{min}.
- No improvement has been found during the application of the last N_{max} moves.

Tabu Search (TS)

Tabu Search [21] is another metaheuristic technique based on the exploration of the neighbourhood of the current solution. In our implementation, the *best* neighbouring solution is always accepted regardless of its associated cost function as well as of the current solution. However, during the execution of TS, a *tabu list* is maintained (containing solutions previously visited) and a solution is accepted for further exploration only if it is not contained in the tabu list. The deployment of a tabu list allows on one hand to avoid cycles during the exploration, on the other hand provides TS with the ability to escape from already visited local minima (since they will be included in the tabu list). The algorithm terminates when all solutions in the neighbourhood of the current one are in the tabu list, or no improvement has been achieved during the last N_{max} moves.

4.5.3 Evaluation

We have compared the different meta-heuristics with the goal of determining the best algorithm for the optimization problem associated with GHV. In our analysis, we have compared the average number of evaluated solutions as well as the average final value of the cost function[1] yielded by each method over 4 different benchmark datasets (which will be introduced more in details in Section 5.1) and on the same hypotheses yielded by the pipeline proposed in Section 3.4. Figure 4.5 reports these results, where the suffix _RM indicates the use of *Replace active hypothesis* moves in addition to *Switch state* moves. We have configured both SA and TS with $N_{max} = 200$. Different than in [4], the initial solution for all algorithms is represented by all hypotheses being deactivated ($x_i = 0, \forall x_i \in \mathcal{X}$), as this tends to yield a faster convergence: indeed, since GHV holds an extremely discriminative power for false positives, typical working conditions will be characterized by a high number of hypotheses fed to GHV, the majority being false positives hence switched off in the final solution representing the achieved minimum.

Regarding the amount of evaluated moves, we can observe that methods based on Local Search (LS_hc and LS_gd) yield a faster convergence than the more complex SA and TS meta-heuristics. Moreover, their performance in terms of average minimum cost — in particular for LS_gd_RM — is surprisingly good. However, in the *Clutter* dataset, the performance of TS is slightly better (see Figure 4.5-(d)). This is caused by the fact that monotonic methods with the set of available moves are unable to explore parts of the solution space involving the simultaneous activation of two or more hypotheses, none of them yielding a cost improvement when independently activated. Since TS always explores the solution space given by the best move (regardless of cost improvements), the case where two or more hypotheses are required to be activated for a positive cost contribution is indeed explored.

This is in general related to the clutter cue deployed within GHV. Please note that the clutter term considers only scene points that are *unexplained* by the current

[1] Being the minimum negative in value, the inverse of the cost is plotted

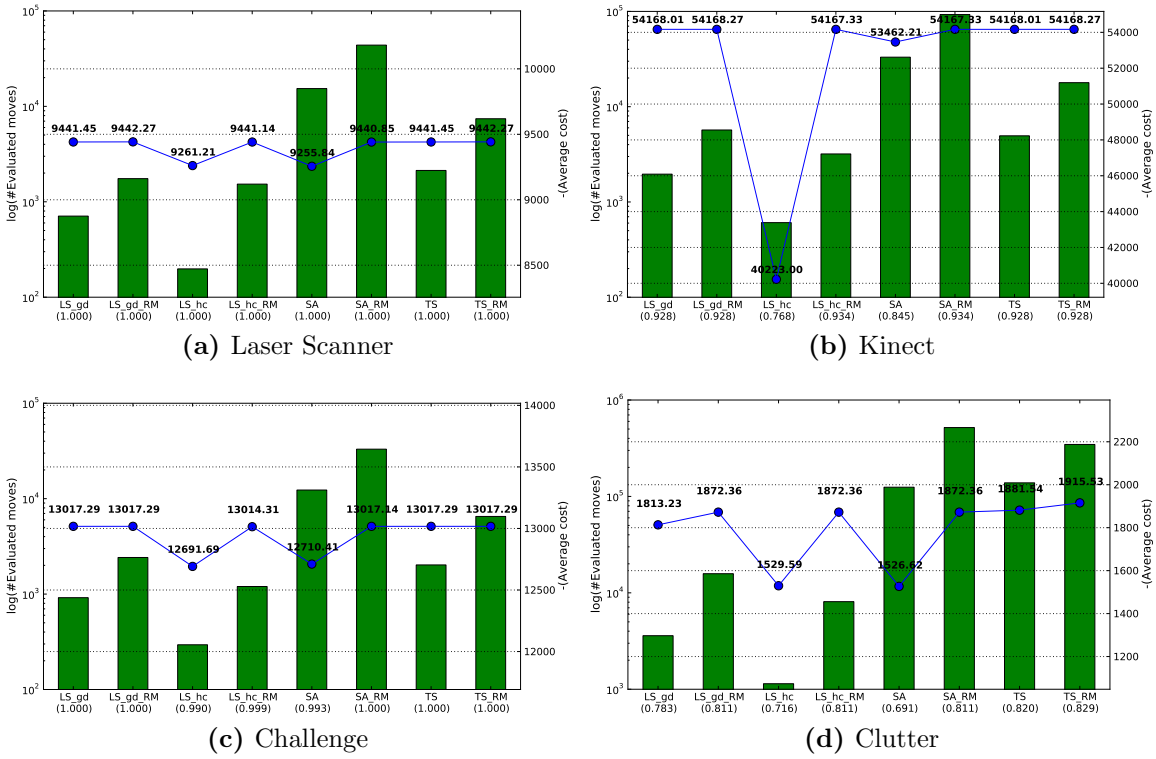
active hypotheses. Therefore, in situations where the smooth segments spread over multiple individual objects, the activation of a single correct hypothesis might result in a substantial increase of the global clutter term (if their surroundings are not yet *explained* by the current solution) and cause an increase in the global cost, indicating that the hypothesis should be rejected. However, if the optimization strategy explores the solution space spanned by the activation of this hypothesis, regardless of the current cost increase (e.g. Tabu Search), the next move might result in the activation of a second correct hypothesis which causes the cluttered areas from the previous hypothesis to become *explained* and thus reduce the clutter term associated with this solution. The increase of the number of scene points being explained without a significant increase of other terms penalizing incorrect hypotheses results in a cost decrease and the current solution (with both hypotheses activated) becomes the current incumbent.

In order to illustrate the performance of the different optimization strategies not solely based on their associated costs but also in terms of recognition capabilities, Figure 4.5 shows the F-score:

$$F = 2 \cdot \frac{(precision \cdot recall)}{(precision + recall)} \tag{4.21}$$

achieved with the different evaluated methods (under each acronym). Observe how some optimization strategies associated with worse costs than other methods, result in an equal performance in terms of F-score (Figure 4.5-(a)). This is explained by similar object hypotheses being present in the hypotheses set, $\mathcal{H}$, causing local minima in the cost function, from which some optimization strategies are unable to escape. Due to the similarities between objects in the model library in the *Kinect* dataset, we can observe in Figure 4.5-(b) that the F-score given by SA_RM and LS_hc_RM is higher than TS_RM (overall best performing method in terms of reached minimum cost) even though their associated cost is slightly worse. Such fact indicates small inaccuracies in the cues deployed within GHV which are unable to robustly differentiate between very similar models. This leads in some situations to better costs being associated with solutions providing a small decrease in terms of the F-score metric.

The bad performance of Simulated Annealing (SA), specially when *replace active hypothesis* are deactivated, is explained on one hand by a suboptimal configuration of the different parameters and cooling strategy associated with it. On the other hand, these results indicate that the design of SA might not be the most suitable for the verification problem at hand. In particular, because the temperature associated with Simulated Annealing is high at the beginning (increasing the probability of accepting moves with worse costs than the current incumbent), it causes the exploration of solution subspaces that are unlikely to be a good representative of the global minimum. Instead, it would be better to have a steepest descent exploration at the beginning (similar to Local Search methods or Tabu Search) and once a good solution is found, explore slightly worse neighbour solutions in order to escape local minima. Additionally, the fact that SA requires additional parameters (compared to



(a) Laser Scanner **(b)** Kinect

(c) Challenge **(d)** Clutter

Figure 4.5: Results for the different optimization strategies on the different datasets. The suffix _RM indicates the use of *replace active hypothesis* moves in addition to *switch state* moves which are always used. F-score reported for the different meta-heuristics between parentheses.

other meta-heuristics) makes its deployment less appealing. Based on this evaluation, hereinafter the method deployed within GHV will be TS, as the best trade-off between accuracy in the yielded solution and required number of moves.

4.6 GHV parameters

Even though the GHV formulation effectively reduces the amount of hard thresholds involved in the verification stage, the algorithm is governed by several parameters. This section aims at providing an overview of the different parameters associated with GHV as well as a set of guidelines to ease their configuration in novel scenarios or different data properties. The following list represents the main parameters in GHV:

- *Inliers threshold* (ρ_e): Represents the maximum distance between model and scene points in order to state that a scene point is explained by a model point. Valid model points that do not have any corresponding scene point within ρ_e are termed *model outliers*.

- *Clutter influence radius* (ρ_c): Represents the maximum distance between an *explained* scene point, p, and other unexplained scene points such that they influence the clutter term associated with p.
- *Outlier penalty multiplier* (λ): Represents a penalty multiplier for *model outliers*. In particular, each model outlier associated with an active hypothesis increases the global cost function by λ.
- *Smooth clutter penalty multiplier* (κ): The penalty multiplier used to penalize unexplained points within the clutter influence radius (ρ_c) of a explained scene points when they belong to the same smooth segment.
- Parameters associated with the deployment of color information:
  - *Color similarity threshold* ($\rho_l \in [0, 1]$): Represents the minimum similarity (in terms of color) to decide if a scene point is explained by a model point.
  - *L-channel variance* (σ_l): Variance associated with the L-channel (LAB color space) to assess similarity between model and scene points.
  - *A,B-channel variance* ($\sigma_a = \sigma_b$): Variance associated with the A and B channels (LAB color space) to assess similarity between model and scene points.

In addition to these, other parameters are indirectly used within GHV. For example, parameters controlling the smooth segmentation procedure or the support radius to estimate point normals as well as parameters associated with the deployed optimization strategy.

Regarding their configuration, the values taken by some of these thresholds and parameters are directly related to the properties of the underlying data being processed. In particular, the *inliers threshold* is chosen considering the amount of noise in the data as well as its resolution. To choose the color similarity threshold and color variances values it is recommended to consider the expected illumination conditions during recognition as well as illumination conditions available during the model acquisition phase. Moreover, if different sensors are used during model acquisition and recognition, one should consider this by increasing the expected variances regarding the A and B channels. To choose appropriate parameter values related to the clutter term, the expected recognition scenarios is of major importance. In particular, if we expect scenes to be heavily cluttered, the parameters are to be carefully chosen in order to avoid the rejection of correct hypotheses. Finally, the penalty associated with model outliers should be chosen depending on several factors such as accuracy of provided poses as well as application tolerance regarding false positives. Chapter 5 provides specific configuration choices for the different datasets used to evaluate the proposed recognition system.

Chapter 5

Evaluation

In addition to the individual experiments carried on for specific parts of this thesis, the main aim of this section is to evaluate the complete recognition system. To this end, we have gathered the most popular public benchmarks for 3D object recognition in clutter and performed several experiments to demonstrate the overall capabilities of the system.

5.1 Datasets

The selected datasets are heterogeneous in the traits of the employed models as well as in those of the sensors used for acquisition, as detailed in Table 5.1: while some datasets only include point clouds and 3D meshes of highly descriptive objects acquired with laser scanners (*Laser Scanner*, *Queens*), others have been acquired with structured light sensors and relate to robotic applications aimed at manipulation of typical household objects(*Kinect*, *Challenge*, *Willow*, *Clutter*). All these datasets have been extensively used in literature, this allowing direct comparison with state-of-the-art approaches. Peculiar to our method is the ability to generalize among different types of data as well as different scene configurations, which is demonstrated by these exhaustive choice of datasets.

5.1.1 Laser Scanner Dataset

The Laser Scanner Dataset, originally proposed by Mian et al. [44], is a well-known benchmark for 3D object recognition. It is composed of 50 scenes, each displaying at least 4 objects (there exist 5 models) in different configurations (see Figure 5.1). The scenes were obtained using a laser scanner and the data is relatively clean. The objects present distinctive geometrical traits and thus, the application of 3D local features is suited for this dataset. The main challenge is represented by the recognition of highly occluded instances (in some cases over 90% occlusion, see Figure 5.7-(a)) under clutter generated by other object instances.

Dataset	Properties				
	#Models	#Scenes	#Inst.	Model data	Test data
Laser Scanner	4	50	188	3D mesh	Point cloud
Laser Scanner¶	5	50	217	3D mesh	Point cloud
Queen's	5	80	240	3D mesh	Point cloud
Kinect	35	50	176	3D mesh	Range image
Challenge	35	176	434	RGB Point cloud	RGB-D
Willow	35	353	1628	RGB Point cloud	RGB-D
Clutter	18	30	120	Point cloud	RGB-D

Table 5.1: Properties of the dataset, including the representation of model and test data. *Laser Scanner*¶ represents the dataset with the *rhino* model.

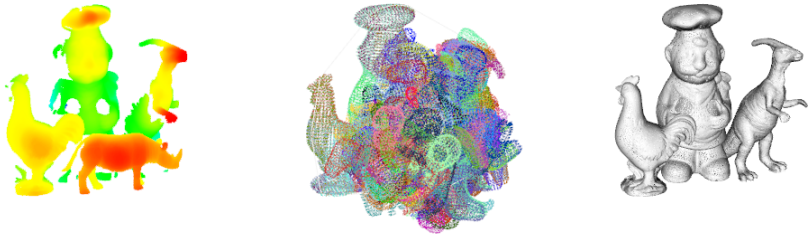


Figure 5.1: Sample scene for the Laser Scanner dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

5.1.2 Queen's Dataset

The Queen's dataset is composed of 80 scenes and a model set of 5 objects. The test data was obtained using a LIDAR scanner and similar to the Laser Scanner dataset, the data is relatively free of noise, presents however an irregular point density (see Figure 5.2). The object models present as well distinctive geometrical traits. Contrary to the Laser Scanner Dataset, some scenes present a planar surface on which the objects are standing, creating an additional amount of clutter around the object instances.

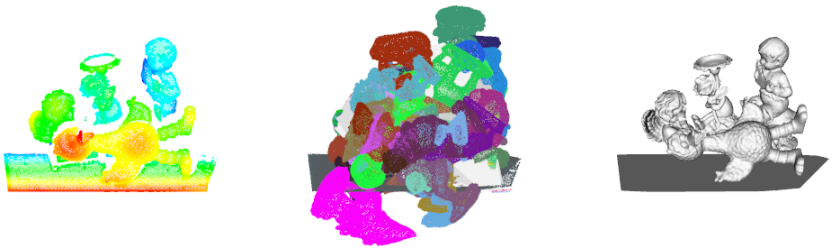


Figure 5.2: Sample scene for the Queen's dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

5.1.3 Kinect Dataset

The Kinect dataset is composed of 50 scenes obtained with a Kinect sensor. The model library is in this case larger than in the previous datasets, totalling 35 object models which are represented as a 3D mesh without color information. The main challenges in the dataset are represented by the presence of typical Kinect artefacts in the test data and the high similarity between some objects models (see Figure 5.3). Because the objects to be recognized are presented in a table-top setup, the objects can be easily segmented which enables the deployment of the global pipeline.

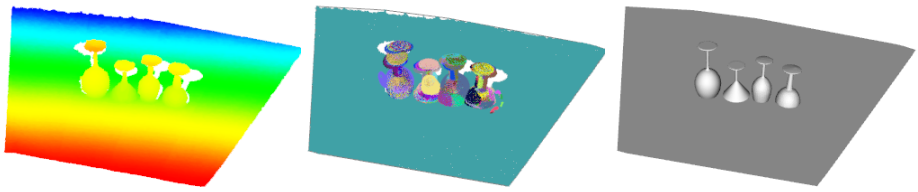


Figure 5.3: Sample scene for the Kinect dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

5.1.4 Challenge Dataset

The Challenge dataset is composed of 176 scenes obtained with a Kinect sensor mounted on a PR2 robot. The model library contains 35 textured household objects represented as a registered RGB point cloud, thus allowing the deployment of textured-based local pipelines (i.e, SIFT). Because of the table-top setup present in the scene (see Figure 5.4), objects are easy segmentable from the background enabling the deployment of global pipelines.

Figure 5.4: Sample scene for the Challenge dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

5.1.5 Willow Dataset

The Willow dataset is composed of 353 scenes with a similar setup to that of the Challenge dataset (see Figure 5.5) and same model library. However, the test scenes in this case present in some cases highly occluded instances (see Figure 5.7-(e)) as well as multiple instances of the same object. In addition, several scenes contain impostor objects (i.e, objects that are not known to the recognition system) such as the two *Odwalla* bottles shown in Figure 5.5.

Figure 5.5: Sample scene for the Willow dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

5.1.6 Clutter Dataset

Finally, the Clutter dataset is composed of 30 scenes obtained with a Kinect sensor and a model library of 18 objects. This dataset is characterized by scenes presenting complex setups where the objects to be recognized strongly interact with each other, this causing severe clutter and occlusions (see Figure 5.6). Even though color information is available for both models and scenes, the misalignments between color and depth during the model acquisition makes it not possible to use color information during the hypothesis verification stage.

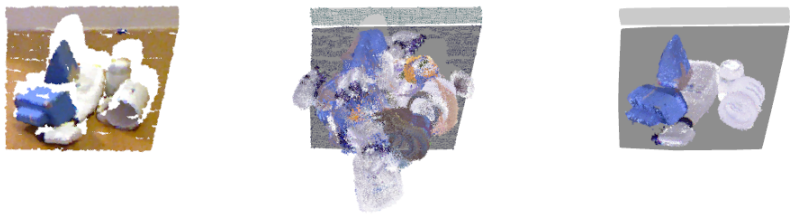


Figure 5.6: Sample scene for the Clutter dataset (left), object and planar hypotheses (middle) and verified hypotheses (right).

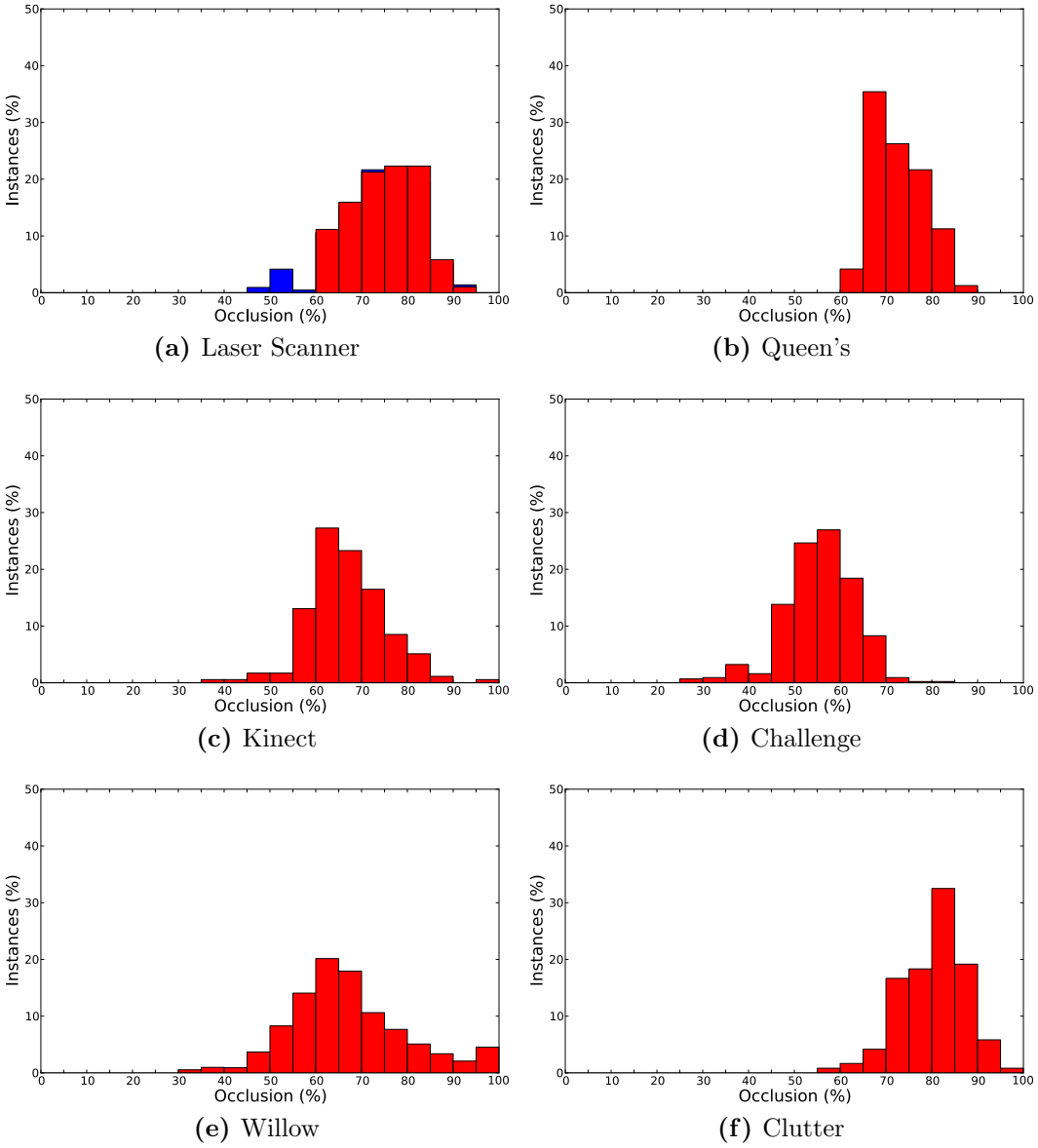


Figure 5.7: Occlusion distribution for the different datasets. The blue bars on the Laser Scanner dataset histogram represent the *rhino* instances.

5.2 Color and tonal specification

To experimentally motivate the proposed color extension of GHV as described in Section 4.2.5, we have compared the results obtained by our method in four different configurations: (i) geometry only, (ii) no tonal registration, (iii) tonal registration using all explained points by an hypothesis and (iv) independent tonal registration for each smooth face of the hypothesis model. Figure 5.8 reports these results on the *Challenge* dataset, showing how the use of color enhances the capability of GHV to

distinguish among objects of similar shapes (e.g., observe the lower precision of the 'geometry only' configuration) as well the effectiveness of tonal mapping to better deal with illumination changes and photometric distortions between models and scene. Moreover, it is worth noting how the use of color without tonal registration results in the rejection of correct hypotheses due to color inconsistencies. Independent tonal registration for different object parts results in the best performance, being able to cope with scene-model illumination transformations that are not globally consistent on the whole object.

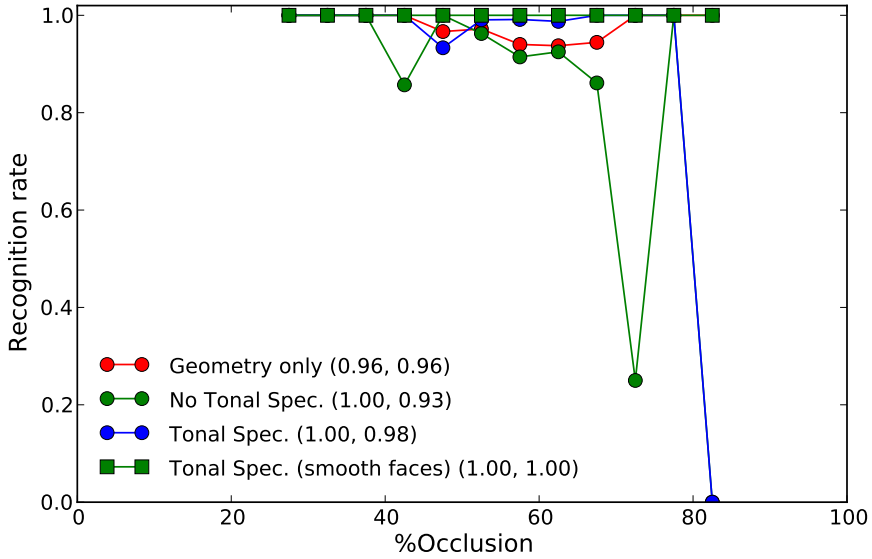
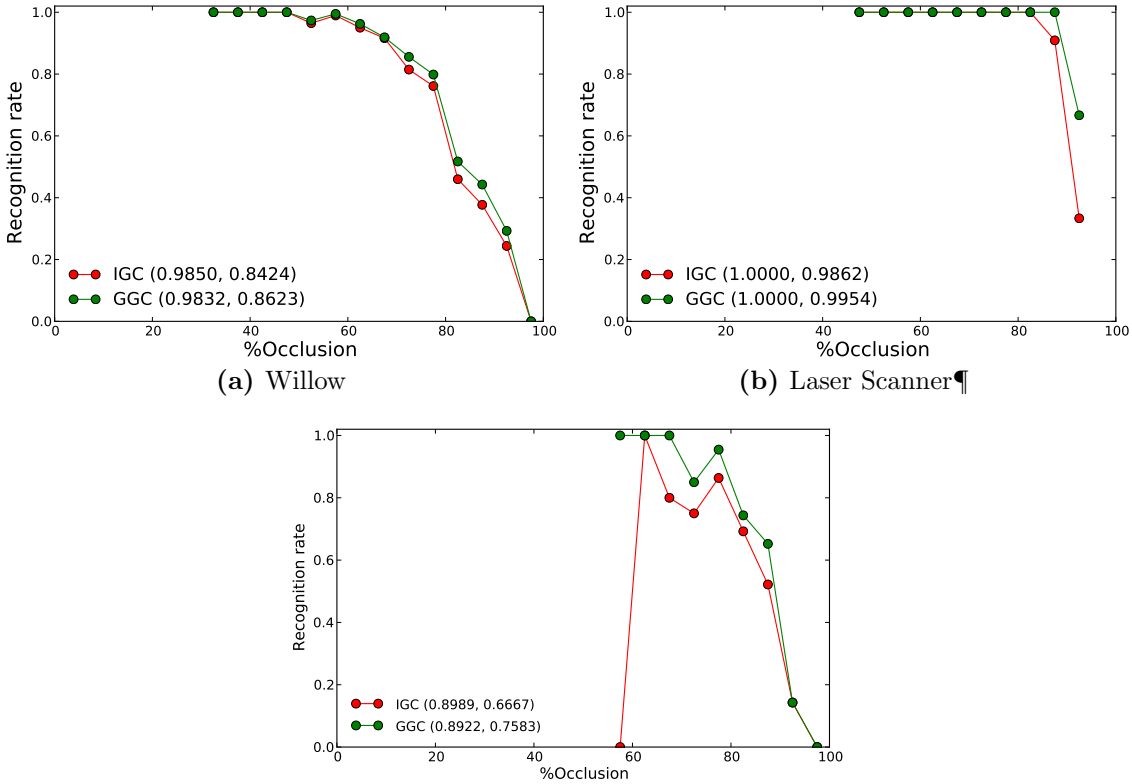


Figure 5.8: Color and tonal specification results on the Challenge dataset. $\rho_l = 0.8, \sigma_L = 0.25, \sigma_{AB} = 0.3$.

5.3 Correspondence grouping (GGC vs IGC)

In this experiment, we compared the proposed GGC Correspondence Grouping method against the IGC method deployed in [4]. To this end, we compared the hypotheses generated by the local recognition pipeline (SHOT and SIFT; when applicable) with GGC and IGC on three datasets. Figure 5.9 reports these results which clearly depict the superior performance of GGC compared to IGC. In particular, we can observe an increase in the recognition rate as the occlusion level increases, causing correspondences to become noisier and smaller in number, and thus disturbing the correspondence seeds for IGC. In the *Clutter* dataset, the improved performance of GGC becomes plausible at even lower occlusion levels due to the significant amount of clutter within the descriptor support, which causes as well a significant deterioration of the associated features. The use of GGC comes however at the price of more generated hypotheses and can eventually cause a small decrease in the overall precision if GHV is unable to reject incorrect hypotheses (see Figure 5.9-(a)). Nevertheless, the benefits in terms of recognition rate compensates for the extra false detections.



(a) Willow **(b)** Laser Scanner¶

(c) Clutter

Figure 5.9: Graph Based Geometric Consistency Grouping vs Iterative Geometric Consistency Grouping. Only the Local Pipeline (SHOT and SIFT features when applicable) was deployed. Precision and recall (between parantheses) reported only for visible instances (Willow dataset). GHV configuration kept constant for both hypothesis sets.

5.4 Performance of different recognition pipelines

Aiming at experimentally highlighting the advantages associated with the deployment of multiple recognition pipelines, we have conducted an experiment evaluating their individual performance. These results have then been compared with those obtained by combining hypotheses generated by individual pipelines during the verification stage. Figure 5.10 presents these results. Observe in all situations how the the combination of global and local pipelines results in the best performance.

Because the model library in the Challenge and Willow dataset is composed of textured objects, the performance of the local pipeline based on SIFT features is remarkably good. By adding correspondences based on 3D shape features (SHOT), the performance of the local pipeline increases even further, resulting almost in ideal performance (1 single false negative) on the Challenge dataset. Regarding the global pipeline (based on OUR-CVFH), it is the best performing individual pipeline on Challenge and Willow. This is facilitated by the table-top scene setup where objects are easily segmentable.

However, observe that the deployment of the global pipeline on the Willow dataset increases the number of false positives. Because of the impostor objects in this dataset, which share similar shape and color attributes with some objects in the model library, the verification stage is unable to reject these wrong hypotheses. As objects undergo stronger occlusions, its performance is, as expected, below that of the local pipelines.

Regarding the Kinect dataset (see Figures 5.10-(c)) the local pipeline (based solely on SHOT features since color is not available) performs best. This is quite remarkable taking into account the common geometrical traits present in the object models. Unfortunately, in this dataset, the segmentation provided by MPS presents several under-segmentation artefacts (i.e, when objects are touching each other) which deteriorate the performance of the global pipeline.[1]

Figures 5.10-(a) and 5.10-(b) highlight as well the additional benefits of merging correspondences from different local pipelines at the correspondence grouping stage, aiming at increasing the consensus of object hypotheses. While in the "SIFT + SHOT" pipeline, the scene-model correspondences obtained by means of both local features are merged at the CG stage, "SIFT and SHOT" presents the results obtained when both pipelines generate object hypotheses on its own. In particular, "SIFT + SHOT" provides an improvement of 0.66% and 2.5% recall for the Challenge and Willow dataset, respectively.

5.5 Computational remarks

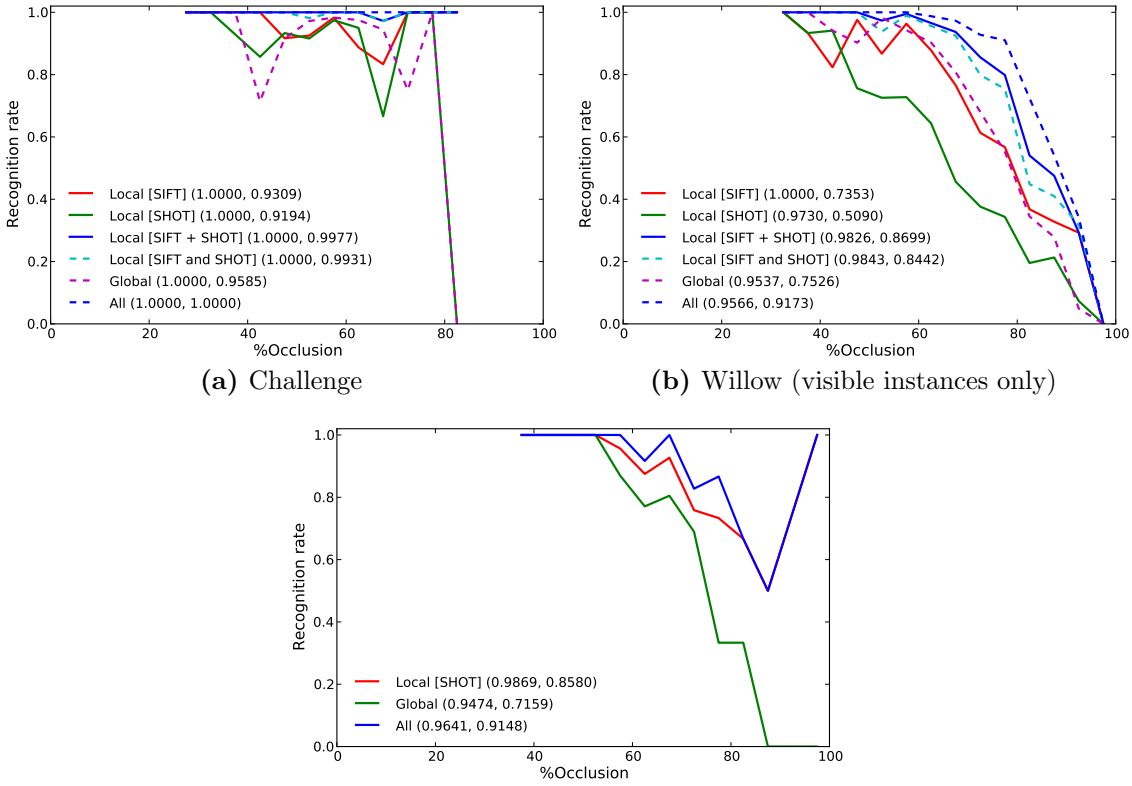
Regarding the computational efficiency of the GHV method and in addition to the evaluation of the different optimization strategies (see Fig. 4.5), we evaluated the total amount of time taken by the two main stages of the GHV algorithm: (i) computation of cues and (ii) optimization of the cost function. These results are reported in Figure 5.11. Figure 5.11-(a) plots the different amount of time taken with respect to the number of hypotheses to be verified. It can be observed that there exists an almost linear dependency between required time and number of hypotheses. However, the time required in a few scenes seems to deviate from this linear relation.

Because the GHV algorithm is based on point operations, the size of the hypotheses being verified influences its performance. In particular, Figure 5.11-(b) reports again the time required with respect to the amount of visible points of the different object hypotheses in a scene. In this case, it is clearly observable that both variables do present a linear dependency without any major deviations.

This linear dependency, in combination with the fact that operations performed at each point are independent of other points, indicate that the GHV algorithm might be suited for massive parallelisation on modern Graphic Processing Units (GPU).

The measured time regarding the computation of cues includes not only operations performed on object hypotheses but also on the scene (smooth segmentation, normal

[1]Note that the segmentation proposed by Richtsfeld requires color information and thus could not be deployed for the Kinect dataset.



(a) Challenge **(b)** Willow (visible instances only)

(c) Kinect

Figure 5.10: Performance of individual and combinations of different recognition pipelines on Challenge, Willow and Kinect datasets.

computation, etc.). Obviously, the amount of time spent on these operations is independent of the number of hypotheses. This fact is observable in Figure 5.11 where the computation of cues is always above two seconds regardless of the number of hypotheses.

5.6 Comparison against the state-of-the-art

Finally, we have carried out an exhaustive comparison of the proposed recognition system with the state of the art. In particular, we have compared it on all six benchmark datasets and against the state-of-the-art algorithms thereby evaluated in the literature. Table 5.3 reports the performance in terms of Precision, Recall and F-score obtained by GHV. For the *Clutter*, *Kinect*, *Willow* and *Challenge* datasets, where methods are typically evaluated in these terms [22, 68, 60, 5], the Table reports the published results from competing proposals. Instead, for what concerns the *Laser Scanner* and *Queen's* datasets, Figure 5.12 shows the Recognition Rate vs. Occlusion Rate charts, being this the standard way deployed in literature for experimental comparison on these datasets [11, 59, 50, 17]. For what concerns the *Queen's* dataset,

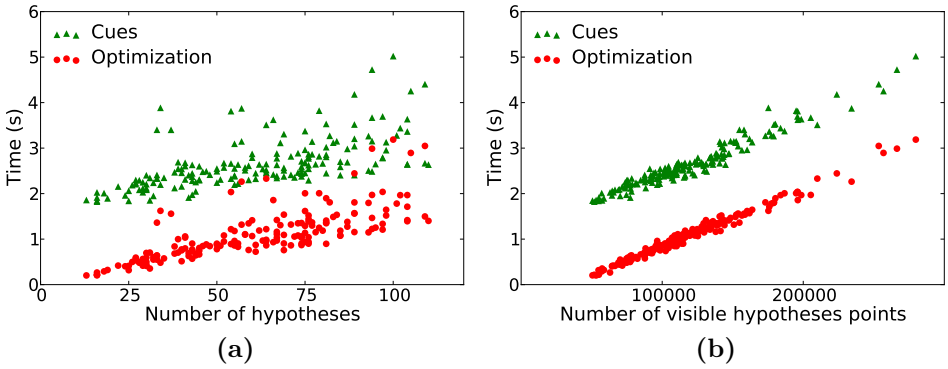


Figure 5.11: Timing results on the Challenge Dataset using Tabu Search (only with *Switch State* moves) as optimization strategy.

Figure 5.12 reports the average recognition rate (in the form of a horizontal dotted line) for [59] rather than the full plot, as not available in the original paper.

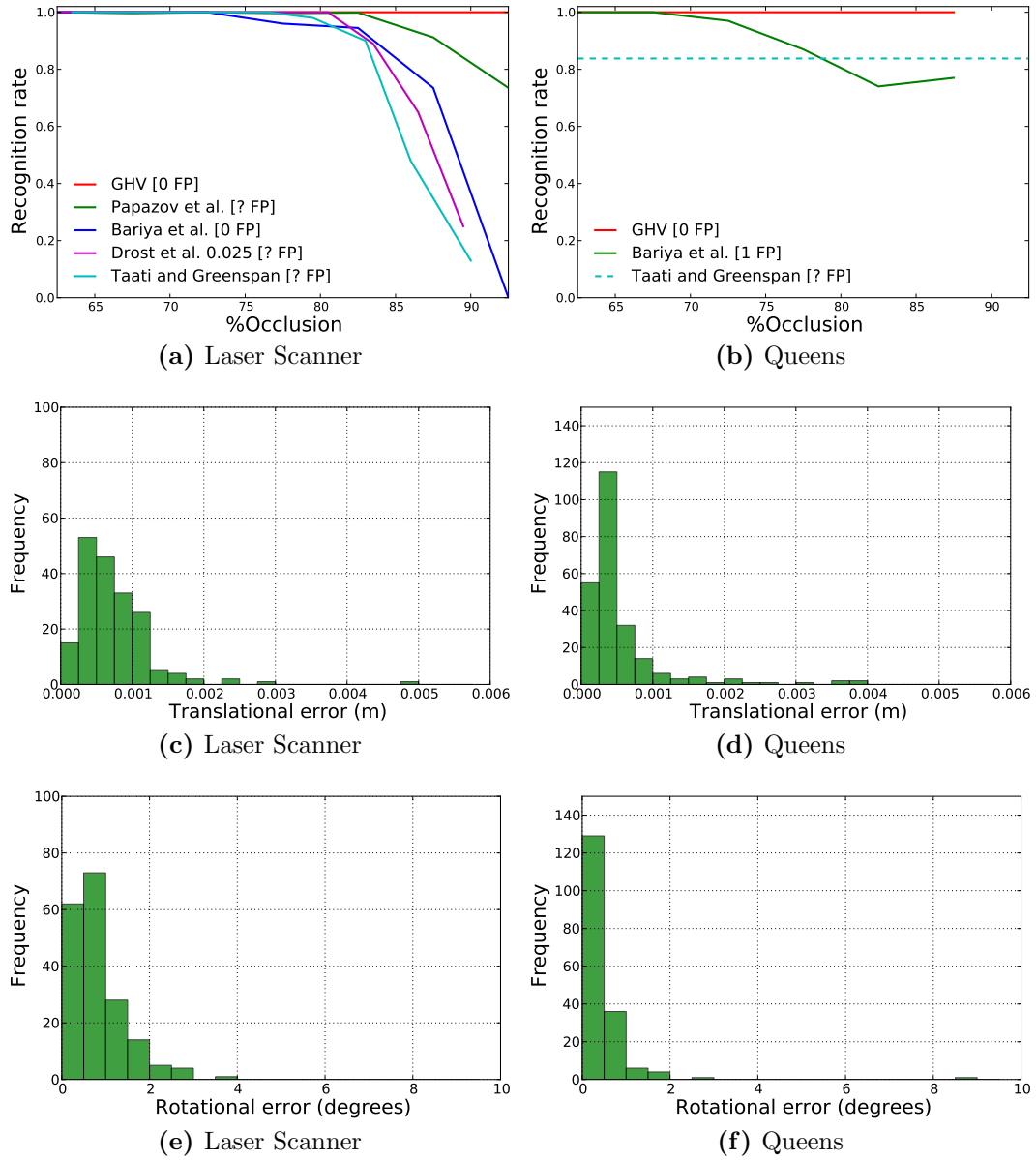
As reported from the Table 5.3 and the Figure 5.12, GHV outperforms all methods on five of the six datasets used in the evaluation, while its performance is comparable to that of the best method on the *Willow* dataset. Also worth mentioning, the proposed pipeline yields the ideal performance (maximum Precision and Recall) on three of the six evaluated datasets. Additionally, Figures 5.12 also show the translational and rotational errors (respectively, in SubFigures b and c) with respect to the ground-truth reported by GHV on the *Laser Scanner* and *Queen's* datasets. As it can be seen, the 6DoF pose estimated by GHV for the recognized models is always extremely accurate, the error being in the great majority of cases below $1mm$ and $2°$. Table 5.2 summarizes the GHV parameters for the different datasets.

Dataset	**Parameters**				**Color Parameters**		
	$\rho_e(mm)$	$\rho_c(cm)$	κ	λ	ρ_l	σ_L	σ_{AB}
Laser Scanner	5	3	5	3	-	-	-
Laser Scanner¶	5	3	5	3	-	-	-
Queen's	8	4	5	3	-	-	-
Kinect	8	4	5	5	-	-	-
Challenge	8	4	5	4	0.8	0.25	0.3
Willow	8	4	5	4	0.8	0.25	0.3
Clutter	8	4	5	4	-	-	-

Table 5.2: GHV parameters for the different datasets. *Laser Scanner*¶ represents the dataset with the *rhino* model.

Dataset	Method	Prec.	Recall	F-score
Laser Scanner	GHV	1.0000	1.0000	1.0000
Laser Scanner¶	GHV	1.0000	0.9954	0.9976
Queen's	GHV	1.0000	1.0000	1.0000
Kinect	GHV	0.9415	0.9148	0.9279
	Glover [22]	0.8940	0.8640	0.8788
	Aldoma [4]	0.9090	0.7950	0.8481
Challenge	GHV	1.0000	1.0000	1.0000
	Tang [60]	0.9873	0.9023	0.9429
	Xie [68]	1.0000	0.9977	0.9988
	Aldoma [5]	0.9977	0.9977	0.9976
Willow	GHV	0.9784	0.8636	0.9173
	Xie [68]	0.9828	0.8778	0.9273
	Aldoma [5]	0.9430	0.7086	0.8091
	Tang [60]	0.8875	0.6479	0.7490
Clutter	GHV	0.8989	0.7583	0.8225
	Glover [22]	0.8380	0.7330	0.7819
	Aldoma [4]	0.8290	0.6420	0.7236

Table 5.3: Summary of results on the different datasets. See Figure 5.12 a for comparison with state-of-the-art for the Laser Scanner and Queen's dataset.



(a) Laser Scanner **(b)** Queens

(c) Laser Scanner **(d)** Queens

(e) Laser Scanner **(f)** Queens

Figure 5.12: Recognition rate vs occlusion on (a) the Laser Scanner Dataset (without *rhino*) and (b) the Queen's Dataset (all 80 scenes). (c-d) and (e-f) respectively report translational error and rotational error for GHV. Please refer to the original papers for accurate plots of their results.

Chapter 6

Automating "Ground Truth" annotations

The availability of large, challenging and varied datasets is a key element to evaluate progress in many robotic tasks. While a few RGB-D datasets are available for object class [35, 43] and object instance recognition [26, 22], more datasets are required to cover a larger spectrum of real-world challenges faced in robotic perception: changing lighting conditions, complex scene layouts as well as objects undergoing occlusions, being not easily segmentable from the background and/or not presenting discriminative features.

Despite the acquisition of RGB-D data being greatly simplified by modern sensing technologies, a major issue holding back the proliferation of benchmark datasets is related to their annotation being time consuming and tedious; in particular, if accurate poses for object instances are required. While it is possible to *automate* the process by means of fiducial patterns, using such techniques results in unnatural scenes and imposes restrictions on their layout (e.g. table-top scenarios). For instance, in the datasets proposed for the ICRA11 Perception Challenge, objects are placed using fixtures on a planar surface equipped with a checkerboard pattern. Since the fixtures' position and orientation relative to the pattern are known, the pose of the objects located at each fixture can be estimated up to the accuracy of the pattern detection algorithm and fixture-pattern relative measurements. Such a method still requires a human operator to provide object-fixture correspondences.

Aiming at reducing the aforementioned burden as well as the limitations of current techniques, this chapter tackles the problem of *automating* "ground truth" annotation for multi-view RGB-D object instance recognition datasets. Specifically, we consider datasets composed of sequences of RGB-D frames where each frame provides an additional viewpoint of the scene under consideration. To simplify the original recognition problem, the main idea consists in exploiting the additional information provided by multiple vantage points to build a richer and integrated representation of the scene as well as the objects therein. Intuitively, while multiple viewpoints increase the probability of seeing the object from an advantageous perspective (i.e, the object becomes fully visible or a distinctive part is revealed), the integrated representation provides a stronger evidence of an object being actually present in the scene and thus facilitates the removal of spurious single-view detections. Under a small set of

Figure 6.1: Annotation examples on two datasets. The exploitation of multiple vantage points facilitates accurate annotations of objects undergoing strong occlusions in complex scene layouts. Images are generated by blending the annotations from our method into the RGB image (with reduced opacity).

assumptions stated in Section 6.1.7, we argue that recognition results obtained on such a representation are close to the actual ground truth of the data. Therefore, the main contributions of this work are related to:

(i) *How to build such a representation?* We do this in two steps. In a first step, a *single-view recognition* system is deployed on each frame. In a second step, single-view detections in combination with visual features originating from single frames are used to *reconstruct* a 3D representation of the scene.

(ii) *How to use it in order to solve the multi-view recognition problem?* By projecting single-view detections (*object hypotheses*) into the reconstructed scene, the problem boils down to selecting a subset of hypotheses that *best* represent the reconstructed sequence, achieved in our proposal by means of a multi-view hypothesis verification stage.

We used the proposed method to automatically annotate more than 95% of the 3500 object instances in two large datasets totalling 516 RGB-D frames, including many frames where some objects were largely occluded (see Figure 6.1. Thus, in combination with a final manual stage to verify and extend automatic annotations, the method is useful to accurately annotate large amounts of data with a significant reduction in the amount of manual intervention.

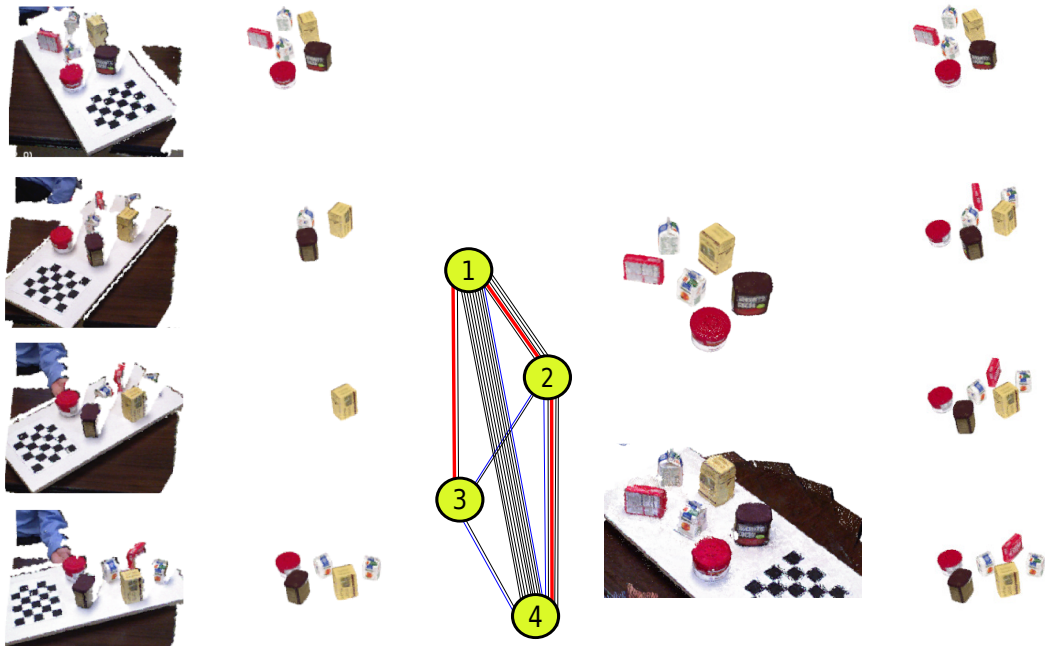


Figure 6.2: Workflow of proposed multi-view recognition method to generate "ground truth" annotations in a sequence consisting of 4 RGB-D frames: a) input RGB-D frames; b) single-view recognition results; c) graph representation of multiple views. If the same object was recognized in two views or the views can be registered by visual features (blue edges), an edge is added to the graph connecting the views with an associated transformation and an appropriate weight. The subsequently calculated Minimum Spanning Tree is shown by red edges; d) reconstructed scene and projected hypotheses remaining after the 3D verification stage; e) verified hypotheses are back-projected to the original frames, generating "ground truth" annotations.

6.1 Proposed approach

Provided with a set of models with m point clouds $\mathcal{M} = \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m\}$ and a set of n RGB-D frames belonging to a sequence $\mathcal{S} = \{\boldsymbol{S}_1 \ldots \boldsymbol{S}_n\}$, the goal of the proposed method is to detect in each frame all objects known to the system together with their pose. Figure 6.2 depicts the overall structure of the multi-view recognition method.

6.1.1 Single-view recognition

The single-view recognizer generates for each scene point cloud $\boldsymbol{S}_k \in \mathcal{S}$ a set of hypotheses $\mathcal{H}_k = \left\{h_1^k, h_2^k, \ldots h_p^k\right\}$, where

$$h_j^k = \left\{o_j^k, \boldsymbol{P}_j^k\right\}, \quad 1 \leq j \leq p \tag{6.1}$$

describes a single hypothesis with the object identity $o_j^k \in \mathcal{M}$ and a 4×4 transformation matrix $\boldsymbol{P}_j^k$ defining the 6DoF object pose with respect to the reference frame of $\boldsymbol{S}_k$. Object instances and poses in single frames are obtained by deploying the

single-view recognition system proposed in this thesis. Single-view hypotheses are generated using the recognition system proposed in Section 3.4. Note that the rest of the method is independent of this stage and other single-view approaches might be deployed (e.g. [26, 68]), provided that they retrieve a set of objects with their poses.

6.1.2 Multi-view graph representation

The multi-view stage starts by creating a set of vertices $\mathcal{V} = \{\mathcal{V}_1 \dots \mathcal{V}_n\}$, where each vertex contains single-view hypotheses with their respective scene point cloud

$$\mathcal{V}_k = \{\boldsymbol{S}_k, \mathcal{H}_k\}, \quad 1 \leq k \leq n. \tag{6.2}$$

By iteratively comparing vertex pairs with respect to their hypotheses sets, vertices sharing a hypothesis with the same model identity o are connected by an edge

$$\begin{aligned} \mathcal{E}_{ij}^{l} &= \left\{o_{ij}^{l}, \boldsymbol{T}_{ij}^{l}, \vartheta_{ij}^{l}, i, j\right\} \\ \forall i, j \mid \left(o_{ij}^{l} \in \mathcal{H}_i\right) \wedge \left(o_{ij}^{l} \in \mathcal{H}_j\right), &\quad 0 \leq l \leq n_{ij}, \end{aligned} \tag{6.3}$$

with an edge weight ϑ_{ij}^{l} resulting from certain quality criteria such as described below. The number of shared hypotheses between vertices $\mathcal{V}_i$ and $\mathcal{V}_j$ is represented by the variable n_{ij}, while the relative pose between view $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ is described by the 4×4 transformation matrix $\boldsymbol{T}_{ij}^{l}$. Given the model identity o_{ij}^{l} is shared amongst both views by hypotheses h_f^i and h_g^j, the transformation is estimated by

$$\boldsymbol{T}_{ij}^{l} = \boldsymbol{P}_f^i \left(\boldsymbol{P}_g^j\right)^{-1}, \tag{6.4}$$

and similarly for the transformation matrix corresponding to edge $\mathcal{E}_{ji}^{l}$,

$$T_{ji}^{l} = (\boldsymbol{T}_{ij}^{l})^{-1}. \tag{6.5}$$

If each vertex has a common object hypothesis with any other vertex, a fully-connected multi-view graph $\mathcal{G}$ can be described by

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}, \tag{6.6}$$

where $\mathcal{E}$ is the set containing all edges from Equation (6.3).

In order to avoid isolated vertices in $\mathcal{G}$ (e.g. no recognized object) or to possibly obtain a better pairwise transformation in case of weak object pose estimates for a pair of vertices, additional edges are created by means of visual features of the scene (*scene to scene* edges). In particular, for each pair of vertices $\{\mathcal{V}_i, \mathcal{V}_j\}$, their respective SIFT features [39] are matched using a *first nearest neighbour* strategy yielding a correspondence set between both frames, which is posteriorly processed by means of a correspondence grouping stage [4]. The output of the grouping stage is a set of geometrically consistent correspondences from which a rigid transformation is estimated. In our implementation, all consensus sets with more than 15 correspondences

are kept and used to create an edge between $\{\mathcal{V}_i, \mathcal{V}_j\}$ effectively extending $\mathcal{E}$. In order to experimentally motivate the creation of edges based on visual features, a small experiment has been conducted evaluating the frequency of edges in the Minimum Spanning Tree (MST) originating from this source. In particular, on the *Willow* and *TUW* datasets, *scene to scene* edges were selected 33.9% and 55.4% of the times, respectively. These results indicate that *scene to scene* edges play an important role during the reconstruction stage.

In its most general form, our method does not require the order of the sequence to be provided. However, if the order is known, significant speed ups can be obtained by avoiding creating edges between views that are too far away. In this work, we did not deploy any edge pruning scheme.

6.1.3 Edge weight and pairwise registration refinement

In order to favor edges in the MST representing a correct and accurate pairwise transformation, the edge weights associated with $\mathcal{E}$ need to be robust and representative for the quality of the estimated transformation. According to Equation (6.3), $\boldsymbol{T}^l_{ij}$ represents the transformation aligning $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$. To accommodate for small inaccuracies of the pair-wise pose estimate $\boldsymbol{T}^l_{ij}$ is refined by means of ICP [56] prior to the computation of the weight associated to the edge.

To assess the registration quality, a quality measure ω is proposed for the refined transformation. To evaluate registration of two point clouds originating from sensors with a single point of projection (such as the recent RGB-D sensors considered in this work), Huber and Hebert [29] introduced *visibility consistency* measures. For example, a *free space violation* (FSV) occurs when a point in $\boldsymbol{T}^l_{ij}\boldsymbol{S}_i$ blocks the visibility of another point in $\boldsymbol{S}_j$ from the sensor's origin of $\boldsymbol{S}_j$. Testing *free space violations* for all points in $\boldsymbol{S}_i$, the FSV fraction becomes

$$f^l_{ij} = \frac{\left|X_{FSV}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|}{\left|X_{FSV}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right| + \left|X_{SS}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|}, \tag{6.7}$$

where the number of points in $\boldsymbol{T}^l_{ij}\boldsymbol{S}_i$ with a free space violation and points on the same surface with respect to $\boldsymbol{S}_j$ are given by $\left|X_{FSV}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|$ and $\left|X_{SS}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|$, respectively. Intuitively, the lower f^l_{ij}, the better is the registration. For an in-depth discussion regarding the FSV fraction, please see [29].

Additionally to the FSV fraction, the computation of ω accounts for the amount of overlap as well as the angle between the normals of each corresponding point pair. In general, transformation estimations of clouds with high overlap are more stable and should therefore be included more often in the MST. While the absolute amount of overlap can be approximated by $\left|X_{SS}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|$, the relative overlap ζ is defined in the following by

$$\zeta^l_{ij} = \min\left(\frac{\left|X_{SS}\left(\boldsymbol{T}^l_{ij}\boldsymbol{S}_i, \boldsymbol{S}_j\right)\right|}{|\boldsymbol{S}_i|}, \zeta_{\max}\right), \tag{6.8}$$

where the parameter $\zeta_{\max}$ indicates the desired amount of overlap between clouds (0.75 in our experiments). The normals' similarity is defined by

$$\psi_{ij}^{l} = \frac{\sum_{\boldsymbol{p} \in \boldsymbol{S}_i} \mathrm{n}\left(\boldsymbol{p}\right) \cdot \mathrm{n}\left(\Gamma\left(\boldsymbol{T}_{ij}^{l}\boldsymbol{p}, \boldsymbol{S}_j\right)\right)}{|\boldsymbol{S}_i|}, \tag{6.9}$$

where $\mathrm{n}(\boldsymbol{p})$ represents the normal vector of point $\boldsymbol{p}$ and $\Gamma\left(\boldsymbol{p}, \boldsymbol{S}_j\right)$ is the *nearest neighbour* of $\boldsymbol{p}$ in view $\boldsymbol{S}_j$.

Combining the previous equations, ω is computed by

$$\omega_{ij}^{l} = \left(1 - f_{ij}^{l}\right) \zeta_{ij}^{l} \psi_{ij}^{l}, \quad 0 \leq \omega_{ij}^{l} \leq \zeta_{\max}. \tag{6.10}$$

Finally, the edge weight is

$$\vartheta_{ij}^{l} = \zeta_{\max} - \min\left(\omega_{ij}^{l}, \omega_{ji}^{l}\right). \tag{6.11}$$

6.1.4 Hypotheses projection and scene reconstruction

Given the graph $\mathcal{G}$ with the edge weights assigned in Subsection 6.1.3, a subgraph $\mathcal{G}'$ is created that connects all vertices $\mathcal{V}$ without cycles and with the lowest total cost in terms of the Prim's Minimum Spanning Tree algorithm [53]

$$\mathcal{G}' = \left\{\mathcal{V}, \mathcal{E}'\right\}, \quad \mathcal{E}' \subset \mathcal{E}. \tag{6.12}$$

Arbitrarily selecting a root node of the MST (i.e., $\mathcal{V}_{\text{root}} \in \mathcal{V}$), a world coordinate system is set to the camera coordinate system of $\mathcal{V}_{\text{root}}$. Starting from $\mathcal{V}_{\text{root}}$ and traversing through $\mathcal{G}'$, the hypotheses set $\mathcal{H}_{\text{root}}$ is augmented by all hypotheses in the graph

$$\mathcal{H}_{\text{root}} \rightarrow \left\{\mathcal{H}'_k\right\}, \quad 1 \leq k \leq n, \tag{6.13}$$

where $\mathcal{H}'_k$ is the set of hypotheses $\mathcal{H}_k$ with pose matrices multiplied iteratively by all the edge transformation matrices from node $\mathcal{V}_k$ to the root. Applying a similar procedure to all n point clouds $\boldsymbol{S}_k$ in the sequence, a 3D reconstruction of the scene, $\boldsymbol{S}$, is obtained:

$$\boldsymbol{S} = \left\{\boldsymbol{T}_k \boldsymbol{S}_k\right\}, \quad 1 \leq k \leq n \tag{6.14}$$

where $\boldsymbol{T}_k$ denotes the transformation bringing the k^{th} frame to the world coordinate system. Even though, the pairwise registration is in general accurate, small errors get accumulated after concatenating a few transformations. To reduce the overall registration error, these errors can be corrected by means of a global registration stage that simultaneously optimizes the poses of all overlapping views. We used the method proposed by Fantoni et al. [20] to refine the transformations. Since distance transforms for large volumes result in a large memory footprint, finite differences are computed using appropriate nearest neighbour searches in an Octree structure. To speed up this process, the refinement is deployed solely with the 3D positions of the visual feature keypoints extracted before.

Figure 6.3: *Left*: Screenshot of the reconstructed scene without filtering; several artefacts are observable due to axial and lateral noise. *Right*: Artefacts are removed after filtering points by means of a suitable noise model, providing a better representation for the verification stage.

Figure 6.4: 3D+RGB Hypothesis Verification; Left: reconstructed point cloud $\boldsymbol{S}$, Middle: extended hypothesis set $\mathcal{H}_{\text{root}}$, Right: selected subset $\mathcal{H}_{\text{verified}} \subset \mathcal{H}_{\text{root}}$ by the verification stage. The unrecognized bottles are not in the training set.

6.1.5 Multi-view GHV

The previous stages result in a set of hypotheses $\mathcal{H}_{\text{root}}$ (obtained by transforming hypotheses generated in single views to a global coordinate system) and a reconstructed scene point cloud $\boldsymbol{S}$ (obtained by registering the different frames in the sequence). Since $\mathcal{H}_{\text{root}}$ might contain wrong or redundant hypotheses, the following stage aims at selecting a subset of $\mathcal{H}_{\text{root}}$ consistent with $\boldsymbol{S}$ (see Figure 6.4). To obtain the best hypothesis subset, the GHV framework method proposed in this thesis is extended to handle multiple vantage points. Because RGB-D sensors present several artefacts that become evident once several clouds are merged together, we apply the RGB-D noise model of Nguyen et al. [49] in order to improve the reconstructed scene $\boldsymbol{S}$ (see Figure 6.3) before the verification stage.

Since the verification stage was originally designed to be deployed on 3D data and does not exploit the grid structure available in RGB-D data (except for reasoning about visible and occluded model points), the multi-view extension turns out to be straightforward. In particular, the definition of visible model points ought to be changed. Thus, for the multi-view case, a model point $\boldsymbol{q} = (q_x, q_y, q_z)^{\mathrm{T}}$ is considered *visible* if it is visible in at least one of the original views reconstructing $\boldsymbol{S}$. Let $\boldsymbol{S}_k$ be a view in the sequence, $\boldsymbol{T}_k$ the transformation bringing $\boldsymbol{S}_k$ to $\boldsymbol{S}$ and $\boldsymbol{q\prime} = \boldsymbol{T}_k^{-1}\boldsymbol{q}$.

Given f, c_x, c_y (focal length and central projection points of the camera), the visibility $V(\boldsymbol{q\prime}, \boldsymbol{S}_k)$ of $\boldsymbol{q}$ in $\boldsymbol{S}_k$ is assessed by

$$V(\boldsymbol{q\prime}, \boldsymbol{S}_k) = \begin{cases} 1, & \text{if} \quad (q'_z - \delta) \leq p_z \\ 0, & \text{elsewhere,} \end{cases} \tag{6.15}$$

where $\boldsymbol{p} = (p_x, p_y, p_z)^{\mathrm{T}} = \boldsymbol{S}_k(u, v)$ represents a point in $\boldsymbol{S}_k$ located at $(u, v) = \left(\frac{fq_x}{q_z} + c_x, \frac{fq_y}{q_z} + c_y\right)$ in the grid structure of the original frame $\boldsymbol{S}_k$, and δ defines a small threshold (3 millimetres) representing the inaccuracy of the data. Thus, $\boldsymbol{q}$ is a *visible* model point in $\boldsymbol{S}$ if $V(\boldsymbol{q\prime}, \boldsymbol{S}_k) = 1$ for any of the original views $\boldsymbol{S}_k \in \mathcal{S}$.

6.1.6 Ground truth annotation: Back-projection to each view

The verification stage results in a verified hypotheses set $\mathcal{H}_{\text{verified}} \subset \mathcal{H}_{\text{root}}$. By means of the respective transformation, these hypotheses can be transferred to the original views and thereby generate "ground truth" annotations for each individual RGB-D image. For instance, the pose of $h_j^{\text{root}} \in \mathcal{H}_{\text{verified}}$ in the k-th frame is given by $\boldsymbol{T}_k^{-1}\boldsymbol{P}_j^{\text{root}}$, where $\boldsymbol{P}_j^{\text{root}}$ represents the pose of the object associated with h_j^{root} in the global coordinate system.

6.1.7 Assumptions

In order for the generated annotations to be complete (all frames annotated) and meaningful (objects annotated with a correct pose), the following assumptions need to hold for the sequence under consideration:

1. The multi-view graph $\mathcal{G}$ contains a single connected component and all edges included in the Minimum Spanning Tree provide an accurate pairwise alignment.

2. Each object (from those in our model library) in the sequence needs to be recognized with the correct pose in at least one frame.

6.2 Results

To demonstrate the performance of the proposed method on real scenarios, we have performed several experiments on three multi-view RGB-D datasets. The availability of ground truth annotations for one of the datasets, allows us to compare automatic annotations provided by our method with the original ones obtained by means of fiducial patterns. For the other two datasets, totalling more than 500 RGB-D frames, we manually verified and when necessary, corrected automatic annotations in order to provide valuable data to the community.

6.2.1 Multi-view datasets

The first two datasets, *Willow* and *Challenge*, respectively contain 24 and 39 sequences of RGB-D frames of a turn-table with several object instances (as well as impostors for *Willow*) on top of it. The training set is composed of 35 models including common textured household objects. Test sequences on *Willow* contain between 11 and 19 frames inducing strong occlusions for some object instances. On the other hand, the objects in the *Challenge* sequences are in general not occluded and the number of frames ranges between 3 and 6. Because of the turn-table setup, the frames in these datasets were processed by first removing any point farther away than 1.5 meters with respect to the camera as well as points below the highest detected plane (i.e, the turn-table). This effectively allowing the algorithm to focus on the part of the data (objects on the table) we are interested in. Notice that even after such a preprocessing stage, some inconsistent data (moving differently than the table) remains unfiltered and thus, motivate the deployment of $\zeta_{\max}$ (desired amount of overlap) to quantify pairwise registration quality.

In order to show the performance of the method in more realistic scenarios (objects on top of each other, multiple supporting surfaces in form of tables or cabinets, high amounts of clutter, etc.), a third dataset, *TUW*, was acquired in our lab using a mobile robot. The model library is composed of 17 objects with different recognition relevant properties, e.g., textured and texture-less objects and geometrically common or unique. Instead of a turn-table setup, this dataset is obtained by moving the robot around a static environment. Statistics of the different datasets are summarized in Table 6.1.

Dataset	Sequences	Objects	Frames	Instances
Challenge	39	97	176	434
Willow	24	110	353	1628
TUW	15	162	163	1911

Table 6.1: Statistics of the multi-view datasets.

6.2.2 Evaluation of the generated "ground truth"

For the *Willow* and *Challenge* datasets, the method was able to detect all objects in the respectively 24 and 39 sequences and did not incur in a single false positive. Regarding pose accuracy, the method had as well an outstanding performance with only 3 sequence-wise inaccurate estimates. All inaccurate poses were related to infamous *object_19* (a specular, almost texture-less and symmetric object) and occurred due to the inability of single-view recognition to estimate an accurate pose in any of the frames composing the three sequences. While the rotation around the symmetry axis was not properly retrieved, the translation of the object was correctly estimated.

Since ground truth annotations were originally provided for the *Challenge* dataset, we performed a quantitative evaluation to compare the annotations provided by the

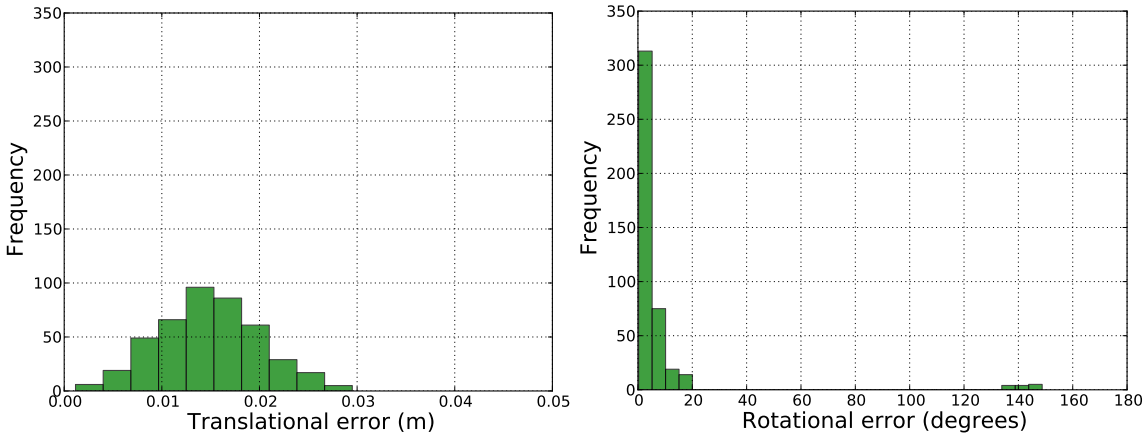


Figure 6.5: Translational and rotational errors for the Challenge Dataset, original annotations by means of fixtures and checkerboard detection versus our automatic annotations. Large rotational errors ($> 20°$) occur due to wrong pose estimates of the proposed method in two sequences where one of the assumptions is not fulfilled.

Figure 6.6: Inaccurate poses on the manual ground truth annotations for the Willow Challenge dataset. Left: original annotations by means of fixtures and checkerboard detection; Right: annotations obtained with our method.

proposed method and the original ground truth (we used the corrected annotations provided by [68])[1]. Figure 6.5 reports these results. Since errors were relatively large for visually pleasant annotations, we carefully analysed the original ground truth annotations to discover that the original poses were in some scenes significantly wrong, especially the translational component (for an example see Figure 6.6). Even though such errors significantly reduce the value of the provided evaluation, we can still observe that the estimated poses are *close* to those obtained by means of fiducial methods as well as single-view recognition methods [68] with excellent performance in the dataset. The errors and inaccuracies on the original annotations motivates even further the need for automating the process.

Regarding the more challenging *TUW* dataset, the method reported 1763 TPs, 0 FPs and 148 FNs, resulting in 100% precision and 92.26% recall. Sequence-wise,

[1] `http://rll.berkeley.edu/2013_IROS_ODP/`

11 objects out of 162 where not detected, resulting in 93.2% recall. Actual ground truth for this dataset was obtained by using the procedure presented in the upcoming section. Errors were mostly caused due to the inability of the single-view recognizer to detect the objects in any frame (assumption 2). Individual frame registration to the reconstructed scene was obtained for all sequences and thus, assumption 1 held. To visualize the automatic annotations obtained with our method, please checkout `goo.gl/qXkBOU`. Ground truth annotations, model library and training and test data are available at the same site.

6.2.3 Manual verification and correction

We have designed a small graphical tool to correct and extend the automatic annotations provided by our method. The tool is able to load the reconstructed scene $\boldsymbol{S}$ and the verified hypotheses $\mathcal{H}_{\text{verified}} \subset \mathcal{H}_{\text{root}}$. The tool provides the user a set of mechanisms to efficiently remove false positives, correct erroneous object poses, and add missing objects. Once the operator has finished, the corrected annotations are back-projected to the single frames as in Section 6.1.6. By means of automatic annotations and directly interacting with the reconstructed 3D scene, the process is greatly simplified.

Chapter 7

Conclusion

The problem of recognizing objects and estimating their pose in 3D scenes has been investigated throughout this thesis. We have addressed all necessary stages involved in the task: (i) acquisition of accurate 3D models from partial views of an object, (ii) generation of object hypotheses through the exploitation of different data modalities as well as different recognition paradigms and (iii) an hypothesis verification stage aiming at finding a globally consistent solution (in terms of the available recognition hypotheses) that best represents the scene under consideration.

In general, the recognition of objects and the estimation of their poses is a difficult problem. Its complexity increases even further in the case where different types of objects ought to be recognized while undergoing occlusions in complex scenes populated by clutter. However, in the context of robotic applications where the agent is confined to a specific environment, we have argued that the object recognition problem can be simplified by explicitly providing an accurate representation of what an object is (i.e, in terms of a 3D model). In addition, enabled by the different data modalities provided by recent RGB-D sensors as well as the availability of different recognition paradigms with complementary strengths, we have seen how the parallel deployment of multiple recognition pipelines increases the recognition capabilities of the overall system.

While the parallel deployment of recognition pipelines enabled the recognition of objects in a diverse range of situations, this choice also incurs in the generation of a large number of false detections. Notably, in cases where the recognition pipelines are *loosely* configured in order to handle challenging situations (i.e, highly occluded objects), false detections are highly problematic. Aiming at mitigating the negative effects of false detections while maintaining the benefits associated with the deployment of multiple pipelines, this thesis has addressed the hypothesis verification stage.

In particular, we have proposed a novel verification framework (GVH) resulting in the formalization of the hypothesis verification stage. Instead of each hypotheses being sequentially accepted or rejected based on different thresholds as usually done in the literature, our formulation involves the minimization of a global cost function aiming at finding a subset of object hypotheses that best explains the current scene while simultaneously considering the interaction between multiple hypotheses. This

cost function is composed of several geometrical cues (as well as appearance cues when color information is available) that enforce a plausible solution. GHV has been shown to be flexible enough to seamlessly integrate, in addition to object hypotheses, planar hypotheses. Because planar elements are commonly found in human-made environments, the ability to verify planar hypotheses provides on one hand a more exhaustive understanding of the scene and on the other hand, it allows to consider the interaction between object and planar hypotheses and thus aid in the verification of objects.

The deployment of multiple pipelines combined with the proposed verification stage has been shown to have an excellent performance on multiple datasets presenting heterogeneous traits. Notably, our method has been able to outperform the state-of-the-art on five of the six public benchmark datasets used to evaluate the system and resulted in ideal performance on three of them. Moreover, our recognition framework has been integrated in the robots being deployed by two European funded projects (HOBBIT and STRANDS) in order to increase their recognition capabilities.

Finally, in the last chapter of this thesis, we have shown how the proposed system can be used to automate the generation of ground-truth annotations for object recognition datasets composed of multi-view sequences of particular scene configurations. On one hand, this facilitates the creation of larger annotated datasets to evaluate recognition methods and motivates on the other hand, specially in the context of mobile robotics, the use of multiple vantage points to reduce ambiguities and challenging cases recurrent in single-view scenarios (e.g. highly occluded objects, objects being observed from undistinguishable feature-less viewpoints, etc.).

7.1 Outlook

Throughout this thesis we have seen how the proposed framework performs accurately in a wide range of recognition scenarios. Despite of these encouraging results, there are several remaining challenges that should be addressed in the near future. Applicable to this work as well as to the majority of successful methods in the literature, a major concern is related to their computational performance. Even though, there has been a dramatic improvement in this aspect in recent years (from processing time of a few minutes just some years ago to current methods being able to recognize complex scenes in just a few seconds), methods are still unable to reliably perform in real-time. This poses several challenges, specially in robotics or industrial scenarios, where computational efficiency is as critical as recognition accuracy.

A major improvement in this aspect could be obtained by exploiting the ability of current Graphic Processing Units (GPU's) to perform operations massively in parallel. We have already briefly mentioned in this work, how the hypothesis verification stage might be massively parallelised thanks to its linear dependency with the amount of hypotheses points being verified as well as most of the point operations within this stage being independent from one another.

An alternative to speeding up current algorithms by means of hardware accelera-

tion is represented by the deployment of simpler and faster single-view methods. The expected decline in recognition accuracy might be overcome by the exploitation of multiple vantage points. Unfortunately, recognition methods from multiple vantage points have been rarely investigated. This will require several research efforts, ideally in combination with other related areas such as best-view planning, key-frame selection, mapping and object search at larger scales.

Beside computational efficiency issues, other challenges need to be addressed. In particular, the current resolution provided by RGB-D sensors is insufficient for the recognition of small or thin objects (e.g. pencils, medicine packages, electronic devices, etc.). Moreover, some daily objects are made of materials from which data cannot be recovered (metal, transparent plastics or glass). In addition, the data coming from these sensors rapidly deteriorates as the distance to targets increases and thus renders the detection of far away objects (i.e., beyond two meters) very challenging. While sensing devices are beyond the area of expertise of the author and is therefore difficult to provide guidelines in this aspect, it is reasonable to expect improved recognition with better sensing capabilities. Another way to overcome issues with current sensing devices might be the exploration of data fusion. Similar to the exploitation of multiple recognition pipelines with complementary strengths, it might be possible to find a combination of sensors able to provide better data for a wider range of situations.

Bibliography

[1] Aitor Aldoma, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, Markus Vincze, and Gary Bradski. CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues. In *3DRR Workshop, ICCV*, 2011.

[2] Aitor Aldoma, Thomas Faeulhammer, and Markus Vincze. Automation of "Ground Truth" Annotation for Multi-View RGB-D Object Instance Recognition Datasets (accepted for publication). In *Proceedings of the 27th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[3] Aitor Aldoma, Z-C Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DoF Pose Estimation. *Robotics & Automation Magazine, IEEE*, 19(3):80–91, 2012.

[4] Aitor Aldoma, Federico Tombari, Luigi Di Stefano, and Markus Vincze. A global hypothesis verification method for 3D object recognition. In *European Conference on Computer Vision (ECCV)*, 2012.

[5] Aitor Aldoma, Federico Tombari, Johann Prankl, Andreas Richtsfeld, Luigi Di Stefano, and Markus Vincze. Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DoF pose estimation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2104–2111, 2013.

[6] Aitor Aldoma, Federico Tombari, R.B. Rusu, and Markus Vincze. OUR-CVFH: Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation. In *Joint DAGM-OAGM Pattern Recognition Symposium*, 2012.

[7] Aitor Aldoma and Markus Vincze. Pose Alignment for 3D Models and Single View Stereo Point Clouds Based on Stable Planes. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 374–380. IEEE, 2011.

[8] Marc Alexa, Johannes Behr, Daniel Cohen-or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on VCG*, 9:3–15, 2003.

[9] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *Trans. PAMI*, 1987.

[10] P. Bariya and K. Nishino. Scale-hierarchical 3D object recognition in cluttered scenes. In *Proc. CVPR*, 2010.

[11] Prabin Bariya, John Novatnack, Gabriel Schwartz, and Ko Nishino. 3D Geometric Scale Variability in Range Images: Features and Descriptors. *IJVC*, (2):232–255, 2012.

[12] Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[13] John Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov 1986.

[14] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724–2729 vol.3, Apr 1991.

[15] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011.

[16] Andrew Fitzgibbon Department and Andrew W. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. In *In British Machine Vision Conference*, pages 411–420, 2001.

[17] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *Proc. CVPR*, 2010.

[18] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D Rigid Body Transformations: A Comparison of Four Major Algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, March 1997.

[19] L. Di Stefano F. Tombari, A. Franchi. BOLD features to detect texture-less objects, 2013.

[20] S. Fantoni, U. Castellani, and A. Fusiello. Accurate and Automatic Alignment of Range Surfaces. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 73–80, 2012.

[21] Fred Glover and Claude McMillan. The general employee scheduling problem. An integration of MS and AI. *Computers & operations research*, 13(5):563–573, 1986.

[22] Jared Glover and Sanja Popovic. Bingham procrustean alignment for object detection in clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[23] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[24] Iryna Gordon and David G. Lowe. What and Where: 3D Object Recognition with Accurate Pose, 2006.

[25] Daniel Herrera C., Juho Kannala, and Janne Heikkil. Joint Depth and Color Camera Calibration with Distortion Correction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):2058–2064, 2012.

[26] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In KyoungMu Lee, Yasuyuki Matsushita, JamesM. Rehg, and Zhanyi Hu, editors, *Computer Vision ACCV 2012*, volume 7724 of *Lecture Notes in Computer Science*, pages 548–562. Springer Berlin Heidelberg, 2013.

[27] Dirk Holz, Alexander J B Trevor, Michael Dixon, Suat Gedikli, and Radu B Rusu. Fast segmentation of RGB-D images for semantic scene understanding.

[28] John Hopcroft and Robert Tarjan. Algorithm 447: Efficient Algorithms for Graph Manipulation. *Commun. ACM*, 16(6):372–378, June 1973.

[29] Daniel Huber and Martial Hebert. Fully Automatic Registration of Multiple 3D Data Sets. In *IEEE Computer Society Workshop on Computer Vision Beyond the Visible Spectrum(CVBVS 2001)*, December 2001.

[30] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *In Proc. UIST*, pages 559–568, 2011.

[31] Andrew E. Johnson and Martial Hebert. Surface matching for object recognition in complex three-dimensional scenes. *IVC*, (9), 1998.

[32] Andrew E. Johnson and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. PAMI*, (5), 1999.

[33] Tapas Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, Jul 2002.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, (4598), 1983.

[35] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *IEEE International Conference on on Robotics and Automation*, 2011.

[36] B. Leibe, A. Leonardis, and B. Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, May 2008.

[37] Ales Leonardis, Alok Gupta, and Ruzena Bajcsy. Segmentation of range images as the search for geometric parametric models. *IJCV*, 1995.

[38] Kok lim Low. Linear least-squares optimization for point-toplane ICP surface registration. Technical report, 2004.

[39] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[40] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[41] Mirko Maischberger. COIN-OR METSlib: a Metaheuristics Framework in Modern C++. `http://www.coin-or.org/metslib/docs/stable/0.5/metslib-tr.pdf`. Accessed: 2014-02-17.

[42] T. Masuda, K. Sakaue, and N. Yokoya. Registration and Integration of Multiple Range Images for 3-D Model Construction. In *Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I - Volume 7270*, ICPR '96, pages 879–, Washington, DC, USA, 1996. IEEE Computer Society.

[43] David Meger and JamesJ. Little. The ubc visual robot survey: A benchmark for robot category recognition. In Jaydev P. Desai, Gregory Dudek, Oussama Khatib, and Vijay Kumar, editors, *Experimental Robotics*, volume 88 of *Springer Tracts in Advanced Robotics*, pages 979–991. Springer International Publishing, 2013.

[44] A. Mian, M. Bennamoun, and R. Owens. 3D Model-based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Trans. PAMI*, (10), 2006.

[45] A. S. MIAN, M. BENNAMOUN, and R. A. OWENS. Automatic correspondence for 3d modeling: An extensive review. *International Journal of Shape Modeling*, 11(02):253–291, 2005.

[46] Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *VISAPP*. INSTICC Press, 2009.

[47] P.J. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on*, pages 130–139, Mar 1997.

[48] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[49] Chuong V. Nguyen, Shahram Izadi, and David Lovell. Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking. In *3DIMPVT*, pages 524–530. IEEE, 2012.

[50] Chavdar Papazov and Darius Burschka. An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In *Proc. 10th ACCV*, 2010.

[51] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Wörgötter. Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds. In *CVPR*, 2013.

[52] A. Petrelli and L. Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2244–2251, Nov 2011.

[53] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957.

[54] T. Rabbani, F.A. van den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. In *IEVM06*, 2006.

[55] A. Richtsfeld, Thomas Mörwald, Johann Prankl, Michael Zillich, and Markus Vincze. Segmentation of Unknown Objects in Indoor Environments. In *IROS*, 2012.

[56] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *3-D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001.

[57] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In *Proc. 23rd IROS*, 10/2010 2010.

[58] R.B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[59] Babak Taati and Michael Greenspan. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding*, 115(5):681 – 694, 2011. Special issue on 3D Imaging and Modelling.

[60] Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A Textured Object Recognition Pipeline for Color and Depth Image Data. In *In the proceedings of the International Conference on Robotics and Automation (ICRA)*, 2012.

[61] F. Tombari and L. Di Stefano. Hough voting for 3D object recognition under occlusion and clutter. *IPSJ Trans. on Computer Vision and Applications (CVA)*, 4:20–29, 2012.

[62] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of Histograms for local surface description. In *Proc. 11th ECCV*, 2010.

[63] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42, 2006.

[64] M. Ulrich, C. Wiedemann, and C. Steger. Combining Scale-Space and Similarity-Based Aspect Graphs for Fast 3D Object Recognition. *PAMI*, 34(10):1902–1914, 2012.

[65] T. Weise, T. Wismer, B. Leibe, , and L. Van Gool. In-hand Scanning with Online Loop Closure. In *IEEE International Workshop on 3-D Digital Imaging and Modeling*, October 2009.

[66] Walter Wohlkinger, Aitor Aldoma, Radu Bogdan Rusu, and Markus Vincze. 3DNet: Large-scale object class recognition from CAD models. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5384–5391, May 2012.

[67] Walter Wohlkinger and Markus Vincze. Shape Distributions on Voxel Surfaces for 3D Object Classification From Depth Images. In *IEEE International Conference on Signal and Image-Processing Applications*, 2011.

[68] Ziang Xie, Arjun Singh, Justin Uang, Karthik S. Narayan, and Pieter Abbeel. Multimodal Blending for High-Accuracy Instance Recognition. In *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[69] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1457–1464, Dec 2013.

[70] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696, Sept 2009.