



# Article Viewpoint Generation Using Feature-Based Constrained Spaces for Robot Vision Systems

Alejandro Magaña \*<sup>0</sup>, Jonas Dirr, Philipp Bauer <sup>10</sup> and Gunther Reinhart

Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

\* Correspondence: alejandro.magana@tum.com

Abstract: The efficient computation of viewpoints while considering various system and process constraints is a common challenge that any robot vision system is confronted with when trying to execute a vision task. Although fundamental research has provided solid and sound solutions for tackling this problem, a holistic framework that poses its formal description, considers the heterogeneity of robot vision systems, and offers an integrated solution remains unaddressed. Hence, this publication outlines the generation of viewpoints as a geometrical problem and introduces a generalized theoretical framework based on Feature-Based Constrained Spaces (C-spaces) as the backbone for solving it. A C-space can be understood as the topological space that a viewpoint constraint spans, where the sensor can be positioned for acquiring a feature while fulfilling the constraint. The present study demonstrates that many viewpoint constraints can be efficiently formulated as C-spaces, providing geometric, deterministic, and closed solutions. The introduced C-spaces are characterized based on generic domain and viewpoint constraints models to ease the transferability of the present framework to different applications and robot vision systems. The effectiveness and efficiency of the concepts introduced are verified on a simulation-based scenario and validated on a real robot vision system comprising two different sensors.

**Keywords:** viewpoint planning; 3D sensors; vision task automation; constraint planning; robot vision system

# 1. Introduction

The increasing performance of 2D and 3D image processing algorithms and the falling prices of electronic components (processors and optical sensors) over the last two decades have motivated not only researchers but also the industry to investigate and automate different machine vision tasks using robot vision systems (RVSs) consisting of a manipulator and a 2D or 3D sensor [1,2]. Whether programmed offline or online, RVSs demand multiple planning modules to execute motion and vision tasks efficiently and robustly. For instance, the efficient and effective planning of valid viewpoints to fulfill a vision task considering different constraints, known as the view(point) planning problem (VPP), still represents an open planning problem within diverse applications [2,3], e.g., camera surveillance, scene exploration, object detection, visual servoing, object reconstruction, image-based inspection, robot calibration, and mobile navigation [2,4].

# 1.1. Viewpoint Generation Problem Solved Using C-Spaces

To tackle the VPP, we first re-examine its reformulation and propose its modularization. Then, this study focuses on the most fundamental sub-problem of the VPP, i.e., the Viewpoint Generation Problem (VGP). The VGP addresses the calculation of valid viewpoints to acquire a single feature while considering the fulfillment of different viewpoint constraints.



Citation: Magaña, A.; Dirr, J.; Bauer, P.; Reinhart, G. Viewpoint Generation Using Feature-Based Constrained Spaces for Robot Vision Systems. *Robotics* **2023**, *12*, 108. https://doi.org/10.3390/ robotics12040108

Academic Editor: Xinjun Liu

Received: 16 June 2023 Revised: 8 July 2023 Accepted: 17 July 2023 Published: 26 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). With this in mind, this paper outlines the VGP as a purely geometrical problem that can be solved in the special Euclidean group denoted as SE(3) (6D spatial space), based on the concept of Feature-Based Constrained Spaces (C-spaces). C-spaces represent the spatial solution space of up to 6D of each viewpoint constraint  $c_i \in \tilde{C}$ , denoted as  $C_i$ , that comprises all valid sensor poses  $p_s$  to acquire a feature f. In other words, it can be assumed that any sensor pose lying within an i C-space fulfills the corresponding i viewpoint constraint. Hence, this solution space can be interpreted as an analytical [5], geometrical solution with an infinite set of valid viewpoints to satisfy the regarded viewpoint constraint. Moreover, the integration of multiple C-spaces spans the jointed C-space denoted as C, where all viewpoint constraints are simultaneously fulfilled. Figure 1 depicts a simplified representation of the VGP, C-spaces, and overview of the most relevant components.



**Figure 1.** Simplified, graphical representation of the Viewpoint Generation Problem (VGP): Which are valid sensor poses  $p_s$  to acquire a feature f considering a set of diverse viewpoint constraints  $\tilde{C}$ ? To answer this question, this study proposes the characterization of Feature-Based Constrained Spaces (C-spaces). The C-space denoted as C can be regarded as the geometrical representation of all viewpoint constraints in the special Euclidean group SE(3). Any sensor pose within it  $\forall p_s \in C$  can be considered to be valid to acquire a feature satisfying all viewpoint constraints  $\tilde{C}$ . The C-space is constituted by individual C-spaces  $C_i(c_i)$ , i.e., geometrical representations of each viewpoint constraint  $c_i \in \tilde{C}$ .

In this context, the most significant challenge behind the conceptualization of C-spaces lies in the generic and geometric formulation and characterization of diverse viewpoint constraints. Throughout this paper, we will use the term formulation refer to the formal, mathematical definition of an individual C-space; however, note that the characterization addresses the concrete implementation or computation of a C-space using a specific algorithm or method. This publication introduces nine C-spaces corresponding to different viewpoint constraints (i.e., sensor imaging parameters, feature geometry, kinematic errors, sensor accuracy, occlusion, multisensors, multi-features, and robot workspace) aligned to a consistent modeling framework to ensure their consistent integration.

#### 1.2. Related Work

Our study treats the VGP as a sub-problem of the VPP. Since most authors do not explicitly consider such a problem separation, this section provides an overview of related research that addresses the VPP in general. In a broader sense, the VPP can be even categorized as a sub-problem of a more popular challenge within robotics, that is, the coverage path planning problem [6].

Over the last three decades, the VPP has been investigated within a wide range of vision tasks that integrate an imaging device, but not necessarily a robot, and require the computation of generalized viewpoints. For a vast overview of the overall progress, challenges, and applications of the VPP, we refer to the various surveys [2,4,7–10] that have been published.

The approaches for viewpoint planning can be classified depending on the knowledge required a priori about the RVS to compute a valid viewpoint. Thus, a rough distinction can be made between model-based and non-model-based approaches [11].

#### 1.2.1. Model-Based

Most of the model-based viewpoint planning methods can be roughly differentiated between *synthesis* and *sampling-based* (related terms: *generate and test*) modeling approaches [5]. While synthesis approaches use analytical relationships to first characterize a continuous or discrete solution space before searching for an optimal viewpoint, sampling techniques are more optimization-oriented and compute valid viewpoints using a set of objective functions.

Since the present study seeks to characterize a solution space, i.e., a *C*-space, for each individual viewpoint constraint using analytical and geometrical relationships, the mathematical foundation of our framework can be classified as a model-based method following a synthesis approach. Hence, this section focuses mainly on the related literature following a similar approach.

Synthesis

Many of the reviewed publications considering model-based approaches have built their theoretical framework based on set theory to formulate either a continuous or discrete search space in a first step. Then, in a second step, optimization algorithms are used to find valid viewpoints within these search spaces and assess the satisfiability of the remaining constraints that were not explicitly considered.

The concept of characterizing such topological search spaces, in our work addressed as *C*-spaces (*related terms: viewpoint space, visibility map, visibility matrix, visibility volumes, imaging space, scannability frustum, configuration space, visual hull, search space*), has been proposed since the first studies addressing the VPP. Such a formulation has the advantage of providing a straightforward comprehension and spatial interpretation of the general problem.

One of the first and seminal studies that considered the characterization of a continuous solution space in  $\mathbb{R}^3$  can be attributed to the publication of [12]. In their work, the authors introduced a model-based method for 2D sensors, which synthesized analytical relationships to geometrically characterize a handful of constraints in terms of resolution, focus, field of view, visibility, view angle, occluding regions, and in later works [13], even constraints on the placement of a lighting source.

Based on the analytical findings provided by the previous work of [13], Ref. [7] introduced a model-based sensor planning system called *Machine Vision Planner* (MVP). On one hand, the MVP can be seen as a synthesis approach that characterizes a feature-based occlusion-free region using surface model decomposition [14,15]. On the other hand, the authors posed the problem in the context of an optimization setting using objective functions to find valid viewpoints within the occlusion-free space that meet imaging constraints.

The MVP was extended by [16] for its use with an industrial robot and moving objects. Their study addressed the drawbacks (non-linearity and convergence guarantee) of the optimization algorithms and opted to characterize 3D search spaces for the sensor's resolution, field of view, and workspace of the robot. Although the authors could not synthesize every constraint in the Euclidean space, they confirmed the benefits of solving the problem in  $\mathbb{R}^3$  instead of optimizing equations for finding suitable viewpoints. Furthermore, in a series of publications Reed et al. [17,18] extended some of the models introduced in the MVP and addressed the characterization of a search space in  $\mathbb{R}^3$  for range sensors, which integrates imaging, occlusion, and workspace constraints. Their study also proposed the synthesis of

an imaging space based on extrusion techniques of the surface models in combination with the imaging parameters of the sensor.

Another line of research within the context of model-based approaches follow the works of Tarbox and Gottschlich, which proposed the synthesis of a discretized search space using *visibility matrices* to map the visibility between the solution space and the surface space of the object. In combination with an efficient volumetric representation of the object of interest using octrees, Refs. [19,20] presented different algorithms based on the concept of visibility matrices to perform automated inspection tasks. The visibility matrices consider a discretized view space with all viewpoints lying on a tessellated sphere with a fixed camera distance. Analogously, under the consideration of further constraints, Refs. [11,21] introduced the measurability matrix extending the visibility matrix of Tarbox and Gottschlich to three dimensions. Within his work, Scott considered further sensor parameters, e.g., the shadow effect, measurement precision, and the incident angle, which many others have neglected. More recent works [3,22–24] confirmed the benefits of such an approach and used the concept of visibility matrices for encoding information between a surface point and a set of valid viewpoints.

In the context of space discretization and feature-driven approaches, further publications [20,25,26] suggested the characterization of the positioning space for the sensor using tessellated spheres to reduce the 6D sensor positioning problem to a 2D orientation optimization problem. Similarly, Refs. [27,28] introduced the concept of visibility maps to encode feature visibility mapped to a visibility sphere. Years later, Refs. [29,30] considered variations of this approach for their viewpoint planning systems. The major shortcomings of techniques considering a problem reduction is that most of them require a fixed working distance, which limits their applicability for other sensors and reduces their efficiency for the computation of multi-feature acquisition.

In the context of laser scanners, other relevant works, such as Refs. [31–33], also considered solutions to first synthesize a search space before searching for feasible solutions. Additionally, the publication of [34] needs to be mentioned, since, besides [21], this is one of the few studies that considered viewpoint visibility for multisensor systems using a lookup table.

Sampling-Based

Many other works [35–39] do not rely on the explicit characterization of a search space and assess the satisfiability of each viewpoint constraint individually by sampling the search space using metaheuristic optimization algorithms, e.g., simulated annealing or evolutionary algorithms. Such approaches concentrate on the adequate and efficient formulation of objective functions to satisfy the viewpoint constraints and find reasonable solutions.

# 1.2.2. Non-Model Based

In contrast, non-model based approaches require no a priori knowledge; the object can be utterly unknown to the planning system. In this case, online exploratory techniques based on the captured data are used to compute the next best viewpoint [26,40–43]. Most of these works focus on reconstruction tasks and address the problem as the *next-best-view* planning problem. Since our work is considered a feature-driven approach requiring a priori knowledge of the system, this line of research will be not further discussed.

#### 1.2.3. Comparison and Need for Action

Although over the last three decades, many works presented well-grounded solutions to tackle the VPP for individual applications, a generic approach for solving the VPP has not been established yet for commercial or industrial applications nor research. Hence, we are convinced that a well-founded framework, comprising a consistent formulation of viewpoints constraints combined with a model-based synthesis approach, while also considering a continuous solution space, has the greatest potential to search for viewpoints that efficiently satisfy different viewpoint constraints. Synthesis vs. Sampling: Within the related works, we have found recent publications following an explicit synthesis and sampling techniques of solution spaces for the same applications. Hence, a clear trend towards any of these model-based approaches could not be identified. On the one hand, sampling methods can be especially advantageous and computationally efficient within simple scenarios while considering few constraints. On the other hand, model uncertainties and nonlinear constraints are more difficult to model using objective functions and, within multi-feature scenarios, the computation efficiency can be severely affected. Therefore, we believe that this problem can be solved more efficiently based on C-spaces composed of explicit models of all regarded viewpoint constraints within applications comprising robot systems and partially known environments with modeling uncertainties.

Continuous vs. Discrete Space: Most of the latest research has followed a synthesis approach based on visibility matrices or visibility maps to encode the surface space and viewpoint space together for a handful of applications and systems [3,22–24,30]. Although these works have demonstrated the use of discrete spaces to be practical and efficient, from our point of view, its major weakness lies in the intrinsic limited storage capacity and processing times associated with matrices. This limitation directly affects the synthesis of the solution space considering just a fixed distance between sensor and object. Moreover, in the context of RVS, limiting the robot's working space seems to be in conflict with the inherent and most appreciated positioning flexibility of robots. Due to these drawbacks and taking into account that the fields of view and working distances of sensors have increased and will continue to do so, we believe that the discretization of the solution space could become inefficient for certain applications at some point.

Problem Formulation: Many of the revised works considering synthesized modelbased approaches posed the VGP formulation on the fundamentals of set theory. However, our research suggests that a consistent mathematical framework, which promotes a generic formulation and integration of viewpoint constraints, has not been appropriately placed. Hence, we consider that an exhaustive domain modeling and consistent theoretical mathematical framework are key elements to provide a solid base for a holistic and generic formulation of the VGP.

# 1.3. Outline

After providing an overview of the related work that has addressed the VPP and VGP in Section 1.2, first, Section 2 presents the domain models of generic RVSs used through the present study. Then, Section 3 poses the formulation VGP from another perspective and introduces the concept of C-spaces to tackle this problem. Section 4 exploits the introduced formulation, and different viewpoint constraints are formulated, characterized, and verified. Then, Section 5 assesses the validity of the proposed formulations and characterization of C-spaces and demonstrate its potential and generalization using a real RVS. Finally, Section 6 presents a summary and conclusions of this paper. In addition, a comprehensive set of supplemental material (e.g., simulation video, C-spaces meshes, renders) is digitally attached to this publication. An overview of the paper outline is presented in Figure 2.



Figure 2. Outline.

#### 1.4. Contributions

Our publication presents the fundamental concepts of a generic framework comprising innovative and efficient formulations to compute valid viewpoints based on C-spaces to solve the fundamental sub-problem of the VPP, i.e., the VGP. The key contributions of this paper are summarized as follows:

- Mathematical, model-based, and modular framework to formulate the VGP based on *C*-spaces and generic domain models.
- Formulation of nine viewpoint constraints using linear algebra, trigonometry, geometric analysis, and Constructive Solid Geometry (CSG) Boolean operations, in particular:
  - Efficient and simple characterization of *C*-space based on sensor frustum, feature position, and feature geometry.
  - Generic characterization of C-spaces to consider bi-static nature of range sensors extendable to multisensor systems.
- Exhaustive supporting material (surface models, manifolds of computed C-spaces, rendering results) to encourage benchmark and further development (see supplementary material).

Additionally, we consider the following principal advantages associated with the formulation of C-spaces:

- Determinism, efficiency, and simplicity: C-spaces can be efficiently characterized using geometrical analysis, linear algebra, and CSG Boolean techniques.
- Generalization, transferability, and modularity: C-spaces can be seamlessly used and adapted for different vision tasks and RVSs, including different sensor imaging sensors (e.g., stereo, active light sensors) or even multiple range sensor systems.
- Robustness against model uncertainties: Known model uncertainties (e.g., kinematic model, sensor, or robot inaccuracies) can be explicitly modeled and integrated while characterizing *C*-spaces. If unknown model uncertainties affect a chosen viewpoint, alternative solutions guaranteeing constraint satisfiability can be found seamlessly within *C*-spaces.

In combination with a suitable strategy, C-spaces can be straightforwardly integrated into a holistic approach for entirely solving the VPP. The use of C-spaces within an adequate strategy represents the second sub-problem of the VPP, which falls outside the scope of this paper and will be handled within a future publication.

# 2. Domain Models of a Robot Vision System

This section outlines the generic domain models and minimal necessary parameters of an RVS, including assumptions and limitations, required to characterize the individual C-spaces in Section 4.

We consider an RVS, a complex mechatronical system that comprises the following domains: a range *sensor* (s) that is positioned by a *robot* (r) to capture a *feature* (f) of an *object of interest* (o) enclosed within an *environment* (e). Figure 3 provides an overview of the RVS domains and some parameters described within this section.



Figure 3. Overview of the RVS domains and kinematic model.

# 2.1. General Notes

- **General Requirements** This paper follows a systematic and exhaustive formulation of the VGP, the domains of an RVS, and the viewpoint constraints to characterize *C*-spaces in a generic, simple, and scalable way. To achieve this, and similar to previous studies [11,33,36], throughout our framework the following general requirements (GR) are considered: generalization, computational efficiency, determinism, modularity and scalability, and limited a priori knowledge. The given order does not consider any prioritization of the requirements. A more detailed description of the requirements can be found in Table A1.
- **Terminology** Based on our literature research, we have found that a common terminology has not been established yet. The employed terms and concepts depend on the related applications and hardware. To better understand the relation of our terminology to the related work and in an attempt towards standardization, whenever possible, synonyms or related concepts are provided. Please note that in some cases, the generality of some terms is prioritized over their precision. This may lead to some terms not corresponding entirely to our definition; therefore, we urge the reader to study these differences before treating them as exact synonyms.
- **Notation** Our publication considers many variables to describe the RVS domains comprehensively. To ease the identification and readability of variables, parameters, vectors, frames, and transformations, we use the index notation given in Table A2. Moreover, all topological spaces are given in calligraphic fonts, e.g., V, P, I, C, while vectors, matrices, and rigid transformations are bold. Table A3 provides an overview of the most frequently used symbols.

# 2.2. General Models

**Kinematic model** Each domain comprises a *Kinematics* subsection to describe its kinematic relationships. In particular, all necessary rigid transformations (given in the right-handed system) are introduced to calculate the sensor pose. The pose p of any element is given by its translation  $t \in \mathbb{R}^3$  and a rotation component that can be given as a rotation matrix  $\mathbf{R} \in \mathbb{R}^{3x3}$ , *Z*-Y-X *Euler* angles  $\mathbf{r} = (\alpha^z, \beta^y, \gamma^x)^T$  or a quaternion  $\mathbf{q} \in \mathbb{H}$ . For readability and simplicity purposes, we use mostly the Euler angle representation throughout this paper. Considering the special orthogonal  $SO(3) \subset \mathbb{R}^{3x3}$ , the pose  $p \in SE(3)$  is given in the special Euclidean group  $SE(3) = \mathbb{R}^3 \times SO(3)$  [44]).  ${}^f p_s \in SE(3)$  in the feature's coordinate system  $B_f$ :

It is assumed that all introduced transformations are roughly known and given in the world (*w*) coordinate system or any other reference system. Moreover, we also consider a summed alignment error  $\epsilon_e$  in the kinematic chain to quantify the sensor's positioning inaccuracy relative to a feature.

**Surface model** A set of 3D surface models  $\kappa \in K$  characterizes the volumetric occupancy of all rigid bodies in the environment. The surface models are not always explicitly mentioned within the domains. Nevertheless, we assume that the surface model of any rigid body is required if this collides with the robot or sensor or impedes the sensor's sight to a feature.

# 2.3. Object

This domain considers an *object* (*o*) (related terms: object of interest, workpiece, artifact, measurement object, inspection object, or test object) that contains the features to be acquired.

- **Kinematics** The origin coordinate system of *o* is located at frame  $B_o$ . The transformation to the reference coordinate system is given in the world coordinate system  $B_w$  by  ${}^w_o T_e$ .
- **Surface Model** Since our approach does not focus on the object but rather on its features, the object may have an arbitrary topology.

#### 2.4. Feature

A *feature* (*f*) (related terms: region, point or area of interest, inspection feature, key point, entity, artifact) can be fully specified considering its kinematic and geometric parameters, i.e., frame  $B_f$  and set of surface points  $G_f(L_f)$ , which depend on a set of geometric dimensions  $L_f$ :

$$f := (B_f, G_f(L_f)). \tag{2}$$

**Kinematics** We assume that the translation  ${}^{o}t_{f}$  and orientation  ${}^{o}r_{f}$  of the feature's origin is given in the object's coordinate system  $B_{0}$ . In the case that the feature's orientation is given by its minimal expression, i.e., just the feature's surface normal vector  ${}^{o}n_{f}$ . The full orientation is calculated by letting the feature's normal to be the basis *z*-vector  ${}^{o}e_{f}^{z} = {}^{o}n_{f}$  and considering the rest basis vectors  ${}^{o}e_{f}^{x}$  and  ${}^{o}e_{f}^{y}$  to be mutually orthonormal. The feature's frame is given as follows:

$$B_f = {}^o T_f ({}^o t_f, {}^o r_f).$$
(3)

**Geometry** While a feature can be sufficiently described by its position and normal vector, a broader formulation is required within many applications. For example, dimensional metrology tasks deal with a more comprehensive catalog of geometries, e.g., edges, pockets, holes, slots, and spheres.

Thus, the present study explicitly considers the geometrical topology of a feature and a more extensive model of it [15,28]. Let the feature topology be described by a set of geometric parameters, denoted by  $L_f$ , such as the radius of a hole or a sphere or the lengths of a square.

**Generalization and Simplification** Moreover, we consider a discretized geometry model of a feature comprising a finite set of surface points corresponding to a feature  $g_f \in G_f$  with  $g_f \in \mathbb{R}^3$ . Since our work primarily focuses on 2D features, it is assumed that all surface points lie on the same plane, which is orthogonal to the feature's normal vector  ${}^on_f$  and co-linear to the *z*-axis of the feature's frame  $B_f$ .

Towards providing a more generic feature model, the topology of all features is approximated using a square feature with a unique side length of  $\{l_f\} \in L_F$  and five surface points  $g_{f,c}$ ,  $c = \{0, 1, 2, 3, 4\}$  at the center and at the four corners of the square. Figure 4 visualizes this simplification to generalize diverse feature geometries.



**Figure 4.** A square feature with the length  $l_f$  comprising five surface points  $g_{f,c}$  is used to generalize any feature topology, e.g., a circle, a slot, or a star (complex geometry).

## 2.5. Sensor

We consider a *sensor* (*s*) (related terms: range camera sensor, 3D sensor, imaging system) a self-contained acquisition device comprising at least two imaging devices  $\{s_1, s_2\} \in \widetilde{S}$  (e.g., two cameras or a camera and a lighting source) capable of computing a range image containing depth information. Such sensors can be classified by the principles used to acquire this type of depth information, e.g., triangulation, intensity, or time of flight [45]. The present work does not explicitly distinguish between these acquisition principles. Moreover, this subsection outlines a generic and minimal sensor model that is in line with our framework. Note that even though the present report focuses primarily on range sensors, the models can also be considered for single imaging devices.

**Kinematics** The sensor's kinematic model considers the following relevant frames:  $B_s^{TCP}$ ,  $B_s^{s_1}$ , and  $B_s^{s_2}$ . Taking into account the established notation for end effectors within the robotics field, we consider that the frame  $B_s^{TCP}$  lies at the sensor's tool center point (TCP). We assume that the frame of the TCP is located at the geometric center of the frustum space and that the rigid transformation  $\frac{ref}{TCP}T_s$  to a reference frame such as the sensor's mounting point is known.

Additionally, we consider that frame  $B_s^{s_1}$  lies at the reference frame of the first imaging device that corresponds to the imaging parameters  $I_s$ . We assume that the rigid transformation  ${}^{ref}_{s_1}T_s$  between the sensor lens and a known reference frame is also known.  ${}^{ref}_{s_2}T_s$  provides the transformation of the second imaging device at the frame  $B_s^{s_2}$ . The second imaging device  $s_2$  might be a second camera considering a stereo sensor or the light source origin in an active sensor system.

**Frustum space** The frustum space  $\mathcal{I}$ -space (related terms: visibility frustum, measurement volume, field-of-view space, sensor workspace) is described by a set of different sensor imaging parameters  $I_s$ , such as the depth of field  $d_s$  and the horizontal and vertical field of view (FOV) angles  $\theta_s^x$  and  $\psi_s^y$ . Alternatively, some sensor manufacturers may also provide the dimensions and locations of the near  $h_s^{near}$ , middle  $h_s^{middle}$ , and far  $h_s^{far}$  viewing planes of the sensor. The sensor parameters  $I_s$  allow only the topology of

the  $\mathcal{I}$ -space to be described. To fully characterize the topological space in the special Euclidean, the sensor pose  $p_s$  must be considered:

$$\mathcal{I}_{s} := \mathcal{I}_{s}(\boldsymbol{p}_{s}, \boldsymbol{I}_{s}) = \{ \boldsymbol{p}_{s} \in SE(3), \\ (\boldsymbol{d}_{s}, \boldsymbol{h}_{s}^{near}, \boldsymbol{h}_{s}^{far}, \boldsymbol{\theta}_{s}^{x}, \boldsymbol{\psi}_{s}^{y}) \in \boldsymbol{I}_{s} \}.$$

$$(4)$$

The  $\mathcal{I}$ -space can be straightforwardly calculated based on the kinematic relationships of the sensor and the imaging parameters. The resulting 3D manifold  $\mathcal{I}_s$  is described by its vertices  $V_k^{\mathcal{I}_s} := V_k(\mathcal{I}_s) = (V_k^x, V_k^y, V_k^z)^T$  with k = 1, ..., l and corresponding edges and faces. We assume that the origin of the frustum space is located at the TCP frame, i.e.,  $B_s^{\mathcal{I}_s} = B_s^{TCP}$ . The resulting shape of the  $\mathcal{I}$ -space usually has the form of a square frustum. Figure 5 visualizes the frustum shape and the geometrical relationships of the  $\mathcal{I}$ -space.



**Figure 5.** Detailed kinematic and imaging model of the sensor in the *x*-*z* plane. The frustum space  $\mathcal{I}_s$  is spanned by the imaging parameters of the sensor  $(d_s, h_s^{near}, h_s^{far}, \phi_s^x, \psi_s^y) \in I_s$  considering a sensor pose  $p_s$ . The  $\mathcal{I}$ -space is described by a minimum of eight vertices  $V_{1-8}^{\mathcal{I}_s}$  (note that in this 2D view the vertices 5–8 lie on the far *x*-*z* plane and that the FOV angle  $\psi_s^y$  is not illustrated).

- **Range Image** A range image (related terms: 3D measurement, 3D image, depth image, depth maps, point cloud) refers to the generated output of the sensor after triggering a measurement action. A range image is described as a collection of 3D points denoted by  $g_s \in \mathbb{R}^3$ , where each point corresponds to a surface point of the measured object.
- **Measurement accuracy** The measurement accuracy depends on various sensor parameters and external factors and may vary within the frustum space [21]. If these influences are quantifiable an accuracy model can be considered within the computation of the C-space. For example, ref. [34] proposed a method based on a Look-Up Table to specify quality disparities within a frustum.
- **Sensor Orientation** When choosing the sensor pose for measuring an object's surface point or a feature, additional constraints must be fulfilled regarding its orientation. One fundamental requirement that must be satisfied to guarantee the acquisition of a surface point is the consideration of the incidence angle  $f \varphi_s$  (related terms: inclination, acceptance, view, or tilt angle). This angle is expressed as the angle

between the feature's normal  $n_f$  and the sensor's optical axis (*z*-axis)  $e_s^z$  and can be

$${}^{f}\varphi_{s}^{max} > |{}^{f}\varphi_{s}|, {}^{f}\varphi_{s} = \arccos\left(\frac{\boldsymbol{n}_{f} \cdot \boldsymbol{e}_{s}^{z}}{|\boldsymbol{n}_{f}| \cdot |\boldsymbol{e}_{s}^{z}|}\right).$$
(5)

The maximal incidence angle  $f \varphi_s^{max}$  is normally provided by the sensor's manufacturer. If the maximal angle is not given in the sensor specifications, some works have suggested empirical values for different systems. For example, ref. [46] propose a maximum angle of 60° [47] suggests 45°, while [48] propose a tilt angle of 30° to 50°. The incidence angle can also be expressed on the basis of the Euler angles (pan, tilt) around the *x*- and *y*-axes:  $f \varphi_s (f \beta_s^y, f \gamma_s^x)$ .

Furthermore, the rotation of the sensor around the optical axis is given by the Euler angle  $\alpha_s^z$  (related terms: swing, twist). Normally, this angle does not directly influence the acquisition quality of the range image and can be chosen arbitrarily. Nevertheless, depending on the lighting conditions or the position of the light source while considering active systems, this angle might be more relevant and influence the acquisition parameters of the sensor, e.g., the exposure time. Additionally, if the shape of the frustum is asymmetrical, the optimization of  $\alpha_s^z$  should be considered.

#### 2.6. Robot

calculated as follows:

The robot (related terms: manipulator, industrial robot, positioning device) has the main task of positioning the sensor to acquire a range image.

**Kinematics** The robot base coordinate frame is placed at  $B_r$ . We assume that the rigid transformations between the robot basis and the robot flange,  ${}_{f_r}^r T_r$ , and between the robot flange and the sensor,  ${}_{s}^{f_r} T_r$ , are known. We also assume that the Denavit-Hartenberg (DH) parameters are known and that the rigid transformation  ${}_{f_r}^r T_r(DH)$  can be calculated using an inverse kinematic model. The sensor pose in the robot's coordinate system is given by

$${}^{r}\boldsymbol{p}_{s} = {}^{r}_{s}\boldsymbol{T}_{r} = {}^{r}_{f_{r}}\boldsymbol{T}_{r} \cdot {}^{f_{r}}_{s}\boldsymbol{T}_{r}. \tag{6}$$

The robot workspace is considered to be a subset in the special Euclidean, thus  $W_r \subseteq SE(3)$ . This topological space comprises all reachable robot poses to position the sensor  ${}^r p_s \in W_r$ .

**Robot Absolute Position Accuracy** It is assumed that the robot has a maximal absolute pose accuracy error of  $\epsilon_r$  in its workspace and that the robot repeatability is much smaller than the absolute accuracy; hence, it is not further considered.

#### 2.7. Environment

The environment domain comprises models of remaining components that were not explicitly included by other domains. Particularly, we consider all other rigid bodies that may collide with the robot or affect the visibility of the sensor, e.g., fixtures, external axes, robot cell components. Thus, if the environment domain comprises rigid bodies, these must be included in the set of surface models  $\kappa_e \in K$ .

#### 2.8. General Assumptions and Limitations

The previous subsections have introduced the formal models and parameters to characterize an RVS. Hereby, we present some general assumptions and limitations considered within our work.

• Sensor compatibility with feature geometry: Our approach assumes that a feature and its entire geometry can be captured with a single range image.

- **Range Image Quality:** The sensor can acquire a range image of sufficient quality. Effects that may compromise the range image quality and have not been previously regarded are neglected include measurement repeatability, lighting conditions, reflection effects, and random sensor noise.
- Sensor Acquisition Parameters: Our work does not consider the optimization of acquisition sensor parameters such as exposure time, gain, and image resolution, among others.
- **Robot Model:** Since we assumed that a range image can just be statically acquired, a robot dynamics model is not contemplated. Hence, constraints regarding velocity, acceleration, jerk, or torque limits are not considered within the scope of our work.

#### 3. Problem Formulation

This section first introduces the concept of generalized viewpoints and briefly describes the viewpoint constraints considered within the scope of our work. Then, the modularization of the VPP and formulation of the VGP as a geometric problem are introduced to understand the placement of the present study. In Section 3.5, *C*-spaces are introduced within the context of *Configuration Spaces* as a practical and simple approach for solving the VGP. Moreover, considering that various viewpoint constraints must be satisfied to calculate a valid viewpoint, we outline the reformulation of the VGP based on *C*-spaces within the framework of *Constraint Satisfaction Problems*.

#### 3.1. Viewpoint and V-Space

Although the concept of generalized viewpoints has been introduced by some of the related works (the concept of a generalized viewpoint was first introduced by [7] and was later used by [11,36]), there seems to be no clear definition of a viewpoint v. Hence, in this study, while considering a feature-centered formulation, we define a viewpoint as being a triple of the following elements: a sensor pose  $p_s \in SE(3)$  to acquire a feature  $f \in F$  considering a set of viewpoint constraints  $\tilde{C}$  from any domain of the RVS:

$$v := (f, p_s, C)$$

Additionally, we consider that any viewpoint that satisfies all constraints is an element of the viewpoint space (V-space):

$$v \in \mathcal{V}.$$
 (7)

Hence, the  $\mathcal{V}$ -space can be formally defined as a tuple comprising a feature space denoted by a feature set F and the  $\mathcal{C}$ -space  ${}^{F}\mathcal{C}(\widetilde{C})$ , which satisfies all spatial viewpoint constraints to position the sensor:

$$\mathcal{V} := (F, {}^{F}\mathcal{C}(\widetilde{C}))$$

Note that within this publication, we only consider spatially viewpoint constraints affecting the placement of the sensor. As given by the limitations of our work in Section 2.8, additional sensor setting parameters are not explicitly addressed. Nevertheless, for purposes of correctness and completeness, let these constraints be denoted by  $\tilde{C}_s$ , then Equation (7) can be extended as follows:

$$(f, p_s, C, C_s) \in \mathcal{V}.$$

#### 3.2. Viewpoint Constraints

To provide a comprehensive model of a generalized viewpoint and assess its validity, it is necessary to formulate a series of viewpoint constraints. Hence, we propose an abstract formulation of the viewpoint constraints needed to acquire a feature successfully. The set of viewpoint constraints  $c_i \in \tilde{C}$ , i = 1, ..., j comprises all constraints  $c_i$  affecting the positioning of the sensor; hence, the validity of a viewpoint candidate. Every constraint  $c_i$  can be regarded as a collection of domain variables of the RVS under consideration, e.g.,

the imaging parameters  $I_s$ , the feature geometry length  $l_f$ , the maximal incidence angle  $\int \varphi_s^{max}$ .

This subsection provides a general description of the constraints; a more comprehensive formulation and characterization are handled individually within Section 4. An overview of the viewpoint constraints considered in our work is given in Table A4.

Although some related studies [11,12,20,36] also considered similar constraints, the main differences in our formulations are found in their explicit characterization and integration with other constraints. While some of these works assess a viewpoint's validity in a reduced 2D space or sampled space, our work focuses on characterizing each constraint explicitly in a higher dimensional and continuous space.

#### 3.3. Modularization of the Viewpoint Planning Problem

The necessity to break down the VPP into two sub-problems can be better understood by considering the following minimal problem formulation based on [20]:

### **Problem 1.** How many viewpoints are necessary to acquire a given set of features?

We believe that considering a multi-stage solution to tackle the VPP can reduce its complexity and contribute to a more efficient solution. Thus, in the first step, we consider the modularization of the VPP and address its two fundamental problems separately: the VGP and the Set Covering Problem (SCP).

First, we attribute to the VGP the computation of valid viewpoints to acquire a single feature considering a set of various viewpoint constraints. Moreover, in the context of multi-feature scenarios and presuming that all features cannot be acquired using a single viewpoint, the efficient selection of more viewpoints becomes necessary to complete the vision task, with which arises a new problem, that is, the SCP.

This paper concentrates on the comprehensively formulation of the VGP; thus, this problem is discussed more extensively in the following sections. Although our work focuses on a feature-based approach, the concept of features can be also be extended to surface points or areas of interest.

#### 3.4. The Viewpoint Generation Problem

The VGP (related terms: optical camera placement, camera planning) and concept of viewpoints can be better understood considering a proper formulation:

# **Problem 2.** Which is a valid viewpoint v to acquire a feature f considering a set of viewpoint constraints $\widetilde{C}$ ?

A viewpoint v exists, if there is at least one sensor pose  $p_s$  that can capture a feature f and only if all j viewpoint constraints  $\tilde{C}$  are fulfilled. The most straightforward way to find a valid viewpoint for Problem 2 is to assume an ideal sensor pose  $p_s^0$  and assess its satisfiability against each constraint using a binary function  $h_i : (p_s^0, c_i) \rightarrow true$ . If the sensor pose fulfills all j constraints, the viewpoint is valid. Otherwise, another sensor pose must be chosen and the process must be repeated until a valid viewpoint is found. The mathematical formulation of such conditions is expressed as follows:

$$v = \{ (f, \boldsymbol{p}_{s}^{0}, \tilde{C}) \in \mathcal{V} \\ | f \in F, \boldsymbol{p}_{s}^{0} \in SE(3), \\ \bigcap_{c_{i} \in \tilde{C}} h_{i} : (\boldsymbol{p}_{s}^{0}, c_{i}) \to true \}.$$

$$(8)$$

The formulation of a generalized viewpoint as given by Equation (8) can be considered one of the most straightforward formulations to solve the VGP, if for each viewpoint constraint, a Boolean condition can be expressed. For instance, by introducing such cost functions for different viewpoint constraints, several works [9,21,36,37,49–51] demonstrated that optimization algorithms (e.g., greedy, genetic, or even reinforcement learning algorithms) can be used to find local and global optimal solutions within polynomial times.

#### 3.5. VGP as a Geometrical Problem in the Context of Configuration Spaces

Although the generalized viewpoint model as given by Equation (8) yields a sufficient and generic formulation to solve the VGP, this formulation is inefficient considering real applications with model uncertainties. System modeling inevitably involves discrepancies between virtual and real-world models, particularly within dynamically changing environments. Due to such model inconsistencies, considering optimal viewpoints to acquire a feature could be regarded as ineffective and inefficient in some applications. Hence, in our opinion, it is more reasonable to treat the VGP as a multi-dimensional problem and to consider multiple valid solutions throughout its formulation.

A sound solution for the VGP will require characterizing a continuous topological space comprising multiple solutions that allow deviation from an optimal solution and efficient choice of an alternative viewpoint. This challenge embodies the core motivation of our work to formulate and characterize C-spaces.

If the VGP can be handled as a spatial problem that can be solved geometrically, we refer to the use of configuration spaces, as introduced by [52,53] and exhaustively studied by [54] in the well-studied motion planning field for solving geometrical *path planning problems*.

"Once the configuration space is clearly understood, many motion planning problems that appear different in terms of geometry and kinematics can be solved by the same planning algorithms. This level of abstraction is therefore very important." [54]

In our work, we use the general concepts of configuration spaces based on the formulation of topological spaces to characterize the manifold spanned by a viewpoint constraint—the C-space. A C-space should not be confused with the configuration space (C-Space) used within the motion planning field to characterize the robot's joint configuration space.

#### 3.6. VGP with Ideal C-Spaces

Following the notation and concepts behind the modeling of configuration spaces, we first consider a modified formulation of Problem 2 and assume an ideal system (i.e., sensor with an infinite field of view, without occlusions and neglecting any other constraint) for introducing some general concepts.

# **Problem 3.** Which is the ideal C-space $C^*$ to acquire a feature f?

Sticking to the notation established within the motion planning research field, let us first consider an ideal, unconstrained space denoted as  $C^* \subseteq SE(3)$ 

$$C^* = \mathbb{R}^3 \times SO(3)$$
  
= { $p_s \in C^*, p_s \in SE(3)$ } (9)  
= { $p_s(t_s, r_s) \in C^*$   
|  $t_s \in \mathbb{R}^3, r_s \in SO(3)$ },

which is spanned by the Euclidean Space  $\mathbb{R}^3$  and the special orthogonal group SO(3), and holds all valid sensor poses  $p_s$  to acquire a feature f. An abstract representation of the unconstrained C-space  $C^*$  is visualized in Figure 6.



**Figure 6.** Abstract and simplified 2D representation of the ideal *C*-space  $C^*$  without viewpoint constraints; if viewpoint constraints are considered, the intersection of the corresponding *C*-spaces, e.g.,  $C_1$ ,  $C_2$ ,  $C_3$ , forms the *C*-space *C*.

# 3.7. VGP with C-Spaces

The ideal C-space as given by Equation (9) considers a sufficient generic model that spans an ideal solution space to solve the VGP. Assuming a non-ideal RVS where a viewpoint must satisfy a handful of requirements, an extended formulation of the C-space admitting viewpoint constraints is introduced within this subsection.

#### 3.7.1. Motivation

The VGP recalls the formulation of decision problems, a class of computational problems, which has been widely researched within different applications. Inspired by other research fields dealing with artificial intelligence and optimization of multi-domain applications, we observed that decision problems, including multiple constraints, can be well formulated under the framework of Constraint Satisfaction Problems (CSP) [55]. This category of problems does not consider an explicit technique to formulate the constraints under consideration. Moreover, a consistent, declarative, and simple representation of the domain's constraints can be decisive for their efficient resolution [56].

# 3.7.2. Formulation

Formulating the VGP as a CSP requires a proper formulation to consider viewpoint constraints in the first step; hence, let Problem Formulation 2 be extended by the following:

**Problem 4.** Which is the C-space C spanned by a set of viewpoint constraints  $\tilde{C}$  to acquire a feature f?

The C-space denoted as

$$\mathcal{C} := \mathcal{C}(f, \widetilde{C}) \tag{10}$$

can be understood as the topological space that all viewpoint constraints from the set  $\tilde{C}$  span in the special Euclidean group so that the sensor can capture a feature f fulfilling all of these constraints. The C-space that a single viewpoint constraint  $c_i \in \tilde{C}$  spans is analogously given:

$$\mathcal{C}_i := \mathcal{C}_i(f, c_i).$$

To guarantee a consistent formulation and integration of various viewpoint constraints, within our framework we consider the following characteristics for the formulation of C-spaces:

1. If an *i* constraint,  $c_i$ , can be spatially modeled, there exists a topological space denoted as  $C_i$ , which can be ideally formulated as a proper subset of the special Euclidean:

$$C_i \subseteq SE(3).$$

In a broader definition, we consider that the topological space for each constraint is spanned by a subset of the Euclidean Space denoted as  $\mathcal{T}_s \subseteq \mathbb{R}^3$  and a special orthogonal group subset given by  $\mathcal{R}_s \subseteq SO(3)$ . Hence, the topological space of a viewpoint constraint is given as follows:

$$C_{i} = \mathcal{T}_{s} \times \mathcal{R}_{s}$$

$$= \{ \boldsymbol{p}_{s} \in C_{i}, \boldsymbol{p}_{s} \in SE(3) \}$$

$$= \{ \boldsymbol{p}_{s}(\boldsymbol{t}_{s}, \boldsymbol{r}_{s}) \in C_{i}$$

$$\mid \boldsymbol{t}_{s} \in \mathcal{T}_{s}, \boldsymbol{r}_{s} \in \mathcal{R}_{s} \}.$$
(11)

- 2. If there exists at least one sensor pose in the *i* C-space  $\exists p_s \in C_i$ , then this sensor pose fulfills the viewpoint constraint  $c_i$  to acquire feature f; hence, a valid viewpoint exists  $(f, p_s, c_i) \in \mathcal{V}$ .
- 3. If there exists a topological space,  $C_i$ , for each constraint  $\forall c_i \in C$  then the intersection of all individual constrained spaces constitutes the joint C-space  $C \subseteq SE(3)$ :

$$C(\widetilde{C}) = \bigcap_{c_i \in \widetilde{C}} C_i(c_i).$$
(12)

4. If the joint constrained space is a non-empty set, i.e.,  $C \neq \emptyset$ , then there exists at least one sensor pose  $\exists p_s \in C$  and consequently a viewpoint  $(f, p_s, \tilde{C}) \in \mathcal{V}$  that fulfills all viewpoint constraints.

An abstract representation of the C-spaces and the resulting topological space C intersected by various viewpoint constraints is depicted in Figure 6. It is worth mentioning that although the framework considers an independent formulation of each viewpoint constraint, the real challenge consists of characterizing each constraint individually to maintain a high generalization and flexibility of the framework (cf. Table A1).

#### 4. Methods: Formulation, Characterization, and Verification of C-Spaces

This section outlines the formulation and characterization of the C-spaces of all viewpoint constraints considered in the scope of this paper (see Table A4). First, Section 4.1 introduces the core constraint, which is built up based on the imaging parameters of the sensor, which are characterized by the  $\mathcal{I}$ -space. Then, Section 4.2 shows how C-spaces can be combined to span a topological space in the special Euclidean group using different sensor orientations. We systematically analyze how the sensor frustum space can be used to describe a C-space in SE(3) and introduce simple and generic formulations for its computation.

While Sections 4.1 and 4.2 introduced the core constraints for characterizing C-spaces, the following Sections 4.3–4.8 deal with the characterization of the remaining viewpoint constraints. Moreover, Section 4.9 presents one possible strategy to integrate all C-spaces, demonstrating the advantages of a consistent and modular characterization.

The formulations presented in this section are motivated by the general requirements (cf. Table A1) that aim to deliver a high generalization of the models to facilitate their use with different RVSs and vision tasks. Hence, whenever possible, the characterization of some constraints using simple scalar arithmetic is prioritized over more complex techniques, and simplifications are introduced for the benefit of pragmatism, efficiency, and generalization of the approaches considered.

#### 4.1. Frustum Space, Feature Position, and Fixed Sensor Orientation

This section shows how the feature position and the frustum space  $\mathcal{I}$ -space can be directly employed to characterize a  $\mathcal{C}$ -space, which fulfills the first viewpoint constraint for a fixed sensor orientation. For the benefit of the comprehension of the concepts introduced within this subsection, we consider a feature as just a single surface point with a normal vector.

#### 4.1.1. Formulation

**Base Constraint Formulation** In the first step, we introduce a simple condition for the first constraint,  $c_1 := c_1(g_{f,0}, p_s, \mathcal{I}_s)$ , which considers the feature (minimally represented by a surface point), and the frustum space, which is characterized by all imaging parameters and the sensor pose. Let  $c_1$  be fulfilled for all sensor poses  $\forall p_s \in SE(3)$ , if and only if the feature surface point lies within the corresponding frustum space at the regarded sensor pose:

$$c_1 \iff g_{f,0} \in \mathcal{I}_s(p_s).$$
 (13)

Problem Simplification with fixed Sensor Orientation Due to the limitations of some sensors regarding their orientation, it is a common practice within many applications to define and optimize the sensor orientation beforehand. Then, the VGP can be reduced to an optimization of the sensor position  $t_s$ . Hence, let condition (13) be reformulated to consider a fixed sensor orientation  $r_s^{fix} \in SO(3)$  and to be true for all sensor positions  $\forall t_s \in \mathbb{R}^3$  that fulfill following condition:

$$c_1 \iff \boldsymbol{g}_{f,0} \in \mathcal{I}_s(\boldsymbol{p}_s(\boldsymbol{t}_s, \boldsymbol{r}_s = \boldsymbol{r}_s^{fix})).$$
 (14)

**Constraint Reformulation based on Constrained Spaces** Recalling the idea of geometrically characterizing any viewpoint constraint (see Section 3.7), we find that the viewpoint constraint formulation of Equation (14) to be unsatisfactory. We believe that this problem can be solved efficiently using geometric analysis and assume there exists a topological space denoted by  $C_1 := C_1(c_1)$ , which can be characterized based on the  $\mathcal{I}$ -space considering a fixed sensor's orientation. If such space exists, then all sensor positions within it fulfill the viewpoint constraint given by Equation (14).

Combining the formulation for C-spaces given by Equation (11) and the viewpoint constraint condition from Equation (14), the formal definition of the topological space  $C_1$  is given:

$$C_{1} = \{ \forall \boldsymbol{p}_{s}(\boldsymbol{t}_{s}, \boldsymbol{r}_{s} = \boldsymbol{r}_{s}^{JX}) \in C_{1} \\ | \forall \boldsymbol{t}_{s} \in \mathcal{T}_{s}, \boldsymbol{r}_{s}^{fix} \in \mathcal{R}_{s}, \boldsymbol{g}_{f,0} \in \mathcal{I}_{s} \}.$$

$$(15)$$

# 4.1.2. Characterization

Within the framework of our research, we found out that the manifold of  $C_1$  can be characterized in different ways. This subsection presents two possible solutions to characterize the *C*-space as given by Equation (15) using analytic geometry. The manifolds of the computed *C*-spaces and additional supporting material from this and the following subsections are found in the digital appendix of this publication.

1. Extreme Viewpoints Interpretation The simplest way to understand and visualize the topological space  $C_1$  is to consider all possible extreme viewpoints to acquire a feature f. These viewpoints can be easily found by positioning the sensor so that each vertex (corner) of the  $\mathcal{I}$ -space,  $V_k^{\mathcal{I}_s}$ , lies at the feature's origin  $B_f$ , which corresponds to the position of the surface point  $g_{f,0}$ . The position of such an extreme viewpoint corresponds to the k vertex  $V_k^{\mathcal{C}_1} \in V^{\mathcal{C}_1}$  of the manifold  $\mathcal{C}_1$ . Depending on the positioning frame of the sensor  $B_s^{TCP}$  or  $B_s^{s_1}$ , the space can be computed for the  $TCP(_{TCP}C_1)$  or the sensor lens  $(_{s_1}C_1)$ . The vertices can be straightforwardly computed following the steps given in Algorithm A1. The left Figure 7 illustrates the geometric relations for computing  $C_1$  and a simplified representation of the resulting manifolds in  $\mathbb{R}^2$  for the sensor TCP  $B_s^{TCP}$  and lens  $B_s^{s_1}$ .

2. Homeomorphism Formulation Note that the manifold  ${}_{ref}C_1$  illustrated in Figure 7a has the same topology as the  $\mathcal{I}$ -space. Thus, it can be assumed there exists a homeomorphism between both spaces such that  $h : \mathcal{I}_s \to \mathcal{C}_1$ . Letting the function h correspond to a point reflection over the geometric center of the frustum space, the vertices of the manifold  $V^{refC_1}$  can be straightforwardly estimated following the steps described in the Algorithm 1. The resulting manifold for the TCP frame is shown in Figure 7b.

**Algorithm 1** Homeomorphism Characterization of the Constrained Space  $C_1$ 

- (a) Consider a constant sensor orientation  $_{ref}r_s^{fix}$  to acquire a feature *f*.
- (b) Position the sensor reference frame at the feature's surface point origin  $_{ref} p_s^f (_{ref} t_s = B_f)$ .
- (c) For each *k* vertex of the frustum space, compute its reflection transformation *h* across the reference pivot frame  $B_s^{ref}$

$$_{ref}\boldsymbol{V}_{k}^{ref}\boldsymbol{\mathcal{C}}_{1}=h(\boldsymbol{V}_{k}^{\mathcal{I}_{s}}(_{ref}\boldsymbol{t}_{s}=B_{f}),B_{s}^{ref}).$$

(d) Connect all vertices from  $V^{ref}{}^{C_1}$  analogously to the vertices of the frustum space  $V^{\mathcal{I}_s}$  to obtain the  ${}_{ref}{}^{C_1}$  manifold.



(a) Extreme Viewpoint Formulation: Characterization of C-space  $C_1$  by positioning all vertices of  $\mathcal{I}_s$  at the feature's frame  $B_f$ .



(**b**) Homeomorphism Formulation: Characterization of C-space  ${}^{TCP}C_1$  by reflecting all vertices of  $\mathcal{I}_s$  around the feature's frame  $B_f$ .

**Figure 7.** Geometrical characterization of the C-space  $C_1$  using the frustum space with two different approaches.

**General Notes** We considered the steps described in Algorithm 1 to be the most traceable strategy using a homeomorphism to compute the constrained space  $_{ref}C_1$ . Nevertheless, we do not refuse any alternative approach for its characterization. For instance, the

same manifold of  $_{ref}C_1$  for any reference frame can also be obtained by first computing the reflection model of the frustum space  $\mathcal{I}_s^*$  over its geometric center at  $B_s^{\mathcal{I}_s}$ :

$$h: (B_s^{\mathcal{L}_s}, \mathcal{I}_s) \to \mathcal{I}_s^*.$$
(16)

The manifold of  $\mathcal{I}_s^*$  can then be simply translated to the desired reference frame so that  $B_s^{\mathcal{I}_s^*} = B_s^{ref}$ , considering that the TCP must be positioned at the origin of the feature  $_{TCP}\boldsymbol{p}_{s}^{f}(_{TCP}\boldsymbol{t}_{s}=B_{f},_{ref}\boldsymbol{r}_{s}^{fix}).$ 

Moreover, our approach considers that the topological space spanned by  $C_1$  exists if the following conditions hold:

- the frames of all vertices of the frustum space  $V_{s,i}(\mathcal{I}_s)$ , i = 1..j are known,
- the frustum space is a watertight manifold,
- and the space between connected vertices of the frustum space is linear; hence, adjacent vertices are connected only by straight edges.

Throughout this paper, we characterize most of the C-spaces considering just the reference frame for the sensor lens  $s_1$ ; hence, if not stated otherwise, consider  $C_1 := {}_{s_1}C_1$ .

# 4.1.3. Verification

Any of the two formulations presented in this subsection can be straightforwardly extended to characterize the C-space  $C_1$  in SE(3). However, we found the homeomorphism formulation to be the most practical way to compute the  $C_1$  manifold. Hence, to verify the characterization of  $C_1$  using this approach, we first defined a sensor orientation in SE(3)denoted as  $f_0 r_s^{fix}$  to acquire the feature  $f_0$ . We then computed the  $\mathcal{I}$ -space manifold using a total of j = 8 vertices with the imaging parameters of  $s_1$  from Table A6 and computed the reflected manifold of the  $\mathcal{I}$ -space  $\mathcal{I}_s^*$  as proposed by Equation (16). Some sensors might consider a variation of the FOV between the near-middle planes and the middle-far planes. Hence, in such cases 12 vertices would be necessary to characterize the frustum space correctly. If the slope between the near and far planes is constant, eight vertices are sufficient. In the next step, we transformed  $\mathcal{I}_{s}^{*}$  using the rigid transformation

$${}_{s_1}^{f} \boldsymbol{T} = {}_{TCP} \boldsymbol{p}_s^f ({}_{TCP} \boldsymbol{t}_s = B_{f_o, TCP} \boldsymbol{r}_s = {}^{f_0} \boldsymbol{r}_s^{fix})$$

to obtain the C-space manifold  $_{TCP}C_1$  for the sensor TCP frame and the transformation

 $\int_{s_1}^{f} T = \int_{T \subset P}^{f} T \cdot \int_{s_1}^{T \subset P} T$  to characterize the manifold  $\int_{s_1} C_1$  for the sensor lens frame. Figure 8 shows the resulting  $C_1$  manifolds considering different sensor orientations. The left Figure 8a visualizes the  ${}_{s_1}C_1$  and  ${}_{TCP}C_1$  manifolds considering the following orientation in *SE*(3):  ${}^{f_0}r_s^{fix}(\alpha_s^z = 170^\circ, \beta_s^y = 5^\circ, \gamma_s^x = 45^\circ)$ . On the other hand, Figure 8b visualizes the C-spaces just for the sensor lens considering two different sensor orientations.

To assess the validity of the characterized C-spaces, we selected eight extreme sensor poses lying at the vertices of each C-space manifold

$$\{p_{s,1}(t_s = V_1^{C_1}, r_s = {}^{f_0}r_s^{fix}), \dots, \\ p_{s,8}(t_s = V_8^{C_1}, r_s = {}^{f_0}r_s^{fix})\} \in C_1$$

and computed their corresponding frustum spaces  $\mathcal{I}_s(\boldsymbol{p}_{s,1}), \ldots, \mathcal{I}_s(\boldsymbol{p}_{s,8})$ . Our simulations confirmed that the feature  $f_0$  lay within the frustum space for all extreme sensor poses, hence satisfying the core viewpoint condition (14). Some exemplary extreme sensor poses and their corresponding  $\mathcal{I}$ -space are shown in the Figure 8. The rest of the renders, manifolds of the C-spaces, frames, and object meshes for this example and following examples can be found in the digital supplement material of this paper.

As expected, the computational efficiency for characterizing one C-space showed a good performance with an average computation time (30 repetitions) of 4.1 ms and





(a) C-spaces manifolds  ${}_{s_1}C_1({}^{f_0}r_s^{fix})$  (blue-filled) and  ${}_{TCP}C_1({}^{f_0}r_s^{fix})$  (transparent with blue edges) in SE(3) considering a fixed orientation  ${}^{f_0}r_s^{fix}(\alpha_s^z = 170^\circ, \beta_s^y = 5^\circ, \gamma_s^x = 45^\circ)$ .

(**b**) *C*-spaces manifolds considering two different sensor orientations  $C_1({}^{f_0}r_{s,1}(\alpha_s^z = 180^\circ, \gamma_s^x = 0^\circ, \beta_s^y = -25^\circ))$  and  $C_1({}^{f_0}r_{s,2}(\alpha_s^z = 180^\circ, \gamma_s^x = 0^\circ, \beta_s^y = 30^\circ)).$ 

**Figure 8.** Characterization of different C-spaces  $C_1(f_0, \mathcal{I}_s, f^0 r_s)$  (blue manifolds) in SE(3) considering different sensor orientations using the homeomorphism formulation. The  $\mathcal{I}$ -spaces (green manifolds) corresponding to different evaluated extreme viewpoints demonstrate that the feature  $f_0$  can be captured even from a sensor pose lying at the vertices of the C-space; hence, any sensor pose within the C-space  $p_s \in C_1$  can also be considered valid.

a standard deviation of  $\sigma = 2.4$  ms. The computation steps comprise a read operation of the vertices (hard-coded) of the frustum space as well as the required reflection and

transformation operations of a manifold with eight vertices.

#### 4.1.4. Summary

This subsection outlines the formulation and characterization of the fundamental C-space  $C_1$ , which is characterized based on the sensor imaging parameters, the feature position, and a fixed sensor orientation. Using an academic example, we demonstrated that any sensor pose (fix orientation) within  $C_1$  was valid to acquire the regarded feature satisfying the imaging sensor constraints. Moreover, two different strategies were proposed to efficiently characterize such a topological space based on fundamental geometric analysis.

The formulations and characterization strategies introduced in this subsection are considered the backbone of our framework. The potential and benefits of the core C-space  $C_1$  are exploited within the following subsections to consider the integration of additional viewpoint constraints.

# 4.2. Range of Orientations

In the previous subsection, the formulation of C-spaces for a fixed sensor orientation was introduced. Based on this formulation, this subsection outlines the formulation of a topological space in the special Euclidean group SE(3), which allows a variation of the sensor orientation.

Within the scope of our work, and taking into account applications that comprise an a priori model of the object and its features as well as the problem simplification addressed in Section 4.1, we consider it unreasonable and inefficient to span a configuration space that

comprises all orientations in  $\mathcal{R}_s \subseteq SO(3)$ . This assumption can be confirmed by observing Figure 8b, which demonstrates that the topological space, which will allow sensor rotations with an incidence angle of  $-25^\circ$  and  $30^\circ$ , does not exist.

For this reason, we consider it more practical to span a configuration space, which comprises a minimal and maximal sensor orientation range  $r_s^{min} \leq r_s \leq r_s^{max} \in \mathcal{R}_s^f$  instead of an unlimited space with all possible sensor orientations. The minimal and maximal orientation values can be defined considering the sensor limitations given by the second viewpoint constraint.

#### 4.2.1. Formulation

First, consider the range of sensor orientations with

$$\mathbf{r}_{s,m} \in R_s, \mathbf{r}_s^{min} \leq \mathbf{r}_{s,m} \leq \mathbf{r}_s^{max}, m = 1, \dots, n,$$

and let the C-space for a single orientation as given by Equation (15) be extended as follows:

$$\mathcal{C}_{2}(R_{s}) = \{ \boldsymbol{p}_{s}(\boldsymbol{t}_{s}, \boldsymbol{r}_{s,m}) \in \mathcal{C}_{2}(R_{s}) \\ \mid \boldsymbol{t}_{s} \in \mathcal{T}_{s'} \boldsymbol{g}_{f,0} \in \mathcal{I}_{s'}, \\ \boldsymbol{r}_{s,m} \in R_{s}, R_{s} \subseteq \mathcal{R}_{s}^{f} \}.$$

$$(17)$$

The topological space, which considers a range of sensor orientations, and denoted by  $C_2(R_s)$ , can be seamlessly computed by intersecting the individual configuration spaces for each *m* orientation:

$$\mathcal{C}_2(R_s) = \bigcap_{r_{s,m} \in R_s,}^n \mathcal{C}_1(\mathbf{r}_{s,m}, \mathcal{I}_s, B_f).$$
(18)

# 4.2.2. Characterization

The C-space  $C_2(R_s)$  as given by Equation (18) can be seamlessly computed using CSG Boolean Intersection operations. Considering that each intersection operation yields a new manifold with more vertices and edges, it is well known that the computation complexity of CSG operations increases with the number of vertices and edges. Thus, to compute  $C_2(R_s)$  in a feasible time, a discretization of the orientation range must be first considered.

One simple and pragmatic solution is to consider different sensor orientations comprising the maximal and minimal allowed sensor orientations, e.g.,  $(\mathbf{r}_s^{min}, \mathbf{r}_{s,}^{ideal}, \mathbf{r}_s^{max}) \in R_s$ . Figure 9a illustrates the 2D  $C_1$  manifolds of five different sensor orientations  $\{-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ\} \in \beta_s^y$ using a discretization step of  $\mathbf{r}_s^d = 10^\circ$  for the positioning frames TCP  $_{TCP}C_2(R_s)$  and sensor lens frame  $_{s_1}C_2(R_s)$ . The  $C_2$  manifolds are characterized by intersecting all individual spaces  $C_1$  as given by Equation (18).

As can be observed from Figure 9b,c, it should be noted that the manifolds  ${}_{s_1}C_2$ and  ${}_{TCP}C_2$  span different topological spaces (here in SE(1)) depending on the selected positioning frame. Contrary to the C-spaces  ${}_{s_1}C_1$  and  ${}_{TCP}C_1$  considering a fixed sensor position, the selection of the reference positioning frame should be considered before computing  $C_2$  and taking into account the explicit requirements and constraints of the vision task.



**Figure 9.** (a) Characterization of C-spaces in  $\mathbb{R}^2_{s_1}C_2(R_s)$  and  $_{TCP}C_2(R_s)$  for different positioning frames with the following range of sensor orientations  $\{-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ\} \in \beta_s^y$ . (b) Verification of two different viewpoints using sensor lens at positioning frame  $\{_{s_1}^{f_0}\boldsymbol{p}_{s,1}(\beta_s^y = 20^\circ), _{s_1}^{f_0}\boldsymbol{p}_{s,2}(\beta_s^y = -20^\circ)\} \in {}_{s_1}C_2(R_s)$ . (c) Verification of two different viewpoints using sensor TCP as positioning frame  $\{_{TCP}^{f_0}\boldsymbol{p}_{s,1}(\beta_s^y = 20^\circ), _{TCP}^{f_0}\boldsymbol{p}_{s,2}(\beta_s^y = -20^\circ)\} \in {}_{TCP}C_2(R_s)$ .

**Discretization without Interpolation** Note that  $C_2(R_s)$  spans a topological space that is just valid for the sensor orientations in  $R_s$  and that the sensor orientation  $r_s$  cannot be arbitrary chosen within the range  $r_s^{min} < r_s < r_s^{max}$ . This characteristic can be more easily understood by comparing the volume form, *Vol*, of the C-spaces  $C_2(r_s^{min}, r_s^{max})$  and  $C_2(r_s^{min}, r_s^{max})$ , which would show that the C-space  $C_2(r_s^{min}, r_s^{max})$  is less restrictive:

$$Vol(\mathcal{C}_2(\mathbf{r}_s^{min}, \mathbf{r}_s^{max})) > Vol(\mathcal{C}_2(\mathbf{r}_s^{min}, \mathbf{r}_s^{ideal}, \mathbf{r}_s^{max})).$$
(19)

This characteristic can particularly be appreciated in the top of the  ${}_{s_1}C_2(r_s^{min}, r_s^{max})$  manifold in Figure 9a. Thus, it should be kept in mind that the constrained space  $C_2(R_s)$  does not allow an explicit interpolation within the orientations of  $R_s$ .

**Approximation of**  $C_2$  However, as can be observed from Figure 9, the topological space spanned while considering a step size of 10°,  $C_2(R_s(r_s^d = 10^\circ))$ , is almost identical to the space if we would relax the step size to 20°,  $C_2(R_s(r_s^d = 20^\circ))$ . Hence, it can be assumed for this case that the *C*-spaces are almost identical and the following condition will hold:

$$C_2(R_s(r_s^d = 10^\circ)) \approx C_2(R_s(r_s^d = 20^\circ)).$$
 (20)

# 4.2.3. Verification

For verification purposes, we consider an academic example with the following sensor orientation ranges:  $\gamma_s^x = \{-5, 0, 5\}^\circ$ ,  $\beta_s^y = \{-5, 0, 5\}^\circ$ , and  $\alpha_s^z = \{-10, 0, 10\}^\circ$ . The resulting *C*-space,  $C_2(R_s)$ , was computed by intersecting the constrained space  $C_1(f_0, \mathcal{I}_{sr}, {}^f r_s)$  for each possible sensor orientation combination, i.e.,  $3^3 = 27$  sensor orientations (cf. Section 4.1). The computation time correspond to  $t(C_2(R_s)) = 15$  s. Figure A1 visualizes the 6D manifold of the *C*-space obtained through Boolean intersection operations. Additionally, the ideal constrained space considering a null rotation is also displayed to show a qualitative comparison of the reduction of the *C*-space considering a rotation space in *SE*(3).

For verifying the validity of the computed manifold, four extreme viewpoints and their corresponding frustum spaces are displayed in Figure A1 considering the following random orientations:

$${}^{f} \mathbf{r}_{s,1} (\gamma_{s}^{x} = 5^{\circ}, \beta_{s}^{y} = 0^{\circ}, \alpha_{s}^{z} = 10^{\circ}),$$
  
$${}^{f} \mathbf{r}_{s,2} (\gamma_{s}^{x} = 0^{\circ}, \beta_{s}^{y} = -5^{\circ}, \alpha_{s}^{z} = -10^{\circ}),$$
  
$${}^{f} \mathbf{r}_{s,3} (\gamma_{s}^{x} = -4^{\circ}, \beta_{s}^{y} = -5^{\circ}, \alpha_{s}^{z} = -8^{\circ}),$$
  
$${}^{f} \mathbf{r}_{s,4} (\gamma_{s}^{x} = 3^{\circ}, \beta_{s}^{y} = -3^{\circ}, \alpha_{s}^{z} = -9^{\circ}).$$

Note that while the first two viewpoints consider an explicit orientation within the given orientation range  $\{{}^{f}\mathbf{r}_{s,1}, {}^{f}\mathbf{r}_{s,2}\} \in R_s$ , the sensor orientation of the third and fourth viewpoints are not elements of  $\{{}^{f}\mathbf{r}_{s,3}, {}^{f}\mathbf{r}_{s,4}\} \notin R_s$ ; however, they lie within the interpolation range. The frustum spaces prove that all viewpoints can capture f satisfactorily. Although the sensor poses  $\mathbf{p}_{s,3}$  and  $\mathbf{p}_{s,4}$  as valid in this case, this assumption cannot be guaranteed for any other arbitrary orientation. Nevertheless, this confirms that the approximation condition as given by Equation (20) holds to some extent.

To provide a more quantifiable evaluation of this approximation, the constrained space,  $C_2$ , considering finer discretization steps  $r_s^d$  of 2.5° and 1° for  $\gamma_s^x$  and  $\beta_s^y$  was computed. The total number of computed manifolds corresponds to  $5 \times 5 \times 3 = 75$  *C*-spaces with  $t(C_2(2.5^\circ)) = 40 s$  for  $r_s^d = 2.5^\circ$  and  $11 \times 11 \times 3 = 363$  *C*-spaces with  $t(C_2(1^\circ)) = 720$  s for  $r_s^d = 2.5^\circ$ . The relative volumetric ratio between the computed spaces is given as follows:  $Vol(C_2(1^\circ))/Vol(C_2(2.5^\circ)) = 0.9999$  and  $Vol(C_2(1^\circ))/Vol(C_2(5^\circ)) = 0.9995$ . These experiments show that the differences between the manifold volume ratios for the selected steps are insignificant and that the approximation with a step of  $r_s^d = 2.5^\circ$  holds for this case.

**General Notes** It is important to note that the validity of the approximation introduced by Equation (20) must be individually assessed for each individual application, imaging parameters, and other constraints. Some preliminary experiments showed that when considering further viewpoint constraints that depend on the sensor orientation, e.g., the feature geometry (see Section 4.3), the differences between the spaces using different discretization steps may be more considerable. A more comprehensive analysis falls outside the scope of this study and remains to be further investigated. We urge the reader to perform some empirical experiments for choosing an adequate discretization step and a good trade-off between accuracy and efficiency.

#### 4.2.4. Summary

Contrary to the previously introduced C-space  $C_1$  which is limited to a fixed sensor orientation, this subsection outlined the formulation and characterization of the C-space  $C_2$  in SE(3) that satisfies the sensor imaging parameters for different sensor orientations. We demonstrated that the manifold  $C_2$  is straightforwardly characterized by intersecting multiple C-spaces with different sensor orientations.

#### 4.3. Feature Geometry

In many applications, the feature geometry is a fundamental viewpoint constraint that may considerably limit the space in SE(3) for positioning the sensor. This subsection shows that the required C-space affected by the feature geometry can be efficiently and explicitly characterized using trigonometric relationships that depend on the feature geometry, the sensor's FOV angles, and the sensor orientation.

#### 4.3.1. Formulation

Taking into account the third viewpoint constraint (see Section 2.8), it can be assumed that all feature surface points  $G_f(L_f)$  must be acquired simultaneously. The *C*-space that fulfills this requirement can be easily formulated by extending the base constraint of  $C_1$  as

given by Equation (15), considering that all surface points must lie within the frustum space:

$$\mathcal{C}_{3} = \{ \forall \mathbf{p}_{s}(\mathbf{t}_{s}, \mathbf{r}_{s}^{fix}) \in \mathcal{C}_{3}(\mathcal{C}_{1}, G_{f}(L_{f}), \mathbf{r}_{s}^{fix}, \mathcal{I}_{s}) \\ | \forall \mathbf{t}_{s} \in \mathcal{T}_{s}, \mathbf{r}_{s}^{fix} \in \mathcal{R}_{s}, G_{f}(L_{f}) \subseteq \mathcal{I}_{s} \}.$$

$$(21)$$

# 4.3.2. Generic Characterization

Taking into account the generic formulation of the *C*-space  $C_3$  from Equation (21), in the simplest case, it can be assumed that  $C_3$  could be obtained by scaling  $C_1$ . Let the required scaling vector be denoted by  $\Delta(\mathbf{r}_s^{fix}, L_f, \theta_s^x, \psi_s^y)$  and depend on the feature geometry, the sensor rotation, and the FOV angles of the sensor. The generic characterization of the  $C_3$  manifold can then be expressed as follows

$$\mathcal{C}_3 = \mathcal{C}_1(\boldsymbol{\Delta}(\boldsymbol{r}_s^{fix}, L_f, \theta_s^x, \boldsymbol{\psi}_s^y)).$$

Recalling that the  $C_1$  manifold is not symmetrical in all planes, hence, the rotation variation assumes that the  $C_3$  manifold cannot be correctly scaled regarding the same scaling vector. Thus, a more generic and flexible approach can then be considered by letting each k vertex of  $\forall V_{v,k}^{C_1} \in V_v^{C_1}$  be individually scaled. Hence, each vertex of  $C_3$  can be computed with

$$\boldsymbol{V}_{k}^{\mathcal{C}_{3}} = \boldsymbol{V}_{k}^{\mathcal{C}_{1}} - \boldsymbol{\Delta}_{k}(\boldsymbol{r}_{s}^{fix}, \boldsymbol{L}_{f}, \boldsymbol{\theta}_{s}^{x}, \boldsymbol{\psi}_{s}^{y}), \qquad (22)$$

considering the following generalized vector:

$$\boldsymbol{\Delta}_{k}(\boldsymbol{r}_{s}^{fix}, L_{f}, \theta_{s}^{x}, \psi_{s}^{y}) = \begin{pmatrix} \Delta_{k}^{x}(\boldsymbol{r}_{s}^{fix}, L_{f}, \theta_{s}^{x}, \psi_{s}^{y}) \\ \Delta_{k}^{y}(\boldsymbol{r}_{s}^{fix}, L_{f}, \theta_{s}^{x}, \psi_{s}^{y}) \\ \Delta_{k}^{z}(\boldsymbol{r}_{s}^{fix}, L_{f}, \theta_{s}^{x}, \psi_{s}^{y}) \end{pmatrix}.$$
(23)

The explicit characterization of the scaling vector from Equation (23) requires an individual and comprehensive trigonometric analysis of each k vertex of  $C_3$ , which depends on the chosen sensor orientation. Moreover, since the scaling vector  $\Delta_k$  depends also on the feature geometrical properties, from now on we will assume the generalization of the feature geometry as introduced in Section 2.4 to characterize any feature by a square of the length  $l_f$ . This simplification contributes to a higher generalization of our models for different topologies and facilitates the comprehension of the trigonometric relationships introduced in the following subsections.

#### 4.3.3. Characterization of the C-Space with Null Rotation

The most straightforward scenario to quantify the influence of the feature geometry on the constrained space is first to consider a null rotation,  ${}^{f}r_{s}^{0}$ , of the sensor relative to the feature, i.e., the feature's plane is parallel to the *xy*-plane of the TCP and the rotation around the optical axis equals zero,  ${}^{f}r_{s}^{0}(\gamma_{s}^{x} = \beta_{s}^{y} = \alpha_{s}^{z} = 0)$ .

First, span the core constrained space,  $C_1$ , considering the feature position and the null rotation of the sensor. Then, parting from one vertex of  $C_1$ , let the  $\mathcal{I}$ -space be translated in one direction until  $\mathcal{I}_s$  entirely encloses the whole feature. This step is exemplary, shown in the *x*-*z* plane in Figure 10 at the third vertex of  $C_1$ , and can be interpreted as an analogy to the *Extreme Viewpoints Interpretation* (cf. Section 1) used to span  $C_1$ . Then, it is easily understood that to characterize  $C_3$  all vertices  $\forall V_{v,k}^{C_1} \in V_v^{C_1}$  of  $C_1$  must be shifted in the x and y directions by a factor of  $0.5 \cdot l_f$ :

$$\Delta_{k}^{x}(\mathbf{r}_{s}^{0}, l_{f}) = 0.5 \cdot l_{f} 
\Delta_{k}^{y}(\mathbf{r}_{s}^{0}, l_{f}) = 0.5 \cdot l_{f} 
\Delta_{k}^{z}(\mathbf{r}_{s}^{0}, l_{f}) = 0.$$
(24)



**Figure 10.** Characterization of the *C*-space  $C_3$  considering a null rotation  ${}^f r_s^0$  over the feature *f*: scale all vertices of  $C_1$  in the *x*- and *y*-axes considering the feature geometric length of  $0.5 \cdot l_f$ .

# 4.3.4. Rotation around One Axis

Any other sensor orientation different from the null orientation requires an individual analysis of the exact trigonometric relationships for each vertex. To break down the complexity of this problem, within this subsection, we first provide the geometrical relationships needed to characterize the constrained space  $C_3$  while considering an individual rotation around each axis. The characterization of the constrained spaces follows the same approach described in the previous subsection, which requires first the characterization of the base constraint,  $C_1$ , and then the derivation of the scaling vectors.

 Rotation around z-axis α<sub>s</sub><sup>z</sup> ≠ 0 Assuming a sensor rotation around the optical axis, <sup>f</sup>r<sub>s</sub><sup>z</sup>(α<sub>s</sub><sup>z</sup> ≠ 0, φ<sub>s</sub>(β<sub>s</sub><sup>y</sup>, γ<sub>s</sub><sup>x</sup>) = 0) (see Figure 11), the *C*-space is scaled just along the vertical and horizontal axes, using the following scaling factors:

$$\Delta_{k}^{x}(\mathbf{r}_{s}^{z}, l_{f}) = \frac{l_{f}}{2} \cdot (\cos(|\alpha_{s}^{z}|) + \sin(|\alpha_{s}^{z}|)) 
\Delta_{k}^{y}(\mathbf{r}_{s}^{z}, l_{f}) = \frac{l_{f}}{2} \cdot (\cos(|\alpha_{s}^{z}|) + \sin(|\alpha_{s}^{z}|)) 
\Delta_{k}^{z}(\mathbf{r}_{s}^{z}, l_{f}) = 0.$$
(25)

The derivation of the trigonometric relationships from Equation (25) can be better understood by looking at Figure A2.



**Figure 11.** Characterization of the vertices of the *C*-space,  $C_3$ , considering a rotation around the optical axis  ${}^f r_s^z(\alpha_s^z \neq 0, \varphi_s(\beta_s^y, \gamma_s^x) = 0)$ .

Rotation around *x*-axis or *y*-axis (γ<sub>s</sub><sup>x</sup> ≠ 0⊻β<sub>s</sub><sup>y</sup> ≠ 0): A rotation of the sensor around the *x*-axis, *r<sub>s</sub>*(γ<sub>s</sub><sup>x</sup> ≠ 0, α<sub>s</sub><sup>z</sup> = β<sub>s</sub><sup>y</sup> = 0), or *y*-axis, *r<sub>s</sub>*(β<sub>s</sub><sup>y</sup> ≠ 0, α<sub>s</sub><sup>z</sup> = γ<sub>s</sub><sup>x</sup> = 0), requires deriving individual trigonometric relationships for each vertex of C<sub>3</sub>. Besides the feature length, other parameters such as the FOV angles (θ<sub>s</sub><sup>x</sup>, ψ<sub>s</sub><sup>y</sup>) and the direction of the rotation must be considered.

The scaling factors for the eight vertices of the *C*-space while considering a rotation around the *x*-axis or *y*-axis can be found in Table A5 regarding the following general auxiliary lengths for  ${}^{f}\mathbf{r}_{s}(\alpha_{s}^{z} = \gamma_{s}^{x} = 0, \beta_{s}^{y} \neq 0)$  (left) and for  ${}^{f}\mathbf{r}_{s}(\alpha_{s}^{z} = \beta_{s}^{y} = 0, \gamma_{s}^{x} \neq 0)$  (right):

$$\begin{split} \rho^{z,y} &= \frac{l_f}{2} \cdot \sin(|\beta_s^y|) & \rho^{z,x} &= \frac{l_f}{2} \cdot \sin(|\gamma_s^x|) \\ \rho^x &= \frac{l_f}{2} \cdot \cos(|\beta_s^y|) & \rho^y &= \frac{l_f}{2} \cdot \cos(|\gamma_s^x|) \\ \varsigma^x &= 2 \cdot \rho^{z,y} \cdot \tan(0.5 \cdot \theta_s^x) & \varsigma^y &= 2 \cdot \rho^{z,x} \cdot \tan(0.5 \cdot \psi_s^y) \\ \varsigma^{x,y} &= 2 \cdot \rho^{z,y} \cdot \tan(0.5 \cdot \psi_s^y) & \varsigma^{y,x} &= 2 \cdot \rho^{z,x} \cdot \tan(0.5 \cdot \theta_s^x) \\ \lambda^x &= \rho^x - \varsigma^x & \lambda^y &= \rho^y - \varsigma^y \\ \sigma^x &= \rho^x + \varsigma^x & \sigma^y &= \rho^y + \varsigma^y. \end{split}$$

The derivation of the trigonometric relationships can be better understood using an exemplary case. First, assume a rotation of the sensor around the *y*-axis of  $\beta_s^y > 0 \land \gamma_s^x = 0$ , as illustrated in Figure 12. The trigonometric relationships can then be derived for each vertex following the *Extreme Viewpoints Interpretation* as exemplary and depicted in Figure A3).



**Figure 12.** Characterization of the vertices of the *C*-space,  $C_3$ , considering a rotation around the *y*-axis  ${}^{f} \boldsymbol{r}_{s}^{y} (\gamma_{s}^{x} = \alpha_{s}^{z} = 0, \beta_{s}^{y} < 0)$ .

4.3.5. Generalization to 3D Features

Although our approach contemplates primary 2D features, the constrained space  $C_3$  can be seamlessly extended to acquire 3D features considering a feature height  $h_f \in L_f$ .

This paper only considers the characterization of the scaling vectors for concave (e.g., pocket, slot) and convex (e.g., cube, half-sphere) features with a null rotation of the sensor,  ${}^{f}r_{s}^{0}$ . For instance, the back vertices (k = 1, 2, 5, 6) of  $C_{3}$  to capture a concave feature, as shown in Figure 13, must be scaled using the following factors:

$$\Delta_k^x = h_f \cdot \tan(0.5 \cdot \theta_s^x) + 0.5 \cdot l_f$$
  

$$\Delta_k^y = h_f \cdot \tan(0.5 \cdot \psi_s^y) + 0.5 \cdot l_f$$
  

$$\Delta_k^z = h_f.$$
(26)

The front vertices are scaled using the same factors as for a 2D feature as given by Equation (24). For convex features, let all vertices be scaled with the factors given by Equation (26), except for the depth delta factor of the back vertices, which follows  $\Delta_k^z = 0$ .

Note that the characterization of the C-space,  $C_3$ , for considering 3D features just guarantees that the entire feature surface lies within the frustum space. We neglect any further visibility constraints that may influence the viewpoint's validity, such as the maximal angles for the interiors of a concave feature. Moreover, it should be noted that the scaling factors given within this subsection hold for just a null rotation. The characterization of the scaling factors for other sensors orientations can also be derived by extending the previously introduced relationships for 2D features in Section 4.3.4.



**Figure 13.** Characterization of the vertices of the *C*-space,  $C_3$ , considering a 3D feature and null rotation  $f \mathbf{r}_s^0$ .

# 4.3.6. Verification

The verification of the geometrical relationships introduced within this subsection was performed based on an academic example to acquire a square feature  $f_1$  and a 3D pocket feature  $f_1^*$ . The *C*-spaces for  $f_1$ , i.e.,  $C_{3,1}(f_1 r_{s,1})$  and  $C_{3,2}(f_1 r_{s,2})$ , consider a sensor orientation of  $f_1 r_{s,1}(\alpha_s^z = \beta_s^y = 0, \gamma_s^x = 30^\circ)$  and  $f_1 r_{s,2}(\gamma_s^x = \beta_s^y = 0, \alpha_s^z = 15^\circ)$ , while the *C*-space  $C_{3,3}(f_1^* r_{s,3})$  for  $f_1^*$  considers a null sensor orientation  $f_1^* r_{s,3}(\alpha_s^z = \beta_s^y =, \gamma_s^x = 0^\circ)$ . All constrained spaces were computed using the imaging parameters of sensor  $s_1$  (cf. Table A6) and geometric parameters of the features from Table A7.

Figure 14 visualizes the scene comprising the C-spaces for acquiring  $f_1$  and  $f_1^*$ . All  $C_{3,1-3}$  manifolds were computed by scaling first the manifold of the frustum space, considering the scaling factors addressed within the past subsections, and then by reflecting and transforming the manifold with the corresponding sensor orientation (for  $C_{3,1}$  see Table A5, for  $C_{3,2}$  see Equation (25), and for  $C_{3,3}$  see Equation (26)).

To verify the geometrical relationships introduced within this subsection, a virtual camera using the trimesh Library [57] and the imaging parameters of  $s_1$  were created. Then, the depth images and their corresponding point clouds at eight extreme viewpoints, i.e., the manifold vertices, were rendered to verify that the features could be acquired from each viewpoint. The images and point clouds of all extreme viewpoints confirm that the features lie at the border of the frustum space and can be entirely captured. Figure A4a–c demonstrate this empirically and show the depth images and point clouds at the selected extreme viewpoints ( $f_1 p_{s,1}$ ,  $f_1 p_{s,2}$ ,  $f_1^* p_{s,3}$ ) from Figure 14.

Our approach provides an analytical and straightforward solution for efficiently characterizing the C-space which is limited by the frustum space and feature geometry. Since the delta factors can be applied directly to the vertices of the frustum space, the

computational cost is similarly efficient to the computation time of  $C_1$  with an average computation time of 5.8 ms and  $\sigma$  = 3.2 ms.



**Figure 14.** Characterization of diverse *C*-spaces in *SE*(3), considering the feature geometry to capture a 2D square feature  $f_1$  and a 3D pocket feature  $f_1^*$ . The exemplary scene displays two *C*-spaces for acquiring feature  $f_1$  with two different sensor orientations,  $C_{3_1}({}^{f_1}\boldsymbol{r}_{s,1})$  and  $C_{3_2}({}^{f_1}\boldsymbol{r}_{s,2})$ , one *C*-space  $C_{3_3}({}^{f_1}\boldsymbol{r}_{s,3})$  for capturing  $f_1^*$ , and the frames of one extreme viewpoint at each constrained space.

# 4.3.7. Summary

This subsection extended the formulation of the core C-space  $C_1$  (see Section 4) to a C-space  $C_3$ , taking into account the feature's geometry dimensions. Using an exhaustive trigonometric analysis, our study introduced the exact relationships to characterize the vertices of the required multi-dimensional manifold, considering a generalized model of the feature geometry, the sensor's orientation, and its FOV angles. Our findings show that the characterization of the C-space constrained by the feature geometry can be computed with high efficiency using an analytical approach.

The trigonometric relationships introduced in this section are sufficient to characterize the *C*-space manifold while taking into account the rotation of the sensor in one axis. Characterizing the explicit relationships for a simultaneous rotation of the sensor in all axes is beyond the scope of this paper. However, the trigonometric relationships and general approach presented in this subsection can be used as the basis for their derivation.

#### 4.4. Constrained Spaces Using Scaling Vectors

If a viewpoint constraint can be formulated using scaling vectors, as suggested for the feature geometry in Section 4.3, then the same approach can be equally applied to characterize the constrained space of different viewpoint constraints. This subsection introduces a generic formulation for integrating such viewpoint constraints and proposes characterizing kinematic errors and the sensor accuracy following this approach.

# 4.4.1. Formulation

If the influence of an *i* viewpoint constraint  $c_i \in \tilde{C}$  can be characterized by a scaling vector  $\Delta(c_i)$  to span its corresponding constrained space,  $C_i$ , then its vertices  $V^{C_i}$  can be scaled using a generalized formulation of Equation (22):

$$\boldsymbol{V}_{k}^{\mathcal{C}_{i}} = \boldsymbol{V}_{k}^{\mathcal{C}_{1}} - \boldsymbol{\Delta}(c_{i}).$$
(27)

**Integrating Multiple Constraints** The characterization of a jointed constrained space, which integrates several viewpoint constraints, can be computed using different approaches. On one hand, the constrained spaces can be first computed and intersected iteratively using CSG operations, as originally proposed in Equation (12). However, if the space spanned by such viewpoint constraints can be formulated

according to Equation (27), the characterization of the constrained space C can be more efficiently calculated by simply adding all scaling vectors:

$$\boldsymbol{V}_{k}^{\mathcal{C}}(\widetilde{C}) = \boldsymbol{V}_{k}^{\mathcal{C}_{1}} - \sum_{c_{i} \in \widetilde{C}, i \neq 1} \boldsymbol{\Delta}(c_{i}).$$
<sup>(28)</sup>

While the computational cost of CSG operations is at least proportional to the number of vertices between two surface models, note that the complexity of the sum of Equation (28) is just proportional to the number of viewpoint constraints.

- **Compatible Constraints** Within this subsection, we propose further possible viewpoint constraints that can be characterized according to the scaling formulation introduced by Equation (28).
  - **Kinematic errors:** Considering the fourth viewpoint constraint and the assumptions addressed in Section 2.8, the maximal kinematic error  $\epsilon$  is given by the sum of the alignment error  $\epsilon_e$ , the modeling error of the sensor imaging parameters  $\epsilon_s$ , and the absolute position accuracy of the robot  $\epsilon_r$ :

$$|\epsilon| = |\epsilon_e| + |\epsilon_s| + |\epsilon_r|.$$

Assuming that the total kinematic error has the same magnitude in all directions, all vertices can be equally scaled. The vertices of the *C*-space  $C_4(\epsilon)$  are computed using the scaling vector  $\Delta(\epsilon)$ :

$$V_k^{\mathcal{C}_4}(\mathbf{\Delta}(\epsilon)) = V_k^{\mathcal{C}_1} - \mathbf{\Delta}(\epsilon)).$$

Sensor Accuracy: If the accuracy of the sensor *a<sub>s</sub>* can be quantified within the sensor frustum, then similarly to the kinematic error, the manifold of the *C*-space C<sub>5</sub>(*a<sub>s</sub>*) can be characterized using a scaling vector Δ(*a<sub>s</sub>*):

$$\boldsymbol{V}_{k}^{\mathcal{C}_{5}}(\boldsymbol{\Delta}(a_{s})) = \boldsymbol{V}_{k}^{\mathcal{C}_{1}} - \boldsymbol{\Delta}(a_{s})).$$

Figure A5 visualizes an exemplary and more complex scenario comprising an individual scaling of each vertex. This example shows the high flexibility and adaptability of this approach for synthesizing C-spaces for different viewpoint constraints according to the particular necessities of the application in consideration.

#### 4.4.2. Summary

The use of scaling vectors is an efficient and flexible approach to characterizing the C-space spanned by any viewpoint constraint. Within this section, we considered a few viewpoint constraints that can be modeled aligned to this formulation. However, it should be noted that this approach requires an explicit characterization of the individual scaling vectors and the overall characterization may be limited by the number of vertices of the base C-space  $C_1(\mathcal{I}_s)$ . For instance, our model of the  $\mathcal{I}$ -space considers just eight vertices. Thus, any viewpoint constraint that requires a more complex geometrical representation could be limited by this.

Moreover, note that if the regarded viewpoint constraint can be explicitly characterized as a manifold in SE(3), this C-space can then be directly intersected with the rest of the constraints as originally suggested by Equation (12); such an example is given for characterizing the robot workspace (see Section 4.8.1). However, note that such an approach is generally computationally more expensive; hence, this study recommends that the characterization of viewpoint constraints using scaling vectors be prioritized whenever possible.

# 4.5. Occlusion Space

We consider the occlusion-free view of a feature (related terms: shadow or bistatic effect) a non-negligible requirement that must be individually assessed for each viewpoint. In the context of our framework, this subsection outlines the formulation of a negative topological space —the occlusion space  $C_6^{occl}$ — to ensure the free visibility of each viewpoint within the C-space. Although other authors [15,18,26] have already suggested the characterization of such spaces, the present study proposes a new formulation of such a space aligned to our framework. Our approach strives for an efficient characterization of  $C_6^{occl}$  using simplifications about the feature's geometry and the occlusion bodies.

#### 4.5.1. Formulation

If a feature *f* is not visible from at least one sensor pose within the *C*-space, it can be assumed that at least one rigid body of the RVS is impeding its visibility. Thus, an occlusion space for *f* denoted as  $C_6^{occl} \subset SE(3)$  exists and a valid sensor pose cannot be an element of it  $p_s \notin C_6^{occl}$ . However, it is well known that the characterization of such occlusion spaces can generally be computationally expensive. Therefore, it seems inefficient to formulate  $C_6^{occl}$  in the special Euclidean group for all possible sensor orientations. For this reason, and by exploiting the available *C*-space spanned by other viewpoint constraints,  $C_6^{occl}$  can be formulated based on the previously generated *C*-space, a given sensor orientation  $r_s^{fix}$ , and the surface models of the rigid bodies  $\kappa \in K$ :

$$\mathcal{C}_6^{occl} := \mathcal{C}_6^{occl}(f, \mathcal{C}, \mathbf{r}_s^{fix}, K).$$
<sup>(29)</sup>

Contrary to all other viewpoint constraint formulations (see Equation (11)), it must be assumed that the occlusion space is not a subset of the *C*-space  $C_6^{occl} \notin C$ . Hence, let Equation (12) be reformulated as the set difference of the *C*-spaces of other viewpoint constraints and the occlusion space:

$$C = \left(\bigcap_{i=1, i \neq 6}^{j} C_{i}\right) \setminus C_{6}^{occl}.$$
(30)

If the resulting constrained space results in being a non-empty set,  $C \neq \emptyset$ , there exists at least one valid sensor pose for the selected sensor orientation with occlusion-free visibility to the feature *f*.

# 4.5.2. Characterization

The present work proposes a strategy to compute  $C_6^{occl}$ , which is broken down into the following general steps. In the first step, the smallest possible number of view rays is computed for detecting potential occlusions. In the second step, by means of ray-casting techniques, view rays are tested for occlusion against all rigid bodies of the RVS. For the interested reader, we refer to [58,59] for a comprehensive overview of ray-casting techniques. Then, the occlusion space is characterized using a simple surface reconstruction method using the colliding points of the rigid bodies and some further auxiliary points. In the last step, the occlusion space is integrated with the *C*-space spanned by other viewpoint constraints, as given by Equation (30).

Algorithm A2 describes more comprehensively all the steps to characterize the occlusion space  $C_6^{occl}$ , and Figure 15 provides an overview of the workflow and visualization of the expected results of the most significant steps, considering an exemplary occluding object  $\kappa_1$ . A more comprehensive description of the characterization of the view rays while using the *C*-space characterized by other viewpoint constraints can be found in the Table A6.



(a) Step 1: A set of view rays  $(\varsigma_{g_{f,c}})$  are computed for each *c* surface point using the limits of the *C*-space  $C_3$ . Step 2: The occluding points  $(q_f^{occl,\kappa})$  of a rigid body are found using ray casting.



(**b**) Step 3: The occlusion rays  $(g_{g_{f,c}}^{occl,\kappa})$  are computed between each *c* surface point and all occlusion points  $(q_f^{occl,\kappa})$ .

Step 4: An additional point  $(*q_f^{occl,k})$  is selected at each occlusion ray beyond the constrained space.



(c) Step 5: The occlusion space is characterized using the convex hull spanned by all colliding points and the additional points lying at the occlusion rays.

**Figure 15.** Overview of the computation steps of Algorithm A2 for the characterization of the occlusion space  $C_6^{occl,\kappa}$  induced by an occluding rigid body  $\kappa$ .

# 4.5.3. Verification

For verification purposes, we consider an academic example, which comprises an icosahedron occluding the sight of feature  $f_1$ . The dimensions and location of the occluding object  $\kappa_1$  are described in Table A7. Figure A6 displays the related scene and the manifolds of the computed occlusion space and corresponding occlusion-free space. In the first step, the *C*-space for feature  $f_1$ , considering its geometry, i.e.,  $C_3$ , and the following sensor orientation  $f_1 \mathbf{r}_s (\alpha_s^z = \gamma_s^x = 0, \beta_s^y = 15^\circ)$  was characterized. In the second step, the manifold of the occlusion space,  $C_6^{occl}$ , was synthesized following the steps described in Algorithm A2. A discretization step of  $d^{\varsigma} = 0.5^\circ$  was selected for computing the view rays  $\varsigma_{g_{f,c}}$ .

Figure 16 shows the rendered point cloud and range image at one extreme viewpoint within the resulting occlusion-free space. The rendered point cloud and image confirm that although the occluding body lies within the frustum space of the viewpoint, the feature and its entire geometry can still be completely captured. As expected, the computation of the collision points using ray casting was the most computationally expensive step with a total time of  $t(q_{f,c}^{occl,\kappa}) = 0.7$  s, and the total time required for characterizing the occlusion-free space corresponded to  $t(C_{6}^{occl,\kappa}) = 1.32$  s.



**Figure 16.** Visualization of the occlusion C-space in SE(3) (red manifold) and the occlusion-free space (blue manifold) to acquire a square feature  $f_1$  considering an occlusion body  $\kappa$  (icosahedron in orange). Verification of occlusion-free visibility: rendered scene (**left image**), depth image (**right image in the upper corner**), and detailed view of the rendered point cloud and object (**right image in the lower corner**) at an extreme viewpoint  $p_{s,1} \notin C_6^{occl,\kappa}$ .

#### 4.5.4. Summary

Within this subsection, a strategy that combines ray-casting and CSG Boolean techniques to compute an occlusion space  $C_6^{occl}$  was introduced. The present study showed that the  $C_6^{occl}$  manifold can be thoroughly integrated with the *C*-space spanned by other viewpoint constraints, complying with the framework proposed within this publication. Moreover, to enhance the efficiency of the proposed strategy, we considered a simplification of the feature geometry, discretization of the viewpoint space, and the use of a previous computed *C*-space.

Due to the aforementioned simplifications, it should be kept in mind that, contrary to the other *C*-spaces, the  $C_6^{occl}$  manifold does not represent the explicit and real occlusion space and should be treated as an approximation of it. For example, a significant source of error regarding the accurate identification of all occluding points is the chosen discretizing step size for the computations of the view rays. This effect is known within ray-casting applications and can also be observed in Figure 15, where the right corner point of the colliding body is missed. Within the context of the present study, comprising robot systems, we assume that such simplifications can be safely taken into consideration if the absolute position accuracy of the robot is considered for the selection of the step size  $d^{\varsigma}$ . For example, assuming a conservative absolute accuracy of a robot of 1 mm and a minimum working distance of 200 mm, it is reasonable to choose a step size of  $d^{\varsigma} \approx 0.3^{\circ}$ , using the arc length formula (1 mm/200 mm) = 0.005 rad. Alternatively, more robust solutions can be achieved by scaling the occlusion space with a safety factor.

Moreover, special attention must be given when computing the occlusion space as suggested in Step 5 of Algorithm A2 for rigid bodies with hollow cavities, e.g., a torus. It should be expected that the result of the occlusion space will be more conservative. A more precise characterization of the occlusion space falls outside the scope of this paper and could be achieved using more sophisticated surface reconstruction algorithms [60,61].

Finally, it is important to mention that the characterization of the occlusion space may lead to a non-watertight manifold, which may complicate the further processing of the jointed C-space. Thus, we recommend computing the occlusion space as the last viewpoint constraint.

#### 4.6. Multisensor

Considering the intrinsic nature of a range sensor, in its minimal configuration, two imaging devices (two cameras or one camera and one active projector) are necessary to acquire a range image. Therefore, a range sensor can be regarded as a multisensor system. Up to this point, it had been assumed that the C-space of a range sensor could be characterized using just one frustum space  $\mathcal{I}_s$  and that the resulting C-space integrates the imaging parameters and any other viewpoint constraints of all imaging devices of a range sensor. On one hand, some of the previous introduced formulations and most of the related work demonstrated that this simplification is in many cases sufficient for computing valid viewpoints. On the other hand, this assumption could also be regarded as restrictive and invalid for some viewpoint constraints. For example, the characterization of the occlusion-free space as described in Section 4.5 will not guarantee free sight to both imaging devices of a range sensor.

For this reason, our study assumes that each imaging device can have individual and independent viewpoint constraints. As a result, we consider that an individual C-space can be spanned for each imaging device. Furthermore, this section outlines a generic strategy to merge the individual C-spaces of multiple imaging devices to span a common C-space that satisfies all viewpoint constraints simultaneously.

To the best of our knowledge, none of the related work has considered viewpoint constraints for the individual imaging devices of a range sensor or for multisensor systems. In Section 5, the scalability and generality of our approach while considering a multisensor system is demonstrated with two different range sensors.

# 4.6.1. Formulation

Our formulation is based on the idea that each imaging device can be modeled independently and that all devices must simultaneously fulfill all viewpoint constraints. First, considering the most straightforward configuration of a sensor with a set of two imaging devices  $\{s_1, s_2\} \in \tilde{S}$ , we can assume that we have two different sensors with two different frustum spaces  $\mathcal{I}_{s_1} = \mathcal{I}_0(s_1 p_s, I_{s_1})$  and  $\mathcal{I}_{s_2} = \mathcal{I}_0(s_2 p_s, I_{s_2})$  (cf. Section 2.5). Thus, aligned to the formulation from Section 4.1, the core C-space for each sensor can be expressed as follows:

$$\mathcal{C}_1^{s_1}(s_1 \boldsymbol{r}_s^{fix}, \mathcal{I}_{s_1}, B_f) \text{ and } \mathcal{C}_1^{s_2}(s_2 \boldsymbol{r}_s^{fix}, \mathcal{I}_{s_2}, B_f).$$

In a more generic definition that considers all viewpoint constraints of  $s_1$  or  $s_2$ , the C-space is denoted as  $C^{s_1} := C_1^{s_1}(f, \tilde{C}^{s_1})$  or  $C^{s_2} := C_1^{s_2}(f, \tilde{C}^{s_2})$ .

Considering that the rigid transformation  ${}^{s_1}_{s_2}T_s$  between the two imaging devices of a sensor is known, it can be assumed that a valid sensor pose for the first imaging device exists, if and only if the second imaging device also lies within in its corresponding C-space simultaneously. The formulation of such condition follows

$$s_1 \boldsymbol{p}_s \in \mathcal{C}^{s_1} \iff s_2 \boldsymbol{p}_s \begin{pmatrix} s_2 \\ s_1 \end{pmatrix} T_s, \boldsymbol{p}_{s_1} \in \mathcal{C}^{s_2},$$
(31)

supposing that the reference positioning frame for the sensor is the first imaging device  $s_1$ .

If the condition from Equation (31) is valid, there must exist a C-space for  $s_1$  that integrates the viewpoint constraints of both lenses being denoted as  $C_7^{s_{1,2}}$ , which formulation follows:

$$\mathcal{C}_{7}^{s_{1,2}} = \mathcal{C}_{7}^{s_{1}}(\mathcal{C}^{s_{2}}) = \{_{s_{1}}\boldsymbol{p}_{s} \in \mathcal{C}_{7}^{s_{1,2}} \\ |_{s_{1}}\boldsymbol{r}_{s}^{fix} \in \mathcal{R}_{s',s_{2}}^{f}\boldsymbol{p}_{s}(_{s_{1}}^{s_{2}}\boldsymbol{T}_{s',s_{2}}\boldsymbol{p}_{s}) \in \mathcal{C}^{s_{2}}\}.$$

The joint space can then be integrated with the C-space for  $s_1$  employing a Boolean intersection:

$$\mathcal{C}^{s_{1,2}} = \mathcal{C}^{s_1} \bigcap \mathcal{C}_7^{s_{1,2}}$$

A more generic formulation of the *C*-space of  $s_1$  being constrained by all imaging devices  $s_t \in \widetilde{S}$  is given:

$$\mathcal{C}^{S_1} = \mathcal{C}^{s_1} \bigcap_{s_t \in \widetilde{S}} \mathcal{C}_7^{s_1, s_t}$$

Analogously,  $C^{\tilde{S}_2}$  denotes the space for positioning the imaging device  $s_2$  including the constraints of  $s_1$ .

# 4.6.2. Characterization

The C-space  $C^{\tilde{S}_1}$ , which comprises all constraints of all imaging devices  $s_t \in \tilde{S}$ , can be straightforwardly characterized following the five simple steps given in Algorithm A3. Figure 17 visualizes the interim manifolds at each step to ultimately characterize the manifold  $C^{\tilde{S}_1}$ . Finally, Figure 17f shows an exemplary extreme viewpoint, where both imaging devices frames are within their respective C-spaces  ${}_{s_1}p_s \in C^{\tilde{S}_1}$  and  ${}_{s_2}p_s \in C^{\tilde{S}_2}$  and the feature geometry lies within both frustum spaces.

The space  $C^{\tilde{S}_2}$  for the second imaging device can be computed analogously following the same steps. However, the topology of the resulting  $C^{\tilde{S}_2}$  will be identical to  $C^{\tilde{S}_1}$ . Hence, instead of repeating the steps described in Algorithm A3 for a second imaging device, a more efficient alternative is to translate the manifold of  $C^{\tilde{S}_1}$  to the position of  $s_2$  at  $s_2 p_s(s_2 t_s = B_f)$  using the rigid translation

$$\mathcal{C}^{\widetilde{S}_2} = translation(\mathcal{C}^{\widetilde{S}_1}, {}^{s_2}_{s_1}T_s)$$

Note that if two imaging devices are the same orientation, the resulting C-spaces are identical, hence  $C^{\tilde{S}_1} = C^{\tilde{S}_2}$ .



(a) Step 1: Compute *C*-space  $C^{s_1}(_{s_1}r_s^{fix}, \widetilde{C}^{s_1})$  for imaging device  $s_1$ .



 $\mathcal{C}\text{-space}\mathcal{C}^{s_2}(_{s_2}r_s^{fix}(_{s_1}r_s^{fix}), \tilde{C}^{s_2})$ for imaging device  $s_2$ .



(c) Step 3: Compute the pose,  $p_{s_1}^{f,s_2}$ , that the first sensor takes when computing  $C^{s_2}$ .

Figure 17. Cont.



(d) Step 4: Duplicate the manifold  $C^{s_2}$  and translate it to  ${}_{s_1}p_s^{f,s_2}$ .  $C_7^{s_{1,2}}$  denotes the resulting manifold at this position.



(e) Step 5: The intersection between  $C_7^{s_{1,2}}$  and  $C_7^{s_2}$  yields the space  $C_{\tilde{S}_1}^{\tilde{S}_1}$  for sensor  $s_1$ , which integrates all viewpoint constraints of  $s_2$ .



(f) Repeat Steps 1–5 for the second imaging device or duplicate and translate manifold  $C^{\tilde{S}_1}$  for computing  $C^{\tilde{S}_2}$ .

**Figure 17.** Overview of the computation steps of Algorithm A3 to characterize the C-space  $C^{\tilde{S}_1}$  that integrates viewpoint constraints of the two imaging devices ( $s_1$ ,  $s_2$ ) from a range sensor.

# 4.6.3. Verification

The joint constrained spaces  $C^{\tilde{S}_1}$  and  $C^{\tilde{S}_2}$  for  $s_1$  and  $s_2$  were computed according to the steps provided in Algorithm 1 for acquiring feature  $f_1$ . The imaging parameters of both sensors and rigid transformation between them are given in Table A6.

First, the C-spaces of both sensors were spanned considering their imaging parameters and a null orientation of the first sensor, i.e.,  ${}^{f}\mathbf{r}_{s,1}(\alpha_{s}^{z} = \beta_{s}^{y} = \gamma_{s}^{x} = 0^{\circ})$ . The individual constrained spaces were computed for each sensor while considering the constrained space affected by the feature geometry, i.e.,  $C_{3}^{s_{1}}$  and  $C_{3}^{s_{2}}$ . Since the frustum space of the second sensor always lies within the first one, we additionally considered a fictitious accuracy constraint for the depth of the second sensor,  $C_{5}^{s_{2}}(a_{s_{2}}(z_{min} = 500 \text{ mm}, z_{max} = 700 \text{ mm}))$ , to limit its working distance.

Figure 18 visualizes the described scene and resulting manifolds of the constrained spaces. Figure 19 displays the frustum spaces, rendered depth images of both sensors, and the resulting point cloud at an extreme viewpoint. The rendered images provide a visual verification of our approach, demonstrating that  $f_1$  is visible from both sensors. Note that the second device represents an active structured light projector (see in Table A6).

The total computation time of the constrained space was estimated at 200 ms. The computation time depended mainly on the intersection from Step 5, which corresponded to 100 ms. The computation results only apply to this case. Such analyses are difficult to generalize since the complexity of Boolean operations depends decisively on the number of vertices of the manifolds of the occluding objects.



**Figure 18.** Characterization of the *C*-space for the first sensor in *SE*(3),  $C^{S_1}$  (blue manifold), being delimited by the *C*-space of the second sensor,  $C^{s_2}$  (orange manifold without fill), to acquire a square feature  $f_1$ . The  $C^{\tilde{S}_2}$  (orange manifold) analogously characterizes the *C*-space for sensor  $s_2$ , considering the constraints of  $s_1$ .



**Figure 19.** Verification of occlusion-free visibility at an extreme viewpoint  $_{s_1} p_s \in C^{S_1}$  and  $_{s_2} p_s \in C^{S_2}$ : rendered scene (left image), depth images of  $s_1$  (right image in the upper corner) and  $s_2$  (right image in the lower corner).

# 4.6.4. Summary

This subsection introduced the formulation and characterization of a C-space, which considers the intrinsic configuration of range sensors comprising at least two imaging devices. Our approach enables the combination of individual viewpoint constraints for each device and the characterization of a C-space, which fulfills simultaneously all viewpoint constraints from all imaging devices.

The strategy proposed in Algorithm A3 was demonstrated to be valid and efficient in characterizing the manifolds of such a *C*-space. Nevertheless, we do not dispose alternative approaches for its characterization. For instance, if the frustum spaces are intersected initially, the resulting frustum space can be used as the base for spanning the rest of the constraints. However, we consider the steps proposed within this subsection more traceable, modular, and extendable to consider further constraints, multisensor systems, or even transferable to similar problems. For example, a variation of the algorithm could be applied to maximize or guarantee the registration space between two different viewpoints, which represents a fundamental challenge within many vision applications [62].

#### 4.7. Robot Workspace

This section outlines the formulation of the robot workspace as a further viewpoint constraint to be seamlessly and consistently integrated with the other C-spaces.

#### 4.7.1. Formulation

A viewpoint can be considered valid if a sensor pose is reachable by the robot; hence, it lies within the robot workspace  $p_s \in W_r$ . The formulation of a constraint can then be straightforwardly formulated as follows:

$$\mathcal{C}_8 = \mathcal{W}_r = \{ \boldsymbol{p}_s \in \mathcal{C}_8 \mid \boldsymbol{p}_s \in \mathcal{W}_r \}.$$
(32)

#### 4.7.2. Characterization and Verification

In our work, we assume that the robot workspace is known and can be characterized by a manifold in the special Euclidean group SE(3). Considering this assumption, the constrained space  $C_8$  can be seamlessly intersected with the rest of the viewpoint constraints. Figure 20 shows an exemplary scene for acquiring feature  $f_1$  and the resulting constrained manifold  $C_3 \cap C_8$ , which considers a robot with a workspace of a half-sphere and a working distance of 1000 mm–1800 mm and the *C*-space manifold  $C_3$  spanned by the feature geometry.



**Figure 20.** Characterization of the robot workspace as a further *C*-space  $C_8$  and integration with other *C*-spaces, e.g., here  $C_3$ , using a CSG Intersection Operation.

#### 4.7.3. Summary

A more comprehensive formulation and characterization of the robot workspace to consider singularities requires more detailed modeling of the robot kinematics. Furthermore, our study has not considered the explicit characterization of the collision-free space of a robot, which had been the focus of exhaustive research in the last three decades. We assume that an explicit proof for collision must be performed in the last step for a selected sensor pose within the C-space. Nevertheless, we consider that our approach contributes substantially to a significant problem simplification by delimiting the search space to compute collision-free robot joint configurations more efficiently.

# 4.8. Multi-Feature Spaces

Up to this point, our work has outlined the formulation and characterization of C-spaces to acquire just one feature. Within this subsection we briefly outline the characterization of a C-space,  ${}^{F}C$ , which allows for the capture of a set of features F and the simultaneous fulfillment of all viewpoint constraints from all features  $f_m \in F$  with m = 1, ..., n.

# 4.8.1. Characterization

The characterization of  ${}^{F}C$  can be seamlessly achieved according to the two steps described in Algorithm A4. In the first step, the *C*-space for all *n* features,  ${}^{f_m}C$ , are characterized considering a fixed sensor orientation  $r_s^{fix}$  and the individual constraints  $\tilde{C}(f_m)$ . Then, the constrained space  ${}^{F}C$  is synthesized by intersecting all individual constrained spaces. Figure 21 shows the characterization of such a space for the acquisition of two features.



**Figure 21.** The characterization of the C-space spanned by two features is computed by intersecting its constrained spaces using the same sensor orientation.

# 4.8.2. Verification

To verify the proposed characterization of a constrained space for acquiring multiple features, we outlined an exemplary use case comprising two features  $\{f_1, f_2\} \in F$  (see Table A7). We computed the space,  ${}^FC$ , according to the steps of Algorithm A4, considering an orientation of  ${}^{f_1}r_s(\alpha_s^z = \beta_s^y = 0^\circ, \gamma_s^x = -10^\circ,)$  relative to the feature  $f_1$ . Figure A7 shows the described scene and visualizes the resulting joint space. The rendered scene and range images of Figure 22 confirm the validity of the  ${}^FC$ , demonstrating that both features can be simultaneously acquired at two extreme viewpoints within this space.



**Figure 22.** Verification of the <sup>*F*</sup>*C* at two extreme viewpoints  $\{p_{s,1}, p_{s,2}\} \in ^{F}C$ : rendered scene (**left image**) and depth images of  $p_{s,1}$ (**right image in the upper corner**) and  $p_{s,2}$  (**right image in the lower corner**).

#### 4.8.3. Summary

Within this subsection, we demonstrated that C-spaces from different features can be seamlessly combined to span a topological space that guarantees the acquisition of these features, simultaneously satisfying the individual feature viewpoint constraints.

The current study assumes that the sensor orientation can be arbitrarily chosen and that the features can be acquired jointly by the sensor. In most applications, such assumptions cannot always be met and the following fundamental questions arise: which features can be acquired simultaneously and which is an adequate sensor orientation? These questions fall outside the scope of this paper and yield the motivation of our ongoing research, which addresses the efficient combination of C-spaces to tackle the superordinated VPP.

# 4.9. Constraints Integration Strategy

The integration of viewpoint constraints can be considered to be commutative, i.e., the order of computation and integration of the constraints do not affect the characterization

of the final constrained space. However, due to the diverse computation techniques that our our framework considers, a well-thought-out strategy may contribute towards increasing the computational efficiency of the overall process. In this publication, we outline one possible and simple strategy described in Algorithm A5 to integrate all viewpoint constraints into a single C-space.

The optimal integration of constraints falls outside the scope of this publication. Moreover, we consider that an optimal and efficient strategy can be tailored just by considering the individual application and its specific constraints.

#### 5. Results

Within this section, a comprehensive evaluation of the constraint formulations and their integration is undertaken. First, Section 5.2 verifies the formulations of all regarded viewpoint constraints of our work based on an academic example. In Section 5.3, the framework's broad generality and applicability is evaluated within an industrial RVS comprising two different sensors.

# 5.1. Technical Setup

This section provides an overview of the hardware and software used for the characterization of the C-spaces within the following sections. We briefly introduce the considered domains, parameters, and specifications that were employed to verify the individual formulations of C-spaces presented in Section 4.

# 5.1.1. Domain Models

- Sensors: We used two different range sensors for the individual verification of the *C*-spaces and the simulation-based and experimental analyses. The imaging parameters (cf. Section 2.5) and kinematic relations of both sensors are given in Table A6. The parameters of the lighting source of the *ZEISS Comet PRO AE* sensor are conservatively estimated values, which guarantee that the frustum of the sensor lies completely within the field of view of the fringe projector. A more comprehensive description of the hardware is provided in Section 5.
- Object, features, and occlusion bodies: For verification purposes, we designed an academic object comprising three features and two occlusion objects with the characteristics given in Table A7 in the Appendix A.
- Robot: We used a *Fanuc M-20ia* six-axis industrial robot and respective kinematic model to compute the final viewpoints to position the sensor.

# 5.1.2. Software

The backbone of our framework was developed based on the *Robot Operating System* (ROS) (Distribution: Noetic Ninjemys) [63]. The framework was built upon a knowledge-based, service-oriented architecture. A more detailed overview of the general conceptualization of the architecture and knowledge-base is provided in our previous works [64,65].

Most of our algorithms consist of generating and manipulating 3D manifolds. Hence, based on empirical studies, we evaluated different open-source Python 3 libraries and used them according to their best performance for diverse computation tasks. For example, the PyMesh Library from [66] demonstrated the best computational performance for Boolean operations. On the contrary, we used the trimesh Library [57] for verification purposes and for performing ray-casting operations, due to its integration of the performance-oriented Embree Library [67]. Additionally, for further verification, visualization, and user interaction purposes, we coupled the ROS kinematic simulation to Unity [68] using the ROS# library [69].

All operations were performed on a portable workstation Lenovo W530 running Ubuntu 20.04 with the following specifications: Processor Intel Core i7-4810MQ @2.80 GHz, GPU Nvidia 3000 KM, and 32 GB Ram.

# 5.2. Academic Simulation-Based Analysis

This subsection presents a simple but thorough academic use case that considers all introduced viewpoint constraints to perform an exhaustive evaluation of the presented formulations. As stated in the introduction, with this present scenario we aim to provide a first draft of a much-needed benchmark for other researchers that can be used as basis for further development, reproducibility, and comparison. The surface models, the resulting manifolds of the computed *C*-spaces, and the frustum spaces can be found attached in the additional material of our publication.

## 5.2.1. Use Case Description

The exemplary case regards an RVS with two sensors and an object of interest containing three different features with different sizes and geometries. Table A8 gives a detailed overview of the considered constraints. In addition to the imaging parameters of both range sensors, all other parameters can be assumed to be fictitious though realistic. The kinematic and imaging models corresponds to the real RVS, which is described in Section 5.3.1.

# 5.2.2. Results

Following the strategy described in Algorithm A5, the joint *C*-spaces of the four imaging devices, i.e.,  ${}^{F}C^{\tilde{S}_{1}}$ ,  ${}^{F}C^{\tilde{S}_{2}}$ ,  ${}^{F}C^{\tilde{S}_{3}}$ , and  ${}^{F_{r}}C^{\tilde{S}_{4}}$ , were computed for acquiring all features and considering all viewpoint constraints from Table A8. Figure 23 shows the complexity of the described case comprising three features and some of the resulting *C*-spaces. The blue manifold  ${}^{F}C^{\tilde{S}_{1}}$  represents the final constrained space of the first imaging device  $s_{1}$ . It can be appreciated that  ${}^{F}C^{\tilde{S}_{1}}$  is characterized by the intersection of all other *C*-spaces. Moreover, Figure 23 shows that the  ${}^{F}C^{\tilde{S}_{1}}$  manifold is mainly constrained by the *C*-space corresponding to the second range sensor, i.e.,  ${}^{F}C^{\tilde{s}_{1},s_{3},s_{4}}$ .

To verify the validity of the computed C-spaces, the depth images and point clouds for all imaging devices at eight extreme viewpoints were rendered. Figure 24 shows the corresponding rendered scene and resulting depth images for each imaging device at one extreme viewpoint  $_{s_1} p_s \in {}^F C^{\tilde{S}_1}$ . The depth images demonstrate that all imaging devices can successfully acquire all features without occlusion simultaneously.

The total computation time for characterizing all C-spaces corresponded to  $t({}^{F}C^{S_1}, {}^{F}C^{S_2}, {}^{F}C^{\tilde{S}_3}, {}^{F}C^{\tilde{S}_4}) \approx 50$  s. However, this time comprises other computation steps (e.g., frames transformation and inverse kinematic operations using ROS-Services) which distort the effective computation time of the C-spaces. A proper analysis of the computational efficiency of the whole strategy remains to be further investigated.



**Figure 23.** Characterization of the C-space spanned by a set of viewpoint constraints (see Table A8) for a multisensor scenario to capture a set of features *F*.



**Figure 24.** (Left): Verification scene visualizing the frames and  $\mathcal{I}$ -spaces of all imaging devices at the extreme sensor pose  $_{s_1}p_s \in {}^F \mathcal{C}^{\widetilde{S}_1}$ ,  $_{s_2}p_s \in {}^F \mathcal{C}^{\widetilde{S}_2}$ ,  $_{s_3}p_s \in {}^F \mathcal{C}^{\widetilde{S}_3}$ , and  $_{s_4}p_s \in {}^F \mathcal{C}^{\widetilde{S}_4}$ ) that fulfills all viewpoint constraints. (**Right**): Depth images of all imaging devices at the corresponding sensor pose.

# 5.2.3. Summary

Despite the complexity of the use case, the framework (models, methods, and integration strategy) presented within this paper demonstrated its effectiveness in characterizing a continuous topological space in the special Euclidean, where all defined viewpoint constraints could be fulfilled. The simulated depth images and point clouds confirmed that all selected viewpoints within the characterized C-space satisfied all regarded constraints. Moreover, the proposed academic example effectively outlines a simple but sufficient complex scenario to benchmark our and future viewpoint planning strategies.

# 5.3. Real Experimental Analysis

To assess the usability and validity of our framework within real applications, the framework presented in this study was utilized to generate automatically valid viewpoints for capturing different features of a car door using real RVS, i.e., the *AIBox* from *ZEISS*. The AIBox is an industrial measurement cell used to automate different vision-based quality inspection tasks such as dimensional metrology and digitization, among others.

# 5.3.1. System Description

The AIBox is an integrated industrial RVS, equipped with a structured light sensor (*ZEISS COMET PRO AE*), a six-axis industrial robot (*Fanuc M-20ia*), and a rotary table for mounting an inspection object. Moreover, to evaluate the use of our approach considering a multisensor system, we additionally attached a stereo sensor (*rc\_visard 65, Roboception*) to the structured light sensor. The imaging parameters of both sensors are given in Table A6. Figure 25 provides an overview of the reconfigured AIBox with the stereo sensor. We assume that the inspection object is roughly aligned, e.g., in [70] we presented a CNN fully trained on synthetic data to automate this task using the RVS sensor.



Figure 25. Overview of the core components of the reconfigured inspection RVS AIBox.

# 5.3.2. Vision Task Definition

The validation of our framework was performed on the basis of two vision tasks while considering diverse viewpoint constraints. For the first task, we considered just the ZEISS sensor's ability to acquire the features  $\{f_1, f_2\} \in F_1$ , which lie on the outside of the door and can be potentially occluded by the door fixture. For the second task, we considered both sensors and the acquisition of two features  $\{f_3, f_4, f_5\} \in F_2$  on the inside of the door. The incidence angle for the first case corresponded to a sensor orientation of  ${}^o r_{s_2}(\alpha_s^z = \gamma_s^x = 0^\circ, \beta_s^y = -15^\circ)$  and for the second of  ${}^o r_{s_1}(\alpha_s^z = \beta_s^y = \gamma_s^x = 0^\circ)$ . To compensate for any kinematic modeling uncertainties, we consider an overall kinematic error of  $\epsilon_{s_1}^{x,y,z} = (70.0, 70.0, 50.0)$  mm for  $s_1$  and of  $\epsilon_{s_3,4}^{x,y,z} = (30.0, 30.0, 30.0)$  mm for  $s_3$  and  $s_4$ .

# 5.3.3. Results

For both vision tasks, we computed the necessary C-spaces aligned to the strategy presented by Algorithm A5. The C-spaces of the first inspection scenario for the camera  $F_1 C^{\tilde{S}_1}$  and projector  $F_1 C^{\tilde{S}_2}$  of the *Comet Pro AE* and its corresponding occlusion spaces are displayed on the left image of Figure 26. To assess the validity of the characterized C-spaces, we chose diverse extreme viewpoints at the vertices of the  $F_1 C^{\tilde{S}_1}$  manifold and performed real measurements. On the right side of Figure 26, the real monochrome images of the camera and the resulting point clouds at two validating viewpoints are displayed. The 2D images and point clouds prove that both features can be successfully acquired from both of these viewpoints, which confirms the free sight for the sensor and the illumination of both features without shadows.



**Figure 26.** (a) Visualization of the characterized C-spaces  $({}^{F_1}C^{\tilde{S}_1}, {}^{F_1}C^{\tilde{S}_2})$  to capture features  $f_1$  and  $f_2$  by the *COMET ProAE*. Right figures: 2D images and corresponding point clouds at two extreme viewpoints: (b) for  ${}_{s_1}p_{s,1} \in {}^{F_1}C^{\tilde{S}_1}$  and (c) for  ${}_{s_1}p_{s,2} \in {}^{F_1}C^{\tilde{S}_1}$ .

Moreover, on the left of Figure 27, the constrained spaces of the first imaging device of each sensor, i.e.,  ${}^{F_2}C^{\tilde{S}_1}$  and  ${}^{F_2}C^{\tilde{S}_3}$ , are visualized for the second inspection scenario. Analogously to the first scenario, two extreme viewpoints at the vertices of the manifolds were selected to assess the validity of the computed *C*-spaces. As expected, the real 2D images of all imaging devices and the resulting point clouds of both sensors at two exemplary extreme viewpoints (shown on the right of Figure 26) demonstrate that all features can be successfully acquired by the four imaging devices of both sensors.



**Figure 27.** (a): Visualization of the characterized C-spaces ( ${}^{F_2}C^{\tilde{S}_1}, {}^{F_2}C^{\tilde{S}_3}$ ) to capture the feature set  $\{f_3, f_4, f_5\} \in F_2$  by the *COMET ProAE* and *rc\_visard 65*. Right figures: 2D images and corresponding point clouds for both sensors at two extreme viewpoints  $\{_{s_1}p_{s,1's_1}p_{s,2}\} \in {}^{F_2}C^{\tilde{S}_1}$  (upper figures (**b**,**d**): *Comet Pro AE*, lower figures (**c**,**e**): *rc\_visard 65*.

#### 5.3.4. Summary

Using an industrial RVS, and regarding real viewpoint constraints, we were able to validate the formulations, characterization, and application of C-spaces for inspection tasks in an industrial context. These experiments show the suitability of our framework for an industrial application on a real RVS with multiple range sensors.

Furthermore, our strategy for merging individual C-spaces to capture more than one feature proved to be effective for the vision tasks under consideration. However, a more complex task such as the inspection of all door features requires a more complex strategy, which considers the search of features that can be acquired together. This question recalls the overall VPP, which falls outside the scope of this publication and we intend to address in our future work.

# 6. Conclusions

# 6.1. Summary

The computation of valid viewpoints considering different system constraints, named VGP in this publication, is considered a complex and unsolved challenge that lacks a generic and holistic framework for its proper formulation and resolution. In this paper, we outline the VGP as a geometric problem that can be solved explicitly in the special Euclidean group SE(3) using suitable and explicit models of all related domains of an RVS and viewpoint constraints. Within this context, much of our effort was devoted to the comprehensive and systematic formulation of the VGP and the exhaustive characterization of domains and viewpoint constraints aligned to the formulation of geometric problems.

The core result of this study is the characterization of C-spaces, which can be understood as topological manifolds that span a space with infinite viewpoint solutions to acquire one feature or a group of features, while considering various viewpoint constraints and modeling uncertainties. Our approach focuses on providing rather infinite valid solutions instead of optimal ones. If the entire a priori knowledge of the RVS can be formalized and integrated into the C-space, then it we can assume that any viewpoint within it is a local optimum. Our work shows that a handful of viewpoint constraints can be efficiently and simply modeled geometrically and integrated in a common framework to span such constrained spaces. Finally, based on a comprehensive academic example and a real application, we demonstrate the usability of such a framework.

#### 6.2. Limitations and Chances

We are aware that the framework proposed in the present study may have some limitations that may prevent its straightforward application for other RVS or use cases. First, it must be acknowledged that our framework can be classified under the category of model-based approaches. Therefore, in a first step, a priori information to model the components of the considered RVS must be examined. We consider that an exhaustive and explicit modeling of the necessary domains is required to deliver solutions that offer a higher generalization for other applications and systems. For the benefit of generalization, complexity reduction, and computational efficiency, we addressed various simplifications which could affect the accuracy of some models and might yield more conservative, though more robust, solutions.

We firmly believe that the VGP can be efficiently solved geometrically. We demonstrated that many constraints can be explicitly and efficiently characterized by combining several techniques, including linear algebra, trigonometry, and geometrical analysis. In the scope of our experiments, we confirmed that the computation of the C-spaces manifolds based on these approaches ran efficiently in linear times. However, we also noted that algorithms comprising CSG Boolean operations are more computationally expensive, especially for calculations considering multiple Boolean operations on the same manifold. Although this limitation can be minimized by filtering and smoothing algorithms for decimating manifolds, this characteristic could still be considered insufficient for some users and applications. Although the shortcomings of CSG Boolean techniques regarding their computational efficiency have been mentioned in some prior works, we also believe that the present available computational performance and paralleling capabilities of CPUs and GPUs require a new reevaluation of their overall performance. Additionally, our work also suggests that combined with efficient imaging processing libraries, approaches requiring heavy use of CSG operations can be efficiently used within many applications. Nevertheless, a comprehensive computational efficiency analysis to find a break-even point between our approach and others remains to be further investigated.

Moreover, we also see room for improvement to increase the efficiency of some of the algorithms presented. For instance, the computation of the occlusion space and integration of constraints could also be improved using an alternative strategy and more efficient algorithms implemented in low-level programming languages. Additionally, we also see potential for improving the efficiency of some algorithms. For instance, the performance of many algorithms could enormously benefit of computational optimization techniques such as parallelization and GPU computation. For replication purposes of our work, we encourage the reader to make a thorough evaluation of the performance of the state-of-the-art libraries available at the present time, according to their application needs and system requirements.

#### 6.3. Outlook

We consider the use of C-spaces appropriate, but not limited to vision tasks that rely on features. For example, we showed how our concept could be extended to applications that generally would not consider features and demonstrate its application for an object detection problem with a certain level of spatial uncertainty. Our ongoing work concentrates on assessing further applications or systems that may benefit from our approach, e.g., feature-based robot calibration or adaption to laser sensors. Further studies should still be undertaken in this direction to verify the usability and explore the limitations of our framework within other applications and RVSs.

Recalling that we neglected any sensor parameters that may directly constrain the C-space, e.g., exposure time, gain, and others, we consider some other lines of research that integrate such a parameter space in the V-space. For instance, our ongoing study investigates the combination of a data-based approach to optimize the exposure times and the use of C-spaces for finding optimized sensor poses.

The most promising future research should be devoted to the overall problem of the VPP, which could be reformulated based on the present study and its findings. Further research that will exploit this and comprises a holistic strategy for its resolution is already in progress.

We believe that our work will serve as a solid base and guideline for further studies to adapt and extend our framework according to the individual requirements of their concrete applications and RVS.

**Supplementary Materials:** The following supporting information can be downloaded at: https: //www.mdpi.com/article/10.3390/robotics12040108/s1, Video S1: Viewpoint\_Generation\_using\_ Cspaces\_Magana\_et\_al.mp4, collection of figures and meshes (an overview of the folder structure is provided at the README file).

Author Contributions: Conceptualization, A.M., J.D., P.B. and G.R.; methodology, A.M.; software, A.M.; validation, A.M. and J.D.; formal analysis, A.M.; investigation, A.M.; resources, G.R.; data curation, A.M.; writing—original draft preparation, A.M. and J.D.; writing—review and editing, A.M., J.D., P.B. and G.R.; visualization, A.M.; supervision, G.R.; project administration, A.M. and P.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (funding code ESB036/001).

**Data Availability Statement:** A comprehensive set of supplemental material is attached to this paper. This set includes a supporting video, C-spaces meshes, object meshes and frames, features and viewpoints frames, and rendered data used for verification and validation.

**Acknowledgments:** The framework presented in this paper was developed and thoroughly evaluated within the scope of the "CyMePro" (cyber-physical measurement technology for 3D digitization in the networked production) project. We thank our research partners AUDI AG and Carl Zeiss Optotechnik GmbH for the fruitful discussions and their cooperation. Moreover, we are deeply grateful to our colleagues Daria Leiber for the fruitful discussions regarding the initial drafting and Benedikt Schmucker for the critical revision of this publication.

Conflicts of Interest: The authors declare no conflict of interest.

#### Abbreviations

The following abbreviations are used in this manuscript:

C-space	Feature-Based Constrained Space
CSG	Constructive Solid Geometry
$\mathcal{I} ext{-space}$	Frustum space
RVS	Robot Vision System
VGP	Viewpoint Generation Problem
VPP	Viewpoint Planning Problem

#### **Appendix A. Tables**

 Table A1. Description of general requirements.

General Requirement	Description
1. Generalization	The models and approaches used should be abstracted and generalized at the best possible level so that they can be used for different components of an RVS and can be applied to solve a broad range of vision tasks.
2. Computational Efficiency	The methods and techniques used should strive towards a low level of computational complexity. Whenever pos- sible, analytical, and linear models should be preferred over complex techniques, such as stochastic and heuristic algorithms. Nevertheless, when considering offline scenarios, the trade-off between computing a good enough solu- tion within an acceptable amount of time should be individually assessed.
3. Determinism	Due to traceability and safety issues within industrial applications deterministic approaches should be prioritized.
4. Modularity and Scalability	The approaches and models should consider in general a modular structure and promote their scalability.
5. Limited a priori Knowledge	The parameters required to implement the models and approaches should be easily accessible for the end-users. Neither in-depth optics nor robotic knowledge should be required.

Notation	Index Description					
$b^r x^n_d$	x = variable, parameter, vector, frame, or transformation d = RVS domain, i.e., (s)ensor, (r)obot, (f)eature, (o)bject, (e)nvironment, or $delement of a list or setn$ = related domain, additional notation, or depending variable $r$ = base frame of the coordinate system $B_r$ or space of feature $f$ $b$ = origin frame of the coordinate system $B_b$					
Notes	The indices $r$ and $b$ just apply for pose vectors, frames, and transformations.					
Example	<ul> <li>The index notation can be better understood consider following examples:</li> <li><i>d</i>: Let the geometry of a feature be described by a surface point g<sub>f</sub> ∈ ℝ<sup>3</sup>.</li> <li><i>d</i>: If the feature comprises more surface points, then let the point with the index 2 be denoted by g<sub>f,2</sub> ∈ ℝ<sup>3</sup>.</li> <li><i>r</i>: Assuming that the position of a surface point g<sub>f</sub> is described in the coordinate system of the feature, B<sub>f</sub>, then it follows: <sup>f</sup>g<sub>f</sub>. In case that the base coordinate frame has the same notation as the domain itself, i.e., r = d, then just the index for the domain is given: g<sub>f</sub> = <sup>f</sup>g<sub>f</sub>.</li> <li><i>b</i>: In case that the frame of the surface point is given in the coordinate reference system of the object B<sub>0</sub>, then following notation applies: <sup>o</sup>g<sub>f</sub> = <sup>g</sup>g<sub>f</sub>.</li> </ul>					

Table A2. Overview of index notations for variables.

Table A3. List of the most common symbols.

Symbol	Description
	General
С	Viewpoint constraint
$\widetilde{C}$	Set of viewpoint constraints
f	Feature
F	Set of features
st	<i>t</i> imaging device of sensor <i>s</i>
υ	Viewpoint
$p_s$	Sensor pose in $SE(3)$
	Spatial Dimensions
В	Frame
t	Translation vector in $\mathbb{R}^3$
r	Orientation matrix in $\mathbb{R}^{3x3}$
V	Manifold vertex in $\mathbb{R}^3$
	Topological Spaces
$egin{smallmatrix} \mathcal{C} \ \mathcal{C}_i \ \mathcal{I}_s \end{bmatrix}$	$C$ -space for a set of viewpoint constraints $\widetilde{C}$ i $C$ -space of the viewpoint constraint $c_i$ Frustum space

Viewpoint Constraint	Description
1. Frustum Space	The most restrictive and fundamental constraint is given by the imaging capabilities of the sensor. This constraint is fulfilled if at least the feature's origin lies within the frustum space (cf. Section 2.5).
2. Sensor Orientation	Due to specific sensor limitations, it is necessary to ensure that the maximal permitted incidence angle between the optical axis and the feature normal lies within an specified range; see Equation (5).
3. Feature Geometry	This constraint can be considered an extension of the first viewpoint constraint and is fulfilled if all surface points of a feature can be acquired by a single viewpoint, hence lying within the image space.
4. Kinematic Error	Within the context of real applications, model uncertainties affecting the nominal sensor pose compro- mise a viewpoint's validity. Hence, any factor, e.g., kinematic alignment, robot's pose accuracy, which affects the overall kinematic chain of the RVS must be considered (see Section 2.6).
5. Sensor Accuracy	Acknowledging that the sensor accuracy may vary within the sensor image space (see Section 2.5), we consider that a valid viewpoint must ensure that a feature must be acquired within a sufficient quality.
6. Feature Occlusion	A viewpoint can be considered valid if a free line of sight exists from the sensor to the feature. More specifically, it must be assured that no rigid bodies are blocking the view between the sensor and the feature.
7. Bistatic Sensor and Multisensor	Recalling the bistatic nature of range sensors, we consider that all viewpoint constraints must be valid for all lenses or active sources. Furthermore, we also extend this constraint for considering a multisensor RVS comprising more than one range sensor.
8. Robot Workspace	The workspace of the whole RVS is limited primarily by the robot's workspace. Thus, we assume that a valid viewpoint exists if the sensor pose lies within the robot workspace.
9. Multi-Feature	Considering a multi-feature scenario, where more than one feature can be acquired from the same sensor pose, we assume that all viewpoint constraints for each feature must be satisfied within the same viewpoint.

Table A4. Overview and description of the viewpoint constraints considered in our work.

**Table A5.** Scaling factors for the vertices of the constrained space  $V_k^{C_3}$ , considering a sensor rotation around the *x*-axis or *y*-axis relative to the feature frame.

	Rotation	around y-a	xis		Rotation ar	ound <i>x</i> -axi	S		
k Vertex		$f_{r_s}(\alpha)$	$q_s^z = \gamma_s^x = 0, \mu$	$\beta_s^y \neq 0$ )		$f_{r_s}(\alpha_s^z)$	$=\beta_s^y=0,\gamma$	$\gamma_s^x \neq 0$ )	
of $V_{k}^{\mathcal{C}_{3}}$	Δ	$x_k$	$\Delta_k^y$	$\Delta_k^z$	$\Delta_k^x$	Δ	$y_k$	$\Delta_k^z$	
	$\beta_s^y < 0$	$\beta_s^y > 0$				$\gamma_s^x < 0$	$\gamma_s^x > 0$		
1	$\lambda^x$	$ ho^x$	$\frac{l_f}{2}$		$\frac{l_f}{2}$	$\lambda^y$	$ ho^y$		
2	$ ho^x$	$\lambda^x$	$\frac{l_f}{2}$		$\frac{l_f}{2}$	$\lambda^y$	$ ho^y$		
3	$\sigma^x$	$ ho^x$	$\frac{l_f}{2} + \varsigma^{x,y}$		$\frac{l_f}{2} + \varsigma^{y,x}$	$ ho^y$	$\sigma^y$		
4	$ ho^x$	$\sigma^x$	$\frac{l_f}{2} + \varsigma^{x,y}$	$\rho^{z,y}$	$\frac{l_f}{2} + \varsigma^{y,x}$	$ ho^y$	$\sigma^y$	$\rho^{z,x}$	
5	$\lambda^x$	$ ho^x$	$\frac{l_f}{2}$		$\frac{l_f}{2}$	$ ho^y$	$\lambda^y$	ı	
6	$ ho^x$	$\lambda^x$	$\frac{l_f}{2}$		$\frac{l_f}{2}$	$ ho^y$	$\lambda^y$		
7	$\sigma^{x}$	$ ho^x$	$\frac{l_f}{2} + \varsigma^{x,y}$		$\frac{l_f}{2} + \varsigma^{y,x}$	$\sigma^y$	$ ho^y$		
8	$ ho^x$	$\sigma^x$	$\frac{l_f}{2} + \varsigma^{x,y}$		$\frac{l_f}{2} + \varsigma^{y,x}$	$\sigma^y$	$ ho^y$		

Range Sensor	1	2	
Manufacturer	Carl Zeiss Optotechnik GmbH,	Roboception, Munich, Germany	
Model	COMET Pro AE		rc_visard 65
3D Acquisition Method	Digital Fringe Projection		Stereo Vision
Imaging Device <i>s</i> <sub>t</sub>	Monochrome Camera: $s_1$ Blue Light LED-Fringe Projector: $s_2$		Two monochrome cameras: <i>s</i> <sub>3</sub> , <i>s</i> <sub>4</sub>
Field of view	$\theta_s^{x} = 51.5^{\circ},  \psi_s^{y} = 35.5^{\circ}$	$\theta_s^{x} = 70.8^{\circ}, \psi_s^{y} = 43.6^{\circ}$	$ heta_s^x\!=\!62.0^\circ$ , $\psi_s^y\!=\!48.0^\circ$
Working distances	$@400 \text{ mm} : (396 \times 266) \text{ mm}^2$	@200 mm : $(284 \times 160) \text{ mm}^2$	@200 mm : $(118 \times 178) \text{ mm}^2$
and near, middle, and far planes relative to	$(600 \text{ mm} : (588 \times 392) \text{ mm}^2)$	$(853 \times 480)  \text{mm}^2$	$@600 \mathrm{mm}: (706 \times 534) \mathrm{mm}^2$
imaging devices lens.	@800 mm : $(780 \times 520)$ mm <sup>2</sup>	@1000 mm : $(1422 \times 800) \text{ mm}^2$	@1000 mm : $(1178 \times 890) \mathrm{mm}^2$
Transformation between sensor lens and TCP ${}^{TCP}_{s_t}T$	${}^{TCP}_{s_1} t: (0, 0, -602) \text{ mm}$ ${}^{TCP}_{s_1} r: (0, 0, 0)^{\circ}$	${}^{TCP}_{s_2} t: (0, 0, -600) \text{ mm} \\ {}^{TCP}_{s_2} r: (0, 0, 0)^{\circ}$	${}^{TCP}_{s_{3,4}} t: (0,0,-600) \text{ mm} \ {}^{TCP}_{s_{3,4}} r: (0,0,0)^{\circ}$
Transformation between imaging devices of each sensor ${}^{S_1}_{S_2}T, {}^{S_3}_{S_4}T$	$\int_{s_2}^{s_1} t : (217.0, 0, 8.0) \text{ mm}$ $\int_{s_2}^{s_1} r : (0, -20.0, 0)^{\circ}$		${}^{s_3}_{s_4} t : (65.0, 0, 0) \text{ mm}$ ${}^{s_3}_{s_4} r : (0, 0, 0)^\circ$
Transformation between both sensors $s_3^{S_1}T$	${}^{s_1}_{s_3}t:(348.0,-81.0,42.0)\mathrm{mm},{}^{s_1}_{s_3}r$	· : (0.52, 0.56, 0.34) °	

**Table A6.** Imaging parameters of the sensors  $s_1$  and  $s_2$ .

**Table A7.** Overview of features and occlusion objects used for verification steps and simulation-based analysis.

Feature	$f_0$	$f_1, f_1^*$	$f_2$	f <sub>3</sub>	κ <sub>1</sub>	κ2
Topology	Point	Slot	Circle	Half-Sphere	Icosahedron	Octaeder
Generalized Topology	-	Square	Square	Cube	-	-
Dimensions in mm	$l_{f_o}=0$	$l_{f_1} = 50, \\ h_{f_1^*} = 30$	$l_{f_2} = 20$	$l_{f_2} = 40, \\ h_{f_2} = 40$	edge length: $\approx 14.0$	edge length: $\approx 20.0$
Translation vector in object's frame $t_o = (x_o, y_o, z_o)^T$ in mm	$\left(\begin{array}{c} 0.0\\ 0.0\\ 0.0\end{array}\right)$	$\left(\begin{array}{c} 0.0\\ 0.0\\ 0.0\end{array}\right)$	$\left(\begin{array}{c} 75.0\\ 150.0\\ 20.0 \end{array}\right)$	$\left(\begin{array}{c}120.0\\30.0\\0.0\end{array}\right)$	$\left(\begin{array}{c} -67.5\\ 0.0\\ 240.0 \end{array}\right)$	$\left(\begin{array}{c}117.5\\100.0\\445.0\end{array}\right)$
Rotation in Euler Angles in object's frame $r_o(\gamma_s^x, \beta_s^y, \alpha_s^z)$ in °	(0,0,0)	(0,0,0)	(0,20,0)	(0,0,0)	(0,0,0)	(0,0,0)

Viewpoint Constraint	Description	Approach
1	Two sensors( $s^1$ , $s^2$ ) with two imaging devices each: $\{s_1^1, s_2^1, s_3^2, s_4^2\} \in \tilde{S}$ . The imaging parameters of all devices are specified in Table A6.	Linear algebra and geometry
2	Relative orientation to the object's frame: ${}^{o}r_{s}(\alpha_{s}^{z}=\gamma_{s}^{x}=0,\beta_{s}^{y}=6.64^{\circ}).$	Linear algebra and geometry
3	A planar rectangular object with three different features $\{f_1, f_2, f_3\} \in F$ (see Table A7).	Linear algebra, geometry, and trigonometry
4–5	The workspace of the second imaging device is restricted in the <i>z</i> -axis to the following working distance $z_{s_2} > 450$ mm.	Linear algebra and geometry
6	Two objects with the form of an icosahedron ( $\kappa_1$ ) and a octahedron ( $\kappa_2$ ) occlude the visibility of the features.	Linear algebra, ray-casting, and CSG Boolean Operations
7	All constraints must be satisfied by all four imaging devices simultaneously.	Linear algebra and CSG Boolean Operations
8	Both sensors are attached to a six-axis industrial robot. The robot has a workspace of a half-sphere with a working distance of 1000 mm–1800 mm.	CSG Boolean Operation
9	All features from the set $G$ must be captured simultaneously.	CSG Boolean Operation

**Table A8.** Overview and description of the viewpoint constraints considered for the simulation-based analysis.

# Appendix B. Algorithms

Algorithm A1 Extreme Viewpoint Characterization of the Constrained Space  $C_1$ .

- 1. Consider a constant sensor orientation  $_{ref}r_s^{fix}$  to acquire a feature *f*.
- 2. Position the sensor so that the *k* vertex of the frustum space lies at the feature's origin,

$$_{ref}\boldsymbol{p}_{s,k}(_{ref}\boldsymbol{t}_{s}(\boldsymbol{V}_{k}^{\mathcal{I}_{s}}=B_{f}),_{ref}\boldsymbol{r}_{s}^{fix}).$$

3. Let the coordinates of the *k* vertex of the constrained space  $_{ref}C_1$  be equal to the translation vector of the sensor frame  $B_s^{ref}$ :

$$_{ref}\boldsymbol{V}_{k}^{ref}\boldsymbol{\mathcal{C}}_{1}={}_{ref}\boldsymbol{t}_{s}(\boldsymbol{B}_{s}^{ref}).$$

- 4. Repeat Steps 2 and 3 for all *l* vertices of the frustum space.
- 5. Connect all vertices from  $V^{ref}{}^{C_1}$  analogously to the vertices of the frustum space  $V^{\mathcal{I}_s}$  to obtain the  ${}_{ref}{}^{C_1}$  manifold.

# **Algorithm A2** Characterization of the occlusion space $C_6^{occl}$ .

1. Compute a set of view rays  $\varsigma_{g_{f,c}}(m, n) \in \Sigma$  for each surface point  $g_{f,c}$  using a set of direction vectors  $\sigma_{m,n}$ :

$$\boldsymbol{\varsigma}_{g_{f,c}}(m,n) = \boldsymbol{g}_{f,c} + \boldsymbol{\sigma}_{m,n}(\sigma_m^x, \sigma_n^y, \boldsymbol{r}_s^{f,x}).$$

The direction vectors span a  $m \times n$  grid of equidistant rays with a discretization step size  $d^{\varsigma}$ . The aperture angles of the view rays correspond to the maximal aperture of a previously characterized C-space C.

2. Test all view rays,  $\forall \varsigma_{g_{f,c}}(m,n) \in \Sigma$ , for occlusion against each rigid body  $\kappa \in K$  using ray casting. Let the collision points at the rigid bodies be denoted as:

$$\boldsymbol{q}_{f}^{occl,\kappa} \in Q_{f}^{occl,\kappa}$$

3. Shoot an occlusion ray,  $\varsigma_{g_{f,c}}^{occl,\kappa}$ , from each surface point  $g_{f,c}$  to all occluding points of the set  $\forall q_f^{occl,\kappa} \in Q_f^{occl,\kappa}$ :

$$\boldsymbol{\varsigma}_{g_{f,c}}^{occl,\kappa}(t) = \boldsymbol{q}_{f}^{occl,\kappa} + t \cdot (\boldsymbol{q}_{f}^{occl,\kappa} - \boldsymbol{g}_{f,c}).$$

4. Select one point,  ${}^*q_f^{occl,k}$ , from each occlusion ray  $\forall \varsigma_{g_{f,c}}^{occl,\kappa} \in \Sigma^{occl,\kappa}$  considering that this must lie beyond the constrained space. Let these points be elements of the set  ${}^*Q_f^{occl,\kappa}$ .

$${}^{*}\boldsymbol{q}_{f}^{occl,k} \in ({}^{*}\boldsymbol{Q}_{f}^{occl,\kappa},\boldsymbol{\varsigma}_{g_{f,c}}^{occl,\kappa}), {}^{*}\boldsymbol{q}_{f}^{occl,k} \notin \mathcal{C}$$

5. Compute the convex hull spanned by all points of  $Q_f^{occl,k}$  and  ${}^*Q_f^{occl,\kappa}$ . The convex hull corresponds to the manifold of  $C_{,6}^{occl,\kappa}$ :

$$\mathcal{C}_{6}^{occl,\kappa} \leftarrow \mathcal{H}_{hull}(Q_{f}^{occl,k}, {}^{*}Q_{f}^{occl,\kappa}).$$

- 6. Compute the occlusion space,  $C_6^{occl,\kappa}$ , for all rigid bodies,  $\forall \kappa \in K$ , repeating Steps 2 until 5.
- 7. The occlusion space for all rigid bodies corresponds to the CSG Boolean Union operation of all individual occluding spaces:

$$\mathcal{C}_6^{occl} = \bigcup_{\kappa \in K} \mathcal{C}_6^{occl,\kappa}.$$

8. The occlusion space is integrated with the *C*-space spanned by other viewpoint constraints using a CSG Boolean Difference operation:

$$\mathcal{C} = \left(\bigcap_{i=1, i\neq 6}^{j} \mathcal{C}_{i}\right) \setminus \mathcal{C}_{6}^{occl}.$$

**Algorithm A3** Characterization of C-space  $C^{\tilde{S}_1}$  to integrate viewpoint constraints of a second imaging device  $s_2$ .

1. Compute the *C*-space of the first device, considering a fixed orientation  ${}_{s_1} r_s^{fix}$  and any further viewpoint constraints  $\tilde{C}^{s_1}$ .

$$\mathcal{C}^{s_1}(_{s_1}\boldsymbol{r}^{fix}_s, \widetilde{C}^{s_1}).$$

2. Compute the *C*-space for the second imaging device, taking into account any viewpoint constraints and the previously defined orientation of the first imaging device, using the rigid orientation between both devices  ${}_{s_7}r_s^{fix}({}_{s_1}r_s^{fix}) = {}_{s_1}^{s_2}R_s \cdot {}_{s_1}r^{fix}$ :

$$\mathcal{C}^{s_2}({}_{s_2}\boldsymbol{r}^{fix}_s({}_{s_1}\boldsymbol{r}^{fix}_s),\widetilde{C}^{s_2}).$$

3. Compute the sensor pose that the first device assumes when computing  $C^{s_2}$ , using the rigid transformation between both devices:

$$_{s_1} p_s^{\mathcal{C}^{s_2}} = {}^{s_1}_{s_2} T_s \cdot {}^{s_2}_{s_2} p_s({}^{s_2} t_s = B_f)$$

4. Duplicate the manifold of  $C^{s_2}$  and translate it to the position vector of  ${}_{s_1}p_s^{C^{s_2}}$ . The C-space  $C_7^{s_{1,2}}$  corresponds to this translated manifold:

$$\mathcal{C}_{7}^{s_{1,2}} = translation(\mathcal{C}^{s_2}, \mathbf{t}_s^{\mathcal{C}^{s_2}}).$$

5. Integrate  $C_7^{s_{1,2}}$  with the *C*-space of the first imaging device using a Boolean Intersection operation:

$$\mathcal{C}^{S_1} = \mathcal{C}^{S_1} \bigcap \mathcal{C}^{S_{1,2}}_7.$$

Algorithm A4 Integration of *C*-space for multiple features.

1. Compute all *n C*-spaces for each feature  $\forall f_m \in F$  with m = 1, ..., n.

$$f_m \mathcal{C} := \mathcal{C}(\mathbf{r}_s^{fix}, \widetilde{C}(f_m))$$

2. Compute the joint *C*-space by intersecting all *n C*-spaces:

$${}^{F}\mathcal{C} = {}^{F}\mathcal{C}_{9} = \bigcap_{f_{m} \in F}^{n} {}^{f_{m}}\mathcal{C}$$

Algorithm A5 Strategy for the integration of viewpoint constraints.

- 1. Consider a fixed sensor orientation  $_{s_1}r_s^{fix}$  for the reference imaging device  $s_1$ .
- 2. Compute the *C*-spaces manifolds

$$f_m \mathcal{C}_{1-6}^{s_t}(s_t \mathbf{r}_s(s_1 \mathbf{r}_s^{fix}))$$

of imaging device  $s_t$  for each feature  $\forall f_m \in F$ , considering the sensor orientation of the first device  $s_1 r_s^{fix}$  and the viewpoint constraints 1 - 6.

3. Compute the *C*-space of all features for sensor  $s_t$ :

$${}^{F}\mathcal{C}^{s_t} = \bigcap^{n} {}^{f_m}\mathcal{C}^{s_t}_{1-6}.$$

- 4. Repeat Steps 1–3 for all imaging devices  $\forall s_t \in \widetilde{S}$ .
- 5. Compute the *C*-space for all *u* imaging devices, e.g., for  $s_1$ :

$${}^{F}\mathcal{C}^{\widetilde{S}_{1}} = {}^{F}\mathcal{C}^{s_{1}} \bigcap_{s_{t} \in \widetilde{S}} {}^{F}\mathcal{C}^{s_{1},s_{t}}_{7}.$$

6. Intersect the robot workspace to obtain the final C-space b, e.g., for  $s_1$ :

$${}^{F}\mathcal{C}^{\widetilde{S}_{1}}={}^{F}\mathcal{C}^{\widetilde{S}_{1}}\bigcap\mathcal{C}_{8}.$$

### Algorithm A6 Computation of View Rays for Occlusion Space.

1. Considering the simplification of the feature topology (cf. Section 2.4), let a set of view rays denoted by  $\tau_{f,c,l,m} \in \Sigma_c, c = \{0, 1, 2, 3, 4\}$  be shot at each feature corner point  $g_{f,c} \in \mathbb{R}^3$  with following direction vectors  $\sigma_{c,l,m}(\sigma_{c,l}^x, \sigma_{c,m}^y) \in \mathbb{R}^3$ :

$$\boldsymbol{\tau}_{f.c.m.n} = \boldsymbol{g}_{f.c} + \boldsymbol{\sigma}_{c.m.n}(\sigma_m^x, \sigma_n^y)$$

2. The direction vectors are characterized by a grid of equidistant rays, which can be expressed by means of the aperture angles  $\sigma_m^x$  and  $\sigma_n^y$ :

$$-\frac{\sigma_{max}^{x}}{2} \leq \sigma_{m}^{x} \leq \frac{\sigma_{max}^{x}}{2} \text{ and } -\frac{\sigma_{max}^{y}}{2} < \sigma_{n}^{y} < \frac{\sigma_{max}^{y}}{2}.$$

The maximal aperture angles  $\sigma_{max}^x$  and  $\sigma_{max}^y$  can simply correspond to the FOV angles of the sensor. An efficient alternative is to consider the aperture angles C, which already comprises the FOV angles and other constraints. The total number of rays depends on the chosen step size  $d^{\sigma}$  for computing the equidistant view rays:

$$l \in [1, \dots, \frac{\sigma_{max}^{x} - \sigma_{min}^{x} + 1}{d^{\sigma}}],$$
$$m \in [1, \dots, \frac{\sigma_{max}^{y} - \sigma_{min}^{y} + 1}{d^{\sigma}}].$$

# **Appendix C. Figures**



**Figure A1.** Characterization of the the *C*-space,  $C_2(R_s)$  in SE(3), comprising multiple sensor orientations  $R_s$ .



Figure A2. Geometrical analysis for rotation around *z*-axis.



(c) 3. Vertex of  $V_3^{\mathcal{C}_3}$ .

(**d**) 4. Vertex of  $V_4^{\mathcal{C}_4}$ .





(a) Rendered scene at extreme viewpoint:  ${}^{f_1}p_{s,1} \in C_{3,1}({}^{f_1}r_{s,1}).$ 



(**b**) Rendered scene at extreme viewpoint:  ${}^{f_1}p_{s,2} \in C_{3,2}({}^{f_1}r_{s,2})$ .



(c) Rendered scene at extreme viewpoint:  $f_1^* p_{s,3} \in C_{3,3}(f_1^* r_{s,3}^0)$ .

**Figure A4.** Rendered scenes at the extreme viewpoints  $f_1 p_{s,1}$ ,  $f_1 p_{s,2}$ , and  $f_1^* p_{s,3}$  for verifying that the whole feature geometry lies entirely within the corresponding frustum spaces. Each figure displays the resulting frustum space (manifold with green edges), corresponding rendering point cloud (green points), and depth image (2D image in color map) at each extreme viewpoint.



**Figure A5.** Exemplary flexible characterization of the viewpoint constraints (e.g., kinematic errors and sensor accuracy) using different scaling vectors for each vertex.



**Figure A6.** Occlusion C-space in SE(3) (red manifold) and the occlusion-free C-space (blue manifold) to acquire a square feature  $f_1$ , considering an occlusion body  $\kappa$  (icosahedron in orange).



**Figure A7.** Characterization of the *C*-space,  ${}^{F}C$ , in *SE*(3) to acquire a set of features  $\{f_1, f_2\} \in F$  being characterized by the intersection of the individual *C*-spaces  ${}^{f_1}C$  and  ${}^{f_2}C$ .

# References

- 1. Kragic, D.; Daniilidis, K. 3-D Vision for Navigation and Grasping. In *Springer Handbook of Robotics*; Siciliano, B.; Khatib, O., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 811–824. [CrossRef]
- 2. Peuzin-Jubert, M.; Polette, A.; Nozais, D.; Mari, J.L.; Pernot, J.P. Survey on the View Planning Problem for Reverse Engineering and Automated Control Applications. *Comput.-Aided Des.* **2021**, *141*, 103094. [CrossRef]
- Gospodnetić, P.; Mosbach, D.; Rauhut, M.; Hagen, H. Viewpoint placement for inspection planning. *Mach. Vis. Appl.* 2022, 33, 2. [CrossRef]
- 4. Chen, S.; Li, Y.; Kwok, N.M. Active vision in robotic systems: A survey of recent developments. *Int. J. Robot. Res.* 2011, 30, 1343–1377. [CrossRef]

- 5. Tarabanis, K.A.; Allen, P.K.; Tsai, R.Y. A survey of sensor planning in computer vision. *IEEE Trans. Robot. Autom.* **1995**, *11*, 86–104. [CrossRef]
- 6. Tan, C.S.; Mohd-Mokhtar, R.; Arshad, M.R. A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. *IEEE Access* **2021**, *9*, 119310–119342. [CrossRef]
- Tarabanis, K.A.; Tsai, R.Y.; Allen, P.K. The MVP sensor planning system for robotic vision tasks. *IEEE Trans. Robot. Autom.* 1995, 11, 72–85. [CrossRef]
- 8. Scott, W.R.; Roth, G.; Rivest, J.F. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.* (*CSUR*) 2003, 35, 64–96. [CrossRef]
- 9. Mavrinac, A.; Chen, X. Modeling Coverage in Camera Networks: A Survey. Int. J. Comput. Vis. 2013, 101, 205–226. [CrossRef]
- 10. Kritter, J.; Brévilliers, M.; Lepagnot, J.; Idoumghar, L. On the optimal placement of cameras for surveillance and the underlying set cover problem. *Appl. Soft Comput.* **2019**, *74*, 133–153. [CrossRef]
- 11. Scott, W.R. Performance-Oriented View Planning for Automated Object Reconstruction. Ph.D. Thesis, University of Ottawa, Ottawa, ON, Canada, 2002.
- Cowan, C.K.; Kovesi, P.D. Automatic sensor placement from vision task requirements. *IEEE Trans. Pattern Anal. Mach. Intell.* 1988, 10, 407–416. [CrossRef]
- Cowan, C.K.; Bergman, A. Determining the camera and light source location for a visual task. In Proceedings of the 1989 International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; IEEE Computer Society Press: Washington, DC, USA, 1989; pp. 509–514. [CrossRef]
- Tarabanis, K.; Tsai, R.Y. Computing occlusion-free viewpoints. In Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15–18 June 1992; IEEE Computer Society Press: Washington, DC, USA, 1992; pp. 802–807. [CrossRef]
- 15. Tarabanis, K.; Tsai, R.Y.; Kaul, A. Computing occlusion-free viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 279–292. [CrossRef]
- 16. Abrams, S.; Allen, P.K.; Tarabanis, K. Computing Camera Viewpoints in an Active Robot Work Cell. *Int. J. Robot. Res.* **1999**, 18, 267–285. [CrossRef]
- 17. Reed, M. Solid Model Acquisition from Range Imagery. Ph.D. Thesis, Columbia University, New York, NY, USA, 1998.
- Reed, M.K.; Allen, P.K. Constraint-based sensor planning for scene modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 2000, 22, 1460–1467. [CrossRef]
- Tarbox, G.H.; Gottschlich, S.N. IVIS: An integrated volumetric inspection system. *Comput. Vis. Image Underst.* 1994, 61, 430–444. [CrossRef]
- Tarbox, G.H.; Gottschlich, S.N. Planning for complete sensor coverage in inspection. *Comput. Vis. Image Underst.* 1995, 61, 84–111. [CrossRef]
- 21. Scott, W.R. Model-based view planning. Mach. Vis. Appl. 2009, 20, 47-69. [CrossRef]
- 22. Gronle, M.; Osten, W. View and sensor planning for multi-sensor surface inspection. *Surf. Topogr. Metrol. Prop.* **2016**, *4*, 024009. [CrossRef]
- Jing, W.; Polden, J.; Goh, C.F.; Rajaraman, M.; Lin, W.; Shimada, K. Sampling-based coverage motion planning for industrial inspection application with redundant robotic system. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 5211–5218. [CrossRef]
- 24. Mosbach, D.; Gospodnetić, P.; Rauhut, M.; Hamann, B.; Hagen, H. Feature-Driven Viewpoint Placement for Model-Based Surface Inspection. *Mach. Vis. Appl.* **2021**, *32*, 8. [CrossRef]
- Trucco, E.; Umasuthan, M.; Wallace, A.M.; Roberto, V. Model-based planning of optimal sensor placements for inspection. *IEEE Trans. Robot. Autom.* 1997, 13, 182–194. [CrossRef]
- 26. Pito, R. A solution to the next best view problem for automated surface acquisition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, 21, 1016–1030. [CrossRef]
- Stößel, D.; Hanheide, M.; Sagerer, G.; Krüger, L.; Ellenrieder, M. Feature and viewpoint selection for industrial car assembly. In Proceedings of the Joint Pattern Recognition Symposium, Tübingen, Germany, 30 August–1 September 2004; Springer: Berlin/Heidelberg, Germany; pp. 528–535. [CrossRef]
- 28. Ellenrieder, M.M.; Krüger, L.; Stößel, D.; Hanheide, M. A versatile model-based visibility measure for geometric primitives. In Proceedings of the Scandinavian Conference on Image Analysis, Joensuu, Finland, 19–22 June 2005; pp. 669–678. [CrossRef]
- 29. Raffaeli, R.; Mengoni, M.; Germani, M.; Mandorli, F. Off-line view planning for the inspection of mechanical parts. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2013**, *7*, 1–12. [CrossRef]
- Koutecký, T.; Paloušek, D.; Brandejs, J. Sensor planning system for fringe projection scanning of sheet metal parts. *Measurement* 2016, 94, 60–70. [CrossRef]
- Lee, K.H.; Park, H.P. Automated inspection planning of free-form shape parts by laser scanning. *Robot.-Comput.-Integr. Manuf.* 2000, 16, 201–210. [CrossRef]
- 32. Derigent, W.; Chapotot, E.; Ris, G.; Remy, S.; Bernard, A. 3D Digitizing Strategy Planning Approach Based on a CAD Model. *J. Comput. Inf. Sci. Eng.* **2006**, *7*, 10–19. [CrossRef]

- Tekouo Moutchiho, W.B. A New Programming Approach for Robot-Based Flexible Inspection Systems. Ph.D. Thesis, Technical University of Munich, Munich, Germany, 2012.
- Park, J.; Bhat, P.C.; Kak, A.C. A Look-up Table Based Approach for Solving the Camera Selection Problem in Large Camera Networks. In Workshop on Distributed Smart Cameras in conjunction with ACM SenSys; Association for Computing Machinery: New York, NY, USA, 2006; pp. 72–76.
- González-Banos, H. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the Seventeenth Annual Symposium* on *Computational Geometry-SCG '01*; Souvaine, D.L., Ed.; Association for Computing Machinery: New York, NY, USA, 2001; pp. 232–240. [CrossRef]
- Chen, S.Y.; Li, Y.F. Automatic sensor placement for model-based robot vision. *IEEE Trans. Syst. Man Cybern. Part Cybern. Publ. IEEE Syst. Man Cybern. Soc.* 2004, 34, 393–408. [CrossRef] [PubMed]
- 37. Erdem, U.M.; Sclaroff, S. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Comput. Vis. Image Underst.* 2006, 103, 156–169. [CrossRef]
- Mavrinac, A.; Chen, X.; Alarcon-Herrera, J.L. Semiautomatic Model-Based View Planning for Active Triangulation 3-D Inspection Systems. *IEEE/ASME Trans. Mechatron.* 2015, 20, 799–811. [CrossRef]
- Glorieux, E.; Franciosa, P.; Ceglarek, D. Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection. *Robot.-Comput.-Integr. Manuf.* 2020, 61, 101843. [CrossRef]
- 40. Chen, S.Y.; Li, Y.F. Vision sensor planning for 3-D model acquisition. *IEEE Trans. Syst. Man Cybern. Part Cybern. Publ. IEEE Syst. Man Cybern. Soc.* **2005**, 35, 894–904. [CrossRef]
- Vasquez-Gomez, J.I.; Sucar, L.E.; Murrieta-Cid, R.; Lopez-Damian, E. Volumetric Next-best-view Planning for 3D Object Reconstruction with Positioning Error. Int. J. Adv. Robot. Syst. 2014, 11, 159. [CrossRef]
- Kriegel, S.; Rink, C.; Bodenmüller, T.; Suppa, M. Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects. *J.-Real-Time Image Process.* 2015, 10, 611–631. [CrossRef]
- Lauri, M.; Pajarinen, J.; Peters, J.; Frintrop, S. Multi-Sensor Next-Best-View Planning as Matroid-Constrained Submodular Maximization. *IEEE Robot. Autom. Lett.* 2020, *5*, 5323–5330. [CrossRef]
- 44. Waldron, K.J.; Schmiedeler, J. Kinematics. In *Springer Handbook of Robotics*, Siciliano, B., Khatib, O., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 11–36. [CrossRef]
- 45. Beyerer, J.; Puente León, F.; Frese, C. Machine Vision; Springer: Berlin/Heidelberg, Germany, 2016. [CrossRef]
- Biagio, M.S.; Beltrán-González, C.; Giunta, S.; Del Bue, A.; Murino, V. Automatic inspection of aeronautic components. *Mach. Vis. Appl.* 2020, 28, 591-605. [CrossRef]
- 47. Bertagnolli, F. Robotergestützte Automatische Digitalisierung von Werkstückgeometrien Mittels Optischer Streifenprojektion; Messtechnikund Sensorik, Shaker: Aachen, Germany, 2006.
- Raffaeli, R.; Mengoni, M.; Germani, M. Context Dependent Automatic View Planning: The Inspection of Mechanical Components. Comput. Aided Des. Appl. 2013, 10, 111–127. [CrossRef]
- 49. Beasley, J.; Chu, P. A genetic algorithm for the set covering problem. Eur. J. Oper. Res. 1996, 94, 392–404. [CrossRef]
- 50. Mittal, A.; Davis, L.S. A General Method for Sensor Planning in Multi-Sensor Systems: Extension to Random Occlusion. *Int. J. Comput. Vis.* 2007, *76*, 31–52. [CrossRef]
- Kaba, M.D.; Uzunbas, M.G.; Lim, S.N. A Reinforcement Learning Approach to the View Planning Problem. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5094–5102. [CrossRef]
- 52. Lozano-Pérez, T. Spatial Planning: A Configuration Space Approach. In *Autonomous Robot Vehicles*; Cox, I.J., Wilfong, G.T., Eds.; Springer: New York, NY, USA, 1990; pp. 259–271. [CrossRef]
- 53. Latombe, J.C. *Robot Motion Planning*; The Springer International Series in Engineering and Computer Science, Robotics; Springer: Boston, MA, USA, 1991; Volume 124. [CrossRef]
- 54. LaValle, S.M. Planning Algorithms; Cambridge University Press: Cambridge, UK, 2006.
- 55. Ghallab, M.; Nau, D.S.; Traverso, P. *Automated Planning*; Morgan Kaufmann and Oxford; Elsevier Science: San Francisco, CA, USA, 2004.
- 56. Frühwirth, T.; Abdennadher, S. *Essentials of Constraint Programming*; Cognitive Technologies; Springer: Berlin, Germany; London, UK, 2011.
- 57. Trimesh. Trimesh Github Repository. Available online: https://github.com/mikedh/trimesh (accessed on 16 July 2023).
- 58. Roth, S.D. Ray casting for modeling solids. Comput. Graph. Image Process. 1982, 18, 109–144. [CrossRef]
- 59. Glassner, A.S. An Introduction to Ray Tracing; Academic: London, UK, 1989.
- 60. Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* **1983**, *29*, 551–559. [CrossRef]
- Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Italy, 26–28 June 2006; Eurographics Association: Goslar, Germany, 2006; pp. 61–70.
- Bauer, P.; Heckler, L.; Worack, M.; Magaña, A.; Reinhart, G. Registration strategy of point clouds based on region-specific projections and virtual structures for robot-based inspection systems. *Measurement* 2021, 185, 109963. [CrossRef]
- 63. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A. ROS: An open-source Robot Operating System. In *Proceedings of the ICRA Workshop on Open Source Software*; IEEE: Piscataway, NJ, USA, 2009; Volume 3, p. 5.

- 64. Magaña, A.; Bauer, P.; Reinhart, G. Concept of a learning knowledge-based system for programming industrial robots. *Procedia CIRP* **2019**, *79*, 626–631. [CrossRef]
- Magaña, A.; Gebel, S.; Bauer, P.; Reinhart, G. Knowledge-Based Service-Oriented System for the Automated Programming of Robot-Based Inspection Systems. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1511–1518. [CrossRef]
- 66. Zhou, Q.; Grinspun, E.; Zorin, D.; Jacobson, A. Mesh arrangements for solid geometry. *ACM Trans. Graph.* **2016**, *35*, 1–15. [CrossRef]
- 67. Wald, I.; Woop, S.; Benthin, C.; Johnson, G.S.; Ernst, M. Embree. ACM Trans. Graph. 2014, 33, 1–8. [CrossRef]
- 68. Unity Technologies. Unity. Available online: https://unity.com (accessed on 16 July 2023).
- 69. Bischoff, M. ROS #. Available online: https://github.com/MartinBischoff/ros-sharp (accessed on 16 July 2023).
- Magaña, A.; Wu, H.; Bauer, P.; Reinhart, G. PoseNetwork: Pipeline for the Automated Generation of Synthetic Training Data and CNN for Object Detection, Segmentation, and Orientation Estimation. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 587–594. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.