

Article

Viewpoint Planning for Range Sensors Using Feature Cluster Constrained Spaces for Robot Vision Systems

Alejandro Magaña ^{1,*} , Michiel Vlaeyen ^{2,3} , Han Haitjema ^{2,3} , Philipp Bauer ¹ , Benedikt Schmucker ¹ and Gunther Reinhart ¹

¹ Institute for Machine Tools and Industrial Management, Technical University of Munich, 85747 Garching, Germany

² Department of Mechanical Engineering, KU Leuven, 3001 Leuven, Belgium

³ Flanders Make—Core Lab MaPS, KU Leuven, 3001 Leuven, Belgium

* Correspondence: alejandro.magana@tum.de

Abstract: The efficient computation of viewpoints for solving vision tasks comprising multi-features (regions of interest) represents a common challenge that any robot vision system (RVS) using range sensors faces. The characterization of valid and robust viewpoints is even more complex within real applications that require the consideration of various system constraints and model uncertainties. Hence, to address some of the challenges, our previous work outlined the computation of valid viewpoints as a geometrical problem and proposed feature-based constrained spaces (\mathcal{C} -spaces) to tackle this problem efficiently for acquiring one feature. The present paper extends the concept of \mathcal{C} -spaces to consider multi-feature problems using feature cluster constrained spaces (${}^G\mathcal{C}$ -spaces). A ${}^G\mathcal{C}$ -space represents a closed-form, geometrical solution that provides an infinite set of valid viewpoints for acquiring a cluster of features satisfying diverse viewpoint constraints. Furthermore, the current study outlines a generic viewpoint planning strategy based on ${}^G\mathcal{C}$ -spaces for solving vision tasks comprising multi-feature scenarios effectively and efficiently. The applicability of the proposed framework is validated on two different industrial vision systems used for dimensional metrology tasks.



Citation: Magaña, A.; Vlaeyen, M.; Haitjema, H.; Bauer, P.; Schmucker, B.; Reinhart, G. Viewpoint Planning for Range Sensors Using Feature Cluster Constrained Spaces for Robot Vision Systems. *Sensors* **2023**, *23*, 7964.

<https://doi.org/10.3390/s23187964>

Academic Editors: Alexander Ferrein and Michael Reke

Received: 28 July 2023

Revised: 3 September 2023

Accepted: 12 September 2023

Published: 18 September 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/>).

Keywords: viewpoint planning problem; vision system automation; constraint planning; robot vision system; range sensor

1. Introduction

The number of applications requiring machine vision capabilities (e.g., image-based inspection, object digitization, scene exploration, object detection, visual servoing, robot calibration, mobile navigation) has rapidly increased within the research and industry over the last decade [1,2]. Depending on the vision task and system complexity, the automation of such applications is necessary to provide effective and efficient solutions [3]. Over the past decade, robot vision systems (RVS) equipped with a range sensor have proven useful for automating these tasks. However, programming RVS to perform such tasks is considered a tedious and complex challenge. Programmers face a common challenge: the computation of necessary and valid viewpoints to perform the required vision task. This challenge is well known as the view(point) planning problem (VPP).

In an attempt to propose an efficient and generic framework to tackle the VPP, this publication extends the concept of feature-based constrained spaces (\mathcal{C} -spaces), introduced in our previous publication [4], and outlines a viewpoint planning strategy for multi-feature scenarios using feature cluster constrained spaces (${}^G\mathcal{C}$ -spaces).

1.1. \mathcal{C} -spaces for Solving the Viewpoint Planning Problem

The VPP can be comprehended by considering the following simple and generic formulation (F):

Problem 1. *What is the minimum number of viewpoints to acquire a set of features considering different viewpoint constraints?*

Figure 1 provides a graphical representation of this problem for acquiring a set of four features. Solving the VPP based on this formulation would require a monolithic strategy to efficiently solve a high-dimensional problem where multiple viewpoint constraints must be satisfied simultaneously. As a first step, we believe that a suitable reformulation and modularization of the VPP is necessary to reduce its complexity and outline more efficient solutions. In the present study, we propose first breaking the VPP into two subproblems, i.e., the viewpoint generation problem (VGP) and the set cover problem (SCP). The detailed reformulation of the VPP and its subproblems is handled within Section 3.

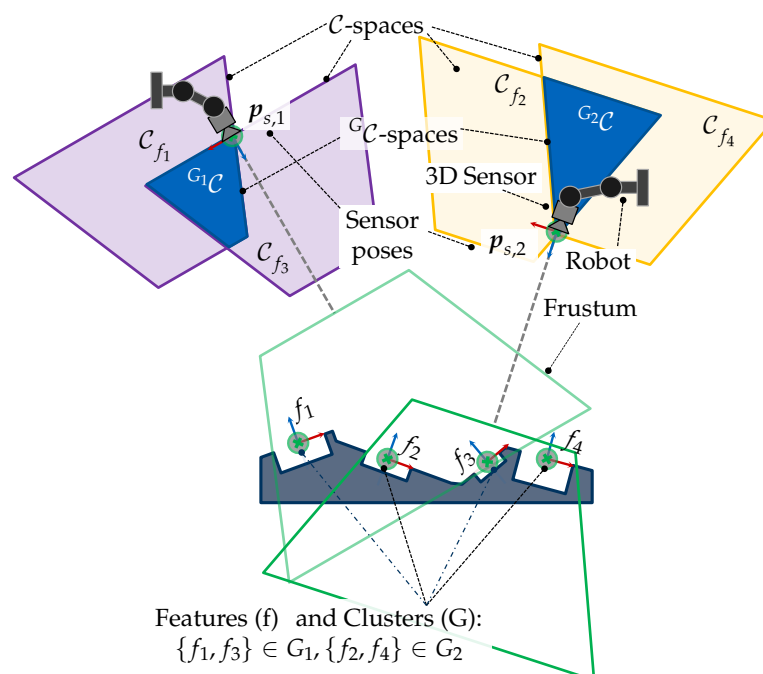


Figure 1. Simplified, graphical representation of the Viewpoint Planning Problem (VPP): how many viewpoints (sensor poses) are needed to acquire all four features? The present study proposes a viewpoint planning strategy based on feature cluster constrained spaces ($\mathcal{G}\mathcal{C}$ -spaces) to answer this question. A $\mathcal{G}\mathcal{C}$ -space spans a topological space in the special Euclidean $SE(3)$ with an infinite number of sensor poses $\forall p_s \in \mathcal{G}\mathcal{C}$ to acquire all features from a cluster G that satisfy a set of viewpoint constraints \tilde{C} , e.g., imaging parameters of the sensor, feature geometry, and orientation. $\mathcal{G}\mathcal{C}$ -spaces are computed based on the intersection of \mathcal{C} -spaces to acquire individual features. This example shows that two sensor poses ($p_{s,1}, p_{s,2}$) are required to capture the four visualized features. The selection of the sensor poses is performed straightforwardly by selecting any sensor pose within the $\mathcal{G}\mathcal{C}$ -spaces $\mathcal{G}\mathcal{C}_1$ and $\mathcal{G}\mathcal{C}_2$. The design of a strategy for selecting which features can be grouped and the characterization of the $\mathcal{G}\mathcal{C}$ -spaces are the focus of the present research.

Magaña et al. [4] focused on the first subproblem of the VPP, i.e., the VGP, which addresses the generation of valid viewpoints to acquire a single feature. The VGP was posed as a pure geometrical challenge that can be solved effectively using feature-based constrained viewpoint spaces (\mathcal{C} -spaces). A \mathcal{C} -space represents the spatial solution space in 6D (translation and rotation) that a set of viewpoint constraints spans. Therefore, \mathcal{C} -spaces represent an analytical, geometric, and closed-form solution that provides infinite valid

viewpoints to capture a feature. Especially within real applications, infinite solution spaces are advantageous for choosing seamlessly alternative solutions to compensate for model uncertainties or unknown constraints affecting the validity of a chosen viewpoint.

Although [4] delivered a sound solution for computing valid viewpoints and demonstrated the potential of \mathcal{C} -spaces for a simplified multi-feature acquisition, the authors did not pose the VGP as a multiple-feature problem. Moreover, a strategy for finding potential features that could be acquired together was not proposed. For example, it is generally not obvious which features could be acquired together in more complex scenarios with several features, such as the one depicted in Figure 1.

For these reasons, the present paper first extends the concept of \mathcal{C} -spaces for acquiring multiple features by introducing ${}^G\mathcal{C}$ -spaces. ${}^G\mathcal{C}$ -spaces can be analogously interpreted as spatial, continuous solution spaces spanned by a set of \mathcal{C} -spaces for acquiring a group of features. Hence, any sensor pose within it fulfills all viewpoint constraints \bar{C} and can be considered valid for capturing all regarded features. Figure 1 provides an exemplary and simplified illustration of the VPP and the spanned ${}^G\mathcal{C}$ -spaces for acquiring four features. The VPP can be easily solved by selecting one sensor pose from each ${}^G\mathcal{C}$ -space. Section 4.1 outlines the characterization of ${}^G\mathcal{C}$ -spaces based on \mathcal{C} -spaces.

In a second step, the present work addresses the second subproblem of the VPP, i.e., the SCP, and introduces a viewpoint plan strategy in Section 4.2 to identify the potential features that can be clustered together. Having identified the minimum number of necessary feature clusters, hence, viewpoints, the corresponding ${}^G\mathcal{C}$ -spaces can then be computed to solve these multi-viewpoint tasks.

In the last Section 5, the applicability and potential of the proposed approach are evaluated in the context of dimensional metrology tasks using two industrial vision systems with different range sensors.

1.2. Related Work

The VPP has been the focus of the research over the last three decades within a wide range of vision tasks that demand the computation of generalized viewpoints. A broad overview of the overall progress, challenges, and applications of the VPP is provided within various surveys [2,5–9]. Depending on the a priori knowledge required, the approaches for viewpoint planning can be roughly classified into *model-based* and *non-model-based* approaches. Since our framework falls under the first category, this section provides a more comprehensive overview of the related research following model-based approaches.

Model-based methods require minimal spatial information from the features, e.g., its position and orientation, can be categorized as *synthesis* or *sampling-based* approaches. On the one hand, synthesis approaches consider first the characterization of a continuous or discrete solution space. On the other hand, sampling-based techniques generate and evaluate viewpoints based on objective functions without the need to span a search space.

1.2.1. Synthesis

The synthesis of solution spaces (related terms: \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces, search space, configuration space, viewpoint space, visibility map, visibility matrix, visibility volumes, imaging space, scannability frustum, visual hull) was posed in the first studies addressing the VPP and has the advantage of providing a straightforward comprehension and spatial interpretation of the general problem.

The pioneering studies of [10,11] are considered among the first to consider the characterization of a continuous viewing space in \mathbb{R}^3 . Their works proposed analytical relationships to characterize multiple constraints geometrically for 2D sensors. However, their research concentrated on generating single viewpoints and did not consider a strategy for more complex applications requiring multiple viewpoints. Based on the analytical findings provided by the previous work of [10,11], Tarabanis et al. [5,12] introduced a model-based sensor planning system that characterizes a continuous search space taking

into account the occlusion constraints. The rest of the constraints are formalized based on objective functions.

The study of [13] extended and addressed some of the drawbacks of [5], such as the non-linearity and convergence guarantee of the objective functions. In their work, they opted to synthesize the 3D search spaces, as suggested by [10]. Since not all viewpoint constraints could be synthesized, the authors proposed a gradient-based approach to find valid sensor poses within the search space and to evaluate the rest of the constraints. In parallel, in a series of publications, Reed et al. [14,15] also followed an explicit characterization of the viewpoint space in \mathbb{R}^3 for range sensors integrating imaging, occlusion, and workspace constraints. However, the final selection of valid viewpoints required a discretization of the search space for finding a proper sensor orientation.

Another significant line of research was introduced by [16], who proposed characterizing a discretized viewpoint space. Analogously, ref. [17] introduced the measurability matrix, which extended the visibility matrix of [16] to three dimensions under the consideration of further constraints. Both publications [18,19] placed the final selection of viewpoints in the context of the SCP and considered various heuristic algorithms for finding optimal viewpoints. More recent works [20–22] followed and extended the work of [17], confirming the usability of visibility matrices and modularization of the VPP.

Furthermore, a handful of works [23–29] also considered the explicit characterization of search spaces for some specific constraints. Similar to others, the search for viable viewpoint candidates was performed using different optimization algorithms to assess the satisfiability of other constraints. Some of these studies, e.g., [24,27,28], also suggested identifying potential groups of features or surfaces first based on their positions and orientations before spanning a search space to increase the planning efficiency.

1.2.2. Sampling-Based

Unlike synthesis methods, sampling-based approaches do not rely on the explicit characterization of a search space. They evaluate the validity of each candidate viewpoint based on objective functions for each constraint on the viewpoint, which are solved by metaheuristic optimization algorithms, e.g., simulated annealing or evolutionary algorithms [30–34]. These approaches focus on the efficient formulation of objective functions to satisfy the viewpoint constraints and find valid solutions. By neglecting the characterization of a search space, which can be computationally expensive, sampling methods can be especially efficient within simple scenarios considering only a few constraints and features.

1.3. Need for Action

Due to the heterogeneity of vision tasks and RVS confronted with the VPP, many researchers (see Section 1.2) have presented tailor-made and sound solutions to address this problem. To our knowledge, a standard approach for solving the VPP has not yet been established within the research or the industry.

Our literature review revealed various ways of characterizing solution spaces to identify valid viewpoints that satisfy the VPP, either explicitly or implicitly. Implicit methods incorporating sampling techniques can be highly advantageous and computationally efficient, especially in straightforward scenarios with fewer constraints. However, modeling uncertainties to compensate for robust solutions in complex applications that consider multiple features and constraints poses a more challenging task when using objective functions. That is mainly because the validation of objective functions must be explicitly proofed for each computed viewpoint, and no alternative viewpoints are initially proposed. This point is particularly crucial in real-world applications, where modeling uncertainties cannot be ignored, and robust approaches for compensating for these uncertainties are necessary. The difficulties of handling the VPP by discretizing the valid viewpoint space can be mitigated by using analytical modeling to establish the validity of constraints in special Euclidean explicitly. Therefore, an approach that proposes modeling every constraint as a solution space has the potential to address the VPP explicitly. However, addressing the problem

this way requires an extensive analysis of the constraints and how they can be spatially modeled. While our previous work [4] addresses this challenge for single features, the current study extends the concept to multiple features and outlines a suitable strategy to tackle the VPP holistically considering the following points:

- **Multi-stage formulation:** The complexity of the VPP can be subdivided into multiple subproblems. Simple and more efficient solutions can then be individually formulated for each subproblem.
- **Model-based solution:** Assuming that a priori information about viewpoint constraints (vision system, object, and task) is available, this knowledge should be used in the most effective and analytical manner. Furthermore, in this study, it is assumed that each viewpoint constraint should be spatially modeled in the special Euclidean (6D) aligned to a synthesis approach (see Section 1.2). This can be carried out by characterizing each constraint as a topological space, i.e., \mathcal{C} -space. If all viewpoint constraints can then be modeled as topological spaces and integrated together into ${}^G\mathcal{C}$ -spaces, the search for viable candidates can be reduced to the selection of a sensor pose within such spaces.
- **Viewpoint planning strategy:** Taking into account the modularization of the VPP and the characterization of ${}^G\mathcal{C}$ -spaces, a superordinated, holistic viewpoint planning strategy must be outlined for delivering a final selection of valid viewpoint candidates.

To the best of our knowledge, no other works have utilized a feature-based approach and combined multiple infinite spaces to compute valid viewpoints while considering several constraints. Furthermore, no generic strategies can be employed in conjunction with ${}^G\mathcal{C}$ -spaces. For this reason, a tailor-made strategy must be outlined.

1.4. Outline

The outline of this research is shown in Figure 2. First, Section 2 introduces a generic domain model for RVSs. Due to the heterogeneity of the definitions of RVSs in different applications, this section limits the scope of the presented research and helps to evaluate the transferability of the proposed models to other systems. Then, the core modules of the framework presented in this report address all these points and are individually addressed in Sections 3–4.2.

Section 3 revises the formulation of the VPP and its sub-problems. Then, a new generic formulation of the VPP is introduced based on the concept of ${}^G\mathcal{C}$ -spaces. Using the introduced formulation of ${}^G\mathcal{C}$ -spaces, their characterization is comprehensively addressed in Section 4.1. Based on the concept of ${}^G\mathcal{C}$ -spaces, a holistic viewpoint planning strategy to tackle the VPP is presented in Section 4.2. Based on an academic example, the characterization of ${}^G\mathcal{C}$ -spaces and the proposed strategy is verified.

Finally, Section 5 evaluates the validity of ${}^G\mathcal{C}$ -spaces and the proposed strategy to solve the VPP using two real RVS and a metrology application. Finally, Section 6 presents a summary and the conclusions of this paper. In addition, a comprehensive data set of multiple supporting files, e.g., ${}^G\mathcal{C}$ -space meshes, renders, frames, can be found in the attachment of this paper.

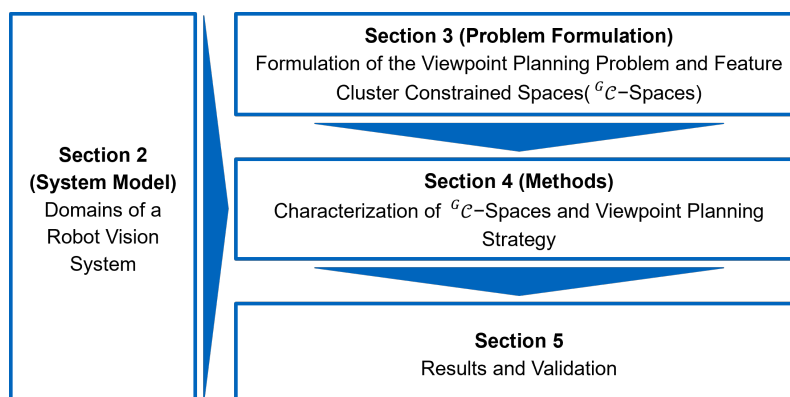


Figure 2. Outline.

2. Robot Vision System Domains

This section provides a brief overview of the most elementary components that build an RVS. A more exhaustive description of the models of all components is given in [4]. Figure 3 provides an overview of these elements. The present study considers many variables to describe the domains of a vision system comprehensively. To ease the identification and readability of variables, parameters, vectors, frames, and transformations, the index notation given in Table A2 is used.

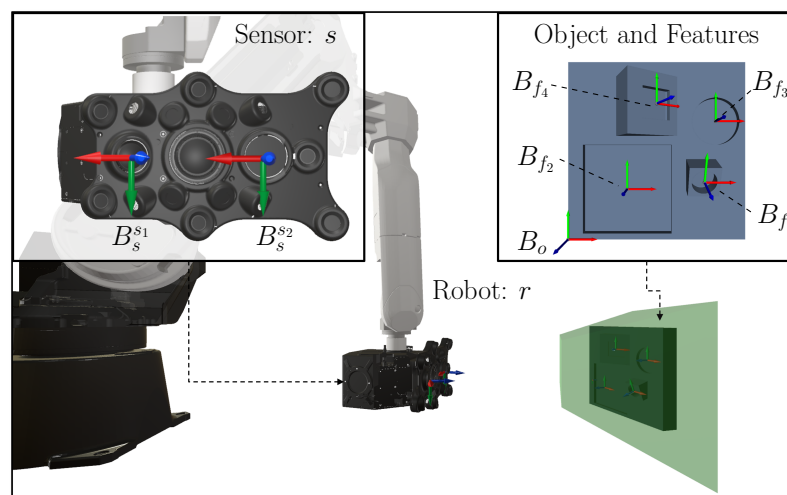


Figure 3. Overview of the most relevant components of a robot vision system.

2.1. Sensor

A sensor (s) (synonym: range camera sensor, 3D sensor, imaging system) is defined as a self-contained acquisition device comprising at least two imaging devices $\{s_1, s_2\} \in \tilde{S}$ (e.g., two cameras or one camera and one coded light source) capable of computing a range image containing depth information. The present study does not explicitly distinguish between the sensing principles of range sensors, e.g., structured light sensors or laser scanners. To ensure the applicability of the outlined solutions with different sensors, a generic and minimal imaging and kinematic model of a generic sensor is presented in this section. Figure A1 illustrates a simplified representation of the sensor's main components.

2.1.1. Frustum Space

The frustum space (\mathcal{I} -space) (related terms: visibility frustum, measurement volume, field-of-view space, sensor workspace) is characterized by a set of various sensor imaging parameters I_s , such as the horizontal and vertical field of view (FOV) angles θ_s^x and ψ_s^y , the middle working distance d_s , and the near h_s^{near} and far h_s^{far} viewing planes. The sensor

parameters I_s describe only the topology of the \mathcal{I} -space. The full characterization of the \mathcal{I} -space in the special Euclidean requires considering the sensor pose \mathbf{p}_s given by [4]. In the present study, the pose \mathbf{p} of any element is given by its translation $\mathbf{t} \in \mathbb{R}^3$ and a rotation. The rotation can be given by the Z-Y-X Euler angles $\mathbf{r} = (\alpha^z, \beta^y, \gamma^x)^T$ or as a rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$. The pose $\mathbf{p} \in SE(3)$ is given in the special Euclidean $SE(3) = \mathbb{R}^3 \times SO(3)$, where $SO(3) \subset \mathbb{R}^{3 \times 3}$ denotes the special orthogonal group [35].

$$\mathcal{I}_s := (\mathbf{p}_s, I_s) = \{ \mathbf{p}_s \in SE(3), \\ (d_s, h_s^{near}, h_s^{far}, \theta_s^x, \psi_s^y) \in I_s \}. \quad (1)$$

The resulting \mathcal{I}_s manifold spanned by the sensor imaging parameters for a given sensor pose is characterized by its vertices $\mathbf{V}_k^{\mathcal{I}_s} := \mathbf{V}_k(\mathcal{I}_s) = (V_k^x, V_k^y, V_k^z)^T$ with $k = 1, \dots, l$ and the corresponding joining edges and faces.

2.1.2. Kinematics

The sensor's kinematic model considers the following frames: B_s^{TCP} , B_s^{s1} , and B_s^{s2} . It is assumed that the frame of the tool center point (TCP) is located at the geometric center of the frustum space and that the frames B_s^{s1} and B_s^{s2} lie at the reference frame (e.g., the lenses or the focal length) corresponding to the imaging parameters I_s .

2.1.3. Sensor Orientation

A basic requirement for detecting a surface point is that the relative angle between a sensor and a surface is within the limits of the specific maximum angle of a sensor. The maximal incidence angle is normally provided by the sensor's manufacturer. This incidence angle, denoted as ${}^f\varphi_s$, is expressed as the angle between the feature's normal \mathbf{n}_f and the sensor's optical axis (z-axis) \mathbf{e}_s^z and can be calculated as follows:

$${}^f\varphi_s^{max} > |{}^f\varphi_s|, {}^f\varphi_s = \arccos\left(\frac{\mathbf{n}_f \cdot \mathbf{e}_s^z}{|\mathbf{n}_f| \cdot |\mathbf{e}_s^z|}\right). \quad (2)$$

Furthermore, the rotation of the sensor around the optical axis is given by the Euler angle α_s^z (related terms: swing, twist). In many cases, this angle can be arbitrarily chosen. Hence, most of the related works assume a fixed angle during the planning process. However, if the shape of the frustum is asymmetrical, it is reasonable to consider its optimization.

2.2. Object

The object (o) domain (related terms: object of interest, workpiece, object-, measurement-, inspection-, test-, or probing-object) holds all features to be captured. Since the framework introduced in this publication can be categorized as a feature-based approach, the object may have an arbitrary surface topology. However, if occlusion constraints are taken into account, then a surface model of the object must be considered if the object itself occludes the features.

2.3. Features

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

$$f := (B_f, G_f(L_f)). \quad (3)$$

Magaña et al. [4] proposed the generalization of the feature's topology using a square with a unique side length $\{l_f\} \in L_f$ to describe any 2D feature. This publication also regards this simplification. Moreover, it is assumed that the translation ${}^o\mathbf{t}_f$ and orientation

${}^0\mathbf{r}_f$ of the feature's origin is given in the object's coordinate system B_o (see Figure 3). Thus, the feature's frame is given as follows:

$$B_f = {}^0T_f({}^0\mathbf{t}_f, {}^0\mathbf{r}_f). \quad (4)$$

2.4. Robot

In the context of multi-feature scenarios and automation of vision tasks, it is common practice to use a robot (related terms: manipulator, industrial robot, positioning device) for positioning the sensor at the selected viewpoint candidates. Since the viewpoint's validity may depend on the robot kinematic, [4] considered the robot workspace as a further viewpoint constraint. The present study considers robots to be an optional element of a vision system. Therefore, robot constraints are not further discussed within this publication.

2.5. Viewpoint and Viewpoint Constraints

In the literature, there seems to be no general definition of a viewpoint v . The present report regards a feature-centered formulation and defines a viewpoint as a triple of the following core elements: a 6D sensor pose $\mathbf{p}_s \in SE(3)$ to acquire a group of features $f \in G$ considering a set of viewpoint constraints \tilde{C} :

$$v := (G, \mathbf{p}_s, \tilde{C}).$$

The set of viewpoint constraints $c_i \in \tilde{C}, i = 1, \dots, h$ includes all constraints c_i affecting the positioning of the sensor. Every constraint c_i can be interpreted as a collection of multiple variables of the vision system, e.g., the imaging parameters I_s , the feature geometry length l_f , the maximal incidence angle ${}^f\varphi_s^{max}$. A description of the viewpoint constraints considered in the scope of the current publication is given in Table A1.

2.6. Vision Task

A vision task is defined by the set of features F (synonyms: feature plan, inspection plan, feature space)

$$f_m \in F, m = 1, \dots, n$$

that must be acquired. A vision task is then fulfilled when there exists a viewpoint plan denoted as P that holds a finite number of k viewpoints

$$(G_j, \mathbf{p}_{s,j}, \tilde{C}) \in P, j = 1, \dots, k,$$

which guarantees the acquisition of all n features from F satisfying all viewpoint constraints \tilde{C} . Note that G_j denotes a subset of features $G_j \subseteq F$ and that the union of all subsets corresponds to

$$F = \bigcup_{j=1}^k G_j.$$

3. Formulation of the Viewpoint Planning Problem Based on $G\mathcal{C}$ -spaces

We are convinced that a multi-stage formulation of the VPP can reduce its overall complexity and enable more efficient solutions. Therefore, in our first study [4], we proposed to split the VPP into the VGP and SCP and focused on the first one.

Magaña et al. [4] attributes to the VGP (related terms: camera planning, optical camera placement) the calculation of a viewpoint to acquire a single feature considering the fulfillment of a set of viewpoint constraints. Furthermore, when considering a multi-feature application, the efficient selection of multiple viewpoints becomes necessary to accomplish the vision task, which results in a new problem, namely, the SCP. A simplified representation of the VPP and its subproblems are visualized in Figure 4.

3.1. The Viewpoint Generation Problem

The generation of viewpoints for acquiring one or multiple features can be treated as an isolated subproblem of the VPP. Hence, our previous study [4] focused on this subproblem, i.e., the VGP, and proposed its consistent and exhaustive formulation. The publication demonstrated that the VGP could be effectively handled as a purely geometrical problem and introduced the concept of \mathcal{C} -spaces as the backbone element to generate viewpoints capable of acquiring one feature.

The present study first summarizes the formulation of the VGP in the context of single-feature vision tasks [4]. Then, an extended reformulation of the VGP is placed to consider more complex vision tasks regarding a multi-feature acquisition. In this context, the formal definition of ${}^G\mathcal{C}$ -spaces based on \mathcal{C} -spaces is introduced as the core element to enable the acquisition of feature clusters.

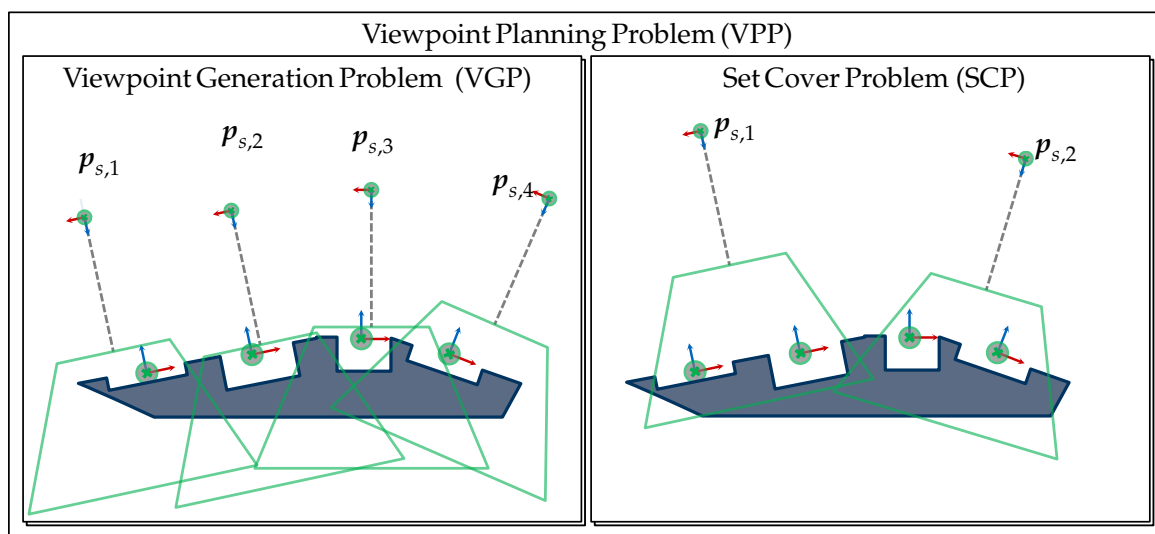


Figure 4. Modularization of the VPP and simplified representation of its subproblems. On the one hand, the VGP addresses the acquisition of a single feature by a viewpoint satisfying a set of constraints. On the other hand, the SCP seeks to reduce the number of required viewpoints to acquire all features.

3.1.1. VGP with \mathcal{C} -spaces

The concept of \mathcal{C} -spaces can be better understood considering a proper formulation of the VGP [4]:

Problem 2. Which is the \mathcal{C} -space ${}^f\mathcal{C}$ to acquire a single feature f considering a set of viewpoint constraints $\tilde{\mathcal{C}}$?

The mathematical definition of the \mathcal{C} -space denoted as ${}^f\mathcal{C} := \mathcal{C}(f, \tilde{\mathcal{C}})$ can now be introduced, considering that there exists a topological space in ${}^f\mathcal{C} \subseteq SE(3)$ that holds all valid sensor poses \mathbf{p}_s to acquire a feature f , considering a set of viewpoint constraints $\tilde{\mathcal{C}}$. This topological space \mathcal{C} is spanned by the Euclidean space \mathbb{R}^3 and the special orthogonal group $SO(3)$ [4]:

$$\begin{aligned}
 {}^f\mathcal{C} &= \mathbb{R}^3 \times SO(3) \\
 &= \{\mathbf{p}_s \in {}^f\mathcal{C}, \mathbf{p}_s \in SE(3)\} \\
 &= \{\mathbf{p}_s(\mathbf{t}_s, \mathbf{r}_s) \in {}^f\mathcal{C} \\
 &\quad | \mathbf{t}_s \in \mathbb{R}^3, \mathbf{r}_s \in SO(3)\}.
 \end{aligned} \tag{5}$$

Moreover, let the formulation of ${}^f\mathcal{C}$ be further specified and assume that for each viewpoint constraint $\forall c_i \in \tilde{\mathcal{C}}$ there exists an individual i \mathcal{C} -space denoted as ${}^f\mathcal{C}_i := \mathcal{C}_i(f, c_i)$, which represents the topological space where any sensor pose $p_s \in {}^f\mathcal{C}_i$ satisfies the constraint c_i . The intersection of all \mathcal{C} -spaces conforms to the joint \mathcal{C} -space ${}^f\mathcal{C}$ that fulfills all viewpoint constraints:

$${}^f\mathcal{C} = \bigcap_{c_i \in \tilde{\mathcal{C}}} \mathcal{C}_i(f, c_i), \tag{6}$$

and ${}^f\mathcal{C}$ can be considered a subset of any viewpoint constrained space, i.e., ${}^f\mathcal{C} \subseteq {}^f\mathcal{C}_i$.

An abstract representation of the \mathcal{C} -space ${}^{f_1}\mathcal{C}$ for acquiring f_1 , being constrained by three viewpoint constraints and its corresponding \mathcal{C} -spaces ${}^{f_1}\mathcal{C}_1, {}^{f_1}\mathcal{C}_2$, and ${}^{f_1}\mathcal{C}_3$, is depicted in Figure 5. Respectively, ${}^{f_2}\mathcal{C}$ spans the topological space for acquiring feature f_2 , being delimited by two \mathcal{C} -spaces, ${}^{f_2}\mathcal{C}_1$ and ${}^{f_2}\mathcal{C}_2$.

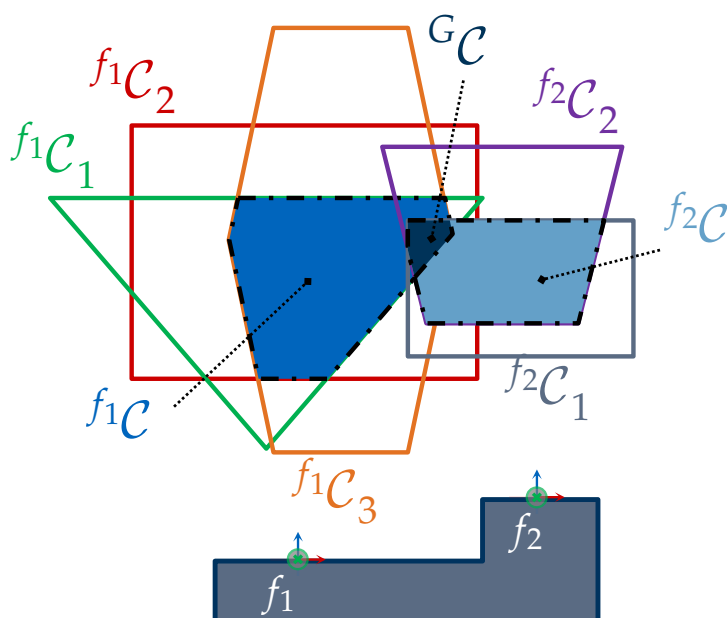


Figure 5. Abstract representation of the \mathcal{C} -spaces ${}^{f_1}\mathcal{C}$ and ${}^{f_2}\mathcal{C}$ that represent the infinite solution space for separately acquiring the features $\{f_1, f_2\} \in G$. ${}^{f_1}\mathcal{C}$ and ${}^{f_2}\mathcal{C}$ are characterized by different viewpoint constraints and their respective \mathcal{C} -spaces $\{{}^{f_1}\mathcal{C}_1, {}^{f_1}\mathcal{C}_2, {}^{f_1}\mathcal{C}_3\} \supseteq {}^{f_1}\mathcal{C}$ and $\{{}^{f_2}\mathcal{C}_1, {}^{f_2}\mathcal{C}_2\} \supseteq {}^{f_2}\mathcal{C}$. The intersection of the two \mathcal{C} -spaces ${}^{f_1}\mathcal{C}$ and ${}^{f_2}\mathcal{C}$ yield the ${}^G\mathcal{C}$ -space ${}^G\mathcal{C}$, which represents the infinite solution space to simultaneously acquire all features of the feature cluster G , fulfilling all viewpoint constraints.

3.1.2. VGP with ${}^G\mathcal{C}$ -spaces

Although the concept of \mathcal{C} -spaces provides a sound approach for characterizing an infinite solution space to acquire one feature, [4] did not comprehensively address its scalability in a multi-feature scenario. Hence, this subsection introduces the proper formulation and characterization of ${}^G\mathcal{C}$ -spaces to acquire a cluster of features based on \mathcal{C} -spaces, considering individual viewpoint constraints for each feature. The concept of \mathcal{C} -spaces can then be straightforwardly extended to a multi-feature problem, considering first the reformulation of Problem 2:

Problem 3. Which is the ${}^G\mathcal{C}$ -space ${}^G\mathcal{C}$ to acquire a cluster of features G considering a set of viewpoint constraints $\tilde{\mathcal{C}}$?

The ${}^G\mathcal{C}$ -space represents the multi-dimensional and continuous topological space to capture a feature cluster G . As its counterpart, i.e., ${}^f\mathcal{C}$, the ${}^G\mathcal{C}$ holds all valid sensor poses \mathbf{p}_s to acquire a subset of features G considering a set of viewpoint constraints \tilde{C} . The present study defines the following terms to formulate ${}^G\mathcal{C}$:

1. The ${}^G\mathcal{C}$ -space represents the intersection of all individual feature \mathcal{C} -spaces as defined by Equation (6):

$${}^G\mathcal{C} = \bigcap_{f_m \in G} \left(\bigcap_{c_i \in \tilde{C}} \mathcal{C}_i(f_m, c_i) \right). \quad (7)$$

2. If a ${}^G\mathcal{C}$ -space is a non-empty manifold, i.e., ${}^G\mathcal{C} \neq \emptyset$, there exists at least one sensor pose $\exists \mathbf{p}_s^G \in {}^G\mathcal{C}$ that fulfills all viewpoint constraints to acquire all features G .

Figure 5 depicts an abstract representation of the ${}^G\mathcal{C}$ -space ${}^G\mathcal{C}$ being characterized by the intersection of the individual constrained spaces ${}^{f_1}\mathcal{C}$ and ${}^{f_2}\mathcal{C}$ of the feature cluster $\{f_1, f_2\} \in G$.

3.2. The Set Cover Problem

${}^G\mathcal{C}$ -spaces provide closed solutions with an infinite set of viewpoints to acquire a feature cluster. However, the main question regarding which features can be acquired from the same viewpoint remains unanswered and is to be investigated within the second sub-problem of the VPP, i.e., the SCP. Hence, this section first outlines an adequate formulation of the SCP.

3.2.1. Problem Formulation

In general, the SCP may be formulated as an optimization problem or a decision problem. On the one hand, the optimization definition strives to find the minimal number of k viewpoints to capture all n features; this formulation was initially introduced in Problem 1. On the other hand, the SCP can be posed as a decision problem:

Problem 4. *Are k viewpoints sufficient to acquire a set of features F ?*

Although any approach addressing the VPP should prioritize the minimization of viewpoints, in our view, an optimization formulation may appear impractical and even ineffective in real applications regarding model uncertainties and feature-rich scenarios. Hence, for the benefit of pragmatism and as other authors, e.g., [15,16,18], have considered, we prefer to relax this requirement and strive for an acceptable number of viewpoints and prioritize robust and computationally efficient solutions.

3.2.2. Solving the SCP

A simple solution for Problem 4 is given by finding k iteratively. In this case, a first attempt is made to solve the problem considering an initial value, e.g., $k = 1$. If the k viewpoints are not sufficient to acquire F , then k is increased by one until there are enough viewpoints to acquire all features.

At this point, the challenge and motivation of the first subproblem of the VPP, i.e., the VGP, arises: how the validity of viewpoints can be efficiently and effectively assessed. A solution to this problem was comprehensively addressed using ${}^G\mathcal{C}$ -spaces (cf. Section 3.1). Hence, if the number of required viewpoints that fulfill a vision task can be previously approximated, its validation can be efficiently assessed based on ${}^G\mathcal{C}$ -spaces.

3.3. Reformulation of the VPP

Having addressed the individual subproblems of the VPP, its reformulation can then be posed aligned to a decision formulation of the SCP and the concept ${}^G\mathcal{C}$ -spaces:

Problem 5. Are k ${}^G\mathcal{C}$ -spaces sufficient to acquire a set of features F ?

Aligning the VPP to a decision formulation, the search for finding k potential feature clusters must be addressed in the first step. The number of k feature clusters can be iteratively found following the ground idea proposed in Section 3.2.2. An adequate strategy to efficiently find feature clusters is comprehensively introduced in Section 4.2.1.

Assuming that an adequate number of potential k feature clusters can be found, then exactly k ${}^G\mathcal{C}$ -spaces must be characterized to capture all features from F . Now, if all ${}^G\mathcal{C}$ -spaces exist, i.e., $\forall \{{}^{G_1}\mathcal{C}, \dots, {}^{G_k}\mathcal{C}\} \neq \emptyset$, it can be assumed there exists at least one valid sensor pose within each ${}^G\mathcal{C}$ -space. The viewpoint plan that fulfills the regarded vision task can be straightforwardly given by selecting one sensor pose from each ${}^G\mathcal{C}$ -space $\{p_{s,1}, \dots, p_{s,k}\} \in P$.

4. Methods

4.1. Feature Cluster Constrained Spaces

This section describes the definition of ${}^G\mathcal{C}$ -spaces based on the intersection of individual \mathcal{C} -spaces. First, a strategy to characterize the most elementary \mathcal{C} -space based on the sensor's imaging parameters, feature position, and sensor orientation is introduced. Then, some general terms are established to compute ${}^G\mathcal{C}$ -spaces and verify their usability based on an academic example.

4.1.1. \mathcal{C} -spaces

Aligned to the formulation of \mathcal{C} -spaces and treating the VGP as a geometrical problem, [4] demonstrated that a handful of viewpoint constraints can be straightforwardly characterized and integrated using linear algebra, geometric analysis, and CSG Boolean operations.

For instance, the fundamental \mathcal{C} -space denoted as \mathcal{C}_1 is characterized based on the \mathcal{I} -space (sensor imaging parameters), a feature of interest and a fixed sensor orientation. Our past study introduced two strategies (extreme viewpoint and homeomorphism interpretation) to characterize the manifold of \mathcal{C}_1 . This paper outlines a simpler variation of the homeomorphism formulation using an alternative reflecting pivot point. The detailed steps to characterize \mathcal{C}_1 are described in Algorithm 1 and visualized in a simplified 2D representation in Figure 6a–c. Moreover, Figure 6d demonstrates that any sensor pose within \mathcal{C}_1 is valid to acquire the feature f .

The \mathcal{C} -space \mathcal{C}_1 manifold represents the basis for characterizing further \mathcal{C} -spaces considering other viewpoint constraints, e.g., the feature geometry, kinematic errors, occlusion, and multi-sensors. The formulation, characterization, and integration of multiple viewpoint constraints fall outside the scope of this publication, see [4] for further details.

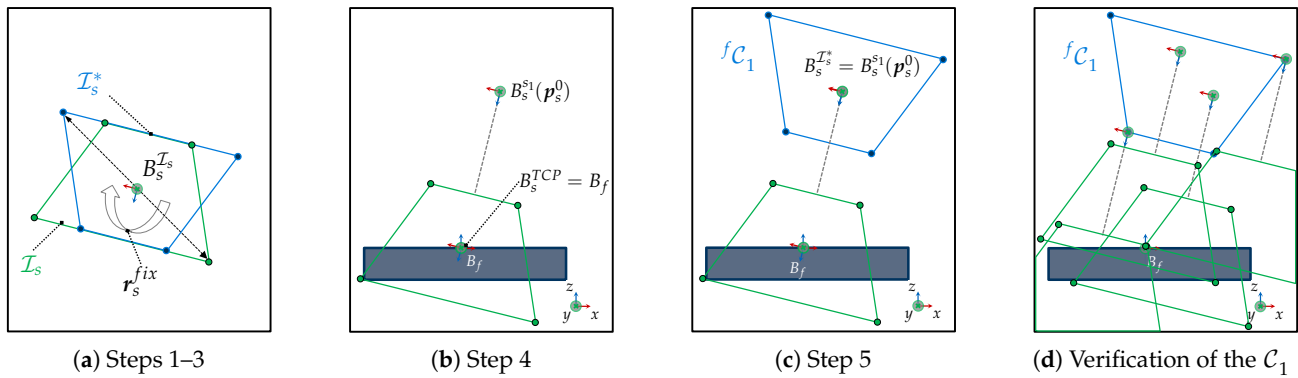


Figure 6. Characterization and Verification of the core \mathcal{C} -space \mathcal{C}_1 for capturing feature f considering a fixed sensor orientation and the \mathcal{I} -space. (a–c): Simplified visualization of the steps of Algorithm 1 to characterize the \mathcal{C} -space \mathcal{C}_1 , which considers the imaging parameters, the feature position, and a fixed sensor orientation. (d): Any sensor pose with $\mathbf{p}_s(\mathbf{r}_s = \mathbf{r}_s^{fix}) \in \mathcal{C}_1$ is valid to capture feature f .

Algorithm 1 Characterization of the \mathcal{C} -space \mathcal{C}_1 by reflection

1. Select a sensor orientation \mathbf{r}_s^{fix} for acquiring a feature f .
2. Rotate the \mathcal{I} -space manifold using the fixed rotation \mathbf{r}_s^{fix} and let this space be denoted as \mathcal{I}_s^* :

$$\mathcal{I}_s^* = rotate(\mathcal{I}_s, \mathbf{r}_s^{fix}).$$

3. Reflect \mathcal{I}_s^* over its origin $B_s^{\mathcal{I}_s^*}$:

$$\mathcal{I}_s^* = reflect(\mathcal{I}_s^*, B_s^{\mathcal{I}_s^*}).$$

4. Position the sensor's TCP frame at the feature's frame considering the sensor orientation \mathbf{r}_s^{fix} , i.e., $B_s^{TCP} = B_f$. Let this sensor pose be denoted as

$$\mathbf{p}_s^0 := \mathbf{p}_s(\mathbf{t}_s(B_s^{TCP} = B_f), \mathbf{r}_s^{fix}).$$

5. Translate the \mathcal{I}_s^* to the sensor lens frame $B_s^{\mathcal{I}_s^*} = B_s^{s1}(\mathbf{p}_s^0)$. The resulting manifold yields the \mathcal{C} -space

$${}^f\mathcal{C}_1 = translate(\mathcal{I}_s^*, B_s^{s1}(\mathbf{p}_s^0)).$$

4.1.2. ${}^G\mathcal{C}$ -spaces

Fixed Sensor Orientation

Recalling that \mathcal{C} -spaces only span a valid solution space for a fixed sensor orientation, it must be assumed that the validity of a ${}^G\mathcal{C}$ -space is also limited to a fixed sensor orientation.

Although an approach to characterize \mathcal{C} -space regarding multiple orientations was introduced in [4], we consider that in multi-feature scenarios, more efficient solutions can be obtained if the optimization of the sensor is first considered. By doing so, the problem complexity can be reduced for finding a valid sensor position in the Euclidean space considering an optimal sensor orientation.

Although \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces are exclusively valid for a defined sensor orientation, it can be assumed that a deviation of the sensor orientation can be implicitly compensated to some extent within the spanned topological spaces. However, the magnitude of the allowed deviation cannot be explicitly given, and it must be assumed that this is inconsistent within the topological spaces and depends on the remaining constraints.

Characterization

Assuming that the sensor orientation is known and the feature clusters are identified, the characterization of the ${}^G\mathcal{C}$ -space composed of individual \mathcal{C} -spaces is performed in two steps.

In the first step, all individual \mathcal{C} -spaces of each feature $f \in G$ are computed considering an individual viewpoint constraint set ${}^f\tilde{\mathcal{C}}$ with a fixed sensor orientation r_s^{fix} . In the second step, the \mathcal{C} -spaces of all features belonging to a cluster are merged with each other using a CSG Boolean intersection operation, as formulated in Equation (7). Figure 7 presents a simplified initial situation, characterization, and validation of a ${}^G\mathcal{C}$ -space ${}^{G_1}\mathcal{C}$ for capturing a set of two features $\{f_1, f_2\} \in G_1$.

In addition, Figure 8 considers a more complex scenario in \mathbb{R}^2 for capturing two feature clusters $\{f_1, f_3\} \in G_1$ and $\{f_2, f_4\} \in G_2$. The feature clusters are acquired with the sensor orientations $r_{s,1}^{fix}$ and $r_{s,2}^{fix}$. The two required ${}^G\mathcal{C}$ -spaces ${}^{G_1}\mathcal{C}(r_{s,1}^{fix})$ and ${}^{G_2}\mathcal{C}(r_{s,2}^{fix})$ are characterized by intersecting the corresponding individual \mathcal{C} -spaces of the regarded features.

A great deal of attention must be paid if the intersection of two \mathcal{C} -spaces yields an empty manifold; it can then be assumed that the corresponding features cannot be acquired simultaneously considering the regarded viewpoint constraints and sensor orientation.

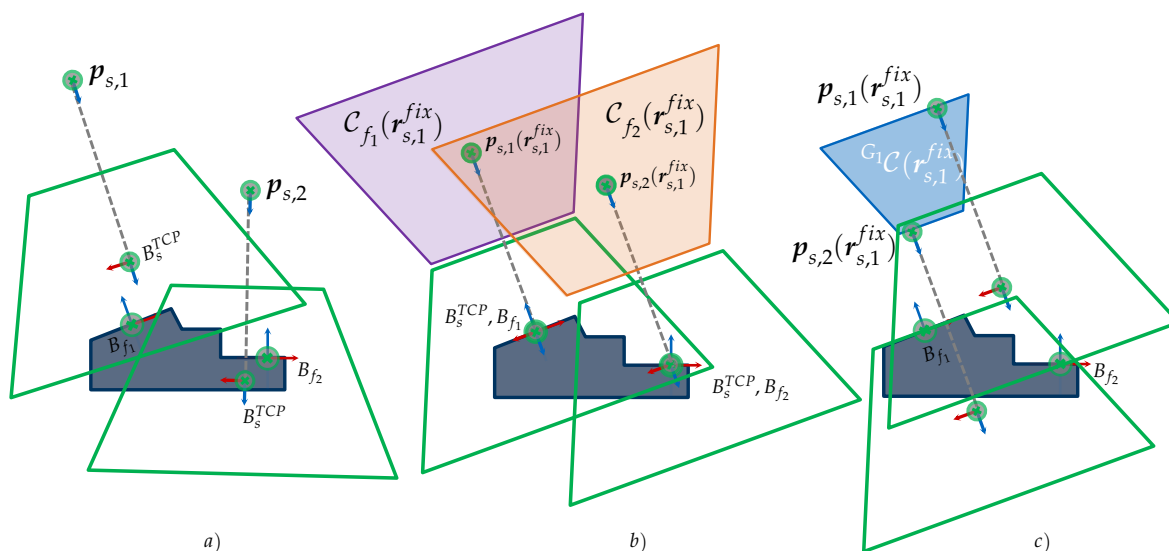


Figure 7. Overview for characterizing ${}^G\mathcal{C}$ -spaces. (a) Initial situation: Two features f_1 and f_2 with two corresponding sensor poses $p_{s,1}$ and $p_{s,2}$ to capture them. (b) Step 1 for ${}^G\mathcal{C}$ -space characterization: Select a sensor orientation $r_{s,1}^{fix}$ for capturing both features and estimate their corresponding \mathcal{C} -spaces \mathcal{C}_{f_1} and \mathcal{C}_{f_2} . (c) Step 2 for ${}^G\mathcal{C}$ -space characterization and verification: Intersect both \mathcal{C} -spaces to characterize the resulting ${}^G\mathcal{C}$ -space: ${}^{G_1}\mathcal{C} = \mathcal{C}_{f_1} \cap \mathcal{C}_{f_2}$. Any sensor pose within $\forall p_s \in {}^{G_1}\mathcal{C}$ is valid for capturing both features.

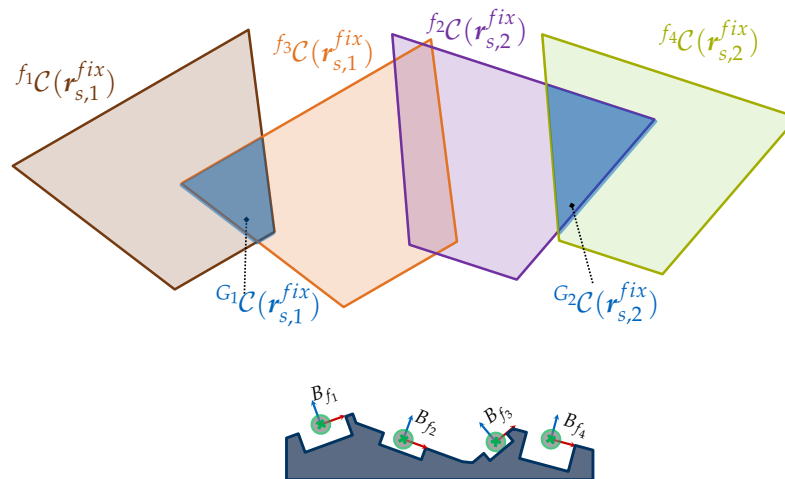


Figure 8. Simplified representation of two ${}^G\mathcal{C}$ -spaces ${}^{G_1}\mathcal{C}(r_{s,1}^{fix})$ and ${}^{G_2}\mathcal{C}(r_{s,2}^{fix})$ to acquire the feature clusters $\{f_1, f_3\} \in G_1$ and $\{f_2, f_4\} \in G_2$.

Verification

To verify the characterization of ${}^G\mathcal{C}$ -spaces for acquiring multiple features, an object comprising four features $\{f_1, f_2, f_3, f_4\} \in F$ was designed (see object in the top right of Figure 3). The features are initially grouped into two clusters: $\{f_1, f_2\} \in G_1$ and $\{f_3, f_4\} \in G_2$. The dimensions and frames of all features are given in Table A3. The features are acquired regarding the following orientation in the object's coordinate system for the first cluster ${}^o r_{s,1}$ and for the second cluster ${}^o r_{s,2}$:

$$\begin{aligned} {}^o r_{s,1}(\alpha_s^z = \beta_s^y = 0^\circ, \gamma_s^x = -150.0^\circ) \\ {}^o r_{s,2}(\alpha_s^z = 90^\circ, \gamma_s^x = -160^\circ, \beta_s^y = 0^\circ). \end{aligned}$$

In the first step, the individual \mathcal{C} -spaces for each feature were computed according to Algorithm 1 considering the imaging parameters of the sensor s^1 (see Table A4) and the selected sensor orientations. Moreover, the feature geometry was integrated as a further viewpoint constraint to ensure the acquisition of the entire geometry (see [4]). In the second step, the resulting ${}^G\mathcal{C}$ -spaces ${}^{G_1}\mathcal{C}$ and ${}^{G_2}\mathcal{C}$ were synthesized by intersecting the corresponding \mathcal{C} -spaces; details on the implementation are given in Section 5.1.4.

Figure 9 shows the described scene and visualizes the characterized manifolds of the individual \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces. To verify the validity of the approach outlined within this section, two extreme sensor poses at each ${}^G\mathcal{C}$ -space, $\{p_{s,1}, p_{s,2}\} \in {}^{G_1}\mathcal{C}$ and $\{p_{s,3}, p_{s,4}\} \in {}^{G_2}\mathcal{C}$, were chosen and rendered, as well as the corresponding depth images and range images at each sensor pose. The results visualized in Figure 10 demonstrate that all features belonging to the same cluster can be simultaneously acquired at the exemplary extreme viewpoints within their corresponding ${}^G\mathcal{C}$ -spaces. The academic example comprising the object's surface model, \mathcal{C} -space, ${}^G\mathcal{C}$ -spaces, and depth images can be found in the digital appendix of this study.

4.1.3. Summary

A \mathcal{C} -space represents an analytical, geometric solution offering infinite valid sensor positions to acquire a single feature satisfying a defined set of viewpoint constraints. Moreover, the experiments demonstrated that the intersection of individual \mathcal{C} -spaces characterizes a topological space, the ${}^G\mathcal{C}$ -space, which provides infinite solutions to acquire a group of features and satisfies all constraints of their \mathcal{C} -spaces simultaneously. Due to these characteristics, the present study regards ${}^G\mathcal{C}$ -spaces as the backbone element for tackling the VPP.

However, ${}^G\mathcal{C}$ -spaces have some limitations, e.g., it must be a priori known which individual \mathcal{C} -spaces can be clustered together, their validity is limited to a fixed sensor

orientation, and their characterization relies on CSG Boolean operations, which are generally considered an expensive computational technique. Therefore, solving the VPP based on ${}^G\mathcal{C}$ -spaces requires a well-thought-out strategy to address these challenges. The formulation of such a strategy is given in Section 4.2.

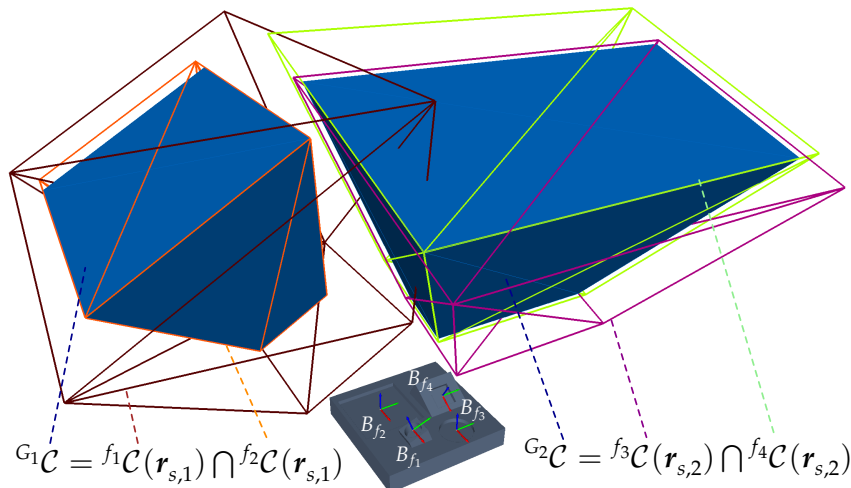


Figure 9. Manifolds of the ${}^G\mathcal{C}$ -spaces ${}^{G_1}\mathcal{C}$ and ${}^{G_2}\mathcal{C}$ in $SE(3)$ for two feature clusters $\{f_1, f_2\} \in G_1$ and $\{f_3, f_4\} \in G_2$ being characterized by the intersection of the corresponding \mathcal{C} -spaces $f_1\mathcal{C}, f_2\mathcal{C}, f_3\mathcal{C}$ and $f_4\mathcal{C}$.

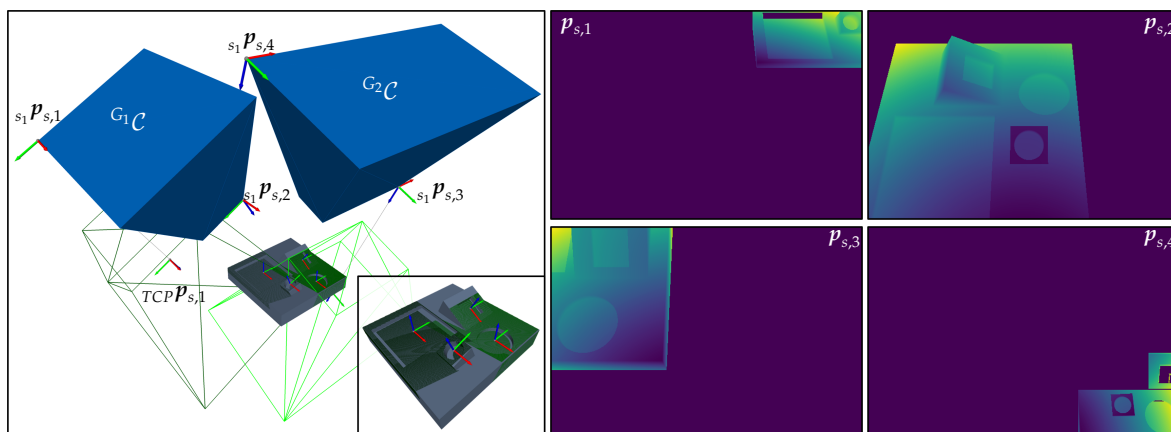


Figure 10. Verification of ${}^G\mathcal{C}$ -spaces considering two sensor poses $\{p_{s,1}, p_{s,2}\} \in {}^{G_1}\mathcal{C}$ and $\{p_{s,3}, p_{s,4}\} \in {}^{G_2}\mathcal{C}$ at the vertices of each manifold. Rendered scene and range images of $p_{s,1}$ and $p_{s,3}$ of (left image) and depth images of all sensor poses (right images).

4.2. Viewpoint Planning Strategy

The present publication poses the VPP in the framework of decision formulation problems and suggests the use of ${}^G\mathcal{C}$ -spaces as the key element for its solution. This section outlines a holistic viewpoint plan strategy based on ${}^G\mathcal{C}$ -spaces to solve generic vision tasks that are confronted with the VPP. Figure 11 provides a simplified overview of the four subordinated modules of the strategy, i.e., the search for potential feature clusters, the optimization of sensor orientation, the characterization of ${}^G\mathcal{C}$ -spaces, and the final selection of sensor poses.

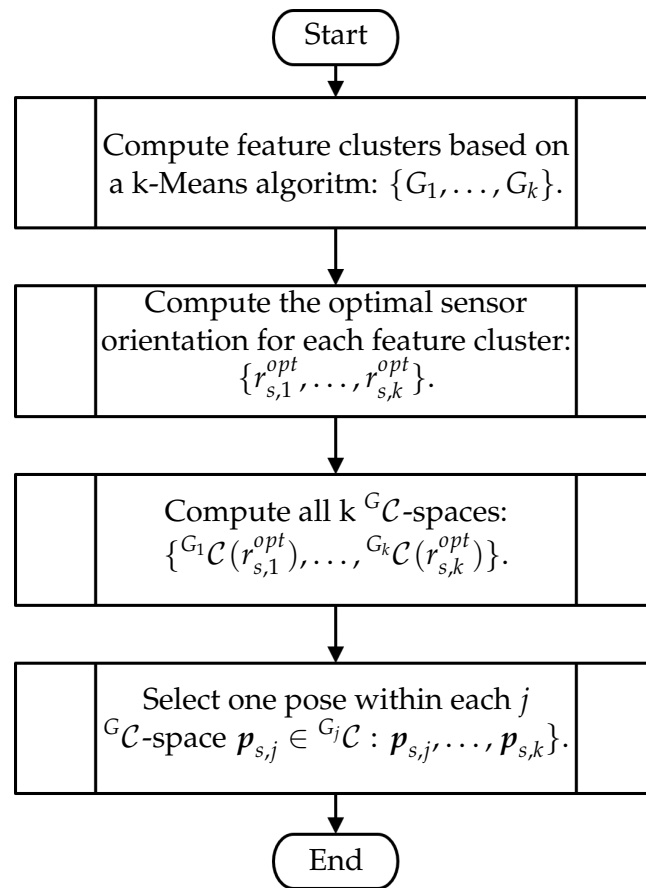


Figure 11. Overview of the viewpoint planning strategy main modules.

4.2.1. Feature Clusters

Addressing the SCP as a decision problem requires identifying feature clusters in the first step, i.e., groups of features that can potentially be acquired together based on their spatial vicinity and orientation similarity. Identifying such feature clusters can be efficiently performed using a clustering algorithm such as a centroid-based k-Means algorithm. This subsection presents a practical and efficient clustering strategy for identifying potential feature clusters. Figure 12 depicts a flow chart of the proposed algorithm.

Data Preparation

First, the strategy proposed within this subsection regards a data-enhancing preprocessing step for combining feature positions and their normal vectors to simplify the search for more effective clusters. In addition to the features' position, we assume that features having similar normal directions are more likely to be clustered together. Hence, a new observation variable is introduced for each feature f_m , denoted as $\mathbf{t}_{f_m}^* \in T^*$, $\mathbf{t}_{f_m}^* \in \mathbb{R}^3$, $m = 1, \dots, n$, that comprises information about the feature position and orientation. The observation variable is obtained by shifting the features' frames along their normal (z-axis) considering a defined distance e^* . This trick has the advantage of spatially separating features based on their orientation without increasing the problem's complexity, see Figure 13. The observation variable $\mathbf{t}_{f_m}^*$ is formally defined as the translation component (*trans*) of the following transformation:

$$\mathbf{t}_f^* = \text{trans} \left({}^0T_f({}^0\mathbf{t}_f, {}^0\mathbf{r}_f) \cdot \begin{pmatrix} I & 0 \\ 0 & 1 \end{pmatrix} \right),$$

where I denotes a 3×3 square identity matrix.

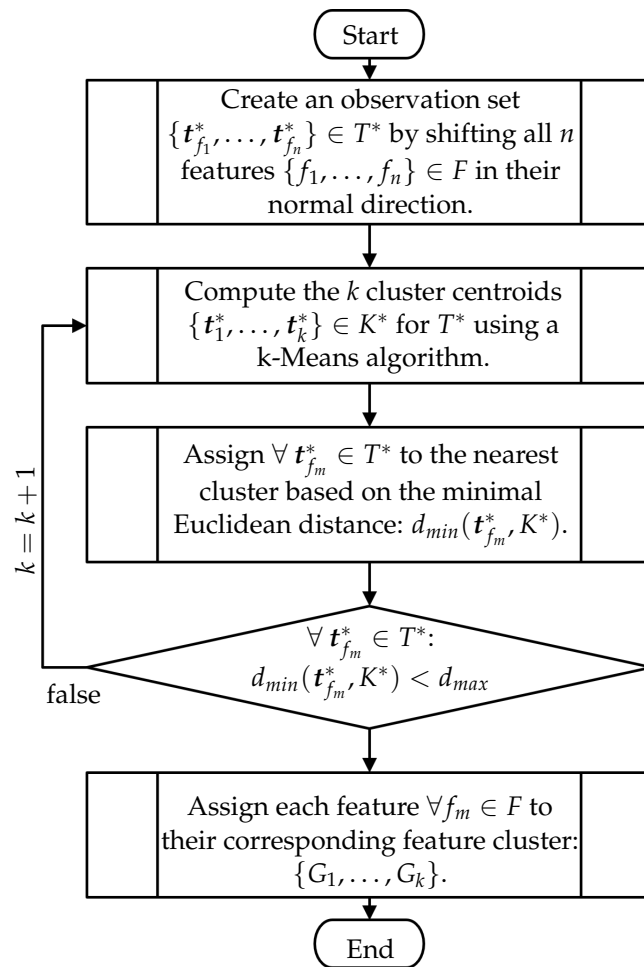


Figure 12. Algorithm for computing feature clusters based on a k-Means algorithm.

Figure 13 depicts a simplified representation of the enhanced variables. The distance e^* can be arbitrarily chosen. However, e^* should not be larger than the sensor's far plane $e^* \leq h_s^{far}$. The empirical experiments in Section 5.1 showed that the sensor's middle working distance ($e^* = d_s$) is a good compromise for generating an adequate number of clusters.

Iteratively Clustering

Finally, using the set of observations $\mathbf{t}_{f_m}^* \in T^*$, all k cluster centroids $\mathbf{t}_j^* \in K^*$, $j = 1, \dots, k$ can be computed using a centroid-based k-Means algorithm. The k-Means aims to choose centroids that minimize the Euclidean distances between a selected cluster centroid and the set of observations T^* :

$$\sum_{j=1}^k \min(\|\mathbf{t}_{f_m}^* - \mathbf{t}_j^*\|^2)$$

Recalling the approach proposed in Section 3.2.2 for iteratively finding a viable number of clusters, a break condition must be first defined to determine when it is necessary to increase the number of feature clusters. Such a condition can be applied for each observation $\forall \mathbf{t}_{f_m}^* \in T^*$ considering the minimal Euclidean distance $d_{min}(\mathbf{t}_{f_m}^*, K^*)$ that an observation has to the closest cluster centroid from K^* . This distance then has to be smaller than a threshold d_{max} :

$$d_{min}(\mathbf{t}_{f_m}^*, K^*) < d_{max}. \quad (8)$$

Taking into account the imaging capabilities of the sensor, d_{max} can be defined using the shortest length of the frustum, e.g., half of the sensor width at the nearest view plane (see Figure A1).

If any element of T^* does not fulfill condition (8), then k is increased by 1. If a minimal number of clusters can be estimated beforehand, this can be given as an initial value to optimize the search process; otherwise, $k = 1$ should be assumed. This process is repeated until Equation (8) is satisfied by all observations $\forall t_{f_m}^* \in T^*$. Figure 13 illustrates the two resulting clusters for a simple scenario in \mathbb{R}^2 .

4.2.2. Sensor Orientation Optimization

Recalling the requirements to compute ${}^G\mathcal{C}$ -spaces, the present strategy considers a two-step approach for selecting an optimized sensor orientation for each feature cluster.

Formulation

The sensor orientation in $SO(3)$ can be fully represented by a normal vector and an optimized swing angle:

$$r_s^{opt} = (n_s^{opt}, \alpha_s^{z,opt}).$$

While the normal vector represents the incidence angle, i.e., the rotation around the x -axis and y -axis of the sensor, the swing angle provides the rotation around the z -axis.

Incidence Angle

The optimized incidence angle for a cluster can be calculated using the arithmetic mean of the normal vectors of all features of a cluster $\forall f \in G$, as follows:

$$n_s^{G,opt} = \left\| \sum_{f \in G} n_f \right\|, n_s^{G,opt} < f \varphi_s^{max}. \quad (9)$$

In the case that the incidence angle between the optimized orientation and a feature is greater than the maximum $f \varphi_s^{max}$ (see Equation (2)), then the feature must be assigned to a new cluster.

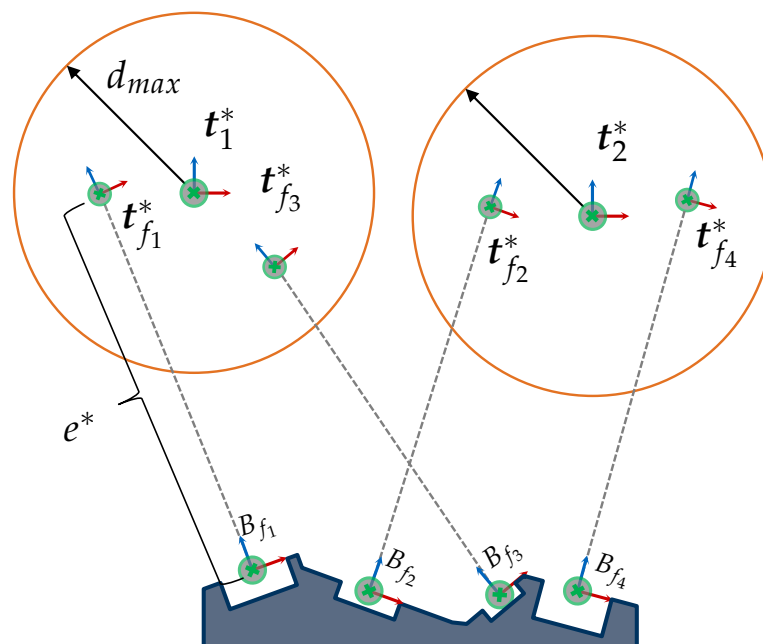


Figure 13. Exemplary characterization of two feature clusters (centroids: t_1^*, t_2^*) considering the observation set $t_{f_1}^*, \dots, t_{f_5}^*$ using a k-Means algorithm.

Swing Angle

Optimizing the rotation angle around the optical axis can be particularly advantageous when the sensor frustum is asymmetrical. Hence, this subsection suggests finding the optimal swing angle using an oriented minimum bounding box (OBB) algorithm. The direction of the longest side of such a bounding box corresponds to the optimized sensor around its z-axis.

First, the 2D projection of all features $t_f^{2D} \in \mathbb{R}^2$ for $\forall f \in G$ is calculated using a perspective projection matrix denoted as $M^{2D}(n_s^{G,opt})$ and the direction vector $n_s^{G,opt}$:

$$t_f^{2D} = M^{2D}(n_s^{G,opt}) \cdot t_f.$$

Next, using all projected points, an OBB algorithm [36] can be applied to compute the four corner points of the bounding box

$$\{g_0^{bb}, \dots, g_3^{bb}\} \leftarrow OBB(t_{f_1}^{2D}, \dots, t_{f_n}^{2D}).$$

The swing angle corresponds to the principal axis vector e_{OBB}^x of the bounding box along the longest side of the bounding box, e.g., $\alpha_s^{z,opt}$ is given in the object's coordinate system (o) as follows:

$${}_o\alpha_s^{z,opt} = \arccos\left(\frac{e_{OBB}^x \cdot {}_o e^x}{|e_{OBB}^x| \cdot |{}_o e^x|}\right),$$

with $e_{OBB}^x = g_0^{bb} - g_1^{bb}$. Figure 14 presents an illustrative representation of the identification of bounding boxes and estimation of their orientation used to determine the optimal swing angle for two feature clusters.

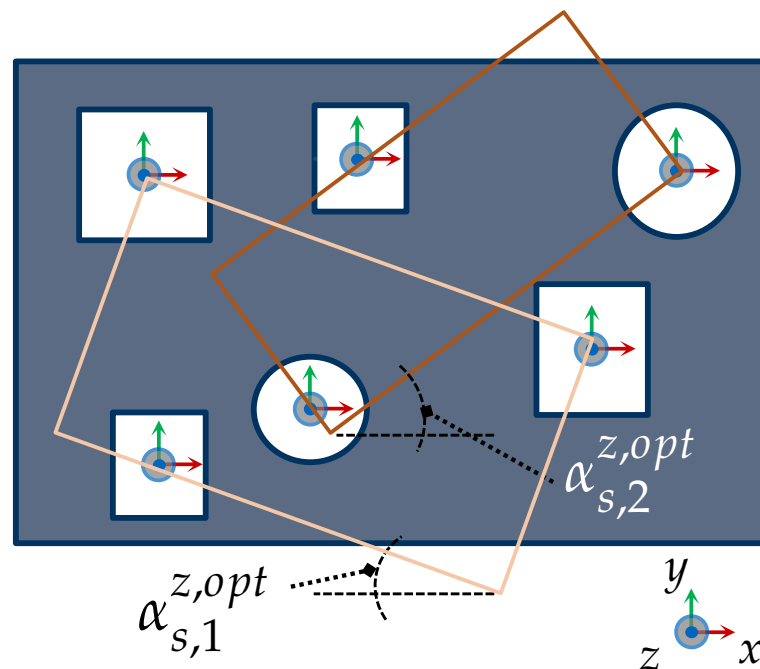


Figure 14. Exemplary optimization of the swing angles $\alpha_{s,1}^{z,opt}$ and $\alpha_{s,2}^{z,opt}$ for two feature clusters using an OBB algorithm.

Orientation of further imaging devices

In the context of range sensors consisting of more than one imaging device (see Section 2.1), note that if the imaging devices are differently oriented to each other, the orientation for

other imaging devices results from the rigid transformation between the devices, e.g., for a second imaging device:

$${}_{s_2}\mathbf{r}_s^{opt} = {}_{s_1}\mathbf{R}_s \cdot {}_{s_1}\mathbf{r}_s^{opt}.$$

The \mathcal{C} -spaces for the second device must be characterized using the sensor orientation ${}_{s_2}\mathbf{r}_s^{opt}$ (see Algorithm 1).

4.2.3. Computation of ${}^G\mathcal{C}$ -spaces

Integration Strategy of \mathcal{C} -spaces

Having identified the number of necessary feature clusters and corresponding optimized sensor orientations, the necessary ${}^G\mathcal{C}$ -spaces can be finally computed. Although the integration of ${}^G\mathcal{C}$ -spaces and \mathcal{C} -spaces can generally be considered commutative, an adequate computation order of the individual steps can considerably improve the overall computational efficiency of the viewpoint planning strategy.

This study extends the strategy from [4] to consider a more efficient characterization of ${}^G\mathcal{C}$ -spaces. Algorithm 2 provides a formal description of these steps. The resulting \mathcal{C} -space and ${}^G\mathcal{C}$ -spaces manifolds of some of these steps are visualized in Figure 15, considering the academic example introduced in Section 4.1.2. Moreover, it is assumed there exists a kinematic error of 50 mm in all directions and that two cubes partly occlude the visibility of the features. The imaging parameters of the second imaging device s_2 are given in Table A4. We refer to our previous publication, which covered the characterization of kinematic errors, occlusion-free spaces, and integration of viewpoint constraints from multiple imaging devices. The manifolds of all \mathcal{C} -space and ${}^G\mathcal{C}$ -spaces are found in the digital appendix of this publication.

Moreover, the present strategy considers the decimation of the resulting \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces manifolds by merging adjacent vertices after each CSG Boolean operation to reduce the overall computational effort. Note, that this simplification step may affect the validity of the ${}^G\mathcal{C}$ -spaces. Therefore, the threshold value for merging adjacent vertices should be carefully chosen. Some particularities of the overall strategy are addressed in the following subsections.

Occlusion-Free ${}^G\mathcal{C}$ -spaces

According to [4], \mathcal{C} -spaces that integrate occlusion constraints can be characterized using ray-casting and CSG Boolean operations. Therefore, their computation can be regarded as one of the most expensive steps within a viewpoint planning strategy. To optimize the computation of such spaces, the same study showed that the computational efficiency could be considerably improved by limiting the validity of the occlusion-free \mathcal{C} -space using the topological space limited by other viewpoint constraints, e.g., imaging parameters, feature geometry, kinematic tolerances.

Based on this insight and taking into account that this step must be repeated multiple times for all \mathcal{C} -spaces, the present publication follows a similar approach by exploiting the concept of ${}^G\mathcal{C}$ -spaces. Hence, in the first step, the required ${}^G\mathcal{C}$ -spaces are computed neglecting any occlusion constraints; let this space be denoted as ${}^{G_j}\mathcal{C}^{s_i,*}$. In the second step, the individual occlusion-free \mathcal{C} -space for each feature can then be more efficiently computed by limiting the occlusion-free visibility to the space spanned by ${}^{G_j}\mathcal{C}^{s_i,*}$ (see Step 6 of Algorithm 2). Note that the same result could have been more inefficiently achieved by first computing the occlusion-free \mathcal{C} -space for each feature and then by intersecting all of them. Figure 15 visualizes the resulting occlusion-free ${}^G\mathcal{C}$ -spaces following this approach.

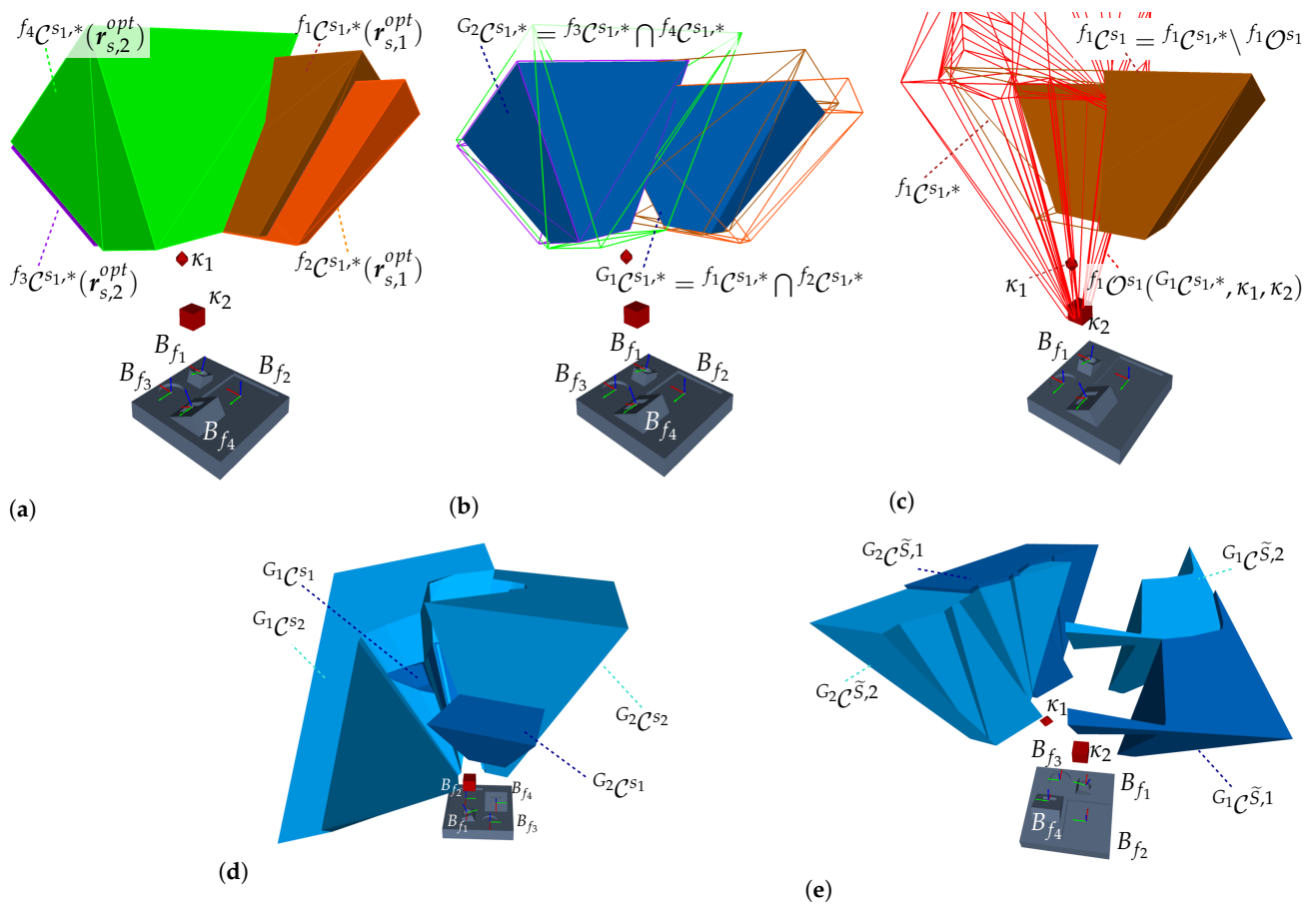


Figure 15. (a–e): Exemplary visualization of some steps of Algorithm 2 to characterize the ${}^G\mathcal{C}$ -space based on three individual \mathcal{C} -spaces corresponding to three features. (a) Step 4: Compute the \mathcal{C} -spaces for s_1 of all features considering the optimized sensor orientations $\mathbf{r}_{s_1}^{opt}$ for $\{f_1, f_2\} \in G_1$ and $\mathbf{r}_{s_2}^{opt}$ for $\{f_3, f_4\} \in G_2$. (b) Step 5: Compute the ${}^G\mathcal{C}$ -spaces ${}^{G_1}\mathcal{C}$ and ${}^{G_2}\mathcal{C}$ by intersecting the corresponding \mathcal{C} -spaces. (c) Step 6: The occlusion-free \mathcal{C} -space for f_1 is characterized by computing the Boolean difference between the occluding space $f_1\mathcal{O}^{s_1}$ (red wireframe manifold) and the \mathcal{C} -space $f_1\mathcal{C}^{s_1,*}$ from Step 4. (d) Step 7: Compute the occlusion-free ${}^G\mathcal{C}$ -spaces for the first imaging device ${}^{G_1}\mathcal{C}^{s_1}$ and ${}^{G_2}\mathcal{C}^{s_1}$. The ${}^G\mathcal{C}$ -spaces of the second imaging device s_2 (${}^{G_1}\mathcal{C}^{s_2}$, ${}^{G_2}\mathcal{C}^{s_2}$) are computed analogously following Steps 3–7. (e) Step 9: Compute the ${}^G\mathcal{C}$ -spaces for s_1 ${}^{G_1}\mathcal{C}^{\tilde{s}_1,1}$ and ${}^{G_2}\mathcal{C}^{\tilde{s}_1,1}$ to consider the viewpoint constraints of the second imaging device.

Strategy against invalid ${}^G\mathcal{C}$ -spaces

Finally, it should be noted that the feature clusters and optimized sensor orientations should be considered as an initial and plausible solution to capture all features. However, the validity of a ${}^G\mathcal{C}$ -space can be first assessed until the individual \mathcal{C} -spaces are intersected and all viewpoint constraints are integrated. Therefore, the existence of the ${}^G\mathcal{C}$ -space must be continuously assessed after intersecting two consecutive \mathcal{C} -spaces. If the intersection yields an empty manifold or the \mathcal{C} -space manifold volume is below a defined threshold, a strategy to overcome this issue must be considered. For instance, in the simplest scenario, the last intersecting \mathcal{C} -space can be assigned to a new cluster, and the rest of the process can be continued with the rest of the features.

The intersection of two consecutive \mathcal{C} -spaces yielding an empty manifold can be caused by the inconvenient combination of diverse viewpoint constraints. Hence, an efficient strategy to address this issue requires an individual analysis of the particular viewpoint constraint. The comprehensive analysis and formulation of such strategies fall outside the scope of this paper.

Algorithm 2 Characterization of ${}^G\mathcal{C}$ -spaces based on \mathcal{C} -spaces

1. Select the j feature cluster $G_j \in F, j = 1, \dots, k$.
2. Select the t imaging device of the sensor $s_t \in \tilde{S}, t = 1, \dots, u$.
3. Consider the sensor orientation ${}_{s_t}r_{s_j}^{opt}$ for the reference imaging device s_t .
4. Compute all n_j \mathcal{C} -spaces of all features from the cluster $\forall f_{m_j} \in G_j, m_j = 1, \dots, n_j$ considering the sensor orientation of the first device ${}_{s_t}r_{s_j}^{opt}$ and a subset of viewpoint constraints $\tilde{C}^* \subseteq \tilde{C}$ that do not require any Boolean operations.

$${}^{f_{m_j}}\mathcal{C}^{s_t,*} := \mathcal{C}(f_{m_j}, \tilde{C}^*, {}_{s_t}r_{s_j}^{opt}).$$

5. Compute the j ${}^G\mathcal{C}$ -space by intersecting all n_j \mathcal{C} -spaces iteratively

$$G_j\mathcal{C}^{s_t,*} = \bigcap^{n_j} {}^{f_{m_j}}\mathcal{C}^{s_t,*}.$$

6. Compute all n_j occlusion spaces ${}^{f_{m_j}}\mathcal{O}^{s_t}$ for each feature $\forall f_{m_j} \in G_j$ based on the previously corresponding ${}^G\mathcal{C}$ -space and the surface models $\kappa \in K$ of all occluding bodies. The occlusion-free \mathcal{C} -space corresponds to the following Boolean difference:

$${}^{f_{m_j}}\mathcal{C}^{s_t} = {}^{f_{m_j}}\mathcal{C}^{s_t,*} \setminus {}^{f_{m_j}}\mathcal{O}^{s_t}(G_j\mathcal{C}^{s_t,*}, K).$$

7. Recompute the j ${}^G\mathcal{C}$ -space iteratively by intersecting all n_j occlusion-free \mathcal{C} -spaces:

$$G_j\mathcal{C}^{s_t} = \bigcap^{n_j} {}^{f_{m_j}}\mathcal{C}^{s_t}.$$

8. Repeat Steps 3–7 for all imaging devices $\forall s_t \in \tilde{S}$.
9. Compute the ${}^G\mathcal{C}$ -space that integrates the viewpoint constraints of all imaging devices, e.g., for s_1 , as follows:

$$G_j\mathcal{C}^{\tilde{S},1} = G_j\mathcal{C}^{s_1} \bigcap_{s_t \in \tilde{S}} G_j\mathcal{C}^{s_1,s_t}(G_j\mathcal{C}^{s_1}, G_j\mathcal{C}^{s_t}).$$

10. Steps 2–11 are repeated for each cluster $\forall G_j \in F, j = 1, \dots, k$.

4.2.4. Sensor Pose Selection

Having computed all required ${}^G\mathcal{C}$ -spaces, the last step considers selecting a sensor pose within each ${}^G\mathcal{C}$ -space. Since the present framework does not explicitly consider the existence of a global optimum within a ${}^G\mathcal{C}$ -space, any sensor pose within it fulfills all defined constraints and is equally valid to any other; see the depth images in Figure 16.

In the simplest case, any vertex of the ${}^G\mathcal{C}$ -space manifold could be used as a valid sensor pose. Alternatively, the geometrical center of the manifold can be considered a local optimum of the ${}^G\mathcal{C}$ -space for some cases. However, it cannot be guaranteed that the geometrical center of the ${}^G\mathcal{C}$ -space manifold lies within it. For this reason, an explicit evaluation is always required. Such scenarios are not rare when considering occlusion constraints.

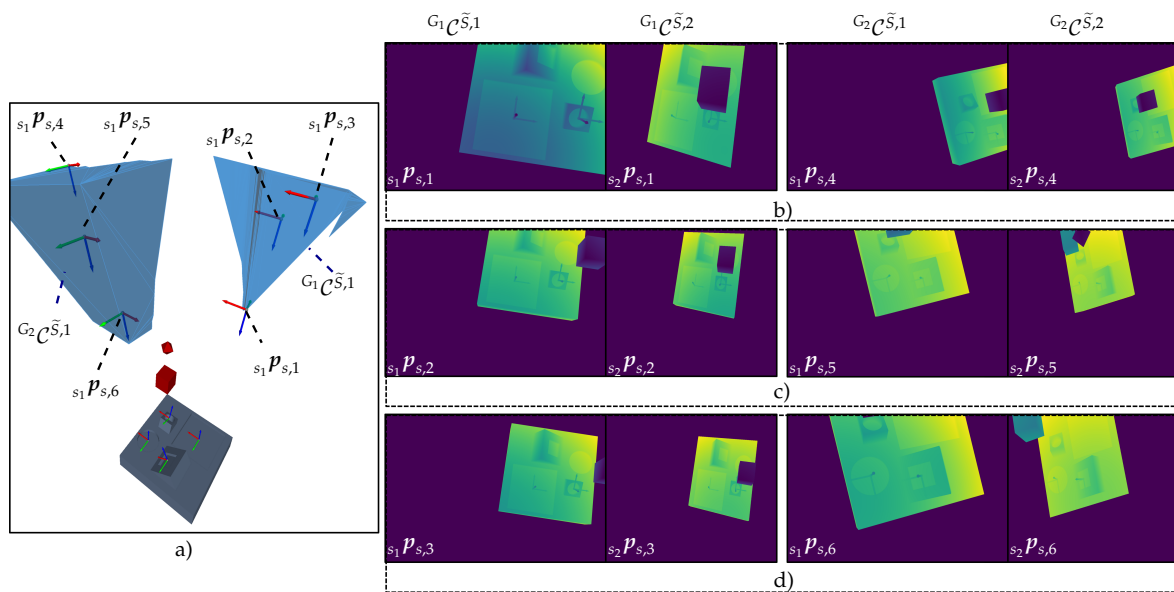


Figure 16. (a) Verification scene comprising two G_C -spaces for acquiring two feature clusters G_1 and G_2 and three potential sensor poses within each G_C -space. The images on the right depict the rendered depth images of the imaging devices s_1 and s_2 corresponding to the three different sensor poses within the G_C -spaces at (b) a vertex, (c) the geometric center, and (d) one random point.

4.3. Summary

This section outlined a generic viewpoint planning strategy to solve the VPP based on the formulation of the SCP as a decision problem and G_C -spaces. First, the strategy uses a k-Means clustering algorithm to identify potential feature clusters. In the second step, a sub-strategy was proposed to consider an optimized sensor orientation in $SO(3)$ for each feature cluster. Having identified potential feature clusters and an adequate sensor orientation, a further generic sub-strategy was introduced to efficiently integrate C -spaces and compute the required G_C -spaces manifolds. Finally, some generic approaches were introduced in the last step to select a valid sensor pose within the characterized G_C -spaces.

5. Results

This section comprehensively analyzes the usability of G_C -spaces and the overall viewpoint planning strategy to solve the VPP in the context of real industrial metrological applications considering two different vision systems. The results confirm that the framework outlined within this publication can be effectively and efficiently used for solving complex vision tasks by providing robust solutions.

5.1. Robot Vision System with Structured Light Sensor

5.1.1. System Description

The framework presented within this publication was utilized for automating the dimensional metrology inspection of a car door comprising up to 670 features using the industrial robot vision system AIBox from ZEISS. The AIBox is an integrated industrial RVS manufactured by ZEISS, equipped with a structured light sensor (ZEISS Comet PRO AE), a six-axis industrial robot (Fanuc M20ia), and a rotary table for mounting an inspection object. The imaging parameters of the imaging sensor and structured light projector are given in Table A4. Figure 17 provides an overview of the AIBox and its core elements.

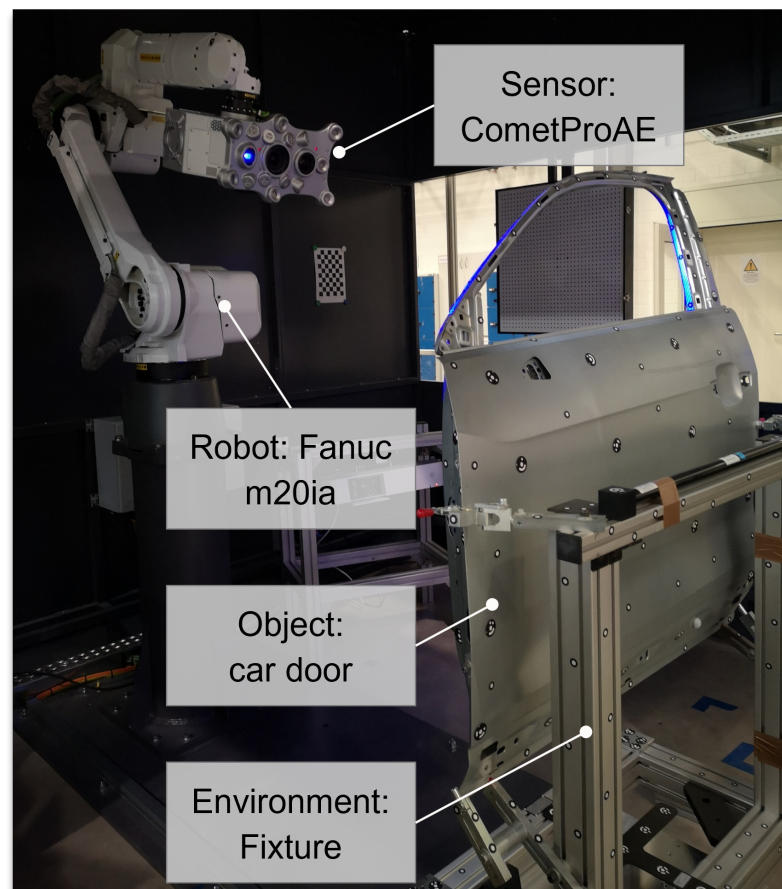


Figure 17. Overview of the core components of the *AIBox*.

5.1.2. Vision Task Description

To thoroughly evaluate the strategy and algorithms presented in this publication, 15 different inspection tasks were considered. The tasks comprise combinations of different features from both door sides and viewpoint constraints. The left columns of Table A5 provide an overview of the vision tasks.

Door Side: To evaluate the usability of the present framework in an industrial context, a car door was used as the probing object. Due to their topological complexity, feature density, and variability, car doors are well-known benchmark workpieces for evaluating metrology tasks and their automation.

Number and Type of Features: The scalability was evaluated using inspection tasks with different numbers and types (points and circles) of features.

Viewpoint Constraints: To analyze the efficacy and efficiency of the overall strategy, vision tasks with different viewpoint constraints were designed. All vision tasks regarded at least the most elementary viewpoint constraints c_1 – c_3 (i.e., the imaging characteristics of the sensor, feature geometry, and the consideration of kinematic errors). Moreover, for some vision tasks, a fourth viewpoint constraint c_4 was considered to ensure the satisfiability of the viewpoint constraints of the projector. Finally, for the vision tasks that included the viewpoint constraint c_5 , it was assumed that all features must have an occlusion-free visibility to the sensor and projector. Table 1 provides an overview of the considered viewpoint constraints.

5.1.3. Evaluation Metrics

Two metrics were designed to properly evaluate the computed viewpoint plans and to compare and benchmark the obtained results.

Measurability Index

To properly quantify the validity of the computed ${}^G\mathcal{C}$ -spaces and selected sensor poses for each viewpoint plan, we introduced a measurability index $C(P)$. This metric represents the ratio between the total number of successfully acquired features and the total number of features per view plan n . The measurability index is given as follows:

$$M = \frac{\sum g(f_m)}{n}.$$

To assess the measurability of a single feature, we considered a qualitative M^{qual} and a quantitative M^{quant} metric.

M^{qual} : The qualitative function assesses the following two conditions.

1. A feature f_m , including its entire geometry, must lie within the calculated frustum space $\mathcal{I}_s(\mathbf{p}_{s,j})$ of the corresponding sensor pose $\mathbf{p}_{s,j} \in {}^G_j\mathcal{C}$
2. Both sensor and projector have free sight to the feature.

If both conditions are fulfilled, the feature is considered to be successfully acquired.

M^{quant} : The validity of each feature was further qualified based on the resulting 3D measurement, i.e., the point cloud. This metric counts the number of acquired points within a defined search radius around a feature. If there exist more points than a specified threshold, the feature can be considered to be valid.

It needs to be noted that the proper evaluation of this condition requires that the measurements are perfectly aligned in the same coordinate system as the features and that the successful acquisition of surface points is guaranteed if all regarded constraints c_1 – c_5 are satisfied. Since our work neglects nonspatial constraints that may affect the quality of the measurement (e.g., exposure times or lighting conditions), the validity of the view plans was mainly assessed based on simulated measurements. The simulated measurements are generated by the proprietary software colin3D (Version 3.12) from ZEISS, which considers occlusion and maximal incidence angle constraints. Moreover, the measurements are perfectly aligned to the car door surface model.

Computational Efficiency

To provide a comprehensive analysis of the computational efficiency of the viewpoint planning strategy, the computation times of the most relevant steps were estimated:

$t_{k,opt}$: computation time for computing the necessary k feature clusters and corresponding optimized sensor orientations,

$t_{\mathcal{C}}$: computation time to characterize all individual \mathcal{C} -spaces (one for each feature) considering the regarded viewpoint constraints,

$t_{G\mathcal{C}}$: computation time to characterize all k ${}^G\mathcal{C}$ -spaces,

t_{total} : total computation time of the vision task, corresponds to the sum of the times mentioned above.

The evaluation tests were developed using the trimesh library [37] and open3D library [38]. All operations were performed on a portable workstation Lenovo W530 running Ubuntu 20.04 with the following specifications: Processor Intel Core i7-3740QM @2.70 GHz, Nvidia Quadro K1000 M, and 24 GB Ram.

Table 1. Overview of viewpoint constraints

Viewpoint Constraints	Description
c_1	The imaging parameters of the structured light sensor comprising the camera and projector must be considered (see Table A4).
c_2	The feature dimensions must be regarded so the whole feature geometry is acquired within the same measurement.
c_3	Due to modeling uncertainties, a kinematic error of 50 mm in all Cartesian directions is assumed.
c_4	The imaging parameters of the structured-light projector must be considered.
c_5	The door fixture may occlude some features. However, a self-occlusion with the car door is neglected.

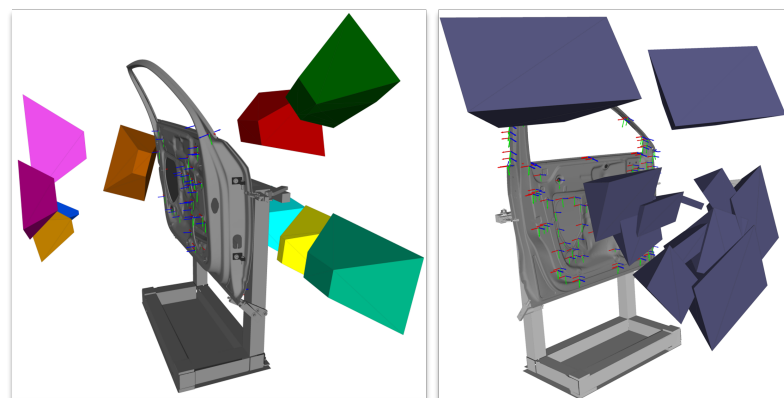
5.1.4. Implementation

The viewpoint planning framework was developed based on the Robot Operating System (ROS) [39], which was primarily used for the frame transformation operations. 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 [40,41].

Moreover, the clustering was performed using the k-Means algorithm of the Sci-Kit library, using the Lloyd implementation [42]. Due to its high efficiency for performing CSG Boolean operations, the PyMesh library from Zhou et al. [43] was utilized for characterizing the \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces. The ray-casting operations for computing the occlusion space were performed using the trimesh library [37].

5.1.5. Results

In the first step, the required ${}^G\mathcal{C}$ -spaces for each vision task of Table A5 were computed using the viewpoint strategy proposed in Section 4.2. An overview of the ${}^G\mathcal{C}$ -spaces for the fourth and seventh inspection tasks is displayed in Figure 18. For the feature clustering, we considered a maximal Euclidean distance of $e_{max} = 200$ mm (see Equation (8)), which approximately represents the frustum's half width-length at its middle plane (see Table A4). Furthermore, the maximal incidence angle was defined as $n_s^{max} = 30^\circ$. The sensor poses used for the evaluation represented the geometric center of the corresponding ${}^G\mathcal{C}$ -space manifolds. The qualitative and quantitative measurability indexes of all viewpoint plans and an overview of the computation times of all vision tasks are shown in the right-handed columns of Table A5. A detailed discussion of the evaluation metrics is discussed as follows.

**Figure 18.** ${}^G\mathcal{C}$ -spaces of inspection tasks 4 (left) and 7 (right).

Measurability

In the first step, the qualitative evaluation of all viewpoint plans was performed based on the previously introduced metrics. As expected, the qualitative evaluation shows very encouraging results, with an average measurability index above 95% for the viewpoint plans that considered the most constraints for each inspection task. The high efficacy of some selected viewpoint plans could be further assessed using the quantitative evaluation of simulated and real measurements. Furthermore, it could also be demonstrated that for some single vision tasks (3–5 and 13–15), the measurability index was improved by considering occlusion constraints for the sensor and projector.

On the one hand, these encouraging results confirm the validity and applicability of the viewpoint planning strategy using ${}^G\mathcal{C}$ -spaces. In particular, the real measurements for vision tasks 5 and 7 demonstrated that the selected sensor poses were robust enough to compensate for kinematic uncertainties in the real measurements. In contrast, the measurability of some individual features could not be guaranteed in some cases. The failed evaluation of these cases, as well as the discrepancies between the qualitative and quantitative evaluations, can be attributed to two leading causes, which require a further discussion:

- **Occlusion:** All failed qualitative evaluations and the decrease in the measurability score if occlusion constraints were regarded can be attributed to the nonexistence of an occlusion-free space for the computed ${}^G\mathcal{C}$ -spaces with the chosen sensor orientation. The strategy proposed in Section 4.2.3 did not explicitly contemplate such cases. However, this problem could be straightforwardly solved by considering an alternative sensor orientation in the 7th step of Algorithm 2 when the intersection of consecutive \mathcal{C} -spaces yields a non-empty manifold. Formulating such a strategy requires a more comprehensive analysis of the occlusion space, which falls outside the scope of this work.
Furthermore, the failed evaluation of most features lying on the inside of the door was occasioned by occlusion with the door itself, which was initially neglected as an occluding object. However, an empirical analysis of some failed viewpoints showed that a positive evaluation could be achieved by recomputing the ${}^G\mathcal{C}$ -spaces of the affected features considering the car door as an occluding object, as seen in Figure 19.
- **Missing points and misalignment:** The quantitative evaluation of some individual viewpoints showed discrepancies between the simulation and the real measurements. These differences can be easily explained considering the requirements of the quantitative evaluation strategy proposed in Section 5.1.3 based on the acquisition of surface points. Due to the high reflectivity of the car door material and the fact that the optimization of the exposure time was neglected during the experiments, the acquisition of enough surface points in some areas could not be achieved, see Figure 20. On the other hand, a detailed evaluation of some failed viewpoints showed that the measurements could not be aligned correctly in other cases, causing a false-positive evaluation of some features. By manually optimizing the number of exposure times and individual values, more dense and better-aligned measurements could be obtained, mitigating most of the mentioned errors.

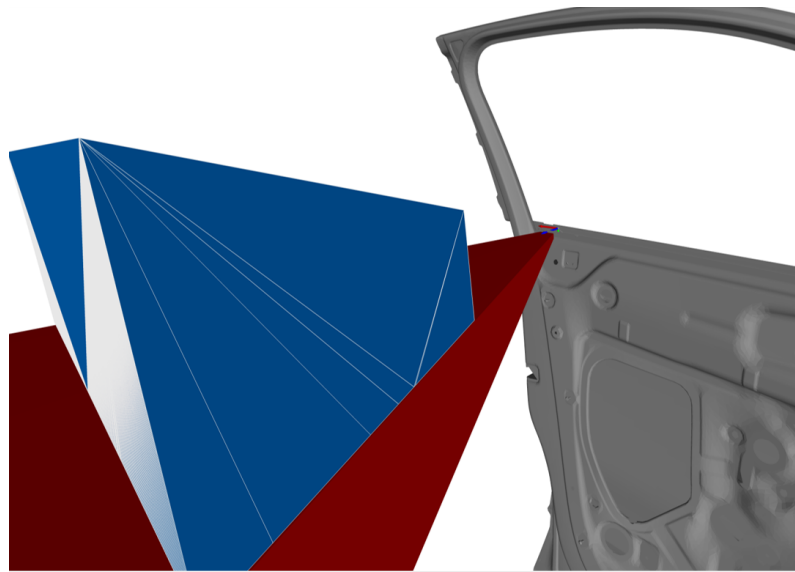


Figure 19. Recomputation of ${}^G\mathcal{C}$ -space regarding the door as an occlusion object, the red manifold represents the occlusion space generated by the door's surface model.

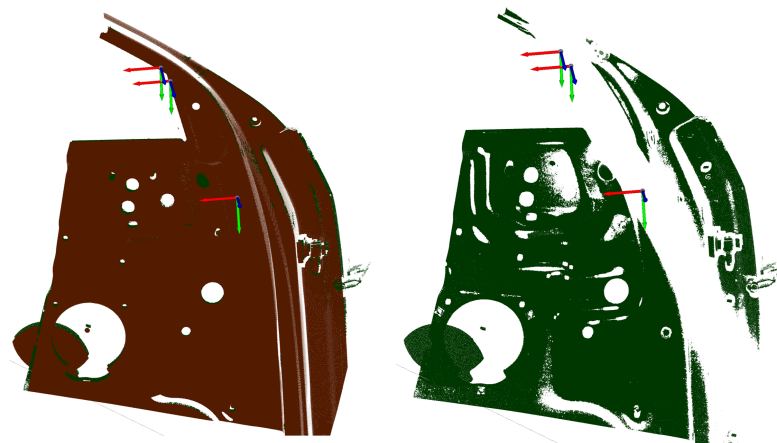


Figure 20. Comparison of the quantitative evaluation of the measurability of three exemplary features using a simulated measurement (left) and a real measurement (right): the quantitative assessment with real measurements fails due to a lack of surface points.

Computational Efficiency

The computational efficiency overview from Table A5 demonstrates that all viewpoint plans neglecting occlusion were computed in linear times. Furthermore, the general time distribution for the vision tasks omitting occlusion constraints shows that, on average, 80% of the total time was required for the characterization of the \mathcal{C} -spaces manifolds, 15% for the ${}^G\mathcal{C}$ -spaces, and only 5% for the clustering and orientation optimization tasks. A more comprehensive analysis of each computed step is discussed below.

t_G : It can be observed that the feature clustering and optimization of the sensor orientation can be regarded as the most efficient step of the strategy and represent, on average, less than 10% of the whole planning process. The experiments show the efficiency of the k-Means algorithm for such tasks, agreeing with the previous findings from [28].

t_C : The vision tasks that only incorporate the fundamental constraints c_1 – c_3 showed a high computational efficiency. These results were to be expected, taking into account that the \mathcal{C} -space characterization of these viewpoint constraints consists mainly of linear operations. This trend can be observed in Figure A2, showing the proportional

increase between the computation time for the required \mathcal{C} -spaces $t_{\mathcal{C}}$ and the total number of features. This behavior can be further observed when the fourth constraint is considered. In this case, each \mathcal{C} -space must be spanned for each imaging device (sensor and projector), increasing the computation by a factor of two. On the other hand, taking into account the occlusion constraint c_5 considerably increased the computational complexity of the task. This behavior is also comprehensible, recalling that the characterization of occlusion-free \mathcal{C} -spaces relies on ray-casting, which is well known to be a computationally expensive process.

Moreover, neglecting occlusion constraints, the average computation time for the characterization of one \mathcal{C} -space was estimated at 60 ms. It needs to be noted that this time estimation includes a non-negligible computational overhead of all required operations, such as frame transformation operations using ROS-Services. In [4] the computation of a single \mathcal{C} -space was estimated, on average, at 4 ms.

$t_{\mathcal{C}}$: The computation of the ${}^G\mathcal{C}$ -spaces using intersecting CSG Boolean operations proved to be highly efficient, requiring 10–15% of the total planning time. The experiments also confirm that the time effort increases with the number of intersecting spaces. However, by applying manifold decimation techniques after each Boolean intersection (cf. Section 4.2.3), the time effort could be considerably reduced. For instance, within the first vision task, the characterization of a ${}^G\mathcal{C}$ -space with six \mathcal{C} -spaces required 0.6 s, while the intersection of a ${}^G\mathcal{C}$ -space with 44 \mathcal{C} -spaces took 2.4 s. Furthermore, the computation times of the ${}^G\mathcal{C}$ -spaces considering occlusion constraints visualized in Figure A2 confirm that the intersection of more complex manifolds was, on average, more time-consuming.

Determinism

Finally, we selected three inspection tasks (4, 6, and 15, marked with * in Table A5) and computed the corresponding viewpoint plan ten times to assess the robustness and determinism of the viewpoint planning strategy. Within these experiments, the heuristic characteristic of the k-Means algorithm can be observed. In particular, within vision task 6, which considers a higher number of features, the number of computed clusters differed between computations with a standard deviation of $\sigma = 0.89$ clusters. However, the necessary ${}^G\mathcal{C}$ -spaces could always be computed, and the computation times showed an acceptable standard deviation.

5.2. CMM with Laser Line Scanner

5.2.1. System Description

To assess the transferability of the viewpoint planning strategy and the usability of ${}^G\mathcal{C}$ -spaces with an alternative kinematic vision system commonly used in industrial metrology applications, we considered a coordinate measuring machine (CMM) model LK Altera 15.7.6 and a laser line scanner (LLS) LC60Dx from Nikon Metrology. A comprehensive system description, including the synthesis of a digital twin and an exhaustive accuracy analysis, can be found in [44,45]. The CMM can be analogously modeled as a serial robot considering three translational and one rotational degrees of freedom. The imaging parameters of the LLS are found in Table A4.

5.2.2. Vision Task Description

A simple inspection task comprising the dimensional inspection of a self-designed academic probing object consisting of three cylindrical features was regarded to evaluate our framework. The object was coated with a matte white spray to facilitate the acquisition of the cylinders' surfaces. The measurement system and object are depicted in Figure 21.

5.2.3. Assumptions and Adaptation of the Domain Models and Viewpoint Planning Strategy

Since [4] did not explicitly address the application of \mathcal{C} -spaces with LLSs, some additional assumptions regarding the modeling of features, sensors, and viewpoint constraints must be first considered.

Features: The measuring object comprises three cylinders with different positions (see Table A6). The feature frame of the cylinders is placed at the bottom. Taking into account the feature model from Section 2.3, which only considers one frame per feature and assumes that the whole cylinder can be acquired with a single measurement, the definition of the feature model must be extended to guarantee the acquisition of the cylinder's surface area. Thus, two further features for each cylinder (f^{+y} , f^{-y}) were introduced. The new feature frames are located at the half-height of the cylinder and their normal vectors (z-axis) are perpendicular to the x-axis of the feature's origin. The geometrical length of these extra features corresponds to the height of the cylinders. An overview of the frames of all features corresponding to one cylinder are shown in Figure 22.

Sensors and Acquisition of Surface Points: It is assumed that the LLS only moves in a straight line in the CMM's workspace with a fixed sensor orientation. For this reason, we assume that all feature surface points can be acquired with a single scanning trajectory as long as the incidence angle constraint between the surface points and the sensor holds.

\mathcal{I} -space and \mathcal{C} -space: Recalling that \mathcal{C} -spaces are built based on the 3D sensor's \mathcal{I} -space, we first considered a modification of the LLS's 2D frustum. Recalling the previous assumption regarding the acquisition of surface points, let the LLS span a 3D \mathcal{I} -space composed of the 2D \mathcal{I} -space and a width corresponding to the distance of the scanning trajectory. The resulting \mathcal{I} -space of one scanning direction and the resulting \mathcal{C} -spaces for one cylinder and its three features are visualized in Figure 22. Having characterized a 3D frustum, the approach presented in Algorithm 1 can be directly applied to span the required \mathcal{C} -space for one feature. The successful acquisition of one feature results from moving the sensor from an arbitrary viewpoint from one end of the \mathcal{C} -space to another arbitrary viewpoint at its other end. This study only considers the characterization of one \mathcal{C} -space for the sensor's laser.

\mathcal{C} -space and Multi-features: Within multi-feature scenarios, it is desirable to acquire as many features as possible during one linear motion. Therefore, in the simplest scenario, the length of all scanning trajectories corresponds to the size of the object's longest dimension. Under this premise, we assume that the width of the \mathcal{I} -space, hence, of each single \mathcal{C} -space, corresponds to this exact length.

Viewpoint Constraints: Equally to other vision systems, the successful acquisition of surface points depends on the compliance of some geometric viewpoint constraints, such as the imaging capabilities of the LLS (c_1) and the consideration of the features' geometrical dimensions (c_2). Since the scope of this study prioritizes the transferability of the viewpoint strategy, only these constraints were considered to guarantee the successful acquisition of the regarded features. The adaptation and validation of further viewpoint constraints lie outside the scope of this publication and remain to be further investigated.

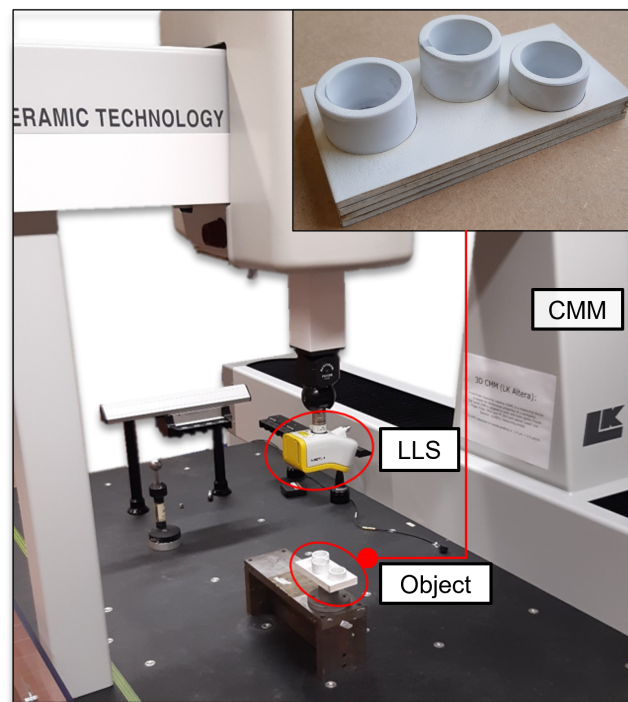


Figure 21. A CMM vision system consisting of two main hardware components: a CMM and the LLS. The probing object composed of three cylinders is positioned within the workspace of the CMM.

5.2.4. Results

Under consideration of the previously mentioned assumptions and modification of the domain models, the necessary ${}^G\mathcal{C}$ -spaces were computed to inspect the nine features of the three cylinders aligned to the viewpoint plan strategy proposed in Section 4.2. The clustering and optimization of the sensor orientation can be performed analogously to the presented methods from Section 4.2. However, unlike what is suggested in Section 2.1.3 and assuming that multiple features could be acquired within the same scanning trajectory, a less conservative condition regarding the maximal Euclidean distance of the clustering algorithm was regarded using the maximal length of the workpiece, i.e., $l_{max} = 120$ mm. Furthermore, the optimization of the swing angle (rotation around the optical z-axis, see Section 4.2.2) was particularly useful for estimating the optimized scanning direction of the sensor.

Considering the minimal established viewpoint constraints, the viewpoint planning strategy yielded three ${}^G\mathcal{C}$ -spaces to acquire all nine features of the three cylinders of the object. Figure 22 illustrates the resulting ${}^G\mathcal{C}$ -spaces manifolds. Each ${}^G\mathcal{C}$ -space corresponds to the acquisition of three features for three different acquisition directions ($z, +y, -y$): one at the top (${}^{G_1}\mathcal{C}^z$) and two at both lateral sides (${}^{G_2}\mathcal{C}^{+y}, {}^{G_3}\mathcal{C}^{-y}$) of the object.

The computation time of the whole strategy corresponded to $t_{total} \approx 610$ ms, confirming the efficiency of the framework. To validate the computed ${}^G\mathcal{C}$ -spaces, four scanning trajectories were selected. Each scanning trajectory's start and end position corresponds to two extreme ${}^G\mathcal{C}$ -spaces vertices, see Table A5. Figure 23 visualizes the corresponding scanning trajectories of all ${}^G\mathcal{C}$ -spaces.

Furthermore, an exemplary representation of the sensor placement of the resulting scanning trajectories of the third ${}^G\mathcal{C}$ -space ${}^{G_3}\mathcal{C}^{-y}$, the corresponding 2D frustum, and the acquired surface points are shown in Figure A3. These extreme scanning trajectories demonstrate the validity of the computed ${}^G\mathcal{C}$ -space, which guarantees that the height of all three cylinders and the outer radii always lie inside the \mathcal{I} -space for all four scans. It can also be assumed that any arbitrary combination of start and end positions within the outer planes of the ${}^G\mathcal{C}$ -spaces will also be valid. The qualitative measurability function

from Section 5.1.3 could be straightforwardly applied to all scanning trajectories and corresponded to the complete acquisition of all features for at least one trajectory.

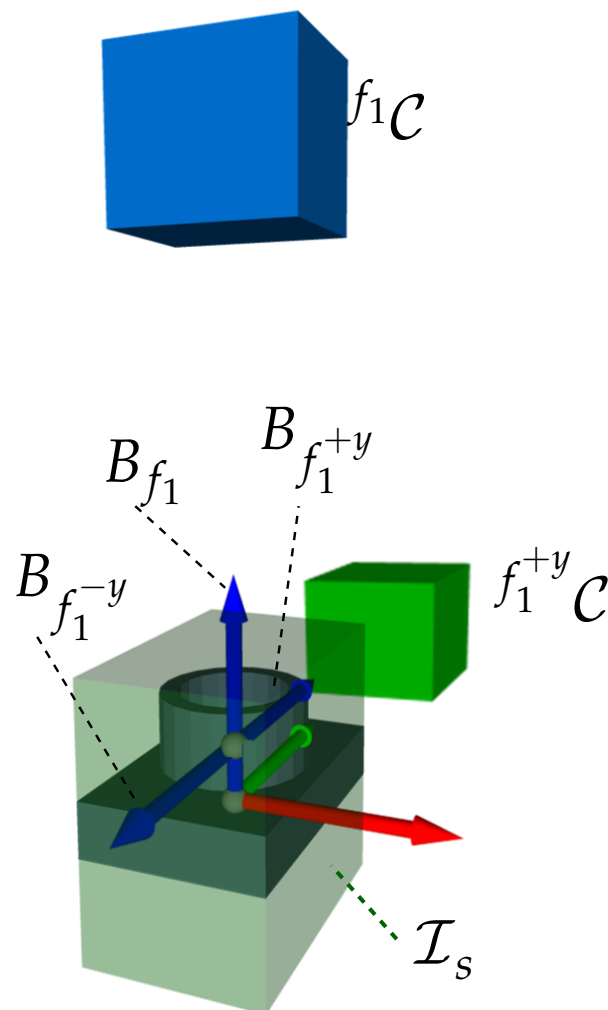


Figure 22. Cylinder with three features, extended model of the sensor \mathcal{I}_s for an LLS, and two exemplary \mathcal{C} -spaces for each feature f_1 and f_1^{+y} .

It should be noted that some surface points could not be successfully acquired for some single extreme scanning trajectories, e.g., the scans 1 and 2 from Figure A3. This failed acquisition can be attributed to the manufacturing tolerances of the self-designed object. However, such tolerances can be seamlessly compensated by selecting alternative sensor positions within the ${}^G\mathcal{C}$ -spaces or considering tolerances implicitly in the synthesis of the \mathcal{C} -spaces, as in the first evaluation case (cf. c_3 from Table 1).

Furthermore, the measurements of all scanning trajectories are depicted in Figure 24, confirming the validity of the viewpoint plan and computed ${}^G\mathcal{C}$ -spaces. The surface model of the object, manifolds of the computed ${}^G\mathcal{C}$ -spaces, and the performed measurements can be found in the digital appendix of this paper. From these experimental results, it can be concluded that the proposed viewpoint planning strategy based on \mathcal{C} -spaces and ${}^G\mathcal{C}$ -spaces could be satisfactorily applied for vision systems comprising LLSs under consideration of certain assumptions.

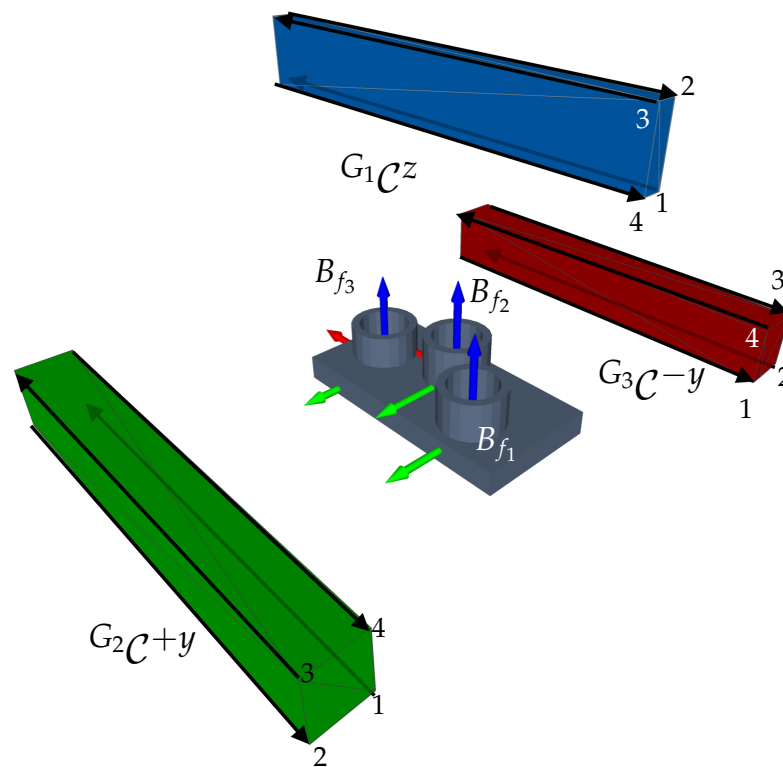


Figure 23. Overview of the three G_C -spaces (G_1C^z , G_2C^+y , G_3C^-y) and visualization of the four scanning tracks (black lines).

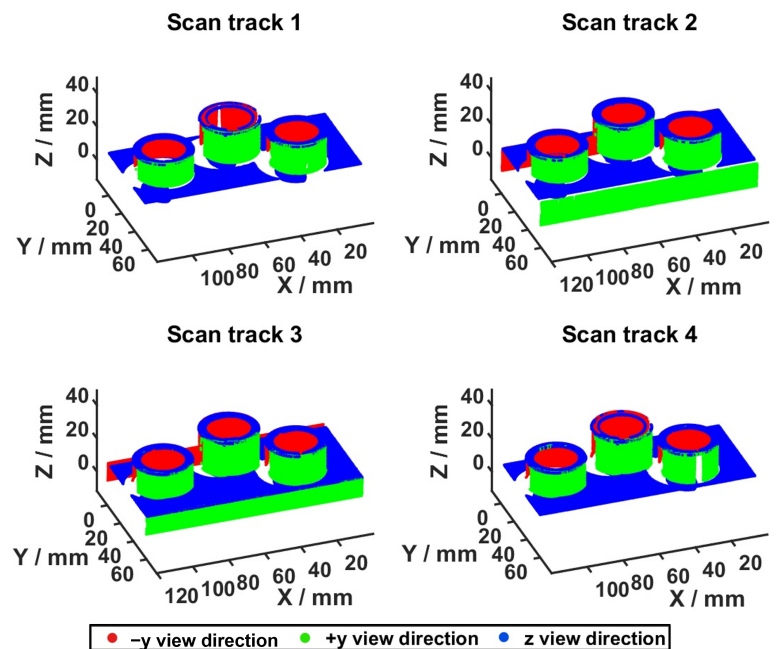


Figure 24. The combined point cloud of the three view directions for all four scan tracks.

5.3. Discussion

A more comprehensive evaluation of the strengths and limitations of the viewpoint planning strategy is discussed in the context of the main goals of this publication:

5.3.1. Efficacy

On the one hand, the effectiveness of the computed viewpoint plans and chosen sensor poses showed an encouraging average measurability satisfiability of over 95% for all regarded vision tasks. It was demonstrated that by considering more viewpoint constraints, the efficacy of the viewpoint plans could be increased towards full measurability. On the other hand, in some cases, a complete measurability could not be achieved due to the occlusion limitations of the planning strategy. Although the present work considered occlusion constraints and the validity of occlusion-free \mathcal{C} -spaces could be verified for individual cases, a proper strategy to consider more complex scenarios remains a stimulus for further research.

Moreover, although the ${}^G\mathcal{C}$ -spaces are valid only for a fixed sensor orientation, the experiments showed that the selected sensor poses were also effective and robust for compensating for minor orientation deviations. In case an orientation range must be explicitly considered, the future works should then extend the strategy to consider ${}^G\mathcal{C}$ -spaces for multiple sensor orientations, as suggested in [4].

Furthermore, most of the selected viewpoints within the ${}^G\mathcal{C}$ -spaces were shown to be valid and sufficient for passing the considered evaluation metrics. However, within the evaluation, we also observed that some real measurements differed from the simulated evaluations due to modeling uncertainties or manufacturing tolerances. An individual analysis of some of these measurements demonstrated that these uncertainties could be compensated for by selecting an alternative sensor pose within the corresponding ${}^G\mathcal{C}$ -space. This characteristic of ${}^G\mathcal{C}$ -spaces embodies the intrinsic benefits of \mathcal{C} -spaces for compensating for uncertainties without compromising the validity of the viewpoint plan. However, these findings also suggest that the further research should be devoted to designing optimization strategies for finding alternative viewpoints within ${}^G\mathcal{C}$ -spaces to compensate for target-oriented uncertainties and neglected constraints.

5.3.2. Computational Efficiency

The comprehensive analysis of the computational complexity demonstrated that most view plans could be computed in near linear time. Due to the complexity of the developed software framework and dependency on external libraries, a more detailed computational analysis is needed to verify the complexity of the strategy and \mathcal{C} -space. However, our experiments confirmed the strength and simplicity of ${}^G\mathcal{C}$ -spaces, showing that effective viewpoint plans could be computed within feasible times despite the complexity of the vision tasks and vision systems.

Moreover, the estimated computational times also showed that if occlusion constraints are regarded, the complexity of the overall strategy increases considerably and depends strongly on the complexity of the considered surface models. Hence, the future research should be devoted to a more efficient characterization and computation of the occlusion spaces.

Furthermore, in terms of overall planning efficiency, the further studies should concentrate on the extension of the present viewpoint planning strategy for considering the proper combination of ${}^G\mathcal{C}$ -spaces in case of overlapping (e.g., see right-handed image of Figure 18) to reduce the number of required viewpoints.

5.3.3. Transferability

This study demonstrated that the usability of \mathcal{C} -spaces, ${}^G\mathcal{C}$ -spaces, and the overall viewpoint planning strategy could be satisfactorily evaluated with different vision systems. In the context of a simplified inspection scenario and considering an adaption of the models introduced in the current study, the applicability of the suggested strategy and potential of ${}^G\mathcal{C}$ -spaces for LLSs was demonstrated.

However, further work remains to be carried out to properly evaluate the use of ${}^G\mathcal{C}$ -spaces for LLS and to consider more complex vision tasks comprising further viewpoint constraints, e.g., occlusion constraints.

6. Conclusions

6.1. Summary

The VPP is a multi-dimensional and challenging problem that any vision task demanding the computation of multiple and valid viewpoints must consider. Based on our literature review, the VPP is still regarded as a complex and unsolved challenge, lacking a generic and efficient framework for computing multiple viewpoints within feature-based applications.

Towards tackling the VPP, the present work proposes its modularization and addresses its subproblems separately, i.e., the VGP and the SCP. First, based on our previous work [4], the present study poses the VGP for multi-feature applications, addressing it as a purely geometric problem that can be solved based on ${}^G\mathcal{C}$ -spaces. ${}^G\mathcal{C}$ -spaces span continuous solution spaces in 6D, providing an infinite number of valid viewpoints that guarantee the successful acquisition of feature clusters, taking into account various viewpoint constraints. The experiments undertaken within this publication demonstrate that spanning an infinite solution space is a powerful technique for implicitly considering model uncertainties within real applications and delivering alternative solutions. Moreover, to address the SCP, we proposed a holistic viewpoint planning strategy based on a heuristic clustering method to identify the sufficient number of ${}^G\mathcal{C}$ -spaces required to fulfill a vision task. The validity, efficiency, and transferability of the viewpoint planning strategy proposed in this study were evaluated using two different industrial vision systems within dimensional metrology tasks. Our evaluation showed that valid viewpoints could be computed for diverse inspection tasks and sensors in linear times, guaranteeing an acquisition of up to 90% of all features.

The key contributions and advantages of a viewpoint planning strategy based on ${}^G\mathcal{C}$ -spaces are summarized as follows:

- Mathematical and generic formulation of the VPP to ease the transferability and promote the extensibility of the framework for diverse vision systems and tasks.
- Synthesis of ${}^G\mathcal{C}$ -spaces built upon \mathcal{C} -spaces, inheriting some of their intrinsic advantages:
 - analytical, model-based, and closed-form solutions,
 - simple characterization based on constructive solid geometry (CSG) Boolean techniques,
 - infinite solutions for the seamless compensation of model uncertainties.
- Generic and modular viewpoint planning strategy, which can be adapted to diverse vision tasks, systems, and constraints.

6.2. Limitations and Future Work

The outlined viewpoint strategy can be categorized as a model-based approach requiring minimal a priori knowledge about the sensor's frustum model and the location and geometry of the features to be acquired. However, the present framework does not consider a stringent definition of features. Hence, the future work should evaluate its usability and adaptability for diverse vision tasks.

Our previous work [4] introduced the core concepts required for synthesizing ${}^G\mathcal{C}$ -spaces and demonstrated that several viewpoint constraints could be modeled geometrically. However, in the context of applications demanding multiple viewpoints, we observed that the potential of ${}^G\mathcal{C}$ -space remains to be further exploited for considering further constraints. On the one hand, based on our ongoing research and preliminary results, we still see the potential for explicitly characterizing registration constraints for maximizing and ensuring the overlapping area between adjacent measurements [46]. On the other hand, we can also imagine that ${}^G\mathcal{C}$ -spaces could be used for considering further robot constraints such as sensor lighting parameters, robot collisions, cycle times, and energy-efficiency constraints. Concretely, our current research concentrates on exploiting the use of ${}^G\mathcal{C}$ -spaces for simultaneously optimizing the sensor exposure time and sensor pose.

Although our experimental results demonstrated encouraging results regarding the efficacy and efficiency of the present framework, further studies should concentrate on the definition of a benchmark scenario and standardized evaluation metrics that facilitate a direct and more comprehensive evaluation between diverse approaches. Due to the nature of the approach presented, which suggests an explicit characterization of the domain models and constraints, a non-negligible effort to implement and parameterize the models as proposed in this research should be considered.

Despite the mentioned limitations, we are convinced that the outlined viewpoint planning strategy based on ${}^G\mathcal{C}$ -spaces provides a springboard for a novel and efficient approach for tackling the VPP, comprising closed and deterministic solutions. We hope that our findings aid researchers and industry in enabling the automation of diverse vision tasks.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s23187964/s1>.

Author Contributions: Conceptualization, A.M., M.V., H.H., P.B., B.S. and G.R.; methodology, A.M.; software, A.M.; validation, A.M. and M.V.; formal analysis, A.M.; investigation, A.M.; resources, H.H. and G.R.; data curation, A.M. and M.V.; writing—original draft preparation, A.M. and M.V.; writing—review and editing, A.M., M.V., H.H., P.B., B.S. and G.R.; visualization, A.M. and M.V.; supervision, H.H. and 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).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A comprehensive set of supplemental material is attached to this paper. This set includes ${}^G\mathcal{C}$ -spaces and \mathcal{C} -spaces meshes, meshes and frames of the used solid models, features and viewpoint frames, and the 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.

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

Abbreviations

The following abbreviations are used in this manuscript:

\mathcal{C} -space	Feature-based constrained space
${}^G\mathcal{C}$ -space	Feature cluster constrained space
RVS	Robot vision system
SCP	Set cover problem
VGP	Viewpoint generation problem
VPP	Viewpoint planning problem

Appendix A. Tables and Figures

Table A1. Overview and description of the viewpoint constraints from [4].

Viewpoint Constraint	Brief Description
1. Frustum Space	The most restrictive and fundamental constraint is given by the imaging capabilities of the sensor. This constraint is fulfilled in its most elementary formulation if at least the feature's origin lies within the frustum space (cf. Section 2.1.1).
2. Feature Orientation and Incidence Angle	The maximal permitted incidence angle between the optical axis and the feature normal is not allowed to exceed a maximal angle determined by the sensor manufacturer. (2).
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 compromise a viewpoint's validity. Hence, any factor, e.g., kinematic alignment, robot's pose accuracy, affecting the overall kinematic chain of the RVS must be considered.
5. Sensor Accuracy	It is assumed that the sensor accuracy may vary within the sensor image space.
6. Feature Occlusion	A viewpoint can be considered valid if a free line of sight exists from the sensor to the feature.
7. Bistatic Sensor and Multisensor	Recalling the bistatic nature of range sensors, all viewpoint constraints must be valid for all lenses or active sources.

Table A2. Overview of index notations for variables.

Notation	Index Description
	<p>x :=variable, parameter, vector, frame or transformation</p> <p>d :=RVS domain, i.e., <u>sensor</u>, <u>robot</u>, <u>feature</u>, <u>object</u>, <u>environment</u> or d element of a list or set</p>
${}^r_b x^d$	<p>n :=related domain, additional notation or depending variable</p> <p>r :=base frame of the coordinate system B_r or space of feature f</p> <p>b :=origin frame of the coordinate system B_b</p>
Notes	The indexes r and b only apply for pose vectors, frames, and transformations.
Example	<ul style="list-style-type: none"> • d: Let a sensor (s) pose be denoted as follows $p_s \in \mathbb{R}^3$ • d: The sensor pose with the index 5 is given as follows $p_{s,5}$. • n: The coordinate system of the of the sensor lens of the first imaging device is denoted by $B_s^{s_1}$. • b: To specify the origin frame of the sensor pose, e.g., in $B_s^{s_1}$, then the following notation applies: ${}_{s_1}p_{s,5}$. • r: Assuming that the sensor pose is given in the coordinate system of the feature, B_f, then it follows that ${}^f p_{s,5}$ or ${}_{s_1}^f p_{s,5}$.

Table A3. Overview of features and occlusion objects used for academic example.

Feature	f_1	f_2	f_3	f_4	κ_1	κ_2
Topology	Circle	Slot	Circle	Slot	Octahedron	Cube
Dimensions in mm	radius = 20	length = 150	radius = 20	length = 50	edge length ≈ 28.3	length = 50.0
Translation vector in object's frame $t_o = (x_o, y_o, z_o)^T$ in mm	$\begin{pmatrix} 225.0 \\ 100.0 \\ 25.0 \end{pmatrix}$	$\begin{pmatrix} 100.0 \\ 85.0 \\ 0.0 \end{pmatrix}$	$\begin{pmatrix} 250.0 \\ 200.0 \\ 0.0 \end{pmatrix}$	$\begin{pmatrix} 150.0 \\ 225.0 \\ 20.0 \end{pmatrix}$	$\begin{pmatrix} 200.0 \\ 200.0 \\ 400.0 \end{pmatrix}$	$\begin{pmatrix} 245.0 \\ 115.0 \\ 200.0 \end{pmatrix}$
Rotation in Euler Angles in object's frame $r_o(\gamma_s^x, \beta_s^y, \alpha_s^z)$ in $^\circ$	(20, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 30, 0)	(0, 0, 0)	(0, 0, 0)

Table A4. Imaging parameters of the two used sensors.

Range Sensor	1		2
Manufacturer	Carl Zeiss Optotechnik GmbH		Nikon
Model	COMET Pro AE		LC60Dx
3D Acquisition Method	Digital Fringe Projection		Laser Scanner
Imaging Devices	Monochrome Camera (s_1)	Blue Light LED-Fringe Projector (s_2)	Laser Diode and Optical Sensor
FOV angles	$\theta_s^x = 51.5^\circ, \psi_s^y = 35.5^\circ$		$\theta_s^x = 35.05^\circ$
Working distances and corresponding near, middle, and far planes.	@400 mm : (396 × 266) mm	@200 mm : (284 × 160) mm	@95 mm : 60 mm
	@600 mm : (588 × 392) mm	@600 mm : (853 × 480) mm	@125 mm : 60 mm
	@800 mm : (780 × 520) mm	@1000 mm : (1422 × 800) mm	@155 mm : 60 mm

Table A5. Overview of viewpoint planning results considering diverse inspection tasks.

Vision Tasks Configuration						Results						
Vision Task	Door Side	Feature Type	Constraints	Nr. of Features	Nr. of Clusters	Measurability Index in %,			Computation Time in s			
						M^{qual}	M_{sim}^{quant}	M_{real}^{quant}	$t_{k,opt}$	t_C	t_{C_C}	t_{total}
1	All	All	1–3	673	75	97.77			8.35	140.94	19.11	168.41
2	All	All	1–4	673	77	97.47			9.28	265.99	43.59	318.86
3	All	Circles	1–3	50	12	92.00			1.53	2.76	0.66	4.94
4 *	All	Circles	1–4	50	10, $\sigma = 0$	92.00			0.98, $\sigma = 0.10$	6.32, $\sigma = 0.66$	1.74, $\sigma = 0.03$	9.04, $\sigma = 0.73$
5	All	Circles	1–5	50	10	96.00	96.00	82.00	1.26	77.27	2.83	81.36
6 *	Inside	All	1–3	157	14, $\sigma = 0.89$	100.00			2.73, $\sigma = 0.27$	8.75, $\sigma = 0.39$	1.78, $\sigma = 0.18$	13.26, $\sigma = 0.66$

Table A5. Cont.

Vision Tasks Configuration						Results						
Vision Task	Door Side	Feature Type	Constraints	Nr. of Features	Nr. of Clusters	Measurability Index in %,			Computation Time in s			
						M^{qual}	M_{sim}^{quant}	M_{real}^{quant}	$t_{k,opt}$	t_C	t_{C_C}	t_{total}
7	Inside	All	1–4	157	14	100.00	96.82	88.54	2.38	18.33	4.12	24.83
8	Inside	Circles	1–3	34	4	100.00			1.04	1.81	0.38	3.24
9	Inside	Circles	1–4	34	4	100.00			1.08	3.64	0.89	5.62
10	Outside	All	1–3	267	26	97.75			3.71	16.72	3.80	24.23
11	Outside	All	1–4	267	26	97.75			4.00	39.02	10.38	53.40
12	Outside	All	1–5	267	26	94.01	88.76		3.68	961.84	50.46	1015.98
13	Outside	Circles	1–3	11	5	63.64			1.16	0.70	0.13	1.99
14	Outside	Circles	1–4	11	5	63.64			1.09	1.28	0.46	2.83
15 *	Outside	Circles	1–5	11	5, $\sigma = 0$	81.82			0.68, $\sigma =$ 0.10	15.12, $\sigma =$ 0.60	2.50, $\sigma =$ 0.15	18.78, $\sigma =$ 0.81

* The results consider the mean and standard deviation (σ) of a ten-times repeated computation.

Table A6. The overview of the cylinders used for CMM.

Feature	f_1	f_2	f_3
Topology	Cylinder	Cylinder	Cylinder
Dimensions in mm	radius = 16, height = 25		
Translation vector in object's frame $t_o = (x_o, y_o, z_o)^T$ in mm	$\begin{pmatrix} 30.0 \\ 40.0 \\ 6 \end{pmatrix}$	$\begin{pmatrix} 60.0 \\ 20.0 \\ 12 \end{pmatrix}$	$\begin{pmatrix} 100.0 \\ 30.0 \\ 3 \end{pmatrix}$
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)	(0,0,0)

Table A7. Coordinates of the start- and end-position of each scanning trajectory.

		Start Coordinate in mm			Start Coordinate in mm		
		X	Y	Z	X	Y	Z
−y view direction	1	−35	34	4	164	34	4
	2	−35	10	4	164	10	4
	3	−35	10	33	164	10	33
	4	−35	34	33	164	34	33
+y view direction	1	−35	26	4	164	26	4
	2	−35	50	4	164	50	4
	3	−35	50	33	164	50	33
	4	−35	26	33	164	26	33
z view direction	1	−35	26	4	164	26	4
	2	−35	26	33	164	26	33
	3	−35	34	33	164	34	33
	4	−35	34	4	164	34	4

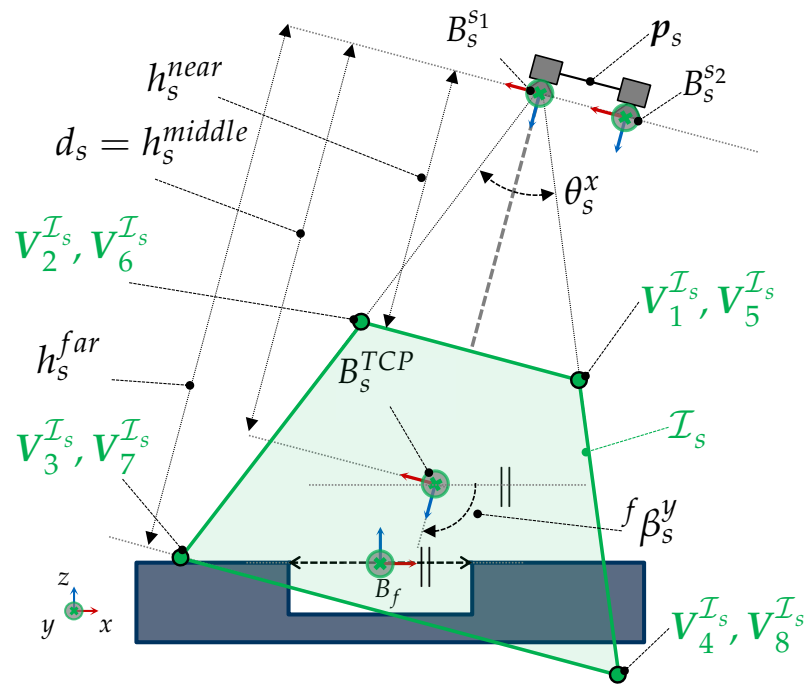


Figure A1. 2D simplified visualization of the kinematic and imaging model of the sensor in the x-z plane. The imaging parameters of the sensor $(d_s, h_s^{near}, h_s^{far}, \theta_s^x, \psi_s^y) \in \mathcal{I}_s$ for a given sensor pose p_s span the frustum space \mathcal{I}_s . A minimum of eight vertices $V_{1-8}^{\mathcal{I}_s}$ are required to characterize the \mathcal{I} -space manifold. (image from Magaña et al. [4]).

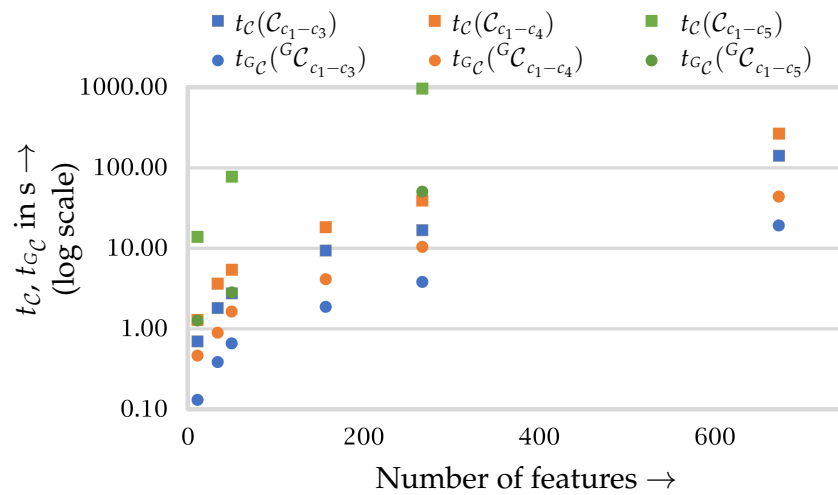


Figure A2. Overview of the computation times for the characterization of \mathcal{C} -spaces (t_c) and ${}^G\mathcal{C}$ -spaces (t_{GC}), depending on the number of features considering different viewpoint constraints.

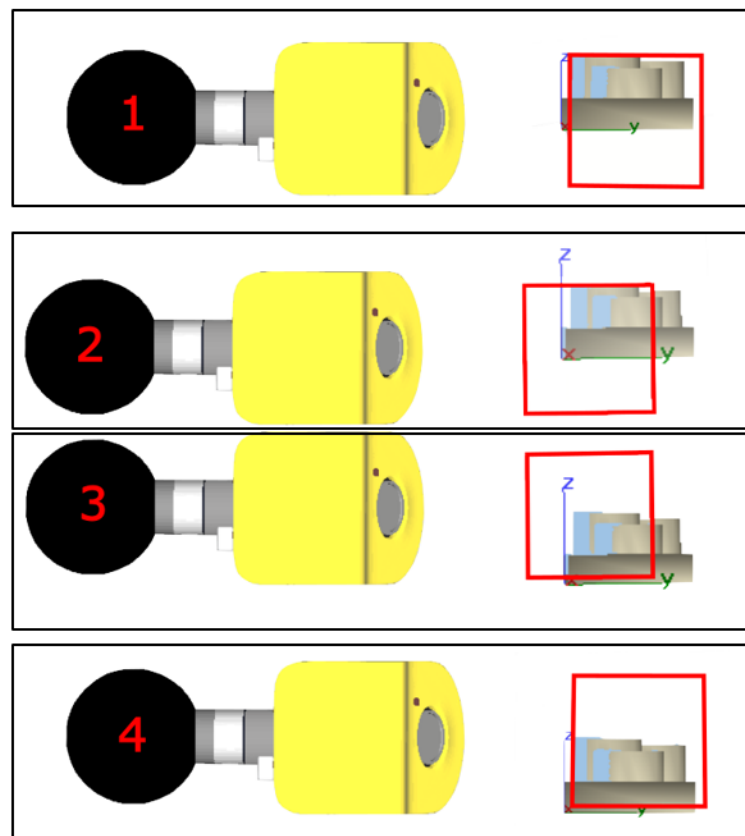


Figure A3. 2D Visualization in the z - y plane of the four scanning trajectories on the boundaries (vertices) of G_3C-y , frustum (red square), and measured points (blue surface).

References

- Müller, C.; Kutzbach, N. *World Robotics 2019—Industrial Robots*; IFR Statistical Department, VDMA Services GmbH: Frankfurt am Main, Germany, 2019.
- 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*. [\[CrossRef\]](#)
- Magaña, A.; Dirr, J.; Bauer, P.; Reinhart, G. Viewpoint Generation Using Feature-Based Constrained Spaces for Robot Vision Systems. *Robotics* **2023**, *12*, 108. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Mavrinac, A.; Chen, X. Modeling Coverage in Camera Networks: A Survey. *Int. J. Comput. Vis.* **2013**, *101*, 205–226. [\[CrossRef\]](#)
- Kritter, J.; Brévilliers, M.; Lepagnet, J.; Idoumghar, L. On the optimal placement of cameras for surveillance and the underlying set cover problem. *Appl. Soft Comput.* **2019**, *74*, 133–153. [\[CrossRef\]](#)
- 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, AR, USA, 14–19 May 1989; IEEE Computer Society Press: New York, NY, USA, 1989; pp. 509–514. [\[CrossRef\]](#)
- Tarabanis, K.; Tsai, R.Y.; Kaul, A. Computing occlusion-free viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 279–292. [\[CrossRef\]](#)
- Abrams, S.; Allen, P.K.; Tarabanis, K. Computing Camera Viewpoints in an Active Robot Work Cell. *Int. J. Robot. Res.* **1999**, *18*, 267–285. [\[CrossRef\]](#)

14. Reed, M. Solid Model Acquisition from Range Imagery. Ph.D. Thesis, Columbia University, New York, NY, USA, 1998.
15. Reed, M.K.; Allen, P.K. Constraint-based sensor planning for scene modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1460–1467. [[CrossRef](#)]
16. Tarbox, G.H.; Gottschlich, S.N. Planning for complete sensor coverage in inspection. *Comput. Vis. Image Underst.* **1995**, *61*, 84–111. [[CrossRef](#)]
17. Scott, W.R. Performance-Oriented View Planning for Automated Object Reconstruction. Ph.D. Thesis, University of Ottawa, Ottawa, ON, Canada, 2002.
18. Scott, W.R. Model-based view planning. *Mach. Vis. Appl.* **2009**, *20*, 47–69. [[CrossRef](#)]
19. Tarbox, G.H.; Gottschlich, S.N. IVIS: An integrated volumetric inspection system. *Comput. Vis. Image Underst.* **1994**, *61*, 430–444. [[CrossRef](#)]
20. Gronle, M.; Osten, W. View and sensor planning for multi-sensor surface inspection. *Surf. Topogr. Metrol. Prop.* **2016**, *4*, 024009. [[CrossRef](#)]
21. 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), IEEE, Vancouver, BC, Canada, 24–28 September 2017; pp. 5211–5218. [[CrossRef](#)]
22. Mosbach, D.; Gospodnetić, P.; Rauhut, M.; Hamann, B.; Hagen, H. Feature-Driven Viewpoint Placement for Model-Based Surface Inspection. *Mach. Vis. Appl.* **2021**, *32*, 1–21. [[CrossRef](#)]
23. 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](#)]
24. Son, S.; Kim, S.; Lee, K.H. Path planning of multi-patched freeform surfaces for laser scanning. *Int. J. Adv. Manuf. Technol.* **2003**, *22*, 424–435. [[CrossRef](#)]
25. 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](#)]
26. Tekou Moutchiho, W.B. A New Programming Approach for Robot-Based Flexible Inspection Systems. Ph.D. Thesis, Technical University of Munich, Munich, Germany, 2012.
27. Fernández, P.; Rico, J.C.; Álvarez, B.J.; Valiño, G.; Mateos, S. Laser scan planning based on visibility analysis and space partitioning techniques. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 699–715. [[CrossRef](#)]
28. Raffaelli, 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](#)]
29. Koutecký, T.; Paloušek, D.; Brandejs, J. Sensor planning system for fringe projection scanning of sheet metal parts. *Measurement* **2016**, *94*, 60–70. [[CrossRef](#)]
30. González-Banos, H. A randomized art-gallery algorithm for sensor placement. In Proceedings of the Seventeenth Annual Symposium on Computational Geometry—SCG '01, Medford, MA, USA, 3–5 June 2001; Souvaine, D.L., Ed.; ACM Press: New York, NY, USA, 2001; pp. 232–240. [[CrossRef](#)]
31. Chen, S.Y.; Li, Y.F. Automatic sensor placement for model-based robot vision. *IEEE Trans. Syst. Man Cybern. Part B Cybern. Publ. IEEE Syst. Man Cybern. Soc.* **2004**, *34*, 393–408. [[CrossRef](#)]
32. 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](#)]
33. Mavrinac, A.; Chen, X.; Alarcon-Herrera, J.L. Semiautomatic Model-Based View Planning for Active Triangulation 3-D Inspection Systems. *IEEE/ASME Trans. Mechatronics* **2015**, *20*, 799–811. [[CrossRef](#)]
34. 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](#)]
35. 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](#)]
36. O'Rourke, J. Finding minimal enclosing boxes. *Int. J. Comput. Inf. Sci.* **1985**, *14*, 183–199. [[CrossRef](#)]
37. Trimesh. Trimesh. 2023. Available online: <https://github.com/mikedh/trimesh> (accessed on 3 September 2023).
38. Zhou, Q.Y.; Park, J.; Koltun, V. Open3D: A Modern Library for 3D Data Processing. *arXiv* **2018**, arXiv:1801.09847. [[CrossRef](#)]
39. 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, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
40. Magaña, A.; Bauer, P.; Reinhart, G. Concept of a learning knowledge-based system for programming industrial robots. *Procedia CIRP* **2019**, *79*, 626–631. [[CrossRef](#)]
41. 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; pp. 1511–1518. [[CrossRef](#)]
42. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [[CrossRef](#)]
43. Zhou, Q.; Grinspun, E.; Zorin, D.; Jacobson, A. Mesh arrangements for solid geometry. *ACM Trans. Graph.* **2016**, *35*, 1–15. [[CrossRef](#)]

44. Vlaeyen, M.; Haitjema, H.; Dewulf, W. Digital Twin of an Optical Measurement System. *Sensors* **2021**, *21*, 6638. [[CrossRef](#)] [[PubMed](#)]
45. Vlaeyen, M.; Haitjema, H.; Dewulf, W. Error compensation for laser line scanners. *Measurement* **2021**, *175*, 109085. [[CrossRef](#)]
46. 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](#)]

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.