Machine learning-based 3D scan coverage prediction for smart-control applications

Abstract

Automatic control of a workpiece being manufactured is a requirement for smart-manufacturing, to ensure in-line correction and reduce rejects. There is therefore a need to develop control strategies which are capable of taking precise account of real working conditions and enabling first-time-right control. As part of such a smart-control strategy, this paper introduces a machine learning-based approach capable of accurately predicting a priori the 3D coverage of a part according to a scan configuration given as input, i.e. predicting before scanning it which areas of the part will be acquired for real. This corresponds to a paradigm shift, where coverage estimation no longer relies on theoretical visibility criteria, but on rules learned from a large amount of data acquired in real-life conditions. The proposed 3D Scan Coverage Prediction Network (3DSCP-Net) is based on a 3D feature encoding and decoding module, which is capable of taking into account the specifics of the scan configuration whose impact on the 3D coverage is to be predicted. To take account of real working conditions, features are extracted at various levels, including geometric ones, but also features characterising the way structured-light projection behaves. The method is thus able to incorporate inter-reflection and overexposure issues into the prediction process. The database used for the training was built using an ad-hoc platform specially designed to enable the automatic acquisition and labelling of numerous point clouds from a wide variety of scan configurations. Experiments on several parts show that the method can efficiently predict the scan coverage, and that it outperforms conventional approaches based on purely theoretical visibility criteria.

Keywords: Structured-light measurement, point cloud acquisition, a priori coverage prediction, first-time-right control.

1. Introduction

The automatic control of a workpiece being manufactured, the analysis of deviations between the manufactured geometry and the theoretical one, and the subsequent correction of the machining parameters have attracted much attention within the smart control and manufacturing community over the last few decades [1]. Today, in the context of the Industry 4.0, many attempts are being made to meet these challenges by means of complex digital twins developed to be able to capture the state of a physical system, understand its behaviour and take decisions to improve it [2]. Unfortunately, not all stages of the workflow described above have yet benefited from these digitisation improvements, and this is particularly true for the measurement step of the manufactured shapes, which is essential for controlling the deviations between the theory and real world.

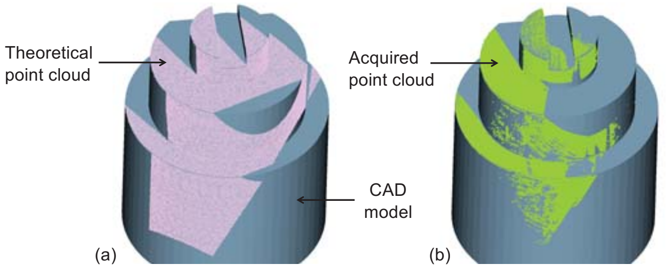

Figure 1: Comparison and discrepancies between the theoretical coverage of a point cloud virtually generated using simple visibility criteria (a), and a real one obtained using a structured-light scanner (b), both obtained from the same viewpoint.

Actually, measurement devices can be divided into two main categories: contact and contactless depending on how they interact with the object [3], or alternatively, active and passive depending on the underlying measurement principles [4]. Within the first category, coordinate measuring machines are certainly the most accurate ones [5], but they suffer from practical implementation difficulties when used for in-line inspection [6], i.e. with a workpiece still in place on the production line, and are therefore more adapted for off-line control. Among contactless technologies, optical-based measurements have become popular in a variety of applications, such as the use of lidar sensors for autonomous driving [7], three dimensional (3D) scanners for part surface inspection [8], or total stations for building modelling [9]. Photogrammetry techniques can also fall into this category, but do not reach performance levels compatible with the requirements of smart control and manufacturing [10]. The optical-based surface measurement scanners are now commonly used in the industry and allow fast 3D acquisitions of dense point clouds [1]. They can be divided into three types depending on the adopted underlying principles: (1) time-of-flight; (2) stereo vision; (3) structured-light [11]. While the first principle is useful to reconstruct large 3D areas such as urban scenes, the latter two are more suited to metrological applications, as it has been demonstrated in recent years [1, 12, 13]. A major advantage of structured-light is that it can achieve relatively high spatial resolution by using off-the-shelf DLP projectors and HD colour cameras, making this technology a good candidate for automatic in-line inspection and control.

However, when integrated within an automatic inspection line, the real behaviour of a structured-light scanner often deviates from the theoretical one. This is particularly true when computing both the field of view and the areas that can be covered, i.e. the so-called coverage, from a given viewpoint and configuration of the scanner. Figure 1 clearly shows that, in comparison to a virtually-generated one, the point cloud obtained using a real acquisition device covers a slightly less important area of the scanned object, here represented by its CAD model. These discrepancies are due to the many external parameters that influence the way light patterns are projected, reflected and measured, as well as to the fact that the simple visibility criteria often used to solve such a problem are based on theoretical principles that do not take real phenomena into account. Indeed, the material of the object, the intensity of the ambient light in the acquisition room, the local shapes that can affect inter-reflections during the acquisition, the camera exposure, are examples of influencing parameters that are not yet considered by the current approaches. Consequently, it is risky to use simple visibility criteria to assess the extent to which an area of a workpiece will actually be covered by points acquired by a scanner, without considering real phenomena. The promised information may simply not be available, with incomplete point clouds that may also be of poor quality, with an obvious impact on the targeted control.

Being able to accurately predict *a priori*, i.e. before scanning, what a structured-light scanner will actually acquire from a specific scanning configuration is therefore crucial for optimising in-line part inspection. This is indeed important when solving the view planning problem, and thus to accurately determine the position(s) and configuration(s) of the scanner for a specific control. This is not an easy task, however, as it not only requires the access to the current and up-to-date real status of the system, process and workpiece, but also the development of an approach that makes it possible to establish potentially complex relationships between the many influencing parameters and their impact on the scan coverage. Unfortunately, conventional heuristic-based approaches are hardly applicable to solve such an open challenge, where there are many couplings between the different phenomena and parameters.

This paper introduces a machine learning-based approach trained on a database containing numerous point clouds obtained from widely varying scanning configurations, and capable of accurately predicting *a priori* the coverage of a part, i.e. predicting before scanning it which areas of the part will be acquired for real. This corresponds to a paradigm shift, where coverage estimation no longer relies on theoretical visibility criteria, but on rules learned from a large amount of data acquired in real-life conditions. To achieve this, several questions are here addressed: *how to define the coverage of a part to be scanned*, *what factors influence the coverage*, and *how to determine the relationships between the coverage and the influencing parameters* ? From these analyses, a novel 3D scan coverage prediction network (3DSCP-Net) is proposed. It is based on a 3D feature encoding and decoding module coupled to a channel attention mechanism, which are able to consider and emphasise the specifics of the scan configuration whose impact on the 3D coverage is to be predicted. To take account of real working conditions, features are extracted at various levels, including geometric ones, but also features characterising the way structured-light projection behaves. The method is thus able to incorporate inter-reflection and overexposure issues into the prediction process, which has never been done in this way before. Experiments on various parts show that the method can efficiently predict the scan coverage, which is particularly useful to prepare the scanning sequences and allow first-time-right control of parts in a smart control and manufacturing context. Results using the proposed *a priori* coverage prediction approach are compared to the ones obtained with approaches based on conventional theoretical visibility criteria.

To sum up, the contribution is threefold: (i) a novel prediction framework based on machine learning capable of predicting *a priori* the coverage of scan configurations on unknown models with the help of a semantic segmentation network; (ii) a method to simulate the projection of patterns in structured-light acquisition, in order to integrate the inter-reflection and overexposure issues that can occur during a real scan; (iii) the formalisation of a methodological framework for the definition of a physical platform and its digital twin capable to semi-automatically acquire many point clouds from different scanning configurations. The complete database is 12 GB and is made publicly available for future benchmarking of other algorithms at the following URL: `https://doi.org/10.5281/zenodo.10807742`.

The paper is organised as follows. Section 2 reviews work related to the three questions raised in the introduction. Section 3 details the proposed machine learning-based 3D scan coverage prediction approach, and notably: the way the data are preprocessed and labelled in a semi-automatic manner, how features are extracted at different levels, in particular how the inter-reflection and overexposure issues have been taken into account by simulating the way structured-light devices project patterns, the architecture of the newly developed 3D Scan Coverage Prediction Network (3DSCP-Net). Section 4 details the methodological framework set up to automatically acquire and label many point clouds from widely varying scan configurations, the experiments and results obtained to validate the proposed model, including a comparison to results obtained with state-of-the-art approaches based on conventional theoretical visibility criteria. The final section concludes this paper, discusses the limits of the model as well as the future works.

2. Related works

A priori predicting 3D scan coverage is of major interest for the development of smart-control applications. Indeed, the precise identification, prior to the scan, of the areas of an object that will really be acquired can be crucial in solving the view planning problem (VPP) [1]. This section successively details the works undertaken to solve the VPP problem, the metrics usually adopted to qualify the point clouds, the parameters that can influence the quality of the results, and the machine learning-based techniques developed to learn from 3D geometric models.

View planning problem and smart control. In the context of part control, solving the view planning problem (VPP) consists in automatically determining the optimal positions and/or trajectories of the acquisition devices to ensure complete coverage of the part to be inspected [1]. To do so, a 3D representation of the object may or may not be available. When available, CAD models or meshes are usually used to optimise the number and location of the scanning viewpoints in order to fully cover a given region of interest (ROI). Scott et al. [14] proposed a method to transpose the VPP into the set covering problem (SCP). This problem can be formulated as follows: If P is a set of elements $\{1, 2, 3, ..., n\}$, and Ps is a list of subsets whose union is equal to P, solving the SCP is equivalent to finding the smallest list of subsets in Ps such that the union of this list is equal to P. The SCP is demonstrated as NP complete, so the problem cannot be solved in polynomial time [15]. From this formalisation, methods can be divided into: (1) methods using optimisation algorithms to solve the SCP in a practical time; (2) methods on covering optimisation; (3) methods seeking next-best-scan, where the prior information is not required. As an output, the surface of the part is segmented, usually using a mesh to reduce the complexity of the algorithm. Many efforts have been made to improve and speed up problem resolution. Scott [16] determines simplification level of the decimating mesh manually, and bobble mesh algorithm is applied to subdivide the surface [17]. Several objective functions have also been imagined and minimised/maximised, such as coverage, scanner orientation, overlap between scans [18, 19, 20]. The approaches based on coverage maximisation firstly segment the outer skin of the part into patches, then propose a set of possible positions, and finally identify the right ones while optimising the acquisition times.

The main issue with these works is that they are mainly based on theoretical and perfect models, and do not take into account the limitations of current acquisition devices, and in particular the fact that acquired point clouds often deviate from the theoretical ones (Fig. 1). To move towards smart-control applications, and better consider real working conditions, further efforts are needed. Thus, in this paper, the objective is to develop a method capable of learning how the acquisition behaves under real scanning conditions, in order to obtain a more realistic prediction of what the system is really able to acquire, and therefore a more accurate resolution of the VPP problem.

Quality metrics and coverage. To assess the quality of acquired point clouds, and possibly related scan configurations, several metrics have been proposed and can be of interest when solving the VPP problem. First of all, indicators intrinsic to the point clouds can be computed, for example density, noise, accuracy and completeness as proposed in [21, 22]. Other metrics try to evaluate the measured area, the surface accessibility, the number of directional measurements, the digitisation time, the total positioning distance, or the surface coverage, for the purpose of comparing several acquisition devices [23, 24]. Catalucci et al. [13] also suggested new indicators such as the acquisition time, or the standard and expanded uncertainty on feature size, and they compared their values for different configurations and devices including photogrammetry and use of structured light-based scanners. Based on these studies, Li et al. [25] have evaluated the sensitivity of these indicators when using structured light-based sensors. As a result, it appears that the coverage indicator is sensitive to the scanning configurations when using structured light-based inspection, and that it can therefore have an impact on optimising the VPP problem resolution. In this context, the coverage is defined on the known CAD model to specify whether a region is acquired/covered or not. To do so, the point cloud needs to be registered with the CAD model, and then depending on the density of points located in a region of the CAD model, the region is considered as covered or not. Based on this definition, the coverage prediction problem comes down to assessing which areas of a CAD model are really covered by an acquisition, and this prior to the effective scan of the part and keeping in mind that the extend of these areas may be affected by influencing parameters, as discussed in the next paragraph.

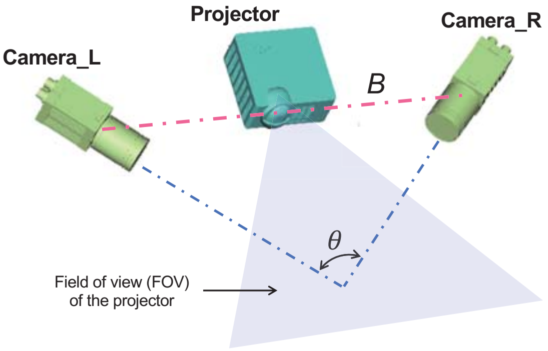

Figure 2: Extrinsic parameters of a binocular configuration used in structured light-based scanners, with baseline length B and angle θ between left and right cameras influencing the reconstruction accuracy, and field of view (FOV) identifying areas where points should be acquired.

Influencing parameters. Several parameters may influence the real coverage of a scanned part, and can be split into four categories, depending on whether they are related to: (a) the reconstruction method, i.e. structured light-based method in the context of this work; (b) the properties of the measured object, including the manufacturing and material characteristics that can also significantly affect the real coverage; (c) the environment parameters, e.g. the ambient light, which also plays a key role on the quality of the acquired point cloud; (d) the scanner configuration, e.g. the position and orientation of the device, as well as all settings, including the exposure. Previous studies have mainly focused on the first category (a), while analysing the influencing parameters in terms of their impact on the precision and accuracy of the reconstruction. In addition to the scan path or multiple viewpoints adopted, the influence of factors related to the lens distortion, as well as to the intrinsic and extrinsic parameters of the hardware have been studied [26] on binocular-based vision devices as shown on Figure 2. First, the lens together with the physical size of the pixel unit of the image sensor chip determine the minimum value of the reconstruction error (i.e. its lower-bound), which is a theoretical value that cannot, however, be reached during real scanning session. Secondly, the extrinsic parameters such as the baseline length B and the angle between the cameras θ (Fig. 2) determine the reconstruction accuracy [27]. The field of view (FOV) also has an impact by limiting the areas in which points are acquired. Here as well, there are discrepancies between the theoretical values and those actually observed in practice. Then, studies on lens distortion revealed that traditional tangential and radial distortion models can be simplified by keeping only the second-order term for radial distortion [27, 28, 29]. However, such simplification cannot be applied for accurate applications, as exact parameters are needed to minimise the deviations between the theoretical and real observed distortions. Lastly, the effective lens focal length, image centre and the pixel cell size taken as intrinsic parameters have slightly larger impact on the reconstruction accuracy with linear correlation [27].

As a conclusion, the influencing parameters in (a) clearly affect the quality of the results, and their potential impact on the deviations between the theoretical and practical acquisitions have been widely studied. Unfortunately, this is not the case for the other three categories (b), (c) and (d), which have received much less attention in the literature. This paper aims at better understanding how these influencing parameters affect the quality of an acquisition, and how they can be used to efficiently model what is happening during real acquisition. In fine, some of these parameters are considered as input of the coverage prediction model developed in this paper, and that uses an ad-hoc machine learning-based approach to discover the intricate relationships involved in the 3D acquisition process.

Machine learning techniques. Learning methods are becoming increasingly popular in many application areas, and in particular in manufacturing and measurement where they have proved effective in solving complex tasks that were previously difficult to achieve in a reasonable time using traditional approaches. This is for instance the case for roughness prediction, for which several attempts have been made to understand the complex underlying phenomena [30, 31, 32, 33], while first analysing the influencing parameters and then introducing the variables into a network for prediction. More recently, Suiyang et al. [33] applied an extreme machine learning method to predict the surface roughness in ultra-precision milling. All these achievements confirm the potential of machine learning-based methods for discovering complex patterns in measurements.

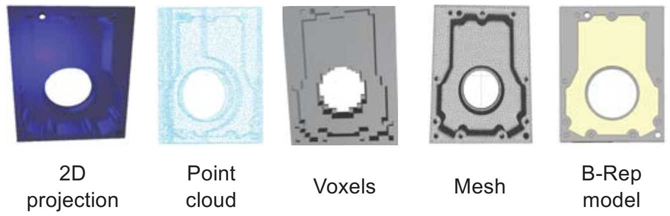

Figure 3: Structured and unstructured 3D representations, traditionally used as input data for current machine learning techniques and influencing the choice of architecture to be implemented.

Throughout the recent decades, neural network-based approaches have demonstrated their efficacy in establishing implicit functions that model the complex relationships among numerous influencing parameters, using a series of samples as input during the training phase. They have acquired a great reputation in two dimensional (2D) tasks such as classification and semantic segmentation. Convolution Neural Network (CNN) is widely used for unstructured data, and Recurrent Neural Network (RNN) for sequence data, while Graph Neural Network (GNN) have become popular for structured data. Researchers have also shifted their focus from the 2D to the 3D domain, and they experimented with different ways of organising the 3D data input to these learning models. Depending on the context, different geometric representations can be considered: 2D projections, point clouds, voxels, meshes and B-Rep models are among the most commonly adopted ones (Fig. 3). The choice of the representation used as input generally determines the choice of the learning architecture to be implemented. Multi-view convolutional neural network (MVCNN) tries converting 3D data into 2D images by projection and applies 2D feature extractors for 3D shape classification. To deal with the problem that the projection loses spatial depth information, SnapNet [34] and SnapNet-R [35] use a virtual camera to generate RGB photos and depth-maps. However, these methods still suffer from a loss of information due to dimensional compression and fine grain changes. Regarding point clouds, Guo et al. [36] reviewed and compared the state-of-the-art methods on deep learning for 3D point clouds. PointNet [37] is a pioneering work in applying machine learning on point clouds, while using a max-pooling symmetric function to solve the problem of disordered point clouds. Addressing its limitation in local feature extraction, PointNet has then undergone enhancements, including the implementation of a multi-layer sampling and grouping technique in PointNet++ [38], along with various other improvements [39, 40]. Regarding voxels, the voxel grid representation is linked to the pixel representation in the 2D domain, allowing it to leverage models similar to those used in 2D. VV-Net [41] is based on an interpolation variational autoencoder (VAE) and proposed to address the problem of missing local geometry within voxels. First, the distribution features of points are obtained for each voxel, and then the distribution of points within each voxel is mapped to the latent space using VAE. But it uses a dense representation that consumes substantial memory. To address this issue, SparseNet introduced sparse convolution as a solution [42]. Overall, volumetric representations preserve the local structural features, however, voxelisation inherently leads to discretisation and a loss of information. High resolution means high memory and computing costs, while low resolution leads to loss of details. Therefore, it's essential to strike a well-balanced trade-off between computational costs and the level of geometric detail. Regarding meshes, SubdivNet [43], a recent work on 3D mesh representations, has defined a general mesh convolutional operation and shown that it can achieve excellent results for 3D tasks. The work is groundbreaking and other applications have yet to be explored. Besides, with regard to the direct use of CAD models, B-Rep representations are increasingly considered as inputs of neural networks for a variety 3D tasks. DeepCAD proposes a deep generative network to extract the core features of the B-Rep model and output a sequence of operations used in CAD modeler to reconstruct a 3D shape [44]. BRepNet [45] is also able to work directly on B-Rep data structures by means of a specific neural network architecture that encodes the boundary curves and surfaces. SolidGen is an autoregressive neural network that models the B-Rep directly by predicting the vertices, edges, and faces using transformer-based and pointer neural networks [46].

As a conclusion, machine learning-based approaches are good candidates to meet the challenge of scan coverage prediction considered in this paper, and for which there are strong correlations between the influencing parameters. The way it has been done is developed in the next section.

3. Machine learning-based coverage prediction framework

This section introduces the components of the machine learning-based coverage prediction framework, including the pre-processing and prediction steps. The way in which the database used for training has been built is explained in the results section.

3.1. Overall framework

The proposed machine learning-based 3D scan coverage prediction framework is illustrated on Figure 4. It is composed of both pre-processing and prediction steps capable of predicting how an object, represented by its input CAD model, can actually be covered by the point cloud acquired from an input scan configuration, and this prior to its real scan.

During the pre-processing steps, the CAD model undergoes a first module that extracts the theoretically visible points from a given scan configuration. Points are defined from a mesh of the CAD model, and only those theoretically visible in the

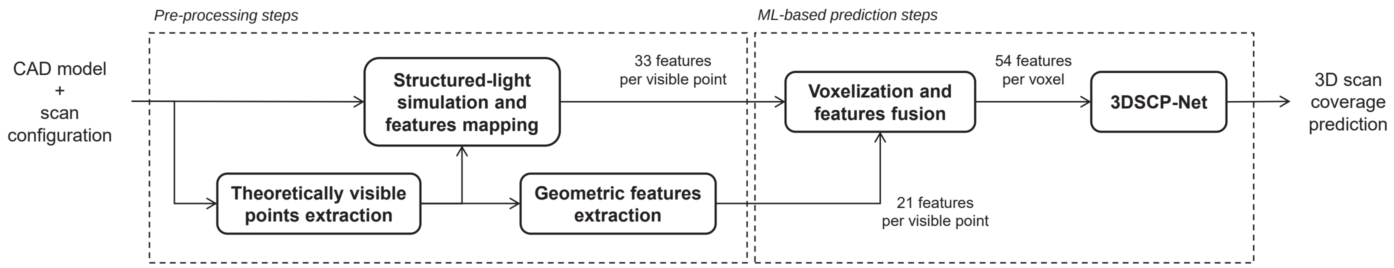
Pre-processing steps
ML-based prediction steps
CAD model
+
scan
configuration
Structured-light simulation and features mapping
33 features per visible point
Theoretically visible points extraction
Geometric features extraction
Voxelization and features fusion
54 features per voxel
21 features per visible point
3DSCP-Net
3D scan coverage prediction

Figure 4: Overall machine learning-based 3D scan coverage prediction framework, composed of pre-processing and ML-based prediction steps, and capable of predicting a priori (i.e. before doing the scan) the areas of an object that will actually be covered by the point cloud acquired from a given scan configuration.

field of view (FOV) of the acquisition device are retained (Section 3.2). In order to take into account local geometric configurations that can influence the inter-reflections of the emitted light patterns, geometric features are extracted on one side (Section 3.3). And, on the other side, the way the structured-light is projected onto the object is simulated and the features obtained from the generated images are mapped to the set of theoretically visible points (Section 3.4). The cloud of theoretically visible points, thus enriched with information characterising the scan configuration, then enters the voxelisation step that is used to prepare the use of convolutional kernels. During the voxelisation process, the features associated with the many points are merged and mapped to the voxels in which points are found (Section 3.5). Spare voxel representation is used considering the large amount of empty voxels. This enriched representation then enter 3DSCP-Net, the newly developed 3D scan coverage prediction model, that is trained to output the real coverage associated to the input CAD model and scan configuration (Section 3.6). In fine, the predicted coverage allows filtering the point cloud of theoretically visible points to retain only the ones that will actually be acquired by the device during the real scan.

The deployment of this complete framework is performed in two successive stages:

> *Training stage.* First, 3DSCP-Net needs to be trained on many samples characterising widely varying scan configurations, for which a real acquisition has been performed and for which a real coverage has been computed. Those real acquisitions and corresponding coverage will also be used as a ground truth to assess the performance of the model in accurately predicting the coverage on unknown scan configurations. The way in which the coverage is computed from a real acquisition and used to label each of the extracted theoretically visible points is explained in Section 3.2, while how the labelled database was built and populated with many scans is explained in the results section.

> *Exploitation stage.* Then, the complete framework, which includes the trained and fine-tuned 3DSCP-Net, can be used to predict the coverage of an object from an unknown scan configuration. As this is performed a priori, i.e. before scanning, there is no need to input a point cloud in this stage, and the prediction is performed on the basis of both an input CAD model and scan configuration as depicted in Figure 4. 3DSCP-Net can be used on its own, or inserted within an optimisation loop and called up several times to identify best scan configurations when solving the VPP, for example.

The way this framework has been validated and the obtained results are discussed in Section 4.

3.2. Extraction and labelling of theoretically visible points

This is the very first module of the pre-processing steps, which consists in identifying the areas of a CAD model $\mathcal{M}_i$ that are theoretically visible for a given scan configuration C_j, and extracting the first batch of associated features. During the training stage, an additional labelling step is carried out.

Adopted network-friendly formalism. As the coverage prediction model needs to be trained using real point clouds obtained from many scan configurations, the CAD model $\mathcal{M}_i$ is first transformed into a point cloud formalism suitable for learning. Indeed, point clouds are particularly well adapted to being processed through an architecture based on convolutional neural networks (CNN) which are capable of extracting features at multiple levels. Thus, the CAD model is first meshed by means of a Delaunay triangulation, with ϵ_ℓ as the edge length, and ϵ_c as the chord error, both of which depend on the characteristics of the acquisition device and scan configuration C_j (see Section 4 for more details on the technology adopted in this paper). The resulting mesh serves as an intermediate representation used to characterise the coverage of the CAD model by means of its connected triangles. Then, barycentre points and normals are extracted from each triangle of the mesh thus generating a raw point cloud. These transformations are illustrated on the example of Figure 5.

Extraction of theoretically visible points. Once all points have been obtained, a filtering allows only the theoretically visible points to be retained. This is necessary to speed up both training and prediction in the areas known to be clearly invisible to the acquisition device, mainly due to the occlusion phenomenon. These points could also have been labelled as uncovered during the labelling process, but this would have created unbalance dataset where there would have been many more uncovered than covered areas. The filtering is performed in two successive steps (top row of Fig. 5):

> Use of the hidden point removal (HPR) algorithm [47] with camera position computed from the scan configuration C_j, and with a search radius R_{HPR} computed with a formula obtained from [48];

> Use of the field of view (FOV) of the scanner, obtained from the scan configuration C_j, to retain only the points located within the field of view, which has previously been scaled by the factor S_{FOV} to take account of positioning errors.

In the end, given both a CAD model $\mathcal{V}_i$ and a scan configuration C_j, the remaining points are appended to the list $\mathcal{V}_{i,j}$ of theoretically visible points. At this stage, each point $\boldsymbol{p}_k$ in the list is characterised by six scalar values, i.e. its 3D coordinates

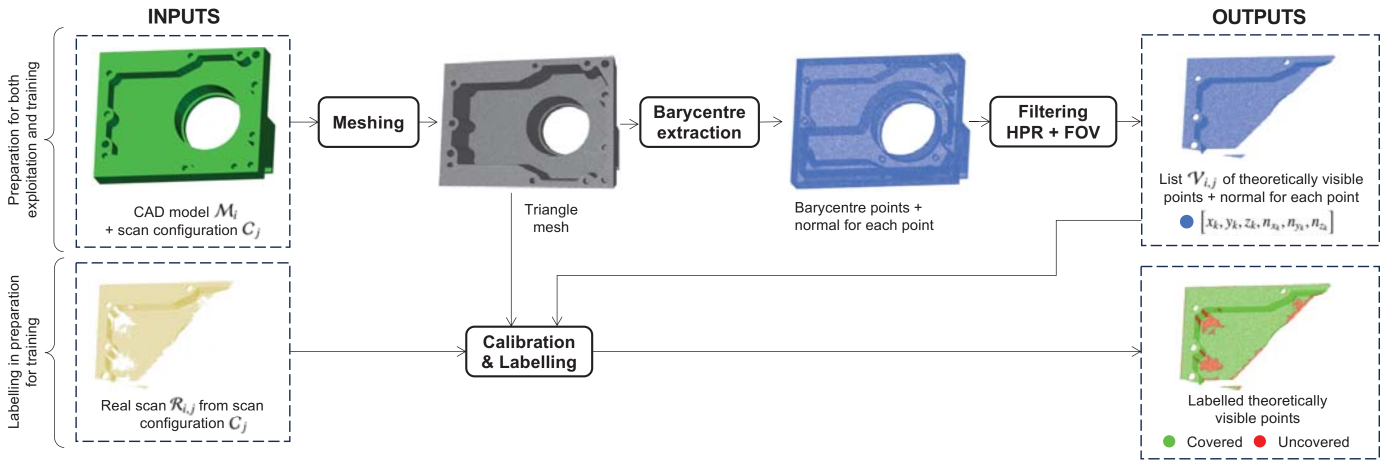
INPUTS
OUTPUTS
Preparation for both exploitation and training
CAD model $\mathcal{M}_i$ + scan configuration C_j
Meshing
Triangle mesh
Barycentre extraction
Barycentre points + normal for each point
Filtering HPR + FOV
List $\mathcal{V}_{i,j}$ of theoretically visible points + normal for each point
$[x_k, y_k, z_k, n_{x_k}, n_{y_k}, n_{z_k}]$
Labelling in preparation for training
Real scan $\mathcal{R}_{i,j}$ from scan configuration C_j
Calibration & Labelling
Labelled theoretically visible points
Covered
Uncovered

Figure 5: Details of the first pre-processing steps to extract the list $\mathcal{V}_{i,j}$ of theoretically visible points and normals from a CAD model $\mathcal{M}_i$ and scan configuration C_j (top row). When preparing the training, the label of each points is also computed from a real scan $\mathcal{R}_{i,j}$ obtained from the same scan configuration (bottom row).

(x_k, y_k, z_k) and its normal $(n_{x_k}, n_{y_k}, n_{z_k})$, forming a first feature vector:

$$\boldsymbol{F}^k_{point} = \left[x_k, y_k, z_k, n_{x_k}, n_{y_k}, n_{z_k}\right] \tag{1}$$

The next two subsections detail how additional features need to be computed to further characterise each point and enable the network to accurately predict whether the areas they represent are covered or not.

Labelling of theoretically visible points. This step is only necessary to prepare for training and is not followed when exploiting a pre-trained model to predict the scan coverage for unknown objects and/or configurations. Here, it is assumed that a real acquisition of the object represented by its CAD model $\mathcal{M}_i$ has been obtained from the scan configuration C_j, resulting in a list of real points $\mathcal{R}_{i,j}$ (bottom row of Fig. 5). The point cloud is then calibrated so as to fit to the mesh of the CAD model, using ICP algorithm as detailed in the result section. The coverage of the virtually generated point cloud $\mathcal{V}_{i,j}$ defers from the real one $\mathcal{R}_{i,j}$ and it is therefore needed to apply a mask to get the label c^k_{point} for each theoretically visible point:

$$c^k_{point} = \mathbb{I}\left[\exists \boldsymbol{p}'_k \in \mathcal{R}_{i,j} \mid d\left(\boldsymbol{p}'_k, T(\boldsymbol{p}_k)\right) < \epsilon_d\right], \quad \forall \boldsymbol{p}_k \in \mathcal{V}_{i,j} \tag{2}$$

where $\mathbb{I}[\cdot]$ is the indicator function, i.e. it is equal to 1 or 0 following the test, which means Covered or Uncovered in this context. Function $T(\cdot)$ returns the triangle whose barycentre is given as a parameter, function $d(\cdot,\cdot)$ computes the distance between a point and a triangle, and ϵ_d is the maximum distance beyond which a point is too far away and cannot be considered as covering the scanned object. This threshold allows artefacts and outliers to be ignored. Here, point-to-triangle distance has been preferred to point-to-point distance, as it allows a better characterisation of the coverage. Indeed, together with the adopted threshold ϵ_d, the former distance identifies points lying on the surface of the object, while the second distance would identify points located at a certain distance from other points on the surface, thus more as a distance measured onto the surface, which does not correspond to the notion of coverage adopted here [25].

3.3. Geometric features extraction

In order to give the network a good insight of the local geometric configurations that can influence the inter-reflections of the emitted light patterns, additional geometric features are extracted. These features depend on the relative positioning of the cameras and projector with respect to the acquired area. Figure 6 shows how they are computed for a theoretically visible point, with respect to the considered scan configuration C_j. For each camera/projector, 5 features are extracted: the norm of the vector between $\boldsymbol{p}_k$ and the centre of the camera/projector, the 3 coordinates of this vector, and the cosine of the angle between this vector and the normal to the point $\boldsymbol{n}_k$. Distance features are used to characterise the proximity of the projector/camera to the surface, and therefore the intensity with which the light is projected/acquired. Vectors are used to characterise the projection/acquisition directions, which is important while considering the inter-reflection issues. Cosines are used to take into account the normal to the theoretically visible point under consideration, which is also an important factor influencing inter-reflections. All these features affect the image quality according to the bidirectional reflectance distribution function (BRDF) in physically based rendering theory [49], which is also consistent with practical experience [1]. In the end, the 15 additional geometric features of a point that relates to the scan configuration are gathered together in another feature vector:

$$\begin{aligned}\boldsymbol{F}^k_{scan} = \Big[& \|\boldsymbol{D}_{R_k}\|, D_{R_{kx}}, D_{R_{ky}}, D_{R_{kz}}, \cos(\alpha_k), \\ & \|\boldsymbol{D}_{P_k}\|, D_{P_{kx}}, D_{P_{ky}}, D_{P_{kz}}, \cos(\beta_k), \\ & \|\boldsymbol{D}_{L_k}\|, D_{L_{kx}}, D_{L_{ky}}, D_{L_{kz}}, \cos(\gamma_k) \Big]\end{aligned} \tag{3}$$

3.4. Structured-light simulation and features mapping

This module aims at further enriching the list of features collected for each theoretically visible point, by extracting new ones from the results of structured-light projection simulations. The objective is to take real account, during a priori prediction, of the way light reflects when scanning a real object, in particular the inter-reflection and overexposure issues that can arise. To achieve this, three aspects need to be detailed: how to move from the 3D coordinates of the theoretically visible points to the pixel coordinates in the local reference frame of the projector/camera images, how structured-light patterns are defined, and how the projection of the pattern sequences can be simulated to get realistic physics-based images of what the cameras see. Features can

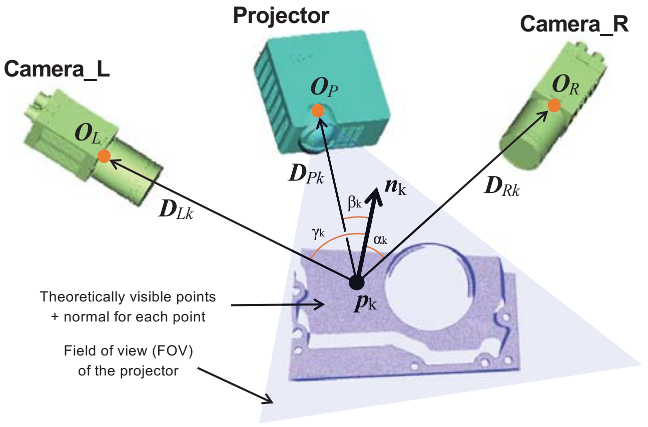

Figure 6: Extraction of the additional 15 geometric features characterising the local geometric configuration of each theoretically visible point, with respect to the considered scan configuration C_j.

then be inferred from the images resulting from the simulations and mapped to the theoretically visible points in 3D space.

It should be emphasised that these simulations are carried out on the whole object in order to take into account all possible inter-reflections, and not only those in the FOV. However, in the end, the extracted features are mapped to the only theoretically visible points obtained in the previous step. This is illustrated on Figure 7.

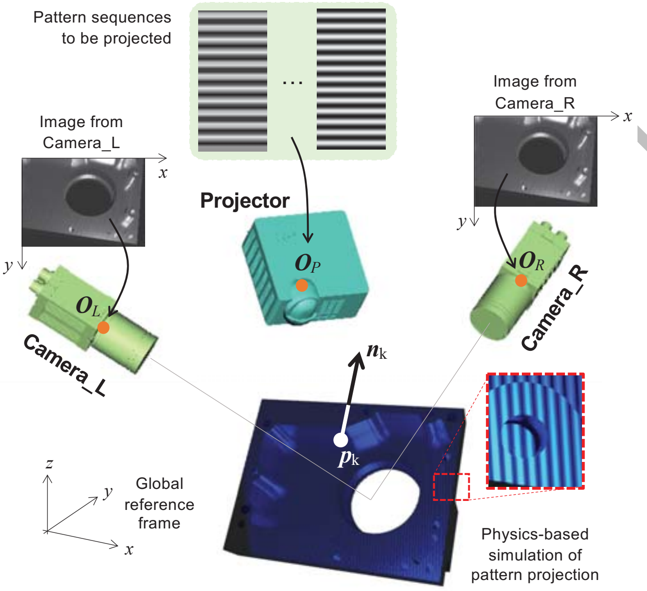

Figure 7: Structured-light physics-based simulation. Pattern sequences are projected onto the object to be scanned, and observed by two cameras capturing two images of the way patterns reflect. Each 3D point on the object has corresponding pixels in the cameras and projector reference frames.

From 3D coordinates to projector/camera images. In order to match the features extracted from the simulated realistic images with each of the theoretically visible point in 3D space, the relationships between the different spaces need to be established. This is obtained using the following projection equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{z_c} \cdot \mathbf{K} \cdot \mathbf{H} \cdot \boldsymbol{p} \tag{4}$$

where vector $\boldsymbol{p}$ contains the 4×1 homogeneous coordinates of a point in the global reference frame, and (x, y) the corresponding pixel coordinates in the reference frame of either the cameras or projector. $\mathbf{H}$ is a 4×4 matrix to transform the 3D coordinates into the camera/projector coordinate system. Here, there are three matrices of this type: $\mathbf{H}_L$, $\mathbf{H}_R$ and $\mathbf{H}_P$ for the left and right cameras and for the projector. $\mathbf{K}$ is a 3×4 intrinsic matrix, projecting 3D coordinates into 2D image coordinates. $\mathbf{K}_C$ is the same for the two cameras, and differs from $\mathbf{K}_P$ for the projector. These matrices are the result of a calibration step that follows Zhang's approach [50]. Finally, z_c is a scale factor, which is the depth value of a 3D point in the camera/projector reference frame, i.e. the third coordinate of the vector $\mathbf{H} \cdot \boldsymbol{p}$.

Definition of the structured-light patterns. The definition of the patterns is based on the multi-frequency heterodyne method [51, 52]. First, the intensity of a pixel (x, y) is defined by the following equation:

$$\begin{cases} I_{k,i}(x,y) = I_0 + I_b \cos\left(\varphi_k(x,y) + \delta_i\right) \\ \text{with } \varphi_k(x,y) = \dfrac{2\pi}{T_k} \times x, \; T_k = \dfrac{C}{f_k} \text{ and } \delta_i = \dfrac{2\pi(i-1)}{N_k} \end{cases} \tag{5}$$

where I_0 and I_b are respectively the mean and amplitude of the intensity, generally defined so that it evolves in the interval $[0, 255]$. However, here $I_0 = 125$ and $I_b = 115$, so that the intensity varies from 10 to 240 to prevent over-brightness or darkness due to material characteristics. $\varphi_k(x, y)$ is the phase principal, C is the number of columns of the image, T_k is the phase cycle determined by the frequency f_k, and N_k is the number of the phase shift steps for the frequency f_k. Figure 8 illustrates how sequences of patterns can be obtained by varying frequencies and steps. This intensity is constant along the y-axis.

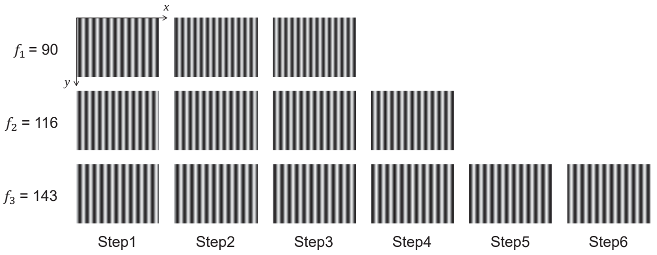

Figure 8: Examples of patterns generated with different frequencies and steps.

Following the four-step phase-shifting method, 3D point cloud can be reconstructed from 2D images [51]. During this process, additional phases $\Phi_k(x, y)$ are computed for the different frequencies f_k:

$$\Phi_k(x,y) = -\arctan\left(\frac{\sum_{i=1}^{N_k} I_{k,i}(x,y)\sin(\delta_i)}{\sum_{i=1}^{N_k} I_{k,i}(x,y)\cos(\delta_i)}\right) \tag{6}$$

where $I_{k,i}(x, y)$ corresponds to the intensity of the pixel (x, y) in the image generated with a frequency f_k and at step i. But more phases can be encoded while combining two frequencies f_k and $f_{k'}$. For instance, the frequency $f_{kk'} = f_{k'} - f_k$ gives rise to the following phase encoding:

$$\Phi_{kk'}(x,y) = \Phi_{k'}(x,y) - \Phi_k(x,y) \tag{7}$$

Recursively, with three frequencies f_k, $f_{k''}$ and $f_{k''}$, the frequency $f_{kk'k''} = f_{k'k''} - f_{kk'}$ gives rise to the following phase encoding:

$$\Phi_{kk'k''}(x,y) = \Phi_{k'k''}(x,y) - \Phi_{kk'}(x,y) \qquad (8)$$

Thanks to this encoding strategy, each column of pixels in the different images has a unique phase, and this is much important for the 3D reconstruction. In fact, the encoding can have countless combinations, and can even be combined with other methods, such as Moiré fringe encoding. In this paper, three frequencies f_k (with $k \in \{1,2,3\}$) have been adopted with common values of 90, 116 and 143, and with a number of steps N_k respectively equal to 3, 4 and 6, thus producing 13 patterns in the end (Fig. 8). This provides access to the recursively defined phase $\Phi_{123}(x,y)$, which will be used to extract features once the patterns have been projected.

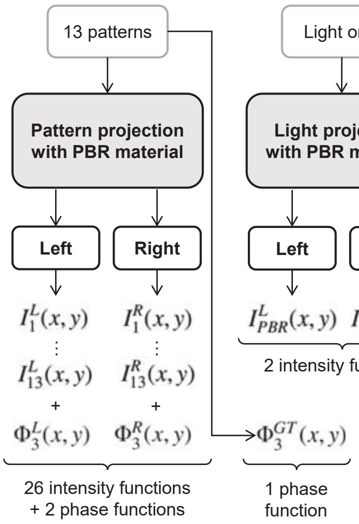
13 patterns
Light only
Light only
Pattern projection with PBR material
Light projection with PBR material
Light projection with HR material
Left
Right
Left
Right
Left
Right
$I_1^L(x,y)$ $I_1^R(x,y)$
$I_{13}^L(x,y)$ $I_{13}^R(x,y)$
+ +
$\Phi_3^L(x,y)$ $\Phi_3^R(x,y)$
$\Phi_3^{GT}(x,y)$
$I_{PBR}^L(x,y)$ $I_{PBR}^R(x,y)$
2 intensity functions
$I_{HR}^L(x,y)$ $I_{HR}^R(x,y)$
2 intensity functions
26 intensity functions + 2 phase functions
1 phase function

Figure 9: Rendering simulations organised in three different categories and producing 30 images from which a total of 30 intensity functions and 3 phase functions can be defined.

Pattern projection simulation and features mapping. The projection of the pattern sequences is simulated in Blender with Cycles as ray-trace based rendering engine [53, 54]. To do so, the 3D scene is configured in the same way as in reality, i.e. by applying all parameters characterising the scan configuration C_j to be simulated. The base colour of the different materials is the one of Unreal Engine [55]. Such a physics-based engine is particularly well suited to simulating the inter-reflections that occur. In order to maximise the extent to which real phenomena are taken into account, three types of simulation are carried out, each producing different images from which several features can be extracted (Fig. 9):

Pattern projection with PBR material. This simulation exploits the metallic PBR (Physically Based Rendering) workflow of Blender, that is based on the use of the Principled BSDF shader [56], to project the 13 patterns defined above (Fig. 8). For each pattern projected, 2 images are formed in the left and right cameras, thus generating a total of 26 projected images and as many intensity functions, $I_1^L(x,y)$ to $I_{13}^L(x,y)$ for the left camera, and $I_1^R(x,y)$ to $I_{13}^R(x,y)$ for the right one. In addition, the 2 phase functions $\Phi_{123}^L(x,y)$ and $\Phi_{123}^R(x,y)$ can be computed from the left and right images. Finally, the 13 initial patterns are also exploited directly (i.e. without projection) to define the phase function $\Phi_{123}^{GT}(x,y)$ that serves as a ground-truth.

Light projection with PBR material. This simulation projects directly the light source without using the patterns, and still with PBR material. The results can be seen in the 2 images formed in the left and right cameras, giving access to the intensity functions $I_{PBR}^L(x,y)$ and $I_{PBR}^R(x,y)$. These additional images are used to explore the light sensitivity of the geometry and material of the part being scanned.

Light projection with HR material. This simulation is similar to the previous one. It projects the light source, but this time with HR (High Reflection) material, giving access to the intensity functions $I_{HR}^L(x,y)$ and $I_{HR}^R(x,y)$. This material has a fairly low roughness value (i.e. 0.1 in our implementation), so specular reflection appears more easily. Such settings lead to overexposure and inter-reflection phenomena, in a similar way to what happens during a real scan, which is particularly interesting to take into account when predicting coverage a priori.

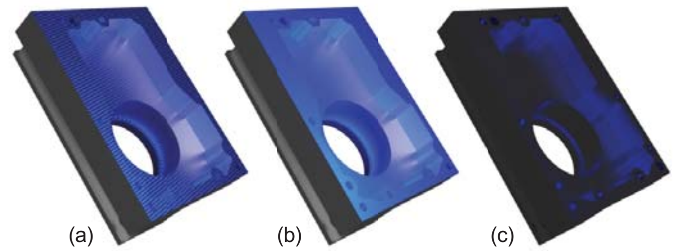

Figure 10: Examples of the three types of simulation using the Cycles engine in Blender: (a) pattern projection with PBR material, (b) light source projection with PBR material, (c) light source projection with HR material.

Figure 10 shows examples of the three types of simulation resulting from the Cycles engine in Blender. From all the 30 images generated, features can be extracted and mapped to each of the theoretically visible points whose coverage needs to be predicted. To do so, for each theoretically visible point $\boldsymbol{p}_k$, equation (4) is used to get the (x_k, y_k) coordinates of the matching pixels in the different images. Here, matrices change depending on whether the matching is to be performed with the projector or with the left/right cameras. Once the (x_k, y_k) coordinates obtained, the functions summarised in Figure 9 can be evaluated to obtain the complete list of 33 features for each theoretically visible points. To simplify notations, the pixel coordinates (x_k, y_k) corresponding to the point k are replaced by an index k in the following feature vector:

$$\begin{aligned} \boldsymbol{F}_{light}^k = \Big[& I_{k,1}^L, \cdots, I_{k,13}^L, I_{k,1}^R, \cdots, I_{k,13}^R, \Phi_{k,123}^L, \Phi_{k,123}^R, \\ & \Phi_{k,123}^{GT}, I_{k,PBR}^L, I_{k,HR}^L, I_{k,PBR}^R, I_{k,HR}^R \Big] \end{aligned} \qquad (9)$$

The intensity features vary from 0 to 255, and are obtained directly from the left and right images by reading the blue channel of the RGB colours. Phase features are in the interval $[0, 2\pi]$.

3.5. Voxelisation and features fusion

This is the final stage in the preparation for coverage prediction. It consists in voxelising all the theoretically visible points at a voxel size V_s, in order to obtain a sparse voxelisation ready for

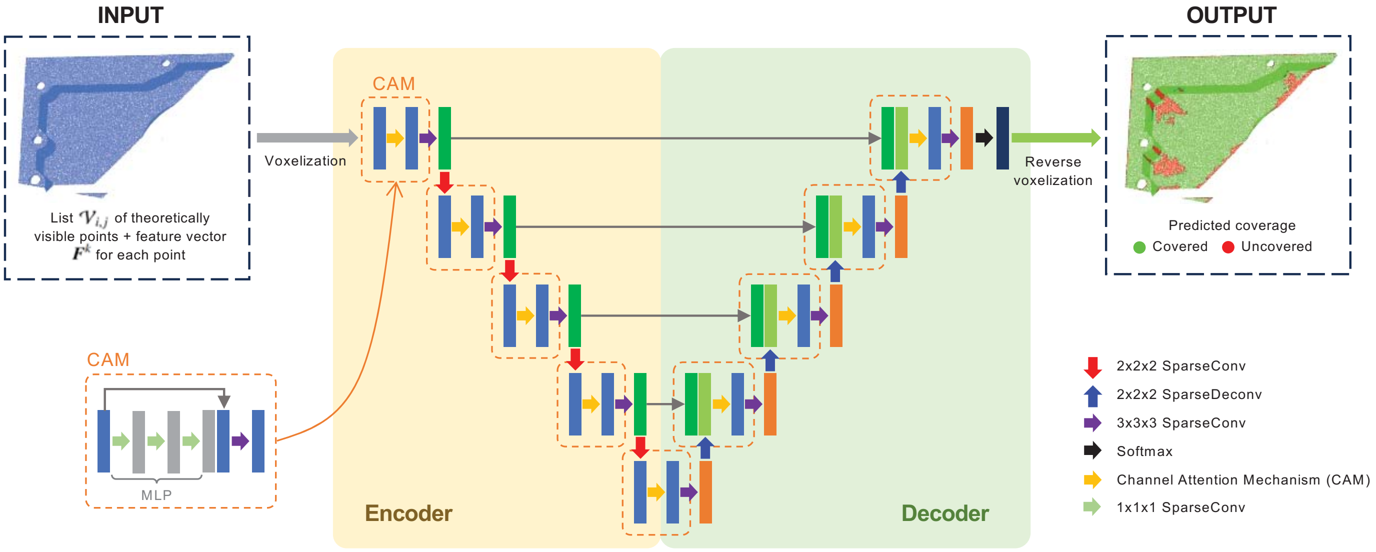

Figure 11: 3DSCP-Net model with a depth of 5 layers and skip connections between encoder and decoder, capable of predicting the coverage of each theoretically visible point, and exploiting a channel attention mechanism (CAM) to selectively emphasise informative features across different channels.

downstream convolutions. Each voxel contains a set of theoretically visible points, and each of these points is assigned a complete feature vector concatenating the three vectors computed in the previous steps:

$$\boldsymbol{F}^k = \boldsymbol{F}^k_{point} \oplus \boldsymbol{F}^k_{scan} \oplus \boldsymbol{F}^k_{light} \tag{10}$$

Features of points in a voxel are then averaged to come out with a single 1×54 feature vector per voxel, and normalised to get all of them in a range $[0, 1]$. For training, each voxel is also assigned a unique label corresponding to the most frequent label among the points included in it. In case of strict equality between the covered and uncovered points of a voxel, the covered label is preferred. Of course, no labels are assigned here during the exploitation stage, as it is up to the prediction model to predict the coverage for each of the theoretically visible points.

3.6. 3D Scan Coverage Prediction Network (3DSCP-Net)

The overall architecture of the coverage prediction model is similar to the one used in U-Net [57], but with several important differences to make 3DSCP-Net specifically dedicated and more efficient in the coverage prediction tasks. Indeed, such an architecture has been proven effective due to its ability to capture both local and global features, by exploiting a sparse representation and sparse convolution mechanisms. Good results have been achieved on semantic segmentation tasks, in particular for 3D reconstruction [58, 59]. Thus, the 3D scan coverage prediction network (3DSCP-Net) has been specifically designed and trained to meet the need for coverage prediction (Fig. 11). The input is the list of theoretically visible points for a given scan configuration, points that are embedded in a spare grid of voxels each containing a normalised 1×54 feature vector extracted during the pre-processing steps (Fig. 4). This is the first important difference with the classical U-Net architecture, for which the feature vectors are conventionally limited to the x, y and z coordinates of the points. The output of 3DSCP-Net is the predicted coverage of each theoretically visible point, obtained by predicting the coverage of each voxel in the grid in which the points lie. Differently from the classical architecture, the convolutional layers are here adapted to perform convolutions on all the features, and not only on 3D coordinates. Pooling operations are also replaced by convolution operations used to downsample and upsample the feature maps at each layer. Differently from the classical U-Net architecture, this helps in capturing more hierarchical features at different scales. Skip connections retain spatial information and gradients during training, enabling more precise prediction. More precisely, two components can be distinguished (Fig. 11):

Voxel-based multi-scale feature extractor consists in an encoder-decoder network trained end-to-end. All the convolutions refer to submanifold sparse convolutions [60, 61] with batch normalisation and ReLu activation function. For each layer in the encoder, features are first processed by the channel attention mechanism (CAM), and then sparsely convoluted by $3 \times 3 \times 3$ kernel without stride to get the output of the layer. The channel attention mechanism emphasises informative features across different channels, and this clearly helps improving model performance when compared to the classical U-Net architecture. Next, a $2 \times 2 \times 2$ convolution kernel with stride 2 is used for subsampling to get the new feature map as input of the next layer. This reduces the size of the feature map and extends feature receptive field [62]. In the decoder, the input of each layer consists of both the up-sampling result of the previous layer, as well as the output of the encoder obtained thanks to the skip connection at the same depth. Each layer in the decoder does like the encoder, applying the channel attention mechanism and submanifold sparse convolution with $3 \times 3 \times 3$ kernel without stride. The up-sampling is lastly done by $2 \times 2 \times 2$ deconvolution kernel to prepare the input for the next layer. The number of layer is set to D, and for each layer the layout of the convolution is $[m, 2m, 4m, \cdots, 2Dm]$ with $m = 80$.

Channel attention mechanism (CAM) has been designed to selectively emphasise informative features across different channels, enhancing the capabilities of the traditional U-Net architecture. It aims to improve model performance by dynamically recalibrating the importance of channels based

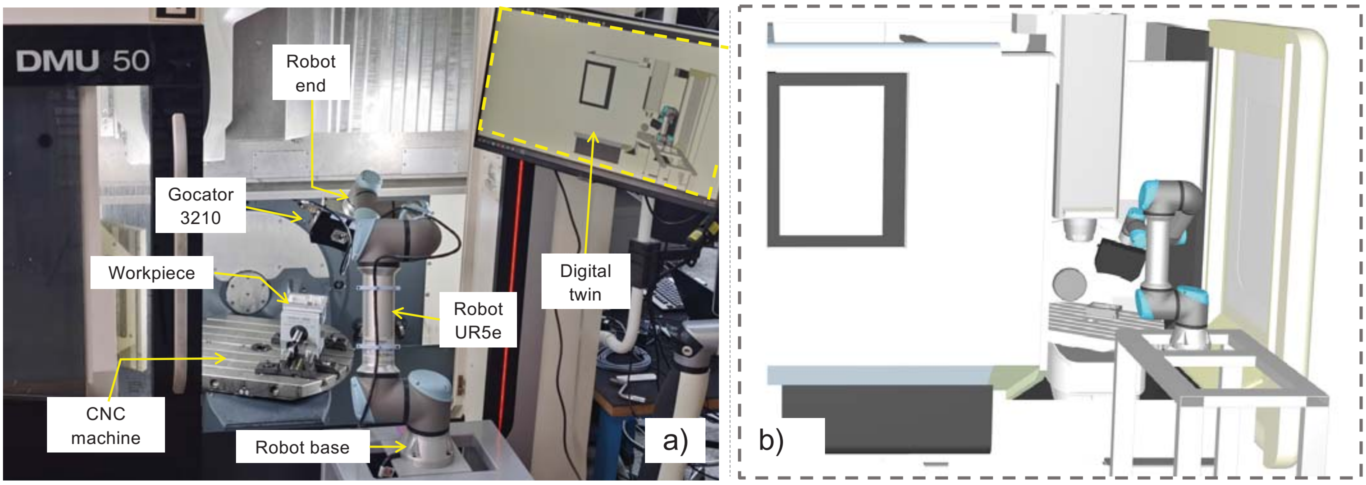

Figure 12: Acquisition platform (a) for automatically collecting and labelling numerous scan data obtained from widely varying scan configurations, and its digital twin (b). Once 3DSCP-Net trained, the digital twin can then predict the coverage for unknown workpieces and/or scan configurations for first-time-right control and smart-manufacturing applications.

on their relevance to the task at hand. The input features pass through a ResBlock made up of a MLP block and concatenation of channels. The MLP block consists of several $1 \times 1 \times 1$ convolution kernels with BatchNorm and ReLu. Then, $3 \times 3 \times 3$ sparse convolutions are applied on the concatenated features. This module is used at each layer, and for both the encoding and decoding phases.

In the end, once trained end-to-end using the cross-entropy loss function, 3DSCP-Net predicts the coverage of a new cloud of theoretically visible points, while predicting the class (covered or not) of the voxels that embed it. As a result, all the points belonging to a voxel are assigned the predicted status of that voxel. Back to Figure 4, the complete framework allows predicting the coverage status of each point in the list $\mathcal{V}_{i,j}$ of theoretically visible points generated from a CAD model $\mathcal{M}_i$ and scan configuration C_j. Hyper-parameters include voxel edge size V_s and depth D of the model, both of which are fine-tuned in Section 4 where an ablation study was also carried out.

4. Results and discussion

This section discusses the results obtained from using the newly developed 3DSCP-Net model to predict a priori, i.e. before scanning it, the coverage of an unknown part $\mathcal{M}_i$ scanned from a given scan configuration C_j. The first section presents the platform set up to automate and record the acquisition of numerous point clouds from different scan configurations. The next section details how this platform was exploited to build the database of automatically labelled point clouds used for training the prediction model. Then, the results of various experiments are presented and discussed in order to fine-tune the prediction model and validate the proposed approach.

4.1. Acquisition platform for collecting real scan data

One of the challenges of implementing such a machine learning approach is the access to a labelled database. To save time and avoid the tedious and error-prone labelling of numerous point clouds, an ad hoc acquisition platform has been set up. The idea is to use a robot to automate the acquisition from numerous positions and orientations of a scanner, while also varying the values of the influencing parameters. Figure 12 presents the acquisition platform composed of a CNC machine DMU50 by DMG MORI, a UR5e robot by Universal Robots, a Gocator 3210 structured-light scanner by LMI installed at the end of the robot. A luminometer and thermometer are also used to control the environmental conditions. The physical platform comes up with its digital twin counterpart, and the whole set up allows the automatic acquisition and labelling of numerous point clouds from widely varying scan configurations C_j. This platform will also be used to validate the proposed approach, and in particular to enable first-time-right control of a workpiece in its manufacturing environment for smart-manufacturing applications.

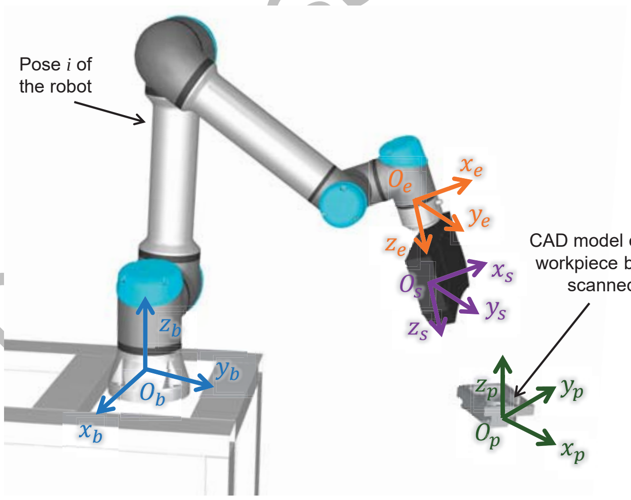

Figure 13: Coordinate systems associated to the acquisition platform and used to locate the point clouds, acquired in the reference frame of the Gocator 3210, onto the CAD model of the workpiece being scanned (DMU50 is not visualised).

As explained in Section 3.2, the labelling of the theoretically visible points requires the acquired point cloud to be correctly located with respect to the CAD model of the workpiece being scanned. To this aim, four coordinate systems are introduced (Fig. 13): robot base (O_b, x_b, y_b, z_b), robot end (O_e, x_e, y_e, z_e), scanner (O_s, x_s, y_s, z_s), and workpiece (O_p, x_p, y_p, z_p) coordinate systems. Then, different transformation matrices are to be computed to be able to link acquired points with points on the workpiece. Firstly, the transformation $\boldsymbol{H}_{se}$ (i.e. scanner to robot end) can be obtained by eye-in-hand calibration [63]. This matrix does not change and remains constant when the robot's poses are varied. The transformation $\boldsymbol{H}_{eb}^{(i)}$ (i.e. robot end to robot base) can be directly obtained for each pose i of the robot. The transformation $\boldsymbol{H}_{ps}^{(i)}$ (i.e. workpiece to scanner) is obtained while performing an acquisition for a given pose $i = 0$ of the robot, and then doing an ICP registration between the obtained point cloud and CAD model. Once these three matrices obtained, the constant transformation matrix $\boldsymbol{H}_{pb}$ (i.e. workpiece to robot base) is obtained with the following equation:

$$\boldsymbol{H}_{pb} = \boldsymbol{H}_{eb}^{(0)} \cdot \boldsymbol{H}_{se} \cdot \boldsymbol{H}_{ps}^{(0)} \tag{11}$$

Finally, for each new pose i characterised by the transformation $\boldsymbol{H}_{eb}^{(i)}$ that is easily available from the robot, the registration matrix $\boldsymbol{H}_{ps}^{(i)}$ between point cloud and CAD model of the workpiece is calculated as follows:

$$\boldsymbol{H}_{ps}^{(i)} = \left(\boldsymbol{H}_{eb}^{(i)} \cdot \boldsymbol{H}_{se}\right)^{-1} \cdot \boldsymbol{H}_{pb} \tag{12}$$

This calibration method allows the acquired point cloud to be first coarsely located on the CAD model for any scan configuration C_j, and then fine-calibrated using the ICP algorithm between the point cloud and CAD model.

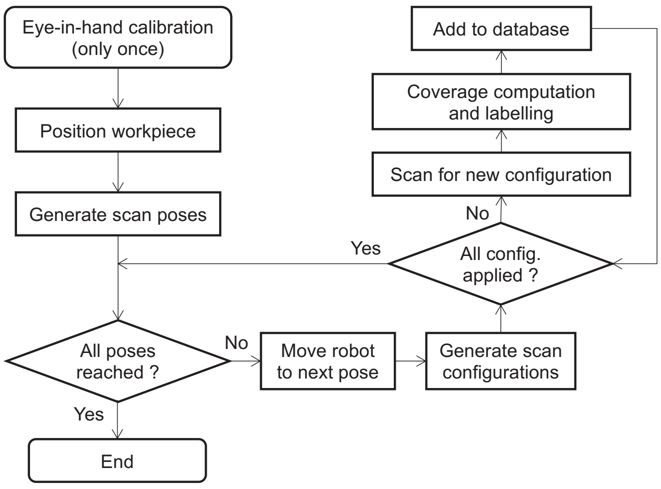

Figure 14: Flowchart of the acquisition protocol with two nested loops, one on the successive robot scan poses, and the other for the various scan configurations to be applied for each pose.

4.2. Data collection and labelling

The database used to train 3DSCP-Net was built following the acquisition protocol depicted on Figure 14. The very first step consists in eye-in-hand calibration [63] that is done only once, to compute the transformation matrices needed for coverage computation and labelling. Then, for a given workpiece, two nested loops are followed:

> *Loop on the scan poses* to be adopted by the robot and Gocator 3210 scanner. In this paper, a simple strategy was adopted to generate many poses of the robot end from a uniform sampling of a sphere cap centred on the barycentre of the workpiece and with a radius so that the centre of the sphere matches the middle of the scanner's FOV. Naturally, the digital twin is exploited to check each pose so as to avoid collision as the robot moves, and inaccessible poses are simply removed from the list. Figure 15 shows the scan poses generated for a given location of a part to be scanned.
>
> *Loop on the scan configurations* to be applied for each robot pose, while varying the scanner exposure time. For each scan, the acquired point cloud is then registered on the CAD model of the workpiece, using Eq. (12) and ICP to fine-tune the calibration, and the coverage is computed from Eq. (2). As a result, each theoretically visible point $\boldsymbol{p}_k$ of a scan configuration C_j is labelled as Covered (1) or Uncovered (0), and its feature vector is extracted following Eq. (10). Here, theoretically visible points result from a mesh of the CAD model generated with an edge length $\epsilon_\ell = 0.5$mm and a chord length $\epsilon_c = 5\%$, in accordance with the 0.08mm XY resolution of the scanner. These points are filtered using HPR algorithm with a radius resulting from [48, 47], and the Gocator's FOV scaled with a factor $S_{FOV} = 105\%$. The labelling is computed with $\epsilon_d = 0.3$mm to manage outliers (three-sigma rule). Each configuration is added to the database, with all of its labelled theoretically visible points and associated feature vectors.

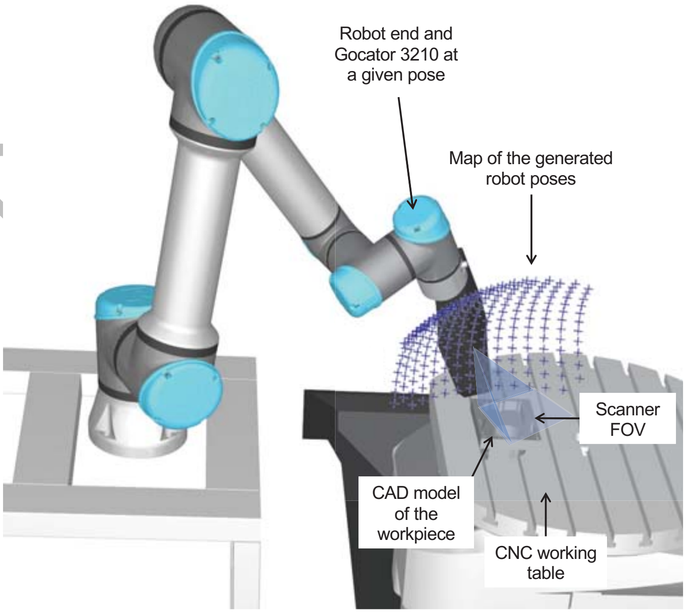
Robot end and Gocator 3210 at a given pose
Map of the generated robot poses
Scanner FOV
CAD model of the workpiece
CNC working table

Figure 15: Map of the scan poses generated from a sphere cap centred on the barycentre of the workpiece to be scanned.

For the experiments, four workpieces have been used, and correspond to the so-called 'Pocket', 'Stair-like', 'Car' and 'Cylinder-like' parts as illustrated on Figure 16. The first was used to train 3DSCP-Net, and the others to test it on an unknown workpiece. These parts were chosen because of their level of complexity, highlighting possible inter-reflection and overexposure issues when scanned with structured-light.

To set the exposure, the Gocator's automatic exposure mode is first used to determine the value considered to be close to ideal, from which a wider range is defined. At the time of the acquisitions, the light intensity was 3.45 Lux and the ambient temperature was 20.5°C, values measured with a luxometer and

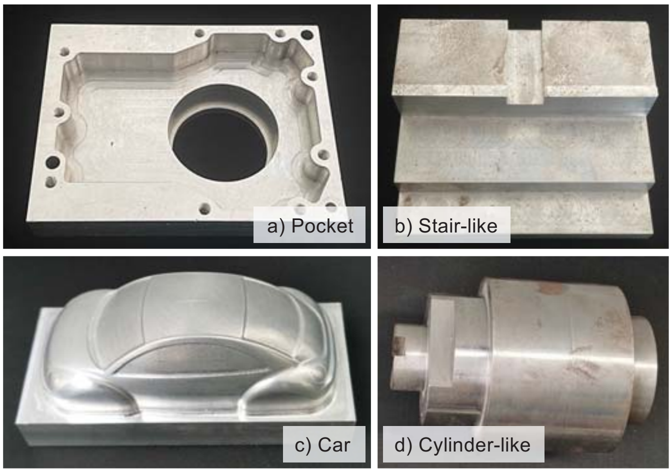

Figure 16: Workpieces used for training (a) and testing (b,c,d) 3DSCP-Net.

thermometer placed close to the workpieces. In the end, a total of 143 acquisitions were made for the Pocket, including 119 acquisitions for training and 24 for testing. In addition, multiple acquisitions have been performed on the Stair-like, Car and Cylinder-like to test 3DSCP-Net on parts never seen during the training. The complete dataset is 13.3GB and is made publicly available at URL: `https://doi.org/10.5281/zenodo.10807742`.

4.3. Evaluation metrics

To evaluate the performance of 3DSCP-Net, the conventional recall, precision and accuracy metrics are used for each scan configuration, taking into account the binary classification (Covered/Uncovered) of each theoretically visible point, and then averaged over the whole dataset with all scan configurations. The $F\beta$-score is then used to summarise precision and recall in a single weighted measure and facilitate the validation task:

$$F\beta\text{-score} = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{13}$$

When $\beta = 1$, this metric gives equal weight to precision and recall, and therefore to false positives and false negatives. However, considering coverage prediction, precision is more important as the areas labelled as Covered are used to optimise the resolution of the view planning problem [1]. Thus, more weight will be given to precision with $\beta = 1/\sqrt{2}$ for all experiments.

4.4. Experimental results and validation

The proposed approach has been implemented in Python language, and the networks are developed in Pytorch. The experiments were conducted with 32 GB NVIDIA Tesla V100-SXM2 GPU with CUDA. The training process uses the Adam optimiser with an initial learning rate of 10^{-4}, and the networks are trained with early stopping, and mini-batch size of 8 for all experiments. The training, validation and test sets are those presented above, so are the metrics used to validate the approach.

Various experiments were conducted to evaluate the performance of 3DSCP-Net in accurately predicting a priori the coverage of a part to be scanned, i.e. before scanning it. They are presented in the following paragraphs. First, several experiments were carried out to fine-tune some of the hyper-parameters, in particular the size V_s of the voxel grid, and the depth D of the multi-scale feature extractor. Then, an ablation study allows to evaluate the contribution of the different blocks, and compare the results to the ones obtained with the conventional U-Net architecture. Finally, the proposed approach is validated on unknown scan configurations, for which a sensitivity analysis is also proposed to study the impact of the robot positioning errors on the performance of 3DSCP-Net, and for first-time-right control in the context of VPP resolution.

Fine-tuning of the hyper-parameters. The first experiment consists in identifying the optimal size V_s of the voxel grid embedding the theoretically visible points whose coverage is to be predicted by 3DSCP-Net. It is important to remember that these points come from the mesh of the CAD model with an edge length of $\epsilon_\ell = 0.5$mm. Therefore, considering equilateral triangles, the theoretical distance between the barycentres of two adjacent triangles is $\epsilon_\ell \times \sin(\pi/3) \times 1/3 \times 2 \simeq 0.2887$mm. The various experiments carried out by varying the voxel size slightly around this theoretical value show that the value $V_s = 0.28$mm is the best when considering $F\beta$-score. This is visible from Table 1 obtained on the Stair-like dataset. In the end, this corresponds to having more or less one theoretically visible point per voxel, thus limiting the effect of averaging the feature vectors within a voxel.

Table 1: Experiment on voxel size (metrics averaged on Stair-like dataset)

V_s (mm)	Recall	Precision	Accuracy	$F\beta$-score
0.35	0.756	0.868	0.787	0.803
0.30	0.804	0.862	0.801	0.826
0.29	0.881	0.827	0.805	0.830
0.28	0.834	0.854	0.798	**0.835**
0.27	0.868	0.817	0.788	0.822

Similarly, different depths D of the model have been compared and the results obtained on the Stair-like dataset are highlighted in Table 2. These tests clearly show that the deepest networks are not necessarily the most effective. Indeed, each downsampling means that the receptive field of the convolution kernel is expanded by 2 times. Since the voxel size is $V_s = 0.28$mm, when depth $D = 6$ the size of the kernel's receptive field for the last layer is 4.48mm. This value seems too large, as the resolution of the scanner is 0.08mm. In conclusion, depth $D = 5$ is a good trade-off.

Table 2: Experiment on model depth (metrics averaged on Stair-like dataset)

Depth D	Recall	Precision	Accuracy	$F\beta$-score
4	0.638	0.794	0.678	0.737
5	0.834	0.854	0.798	**0.835**
6	0.645	0.794	0.683	0.734

Ablation study and comparison. The contribution of each block of 3DSCP-Net is then evaluated through an ablation study, with $V_s = 0.28$mm and $D = 5$ from the previous experiments. For this purpose, various characteristics of the model are grouped together, and the value of combining them is assessed on the Stair-like dataset. Group (1) corresponds to the use of all the geometric features $\boldsymbol{F}^k_{point} \oplus \boldsymbol{F}^k_{scan}$ associated with each theoretically visible point $\boldsymbol{p}_k$. Groups (2) to (4) refer to the use of more or less features in the feature vector $\boldsymbol{F}^k_{light}$ of Eq. (9). More precisely, Group (2) uses the first 26 intensity values, Group (3) the following 3 phase values, and Group (4) the last 4 intensity values with PBR and HR materials. Finally, Group (5) corresponds to the use of the channel attention mechanism (CAM).

Table 3: Results of the ablation study involving groups (1) to (5), and comparison with the conventional U-Net (metrics averaged on Stair-like dataset).

3DSCP-Net	Recall	Precision	Accuracy	$F\beta$-score
1	0.889	0.751	0.727	0.779
1+2	0.839	0.824	0.761	0.807
1+2+3	0.782	0.829	0.733	0.784
1+2+4	0.851	0.839	0.795	0.827
1+2+3+4	0.884	0.822	0.793	0.830
1+2+3+4+5	0.834	0.854	0.798	**0.835**
U-Net	0.531	0.524	0.644	0.486

The results of the ablation study involving groups (1) to (5) are available in Table 3, together with the results obtained with the conventional U-Net architecture that only rely on the 3D coordinates of the points. They clearly show that each block of 3DSCP-Net brings an improvement in terms of $F\beta$-score. Only when considering the geometric features (Group 1), the first 26 intensity values (Group 2) and following 3 phase values (Group 3), the $F\beta$-score decreases. But this reduction vanishes by adding the other groups, thus demonstrating the interest of all the components of the proposed machine learning-based 3D coverage prediction framework. Furthermore, the results clearly demonstrate that 3DSCP-Net performs much better than the conventional U-Net on coverage prediction tasks, thus validating the network architecture, the feature vectors selection as well as the use of the channel attention mechanism to selectively emphasise them.

Table 4: Extract of results obtained for unknown poses and scan configurations.

Part	Pose	Recall	Precision	Accuracy	$F\beta$-score
Stair-like	#1	0.855	0.731	0.770	0.768
	#2	0.966	0.935	0.932	0.946
	#3	0.790	0.886	0.800	0.852
	#4	0.729	0.830	0.709	0.793
Pocket	#1	0.946	0.939	0.935	0.940
	#2	0.933	0.930	0.923	0.931
	#3	0.877	0.929	0.901	0.911
	#4	0.918	0.884	0.880	0.895
Car	#1	0.878	0.947	0.877	0.899
	#2	0.929	0.914	0.873	0.919
	#3	0.855	0.902	0.878	0.886
	#4	0.817	0.961	0.801	0.848
Cylinder-like	#1	0.794	0.884	0.784	0.842
	#2	0.872	0.819	0.781	0.800
	#3	0.796	0.797	0.704	0.780
	#4	0.866	0.812	0.776	0.795

Validation on unknown scan configurations. The proposed approach was validated on the four workpieces, and according to viewpoints and scan configurations that had never been seen when training 3DSCP-Net. The results obtained for the Stair-like dataset are presented in Table 3. Metrics are averaged on the dataset composed of more than 180 poses. Overall, the prediction model successfully reached 83.5% on $F\beta$-score, which is pretty good considering that the tested scan configurations have never been seen. These experiments demonstrate that the model has good generalisation performance, and is able to accurately predict the coverage on unknown workpieces and scan configurations. Four poses of each workpiece are also presented in Figure 17 with metric values gathered in Table 4. Here again, results are very good, with highest values close to 95% on $F\beta$-score. Two of the unknown poses are depicted in Figure 18 for each workpiece. The comparison between the theoretical point cloud and the one predicted by 3DSCP-Net clearly shows that the prediction model performs better than a purely theoretical approach based only on the FOV. This is visible from the many points that the theoretical approach sees, and that are finally not acquired by the scanner. When comparing the predicted point clouds to the ground truth obtained by scanning the workpieces for real, most of the points belong to true-positive (TP in green, with both the ground truth and prediction as Covered) or true-negative (TN in blue, with both the ground truth and prediction as Uncovered). This is very good. Then, there are few false-negative (FN in yellow) for which the model predicts points as Uncovered although they appear as Covered in the ground truth. This is not so bad, as the model somehow underestimates what is really acquired in the end. Finally, there are also few false-positive (FP in red) for which the model predicts points as Covered although they are not in real, thus overestimating what the scanner can obtained in real conditions. This last case is certainly the worst when considering the overall objective of optimising the scanner

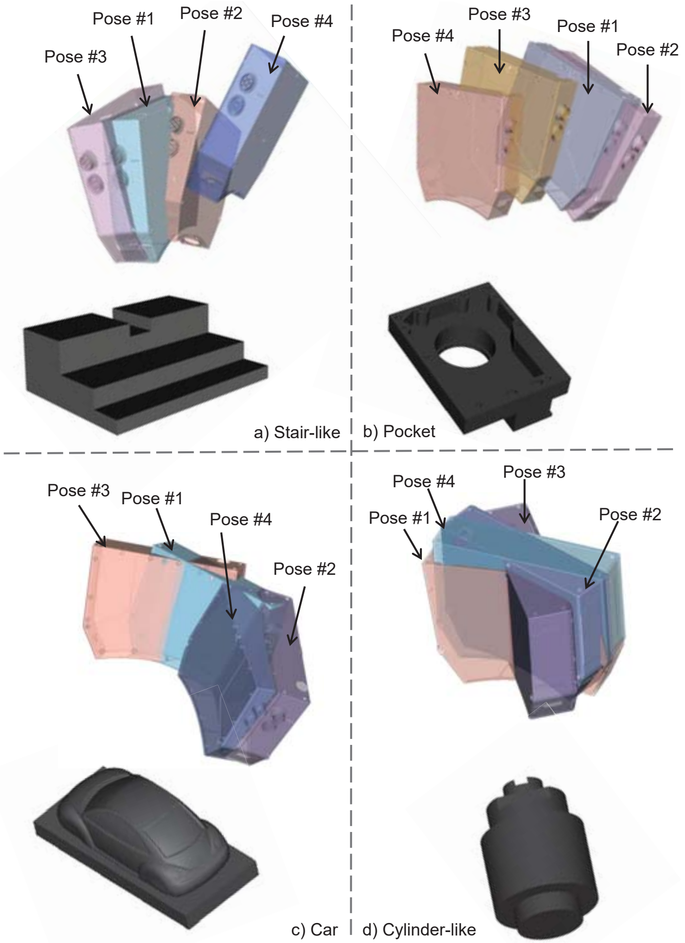

Figure 17: Poses used to validate 3DSCP-Net on unknown scan configurations for the Stair-like (a), Pocket (b), Car (c) and Cylinder-like (d) workpieces.

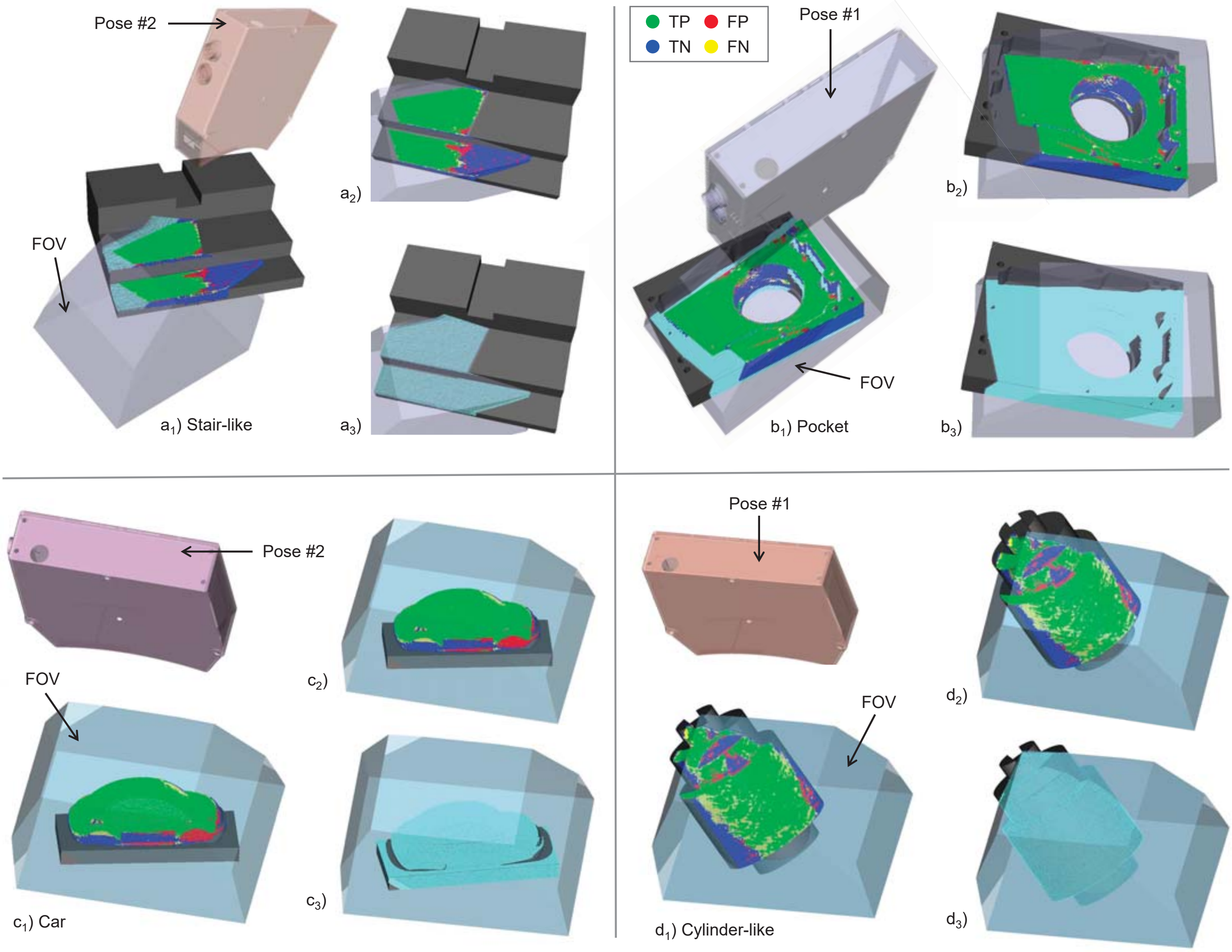

Figure 18: Examples obtained from Stair-like (a_1), Pocket (b_1), Car (c_1) and Cylinder-like (d_1) unknown poses: (a_3, b_3, c_3, d_3) are the theoretical point clouds obtained only using a filter based on the FOV of the Gocator, (a_2, b_2, c_2, d_2) are the comparisons between the coverage predicted by 3DSCP-Net and the coverage computed from the real acquisition, clearly highlighting that most of the points are well-predicted (TP in green and TN in blue) with a $F\beta$-score of respectively 94.6%, 94.0%, 91.9% and 84.2% for the four workpieces (Table 4).

position before scanning the manufactured part. Indeed, if the optimiser uses points predicted to be Covered and which are finally not after scanning, there is a risk of not obtaining a good scan in real. However, there are very few red dots, and it should be borne in mind that a check is generally based on several overlapping scans, whose deficits can compensate for each other. This is evaluated with the last experiment on first-time-right control.

Analysis of the prediction sensitivity. This analysis and associated experiments aim at characterising the impact of robot positioning errors on the performance of 3DSCP-Net in accurately predicting the coverage on unknown parts. This means assessing the model's sensitivity to positioning errors. The literature shows that after motion error compensation, the pose error of the robot's end-effector can obtain good quality [64, 65]. Thus, this analysis only focuses on the impact of the robot positioning error (i.e. only the error on the (x,y,z) coordinates of the end-effector) on the sensitivity of the predictions, and does not take into account pose error, which would also include rotations. To do so, starting from several reference poses (error of 0.0mm), various positioning errors are introduced (i.e. 0.1mm [66], 0.5mm [67] and 1.0mm [68, 69]) and used to randomly affect the position of the end-effector. Then, the coverage predicted by 3DSCP-Net for the perturbed positions are compared to the reference ones, and the results are summarised in Table 5. $RMSD$ refers to the root mean square deviation between the prediction results with the positioning errors introduced, and the reference ones. It reveals that the coverage prediction framework is not sensitive to small errors in the positioning of the robot's end-effector, with a max of 0.006 on $RMSD$, which is good.

Table 5: Analysis of the sensitivity of the prediction model to positioning errors of the robot's end-effector ranging from 0.1mm to 1.0mm on four reference poses (error 0.0mm) of the Stair-like workpiece.

Pose	Positioning error (mm)	0.0	0.1	0.5	1.0	$RMSD$
#1	Recall	0.643	0.646	0.637	0.638	0.005
	Precision	0.893	0.891	0.884	0.887	0.006
	Accuracy	0.774	0.774	0.768	0.769	0.004
	$F\beta$-score	0.791	0.791	0.783	0.785	0.004
#2	Recall	0.760	0.758	0.757	0.758	0.003
	Precision	0.976	0.974	0.972	0.966	0.006
	Accuracy	0.817	0.815	0.813	0.811	0.004
	$F\beta$-score	0.892	0.890	0.888	0.885	0.004
#3	Recall	0.839	0.840	0.841	0.841	0.002
	Precision	0.985	0.984	0.982	0.982	0.002
	Accuracy	0.835	0.836	0.836	0.836	0.001
	$F\beta$-score	0.931	0.931	0.930	0.930	0.000
#4	Recall	0.942	0.944	0.946	0.951	0.005
	Precision	0.937	0.938	0.934	0.933	0.003
	Accuracy	0.906	0.908	0.907	0.910	0.002
	$F\beta$-score	0.939	0.940	0.938	0.939	0.001

First-time-right control. This last experiment aims at evaluating how 3DSCP-Net can help better anticipating points that are not acquired for real when compared to the theoretically visible ones traditionally used to solve the VPP problem. Indeed, accurately predicting missing points in the acquired point clouds can avoid time-consuming additional acquisitions that would be needed to complete the point clouds and fully cover the region of interest (ROI). Figure 19 illustrates a scenario where several geometrical product specifications (GPS) need to be checked. To do this, the surfaces involved must be scanned, and the grouping of these surfaces defines the ROI used as input of the VPP resolution. The way the VPP can be solved is not part of this paper, and the reader can refer to the extended survey of Jubert et al. [1] for more details on the existing approaches. However, solutions to this problem have been explored manually, using solely the FOV to define the points theoretically visible from a viewpoint, and one of these solutions is illustrated by the set of 3 poses depicted on Figure 20. These 3 poses are therefore theoretically sufficient to cover the entire ROI. Unfortunately, and as explained all along this paper, the theoretically visible points will not be all acquired for each of the pose. This is visible from Figure 20 where the points predicted to be acquired (green) cover slightly smaller areas than the theoretically visible ones (cyan). As a consequence, when overlapping the three point clouds all together on Figure 21, it appears that even though the 3 poses have been manually chosen so as to cover perfectly the ROI (cyan), the points that will be acquired for real, and which have been predicted by 3DSCP-Net, cover much smaller areas (green). This clearly demonstrates that relying on FOV and simple visibility criteria, as many approaches do, does not allow an accurate resolution of the VPP, even manually, with a high risk of having to carry out additional and time-consuming unpredictable acquisitions from several other poses to complete the point clouds. On the contrary, solving the VPP using 3DSCP-Net to predict the areas that will be acquired for real allows more realistic and accurate predictions. This is because these predictions are based on a large number of previous acquisitions with a wide variety of scan configurations, and which have been used to train the network end-to-end.

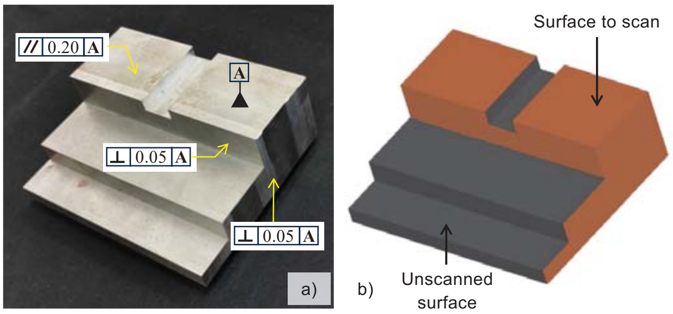

Figure 19: Identification of the region of interest (ROI) and surfaces to be scanned on the Stair-like part (b) in order to be able to control several geometrical product specifications (a).

As a conclusion, the results of these experiments validate the model and its capacity to predict a priori the coverage, i.e. before scanning. 3DSCP-Net clearly outperforms traditional approaches based on simple visibility criteria, taking into account local configurations such as inter-reflection issues and other complex phenomena occurring during real scans (e.g. occlusion, overexposure). It has a good ability to fit the learned dataset, and a good generalisation ability to unseen parts $\mathcal{M}_i$ and/or scan configurations C_j. This model can thus be used to better plan the control of manufactured parts, ushering in a new era of first-time-right smart control.

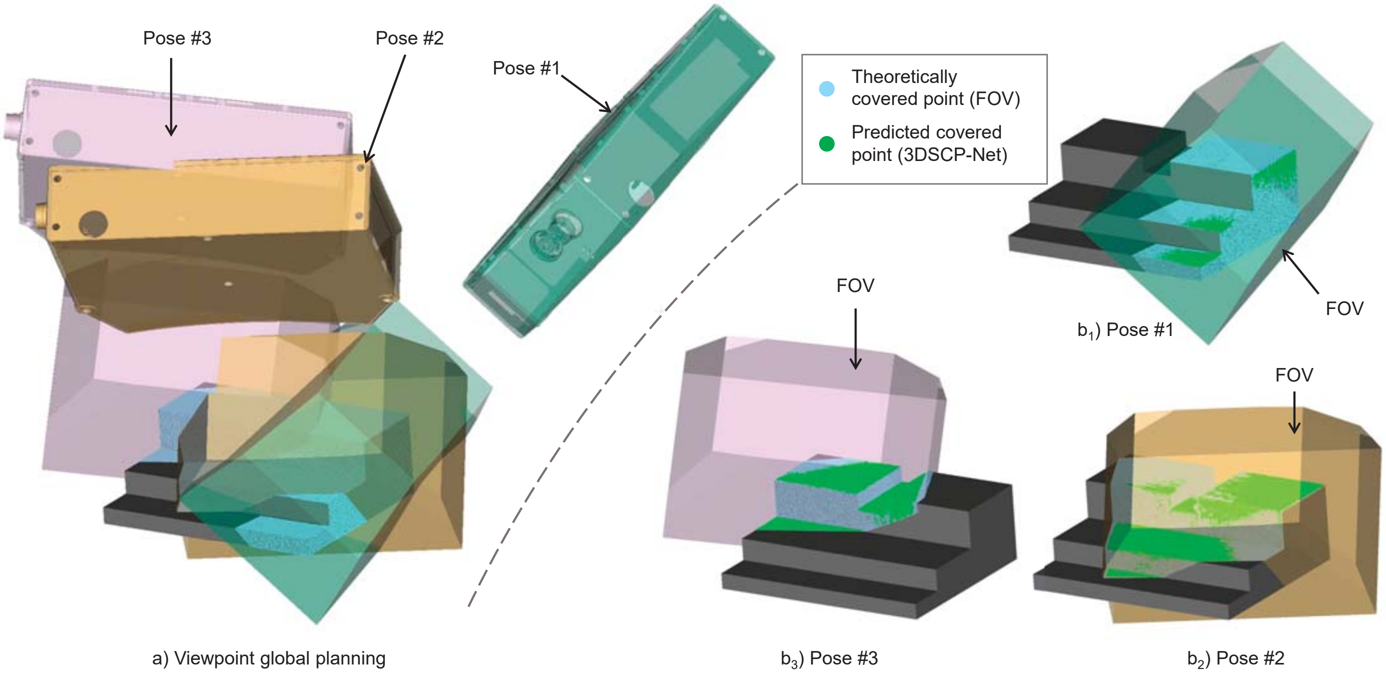
Pose #3
Pose #2
Pose #1
Theoretically covered point (FOV)
Predicted covered point (3DSCP-Net)
FOV
b_1) Pose #1
FOV
FOV
a) Viewpoint global planning
b_3) Pose #3
b_2) Pose #2

Figure 20: Results of the simulation of several acquisitions obtained from 3 poses (a) considered as one of the solution to the VPP defined from the control scenario of Figure 19 and manually located using the FOV and corresponding theoretical visibility criterion. For each of the 3 poses (b_1 to b_3), the coverage is predicted using either the FOV (cyan points), or using 3DSCP-Net to predict the real coverage (green points).

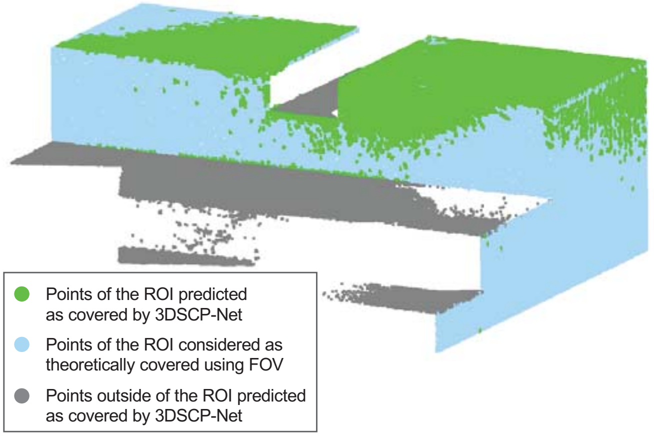
● Points of the ROI predicted as covered by 3DSCP-Net
● Points of the ROI considered as theoretically covered using FOV
● Points outside of the ROI predicted as covered by 3DSCP-Net

Figure 21: Comparison between the overlap of the areas theoretically covered by the 3 poses of Figure 20 (cyan and green points), and the ones predicted as covered by 3DSCP-Net (green points) with a focus on the ROI, except for 3DSCP-Net for which points predicted as covered outside of the ROI are also displayed.

5. Conclusion and future works

This paper introduced a new machine learning-based framework capable of accurately and a priori predicting the scan coverage of a workpiece from a given scan configuration. Such a model is particularly useful for optimising VPP resolution and optimising poses and scan configurations before scanning. The approach exploits an architecture capable of extracting features at different levels. Features relate to the geometry, to the acquisition devices as well as from simulating the projection of patterns in structured-light acquisition. These last features are capable of incorporating inter-reflection and overexposure issues during the prediction. The database used for training has been built thanks to the development of an acquisition platform capable of automatically acquiring and labelling point clouds from widely varying scan configurations. Results demonstrated the performance of the approach, which outperforms what a theoretical approach based on simple visibility criteria can do.

It is important to stress that the method developed in this paper is generic, even though 3DSCP-Net was trained using point clouds obtained from a specific scanner, namely the GOCATOR 3210, which was used in this work. The approach remains valid for any other binocular structured-light-based scanners (i.e. GOCATOR 3520, GOM Scan 1, GOM ATOS Q, GOM ATOS Compact Scan, HEXAGON PrimeScan). Thus, if a new scanner is to be used, the model must be trained again, following the methodology developed in this paper, but using a new database of point clouds obtained from this new scanner. How 3DSCP-Net could directly adapt to other binocular structured-light-based scanners needs to be be studied in more detail. This would need training on a larger database obtained from various scanners, and using one more feature as input of the network to characterise which scanner has been used for a particular sample. In this way, 3DSCP-Net could directly estimate the coverage obtained from various scanners.

However, there is still room for other improvements. First, although the proposed approach already integrates important factors related to the light and surface roughness when simulating the projection of patterns with structured-light, the addition of new features related to the machining should be investigated. For example, the depth of cut, the tool rotation speed, but also the tool wear are all parameters that can influence the quality of the surface of the part being manufactured, and therefore the quality of the scan. Taking these additional features into account requires the database to be populated with many other acquisitions obtained from even more varied manufacturing and scan configurations. The proposed acquisition platform is ready for this and, thanks to the digital twin developed, it can easily be used to carry out on-line acquisitions directly within a machining centre.. Second, smart-control applications need to be further developed, and in particular the integration and use of the proposed prediction model when solving the VPP. This would definitively ease decision-making by relying on information that the system can actually obtained, rather than on theoretical information that do not take into account real working conditions, thus limiting the risks for not being able to aggregate the scans obtained from the different viewpoints. Finally, with the growth of developments in machine learning, there is a need for an increasing number of labelled datasets, which can be difficult to obtain in a reasonable time-frame and with sufficient quality. The proposed approach can definitively contribute to this task, while automatically generating large datasets of as-scanned point clouds, incorporating artefacts that would appear with real acquisition devices, and in particular by accurately predicting the areas covered.

References

[1] M. Peuzin-Jubert, A. Polette, D. Nozais, J. L. Mari, J. P. Pernot, Survey on the view planning problem for reverse engineering and automated control applications, Computer-Aided Design 141 (2021) 103094. doi:10.1016/J.CAD.2021.103094.

[2] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches (2016) 85–113doi:10.1007/978-3-319-38756-7_4.

[3] P. Kaťuch, M. Dovica, S. Slosarčʼk, J. Kováč, Comparision of contact and contactless measuring methods for form evaluation, Procedia Engineering 48 (2012) 273–279, modelling of Mechanical and Mechatronics Systems. doi:10.1016/j.proeng.2012.09.514.

[4] S. Zhang, High-speed 3d shape measurement with structured light methods: A review, Optics and Lasers in Engineering 106 (2018) 119–131. doi:10.1016/j.optlaseng.2018.02.017.

[5] S. Carmignato, A. Voltan, E. Savio, Metrological performance of optical coordinate measuring machines under industrial conditions, CIRP Annals 59 (1) (2010) 497–500. doi:10.1016/j.cirp.2010.03.128.

[6] G. Moroni, S. Petrò, Coordinate Measuring Machine Measurement Planning, Springer London, London, 2011, pp. 111–158. doi:10.1007/978-1-84996-311-4_4.

[7] Y. Li, J. Ibanez-Guzman, Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems, IEEE Signal Processing Magazine 37 (4) (2020) 50–61. doi:10.1109/MSP.2020.2973615.

[8] Y. Wang, F. Xie, S. Ma, L. Dong, Review of surface profile measurement techniques based on optical interferometry, Optics and Lasers in Engineering 93 (2017) 164–170. doi:10.1016/j.optlaseng.2017.02.004.

[9] A. A. Tarvo Mill, R. Liias, Combined 3d building surveying techniques – terrestrial laser scanning (tls) and total station surveying for bim data management purposes, Journal of Civil Engineering and Management 19 (sup1) (2013) S23–S32. doi:10.3846/13923730.2013.795187.

[10] J. Li, J. Berglund, F. Auris, A. Hanna, J. Vallhagen, K. Åkesson, Evaluation of photogrammetry for use in industrial production systems, in: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), 2018, pp. 414–420. doi:10.1109/COASE.2018.8560496.

[11] S. Giancola, M. Valenti, R. Sala, A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies, Springer Briefs in Computer Science, Springer, 2018. doi:10.1007/978-3-319-91761-0.

[12] J. Eastwood, H. Zhang, M. Isa, D. Sims-Waterhouse, R. K. Leach, S. Piano, Smart photogrammetry for three-dimensional shape measurement, https://doi.org/10.1117/12.2556462 11352 (2020) 43–52. doi:10.1117/12.2556462.

[13] S. Catalucci, N. Senin, D. Sims-Waterhouse, S. Ziegelmeier, S. Piano, R. Leach, Measurement of complex freeform additively manufactured parts by structured light and photogrammetry, Measurement 164 (2020) 108081. doi:10.1016/J.MEASUREMENT.2020.108081.

[14] W. R. Scott, G. Roth, J.-F. Rivest, View planning as a set covering problem, NRC Publ Arch 2001.

[15] R. M. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations (1972) 85–103doi:10.1007/978-1-4684-2001-2_9.

[16] W. R. Scott, Model-based view planning, Machine Vision and Applications 20 (2009) 47–69. doi:10.1007/S00138-007-0110-2/METRICS.

[17] K. Shimadat, D. C. Gossard, Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing, in: SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications, Association for Computing Machinery (ACM), 1995, pp. 409–419. doi:10.1145/218013.218095.

[18] S. E. Sadaoui, C. Mehdi-Souzani, C. Lartigue, Computer-aided inspection planning: A multisensor high-level inspection planning strategy, Journal of Computing and Information Science in Engineering 19. doi:10.1115/1.4041970/422052.

[19] M. Germani, F. Mandorli, M. Mengoni, R. Raffaeli, Cad-based environment to bridge the gap between product design and tolerance control, Precision Engineering 34 (2010) 7–15. doi:10.1016/J.PRECISIONENG.2008.10.002.

[20] Q. Wu, J. Lu, W. Zou, D. Xu, Path planning for surface inspection on a robot-based scanning system, 2015 IEEE International Conference on Mechatronics and Automation, ICMA 2015 (2015) 2284–2289doi:10.1109/ICMA.2015.7237842.

[21] C. Lartigue, A. Contri, P. Bourdet, Digitised point quality in relation with point exploitation, Measurement 32 (3) (2002) 193–203. doi:10.1016/S0263-2241(02)00008-8.

[22] C. Mehdi-Souzani, F. Thiébaut, C. Lartigue, Scan planning strategy for a general digitized surface, Journal of Computing and Information Science in Engineering 6 (2006) 331–339. doi:10.1115/1.2353853.

[23] A. Zuquete-Guarato, C. Mehdi-Souzani, Y. Quinsat, C. Lartigue, L. Sabri, Towards a new concept of in-line crankshaft balancing by contact less measurement: Process for selecting the best digitizing system, ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, ESDA 2012 4 (2013) 17–25. doi:10.1115/ESDA2012-82166.

[24] M. Karaszewski, M. Adamczyk, R. Sitnik, Assessment of next-best-view algorithms performance with various 3d scanners and manipulator, ISPRS Journal of Photogrammetry and Remote Sensing 119 (2016) 320–333. doi:10.1016/J.ISPRSJPRS.2016.06.015.

[25] T. Li, R. Lou, A. Polette, D. Nozais, Z. Shao, J.-P. Pernot, On the use of quality metrics to characterize structured light-based point cloud acquisitions, Computer-Aided Design and Applications 20(6). doi:10.14733/cadaps.2023.1190-1203.

[26] M. Senthilvel, R. K. Soman, K. Varghese, Comparison of handheld devices for 3d reconstruction in construction, in: M.-Y. N. T. U. o. S. Cheng, Technology), H.-M. N. T. U. o. S. Chen, Technology), K. C. N. T. U. o. S. Chiu, Technology) (Eds.), Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC), Tribun EU, s.r.o., Brno, Taipei, Taiwan, 2017, pp. 698–705. doi:10.22260/ISARC2017/0097.

[27] Y. Xu, Y. Zhao, F. Wu, K. Yang, Error analysis of calibration parameters estimation for binocular stereo vision system, in: 2013 IEEE International Conference on Imaging Systems and Techniques (IST), 2013, pp. 317–320. doi:10.1109/IST.2013.6729713.

[28] W. Sun, J. Cooperstock, Requirements for Camera Calibration: Must Accuracy Come with a High Price?, in: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1, Vol. 1, 2005, pp. 356–361. doi:10.1109/ACVMOT.2005.102.

[29] P. Swapna, N. Krouglicof, R. Gosine, The question of accuracy with geometric camera calibration, in: 2009 Canadian Conference on Electrical and Computer Engineering, 2009, pp. 541–546. doi:10.1109/CCECE.2009.5090189.

[30] W.-J. Lin, S.-H. Lo, H.-T. Young, C.-L. Hung, Evaluation of deep learning neural networks for surface roughness prediction using vibration signal analysis, Applied Sciences 9 (7). doi:10.3390/app9071462.

[31] Žarko Ćojbašić, D. Petković, S. Shamshirband, C. W. Tong, S. Ch,

P. Janković, N. Dučić, J. Baralić, Surface roughness prediction by extreme learning machine constructed with abrasive water jet, Precision Engineering 43 (2016) 86–92. doi:10.1016/J.PRECISIONENG.2015.06.013.

[32] W. Guo, C. Wu, Z. Ding, Q. Zhou, Prediction of surface roughness based on a hybrid feature selection method and long short-term memory network in grinding, International Journal of Advanced Manufacturing Technology 112 (2021) 2853–2871. doi:10.1007/S00170-020-06523-Z/TABLES/10.

[33] S. Shang, C. Wang, X. Liang, C. F. Cheung, P. Zheng, Surface roughness prediction in ultra-precision milling: An extreme learning machine method with data fusion, Micromachines 14 (11). doi:10.3390/mi14112016.

[34] A. Boulch, J. Guerry, B. L. Saux, N. Audebert, Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks, Computers & Graphics 71 (2018) 189–198. doi:10.1016/J.CAG.2017.11.010.

[35] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, D. Filliat, Snapnet-r: Consistent 3d multi-view semantic labeling for robotics, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 669–678. doi:10.1109/ICCVW.2017.85.

[36] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3d point clouds: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2021) 4338–4364. doi:10.1109/TPAMI.2020.3005434.

[37] R. Q. Charles, H. Su, M. Kaichun, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85. doi:10.1109/CVPR.2017.16.

[38] C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: deep hierarchical feature learning on point sets in a metric space, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 5105–5114.

[39] Y. Xu, T. Fan, M. Xu, L. Zeng, Y. Qiao, Spidercnn: Deep learning on point sets with parameterized convolutional filters, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11212 LNCS (2018) 90–105. doi:10.1007/978-3-030-01237-3_6.

[40] H. Zhao, L. Jiang, C.-W. Fu, J. Jia, Pointweb: Enhancing local neighborhood features for point cloud processing, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5560–5568. doi:10.1109/CVPR.2019.00571.

[41] H.-Y. Meng, L. Gao, Y.-K. Lai, D. Manocha, Vv-net: Voxel vae net with group convolutions for point cloud segmentation, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8499–8507. doi:10.1109/ICCV.2019.00859.

[42] W. Liu, K. Zeng, Sparsenet: A sparse densenet for image classification, arXiv.org.

[43] S.-M. Hu, Z.-N. Liu, M.-H. Guo, J.-X. Cai, J. Huang, T.-J. Mu, R. R. Martin, Subdivision-based mesh convolution networks, ACM Trans. Graph. 41 (3). doi:10.1145/3506694.

[44] R. Wu, C. Xiao, C. Zheng, Deepcad: A deep generative network for computer-aided design models, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 6752–6762. doi:10.1109/ICCV48922.2021.00670.

[45] J. G. Lambourne, K. D. Willis, P. Jayaraman, A. Sanghi, P. Meltzer, H. Shayani, Brepnet: A topological message passing system for solid models, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 12768–12777. doi:10.1109/CVPR46437.2021.01258.

[46] P. K. Jayaraman, J. G. Lambourne, N. Desai, K. D. D. Willis, A. Sanghi, N. J. W. Morris, Solidgen: An autoregressive model for direct b-rep synthesis, CoRR abs/2203.13944. arXiv:2203.13944, doi:10.48550/ARXIV.2203.13944.

[47] S. Katz, A. Tal, R. Basri, Direct visibility of point sets, ACM Trans. Graph. 26 (3) (2007) 24–es. doi:10.1145/1276377.1276407.

[48] R. Mehra, P. Tripathi, A. Sheffer, N. J. Mitra, Visibility of noisy point cloud data, Comput. Graph. 34 (3) (2010) 219–230. doi:10.1016/J.CAG.2010.03.002.

[49] B. Burley, W. D. A. Studios, Physically-based shading at disney, in: Acm Siggraph, Vol. 2012, vol. 2012, 2012, pp. 1–7.

[50] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334. doi:10.1109/34.888718.

[51] L. Song, X. Li, Y. Yang, X. Zhu, Q. Guo, H. Liu, Structured-light based 3d reconstruction system for cultural relic packaging, Sensors 18 (9) (2018) 2981. doi:10.3390/S18092981.

[52] J. Li, J. Guan, H. Du, J. Xi, Error self-correction method for phase jump in multi-frequency phase-shifting structured light, Appl. Opt. 60 (4) (2021) 949–958. doi:10.1364/AO.413506.

[53] X. Zhu, Z. Zhang, L. Hou, L. Song, H. Wang, Light field structured light projection data generation with blender, in: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), 2022, pp. 1249–1253. doi:10.1109/CVIDLICCEA56201.2022.9824921.

[54] A. Puljčan, D. Zoraja, T. Petković, Simulation of structured light 3d scanning using blender, in: 2022 International Symposium ELMAR, 2022, pp. 215–220. doi:10.1109/ELMAR55880.2022.9899809.

[55] B. Karis, E. Games, Real Shading in Unreal Engine 4, Proc. Physically Based Shading Theory Practice 4 (3) (2013) 1.

[56] G. Moioli, Introduction to Blender 3.0: Learn Organic and Architectural Modeling, Lighting, Materials, Painting, Rendering, and Compositing with Blender, Apress Media LLC, 2022. doi:10.1007/978-1-4842-7954-0/COVER.

[57] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. W. III, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, Vol. 9351 of Lecture Notes in Computer Science, Springer, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

[58] H. Nguyen, Y. Wang, Z. Wang, Single-shot 3d shape reconstruction using structured light and deep convolutional neural networks, Sensors 20 (13). doi:10.3390/s20133718.

[59] S. Hu, A. Polette, J. Pernot, Sma-net: Deep learning-based identification and fitting of CAD models from point clouds, Eng. Comput. 38 (6) (2022) 5467–5488. doi:10.1007/S00366-022-01648-Z.

[60] B. Graham, M. Engelcke, L. van der Maaten, 3d semantic segmentation with submanifold sparse convolutional networks, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 9224–9232. doi:10.1109/CVPR.2018.00961.

[61] B. Graham, L. van der Maaten, Submanifold sparse convolutional networks (2017). arXiv:1706.01307.

[62] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. A. Riedmiller, Striving for simplicity: The all convolutional net, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, 2015.

[63] A. Tabb, K. M. A. Yousef, Solving the robot-world hand-eye(s) calibration problem with iterative methods, CoRR abs/1907.12425. arXiv:1907.12425.

[64] W. Wang, W. Tian, W. Liao, B. Li, J. Hu, Error compensation of industrial robot based on deep belief network and error similarity, Robotics and Computer-Integrated Manufacturing 73 (2022) 102220. doi:10.1016/j.rcim.2021.102220.

[65] Z. Jiang, M. Huang, X. Tang, Y. Guo, A new calibration method for joint-dependent geometric errors of industrial robot based on multiple identification spaces, Robotics and Computer-Integrated Manufacturing 71 (2021) 102175. doi:10.1016/j.rcim.2021.102175.

[66] A. Angelidis, G.-C. Vosniakos, Prediction and compensation of relative position error along industrial robot end-effector paths, International journal of precision engineering and manufacturing 15 (2014) 63–73. doi:10.1007/s12541-013-0306-5.

[67] W. Tian, D. Mei, P. Li, Y. Zeng, P. Hong, W. Zhou, Determination of optimal samples for robot calibration based on error similarity, Chinese Journal of Aeronautics 28 (3) (2015) 946–953. doi:10.1016/j.cja.2015.03.003.

[68] M. Slamani, A. Nubiola, I. A. Bonev, Assessment of the positioning performance of an industrial robot, Ind. Robot 39 (1) (2012) 57–68. doi:10.1108/01439911211192501.

[69] J. Qi, B. Chen, D. Zhang, Compensation for absolute positioning error of industrial robot considering the optimized measurement space, International Journal of Advanced Robotic Systems 17 (2) (2020) 1729881420921644. doi:10.1177/1729881420921644.