The International Journal of Advanced Manufacturing Technology (2021) 116:1895–1911
https://doi.org/10.1007/s00170-021-07432-5

ORIGINAL ARTICLE

Autonomous 3D geometry reconstruction through robot-manipulated optical sensors

Carmelo Mineo[1] · Donatella Cerniglia[1] · Vito Ricotta[1] · Bernhard Reitinger[2]

Received: 2 April 2021 / Accepted: 3 June 2021 / Published online: 5 July 2021

Abstract

Many industrial sectors face increasing production demands and the need to reduce costs, without compromising the quality. The use of robotics and automation has grown significantly in recent years, but versatile robotic manipulators are still not commonly used in small factories. Beside of the investments required to enable efficient and profitable use of robot technology, the efforts needed to program robots are only economically viable in case of large lot sizes. Generating robot programs for specific manufacturing tasks still relies on programming trajectory waypoints by hand. The use of virtual simulation software and the availability of the specimen digital models can facilitate robot programming. Nevertheless, in many cases, the virtual models are not available or there are excessive differences between virtual and real setups, leading to inaccurate robot programs and time-consuming manual corrections. Previous works have demonstrated the use of robot-manipulated optical sensors to map the geometry of samples. However, the use of simple user-defined robot paths, which are not optimized for a specific part geometry, typically causes some areas of the samples to not be mapped with the required level of accuracy or to not be sampled at all by the optical sensor. This work presents an autonomous framework to enable adaptive surface mapping, without any previous knowledge of the part geometry being transferred to the system. The novelty of this work lies in enabling the capability of mapping a part surface at the required level of sampling density, whilst minimizing the number of necessary view poses. Its development has also led to an efficient method of point cloud down-sampling and merging. The article gives an overview of the related work in the field, a detailed description of the proposed framework and a proof of its functionality through both simulated and experimental evidences.

Keywords View planning · 3D reconstruction · Adaptive mapping · Metrology · Inspection · Robotics

1 Introduction

1.1 Motivation

This work is motivated by the need to develop an effective approach to measure the geometry of workpieces. In recent years, the use of robotics has increasingly penetrated the manufacturing and the construction industries [1–3]. Besides being attractive to make production phases more cost-effective, robotics and automation have been used to speed up quality inspections [4, 5] and to operate in hazardous environment precluded to human access [6, 7]. Many industrial automated systems are based on robotic arms that manipulate actuators and sensors through predefined tool paths in structured environments. The robot tool paths are typically defined on the digital Computer-Aided Design (CAD) models of the parts to be machined, assembled, disassembled and/or inspected. The process of generating robot tool-paths using simulation software is known as Off-Line Path-planning (OLP) [8]. Unfortunately, the digital models often differ from their respective real counterparts and time-consuming human intervention is required to correct the software OLP robot paths and ensure they meet the required levels of accuracy [9]. Therefore, highly versatile robotic arms that could be used for flexible autonomous systems are still mainly used to automate repetitive tasks in large industries with well-structured environments. Indeed, besides of the investments required to enable efficient and profitable use of robot technology,

✉ Carmelo Mineo
carmelo.mineo01@unipa.it

[1] Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 8, 90128 Palermo, Italy

[2] Research Center for Non-Destructive Testing GmbH (RECENDT), Science Park 2 / 2.OG, Altenberger Straße 69, A-4040 Linz, Austria

the efforts needed to program robots are only economically viable in case of large lot sizes. Research efforts have been put into developing more intuitive programming methods to reduce the programming time [10]. In some specific scenarios (e.g. robotic welding), the path inaccuracy is corrected by seam tracking based on laser profiling sensors for real-time program adaptation [11]. However, the adaptation strategy is limited to simple workpiece geometries. More promising approaches use computer vision to reconstruct the real workpiece geometry and automatically generate robot programs for each new part [12]. Besides three-dimensional (3D) object reconstruction becoming important in numerous industrial applications such as smart manufacturing, industrial automation and Industry 4.0 [13], there exists a wide variety of applications that would benefit from real-time computer vision systems, capable of autonomous object reconstruction. It is the case of virtual reality (VR) games and simulations, augmented reality (AR) applications or systems that include obstacle detection [14].

1.2 Related work

A plethora of methods and systems have been proposed for the acquisition of the geometry of real-life objects, ranging from those which employ active sensor technology, passive sensor technology or a combination of various techniques. The data produced by a 3D scanner is point cloud of the object surface. A well-established classification of the sensors used for 3D reconstruction divides them into two types: contact and non-contact sensors [15]. Contact 3D scanners probe the subject through physical touch, while the object is firmly held in place [16, 17]. Non-contact solutions can be further divided into two main categories: active and passive. Passive 3D scanning solutions rely on detecting reflected ambient radiation. Most solutions of this type detect visible light because it is a readily available ambient radiation, but other types of radiation (e.g. infrared) could also be used. Passive methods can be very cheap, because in most cases they do not need particular hardware but simple digital cameras. On the other hand, active scanners emit some kind of radiation or light and detect its reflection or attenuation [18]. Regardless of the deployed technique, 3D scanners have much in common with cameras. Like most cameras, they have a cone-like field of view and can only collect information about surfaces that are not obscured. While a camera collects colour information about surfaces within its field of view, the main objective of a 3D scanner is to collect distance information about the surfaces within its field of view. Many types of 3D scanning sensors have been designed and used in real applications. Among the scanning sensors, the ones that can be easily integrated with robotic arms to perform automated object reconstruction, can be divided into two categories. The first category comprises the depth cameras (also known as 3D cameras). Depth cameras are designed to return point clouds. Such devices can consist of two conventional grey-scale cameras (stereo-cameras [19]) or sensors that provide RGB colour and depth for each pixel (RGB-D cameras [20]). The second category comprises all those devices that use the controlled emission and reception of light signals (laser beams) as fundamental measurement tool [21]. In the reception phase, a laser scanner can use different techniques for calculating the distance between the laser source and the point hit by the laser beam. According to the technique used, laser scanners are based on trigonometric calculation (triangulation), time-of-flight (when they calculate the distance through the time elapsed between the emission of the laser and the reception of the return signal [22]), or on phase difference (when the calculation is performed by comparing the phase of the emitted signal and the return signal [23]). For most practical situations, a single acquisition from one point of view will not produce a complete model of the subject of interest. Multiple scans, even hundreds, from many different directions are usually required to obtain information about all sides of the subject. Several works have advanced the process of bringing the point clouds, originating from multiple scans, into a common reference system (a process that is usually called alignment or registration). The merged point clouds create the complete 3D model. This whole process, going from the single range map to the whole model, is usually known as the 3D scanning pipeline [24, 25]. Complete 3D reconstruction of a scene is typically achieved by establishing a relative motion between the scanning system and the object to reconstruct, while data is captured by the system. Hand-held 3D scanners rely on the user to move slowly around the object, visiting all object areas of interest, while data is acquiring. When a scanning system is manipulated by a robotic arm, the problem of determining the scanning path arises. Previous works have obtained good automated 3D reconstructions of parts by moving a robot-manipulated 3D scanner around a given component through a predefined path, along which multiple views of the scene are collected. In [26], the authors proposed using a robot arm to move a non-contact passive 3D scanning system, following spiral paths lying on paraboloid primitives and stopping at regular intervals with the camera pointing at the centre of the paraboloid, to collect photogrammetric views of relatively small industrial parts. Although this may be an acceptable scanning path for some objects, it can cause some portions of the part to not be scanned at all, some other areas to not be scanned to a satisfactory or acceptable extent and/or, on the contrary, some remaining areas to be over-sampled. Fixing the path trajectory and the spacing with which data is captured produces sub-optimal 3D reconstructions,

since the acquisition path is not targeted to any specific object. Manual determination of optimal view poses for surface scanning is a time-consuming and expert-dependent task and, despite of the efforts, redundant views are usually deployed. OLP software allows simulating the reachability of view poses and avoiding collisions, when the approximate CAD model of the part to reconstruct is available. Nevertheless, finding the optimum set of view poses for a robot-manipulated 3D scanning system, in order to efficiently reconstruct a given scene using the minimum number of views is still an open problem [27, 28]. It is known under the name of View Pose Planning (VPP) [17, 29].

1.3 Contribution

This work presents a mathematical framework for adaptive and incremental 3D reconstruction of specimens, through the use of a robot-manipulated optical 3D scanner. It allows computing the next optimal view pose after each measurement view. Compared with previous works [30, 31], the proposed approach does not require any prior knowledge about the shape of the object, meaning that the formulation creates a best-guess representation of the subject of the 3D scanning and updates it after each measurement data. Crucially, the method is suitable to obtaining measurable/quantitative results, since it seeks to reach a user-defined target sampling density, which is provided as fundamental input parameter. Such sampling density is expressed as number of points per surface unit (e.g. points/mm^2). Compared to other recently published works, the present approach does not make use of neural-network paradigms [28], exhibiting more deterministic performance. The framework is accompanied by the definition of meaningful stopping criteria, whose fulfilment leads to the termination of the iterative computation of the next view pose and the output of the final result in the form of merged point cloud and reconstructed tessellated model (triangular mesh surface). The framework has not been developed to work only with specific sensor hardware and is adaptable to operate with data streams obtained through a generic range scanning sensor, either depth camera or 3D laser scanner type sensor. Its development has also led to an efficient method for point cloud down-sampling and merging. The framework functionality has been tested through MATLAB-simulated data, obtained from synthetic views of a computer graphics 3D test model developed at Stanford University [32]. The MATLAB-based code is openly available (https://doi.org/10.5281/zenodo.4646850) and can be used by the research community for future developments. In order to validate the framework in experimental scenarios, the control computer has been interfaced with a robot arm and a low-cost RGB-D camera to reconstruct the geometry of a 3D printed version of the Stanford University test model and of an additional industrial test piece.

1.4 Article structure

The remaining of the article is structured as follows. Section 2 describes the theoretical foundations of the framework. Section 3 illustrates the experimental setup, the hardware components and the interfacing platforms utilized for the validation tests. The results arising from simulations and synthetic data sets are illustrated in Section 4. The results obtained through real sensor data sets are presented in Section 5. Finally, Section 6 draws the conclusions and a prospect of future work.

2 Theoretical foundations

This section starts defining all the metrics of 3D scanning sensors and of point clouds, which are used herein to describe the theoretical foundations of the approach presented in this work and discuss the simulations and the experimental results. Then, it describes the approach used for incremental merging of the point clouds acquired from different view poses. Finally, this section focuses on explaining the method elaborated to select the next best acquisition view pose and suitable stopping criteria for adaptive incremental 3D reconstruction.

2.1 Definition of metrics

Before any algorithm can be described, it is necessary to define all the parameters and variables that intervene in the mathematical formulation of the problem of interest. Figure 1a and b show, respectively, point clouds collected through a depth camera type sensor and a laser scanner type sensor. An orthogonal reference system is centred at the sensor data origin. Like a conventional RGB camera, a depth camera has a pyramidal sampling volume, whose dimension depends on the horizontal field-of-view angle (ϑ) and on the vertical field-of-view angle (θ). These angles are bisected by the $\overrightarrow{w}$ vector. Like in conventional RGB cameras, depth cameras allow obtaining equally spaced 3D point samples arranged in a rectangular grid, whose number is equal to the product of the sensor horizontal and vertical pixel resolution (respectively R_h and R_v), when sampling a flat surface parallel to the $\overrightarrow{u} - \overrightarrow{v}$ plane. The total surface area sampled on such plane, at distance d from the $\overrightarrow{u} - \overrightarrow{v}$ plane, is equal to:

$$A_{depth-camera} = a * b = \left(2d * tan\left(\frac{\vartheta}{2}\right)\right)\left(2d * tan\left(\frac{\theta}{2}\right)\right) = 4d^2 * tan\left(\frac{\vartheta}{2}\right) tan\left(\frac{\theta}{2}\right) \quad (1)$$

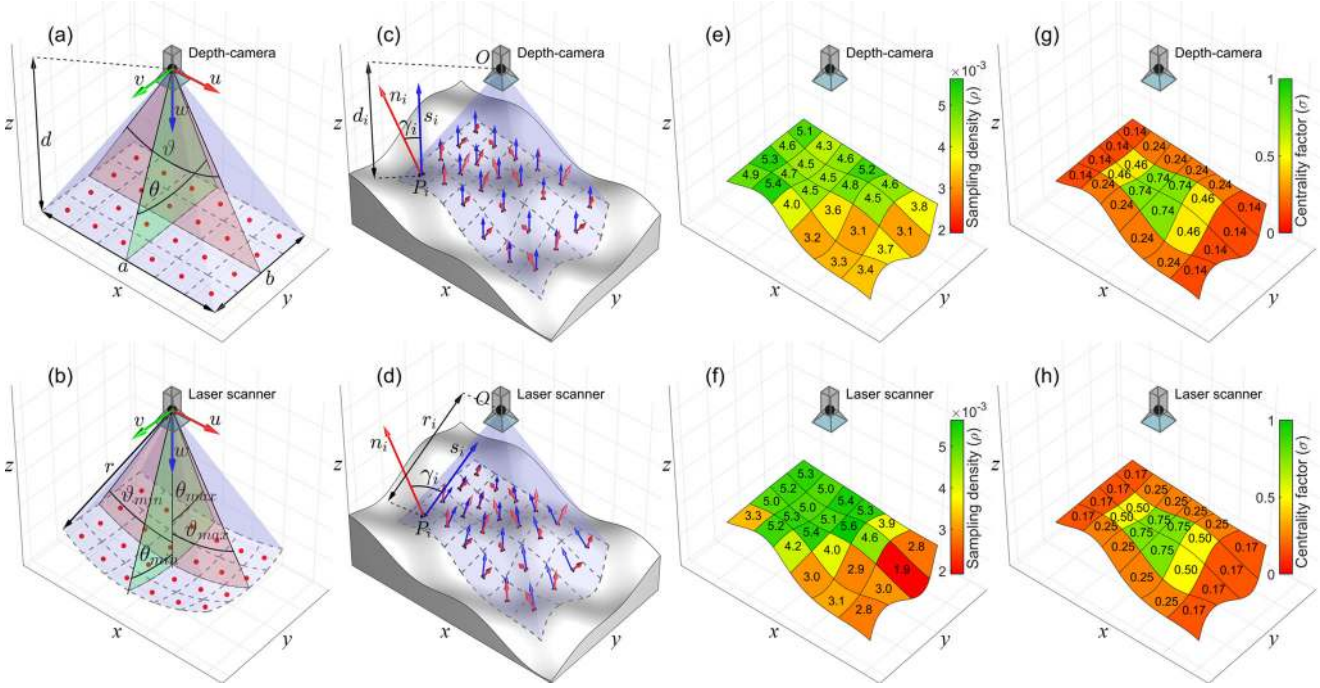

Fig. 1 Fundamental working parameters for a depth camera sensor (**a**) and a laser scanner sensor (**b**), representation of the vectors for the computation of the local sampling density on an example surface (**c–d**), local sampling densities (**e–f**) and centrality factors (**g–h**) computed for all points collected by the generic depth camera and laser scanner

Assuming that the Cartesian coordinates of the sampled point $P_i \equiv [x_i, y_i, z_i]$ are given with respect to the reference system $(\overrightarrow{u}, \overrightarrow{v}, \overrightarrow{w})$ of the 3D scanning, the distance d_i between the plane $\overrightarrow{u} - \overrightarrow{v}$ and the parallel plane for a sampled point, is $d_i = z_i$. A laser-based 3D scanner, schematically represented in Fig. 1b, operates the deflection of the sampling laser beam in angular coordinates. In this work, the angles θ and ϑ are defined as the angles that the position vector forms with the x-z and y-z plane, respectively. Typically, the user can set the desired scanning range, defining lower and upper limits, with $-\pi \leq \vartheta_{min} < 0$, $0 \leq \vartheta_{max} \leq \pi$, $-\pi \leq \theta_{min} < 0$ and $0 \leq \theta_{max} \leq \pi$. Moreover, the user can usually set the number of points to be captured in such angular ranges. As a result, when sampling the detectable portion of the inner surface of the sphere with radius r centred at the sensor origin, a laser scanner allows obtaining equally spaced 3D point samples arranged in a rectangular spherical grid. The total area sampled on such portion of the spherical surface is equal to:

$$\begin{aligned} A_{laser-scanner} &= r^2 * \int_{\vartheta_{min}}^{\vartheta_{max}} d\vartheta * \int_{\theta_{min}}^{\theta_{max}} \cos\theta \, d\theta \\ &= r^2 * (\vartheta_{max} - \vartheta_{min}) * [\sin(\theta_{max}) - \sin(\theta_{min})] \end{aligned} \tag{2}$$

For the purposes of this work, it is crucial to define the local sampling density, given as sampled number of points per squared unit of length (e.g. points/mm^2), for every sampled point. Figure 1c and d represent the points captured by a depth camera and a laser-based sensor through scanning a generic surface. A *sampling vector* ($\overrightarrow{s}_i$) is defined for the i^{th} sampled point (P_i), as the unitary vector normal to that surface for P_i where the sensor would acquire equally spaced samples. Whereas $\overrightarrow{s}_i$ is always perpendicular to the flat surface parallel to the $\overrightarrow{u} - \overrightarrow{v}$ plane at distance d_i for the depth camera type sensor, it is always normal to the surface of sphere centred at the sensor origin with radius r_i, for the laser-scanner type sensor. Therefore, in the case of a depth camera, $\overrightarrow{s}_i$ is equal to $-\overrightarrow{w}$, while it is always the radial vector pointing to the sensor origin ($\overrightarrow{s}_i = O - P_i$), in the case of a laser scanner. Indicating with $\overrightarrow{n}_i$ the vector normal to the scanned surface and with γ_i the angle that this vector forms with $\overrightarrow{s}_i$, the local sampling density (ρ_i) at the i^{th} sampled point, in case a depth camera or a laser scanner is used, is herein defined as:

$$\begin{aligned} \rho_i &= \frac{R_h R_v}{A_i^{depth-camera}} * \cos(\gamma_i) \\ &= \frac{R_h R_v}{4 d_i^2 * tan\left(\frac{\vartheta}{2}\right) tan\left(\frac{\theta}{2}\right)} \frac{\overrightarrow{s}_i \overrightarrow{n}_i}{|\overrightarrow{s}_i| |\overrightarrow{n}_i|} \end{aligned} \tag{3}$$

$$\begin{aligned} \rho_i &= \frac{R_h R_v}{A_i^{laser-scanner}} * \cos(\gamma_i) \\ &= \frac{R_h R_v}{r_i^2 (\vartheta_{max} - \vartheta_{min}) [\sin(\theta_{max}) - \sin(\theta_{min})]} \frac{\overrightarrow{s}_i \overrightarrow{n}_i}{|\overrightarrow{s}_i| |\overrightarrow{n}_i|} \end{aligned} \tag{4}$$

It is worth highlighting that R_h, R_v, ϑ, θ, ϑ_{max}, ϑ_{min}, θ_{max} and θ_{min} are known working parameters of the sensor

and d_i, r_i and $\overrightarrow{s}_i$ can be easily computed using the coordinates of the acquired point and the known pose of the scanning device. The only variable that must be approximated is $\overrightarrow{n}_i$, which is the local normal of the scanned surface at the point P_i. Indeed, the surface is not analytically known before the scan and the objective of the scan is to reconstruct the shape of the surface. In this work, the local normal is inferred through fitting a local plane to neighbouring points [33], in order to approximate its perpendicular vector. The orientation of the normal is set based on the knowledge of the sensor pose, making sure that the absolute value of γ_i (the angle formed by $\overrightarrow{n}_i$ with $\overrightarrow{s}_i$) is smaller than $\pi/2$. Figure 1e and f give a representation of the local sampling densities computed for all points collected on the example surface by the generic depth camera and laser scanner. Referring to the notation given in Fig. 1a and b, the same scanning resolutions and angular ranges are used for the depth camera and laser sensor ($R_h = 6$, $R_v = 4$, $\vartheta/2 = \vartheta_{max} = -\vartheta_{min} = \pi/6$ and $\theta/2 = \theta_{max} = -\theta_{min} = \pi/9$). The same colormap and colour bar limits have been set in Fig. 1e and f to facilitate the comparison of the different local sampling densities relative to the points sampled through the depth camera and the laser scanner. As expected, the low values ($\sim 10^{-3}$) are due to the low horizontal and vertical resolution used for the sake of producing clear schematic representations. Much higher resolutions are typically used to obtain useful results in real applications. The last metric used by this work is named as *centrality factor* (σ). The centrality factor is a nondimensional parameters, whose value is comprised between 0 and 1, being $\sigma = 1$ for a point measured at the centre of the sensor field-of-view and $\sigma = 0$ for points measured at the boundary of the field of view. This factor is computed as in Eqs. 5 and 6 for depth cameras and laser scanners, respectively:

$$\sigma_i = \min\left(1 - \left|\tan^{-1}\left(\frac{x_i}{z_i}\right)\right| \Big/ \left(\frac{2}{\vartheta}\right),\ 1 - \left|\tan^{-1}\left(\frac{y_i}{z_i}\right)\right| \Big/ \left(\frac{2}{\theta}\right)\right) \tag{5}$$

$$\sigma_i = \min\left(\begin{array}{c} 1 - \left|\tan^{-1}\left(\frac{x_i}{z_i}\right) - \frac{\vartheta_{max}+\vartheta_{min}}{2}\right| \Big/ \left(\frac{2}{\vartheta_{max}-\vartheta_{min}}\right), \\ 1 - \left|\tan^{-1}\left(\frac{y_i}{z_i}\right) - \frac{\theta_{max}+\theta_{min}}{2}\right| \Big/ \left(\frac{2}{\theta_{max}-\theta_{min}}\right) \end{array}\right) \tag{6}$$

2.2 Incremental down-sampling and merging

As it was said in the introduction, in most situations, the acquisition of a single point cloud from one point of view cannot produce a complete 3D reconstruction of an object. Multiple point clouds, collected with different sensor poses are typically required. The alignment/registration process of bringing the multiple point clouds into a common reference system is quite straightforward, when the accurate position and orientation of each sensor pose are available, which is always the case for robot-manipulated 3D scanners. In this work, it is assumed that the sensor data origin is accurately calibrated as robot Tool Central Point (TCP) and all collected point clouds get registered into the manipulation robot base reference system, using the sensor pose (position Cartesian coordinates and orientation Euler angles), obtained as feedback from the robot controller. Therefore, the resulting merged point cloud may be intended as the set of all points collected through all sensor views. At first glance, it would be possible to think the sensor should be positioned at a distance from an object surface that allows capturing as many points as are needed to reach the desired target density. If such target density is denoted with ρ^*, expressed as number of points per surface unit (e.g. points/mm^2), the optimum sensor view distance (d_g^*) or view radius (r_g^*) can be extrapolated from Eqs. 7 and 8, for depth cameras and laser scanners respectively:

$$d_g^* = \frac{1}{2}\sqrt{\frac{R_h R_v}{\rho^* tan\left(\frac{\vartheta}{2}\right) tan\left(\frac{\theta}{2}\right)}} \tag{7}$$

$$r_g^* = \sqrt{\frac{R_h R_v}{\rho^* \left(\vartheta_{max} - \vartheta_{min}\right)\left[\sin\left(\theta_{max}\right) - \sin\left(\theta_{min}\right)\right]}} \tag{8}$$

The subscript "g" is given to d_g^* and r_g^*, since they purely derive from geometrical considerations. Placing a depth cameras at distance d_g^* or a laser scanner at radial distance r_g^* allows reconstructing the object geometry exactly at target density only when a planar (for depth cameras) or a spherical surface (for laser scanners) is the surface under inspection. This is far from any real applications, when a generic surface is to be mapped. Moreover, most manufacturers of 3D scanners specify that the sampling inaccuracy/noise of their sensors depends on the distance of the captured points. Assuming the expected measurement noise of a 3D scanner is defined as a percentage of sampling distance ($\varepsilon = noise/d$ or $\varepsilon = noise/r$), it is possible to compute the maximum distance that allows mapping a surface with measurement noise smaller than or equal to n^*:

$$d_n^* = \frac{n^*}{\varepsilon} \quad or \quad r_n^* = \frac{n^*}{\varepsilon} \tag{9}$$

These limit values are denoted with the "n" subscript, since they originate from measurement noise considerations. Thus, in practical application, the optimum view distance (d^* or r^*) is chosen as the lower value between d_g^* and d_n^* ($d^* = \min(d_g^*, d_n^*)$) or r_g^* and r_n^* ($r^* = \min(r_g^*, r_n^*)$). Some sensors with high values of percentual noise (ε) force mapping objects/environments at distances that lead to sampling densities much higher than the target density (e.g. when $d_n^* \ll d_g^*$ or $r_n^* \ll r_g^*$). Moreover, due to the overlap between the field of view of the 3D scanning sensor positioned at different locations, simply appending all collected points to a comprehensive point cloud may lead

to vast regions with too many redundant points. This means that many more points, compared to those required to fulfil the target sampling density, are collected in some regions of an object, making the merged point cloud difficult to process in timely fashions and to store in physical memories. For these reasons, solutions to down-sample the collected points and obtain a uniform point density across the resulting point cloud are typically found in many works [14, 34]. Although down-sampling algorithms have been presented elsewhere, it is worth describing what down-sampling and merging algorithms were implemented in this work, for the sake of making the entire incremental 3D reconstruction pipeline as clear as possible. Figure 2a gives an explanatory scene, showing an initial state point cloud (originating from two sensor data sets captured at O_1 and O_2) and a dense point cloud, newly received from the sensor at O_3. The new point cloud is intentionally assumed to have a point density much higher than the target sampling density and captured with a noticeable spatial overlap with the field of view of the sensor in O_1 and O_2. Therefore, referring to this scene, it is possible to describe the process of merging the j^{th} point cloud data set with the initial state point cloud, originating from all previously acquired data sets (from the 1^{st} to the $(j-1)^{th}$ sensor pose). The average distance between any point of an ideal point cloud, which maps the surface of an object with the target density (ρ^*), and its closest neighbour point should be equal to $l = \rho^{*-1/2}$. Indeed, any square of area l^2 lying on the surface of the object should contain only one of the sampled points. This assumes that approximating the object surface to a plane is acceptable, in the neighbourhood of the square. In these terms, down-sampling a point cloud to meet the target density requirement would consist in finding all squares with side equal to l that lie on the reconstructed object surface and contain more than one sampled point. Wherever multiple points are detected within a square, only one point should be kept as a representative of them. This process is quite computationally expensive for large point clouds.

In this work, a much more efficient sub-optimal algorithm has been found, which uses cubic containers rather than squares. The area of the largest planar surface that can be inscribed in a cube is $\sqrt{2}$ times larger than area of the square face of the cube. Therefore, in this work the volume containing the points of both the initial and new cloud is partitioned with cubes of side $l^* = \left(\sqrt{2}\rho^*\right)^{-1/2}$. Indicating with $P_i \equiv [p_i^x, p_i^y, p_i^z]$ the i^{th} point of the initial state cloud and with $Q_k \equiv [q_k^x, q_k^y, q_k^z]$ the k^{th} point of the new cloud, being $k \in \mathbb{N} \mid 1 \leq k \leq (R_h * R_v)$, the local normals ($\overrightarrow{n}_i$ and $\overrightarrow{n}_k$) are computed as described in

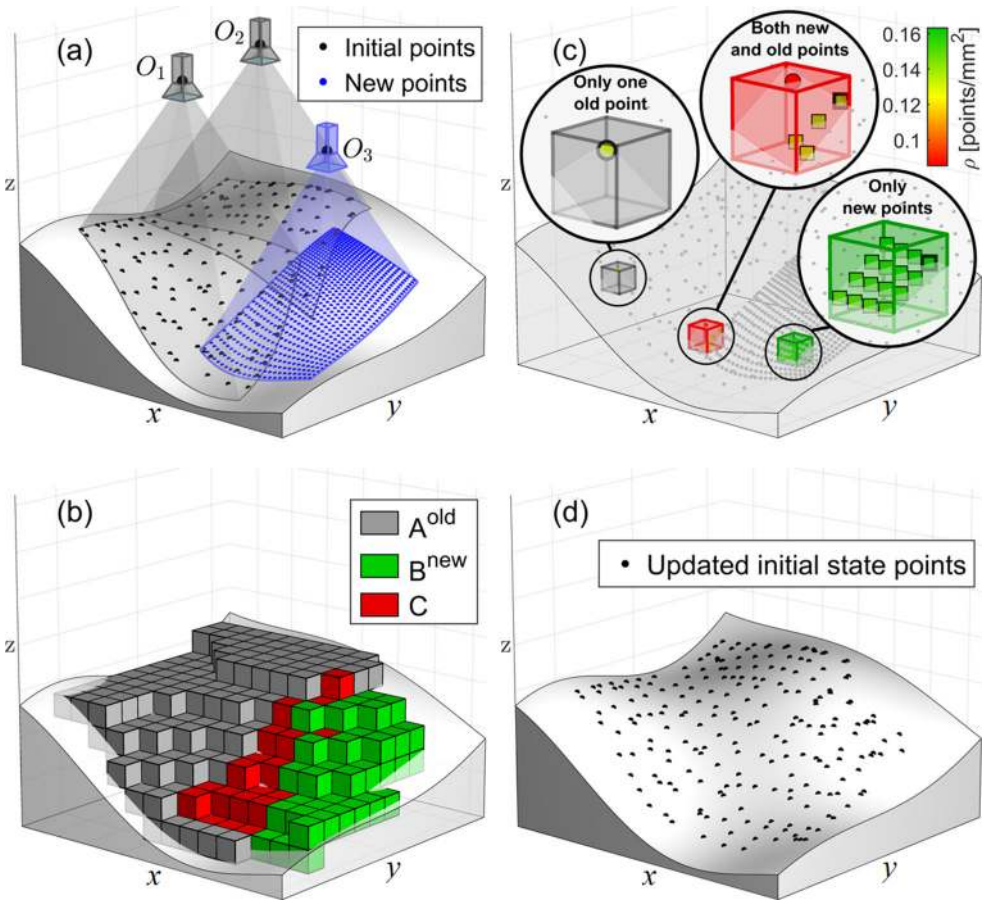

Fig. 2 Initial state points and new incoming points (**a**). Grouping points into cubes of side equal to the target sampling density (**b**). Example of selection of maximum sampling density point in a cube containing only one old point, only new points and both old and new points (**c**). Resulting merged and down-sampled new initial state (**d**)

Section 2.1 through fitting a local plane to the six closest neighbouring points, taken from the whole set of points (old and new), before the down-sampling of the new cloud is performed. Thus, the stack indices, along the x, y and z direction (a_i^x, a_i^y, a_i^z, b_k^x, b_k^y and $b_k^z \in \mathbb{Z}$), of the respective cubes (A_i and B_k) that contain the two points are calculated, dividing their Cartesian coordinates by l^* and rounding to the closest integer numbers. As it is illustrated in Eqs. 10 and 11, working with arrays, a computer can efficiently compute the set of all cubes comprising the initial state points (A) and the set of cubes for new points (B). Through the intersection of A and B (12), it possible to identify the set C of cubes that contain both initial state points and new points. The set A^{old} (subset of A), which contains cubes with only one initial state point, is defined as the difference between set A and set C (13). Finally, the set B^{new} (subset of B), which contains cubes with only new points, is defined as the difference between set B and set C (14). The cubes belonging to these sets are represented in Fig. 2b.

The merged initial state point cloud is assumed to be already down-sampled, since it is intended to be the result of the down-sampling and merging operations performed right after the acquisition of the $(j-1)^{th}$ point cloud. Figure 2c gives close up examples of the points found within cubes belonging to A^{old}, B^{new} and C, where the points from the initial state cloud are displayed as circles, the points from the new cloud are showed as squares and the colour of the points is related to their respective local sampling densities (ρ).

$$A = \begin{bmatrix} \cdots \\ A_{i-1} \\ A_i \\ A_{i+1} \\ \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \cdots \\ a_{i-1}^x & a_{i-1}^y & a_{i-1}^z \\ a_i^x & a_i^y & a_i^z \\ a_{i+1}^x & a_{i+1}^y & a_{i+1}^z \\ \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \cdots & \cdots & \cdots \\ p_{i-1}^x & p_{i-1}^y & p_{i-1}^z \\ p_i^x & p_i^y & p_i^z \\ p_{i+1}^x & p_{i+1}^y & p_{i+1}^z \\ \cdots & \cdots & \cdots \end{pmatrix} / l^* \end{bmatrix} \tag{10}$$

$$B = \begin{bmatrix} \cdots \\ B_{k-1} \\ B_k \\ B_{k+1} \\ \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \cdots \\ b_{k-1}^x & b_{k-1}^y & b_{k-1}^z \\ b_k^x & b_k^y & b_k^z \\ b_{k+1}^x & b_{k+1}^y & b_{k+1}^z \\ \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \cdots & \cdots & \cdots \\ q_{k-1}^x & q_{k-1}^y & q_{k-1}^z \\ q_k^x & q_k^y & q_k^z \\ q_{k+1}^x & q_{k+1}^y & q_{k+1}^z \\ \cdots & \cdots & \cdots \end{pmatrix} / l^* \end{bmatrix} \tag{11}$$

$$C = A \cap B = \{x \mid (x \in A) \wedge (x \in B)\} \tag{12}$$

$$A^{old} = A - C = \{x \mid (x \in A) \wedge (x \notin B)\} \tag{13}$$

$$B^{new} = B - C = \{x \mid (x \in B) \wedge (x \notin A)\} \tag{14}$$

The down-sampled and merged point cloud, which will constitute the updated initial state cloud, will have a number of points equal to the sum of the cubes in all three sets, since only one point per cube is to be selected. This allows a computer to allocate the memory space required for such point cloud. Each cube of A^{old} contains one and only one initial state point, which is transferred to the updated initial state. Every cube in B^{new} comprises points of the new cloud and the point which presents the maximum local sampling density is selected to become part of the updated initial state. Finally, each of the cubes in C always contains one point from the old cloud and one or more points from the new cloud; among them, the point with the maximum local sampling density is selected as representative. Therefore, in this work, the point representative of each volumetric partition is not randomly selected among those present in every cube, but the local sampling density (ρ) is used as a quality propriety to select the best point. This typically allows only the points that carry lower measurement noise levels to be transferred to the updated initial state cloud and to progress along the 3D reconstruction pipeline. It should be noted that the approach used in this work performs efficient incremental down-sampling and merging in a single pass, since merging takes place during down-sampling. Furthermore, the indexing of the points, operated through Eqs. 10 and 11, minimizes the computational effort. Figure 2d shows the updated initial state point cloud.

2.3 Next best view pose computation

In order to automate the acquisition of data for object reconstruction, it is necessary to be able to select the sensor poses through a suitable algorithm. Assuming the first sensor pose is human-defined and no additional information about the object geometry is provided to the algorithm, this work introduces an approach able to maximize the 3D reconstruction of the object surface, while minimizing the number of sensor poses required to achieve this objective. The 3D geometry mapping is operated incrementally, meaning that the system updates the object reconstruction, in the form of a merged point cloud and a tessellated triangular surface, right after each new point cloud is acquired by the sensor from a new pose. Following the acquisition of the J^{th} point cloud from the sensor at pose O_J, the set of all visited sensor poses (O_1, $O_2, \ldots,$ O_J) and the updated initial state point cloud (as illustrated in Section 2.2) are used to compute the next best view pose (O_{J+1}). This is the pose that allows maximizing the mapping information that can be retrieved from the sensor to reconstruct the real geometry.

In this work, a tessellated mesh that reconstructs the mapped object surface (with a level of detail corresponding to the user-defined target sampling density) is computed at each step, by applying the Poisson-based surface reconstruction algorithm described in [35] to the updated initial state point cloud. As example, the subplots in Fig. 3 show the reconstructed surface relative to the updated initial state point cloud given in Fig. 2d.

Therefore, it is checked if line-of-sight exists between the barycentre of each mesh triangle and every visited sensor pose. For the barycentres that are within the field of view of the sensor at a given pose, the ray casting method presented in [36] is used, determining whether the line segment that links each barycentre to the sensor pose has only one intersection with the mesh and if this intersection is at the barycentre. Therefore, the sampling densities relative to each sensor pose are computed according to Eq. 3 for depth cameras and Eq. 4 for laser scanners (see Fig. 3a–c). Indicating with $\rho_{i,j}$ the sampling density of the i^{th} barycentre, relative to the j^{th} sensor pose, the cumulative value (see Fig. 3d) at the i^{th} barycentre is computed as:

$$\widehat{\rho}_{i,J} = \min\left(\rho^* \quad , \quad \sum_{1}^{J} \rho_{i,j}\right) \tag{15}$$

The selection of the minimum value between ρ^* and $\sum_1^J \rho_{i,j}$, which is operated in Eq. 15, should not surprise the reader, since it is promptly justifiable as the mathematical consequence of the down-sampling described in Section 2.2.

2.3.1 Objective function definition

In this work, it has been observed that all values of $\widehat{\rho}_i$, with $1 \leq i \leq T$ (where T is the number of triangles in the Poisson reconstruction mesh), may exceed the target sampling density ρ^* even when some areas of the object are still to be mapped. This is likely to happen when the object surface is sampled with a standoff distance smaller than d_g^* (for depth cameras) or r_g^* (for laser scanners). In such case, it is difficult to use $\widehat{\rho}_i$ alone to formulate an objective function, which is suitable to determine the next best sensor pose (O_{J+1}) and valid stopping criteria for the incremental 3D reconstruction. Moreover, it is important that the next sensor pose does not coincide with any of the previously visited poses ($O_1, O_2, \ldots, O_J$). However, $\widehat{\rho}_i$ does not convey enough information about such previous poses. This problem is solved by defining the *cumulative centrality factor* $\widehat{\sigma}_i$ (see Fig. 3h) as:

$$\widehat{\sigma}_{i,J} = \max(\sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,J}) \tag{16}$$

where $\sigma_{i,j}$ is the centrality factors of the i^{th} barycentre,

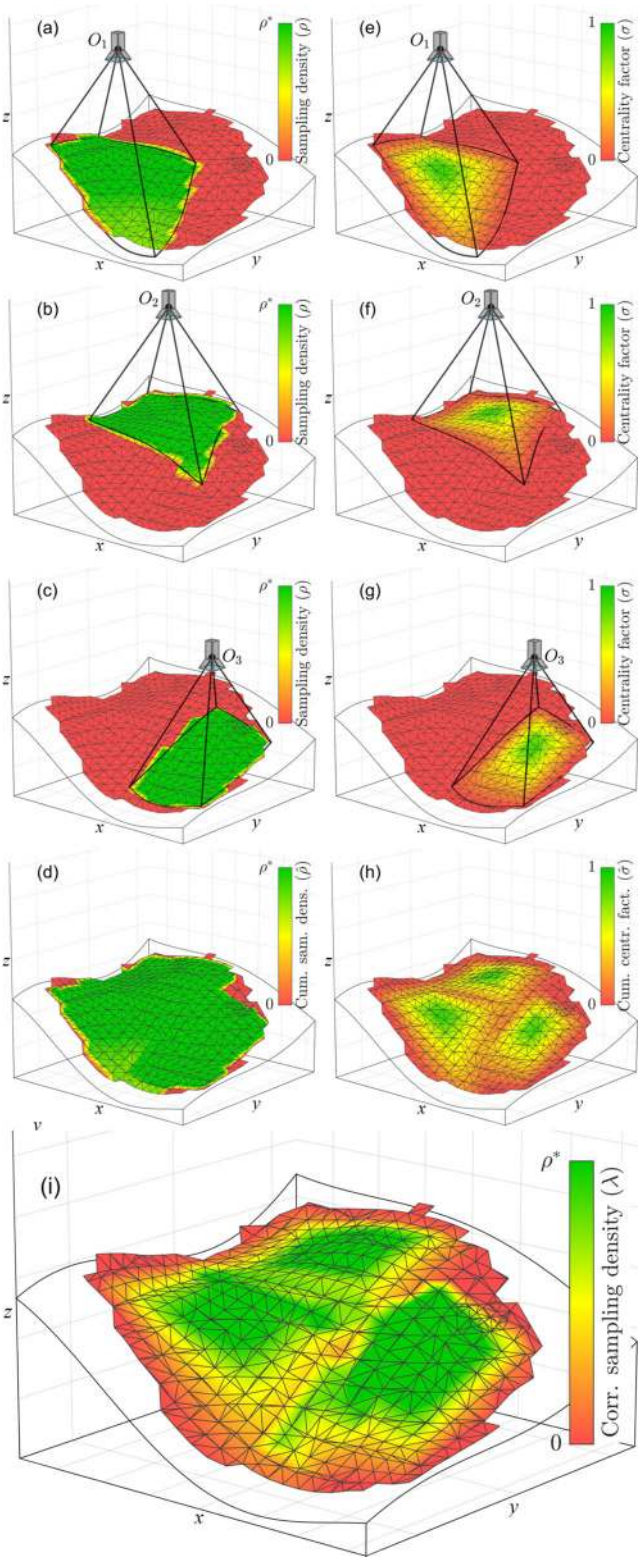

Fig. 3 Sampling density relative to each sensor pose (**a**–**c**), cumulative sampling distance (**d**), centrality factor relative to each sensor pose (**e**–**g**), cumulative centrality factor (**h**) and corrected cumulative sampling density (**i**)

relative to the j^{th} sensor pose, as it is defined in Eq. 5 for depth cameras and Eq. 6 for laser scanners (see Fig. 3e–g). The value of the cumulative centrality factor is always comprised between 0 and 1, being equal to 0 at the boundary of the cumulative surface mapped from all sensor poses and equal to 1 at the intersection between the sensor pose view directions and the mapped surface (see Fig. 3h). As a result, $\widehat{\sigma}_i$ is rich of information about all previous sensor poses.

Therefore, a parameter herein named as *corrected cumulative sampling density* ($\lambda_{i,J}$) is introduced for the definition of the objective function. $\lambda_{i,J}$ is the product of the cumulative sampling density and centrality factor ($\lambda_{i,J} = \widehat{\rho}_{i,J} * \widehat{\sigma}_{i,J}$) (see Fig. 3i). The corrected sampling density inherits its unit from $\widehat{\rho}_i$ (e.g. points/mm^2), since $\widehat{\sigma}_i$ is nondimensional. Whereas the colour of the triangles in the surface reconstruction mesh shown in Fig. 3a–d depends on the barycentres sampling density and cumulative sampling density, it depends on the centrality factor and cumulative centrality factor in Fig. 3e–h and on the corrected cumulative sampling density in Fig. 3i. It is worth highlighting that the same colormap and colour bar limits ([0, ρ^*] for Fig. 3a–d and i and [0,1] for Fig. 3e–h) are used to facilitate the comparison of the plots.

Thus, this work defines the objective function $F\left(O_{J+1}\right)$ as the difference between the theoretical number of points necessary to map the surface represented by the reconstructed mesh, with uniform target density equal to ρ^*, and the prediction of number of points sampled at the $(J+1)^{th}$ step. Indicating with a_i the area of the i^{th} triangle of the mesh, calculated through Heron's formula, we have:

$$F\left(O_{J+1}\right) = \left(\rho^* \sum_{1}^{T} a_i\right) - \left(\sum_{1}^{T} \lambda_{i,J+1} a_i\right) \tag{17}$$

where $\sum_1^T a_i$ is, recognisably, the total mesh area and $\lambda_{i,J+1}$ is the cumulative corrected sampling density, inferred through assuming a new point cloud is collected with the sensor positioned at the pose O_{J+1}.

2.3.2 Searching through the multi-dimensional space

The best next sensor pose is the pose that minimizes the objective function, given in Eq. 17. A sensor pose is a vector with six coordinates ($O = \left[o_x, o_y, o_z, o_A, o_B, o_C\right]$), being o_x, o_y and o_z the Cartesian coordinates of the sensor origin and o_A, o_B and o_C the Euler angles of the sensor reference system. Since $F\left(O_{J+1}\right)$ is a non-continuous function of six variables, it is not possible to find its minimum analytically. In this work, the multi-dimensional search space is probed through computing the value of the objective function at several test poses. The test poses are chosen conveniently, to speed up the selection of the optimum next sensor pose. The approach deployed in this work consists in offsetting the barycentres of the mesh triangles, where $\lambda_{i,J} < \rho^*$, along the triangles normals by d^* (for depth cameras) or r^* (for laser scanners). The resulting points are sorted according to the ascending order of the corrected cumulative sampling density of their parent triangles and the first K points are selected as suitable positions. This defines the poses in Cartesian coordinates. Figure 4a shows the first five test positions for the example mesh surface. The definition of the Euler angles, which describe the orientation of the test sensor poses, requires particular attention. Indeed, since the field of view of depth cameras and laser scanners does not present axial symmetry, the amount of surface a sensor can map is affected by the rotation of the sensor around its view axis. Therefore, a number (H) of different orientations of the field of view with respect to the view axis are considered for each test position, for the sake of better probing the search space. Adopting the opposite of the parent triangle normal vector as view axis direction ($\overrightarrow{w}_k$) for the k^{th} test position, the other two vectors $\overrightarrow{u}_{k,h}$ and $\overrightarrow{v}_{k,h}$ (relative to the h^{th} orientation of the sensor pose reference system with respect to $\overrightarrow{w}_k$) are computed through Rodrigues' formula [37]. Indicating with α_h the angle that defines the h^{th} orientation, it is possible to assume that the orientation at α_h and at $\alpha_h \pm \pi$ would map the same amount of surface, for depth cameras and laser scanners. It is worth noting that this assumption implies $-\vartheta_{min} = \vartheta_{max}$ and $-\theta_{min} = \theta_{max}$, for laser scanner type sensors. Therefore, in this work, α_h has been defined as:

$$\alpha_h = \pi * \frac{(h-1)}{H} \tag{18}$$

with $1 \leq h \leq H$. This produces constant-spaced test orientations in the range [0 , π). This concept is illustrated in Fig. 4b. Once the vectors of the sensor pose orthogonal reference system are known, it is straightforward to extract the Euler angles from the relative rotation matrix ($\boldsymbol{R}_{k,h} = \left[\overrightarrow{u}_{k,h} \ \overrightarrow{v}_{k,h} \ \overrightarrow{w}_k\right]$) [38].

Thus, the total number of test poses is equal to $K * H$, since we have H sensor orientations for each of the K positions. The experimental validation undertaken by this work has led to determine that $K = 20$ and $H = 5$ are good values for practical applications, resulting in a total of up to 100 test poses. All constraints given by real physical setups are considered by discarding any positions that cannot be reached by the sensor manipulator, due to kinematic limitations and or collisions. There, unsuitable positions are prevented from being used as test poses. Therefore, the number of items belonging to the set of test poses (T) may be limited by the physical constraints (robotic reachability and/or collision avoidance).

Figure 5a shows the evaluation of the objective function value at the test poses for the given example. The minimum function value is obtained at the 45^{th} test pose, relative

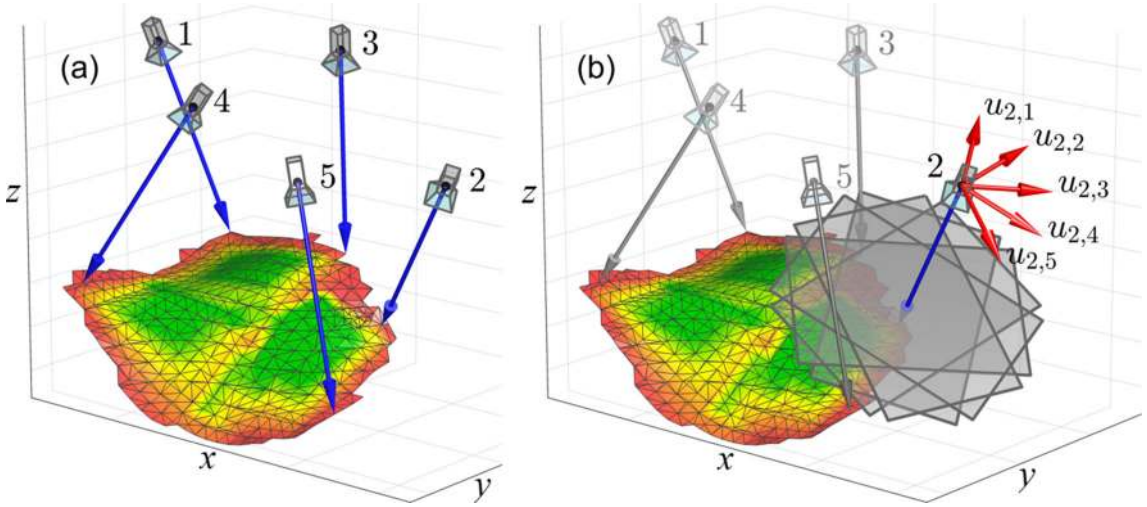

Fig. 4 The first five test view point and direction (**a**) and illustration of the selection of four sensor orientations for each selected test direction (**b**)

to k=9 and h=5 ($\alpha_h = \frac{4}{5}\pi$ radians). Then, this is taken as the next best pose (O_{J+1}). Figure 5b illustrates the sensor fiend of view at O_{J+1}. Interestingly, this approach conveniently defined the next best pose to map the portion of the objective surface that has been sampled the least by previous poses, due to the high local surface gradient.

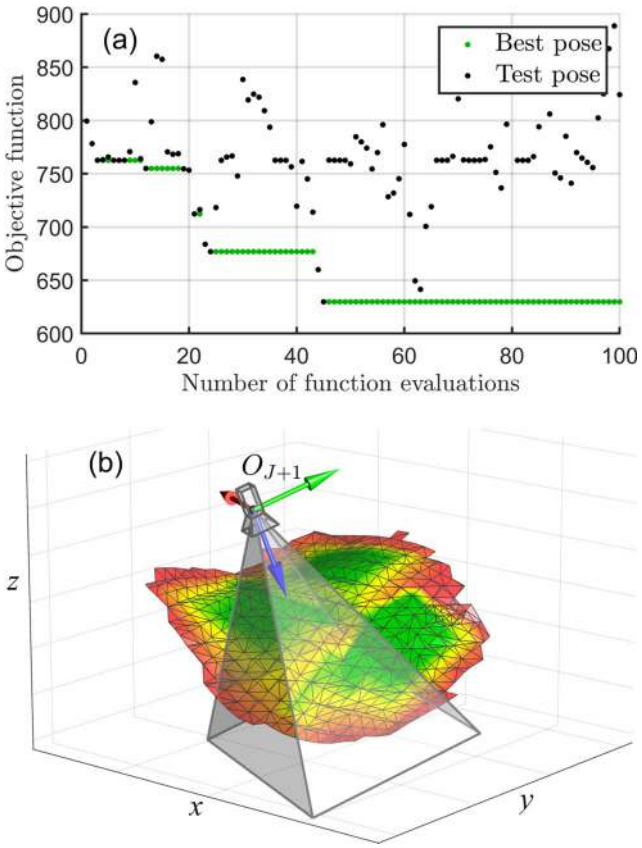

Fig. 5 Evaluation of the objective function value at the test poses (**a**) and illustration of the determined next best pose for the given example (**b**)

Undoubtedly, selecting the best next pose among a large but finite number of test poses, used to probe the objective function in the multidimensional search space, may lead to choosing a pose corresponding to a local minimum of the objective function rather than the absolute minimum. This has been deemed acceptable for the scope of this work.

2.4 Stopping criteria

Once the next best pose is defined, it is used to control the sensor manipulation system and acquire a new point cloud at the specified location. Then, the new point cloud is down-sampled and merged with the initial state point cloud and these steps can repeat again, incrementally generating a 3D reconstruction of the object of interest. Hence, it is immediate to understand the need of defining suitable stopping criteria, which regulate the interruption of the iterative reconstruction process. The described framework exposes meaningful variables that are suitable for this scope. In this work, it was deemed satisfactory to stop the iterative data capture and 3D reconstruction when the objective function (evaluated at O_J) is null or when the set of test poses is empty.

$$F(O_J) = 0 \quad \vee \quad T = \{\} \tag{19}$$

3 Experimental setup

The presented framework has been validated through simulated and real data sets. The experimental setup consists of an Intel® RealSenseTM Depth Camera D435i. It is a low-cost 3D active infrared stereo camera with expected measurement noise $\varepsilon = 0.02$ (2% of distance), a minimum depth distance of 280mm at maximum resolution (1280×720) and of 175 mm at lower resolution (640×480).

The depth camera is manipulated through a KUKA KR10-R1100-2 robot. The robot has six degrees of freedom, a reach of up to 1100 mm and a stated pose repeatability of ± 0.02 mm. Given the limited working envelope of the robot in use, the depth camera was used with a depth frame resolution of 640 $\times$ 480 points, in order to allow all-round mapping of small objects. The sensor horizontal and vertical field-of-view angles were, respectively, $\vartheta = 74°$ and $\theta = 62°$. A bespoke data acquisition software module was developed, using the Interfacing Toolbox for Robotic Arms (ITRA) [39, 40], to synchronize the robotic sensor manipulation with data collection. The depth camera data origin was calibrated as robot TCP, using the hand-eye calibration procedure described in [41]. Collision avoidance wa ensured for all the robotic trajectories, to move from any actual robot pose to the next pose, implementing the effective solution proposed in [42]. A MATLAB-based simulation environment was developed through integrating the virtual CAD model of the camera with the virtual model of the robot. In order to make the results of this work replicable and comparable with the outcomes of future investigations, an openly available computer graphics 3D test model, developed in 1994 at Stanford University [32, 43], was used. The model, often referred as Stanford Bunny consists of a tessellated surface with 69451 triangles, determined by 3D scanning a ceramic figurine of a rabbit. The model was imported in the virtual simulation environment. Figure 6 shows the real and the virtual experimental setup used for the investigations of this work.

Both the robot and a true-scale 3D printed version of the reference sample are placed onto a levelled optical table. The robot manipulator is firmly bolted onto the table by means of a 20-mm-thick steel flange. The sample is supported and raised from the table surface through an 80-mm-high plinth that positions the barycentre of the Stanford Bunny base at an offset of 435 mm along the x-axis and the y-axis and an offset of 60 mm along the z-axis, with respect to the robot base reference system. The simulation environment is a virtual twin version of the real environment.

4 Simulations

A MATLAB-based function was developed to generate a synthetic sensory point cloud for any given pose of the sensor. This was achieved by implementing a ray casting algorithm (based on [36]) to find the intersection points between the sampling directions (originating from the sensor) and the triangular mesh of the reference sample. The simulations of this work have the objective of validating the robustness of the 3D reconstruction approach. Given the stated maximum measurement noise of the utilized sensor (2% of distance), the distance between the test

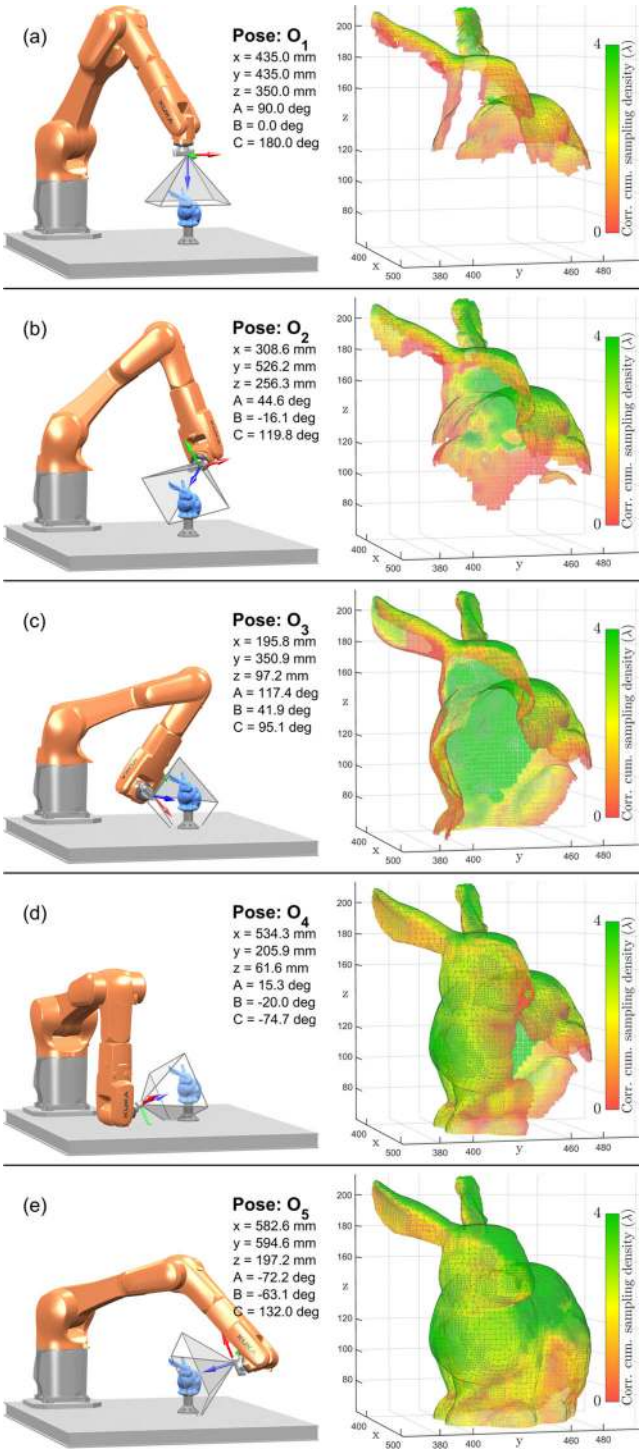

Fig. 6 Real (**a**) and virtual (**b**) experimental setup, showing the Intel® RealSense™ Depth Camera D435i, mounted onto the KUKA KR10-R1100-2 robot, and the 3D printed Stanford Bunny test model

poses and the target surface was limited to 200 mm, which gives an expected maximum deviation of 4 mm between the sampled point clouds and the real geometry. Figure 7 shows the simulated incremental 3D reconstruction of the Stanford Bunny, using the presented framework to meet

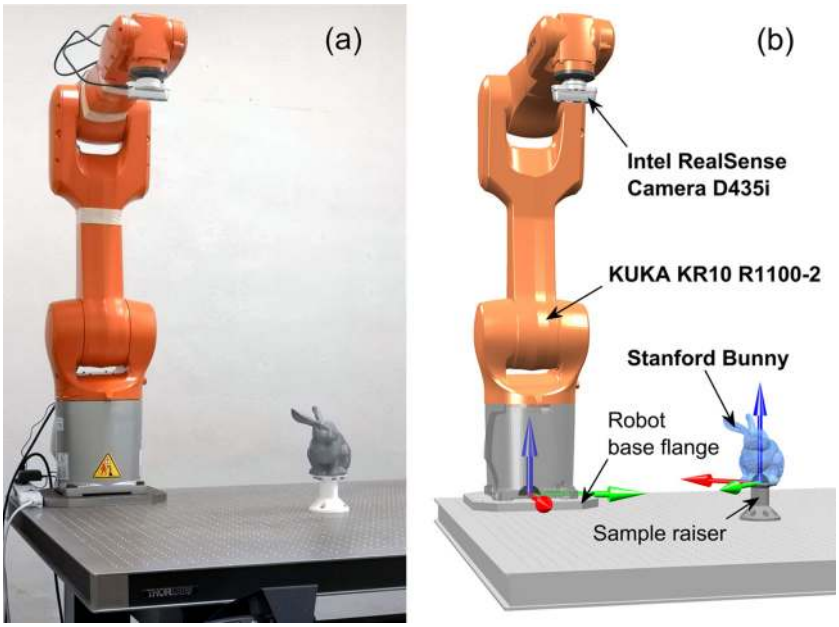

Fig. 7 Simulated full 3D reconstruction of Stanford Bunny with target density $\rho^* = 0.05$ points/mm^2, through an initial starting pose (**a**) and four autonomously generated sensor poses (**b**–**e**)

a user-specified target sampling density of $\rho^* = 0.05$ points/mm^2 (5×10^4 points/m^2). This value of target density was chosen, since it corresponds to a length of a down-sampling cube side edge $l^* = 3.76$ mm, which is similar to the expected amplitude of the measurement noise of the sensor in use, when mapping surfaces at average distance of 200 mm. Indeed, measurement noise much higher than the average distance between the points may negatively affect the accurate estimation of the surface normals. Only the first pose was defined a priori. All following poses were autonomously defined as best next poses, using the approach described in Section 2. The sensor poses were constrained to stay above the base of the sample ($o_z > 60$mm), in order to avoid collisions between the robot and the optical table and map the visible surface of the object (the whole surface excluding the sample base). Figure 7 illustrates the achieved reconstruction process, which was simulated using the pose given in Fig. 7a as starting pose. The simulation demonstrates the possibility enabled by the presented framework to reconstruct complex surface geometries, with a minimum number of effective and autonomously chosen sensor poses. The simulation was repeated using four other starting poses. All simulated reconstructions met the first stopping condition in Eq. 19, effectively reaching the target sampling density throughout the surface of the reference sample. Although the screenshots relative to these additional simulations are not presented here, in order to limit the length of this article, all relevant quantitative results are summarized in Table 1.

As it was expected, some user-specified initial poses are more convenient than others and this influences the whole reconstruction process. This causes the number of necessary sensor poses to vary. Nevertheless, it is interesting to note that all simulated reconstructions led to very similar results, in terms of number of points in the down-sampled cloud, number of triangles in the reconstruction mesh and extension of the mapped surface, despite of the difference in the starting pose and consequent next best poses used

Table 1 Simulations quantitative results. The first column relates to the simulation illustrated in Fig. 7. The following columns regard the other simulations, which were run using different starting poses

Initial pose (coordinates in [mm] and angles in [deg])	x = 435 y = 435 z = 350 A = 95 B = 0 C = 180	x = 635 y = 435 z = 150 A = -90 B = 0 C = 90	x = 435 y = 635 z = 150 A = 0 B = 0 C = 90	x = 235 y = 435 z = 150 A = 90 B = 0 C = 90	x = 435 y = 235 z = 150 A = 180 B = 0 C = 90
Num. poses required	5	5	4	6	7
Num. raw points	231707	227070	191858	297023	343165
Down-sampled points	5131	5234	5109	5242	5256
Num. mesh triangles	35433	35313	35133	37459	36389
Reconstructed surface [mm^2]	51163	52106	51206	51824	51904

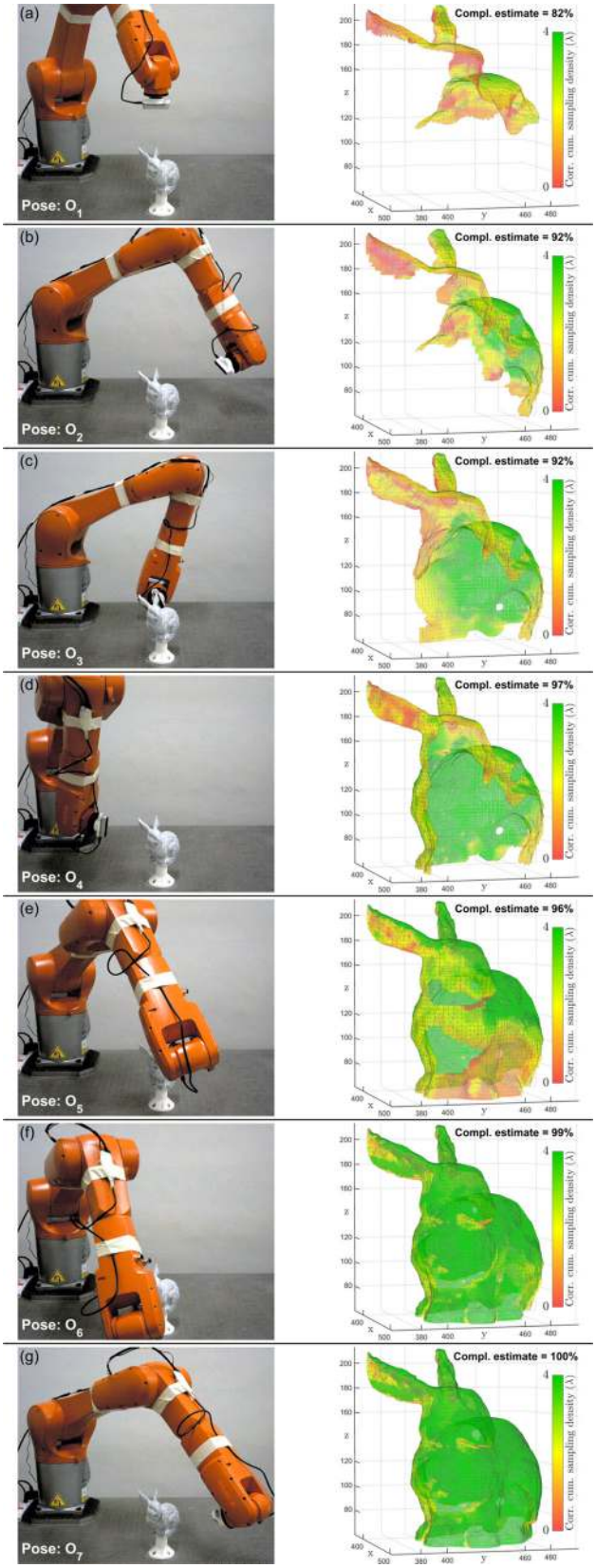

Fig. 8 Full 3D reconstruction of Stanford Bunny with target density $\rho^* = 0.05$ points/mm^2, through an initial starting pose (**a**) and six autonomously generated sensor poses (**b**–**g**)

in the reconstruction pipeline. The values of the mapped surface extension are very close to the area of the Stanford Bunny surface (excluding its base), which is 51954 mm^2, as measured from the reference sample original tessellated mesh. The small deviations, between the extension of the reconstructed surfaces and the reference area, are smaller than 1.6%. They are thought to have been caused by the fact that the reference virtual model and the reconstructed model are, obviously, not represented by the same set of triangulated points.

5 Sensor data results

Real-data reconstructions were undertaken by means of the physical laboratory setup described in Section 3. Figure 8 illustrates the reconstruction of the Stanford Bunny, with target sampling density $\rho^* = 0.05$ points/mm^2, using the first pose in Fig. 7 as initial sensor pose. The real system required a total of seven poses to obtain the full reconstruction of the reference sample, which exceed the respective simulation by two poses. This is caused by the fact that the real sensor typically fails to return some of the surface points that are within the sensor field of view. This is evident if one compares Fig. 8a with Fig. 7a. The extension of the surface mapped through the real data in Fig. 8a is smaller than the ideal reconstruction relative to the same view pose, given in Fig. 7a. The variable reflectivity of the sample causes some areas of the surface to reflect too little or too much light, impeding accurate sampling (within the sensor acceptance thresholds). This leads to the deviation of the real deployed sensor poses from the simulated poses. It is worth pointing that more sophisticated sensors, capable of returning less compromised point clouds, would produce better adherence with simulated pose coordinates and pose sequencing. Nevertheless, the real data reconstructions

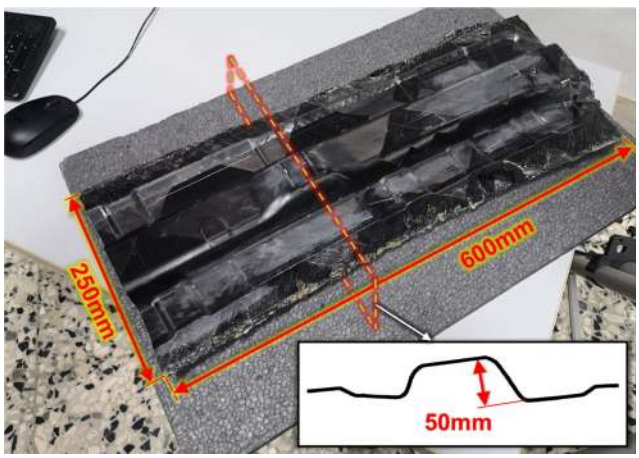

Fig. 9 CFRP automotive sample used as additional test case

performed in this work proved the capability of the proposed framework to flexibly adapt to real scenarios and different starting poses and to be used with low-cost sensors.

In order to further demonstrate the flexibility of the proposed framework, an industrial specimen was reconstructed using the same data acquisition setup. The specimen was a 4-mm-thick carbon fibre reinforced plastic (CFRP) shell sample, moulded into a curved contour by the automotive industry. Composite parts often suffer geometry distortion due to their elastic spring back when they are extracted from the curing mould, which makes geometry mapping a requirement for dimensional assessment or for programming successive robotic machining. The sample had a rectangular size of circa 250 × 600 mm (Fig. 9). For the curvatures of the sample surface, this specimen was deemed representative of the challenging geometries often found in composite samples, where the mapping of the lateral surface of stiffening stringers and ribs requires bespoke sensor view pose planning. Figure 9 shows the contour of the sample surface for the section corresponding to the maximum geometry height.

The sample was uniformly sprayed with a removable white matte powder (Spray-Rotrivel U, manufactured by CGM s.r.l), which gave an approximately Lambertian finish with a reflectance spectrum flat in the visible spectral region [44]. This maximized the mapping performance of the depth camera in use. Figure 10 shows the reconstruction of the test sample through the approach presented in this work. The first point cloud was acquired through a user-defined pose, capturing the central part of the sample (Fig. 10a). The target sampling density (0.05 points/mm^2) was achieved throughout the sample surface, through eleven successive autonomously computed poses (Fig. 10b–l).

The resulting reconstructed surface was compared with the ground-truth point cloud, which was acquired by a Hexagon ROMER Absolute Arm RA-7520SE (Fig. 11). This is a metrology tool, based on a passive arm equipped with a laser profiler and high-accuracy encoders. The stated precision of the scanning system is 53μm. Figure 11b shows the deviation map, between the reconstructed geometry and the ground-truth point cloud. The deviations are within the expected range of 0–4 mm, since the sensor had an accuracy of 2% and the average sensor standoff used for the data collection was set to 200 mm. Nevertheless, the discontinuities in the error distribution in the deviation map

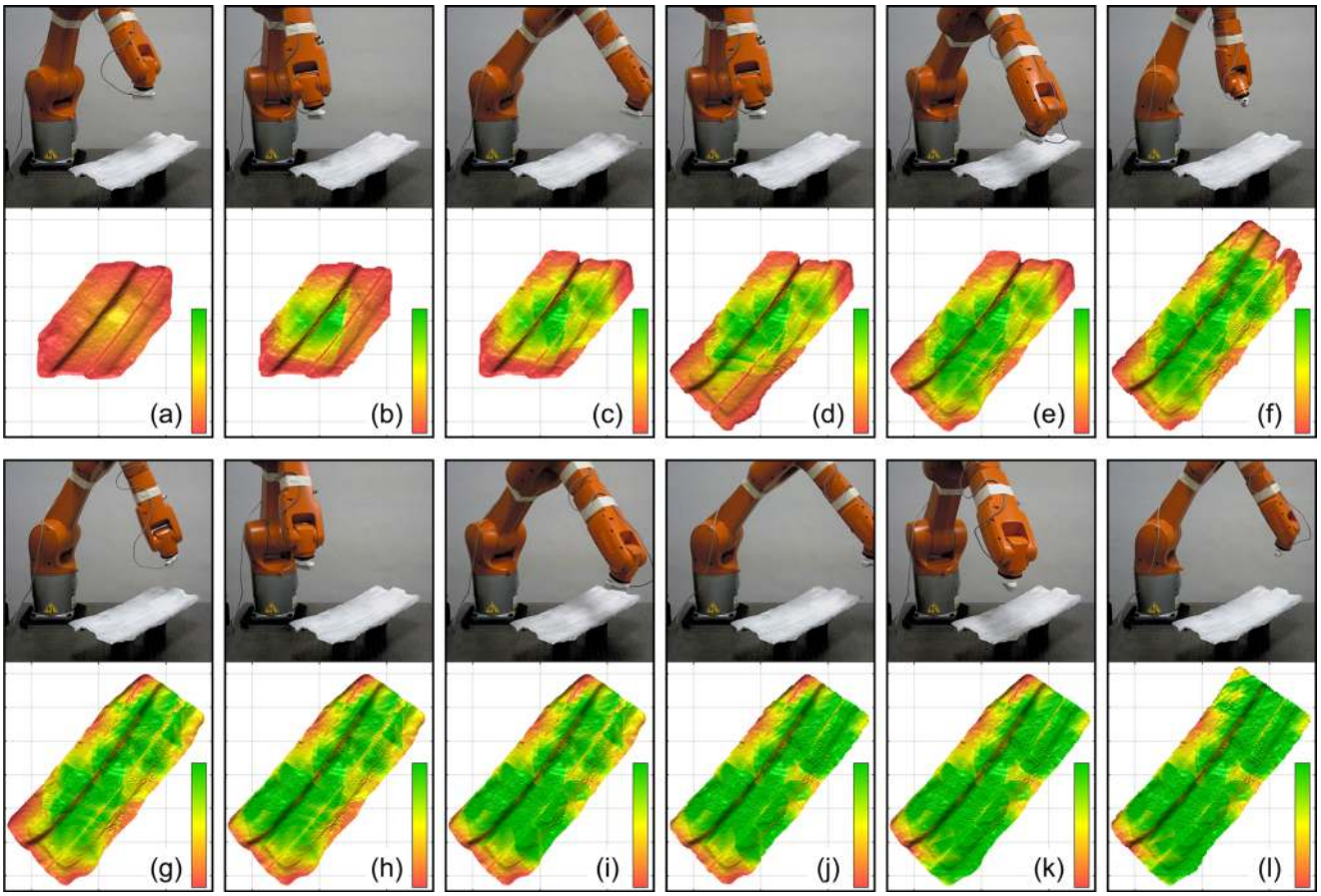

Fig. 10 Full 3D reconstruction of CFRP automotive test sample with target density $\rho^* = 0.05$ points/mm^2, through an initial starting pose (**a**) and eleven autonomously generated sensor poses (**b–l**)

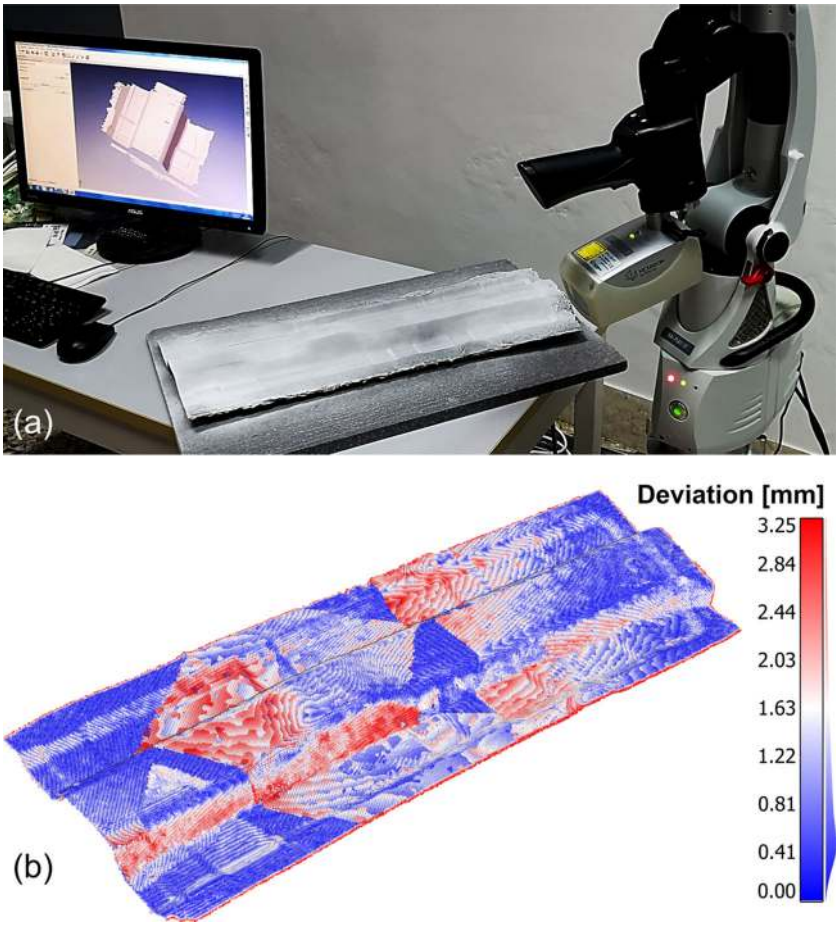

Fig. 11 Acquisition of ground-truth point cloud through the Hexagon ROMER Absolute Arm (**a**). Map of deviation between the reconstructed geometry and the ground-truth (**b**)

seems to suggest that it may also be partially caused by the propagation of the inaccuracy in the calibration of the robot TCP (the camera centre) onto the registration of the point clouds.

6 Conclusions and future work

Several applications require a digital model of an object to create a virtual twin of the part and/or to inform automated systems that need to interact with it. In most situations, the acquisition of a single point cloud from one point of view cannot produce a complete 3D reconstruction of an object. Multiple point clouds, collected from different poses are typically required. Manual determination of optimal view poses for surface scanning is time-consuming and expert-dependent. Moreover, when the scanning sensor is manipulated by a robotic arm, it is necessary to consider the robot kinematic constraints and avoid collisions. Finding the optimum set of view poses for a robot-manipulated 3D scanning system, in order to efficiently reconstruct a given object using the minimum number of views, is still an open problem. This article presented a mathematical framework for automating the 3D reconstruction of specimens. The approach is suitable to be used with two large families of 3D scanners: depth cameras and laser scanners. Compared with previous works, the presented framework does not need a priori information about the shape of the object, since it incrementally creates and updates the digital reconstruction of the part. The method allows mapping the surface of an object to meet a user-defined target sampling density. Efficient incremental down-sampling and merging is performed in a single pass, through an indexing algorithm that minimizes the computational effort. The framework code is made publicly available, at https://doi.org/10.5281/zenodo.4646850, and can be used by the research community for future developments. The robustness of the approach was tested through simulated data. In order to validate the framework in experimental scenarios, a computer was interfaced with a robot arm and an RGB-D camera to reconstruct the geometry of a 3D printed version of a reference test model and of an industrial test piece. The investigations proved the capability of the proposed framework to flexibly adapt to real scenarios and different starting view poses and to be used with low-cost sensors.

The selection of the best next pose among a large but finite number of test poses, used to probe the objective function in the multidimensional search space in this work,

may lead to choosing a pose corresponding to a local minimum of the objective function rather than the absolute minimum. Although this has been deemed acceptable for the scope of this work, future work should focus on enhancing the ability to converge to deployable poses corresponding to the absolute minimum of the objective function for all sampling steps.

Author contribution C.M. conceived and developed the theory, performed data acquisition and processing and wrote the article manuscript. D.C. supervised the findings of this work at all stages. V.R. performed the acquisition of the ground-truth point cloud of the industrial CFRP specimen. B.R. provided the industrial CFRP specimen. All authors discussed the results and contributed to the review of the final manuscript

Funding Open access funding provided by Università degli Studi di Palermo within the CRUI-CARE Agreement. This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement no. 835846.

Availability of data and material Reference model available through the Stanford 3D Scanning Repository (http://graphics.stanford.edu/data/3Dscanrep/#bunny).

Code availability https://doi.org/10.5281/zenodo.4646850.

Declarations

Ethics approval This work did not involve human subjects and/or animals. Thus, no ethical approval was required.

Consent to participate This work did not involve collection of information from human subjects.

Consent for publication This work did not involve collection of information from human subjects. All authors of this work have expressed consent for their names and affiliations to appear in this journal publication.

Conflict of interest The authors declare no competing interests.

References

1. Kumar A (2018) Methods and materials for smart manufacturing: additive manufacturing, internet of things, flexible sensors and soft robotics. Manufacturing Letters 15:122–125
2. Willette A, Brell-Cokcan S, Braumann J (2014) Robotic fabrication in architecture, art and design 2014. Springer, Cham. https://doi.org/10.1007/978-3-319-04663-1
3. Ingrassia T, Nigrelli V, Ricotta V, Tartamella C (2017) Process parameters influence in additive manufacturing. In: Advances on Mechanics, Design Engineering and Manufacturing. Springer, pp 261–270
4. Mineo C, Pierce S, Wright B, Cooper I, Nicholson P (2015) PAUT inspection of complex-shaped composite materials through six DOFs robotic manipulators. Insight-Non-Destructive Testing and Condition Monitoring 57:161–166
5. Mineo C, Pierce SG, Nicholson PI, Cooper I (2016) Robotic path planning for non-destructive testing – A custom MATLAB toolbox approach. Robot Comput Integr Manuf 37:1–12
6. Marturi N et al (2016) Towards advanced robotic manipulations for nuclear decommissioning. In: 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA). IEEE, Amritapuri (India)
7. Burrell T, West C, Monk SD, Montezeri A, Taylor CJ (2018) Towards a cooperative robotic system for autonomous pipe cutting in nuclear decommissioning. IEEE, Sheffield (United Kingdom)
8. Ahmad H (2004) Feasibility study on robot off-line programming and simulation using matlab tools: simmechanics and simulink packages. Dissertation, Universiti Tun Hussein Onn
9. Mineo C, Pierce SG, Nicholson PI, Cooper I (2017) Introducing a novel mesh following technique for approximation-free robotic tool path trajectories. J Comput Des Eng 4(3):192–202
10. Andersen RS, Bøgh S, Moeslund TB, Madsen O (2015) Intuitive task programming of stud welding robots for ship construction. In: International Conference on Industrial Technology (ICIT). IEEE, Seville (Spain)
11. Fang Z, Xu D, Tan M (2010) A vision-based self-tuning fuzzy controller for fillet weld seam tracking. IEEE/ASME Trans Mechatron 16(3):540–550
12. Bi Z, Kang B (2014) Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly. Enterprise Inf Sys 8(2):225–245
13. Bitzidou M, Chrysostomou D, Gasteratos A (2012) Multi-camera 3D object reconstruction for industrial automation. Springer, Rhodos (Greece)
14. Kulikajevas A, Maskeliūnas R, Damaš evičius R, Ho ES (2025) 3D object reconstruction from imperfect depth data using extended YOLOv3 network. Sensors 20:7
15. Curless B (1999) From range scans to 3D models. ACM SIGGRAPH Computer Graphics 33(4):38–41
16. Vermeulen M, Rosielle P, Schellekens P (1998) Design of a high-precision 3D-coordinate measuring machine. CIRP Annals 47(1):447–450
17. Chen S, Li Y, Kwok NM (2011) Active vision in robotic systems: A survey of recent developments. Int J Robot es 30(11):1343–1377
18. Fossum ER (1997) CMOS image sensors: Electronic camera-on-a-chip. IEEE Trans Electron Devices 44(10):1689–1698
19. Abe T, Sensui T (2007) Stereo camera. US Patent 7 190 389, USA
20. Litomisky K (2012) Consumer rgb-d cameras and their applications, University of California. http://alumni.cs.ucr.edu/klitomis/files/RGBD-intro.pdf. Accessed 31 March 2021
21. Gerald GF, Stutz GE (2004) Marshall Handbook of Optical and Laser Scanning, Taylor & Francis. https://doi.org/10.1201/9781315218243
22. Kilpelä A, Pennala R, Kostamovaara J (2001) Precise pulsed time-of-flight laser range finder for industrial distance measurements. Rev Sci Instrum 72(4):2197–2202
23. Journet BA, Poujouly S (1998) High-resolution laser rangefinder based on a phase-shift measurement method. In: Three-dimensional imaging, optical metrology, and inspection, IV Ed,

International Society for Optics and Photonics, pp 123-132. https://doi.org/10.1117/12.334326
24. Bernardini F, Rushmeier H (2002) The 3D model acquisition pipeline. In: Computer graphics forum, Wiley Online Library, vol 21(2), pp 149-172. https://doi.org/10.1111/1467-8659.00574
25. Kraus K (2011) Photogrammetry: geometry from images and laser scans. Walter de Gruyter, Germany
26. Khan A, Mineo C, Dobie G, Macleod C, Pierce G (2020) Vision guided robotic inspection for parts in manufacturing and remanufacturing industry. Journal of Remanufacturing. https://doi.org/10.1007/s13243-020-00091-x
27. Engin S, Mitchell E, Lee D, Isler V, Lee DD (2020) Higher order function networks for view planning and multi-view reconstruction. In: 2020 IEEE International conference on robotics and automation (ICRA). IEEE, Piscataway (USA). https://doi.org/10.1109/ICRA40945.2020.9197435
28. Landgraf C, Meese B, Pabst M, Martius G, Huber MF (2021) A reinforcement learning approach to view planning for automated inspection tasks. Sensors. https://doi.org/10.3390/s21062030
29. Scott WR, Roth G, Rivest J (2003) View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys (CSUR) 35(1):64–96
30. Kaba MD, Uzunbas MG, Lim SN (2017) A reinforcement learning approach to the view planning problem. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu (Hawaii)
31. Jing W et al (2018) A computational framework for automatic online path generation of robotic inspection tasks via coverage planning and reinforcement learning. IEEE Access 6:54854–54864
32. Riener R, Harders M (2012) Virtual reality in medicine. Springer, London. https://doi.org/10.1007/978-1-4471-4011-5
33. Weinmann M, Jutzi B, Hinz S, Mallet C (2015) Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J Photogramm Remote Sens 105:286–304
34. Cao YP, Kobbelt L, Hu SM (2018) Real-time high-accuracy three-dimensional reconstruction with consumer RGB-D cameras. ACM Transactions on Graphics (TOG) 37(5):1–16
35. Kazhdan M, Chuang M, Rusinkiewicz S, Hoppe H (2020) Poisson surface reconstruction with envelope constraints. Computer Graphics Forum, Wiley Online Library 39(5):173-182
36. Möller T, Trumbore B (1997) Fast, minimum storage ray-triangle intersection. Journal of Graphics Tools 2(1):21–28
37. Mebius JE (2007) Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations. Xiv General Mathematics. arXiv:math/0701759. Accessed 31 March 2021
38. Slabaugh GG (1999) Computing Euler angles from a rotation matrix. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.371.6578&rep=rep1&type=pdf. Accessed 31 March 2021
39. Mineo C, et al. (2019) Interfacing toolbox for robotic arms with real-time adaptive behavior capabilities. University of Strathclyde. https://doi.org/10.17868/70008
40. Mineo C et al (2020) Enabling robotic adaptive behaviour capabilities for new industry 4.0 automated quality inspection paradigms. Insight-Non-Destructive Testing and Condition Monitoring 62(6):338–344
41. Khan A, Aragon-Camarasa G, Sun L, Siebert JP (2016) On the calibration of active binocular and RGBD vision systems for dual-arm robots. In: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, Qingdao (China)
42. Wong C, Mineo C, Yang E, Yan XT, Gu D (2020) A novel clustering-based algorithm for solving spatially-constrained robotic task sequencing problems. IEEE/ASME Transactions on Mechatronics. https://doi.org/10.1109/TMECH.2020.3037158
43. Turk G, Levoy M (1994) Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques. Association for Computing Machinery, New York, pp 311-318
44. Lu R (2017) Light scattering technology for food property, quality and safety assessment. Contemporary Food Engineering Series. CRC Press, Boca Raton. https://doi.org/10.1201/b20220