

View Planning for Robotic Inspection of Tolerances Through Visual Tracking of Manual Surface Finishing Operations

Eirik B. Njaastad

*Department of Mechanical and Industrial Engineering,
Norwegian University of Science and Technology (NTNU),
Richard Birkelands vei 2B, Trondheim NO-7491, Norway
eirik.njaastad@ntnu.no*

Olav Egeland

*Department of Mechanical and Industrial Engineering,
Norwegian University of Science and Technology (NTNU),
Richard Birkelands vei 2B, Trondheim NO-7491, Norway
olav.egeland@ntnu.no*

Received (25 July 2019)

Accepted (29 August 2019)

This article presents an approach for determining suitable camera view poses for inspection of surface tolerances based on visual tracking of the tool movements performed by a skilled worker. Automated surface inspection of a workpiece adjusted by manual operations depends on manual programming of the inspecting robot, or a time-consuming exhaustive search over the entire surface. The proposed approach is based on the assumption that the tool movements of the skilled worker coincide with the most relevant regions of the underlying surface of the workpiece, namely the parts where a manual process has been performed. The affected region is detected with a visual tracking system, which measures the motion of the tool using a low-cost RGBD-camera, a particle filter, and a CAD model of the tool. The main contribution is a scheme for selecting relevant camera view poses for inspecting the affected region using a robot equipped with a high-accuracy RGBD-camera. A principal component analysis of the tracked tool paths allows for evaluating the view poses by the Hotelling's T-squared distribution test in order to sort and select suitable camera view poses. The approach is implemented and tested for the case where a large ship propeller blade cast in NiAl bronze is to be inspected by a robot after manual adjustments of its surface.

Keywords: Manufacturing Automation; Inspection Planning; Computer Vision for Manufacturing; Visual Tracking.

1. Introduction

For small batch production with frequent change of product dimensions and product types, it is vital that the product tolerance inspection can be planned and executed efficiently and accurately without unnecessary effort. Robotic inspection with optical sensors can significantly alleviate the problem of achieving sufficient quality assurance of the fabricated products in a minimum amount of time. The workpiece geometry and any deviations are then possible to determine by observing

the workpiece surface from various angles and distances with a 3D camera held by a robot, or by letting the robot carry a line scanner along the surface. Then an effective inspection process depends on the robot being programmed to scan the relevant parts of the surface, with the optical sensor being placed sufficiently close to the surface during scanning in order to achieve the required accuracy.

When a CAD model of the workpiece is available, definitions of the part's geometry can be retrieved and used as a basis for automatically determining necessary sensor view poses in order to obtain a complete and accurate 3D image of a surface or the whole part [1,2]. This bridge between CAD/CAM and automated inspection is called Computer-aided inspection planning (CAIP) [3,4]. CAD-based tolerance inspection originates from software developed for Coordinate Measuring Machines (CMM), but its application has spread, and it has proven to be suitable for inspections performed by 6 DoF robotic arms coupled with 3D optical scanners [5,6]. A thorough review of suggested and applied methodologies and systems can be found in [3,7,8].

In scenarios where the geometry of the workpiece is unknown, typically if there is no CAD model of the workpiece available, the challenge turns into a search for determining suitable sensor view poses while exploring the unknown object. There is much previous work on robotic exploring and automatic determination of sensor view poses. In computer vision literature, the problem of determining the best camera poses is named Next Best View (NBV) planning [9]. Typical uses of NBV includes determining the minimum amount of view poses necessary for a complete 3D capture of an object [10], or finding a sequence of views which allows optimal reconstruction of an object [11], i.e., which minimizes some reconstruction error metric. NBV approaches are classified as online or offline methods, furthermore as model-based or non-model based methods.

If the product to be inspected is large, a complete scan of its surface may be time-consuming. Moreover, only parts of the product may be required to undergo an inspection, like a weld seam or some parts of the surface that have undergone manual surface finishing. In this case, it may be advantageous to track the tool paths of the manual operation and then use this as input for the robotic inspection. Provided that the tool paths allow for selecting suitable view poses, the inspection can achieve higher performance by focusing only on the affected areas, especially when a manual surface finishing process is to be inspected.

There are several successful methods available for contacting and non-contacting tracking of the tool movements of skilled workers. Motion capturing systems like exoskeletal systems are an example of a well-known approach. However, the contacting methods may hinder the human motion or the production process. Non-contacting alternatives such as vision-based techniques may not hinder human gestures [12]. Fiducial markers have proved to be successful in vision-based interfaces between humans and robots for multiple applications, including teleoperation, where markers placed on the hand of a human operator can remotely control a robot [13]. It may, however, be favorable to avoid interfering with the process when recording it, e.g.,

by avoid using markers that are attached to the tool while the worker is performing the process [14].

A research topic of constant relevance is the challenge of capturing surface finishing processes and other complex actions from human demonstration [15–19]. Automatic inspection of a processed workpiece based on tool paths learned from a skilled worker is, however, not much reported in the literature.

In this paper, we propose a new approach to robotic inspection. We intend to focus the inspection on the parts of the workpiece that has undergone manual surface geometry corrections by a skilled worker. The main contribution is a scheme for selecting the most relevant inspection view poses from the tool paths that are performed by the worker. The paper is an extended and revised version of a preliminary conference article that was presented on IEEE IRC 2019 [20].

We suggest using a low-cost RGBD-camera for tracking the tool movements of the skilled worker. In order to handle occlusions and cluttered environments, we employ a particle filter for tracking a CAD model of the tool and recording the time history of the tracked tool movements. Based on the recorded tool paths, a robot holding a second, high-accuracy RGBD-camera inspects the relevant sections of the workpiece. An implementation of a working system is studied through simulations and experiments, and its performance is investigated and validated.

The rest of the paper is organized as follows: Section 2 presents the particle filter used for tracking the tool movements of the skilled worker. In Section 3, a description of the proposed motion tracking and inspection planning approach is given, while an analysis of the computational performance of the algorithm is described in Section 4. Simulated and experimental verification of the overall scheme is detailed in Section 5, followed by a summary of the approach and proposals for further work in Section 6.

2. Preliminaries

When tracking the manual surface adjusting operations, the recorded demonstration consists of a tool path $T = \{t_0, \dots, t_{n-1}\}$ which represents the $n \in \mathbb{N}$ samples (via-points) of the tool center point t_i in the form of a unit quaternion and accompanying translation vector. We employ a Rao-Blackwellized particle filter for tracking the tool movements of the skilled worker [21].

2.1. Particle filter tracking

The Rao-Blackwellized Particle Filter (RBPF) [22] improves the performance of particle filtering by sampling over a subspace of the probability distribution of the state. RBPF is based on the assumption that it is reasonable to evaluate some filtering equations analytically and the remaining with a particle filter rather than computing everything with pure sampling.

Let z_k be a Markov process of initial distribution $p(z_0)$ and transition equation $p(z_k|z_{k-1})$. The set of observations $\{y_1, \dots, y_k\}$ is denoted by $y_{1:k}$. It is possi-

ble to divide the hidden variables z_k into two groups called r_k and x_k , such that $p(z_k|z_{k-1}) = p(x_k|r_{k-1:k}, x_{k-1})p(r_k|r_{k-1})$ and, depending on $r_{0:k}$, the conditional posterior distribution $p(x_{0:k}|y_{1:k}, r_{0:k})$ is thus possible to handle analytically [23]. As a result, we can marginalize out $x_{0:k}$ from the posterior, and only need to focus on estimating $p(r_{0:k}|y_{1:k})$, which now is in a lower-dimensional space. In other words: $p(r_{1:k}|y_{1:k})$ can be predicted with a particle filter, while $p(x_k|r_{1:k}, y_{1:k})$ can be updated using a Kalman filter.

At time step $k - 1$ it is assumed that an approximation to the filter density is given by

$$p(r_{1:k-1}|y_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^{(i)} \delta(r_{1:k-1} - r_{1:k-1}^{(i)}) \quad (1)$$

where $\delta(\cdot)$ is the Dirac delta function, and the weights $w_{k-1}^{(i)}$ are scalars, and $r_{k-1}^{(i)} \in \mathbb{R}^n$. Given N particles (samples) $\{r_{0:k-1}^{(i)}, x_{0:k-1}^{(i)}\}$ at time $k - 1$, approximately distributed according to the distribution $p(r_{0:k-1}^{(i)}, x_{0:k-1}^{(i)}|y_{0:k-1})$, the RBPF is used to compute N particles $(r_{0:k}^{(i)}, x_{0:k}^{(i)})$ approximately distributed according to the posterior $p(r_{0:k}^{(i)}, x_{0:k}^{(i)}|y_{1:k})$, at time k . This is accomplished with the following steps:

(1) Sequential Importance Sampling

- For each particle i a new state vector is predicted by taking a sample: $r_k^{(i)} \sim p(r_k|r_{0:k-1}^{(i)}, y_{1:k})$.
- For each particle i , the importance weights are evaluated up to a normalizing constant

$$w_k^{(i)} = \frac{p(r_{0:k}^{(i)}|y_{1:k})}{q(r_k^{(i)}|r_{0:k-1}^{(i)}, y_{1:k})p(r_{0:k-1}^{(i)}|y_{1:k-1})}. \quad (2)$$

- For each particle i , the importance weights are normalized

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}. \quad (3)$$

(2) Selection

- Multiply samples $r_{0:k}^{(i)}$ with high importance weights $\tilde{w}_k^{(i)}$ to obtain N random samples $\tilde{r}_{0:k}^{(i)}$ distributed approximately according to $p(\tilde{r}_{0:k}^{(i)}|y_{1:k})$. In the same way, samples with low importance weights $\tilde{w}_k^{(i)}$ are suppressed.

(3) Markov Chain Monte Carlo (MCMC)

- The set of particles is resampled using MCMC. A Markov transition kernel is applied with invariant distribution, which is given by $p(r_{0:k}^{(i)}|y_{1:k})$ in order to obtain $r_{0:k}^{(i)}$. We can now go to the next time-step and repeat the procedure.

Finally, we have the particle filter approximation at time step k :

$$p(r_{1:k}|y_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta(r_{1:k} - r_{1:k}^{(i)}) \quad (4)$$

3. Motion Tracking and View Planning

The goal of the approach is to enable the inspection robot to execute a scanning program based on the tool paths learned from a skilled worker. Recorded tool paths are assumed to follow the surface of the object to inspect. The robot will thus be able to focus the inspection to the surface sections on which the worker has performed manual processing.

3.1. System overview

An overview of the proposed approach is shown in Fig. 1. We start by capturing the tool paths with the particle filter described in Section 2. Input to the particle filter is depth images from a consumer-grade and low-accuracy RGBD-camera at 30 Hz, combined with a CAD model of the tool to be tracked. After various filtering of the recorded tool paths, the scanning view poses are selected based on a set of basic camera parameters: Its field of view and optimal scanning distance.

The cameras and robot are calibrated with respect to each other using an offline calibration procedure.

3.2. Tool path filtering

Many surface treatment processes consist of repetitive movements over the same surface patch. The tracked tool path would then be too detailed and ineffective as input for the inspection robot. A typical learned process tool path could potentially contain several thousand tracked tool poses. It would be a highly time-consuming task for the inspecting robot to visit and scan all of them. In many cases, the camera can cover large parts of the tracked path simultaneously. It is then necessary to use filtering methods in order to refine the inspection process.

Our primary strategy for filtering the view poses is to divide them into a voxel grid structure, effectively grouping nearby view poses and replacing them with their centroid. This step ensures that repetitive tool movements are filtered and combined into a set of unique view poses. By adjusting the voxel size, the resolution of the subsequent selection of camera poses can be adjusted. Increasing the grid resolution comes with a computational cost.

Before the voxel grid filtering step, we do a simple Gaussian smoothing of the recorded data, in both the forward and reverse directions.

3.3. Camera orientation by averaging quaternions

Distributing the recorded tool poses into a voxel grid does not involve determining a camera orientation for the camera viewpoints. For each centroid camera position

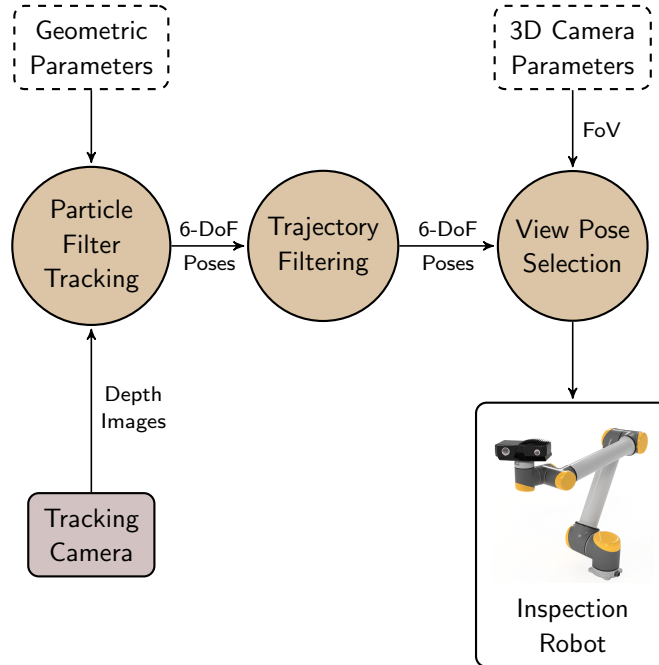


Fig. 1: Overview of the information flow in the proposed approach. Inputs to the tracking process are depth images from a low-accuracy RGBD-camera with the workspace in its field of view, and a CAD model of the tool to be tracked. In order to generate the camera views for inspection, information about depth range and field of view of the second RGBD-camera used for inspection is necessary.

present in the voxel grid, a camera orientation must be chosen.

A simple strategy for selecting the camera orientation would be to choose the view orientation from the viewpoint laying closest to the center of each of the voxels in the voxel grid. The information about view orientation from all the disregarded original viewpoints would then be neglected.

We use the average of the local set of viewpoints around each filtered pose in order to account for more of the local orientation information. We employ a quaternion based fast averaging technique, as proposed in [24]. The local group is determined by doing a linear k-nearest neighbor search for each of the poses present in the voxel grid.

Given \mathbf{q}_i , a set of quaternions, we form the weighted dot product matrix:

$$\mathbf{B} = \frac{1}{n_q} \sum_{i=1}^{n_q} w_i^q (\mathbf{q}_i^T \cdot \mathbf{q}_i) \quad (5)$$

where n_q is the number of poses in the local group, and w_i^q is the associated weights,

given the pose \mathbf{q}_i . The mean quaternion \mathbf{q}_{avg} is given by the eigenvector \mathbf{e}_{max} corresponding to the maximum eigenvalue of \mathbf{B} , λ_{max} .

3.4. Hotelling's T -squared distribution

We employ a multivariate statistical distribution in order to select the most relevant view poses for inspection. The Hotelling's T^2 values represent a measure of the variation in each sample within the model [25]. It indicates how far each sample is from the center of the model.

The statistics are calculated for a principal component analysis model (PCA), which is a well-established technique for unsupervised dimensionality reduction [26]. PCA is used for data compression and information extraction. The idea behind is to reduce the dimensionality of the original data by forming a new set of latent variables which are a linear combination of the original data, without losing essential information. PCA explains the amount of variability in the data.

The T^2 value for the i^{th} observation is defined as:

$$T^2 = \sum_{a=1}^{a=A} \left(\frac{t_{i,a}}{s_a} \right)^2 \quad (6)$$

where the s_a^2 values are constants and are the variances of each component. A is the number of PCA components, with accompanying score $t_{i,a}$. The T^2 measure can be interpreted as a scalar number that summarizes all the score values. The method requires that the data displays a normal distribution.

After calculating the T^2 , we sort the view poses in descending order and uses the resulting list as input for checking overlap between adjacent camera views.

3.5. Camera overlap calculation

In order to reduce the number of camera views such that there is only a minimum camera overlap, we iterate through all view pose candidates generated from the statistical selection step. An illustration of the geometric method of comparison is shown in Fig. 2. The camera views are modeled as pyramids with sides and height corresponding to the field of view and optimal scanning distance of the inspecting RGBD-camera.

Camera view overlap is determined using a Jaccard similarity coefficient between adjacent camera views in voxel space, where a fraction of superimposed voxels expresses the overlap. Each attribute of C_k and C_{k-1} can either be 0 or 1.

$$\begin{aligned}
J(C_k, C_{k-1}) &= \frac{|C_k \cap C_{k-1}|}{|C_k \cup C_{k-1}|} \\
&= \frac{|C_k \cap C_{k-1}|}{|C_k| + |C_{k-1}| - |C_k \cap C_{k-1}|} \\
&= \frac{\sum_{i=1}^n (C_k[i] \cdot C_{k-1}[i])}{\sum_{i=1}^n (C_k[i]) + \sum_{i=1}^n (C_{k-1}[i]) - \sum_{i=1}^n (C_k[i] \cdot C_{k-1}[i])}
\end{aligned} \tag{7}$$

The similarity coefficient is defined $0 \leq J(C_k, C_{k-1}) \leq 1$.

View pose candidates with a similarity coefficient over a specified threshold are disregarded. A list of reasonable camera view poses for automated inspection then emerges.

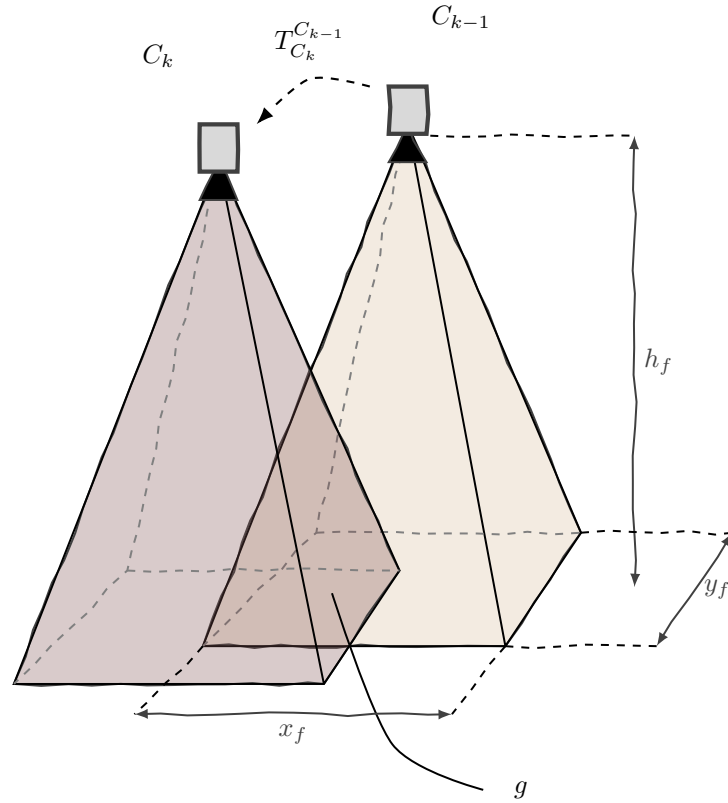
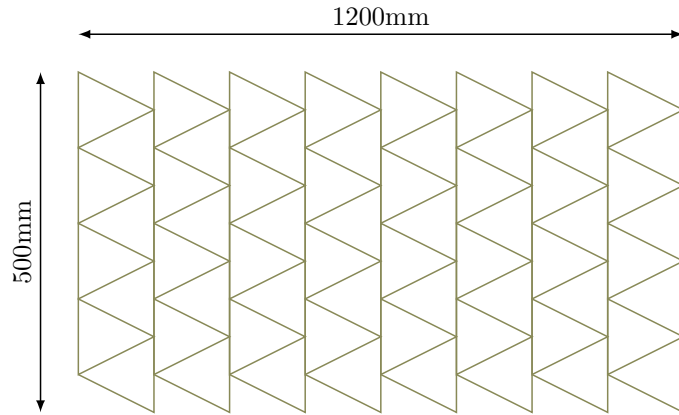


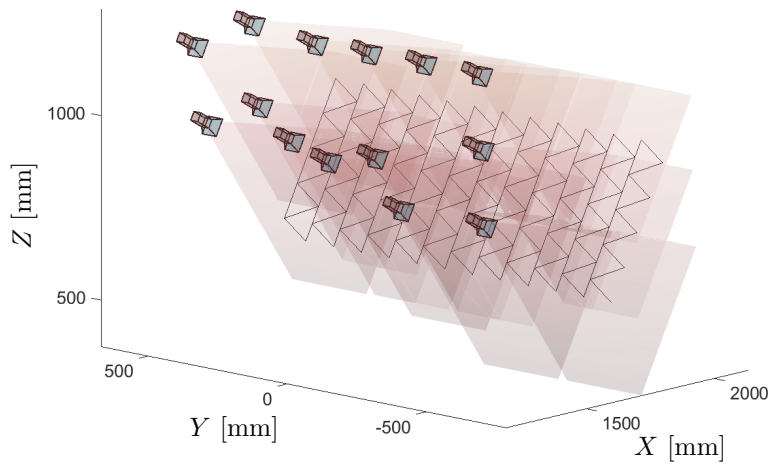
Fig. 2: The overlap g between adjacent camera poses. The transformation between the camera poses C_k and C_{k-1} is $T_{C_k}^{C_{k-1}}$. The camera views are modeled as pyramids representing the camera field of view, with base sides x_f , y_f , and height h_f .

4. Computational Analysis

A scalable but straightforward surface finishing scenario is used for evaluating the computational capabilities of the proposed procedure. The test scenario consists of



(a)



(b)

Fig. 3: The triangular pattern used for analyzing the algorithm characteristics (a). The resolution of the pattern is varied, yet the surface area it covers is kept constant. Using the proposed approach, an example of a resulting set of camera view poses covering the pattern is shown in (b).

a tool path where the tool movements follow an equilateral triangular pattern. The pattern has a total height of 500mm, a total width of 1200mm, and it is arranged in a plane, as shown in Fig. 3a. The surface corresponds to a total area of 0.6m^2 , filled with a variable number of equilateral triangles. Effectively, this corresponds to different tool path lengths and resolutions over a fixed area, and consequently, a varying amount of input view poses. The triangular pattern is to be inspected by a simulated camera that has a field of view imaging area of $(y_{far} \times x_{far})$ of $430\text{mm} \times 270\text{mm}$ and an optimal scanning distance of 700 mm.

The analysis is performed by measuring the time spent on each part of the algorithm when varying the number of input view poses or the resolution of the voxel grid filter. Simulations were performed on a computer with a 3.6 GHz CPU running Windows 10, and the view selection algorithm was implemented in a single-thread MATLAB program. The particle filter based tool tracking and recording parts of the proposed system are hence, not evaluated in this test.

Fig. 3b gives an example of a set of camera view poses generated by the proposed approach in order to inspect the triangular pattern shown in Fig. 3a. Total time consumption and the timing of each step of the algorithm is shown in Fig. 4. The total running time assumes linear behavior around 1000 input tool path points (Fig. 4a), while increased voxel grid resolution (decreasing voxel size) results in exponential behavior when approaching zero (Fig. 4b). From the figures, it can be seen that the initial filtering of the tool path takes a negligible amount of time. The T^2 and camera overlap calculations that are performed for sorting and selecting the view poses is the most time-consuming part of the algorithm.

The performance of the proposed approach is also assessed by comparing it with a brute force approach, where all the recorded tool path points, after the initial voxel grid filtering, are considered as possible camera view poses for surface inspection. The exhaustive approach examines all the view poses in order to find the required number of views. As shown in Fig. 5, the exhaustive strategy demonstrates high dependence on the number of recorded tool path points. Varying the resolution of the initial voxel grid filtering displays similar effects on running times for both procedures.

5. Evaluation Test Cases

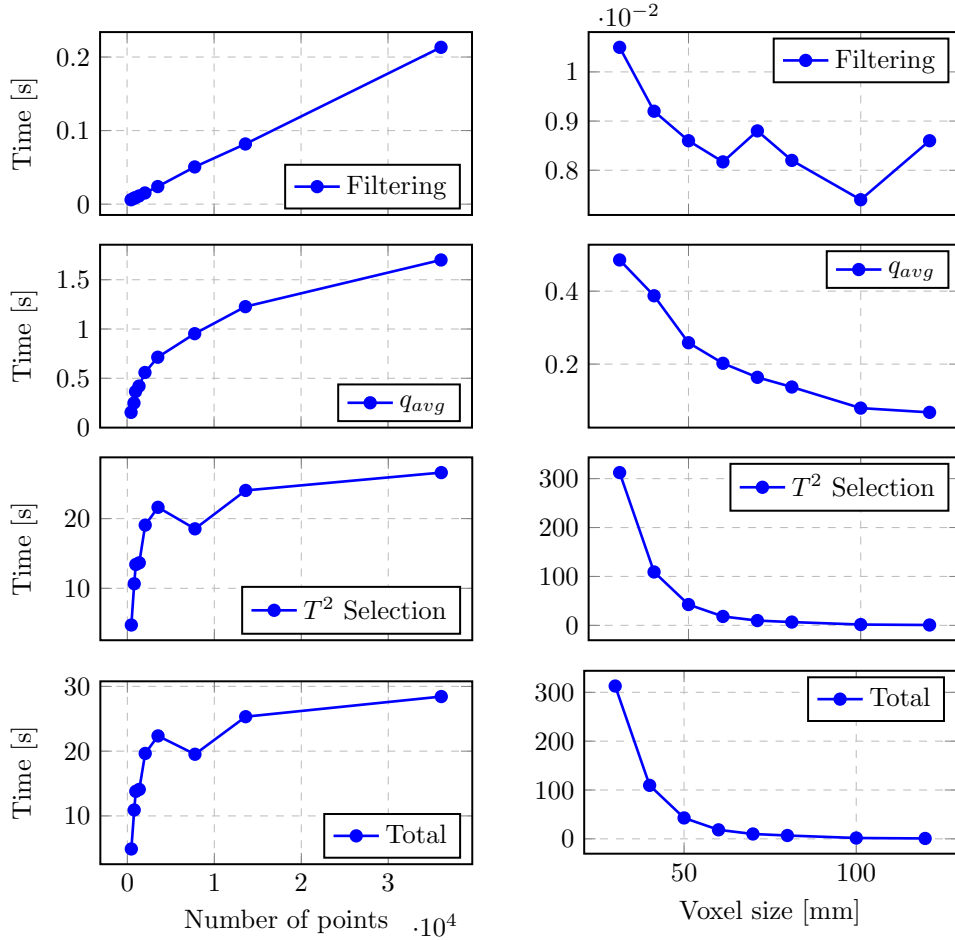
The proposed approach has been evaluated through three different test cases. In the first case, an inspection of a simulated CNC surface machining operation is under consideration. The two subsequent cases deal with different cases of inspection of a large ship propeller blade cast in NiAl bronze.

5.1. *Simulating a complex surface machining operation*

In the first evaluation case, a CNC machining process is simulated, where camera view poses are generated from the tool paths of a machining process performed on a double-curved surface. The simulation was performed with the same computational

resources and camera parameters as in the computational analysis in the previous section. The particle filter based tool tracking and recording parts of the proposed approach are not evaluated in this test.

A curved and bent surface with associated tool path is shown in Fig. 6a. The



(a) Increasing amount of input tool path points.

(b) Decreasing resolution of the voxel grid filter.

Fig. 4: Computational analysis of the proposed approach. Time consumption is measured for each of the essential steps: The voxel grid filtering step, the camera orientation selection step (q_{avg}), and the final step where the statistical T^2 measure and camera view overlap calculations are used for sorting and selection of camera view poses. The total running time is also considered. In (a), the amount of examined tool path points is varied, while the voxel size for the initial filtering is varied in (b).

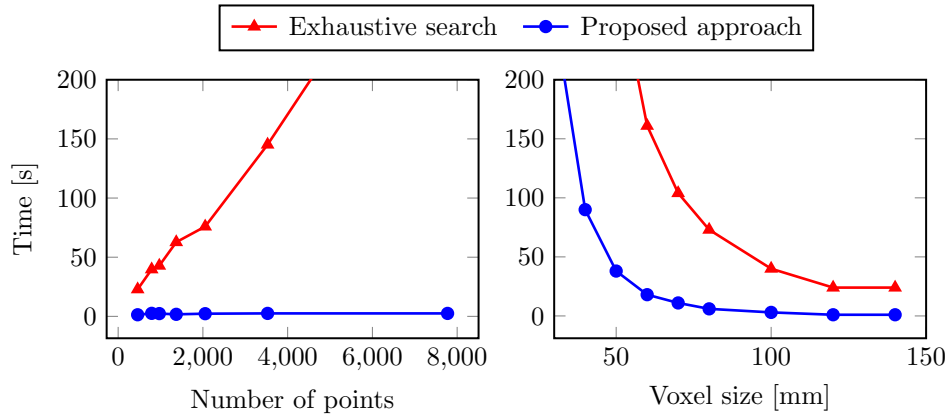


Fig. 5: Running times of the proposed approach compared with an exhaustive strategy, where all the recorded tool path points remaining after the initial voxel grid filtering are considered as possible camera view poses for inspecting the triangular pattern shown in Fig. 3a. The exhaustive strategy demonstrates high dependence on the number of recorded tool path points (left), but similar performance as the proposed approach regarding voxel grid filter resolution (right).

tool path, which is sampled from the simulated CNC machining process, is also shown in Fig. 6b. The tool path was sampled at a rate of 15 Hz. A representation of the tool path after initial voxel grid filtering is shown in Fig. 6c. Fig. 6d illustrates the 17 resulting camera views.

The proposed system was able to successfully generate camera viewpoints covering the whole surface area, based on the simulated machining tool path.

5.2. Capturing a surface finishing process

We tested our approach in an experimental setup using a Universal Robots UR10 robot equipped with a Zivid high-accuracy RGBD-camera for inspection of a large ship propeller blade. A second RGBD-camera (Microsoft Kinect v2) was used to track the motions of an angle grinder using the particle filter described in Section 2 and a CAD model of the angle grinder. The tool path was obtained while the skilled worker performed surface polishing on the side of a ship propeller blade, as shown in Fig. 7a.

This experiment was conducted in order to evaluate the qualitative results of the approach. The motions performed by the skilled worker in the surface polishing process is characterized by repetitive, alternating motions over the double-curved surface of the propeller blade. The captured raw tool path is shown in Fig. 7b. A representation of the tool path after initial voxel grid filtering is shown in Fig. 7c. Fig. 7d illustrates the generated inspection camera view poses, which is used by the robot to capture the point cloud of one side of the ship propeller blade. The

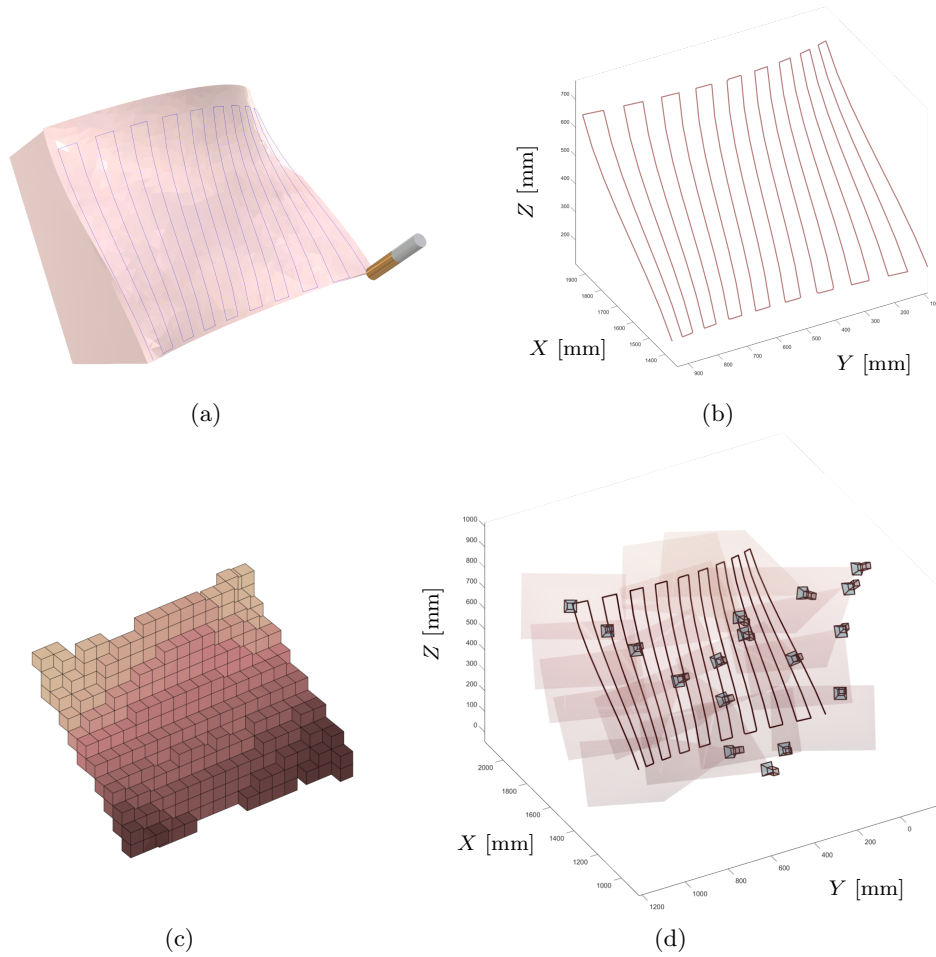


Fig. 6: First evaluation test case: Simulation of a complex surface machining operation. In (a), the original surface with the CNC machining tool path is shown. The sampled tool path is shown in (b). In (c), a voxel representation of the tool path after the initial voxel grid filtering is shown. The resulting camera view poses generated by our approach is illustrated in (d).

resulting point cloud is shown in Fig. 8. The selected inspection camera view poses provide satisfying coverage of the captured tool path and underlying ship propeller surface. The generated camera view poses have high coverage of the captured tool path.

5.3. Capturing a surface adjusting process

The approach was in the following round evaluated for the case where a skilled worker is performing minor geometry adjustments of a ship propeller blade sur-

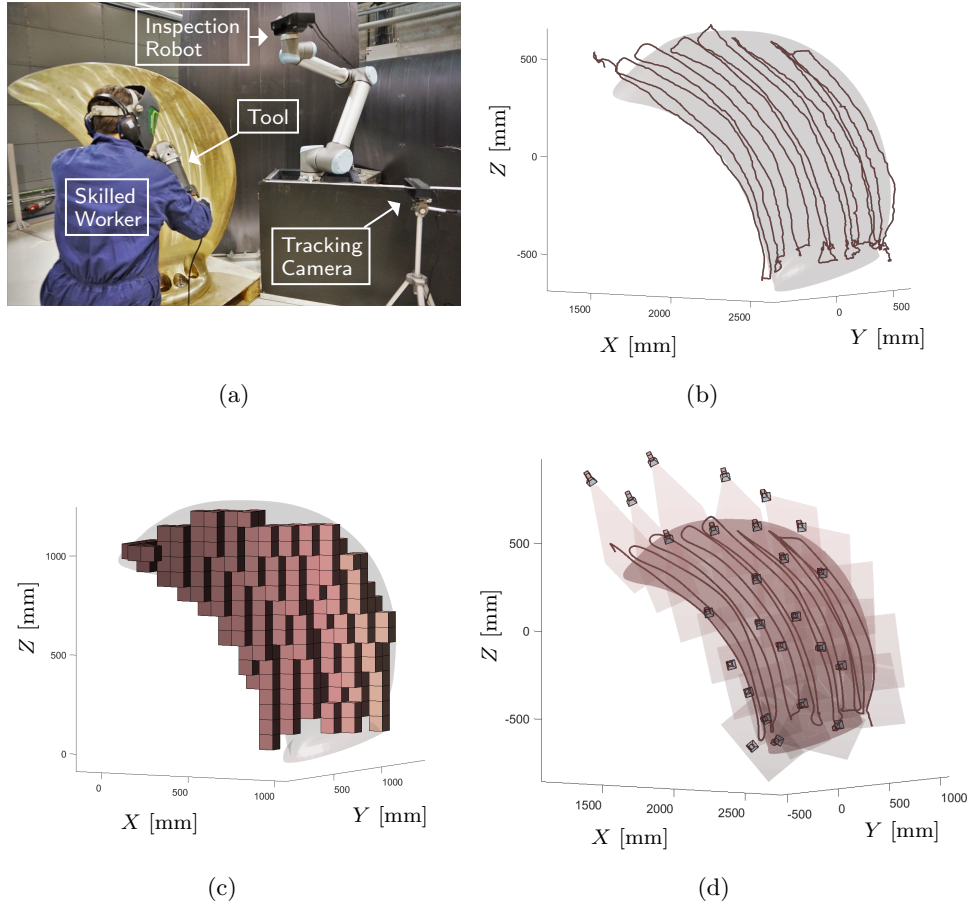


Fig. 7: Second evaluation test case, where an actual surface finishing process is inspected. In (a), a skilled worker is polishing a propeller blade surface using an angle grinder which is tracked by the RGBD-camera and particle filter. The resulting raw tool path from the manual process is shown in (b). In (c), the tool path after initial voxel grid filtering is represented as a voxel grid. The resulting inspection camera view poses generated by our approach is shown in (d). The point cloud produced by the inspection is given in Fig. 8.

face. The setup is similar to the test case described in the previous instance, except for that the recorded tool path has different characteristics compared to the previous case. An alternative attitude was also tested for the tracking RGBD-camera (Microsoft Kinect v2) and the inspection robot. The setup is shown in Fig. 9.

This experiment was conducted in order to demonstrate how the procedure for selecting the camera view poses performs when subjected to a highly repetitive tool path. Compared to the previous test case, performing adjustments to the surface



Fig. 8: 3D inspection results using the proposed approach with an industrial robot and a high-accuracy RGBD-camera.

geometry involves even more repetitive tool movements yet only in a small region of the total blade surface. Fig. 9b shows the skilled worker performing the adjustments by using an angle grinder. The illustration is a frame from the depth stream captured by the tracking RGBD-camera. Marked in red is the tool CAD model, which is used for particle filter tracking of the tool. The recorded raw tool path is shown in Fig. 9c. A voxel representation of the tool path following the initial filtering is illustrated in Fig. 9d. The resulting inspection camera view poses are given in Fig. 9e, next to an image of the robot inspecting the surface (Fig. 9f).

6. Summary & Conclusions

This paper presents a novel approach for determining camera view poses for automated inspection of the results of manual processes performed by skilled workers. The approach is based on the assumption that tool movements performed by a skilled worker coincide with the most relevant areas of the underlying surface of the

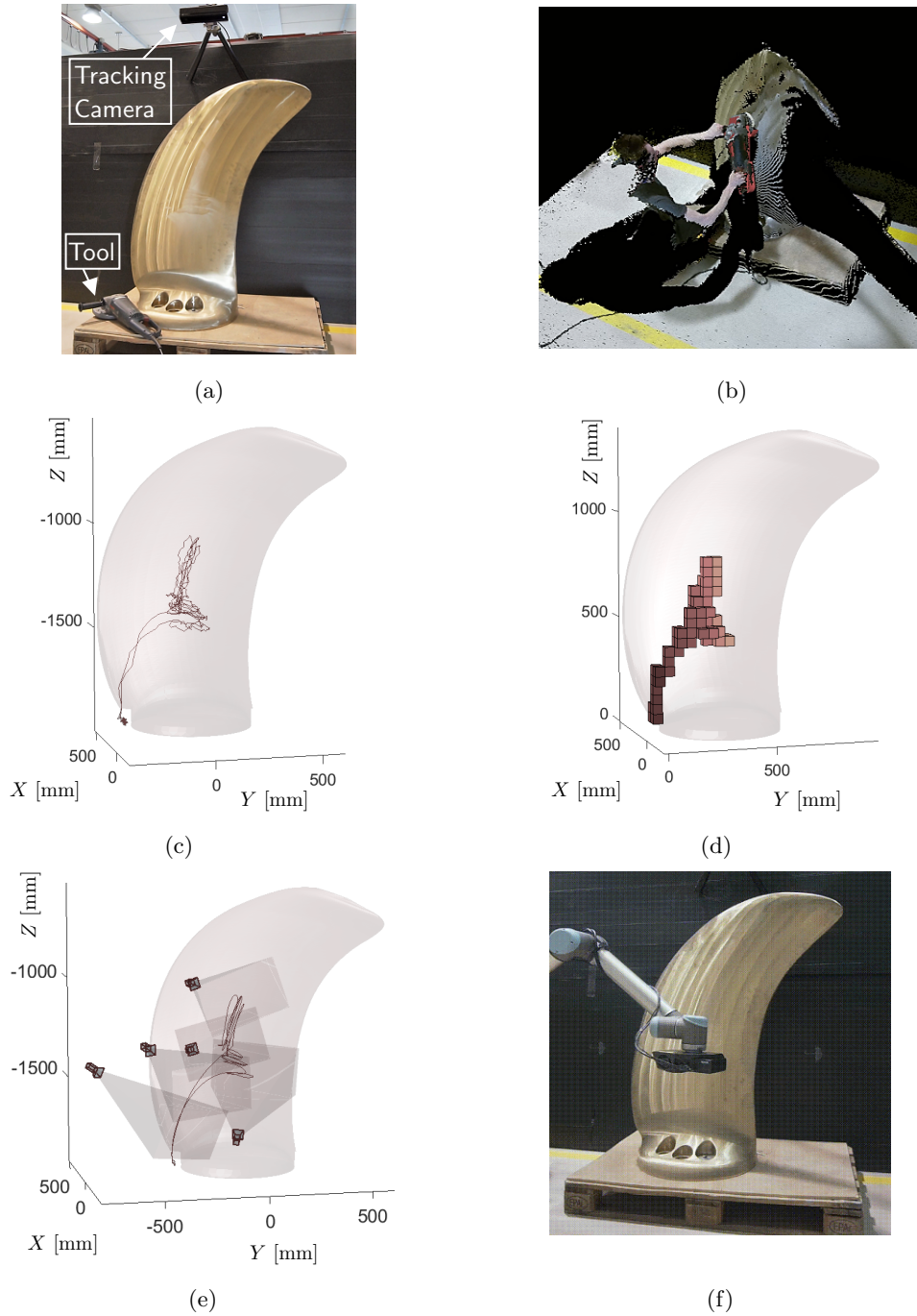


Fig. 9: Third evaluation test case, where a propeller blade is inspected after manual surface geometry adjustments. In (a), the evaluation setup is shown. A point cloud representation of a skilled worker adjusting the surface geometry is given in (b). The tracked raw tool paths from the manual process is shown in (c). In (d), the tool path after initial voxel grid filtering is represented as a voxel grid. The inspection camera view poses generated by our approach is shown in (e), with a robot performing the inspection in (f).

workpiece, namely the parts where a manual process has been performed.

The technique is also suitable for automated inspection planning in the case where a robotic welding or polishing operation has been performed, and sparse or no CAD information of the workpiece is available. Such scenarios can occur if online teach- or lead-through methods are used for robot programming.

In contrast to existing work, the algorithm expects a tool path and is capable of adapting the orientation of the robot. In order to select meaningful camera view poses, camera parameters such as field of view and optimal scanning distance are necessary input to the algorithm. The procedure is not aware of what process it has been shown. It does not recognize, that for example, a surface polishing task has been demonstrated. There is neither deduction of the operations nor a priori knowledge.

We provided qualitative results by simulating various tool paths and quantitative results by recording data from a skilled worker. The approach can easily be extended to other applications, such as robotic welding, machining, or painting.

While the different operations described in this paper have been functionally implemented and successfully experimented with, the integration into a complete process cycle remains to be done. Collision avoidance could be added to the system, taking into account the complete robot model and the continuous stream of information originating while inspecting the workpiece. It would also make sense to take into account the reach of the robot when selecting the inspection camera view poses, or adapting generated inspection programs to the reach of the robot used. Further work may also focus on improving the scanning execution time or on integrating the approach into our ship propeller inspection system [27].

Acknowledgements

The authors wish to thank the OLIVER project and industrial partner Oshaug Metall AS. The project is supported by the Norwegian Research Council under the BIA program, project number 261639. This paper is an extended and revised version of a preliminary conference article that was presented on IEEE IRC 2019 [20].

References

- [1] W. Sheng, N. Xi, M. Song, Y. Chen and J. S. Rankin, Automated CAD-guided automobile part dimensional inspection, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)* **2** 1157–1162 (2000).
- [2] F. Prieto, R. Lepage, P. Boulanger and T. Redarce, A CAD-based 3D data acquisition strategy for inspection, *Machine Vision and Applications* **15**(2) 76–91 (2003).
- [3] F. Zhao, X. Xu and S. Q. Xie, Computer-aided inspection planning the state of the art, *Computers in industry* **60**(7) 453–466 (2009).
- [4] S. E. Sadaoui, C. Mehdi-Souzani and C. Lartigue, Computer-aided inspection planning: A multisensor high-level inspection planning strategy, *Journal of Computing and Information Science in Engineering* **19**(2) p. 021005 (2019).

- [5] G. Reinhart and W. Tekouo, Automatic programming of robot-mounted 3D optical scanning devices to easily measure parts in high-variant assembly, *CIRP annals* **58**(1) 25–28 (2009).
- [6] R. Raffaeli, M. Mengoni, M. Germani and F. Mandorli, Off-line view planning for the inspection of mechanical parts, *International Journal on Interactive Design and Manufacturing (IJIDeM)* **7**(1) 1–12 (2013).
- [7] G. Sansoni, M. Trebeschi and F. Docchio, State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation, *Sensors* **9**(1) 568–601 (2009).
- [8] Y. Li and P. Gu, Free-form surface inspection techniques state of the art review, *Computer-Aided Design* **36**(13) 1395–1417 (2004).
- [9] C. Connolly, The determination of next best views, *Robotics and automation. Proceedings. 1985 IEEE international conference on* **2** 432–435 (1985).
- [10] J. E. Banta, L. Wong, C. Dumont and M. A. Abidi, A next-best-view system for autonomous 3-D object reconstruction, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **30**(5) 589–598 (2000).
- [11] S. Wenhardt, B. Deutsch, E. Angelopoulou and H. Niemann, Active visual object reconstruction using d-, e-, and t-optimal next best views, *2007 IEEE Conference on Computer Vision and Pattern Recognition* **1–7** (2007).
- [12] G. Du and P. Zhang, A markerless human-robot interface using particle filter and kalman filter for dual robots., *IEEE Trans. Industrial Electronics* **62**(4) 2257–2264 (2015).
- [13] J. Kofman, X. Wu, T. J. Luu and S. Verma, Teleoperation of a robot manipulator using a vision-based human-robot interface, *IEEE transactions on industrial electronics* **52**(5) 1206–1219 (2005).
- [14] R. Dillmann, T. Asfour, M. Do, R. Jäkel, A. Kasper, P. Azad, A. Ude, S. R. Schmidt-Rohr and M. Lösch, Advances in robot programming by demonstration, *KI-Künstliche Intelligenz* **24**(4) 295–303 (2010).
- [15] W. Ng, H. Chan, W. K. Teo and I.-M. Chen, Programming robotic tool-path and tool-orientations for conformance grinding based on human demonstration, *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on* **1246–1253** (2016).
- [16] B. Nemeč, K. Yasuda, N. Mullennix, N. Likar and A. Ude, Learning by demonstration and adaptation of finishing operations using virtual mechanism approach, *2018 IEEE International Conference on Robotics and Automation (ICRA)* **7219–7225** (2018).
- [17] C. W. Ng, K. H. Chan, W. K. Teo and I.-M. Chen, A method for capturing the tacit knowledge in the surface finishing skill by demonstration for programming a robot, *Robotics and Automation (ICRA), 2014 IEEE International Conference on* **1374–1379** (2014).
- [18] E. KALT, R. MONFARED and M. JACKSON, Towards an automated polishing system – capturing manual polishing operations, *International Journal of Research in Engineering and Technology* **5**(07) 182–192 (2016).
- [19] A. Irish, I. Mantegh and F. Janabi-Sharifi, A PbD approach for learning pseudo-periodic robot trajectories over curved surfaces, *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on* **1425–1432** (2010).
- [20] E. B. Njaastad and O. Egeland, Planning of robotic inspection from visual tracking of manual surface finishing tool, *2019 Third IEEE International Conference on Robotic Computing (IRC)* **277–281** (2019).
- [21] M. Wüthrich, P. Pastor, M. Kalakrishnan, J. Bohg and S. Schaal, Probabilistic object tracking using a range camera, *Intelligent Robots and Systems (IROS), 2013*

- IEEE/RSJ International Conference on* **3195–3202** (2013).
- [22] A. Doucet and A. M. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering* **12** 656–704 (2009).
 - [23] A. Doucet, N. De Freitas, K. Murphy and S. Russell, Rao-blackwellised particle filtering for dynamic bayesian networks, *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* **176–183** (2000).
 - [24] F. L. Markley, Y. Cheng, J. L. Crassidis and Y. Oshman, Averaging quaternions, *Journal of Guidance, Control, and Dynamics* **30**(4) 1193–1197 (2007).
 - [25] H. Hotelling, The generalization of student’s ratio, *Ann. Math. Statist.* **2** 360–378 (08 1931).
 - [26] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**(11) 559–572 (1901).
 - [27] E. B. Njaastad, N. H. Munthe-Kaas and O. Egeland, Robotic autoscanning of highly skewed ship propeller blades, *IFAC-PapersOnLine* **51**(22) 435–440 (2018).