The Integration of Sensing Necessary for Physical Collaborative Implementation Is Typically the Main Focus of Industrial Spraying Operations That Necessitate Integrated Robotic Route Planning for Complicated Geometry Collaborative Frameworks.

Pranjal Garg[1], Piyush Kumar Jain[2], Harimohan Soni[3]

[1]*Mechanical Engineering Department, Bansal Institute of Science and Technology, Bhopal (M.P.), India.*
[2]*Mechanical Engineering Department, Bansal Institute of Science and Technology, Bhopal (M.P.), India.*
[3]*Mechanical Engineering Department, Bansal Institute of Science and Technology, Bhopal (M.P.), India.*

---***---

Abstract - In the present instant, there are a number of important obstacles that are preventing the broad use of automated robotic solutions to complex activities. For instance, in high-mix/low-volume activities, there is frequently an excessive amount of uncertainty that makes it impossible to correctly hard-code a robotic work cell. This is because of inefficient sensing and the unpredictability of the job. A significant number of the currently available frameworks for collaboration are focused on including the senses that are required for successful physical cooperation. A route planner, a route simulator, and a result simulator are all essential elements that come together to form the framework that has been proposed. Through the use of an integrated user interface, the operator is able to connect with these modules, making modifications to the path plan before automatically authorising the job for execution by a manipulator that does not require collaboration. The collaborative framework is presented for a pressure washing task in a situation that falls under the category of remanufacturing, where each component requires one-off route planning. Shot peening, deburring, grinding, sandblasting, and spray painting are some of the additional operations that might be incorporated into the configuration of the framework. It is possible that surface preparation and coating might be automated in such environments through the utilisation of automated route planning for industrial spraying operations. In the literature, autonomous spray route planners have concentrated on continuous and convex surfaces; however, the majority of the components that are found in the actual world do not adhere to the assumptions that they have established. Adjustments can be made to the movement speed or offset distance at certain points along the route in order to accommodate the requirements of the specific path. In addition, the creation of the route planner takes into consideration the trade-offs that are associated with path adaptation as well as the relative efficacy of various adaptive strategies.

Key Words: collaborative framework, cognitive function spraying operation,reliability,robotics.

1.INTRODUCTION

Building an automated pressure washing work cell capable of handling the majority of the Army's rework and rebuild depot components was the initial objective of this project. The large variety of part sizes and geometries necessitating frequent cleaning made this task challenging. Tank bodies and minor parts are cleaned on pallets at this facility. The problem was made worse by the fact that determining the part's geometry was next to impossible. This is particularly challenging for repair and rebuild facilities due to a lack of data, variations in parts that are easy to detect, and the use of a manual technique with unique parts. Most organisations have opted to complete the operation manually rather than automate it because of these difficulties. Unfortunately, the technology required to automate these physically demanding vocations was either prohibitively expensive or overly complex until recently, making this method the most frequent. Automating full coverage path planning in a consistent and cost-effective manner is a challenging task. It's certainly doable, but a new path design isn't necessary for the majority of automated tasks. Preprogrammed parts are typically repeated using them. Collaborative robotics pioneered the notion of combining human cognitive function with automated robotic system accuracy and endurance since humans are skilled at determining what needs cleaning and automated systems can make effective path plans on the fly. The quality of the path was enhanced with the use of adaptive path planning in comparison to the naïve technique. This project was divided into two issues because of its growing breadth. Describe the collaborative system in detail, beginning with input and ending with process execution after user verification. Could you please describe a pressure washing adaptive path planner.

Figure 1: Spray Optimization.

1.1 A COLLABORATIVE FRAMEWORK FOR ROBOTIC TASK SPECIFICATION

Since the first industrial implementations of robotic solutions in manufacturing environments, task specification has been one of the toughest and most time-consuming parts of the implementation process. As robotics has advanced, so has the technology surrounding task specification; however, there is still a need for the operator to physically program the robot. While this is fine for low-mix, high-volume production processes, it is a very restrictive requirement for the automation of lower-volume processes. Automated task specification would go a long way toward alleviating some of the hurdles faced by high-mix, low-volume processes. However, the implementation of automated robotic solutions for complex tasks currently faces a few major hurdles. Lack of effective sensing and task variability create too much uncertainty to reliably hard-code a robotic work cell. Collaborative robotics have proven effective for mitigating uncertainty by mixing human cognitive function and fine motor skills with robotic strength and repeatability. Yet, there are many instances where physical interaction is impractical, as human reasoning and task knowledge are still needed. The solution is a framework that blends the latest developments in automated task specification with the experience and cognition of a human operator to provide a more accurate task specification. While this chapter does focus on surface finishing tasks such as pressure washing, sandblasting shot peening, deburring, grinding, sanding, and wire brushing, the framework can also be applied to any robotic task that does not have a predefined path, such as assembly, inspection, packaging, and pick-and-place operations.

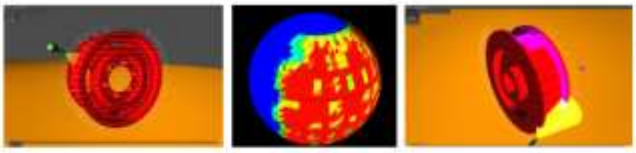


Figure 2: (a) Path with process simulation; (b) Process analysis visualization; (c) User interface with process simulation

2. Methodology

This approach involves taking some initial 3D data, given as an STL file in this case because the algorithm requires normal vector, and building a convex hull around it to eliminate the collision and accessibility issues created by non-continuous and concave surfaces. However, given a method for determining normal vectors and building a tessellated mesh from a point cloud, any number of 3D data gathering methods could be used. At this point, the mesh is converted into a point cloud where the centroid of each facet is linked to the normal vector of that facet. The path is then built based on the convex hull using a slicing based method that relies on the following input parameters: a rotation axis and the degrees of rotation, which serve to modify the slicing direction, as it is unlikely that the part will actually be moved if the scanner is calibrated and registered correctly; and a slice thickness, an offset distance, and an overlap percentage, which serve to quantify how much work will be applied to the part. Once the path has been built, the points from the original mesh, now represented as a point cloud, are mapped to specific segments of the path. In some cases, a point can be mapped to multiple segments.

Figure 3: Overview of the Adaptive Path Planner

2.1 Path Building on the Slice

Within each slice, which are represented as planes defined by $Z=hs$, the intersecting facets Fs of the convex hull H are found by,

$$F_s = \{f \mid u_f = 1 \text{ or } u_f = 2\} \qquad \forall f \in H$$

where u_f is the number of vertices above the slicing plane defined by,

$$u_f = \sum_{i=1}^{3} \begin{cases} 1 & , q_{fiz} \geq h_s \\ 0 & , q_{fiz} < h_s \end{cases} \qquad \forall f \in H$$

where $\boldsymbol{q}fiZ$ is the Z value of vertex i of facet f on the convex hull H and hs is the height of slice s. If a facet has one vertex above or below the plane and the other two are on the opposite side it is considered to be an intersecting facet and is included in the set. When a facet is sliced directly on a single vertex, it is included, and the following interpolation is not necessary. This produces a geometrically finalised toolpath for slices, excluding the inversion of the normal vector to properly represent the end effectors' orientation. However, each data point is still lacking the seventh data point, the time to move between points, needed for a full representation of the trajectory. The resulting trajectory for the slice is defined as,

$$Gs = \{[\boldsymbol{p}, \hat{\boldsymbol{n}}_p, \Delta t_p] \mid \Delta t_p = dp/v_0 \qquad \forall \boldsymbol{p} \in P_s\}$$

where Δtp is the time to move from $\boldsymbol{p}_{x-1}$ to $\boldsymbol{p}_x$, d_p is the distance between the two points, $\hat{\boldsymbol{n}}_p$ is the normal vector corresponding to point p, and Ps is the ordered set of all path points on slice s. It should be noted that the time to move to the first point in the path, $\boldsymbol{p}_0$, is zero.

The true surface is represented by a set of ordered pairs of points and normal vectors, $C=\{\boldsymbol{c}, \hat{\boldsymbol{n}}\}$. The points are sampled from the workpiece's tessellated mesh, and each point is paired with the unit normal vector of the facet from which it was sampled. To avoid ambiguity regarding the unit, to this point, all path planning has been done on the convex hull. This will henceforth be referred to as the "naïve" tool trajectory since it does not consider the underlying surface topology. The following sections describe methods for adapting the naïve trajectory based on the actual part surface. This is accomplished by associating features of the underlying surface with segments of the toolpath and adjusting the tool offset distance and/or velocity based on aggregate descriptions of the underlying surface's position and orientation with respect to the convex hull. The true surface is represented by a set of ordered pairs of points and normal vectors, $C=\{\boldsymbol{c}, \hat{\boldsymbol{n}}\}$. The points are sampled from the workpiece's tessellated mesh, and each point is paired with the unit normal vector of the facet from which it was sampled. To avoid ambiguity regarding the unit normal, points should not be sampled from facet edges. Sampling strategy is a tradeoff between resolution and computational load, and it has implications for how each surface facet influences the adjusted tool trajectory. A simple method is to take the centroid of each facet. This insures that each facet is represented in the ensuing calculations but has disadvantages when facet size varies significantly or facet aspect ratios are high: Areas of high curvature (many small facets) can dominate areas of low curvature (fewer, larger facets), and the centroids of high-aspect-ratio facets can be far from their associated vertices. The disadvantages may be mitigated by enforcing a uniform sampling resolution within facets; however, this can significantly increase the number of points and thus the computational load.

Ideally, all points, $c \in Cs$, within the volume swept by the spray cone as it moves from point pi to pi+1 would be mapped to that toolpath segment; however, this is computationally expensive both in terms of mapping points and the subsequent calculations where points are associated with multiple segments. Instead, this research opts for a binned approach, whereby each point is assigned to a single segment according to its polar coordinate relative to the origin, as in Figure 7. This assumes that the part has been centered on the origin, otherwise the centroid of the convex polygon could be used in its place. It is further required that the surface facet(s) represented by each point are oriented so as to be exposed to the spray. This is accomplished by defining the segment normal.

Using this collection of binned points, the algorithm has the option to adapt the path by adjusting the distance from the part, the velocity of the move or both for each bin within each slice. The process for using both adaptive methods is relatively straight forward when used separately, the distance-based method looks at the aggregate distance from the end effector to all of the affected points and adjusts accordingly and the time-based method looks at the aggregate incidence angle between the orientation of the end effector and the normal vector of each affected point. However, there is one constraint on how they can be used together. Due to the change in positions created by the distance-based adapting method, which can have an effect on the distance between points and thus the time needed to complete the move, the time-based adapting method must be used after the distance-based method for accurate results.

One of the shortcomings of this methodology is the fact that each adaptive algorithm only considers one data type when making a decision on how to adapt the path. Ideally, every decision should be made with all of the available data, but sometimes different data types can skew the results. In this case, using the incidence angle as an indicator of how much the path offset distance should be adjusted would cause the distance to be adapted too much to the point where the sprayer width is narrower than the offset width. This causes gaps in the raster pattern which then requires replanning later on. Alternatively, using distance from the part to influence the velocity of the end effector does not cause the same path breaking issues, but it doesn't necessarily help all that much either. If the end effector is already too far away, the overall performance of the path would be much better served by adjusting the distance rather than extending the exposure time and thus lengthening the overall execution time.

The adaptive process can begin by defining the adjustment value for each bin as,

$$\psi b = \min(d_{AGb} - d_{O}, x d_{O}) \; \forall \; b \in B_s$$

where Bs is the set of all bins on slice s, x represents the maximum percentage that the path can adapt by with regards to d_O, and d_{AGb} is defined as the aggregate distance value of all points in the bin, which can vary based on the aggregation method. Ideally, this returns a value of zero, meaning no adjustment is needed and the aggregate distance is equal to the desired offset distance. The adjustment value is limited in range so that the path will not be moved within the convex hull by over adjusting. In this case, x = 0.95. This is necessary because a value greater than or equal to 1 would result in an adjustment greater than the offset distance itself which would cause the new position to be inside or on the convex hull. While this may not create any collision issues, there is always the possibility and thus is must be accounted for. Alternatively, a minimum distance could be defined and the adjustment would occur on the remaining distance beyond the minimum distance. It should also be noted that the aggregate method used depends on the user's initial input. If necessary, the new path points are determined by,

$$\boldsymbol{p} = \boldsymbol{p} - (\boldsymbol{n}_{p} \, \psi_p) \; \forall \; \boldsymbol{p} \in P_s$$

where $\boldsymbol{n}_p$ is the unit vector of the corresponding surface normal, Ps is the path for slice s, and ψp is the adjustment value of the path point which is defined as,

$$\psi_p = \max(\psi b_p, \psi b_{p-1})$$

where ψb_p is the adjustment value of point p in bin b. The max of the two bins that share the point is used to ensure that the bin needing the most adjustment gets it. For path smoothness, a moving average of the adjacent bins can be used to determine how much adjustment is needed for each bin by taking the mean of a few adjacent values on both sides of the bin. An example of distance-based adaptation is shown below in Figure 10 as an individual slice of Part C, where both the naïve path, in blue, and the distance adapted path, in red, have been plotted along with the facets that are captured within the slice, the exact slicing height is shown to the right of the path analysis.

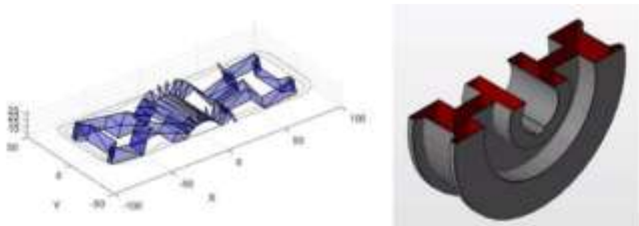
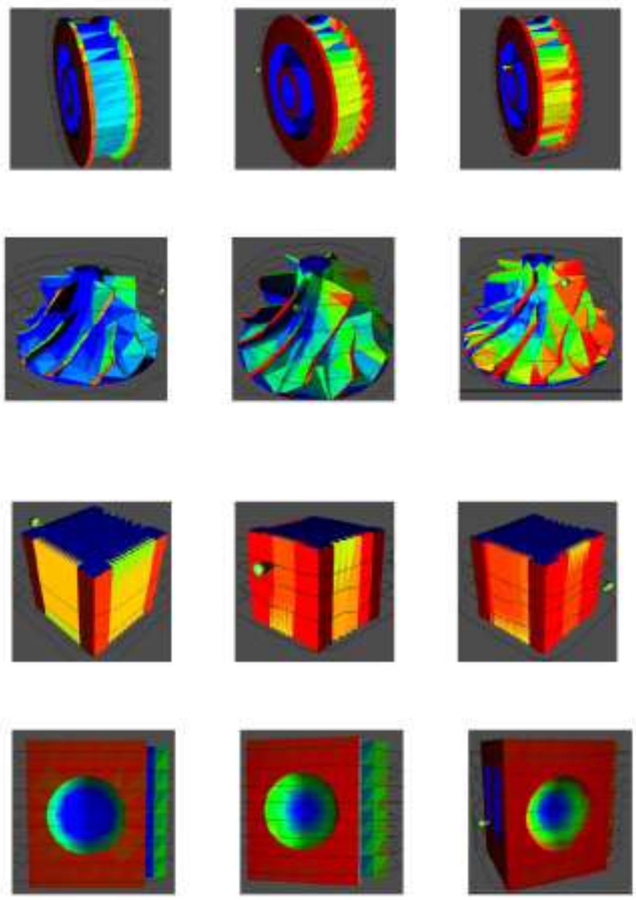


Figure 4: Distance vs Native path plan

In order to determine the best method for producing better toolpaths, two factors were considered. The type of adaptive algorithm used has two levels: Distance and Distance + Time, and the type of statistical aggregate method used, which has four levels: Mean, Mode, Min and Max.

3.Results and discussion

The findings presented here are derived from a collection of five test pieces that are presented in Appendix A. The other three pieces were acquired from online 3D CAD databases to reflect common geometries that are found in the real world. Two of the parts were particularly built to show how the planner makes alterations to the path, while the other three were borrowed from the internet databases. In order to investigate whether or not there were any interactions between the two components, preliminary analysis was carried out using ANOVA. Following a two-way study, the statistical approach was deemed to be multi-colinear, and it was therefore rejected. Due to the fact that the aggregation technique directly affects all data in the same and typically equal manner, it is not surprising that a one-way analysis of variance (ANOVA) investigation revealed that there was some importance to the type of statistical approach that was utilised. In other words, if one were to use common sense, it would be reasonable to assume that adaptation based on the minimum values would be higher than adaptation based on the mean, which would be higher than adaptation based on the maximum values. From this point forward, the study will concentrate on the distinctions between adaptive techniques by averaging the results obtained from each treatment that is categorised according to the adaptive approach.



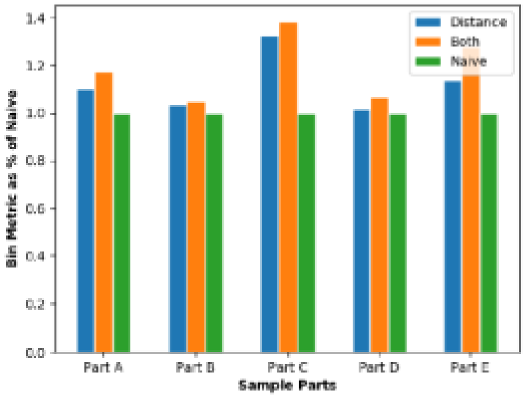
Part B a specifically designed test cube intended to show how the algorithm reacts to changes in incidence angle and thus how it adapts velocity and time. While the results are a bit difficult to see as the mean surface is mostly convex and requires little path variation, when viewed in real time, a real change in velocity can be observed. This part also demonstrates one of the key issues with flat surfaces on the convex hull. There is a distinct difference of coloration between facets on the same plane due to the large facet size which causes the centroids to be captured in different slices and thus adapted differently. This also causes the path to be unevenly adapted despite the mean surface being consistent throughout.

Part C is a simplified wheel rim that has been sliced in the vertical orientation to provide more complexity as opposed to being sliced laying down. One of the unique features of this part is that, due to the edge of the rim, the bins on the path tend to catch points with very different normal vectors and distances. This part experiences some inconsistent impingements due to the sampling error described above. The interior of the wheel also showcases facets within the effective spraying distance, but with perpendicular normal that receive almost no significant cleaning as well as facets that have completely parallel surface normal but still can't meet the minimum effective spray distance despite distance adaptation.

Part D is a rotor blade from an engine that has been reduced from its original facet count for computational efficiency. The original part is shown in Appendix A. While this may look ugly, it is a good approximation of the surface for the algorithm's purposes and allows for each facet to be considered and given an impingement value, which is very difficult to do when the part is down sampled. Due to the shape of the rotor, distance-based adaptation for the full part is hindered by the edges of each fin. Here the reduction does not have a large effect on the visible geometry. This part represents an extreme challenge of tool point accessibility. In this case, the sprayer cannot invade the convex hull, which makes it very hard to effectively cover a large portion of the part.

3.1. DISCUSSION

From these results, the most consistent, effective and cheapest, in terms of computation time, method for adapting the naïve path is as follows: Begin by adapting the naïve path using the distance-based method, evaluate the quality of the path and if a standard is not met, adapt the path based on time if required. Otherwise, the time-based adaptive method is not needed. This can be seen across all of the adaptive algorithm charts in column 1 of Figure 13 which shows a significant decrease in low impingement values from the naïve to the distance adapted path and again when adapting by both distance and time and is summarized in Figure 20. While the values in the figure for parts B and D are not exactly convincing, they can be explained by geometric abnormalities as discussed above. Overall, these findings are consistent with the fact that work done by spraying is highly nonlinear when adjusting distance, which makes it a critical part of the process when calculating actual work done. This method was also chosen because the benefits of adapting based on time are nowhere close to the benefits of adapting by distance and choosing only to use it if the distance method cannot meet the requirements saves significant computational time. Aside from computational time, adjusting the time for each bin generally creates a longer path completion time, which is not ideal. Further compounding the issue of time, some distance adjustments can actually lessen the overall completion time. It should also be noted that the mean aggregation method is preferred because it gives a better representation of all points in a bin. While taking the minimum definitely returns better values, it can be unnecessarily affected by outliers. The same holds true for the maximum value. And while the mode might make sense, these are continuous values and thus it is not very likely to consistently find multiple occurrences of the same value. The difference between using max and min values can also be made up by increasing the sprayer pressure.



At the beginning, the rationale for developing this algorithm was to allow for smooth path planning for very complex parts, which is something it excels at. No matter what geometry is provided to the planner, the resulting path will always resemble a convex part with accessible tool points barring any robot specific accessibility issues. Other than reach, there are no geometrical constraints preventing the execution of the resulting path. For the largest parts, placing them on a rotary table would allow for a robot with a smaller accessible workspace to fully execute the path. While this path will certainly cover the part, the adaptive methods used allow for more contextualized paths that would not be possible with a simple slicing method on the convex hull. The natural alternative to this would be to forgo any convex hull path and then slice the part itself directly. However, the idiosyncrasies of the resulting path would more than likely be very difficult for a robot to achieve in a smooth fashion.

This adapted path is especially useful in situations where quality path plans are needed in a timely manner that do not require perfect precision, without any or minimal human interaction. As mentioned previously, one-off path planning is a good application for this methodology, along with being used as a starting point for mass produced toolpaths that can be tweaked by a human operator. While these adapted paths do a good job of accommodating the surface topology of the original part despite being built from the convex hull, the paths could be improved by defining a safe way to invade the convex hull without creating any collision issues. There are a variety of tool accessibility and visibility algorithms available to verify whether or not a defined path is achievable or not [41]. However, the computational load required for these methods would significantly slow down the path building process. With that being said, it is certainly possible to improve computational time by reworking the algorithms and using better processors to a point where this difference is negligible.

As discussed previously, there are some shortcomings to this methodology. Ideally, every decision should be made with as much relevant data as possible, but sometimes different data types can skew the results. In this case, using the incidence angle as an indicator of how much the path offset distance should be adjusted would cause the distance to be adapted too much to the point where the sprayer width is narrower than the offset width. This causes gaps in the raster pattern which then requires replanning later on. Alternatively, using distance from the part to influence the velocity of the end effector does not cause the same path breaking issues, but it doesn't necessarily help all that much either. If the end effector is already too far away, the overall performance of the path would be much better served by adjusting the distance rather than extending the exposure time and thus lengthening the overall execution time. Another drawback to slicing based methods is that there are usually two sides left uncovered. When sliced along the Z axis, the top and bottom sides are the ones left uncovered. For complete coverage, the part can be sliced along a different axis to cover the uncovered sides. Currently, the slicing direction is determined from user input and is derived as being perpendicular to one of the three axes. For more accurate slicing, the slicing planes could be derived as being perpendicular to any arbitrary vector and methods could be developed for defining that vector based on the overall orientation of all of the facets. Theoretically, the ideal slicing direction would be perpendicular to the average normal vector of all facets, but the specifics require a more thorough investigation.

As more and more complex path planners are developed, the benefit of saving time by only adapting the distance may fade depending on the process parameters. When the distance is shortened too much and the overlap percentage is not large enough to compensate, there is the possibility of gaps being left in between passes of the sprayer as described above. In order to ensure complete coverage, the adjusted section would need to be replanned with a thinner slice thickness, which would result in multiple passes and ultimately more time used. Currently, this replanning is achieved through a separate process that requires user feedback, but in the future a dynamic system that can achieve this on the fly is ideal. Regardless, these path plans provide a good starting point for human iteration by identifying the areas on the part that need more cleaning, whether this be in a collaborative system or as the beginning of a path plan for a mass-produced part. Eventually, these systems will get to the point where they can effectively cover any part placed within their reachable volume on their own with no human intervention required.

4. Conclusion

The widespread use of robots in manufacturing will inevitably lead to completely autonomous systems being the standard. By first considering the convex hull and then making adjustments for the original section, a great deal of the challenging geometrical issues are eliminated from the equation in this two-part method of generalised path planning. Although both approaches do improve toolpaths and may be justified depending on the time trade-off on that part, adjusting based on distance is clearly the most effective way to achieve better results when comparing adaptive methods and considering tradeoffs for the simplified paths. If nothing else, these results show that good toolpaths may be achieved with some sacrifice of precision, even without complex and time-consuming path design methods. This method does offer a solid foundation for agile route planning in industrial spraying operations involving complex, unique components, but there is certainly room for improvement in many areas going forward. Collaborative robotic pressure washing work cell installation was initiated. Both the initial planning and rework modules of the framework may make use of the path planner discussed in chapter 3, while the framework outlined in chapter 2 establishes standards for system design and infrastructure. The pressure washing work cell served as the inspiration for this project, and a prototype system was developed to represent it while designing the framework and path planner. As it stands, the prototype system incorporates all the ideas presented in the papers with the exception of the work cell's actual implementation. Before the adaptive path planner takes control and develops an initial path plan, the user is asked to submit the first 3D data and input settings. Once the user has seen the visualisation, they may either approve the path or choose which parts to replanning. You may either add the newly planned route to the existing one or completely change it if replanning is necessary; either way, you'll be asked for certain input parameters again. In the future, there is still room for improvement in the analytical methods used to predict the route plan's coverage of facets. This is particularly true for parts with complicated geometries that the sprayer could miss. When adaptive measures reveal previously covered components, there needs to be a more sophisticated path planner that can dynamically alter the path's structure. Creating a real prototype from a digital one is the next logical step, after the research-based advancements. Two things must occur for this to be accomplished. Before the robot can begin the work, the route planner's output must be reviewed and adjusted to account for its physical constraints. Secondly, a method must be established to collect the part's 3D data in relation to the robot.

REFERENCES

[1] A. Woods and H. Pierson, "Developing an Ergonomic Model and Automation Justification for Spraying Operations," in Institute of Industrial and Systems Engineers Annual Conference, Orlando, FL, 2018.
[2] H. Chen, T. Fuhlbrigge and X. Li, "Automated Industrial Robot Path Planning for Spray Painting," in 4th IEEE Conference on Automation Science and Engineering, Washington, D.C., 2008.
[3] J. S. Oh, Y. H. Choi, J. B. Park and Y. F. Zheng, "Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, vol. 51, pp. 718-726, 2004.
[4] Z. He, B. Lu, J. Hong, Y. Wang and Y. Tang, "A novel arc-spraying robot for rapid tooling," International Journal of Advanced Manufacturing Technologies, 2007.
[5] W. Chen and D. Zhao, "Path Planning for Spray Painting Robot of Workpiece Surfaces," Mathematical Problems in Engineering, 2013.
[6] W. Sheng, N. Xi, M. Song, Y. Chen and P. Macneille, "Automated CAD-guided robot path planning for spray painting of compound surfaces," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000.
[7] A. M. Kabir, J. D. Langsfeld, S. Shriyam, V. S. Rachakonda, C. Zhuang, K. N. Kaipa, J. Marvel and S. K. Gupta, "Planning Algorithms for Multi-Setup Multi-Pass Robotic Cleaning with Oscillatory Moving Tools," in IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, 2016.
[8] Y.-H. Choi, T.-K. Lee, S.-H. Baek and S.-Y. Oh, "Online Complete Coverage Path Planning for Mobile Robots Based on Linked Spiral Paths Using Constrained Inverse Distance Transform," in The IEEE/RSJ International Conference on Intelligent Robots and Systems , St. Louis, MO, 2009.
[9] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," Rovotics and Autonomous Systems, vol. 61, pp. 1258-1276, 2013.
[10] C. Luo, S. X. Yang, D. A. Stacey and J. C. Jofriet, "A Solution to Vicinity Problem of Obstacles in Complete Coverage Path Planning," in IEEE International Conference on Robotics 8 Automation, Washington DC, 2002.
[11] J. H. Lee, J. S. Choi, B. H. Lee and K. W. Lee, "Complete Coverage Path Planning for Cleaning Task using Multiple Robots," in IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, 2009.
[12] S. X. Yang and C. Luo, "A Neural Network Approach to Complete Path Planning," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, vol. 34, pp. 718-724, 2004.
[13] A. UGUR, "Path planning on a cuboid using genetic algorithms," Information Sciences, vol. 178, pp. 3275-3287, 2007.
[14] J. C. Rubio, J. Vagners and R. Rysdyk, "Adaptive Path Planning for Autonomous UAV Oceanic Search Missions," in AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, 2004.
[15] F. Schwarzer, M. Saha and J.-C. Latombe, "Adaptive dynamic collision checking for single and multiple articulated robots in complex environments," IEEE Transactions on Robotics, vol. 21, no. 3, pp. 338-353, 2005.
[16] R. Bohlin and L. Kavraki, "Path planning using lazy PRM," in IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.
[17] A. Pichler, M. Vincze, K. Hausler, H. Andersen and O. Madsen, "A Method for Automatic Spray Paiiitiiig of Uiiknowii Parts," in IEEE International Conference on Robotics & Automation, Washington DC, 2002.
[18] E. U. Acar, H. Choset, Y. Zhang and M. Schervish, "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods," The International Journal of Robotics Research , vol. 22, pp. 441-466, 2003.
[19] M. Hebert and E. Krotkov, "3-D Measurements From Imaging Laser Radars: How Good Are They?," in International Workshop on Intelligent Robots and Systems , Osaka Japan, 1991.
[20] F. Remondino and S. El-Hakim, "IMAGE-BASED 3D MODELLING: A REVIEW," The Photogrammetric Record, vol. 21, pp. 269-291, 2006.
[21] F. Blais, "Review of 20 years of range sensor development," Journal of Electronic Imaging, vol. 13, pp. 231-240, 2004.
[22] D. K. Pai, K. v. d. Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond and S. H. Yau, "Scanning Physical Interaction Behavior of 3D Objects," in SIGGRAPH, Los Angeles, CA, 2001.
[23] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," in SIGGRAPH, New Orleans, LA, 2000.
[24] C.-F. Huang, Y.-C. Tseng and L.-C. Lo, "The Coverage Problem in Three-Dimensional Wireless Sensor Networks," in IEEE Globecom, Dallas, 2004.
[25] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes and H. Bruyninckx, "Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty," The International Journal of Robotics Research, vol. 26, pp. 433-455, 2007.
[26] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation – a review," Industrial Robot: An International Journal , pp. 64-70, 2005.
[27] P. Meng, E. S. Geskin, M. C. Leu, F. Li and L. Tismeneskiy, "An Analytical and Experimental Study of Cleaning With Moving Waterjets," J. Manuf. Sci. Eng., vol. 120, no. 3, pp. 580-589, 1998.
[28] S.-H. Suh, I.-K. Woo and S.-K. Noh, "Development of An Automatic Trajectory Planning System (ATPs) for Spray Painting Robots," in Intemational Conference on Robotics and Automation, Sacremento, CA, 1991.
[29] H. Chen, W. Sheng, N. Xi, M. Song and Y. Chen, "Automated Robot Trajectory Planning for Spray Painting of Free-Form Surfaces in Automotive Manufacturing," in IEEE International Conference on Robotics 8 Automation, Washington DC, 2002.
[30] A. Djuric, R. J. Urbanic and J. L. Rickli, "A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems".
[31] ISO/TS 15066:2016 Robots and robotic devices -- Collaborative robots, International Organization for Standardization, 2016.
[32] P. Waurzyniak, "Putting Safety First in Robotic Automation," Manufacturing Engineering, pp. 61-66, September 2016.
[33] T. Anandan, "Robotic Industry Insights: Collaborative Robots and Safety," Robotic Inustries Association, 26 Jan 2016. [Online].

Available: https://www.robotics.org/content-detail.cfm?content_id=5908.
[34] Universal Robots, "Afraid to Commit? No Problem, with Flexible, Redeployable Cobots," 3 June 2016. [Online]. Available: https://blog.universal-robots.com/flexible-redeployable-cobots. [Accessed 30 April 2017].
[35] Robotiq, "Teaching Robots Welding," [Online]. Available: http://robotiq.com/solutions/robot-teaching/. [Accessed 6 May 2017].

[36] P. Stone and M. Veloso, "Towards collaborative and adversarial learning: a case study in robotic soccer," International Journal of Human-Computer Studies, vol. 48, pp. 83-104, 1998.
[37] R. Mitnik, M. Recabarren, M. Nussbaum and A. Soto, "Collaborative robotic instruction: A graph teaching experience," Computers and Education, vol. 53, pp. 330-342, 2009.
[38] S. Brown and H. Pierson, "A Collaborative Framework for Robotic Task Specification," Procedia Manufacturing, vol. 17, pp. 270-277, 2018.
[39] C. Connolly, "Cumulative generation of octree models from range data," in IEEE International Conference, 1984.
[40] Roman, "Find the shortest distance between a point and line segments (not line)," August 2017. [Online]. Available: https://stackoverflow.com/questions/27161533/find-the-shortest-distance-between-a-point-and-line-segments-not-line.
[41] D. Dakdouk, "Tool Accessibility with Path and Motion Planning for Robotic Drilling and Riveting," Ryerson University, 2016.
[42] D. Vinayagamoorthy, "Rotor Blade," 29 Ocotober 2017. [Online]. Available: https://grabcad.com/library/rotor-blade-8.
[43] C. Domingos, "Wing_Section_NACA23015C200," 2 January 2018. [Online]. Available: https://grabcad.com/library/wing_section_naca23015c200-1.
[44] F. H. Macias, "rueda motriz e inducida grua viajera," 21 September 2017. [Online]. Available: https://grabcad.com/library/rueda-motriz-e-inducida-grua-viajera-1.