Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

AUTOMATIC COLLISION FREE PATH PLANNING IN HYBRID TRIANGLE AND POINT MODELS: A CASE STUDY

Sebastian Tafuri
Evan Shellshear
Robert Bohlin
Johan S. Carlson

Fraunhofer-Chalmers Research Centre
Sven Hultins Gata
SE-41288 Gothenburg, SWEDEN

ABSTRACT

Collision free path planning is a key technology for assembly analysis, robot line optimization, and virtual assessment of industrial maintenance and service. The ability to compute collision free paths relies on the ability to quickly and robustly query the proximity of the planning object to its surroundings. Path planning with triangulated models is a well studied problem, however, hybrid models comprising both points and triangles present new and difficult challenges. Working directly with point clouds is becoming more relevant because it allows one to scan existing industrial installations and path plan with the scan data instead of possibly incorrect planned layouts. In this paper we implement and analyze a new hybrid path planning interface on a case study in robot line manufacturing and demonstrate its feasibility in comparison to an existing CAD model of the work environment and show that triangulating the original point cloud is undesirable for path planning.

1 INTRODUCTION

Point clouds have become a ubiquitous part of many industrial models in the last 15 years. The ability to visualize billions of points in order to accurately model real world industrial settings has become a reality. More and more software tools are now offering the ability to work with enormous point clouds in real time (Advanced Technical Solutions AB 2012). This has rendered the use of point clouds for simulations in industrial settings feasible, in particular within and around factories. Point clouds offer the user the ability to acquire valid representations of the real state of the environment under consideration and not just the nominal or planned appearance (Berlin 2002). Scanned point cloud models are also able to provide up-to-date information about local changes, which is often much easier to acquire than updating a given CAD model. Most importantly, however, for industrial settings that lack a CAD (or similar) model, one can scan an entire factory much easier than building a new CAD model from scratch.

The use of scanned point clouds, however, requires that the object to be scanned exists and hence for conceptual or unrealized factory environments or parts, this is a problem. Most such virtual models are therefore created using CAD or similar software environments. Hence, one cannot completely rely on point clouds for all modeling purposes and the challenge has been to combine the two styles of representation in a sensible manner. Such a synergy can allow greater flexibility and accuracy for scenarios where one wishes to augment a given point scanned industrial installation with new CAD structures. Such developments have given designers unprecedented levels of accuracy that were not available merely with the original CAD models.

With the rise of such models, one desire of numerous simulators is to be able to perform his or her chosen simulation in a hybrid point-triangle environment (or whatever other combination of point clouds

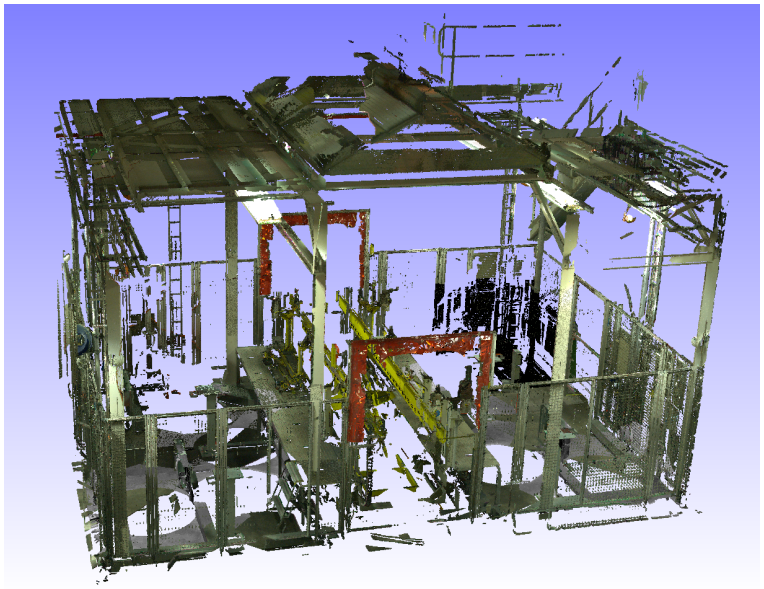

Figure 1: A point cloud with tens of millions of points.

one is interested in). One such simulation that is essential for numerous industries is robot path planning. The goal of robot path planning is to be able to compute a valid motion for a robot to carry out a set of given tasks, such as welding, bending, assembling, etc. (Latombe 1991). Efficiently being able to path plan in a real factory environment, where the robot's work environment is filled with obstacles, is still a major challenge for current research (Spensieri et al. 2010). With the current ability to represent the work environment as a point cloud, one is presented with new challenges to path plan in such hybrid environments.

The rest of the paper is organized as follows. In the next section we give a literature review of path planning in point clouds and triangle meshes as well as a review of the geometric data structures used to carry out proximity queries in such situations. In Section 3 we describe our interface to a path planning algorithm to make path planning in hybrid triangle point cloud geometries possible. In Section 4 we investigate and implement a new hybrid point-triangle path planner and test it on a well-known industrial test case (Spensieri et al. 2008; Segeborn et al. 2010) and present our results. In Section 5 we conclude.

2 LITERATURE REVIEW

Path planning for triangle meshes is a well studied problem and numerous books and reviews on the subject exist (Latombe 1991; LaValle 2006). Popular methods used in real life applications are often based on some type of sampling method, e.g. Probabilistic Roadmap Method (Kavraki et al. 1996; Bohlin and Kavraki 2000), Rapidly Exploring Random Trees (LaValle and Kuffner Jr 2001), etc. In most cases these algorithms can be adapted without much effort to the case of point clouds. Note, however, there exist path planning methods that are designed to work directly with point clouds, e.g. for autonomous robots that sample their environments with stereo vision (Landa et al. 2007; Landa 2008; Sucan, Kalakrishnan, and Chitta 2010). The difference between these methods and the situation examined in this paper is that we are interested in applying such algorithms to hybrid combinations of triangle meshes and point clouds and not just one or the other. There are, however, few research papers that address this novel combination of points and triangles, exceptions being Sucan, Kalakrishnan, and Chitta (2010) and Pan, Chitta, and Manocha (2012). This is most likely due to the ability to triangulate the scanned point cloud and then path plan with the computed triangle mesh. For small point clouds this method is feasible and has been shown to work well (Dupuis et al. 2008), however for point clouds comprising tens or hundreds of millions (even

billions) of points, accurately and reliably triangulating them can be computationally intractable, as well as unnecessary, if one can work directly with the points.

The first step towards being able to carry out path planning between hybrid triangle meshes and point clouds, is the ability to detect collision and measure distances between a point cloud and a triangle mesh. Few solutions to this problem have been suggested in the literature. There exist efficient methods to carry out proximity queries between two triangle meshes (Larsen et al. 2000) as well as between two point clouds (Klein and Zachmann 2004; Pan, Chitta, and Manocha 2011), however, for proximity queries between hybrid triangle meshes and point clouds little has been done. In Johnson and Cohen (2004), a new method is presented to measure distances between numerous types of geometric primitives, however, one is only guaranteed a local shortest distance with potentially long query times (due to convergence issues). In Pan, Chitta, and Manocha (2012), a general purpose library is presented to handle many different types of geometric primitives as well as different types of queries between them. However, because our requirements are not as general as Pan, Chitta, and Manocha (2012), we have developed a solution tailored specifically to our needs[1]. This was done in order to simplify the structure of our algorithms as well as to utilize fast, specialized routines for the geometric primitives we are working with (see Section 3).

3 PATH PLANNING IN HYBRID TRIANGLE POINT CLOUDS GEOMETRIES

Our solution and method for path planning is centered around creating an appropriate interface between well-known path planning algorithms and hybrid triangle and point clouds environments. Hence, we will not focus on a specific path planning algorithm but simply use an existing algorithm that only requires information about distances between objects. An appropriate such algorithm is the one used in the IPS software package (Industrial Path Solutions 2012) and hence from hereon all path planning results will be with respect to the path planning algorithm in IPS.

In IPS the path planning and optimization of multi robot stations is divided up into numerous steps and we will indicate here where our new hybrid proximity query package is involved. Given a set of robots and tasks for them to complete, one must first compute which tasks are feasible to reach in a collision-free configuration, hence this step requires our new hybrid approach. The tasks are then balanced between the robots (so called load balancing) to even out the work load between robots and reduce the total cycle time for the set of tasks. The load balancing step does not require explicit distance queries so this step is not affect by our new approach. With the workload balanced between the robots, the robot motions are generated by finding collision free paths between the tasks using the task planner coupled with our new hybrid approach. Finally the movements of the robots are coordinated and synchronized so that they do not collide with each other. This is done by rearranging execution order and modifying robot velocities. Since the robots are usually represented as triangle meshes, this final step can be done with triangle to triangle proximity queries and also does not influence the other steps, so analyzing its computation is superfluous and hence will be omitted in Section 4. The path planning and optimization algorithm is represented schematically in Figure 2.

In order to be able to measure distances between triangles and points we chose to adapt the well-known bounding volume hierarchy (BVH) solution (Hubbard 1993), with rectangular swept spheres (Larsen et al. 2000) (RSSs) as our chosen bounding volume (BV), see Figure 3. Rectangular swept spheres are constructed by taking the Minkowski sum of some core primitive shape and a sphere. To create the set of bounding volumes that can be defined by RSSs one can have a point, line or rectangle as the core primitive (resulting in a sphere, lozenge or RSS), the first two cases being considered as degenerate rectangles. The superiority of computing proximity queries with RSSs over other types of bounding volumes has been demonstrated in previous publications (Larsen et al. 2000; Pan, Chitta, and Manocha 2012). By enclosing the entire triangle or point model in an RSS one is then able to build a BVH top-down by repeatedly splitting the

[1]We do not require testing between cylinder, boxes, spheres, capsules, cones, etc. as Pan, Chitta, and Manocha (2012) have included. This allows us to avoid slower general algorithms such as the GJK algorithm that is used there. See Pan, Chitta, and Manocha (2012) for more details.

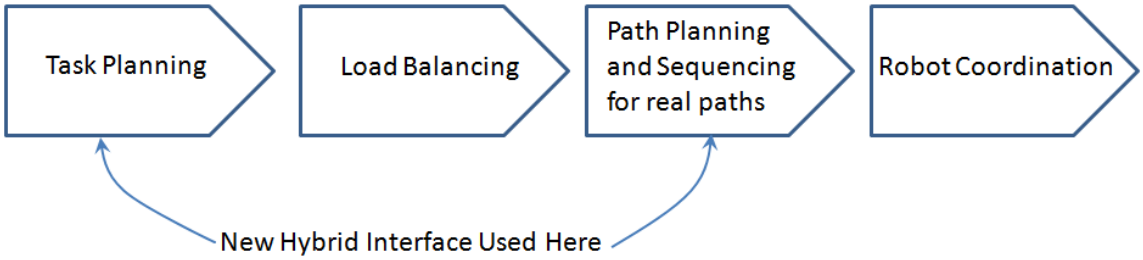

Figure 2: The path planning algorithm in IPS.

set of triangles or points into two groups and computing RSSs to enclose the groups of split triangles or points until one has reached a predefined number of triangles or points that constitute a leaf of the BVH. For more details on BVHs as well as using them for different types of proximity queries, the reader is referred to Larsen et al. (2000).

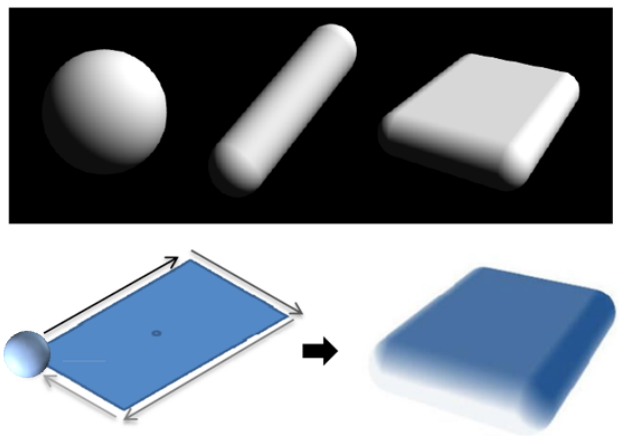

Figure 3: Different examples of rectangular swept spheres as well as their construction.

When working with points, we chose not to use surfels or implicit functions to represent the surface as they can be prone to robustness issues (Steinemann et al. 2007). Hence, to compute distances between points and triangles we chose to work directly with the points. This obviously presented numerous challenges and affected the way we built our BVHs. This forced us, first and foremost, to create our BVHs in such a way that we can simultaneously process points and triangles within the same BVH. Because or code was written in C++ one has a number of options to overcome these difficulties. The method chosen by Pan, Chitta, and Manocha (2012) is to use virtual inheritance to be able to choose the type of object contained in the BVH. The use of virtual methods, however, can impose performance penalty by not being able to inline short simple functions (such as distance computations) and also increase memory costs (at least 4 additional bytes per object) via use of pointers to virtual function objects. To avoid these drawbacks we instead decided to use as much compile time templates as possible which also allows the compiler to create more optimized code.

A major issue one can have when path planning in point clouds is the handling of the enormous numbers of points. Because we chose to work directly with the scanned point data, we required a high density of point sampling to be able to acquire accurate results. This results in high memory requirements and

potential memory bottlenecks. To avoid such situations, we allowed a level of flexibility in choosing the number of points in each leaf of the BVH, i.e. we allowed more than one point hence reducing the number of levels in the BVH. This will however potentially increase the size of the leaf nodes causing decreased performance. To limit this effect we added an additional density measurement to each leaf BV (number of points to a given volume) and if the density was too low, then we subdivide the leaf. In addition, we also had the option to build our BVH in a lazy fashion, only building the first few levels of the BVH and then building the other levels as needed, on the fly. This allowed us to avoid building BVHs for sections of the scene that could not be reached by any robot in the path planning simulations resulting in significant memory savings.

4 SIMULATION RESULTS

The simulation scenario that we consider is a well-known stud weld station, (Spensieri et al. 2008; Segeborn et al. 2010), see Figure 4. The station was scanned with a high resolution and then filtered to reduce noise and other scanning artifacts. The scanned points were then transformed into a point cloud with a 3mm uniform point spacing. While the scanning was being done, the stud weld robots were located in the point cloud. To be able to carry out our analysis we separated the scanned robots manually and replaced them by CAD models from the robot manufacturer.

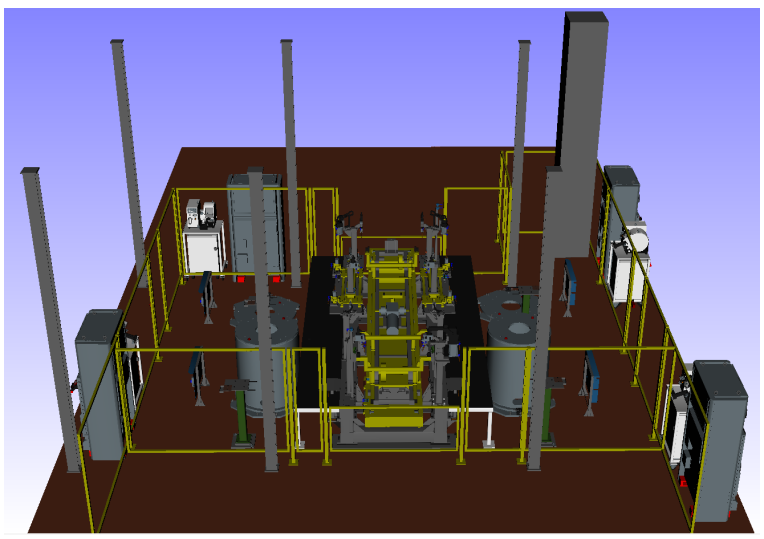

Figure 4: CAD Stud weld station.

An important difference between the CAD model and the scanned point cloud is the additional geometry such as the roof over the robot cell, extra side caging, etc. (compare Figure 4 and Figure 1) in the scanned version. After creating the scanned model one has the option of removing such parts of the point cloud because they do not directly effects whether a robot can carry out the path planning tasks or not. However, we will show later that the effect of the additional geometry has no negative effect on the running time of the algorithm in our examples. Hence, for scenes such as ours, which do not have very large dimensions in comparison to the robot's work environment, one can use the entire environment (if memory and other factors permit).

The robot models used for our simulations were CAD models of two ABB stud weld robots each comprising about 200,000 triangles. The robots have six axes and hence six degrees of freedom to be able to reach a large number of locations, see ABB (2012) for more details. For our path planning example, we created 24 stud weld tasks distributed over the surface of a car chassis such that all 24 tasks were reachable by both robots. Each robot could reach each task with multiple kinematic configurations. This was, in

part, due to the stud weld tool being symmetric and can therefore be rotated around an axis normal to the plane in which the stud weld lies. By discretizing the axis of rotation one can find an arbitrarily accurate discretization of all valid collision free poses of a robot for each task. See Figure 5 for a picture of the robots and chassis. In all simulations the set of stud weld locations remained the same.

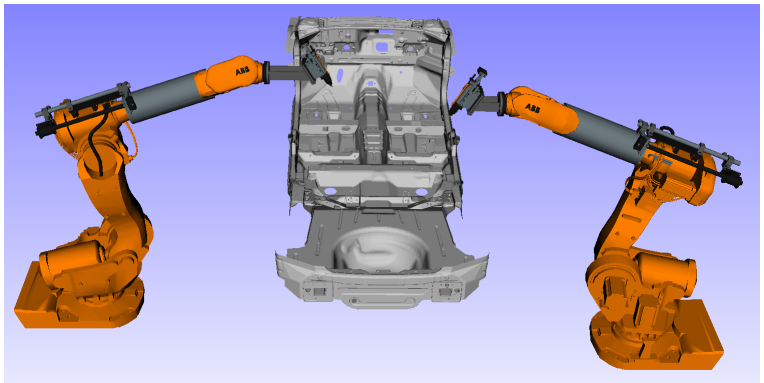

Figure 5: Robots and car chassis.

To test our new hybrid proximity query interface and also to test the viability of working directly with point clouds instead of triangle meshes, we set up six different simulation scenarios with the two robots and car chassis using different models of the stud weld station described above. The actual path planning process was carried out as described in the previous section. In the first scenario we path planned with the surrounding environment being a CAD model of a robot cell, Figure 4. The CAD model contained 1.3 million triangles. In the next scenario we did the same but with a high resolution point cloud model of the same robot cell, Figure 1. The point cloud contained a total of just over 15 million points (3mm uniform density). The next scenario involved the same point cloud model of the robot cell but with a lower point resolution containing 4 million points (6mm uniform density). In the next scenarios we triangulated the high and low resolution point cloud models and path planned with it as the surrounding environment. This triangulation was carried out in the Geomagic Studio software (Geomagic studio 2012), where we used the settings ”Optimize for evenly spaced triangles” as well as not deleting small components. The triangulation of the high resolution and low resolution point clouds contained 24 million and 6.7 million triangles respectively, see Figure 6 for a picture of the low resolution triangulation. Both of these triangulations were about three times larger, in terms of memory costs, than the original point clouds. This was due to the triangulation needing to repeat points for each triangle and resulting in little saving via indexation of points. In addition, the BVH memory costs were approximately four times as much for the high and low resolution triangle meshes in comparison to the point clouds.

To test the affect of the additional points and geometry around the station that is not present in the CAD model, we created a fifth model of the stud weld station by removing 2.4 meters of height from the top of the scene, leaving the new point cloud with 8 million points. This was done in an effort to mimic the layout of the CAD geometry, see Figure 7. Note that even after removing the top part of the scene the point cloud still contained geometry not present in the CAD model. This extra geometry, however, comprised only approximately 10 percent of the remaining points and did not affect the reachability of our stud weld tasks. Finally to be able to have a base case to compare the six cases with, we also carried out the simulation with just the robots and car chassis and no extra surroundings.

In each of the seven previous scenarios we measured the time it took for the path planning algorithm to compute each of the first three steps in the path planning process described in Section 3 as well as the total time, the number of reachable tasks (No. of features reached in Table 2), the total number of achievable robot axes configurations for all stud weld tasks (based on a fixed discretization of each degree of freedom, No. of configurations in Table 2) as well as the time for each robot to carry out the computed path (Stud

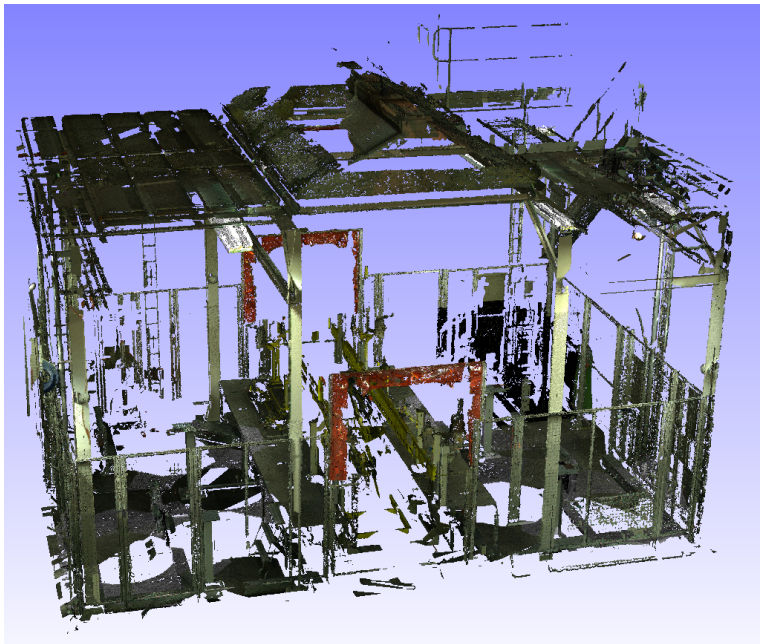

Figure 6: Triangulation of Figure 1.

Table 1: Simulation Results. High Res. and Low Res. Point Cloud are the high resolution and low resolution point cloud respectively. High Res. and Low Res. Triangulation are the triangulations of the high and low resolution point clouds.

Surrounding Geometry	Task Plan (s)	Load Balance (s)	Path Plan (s)	Total Time (s)
Original CAD Geometry	1.438	12.03	116.94	128.97
High Res. Point Cloud	1.531	11.00	142.58	153.58
Low Res. Point Cloud	1.64	11.50	193.08	204.58
High Res. Triangulation	1.437	10.94	200.13	211.07
Low Res. Triangulation	1.297	82.84	760.41	843.25
Point Cloud with no Roof	2.141	11.31	244.27	255.58
None	1.157	11.80	43.41	55.21

Weld Time (s) in Table 2). The simulations were carried out on an Intel® Core™ i7 CPU (quad core), 2.67 GHz with 8 GB of RAM under Windows 7 in C++ using Microsoft Visual Studio 2010. The results of each of the seven simulations are presented in Table 1 and 2. Note that when computing the time for the robot to complete the path planned motions (Stud Weld Time in Table 2), we assumed a constant speed for the robot links as well as infinite acceleration and deceleration so as to have a fair comparison between paths. Figure 8 shows the robots in the low resolution point cloud carrying out the stud weld tasks.

4.1 Discussion of Results

As can be seen from the results there are no major differences between the four scenarios, Original CAD Geometry, Low Res. Point Cloud, High Res. Point Cloud, High Res. Triangulation and Point Cloud with No Roof (the last case being the slowest to compute the collision free paths, see Path Plan (s) in Table 1). This demonstrates the claim that path planning in point scanned versions of the original CAD Geometry is a viable option. In addition, one notices that in our case study there is no major advantage in removing the roof from the point cloud scene nor to reduce the density of points. We notice, however, in the triangulation

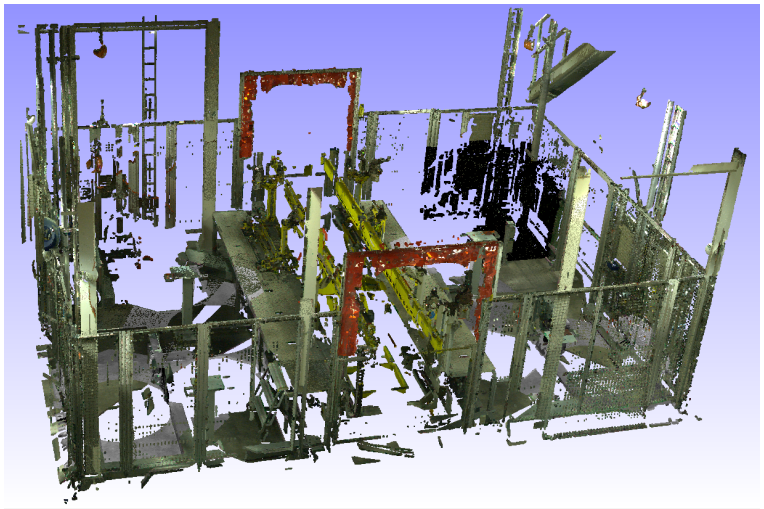

Figure 7: Figure 1 with the roof and ceiling removed.

Table 2: Simulation Results cont.

Surrounding Geometry	No. of features reached	No. of configurations	Stud Weld Time (s)	
			Robot1	Robot 2
Original CAD Geometry	24	30591	33.203	34.411
High Res. Point Cloud	24	27609	32.995	34.294
Low Res. Point Cloud	24	27605	33.008	34.426
High Res. Triangulation	24	27628	32.618	34.206
Low Res. Triangulation	24	27686	33.545	33.693
Point Cloud with No Roof	24	27609	32.995	33.608
None	24	30591	33.462	34.246

of the low resolution point cloud, the path planner required significantly more time to find collision free paths between the stud weld tasks (Path Plan (s) in Table 1). This time difference was caused by larger triangles making robot movements more difficult (which were not present in the triangulation of the high resolution point cloud) and resulted in higher load balancing times. The higher load balancing times were due to the path planning having to attempt five different allocations of stud weld tasks between the two robots, with different tasks sequences, to finally find a collision free path between them. This second result also confirms our other claim that it is unnecessary to triangulate the point clouds to path plan in them and in fact, for one of the cases in our case study, it was disadvantageous.

One does notice, however, that in the original CAD model the robots have 10 percent more available axes configurations for each stud weld task (i.e. 30591 instead of approximately 27650), due to the CAD geometry not containing obstructive geometry present in the real and scanned station. This combined with the lower triangle count resulted in the path planning in the CAD environment being faster. The times to carry out the stud welding (i.e. the quality of the computed paths, Stud Weld Time (s) in Table 2) are not significantly different amongst all test cases so that one can differentiate the methods in this respect. In addition, the computation times for the task planning step were all similar enough also to be able to draw the conclusion that the surroundings in each case did not significantly influence this step.

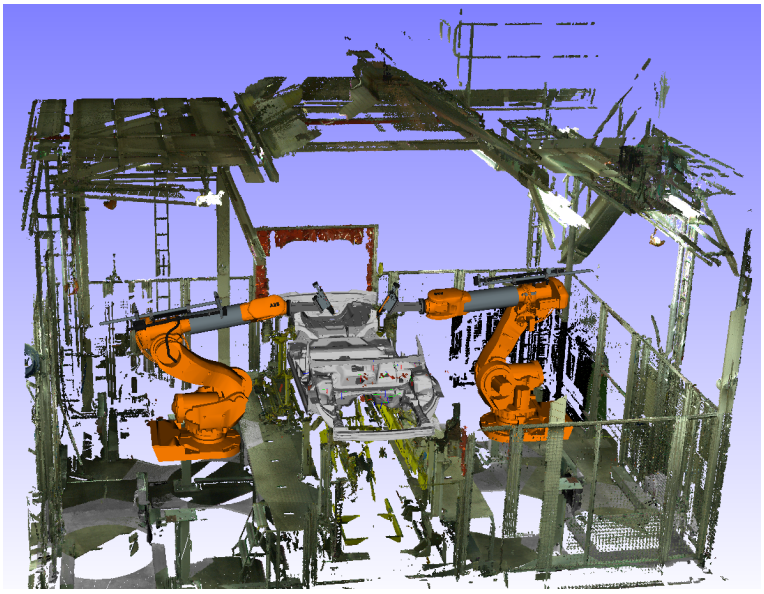

Figure 8: The robots carrying out the stud weld tasks in the low resolution point cloud.

4.2 Drawbacks and Limitations

One of the major drawbacks of the results presented here is that we only used one case study to examine the different models. However, based on the numerous other cases the authors have examined, we expect that the results presented here would carry over to many other scenarios nonetheless. It should be noted that it may be possible to employ improved filtering to increase the path planning performance further and reducing the performance difference between scanned data and CAD.

Another drawback of using scanned pointclouds is that moving parts, like robots, clamps, tracks and other moving equipment, must either be scanned separately, replaced with CAD geometry or extracted from the complete pointcloud. This to allow for independent movement and collision testing during path planning.

5 CONCLUSION

In this article we have demonstrated a new method to carry out proximity queries to facilitate path planning in hybrid triangle and point cloud geometries. As was seen in our case study presented here, it can actually be a disadvantage to triangulate the scanned point cloud. We have also demonstrated that a scanned environment can be used as a substitute to CAD for path planning purposes. Future directions involve increasing the set of examples studied here to many more path planning cases. It would be interesting to see if the results presented here carry over to other hybrid triangle and point cloud environments and other types of path planning tasks.

ACKNOWLEDGMENTS

We sincerely thank Rolf Berlin and David Johansson for their assistance with and provision of the point based models and triangulations thereof used in this article. This work was carried out at the Wingquist Laboratory VINN Excellence Centre, and is part of the Sustainable Production Initiative and the Production Area of Advance at Chalmers University of Technology. It was supported by the Swedish Governmental Agency for Innovation Systems.

REFERENCES

ABB 2012. "ABB". Accessed April 30, 2012. http://www.abb.com/product/seitp327/B863009BB251FE17C1257227002F552F.aspx.

Advanced Technical Solutions AB 2012. "Advanced Technical Solutions AB". Accessed April 30, 2012. http://www.ats.se/.

Berlin, R. 2002. "Accurate robot and workcell simulation based on 3D laser scanning". In *Proceedings of the 33rd ISR (International Symposium on Robotics)*.

Bohlin, R., and L. Kavraki. 2000. "Path planning using lazy PRM". In *IEEE International Conference on Robotics and Automation*, Volume 1, 521–528. IEEE.

Dupuis, E., I. Rekleitis, J. Bedwani, T. Lamarche, P. Allard, and W. Zhu. 2008. "Over-the-horizon Autonomous Rover navigation: experimental results". In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Los Angeles, CA*.

Geomagic studio 2012. "Geomagic studio". Accessed April 30, 2012. http://www.geomagic.com/en/products/studio/overview/.

Hubbard, P. 1993. "Interactive collision detection". In *IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, 24–31. IEEE.

Industrial Path Solutions 2012. "Industrial Path Solutions". Accessed April 30, 2012. http://www.industrialpathsolutions.com/.

Johnson, D., and E. Cohen. 2004. "Unified distance queries in a heterogeneous model environment". In *ASME DETC*. ASME.

Kavraki, L., P. Svestka, J. Latombe, and M. Overmars. 1996. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". *IEEE Transactions on Robotics and Automation* 12 (4): 566–580.

Klein, J., and G. Zachmann. 2004. "Point cloud collision detection". In *Computer Graphics Forum*, Volume 23, 567–576. Wiley Online Library.

Landa, Y. 2008. *Visibility of point clouds and exploratory path planning in unknown environments*. Ph.D. thesis, Department of Mathematics, University of California, Los Angeles, California. Available via ftp://ftp.math.ucla.edu/pub/camreport/cam08-20.pdf [accessed April 19, 2012].

Landa, Y., D. Galkowski, Y. Huang, A. Joshi, C. Lee, K. Leung, G. Malla, J. Treanor, V. Voroninski, A. Bertozzi et al. 2007. "Robotic path planning and visibility with limited sensor data". In *American Control Conference*, 5425–5430. IEEE.

Larsen, E., S. Gottschalk, M. Lin, and D. Manocha. 2000. "Fast distance queries with rectangular swept sphere volumes". In *IEEE International Conference on Robotics and Automation*, Volume 4, 3719–3726. IEEE.

Latombe, J.-C. 1991. *Robot motion planning*. Springer.

LaValle, S. 2006. *Planning algorithms*. Cambridge Univ Press.

LaValle, S., and J. Kuffner Jr. 2001. "Randomized kinodynamic planning". *The International Journal of Robotics Research* 20 (5): 378–400.

Pan, J., S. Chitta, and D. Manocha. 2011. "Probabilistic Collision Detection between Noisy Point Clouds using Robust Classification". In *International Symposium on Robotics Research*. IFRR. Available via http://www.isrr-2011.org/ISRR-2011//Program_files/Papers/Pan-ISRR-2011.pdf [accessed April 25, 2012].

Pan, J., S. Chitta, and D. Manocha. 2012. "FCL: A General Purpose Library for Collision and Proximity Queries". In *IEEE International Conference on Robotics and Automation*. IEEE. Available via http://cs.unc.edu/~panj/fcl.pdf, [accessed April 25, 2012].

Segeborn, J., D. Segerdahl, J. S. Carlson, A. Carlsson, and R. Söderberg. 2010. "Load Balancing of Welds in Multi Station Sheet Metal Assembly Lines". *ASME Conference Proceedings* 2010 (44274): 625–630.

Spensieri, D., J. S. Carlson, R. Bohlin, and R. Söderberg. 2008. "Integrating Assembly Design, Sequence Optimization, and Advanced Path Planning". *ASME Conference Proceedings* 2008 (43253): 73–81.

Spensieri, D., F. Ekstedt, J. Torstensson, R. Bohlin, and J. Carlson. 2010. "Throughput Maximization by Balancing, Sequencing and Coordinating Motions of Operations In Multi-Robot Stations". In *Proceedings of the 8th International NordDesign Conference 2010*, 455–465.

Steinemann, D., M. Otaduy, M. Gross, B. Chen, and M. Zwicker. 2007. "Efficient bounds for point-based animations". In *Proc. IEEE/Eurographics Symposium on Point-Based Graphics*, 57–64.

Sucan, I., M. Kalakrishnan, and S. Chitta. 2010. "Combining planning techniques for manipulation using realtime perception". In *IEEE International Conference on Robotics and Automation*, 2895–2901. IEEE.

AUTHOR BIOGRAPHIES

SEBASTIAN TAFURI is a researcher at the Fraunhofer-Chalmers Research Centre for Industrial Mathematics. He received his master degree in Computer Science in 2004 from Chalmers University of Technology, Sweden. His main research interests are in computational geometry, computer graphics and algorithms.

EVAN SHELLSHEAR is a researcher at the Fraunhofer-Chalmers Research Centre for Industrial Mathematics. He received his B.Sc./B.A. at the University of Queensland in Australia in 2004 and his Diplom and Ph.D. at the University of Bielefeld in Germany in 2008. His research interests are in computational geometry and its applications, especially within computer graphics.

ROBERT BOHLIN is a senior researcher at Fraunhofer-Chalmers Research Centre for Industrial Mathematics. He received his master degree in Automation Engineering in 1997 and his Ph.D. in Industrial Mathematics in 2002 from Chalmers University of Technology, Sweden. His main research interests are in robot path planning, computational geometry and optimization.

JOHAN S. CARLSON is the Vice Director of the Fraunhofer-Chalmers Research Centre for Industrial Mathematics, FCC, and is heading the department of Geometry and Motion Planning. He received a PhD in mathematical statistics in 2000 from Chalmers University of Technology, Sweden. His research interests includes methods, algorithms and tools for virtual product realization and in particular geometric simulation and assurance. He is, together with Dr. Robert Bohlin, originator and owner of the commercial software tool Industrial Path Solutions (IPS). He is a Member of the Governance Board Blekinge Schools of Engineering, of the board of Wingquist Laboratory at Chalmers and Deputy Director in the management group of Wingquist Laboratory Vinn excellence center at Chalmers.