Robotica (2001) volume 19, pp. 137–148. Printed in the United Kingdom © 2001 Cambridge University Press

CAD-based CMM dimensional inspection path planning – a generic algorithm

Yueh-Jaw Lin,* Rahul Mahabaleshwarkar* and Elena Massina†

(Received in Final Form: August 31, 2000)

SUMMARY

This paper describes a newly developed algorithm for CAD-based dimensional inspection path planning utilizing coordinate measuring machines (CMMs). The algorithm guarantees to provide optimum collision-free inspection probe paths by using the topological structure of the boundary representation (B-rep) solid CAD models of the inspected parts. A concept of automatic generation of optimum and collision free path in three dimensional space using available CAD database is proposed. The algorithm is generic for generating probe path over prismatic polyhedral parts. It serves as a principal part of the inspection path planning system. It is based on the modified ray tracing technique which uses a B-Rep data from any geometric modeling systems. Between start point and target point, an imaginary ray is established and if an intersection with part is encountered, an optimal detour path is created avoiding interference of the probe with the part. The generated path consists of linear segments joining start point and target point by various intermediate points. To locate these intermediate points in the consideration space, topological and geometrical structures of the part models are used at the time of decision making. To examine and implement the algorithm, a user-friendly application is developed employing AutoCAD Runtime Extension (ARX) development environment with object oriented programming (OOP) techniques, running on a Windows NT workstation. The effectiveness of the proposed algorithm is verified by the results of the implementation demonstrating optimum collision-free dimensional inspection path generation for four representative prismatic part models. All in all, this work contributes to the knowledge-base formation of automated dimensional inspection research area and paves a way for the integration of CMMs into a CAD/CAM environment, thus automate the process of design, manufacturing and quality assurance.

KEYWORDS: Generic algorithm; Path planning; CAD; CMM inspection

1. INTRODUCTION

With the emergence of Computer Integrated Manufacturing (CIM) technology, Coordinate Measuring Machines (CMM) are widely used as automatic inspection tools to ensure the quality of industrial products. To meet sophisticated functional specifications of products in modern manufacturing, it becomes cost effective to have faster and more accurate dimensional inspection systems. This demand forces CMM to inspect parts faster than ever before and their inspection procedures to be generated automatically.

CMM inspection has many advantages over conventional inspection approaches, which includes flexibility, efficiency, good accuracy and repeatability, higher level automation and capability of integration. Despite these advantages, many CMMs in industry are not fully utilized to their maximum capability, especially for programmability and inspection automation. This fact reduces these expensive and advanced machines to being operated at a lower level of automation. To efficiently and effectively employ CMMs in a computer integrated manufacturing environment, they must be used in coordination with CAD systems. CAD directed CMM inspection includes CAD directed inspection planning, CMM off-line programming and simulation, and sensor guided inspection plan execution and monitoring.

One of the most important issues of CAD directed dimensional inspection planning is path planning, i.e. generating the trajectory for the movement of CMM probe while measuring the required contact points on the part surface. The probe path should satisfy two primary conditions, namely, collision free and optimality. The path should avoid any interference of the probe with the part, fixtures, and CMM structure. The optimality condition contributes significantly towards higher throughput and more reliable data. Both these achievements ensure cost effectiveness of CMM operations.

From the review of the related past work done in CAD directed inspection area, it is observed that the area of finding optimal CMM path needs more research work. Past research efforts have emphasized generation of a collision free path, where major concern is to avoid interference of the probe and part. Aimed at improving the current level of technology in probe path generation, an effort has been made in this work to develop an automatic CMM path planning system, incorporating a new algorithm to produce not only a collision free, but also an optimal path. In light of this, the main objectives of the work are outlined as follows: (1) Design an automated probe path generation system which can serve as an integral part of the overall computer-aided inspection planning (CAIP) system. (2) Develop a generic algorithm for generation of optimum and collision free CMM probe path. (3) Implement the algorithm using commercially available CAD system and simulate the path

* Department of Mechanical Engineering, The University of Akron, Akron, OH 44325–3903 (USA)
† Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD 20899 (USA)

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

graphically. (4) Generate the data required for DMIS output to be used for an on line CMM analysis.

Based on the main objectives outlined above, this paper is organized as follows. Section 2 discusses the related work done by various researchers. In Section 3, a generic algorithm is explained which has the primary role of generating an optimum and collision free path for CMM probe movements. An automated probe path planning system is described which can work as an integral part of a CAIP system. Section 4 summarizes the implementation work and the application for testing the algorithm. The test suites and results after applying the algorithm are also given in this portion of the paper. Finally, conclusions and the scope of future work are presented in Section 5.

2. RELATED WORK

The study of probe path planning was necessary to understand the overall concept of a CAIP system. In this section, the relevant work in two aspects of CMM dimensional inspections, namely, *inspection process planning* and *probe path planning*, will be reviewed.

2.1 Inspection Process Planning

The objectives of a Computer Aided Inspection Planning (CAIP) system are to establish the best sequence of inspection steps, the detailed inspection procedure of each feature, feature accessibility by probes, probe path planning and collision checking, generating the CMM control commands (DMIS interface) and post-processing of measured data such as statistical and cost analysis. The research in this work focuses on developing optimal collision free CMM probe paths using B-Rep CAD models. For such a system, in addition to inspection knowledge and CMM related rules, information regarding geometry, tolerances and datum are needed. The analysis of traditional inspection processes show that the following generic steps are essential for any expert CAIP system.[1]

(i) Understanding part geometry and its inspection criteria as specified in the given CAD model.
(ii) Making decisions regarding inspection procedures to be followed, such as measurement sequence and probe path planning.
(iii) Producing machine instructions and measurement analysis.

Automatic inspection planning is a hierarchical heuristic process incorporating Artificial Intelligence (AI) techniques to make decisions wherever necessary. Prominent research efforts include the generic Object Oriented Inspection Planner (OOIP) reported in reference [2]; an Intelligent Planning Environment using CMM;[3] an Intelligent Inspection Planning in OOP environment demonstrated in reference [4].

In reference [2], the proposed generic Object Oriented Inspection Planner (OOIP) is a subsystem of an integrated STEP based product modeling system. It consists of two knowledge based planning subsystems, Inspection PRrocess Planner (IRIP) and Inspection PAth Planner (IPAP). The IPRP starts with generating a number of feasible plans and finding accessibility of each feature. It then minimizes part and probe orientation changes by inspecting each step in each initial plan, and sequences each inspection item with respect to each probe orientation according to the location and datum requirements. Finally, it removes duplicate items, finalizes probe order, determines measurement points and final part orientation. The IPAP subsystem uses the output of IPRP to perform local and global path planning. The local path planning finds the path to be followed on the same feature. While global path planning gets movements of the probe from one feature to another. Graphical simulations were done to validate the proposed algorithm in this work.

The intelligent planning environment system proposed in reference [3] consists of three key elements, namely, CATIA/CMM inspection planning module, CAD model based localization algorithm, and comparative analysis. Inspection attributes are processed along with CAD model by inspection planner according to the inspection knowledge and artificial intelligence (AI) techniques. Then, the inspection module determines inspection points and probing vectors on features by providing user interface for selecting target features. This is followed by a comparative analysis which statistically compares measurement data against specified dimensions and tolerances. Finally, it verifies inspection specifications and generates plans containing the number of inspection points, locations, and probing sequences.

The intelligent inspection planning in OOP environment system reported in reference [4] allows users to create features as objects which store all information about tolerances, process and functional requirements, geometrical and topological information based on CSG representations. User creates a solid model of desired object in the modeler, then the information extraction module retrieves all necessary information for planning. The system builds special data structures and links for searching. Inspection features are identified into three categories: primitives, surfaces and marked dimensions. Relationships are established between these data structures to facilitate searching information links and establishes inspection sequences.

The off line measurement planning system proposed in reference [5] has three modules. The data input module takes 3D solid models to establish measurement plans. The system requires user selection at decisions such as inspection features, probes and data, etc. Measurement planning module generates measuring sequence for selected tolerance. Finally, the statistical analysis module determines appropriate sample size and estimates form errors by comparing measured data with required data.

The generative inspection process planner described in reference [6] is a prototype CAIP system (IPPEX resembling a rule based expert system. IPPEX uses a product geometric modeler coupled with a dimensional and tolerance modeler to generate inspection instructions in the form of an operation plan and a DMIS part program. Resource data regarding machines, probes and fixtures are queried from a relational database. The IPPEX system consists of four major components. The user interface helps to define

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

inspection task, examine product model, direct inspection activities, interact with knowledge bases and modify system decisions. Product modeling system uses Cranfield B-Rep modeler for geometric information and CAM-I modeling system for storing dimension and tolerance information. These two modelers together provide applications with a product model.

2.2 Probe Path Planning

CMM probe path planning is nothing but to specify a suitable path for collision-free movement of a probe over a workpiece. Following this specific path, probe reaches the measurement point on the part surface and gets the coordinates of that point for further processing. CMM path planning can be considered as that of a Cartesian robot.[7] The primary approaches used by researchers in CMM probe path generation are,

(i) Ray tracing in octree database approach is highly dependent on hardware, more time consuming, uses old technique, can generate non-optimum jazzy path.[7]
(ii) Path on sculptured surface approach generates path only on free form surfaces, does not consider the whole solid together.[8]
(iii) Heuristics based path generation is a very limited approach as all possible collision conditions cannot be stored in the database. This method is unlikely to generate an optimum path.[3]
(iv) Polygon interfering with B-rep object model approach is very new and is under development. But this approach uses heuristics to detour the probe path which may not result in an optimal condition.[2]

In reference [7], Lu *et al.* proposed an algorithm for generation of an optimum CMM path primarily served for polyhedral solids. The algorithm uses a three dimensional ray tracing technique to search for colliding obstacles on a selected path. It uses the perspective projected silhouette contour of part to select vertices and edges of the polyhedra which the optimum path should follow in order to avoid collisions. The approach is based on configuration space method where the CMM probe is assumed as a point and part geometry is suitably modified so as to facilitate Octree representation. It simplifies the interference detection to collision of a point with the constraint space.

First, a tracing ray is formed between the start node and end node which searches through the octree database for collisions with FULL node. After that, the face of the octant where ray leaves and face of penetration with neighboring octant are identified from CAD data. Then, the adjacent node is again checked for collision with the same procedure. This algorithm is a complicated, computationally intensive methodology due to the extensive usage of the octree databases.

The CMM features accessibility and path generation algorithm developed by Lim and Menq[8] determines all feasible probe orientations for inspection of sculptured surfaces, as well as number of probe orientations and generates a collision free probe path. The feasible probe orientations ensure no collision between the probe and the part. Then, the number of probe orientations required to inspect the part are determined using some heuristic rules. Consequently, the algorithm creates a collision free path for simple or complex surfaces with holes.

The overall approach of this algorithm mainly involves the accessibility analysis to form the global and local accessibility cones of each inspection point. The accessibility cones are determined by the half space concept using the tangent plane to surface at the inspection point. Within the accessibility cone, probe is safe from collision. The accessibility cones are simplified to accessibility maps which can be implemented during motion of probes. Then, the algorithm forms a set of probe angles for required inspection and a heuristic method is used for an optimal angle search. Finally, probe path is generated and the module performs interference tests between the probe and part surfaces. If a collision is detected, module matches the collision conditions and heuristically modifies the path until a collision free path is generated. The motion of probe is continuously monitored to match with any preset collision condition.

This algorithm generates a probe path which guarantees a collision free trajectory of the probe, but the generated probe path does not necessarily be optimum.

Along the same line, the inspection path planner (IPAP) system is developed by Gu and Chan[2] to generate a collision free path. The IPAP uses output from inspection process planning module such as measurement points, part and probe orientations, inspection sequence, etc. The IPAP system consists of two modules, local path planning which gives path to be followed on the same feature and global path planning which gives movement of probe from one feature to another.

The overall approach is mainly based on interference calculations in which the motion of the probe is considered as a planer parallelogram and the object is considered as a B-Rep solid. The probe is abstracted as a straight line so that when it moves from one point to the other, the motion covers a plane surface in 3-D. The part is represented by a number of boundary surfaces. If the plane surface of moving line is intersected by edge of surface, it marks collision of probe with the part. If interference is found, then another set of offset points and measurement points is generated to replace the points that cause interference. Again, this approach generates a collision free path which does not necessarily be optimum.

MOST CAIP systems are concerned with automatic planning for dimensional and geometric inspection at either high level or low level. The high level is concerned with producing a collection of setups. Each setup is related to the accessibility of features to be inspected, the probes to inspect each feature and relative orientations of parts. From this, measurement sequence is generated. The low level planning deals with the issue of point selection, path generation and generation of executable code.

In the past, most of the efforts taken toward CMM probe path generation have emphasized on that of collision free paths, where major concerns are the recognition of geometric entities and probe interference. However, the optimization of the path between every pair of measurement

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

points should be considered so as to improve the throughput and quality of an inspection task.

The reviewed literature shows the demand for research in developing a generic algorithm which will provide an optimum path for collision free movements of the CMM probe. Aimed at this, an effort is made towards developing an effective CMM probe path planning system which will use an algorithm providing optimal probe paths over the related past work.

3. THE PROPOSED PATH GENERATION ALGORITHM

The development of an automated CMM probe path generation system is described in this section. A new algorithm for optimal and collision free path generation within the overall system is proposed considering the requirements, assumptions and descriptions.

3.1 Path generation system

The automated probe path generation system requires inspection planning output data such as measurement sequence, number and location of inspection points, probe and part orientation. In addition, it needs CAD modeling database which provides geometrical and topological information of the part modeled in any suitable solid modeling system. The database based on B-Rep solid modeler is used in this work due to its surface representation thoroughness.

The path planning module generates optimum and collision free path between two successive measurement points based on a particular probe and part orientation. During the process, probe is always checked for collision with respect to obstacles. The points lying on the detoured probe path are determined by the developed optimality conditions. Once an optimal solution is reached, generated optimal and collision free path is forwarded to the graphical simulation module where user can analyze and modify path for corrections, if necessary. Otherwise, probe path output is sent to the CMM commands (e.g. DMIS) generation module. The conceptual idea of the proposed system is shown in Fig. 1.

The interface module is responsible for getting three types of interactions, namely, the interactions with (a) the inspection process planner, (b) the CAD database of modeling space, and (c) the user. The system structure has two loops, one is used for getting the actual optimum and collision free path and the other involves user analysis. The first loop is nothing but a generic algorithm which should find a path in the modeling space under consideration. The second loop is required to make use of user knowledge.

3.2 Algorithm Concepts

According to the objective of the optimal and collision free probe path generation, an algorithm using B-Rep CAD database and modified ray tracing technique is proposed. The reason for using B-Rep CAD models is that most of today's commercially available solid modelers are based on Boundary representation (B-Rep) scheme for their geometrical database construction.[9] So the implementation scheme for any algorithm using B-Rep can be exchanged and accessed across various solid modeling platforms. Also, modified ray tracing method has been proven to be suitable for graphical applications,[10] which also serves its purpose to a great extent in this particular application area.

The ray tracing method establishes a parametric straight line between the start point and target point. Along this path, the algorithm detects any intersections of the ray with obstacles, such as fixtures or the part itself. If any intersection is found, then it detours the probe path so as to avoid that obstacle in an optimal manner. Afterwards, the path is finalized which consists of the start point and target

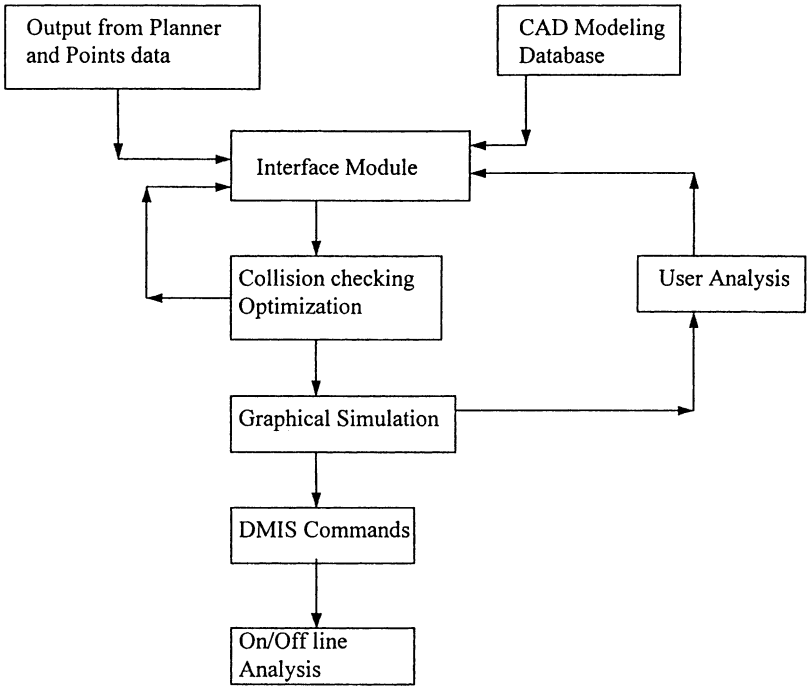


Fig. 1. CMM probe path generation system.

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

point as two end points and intermediate points as the detours for the probe movement.

This algorithm is expected to generate automatically an optimum and collision free CMM probe path which will be efficient, dynamic and on-line. An initial development stage, polyhedral and prismatic objects are considered for application. It requires that the end user has a CAD database system which can provide solid modeling data of the objects and fixtures in a proper format. The overall picture of the CMM probe planning problem can be described as follows:

(i) Generate an algorithm to find an optimum and collision free CMM probe path on parts in three dimensional workspace.
(ii) Use modified ray tracing method with B-Rep solids.
(iii) Implement the algorithm using ARX development environment supported by Mechanical Desktop and C++ as the development language.
(iv) Prepare test suits which comprise all feasible cases for implementing the algorithm.
(v) Check feasibility of the algorithm for extending it to sculptured surfaces.

3.3 Assumptions and requirements

The major assumptions made during the development work and requirements for the working of algorithm are as follows:

(i) The workspace under consideration is a simple three dimensional Euclidean (E^3) space.
(ii) The CMM bed is assumed to be a simple horizontal plane on which the parts to be inspected are placed.
(iii) The CMM probe is considered to be a small spherical object. This simplifies the interference detection problem to collision with a part surface.
(iv) The objects (part or fixture) are of simple prismatic polyhedral shapes. For non-polyhedral objects, the enclosing polyhedral surface must be approximated.
(v) The start point and measurement destination points of the probe are known a priori from the inspection planning module.
(vi) The simultaneous motion of the probe is possible in all three coordinate axes.
(vii) Interactive CAD database with Boundary Representation (B-Rep) scheme is required for the algorithm to work.

3.4 Algorithm description

3.4.1 Nomenclature. The nomenclature used in the algorithm description is explained as follows, for better understanding.

Objects, are simple prismatic polyhedral parts which are placed in three dimensional space for CMM inspection.
Ray, is an imaginary line joining the two points under consideration in space.
Start Point, is the point where Ray is originated, could be in space or on the surface of an object.
Target Point, is the point of measurement, supposed to be on planer surface of any object where the probe should touch and record the coordinates.
Intermediate Points, are points of detour on path between Start Point and Target Point so that collision of probe with any object feature is avoided. Usually these points will be on the edges of objects.
Target Plane, is the planer surface of object which consists of the Target Point.
Target Edges, are the edges of the Target Plane.
Intersection Point, is the point of intersection of the Ray and planer surface of object to be inspected.
Intersection Plane, is the planer surface of object which considers Intersection Point.
Adjacent Planes, are planer surfaces of object which are adjacent to Intersection Plane.
Common Edge, is the edge common to the Intersection Plane and the Adjacent Plane to be traveled.

3.4.2 The Proposed Algorithm Details. The following describes in detail the execution steps of the proposed algorithm.

Step 1: Shoot a Ray R_0 from the Start Point to the Target Point.

The parametric equation of a line which starts at Start Point P_0 and ends at Target Point P_t can be expressed as

$$P = P_0 + (P_t - P_0) \times t \tag{1}$$

where $0 < t < 1$, and $P = P(x, y, z)$. The three component equations can be written as

$$x = x_0 + (x_t - x_0) \times t \tag{2}$$

$$y = y_0 + (y_t - y_0) \times t \tag{3}$$

$$z = z_0 + (z_t - z_0) \times t \tag{4}$$

Step 2: For the particular Ray, obtain Intersection Point and Intersection Plane with the object in the underlying CAD database. Information about all the planes of the objects is stored in a CAD library. Using that information, traverse over all the faces of the object and check for intersection with the parametric Ray. In case of a possible collision on this imaginary path, there would be an intersection between ray and one of the faces of the object. That intersection point will be returned. There could be more than one intersection point, then the nearest one to the Start Point should be considered as the Intersection Point. Also consider the particular plane consisting of the Intersection Point as the Intersection Plane. The intersection point between a line and a plane can be found as follows. The equation of a plane is given by,

$$a \times x + b \times y + c \times z + d = 0 \tag{5}$$

To find out the intersection point, we have four equations with four unknowns. Solving those equations gives

$$t = \frac{d - a \times x_0 - b \times y_0 - c \times z_0}{a \times (x_t - x_0) + b \times (y_t - y_0) + c \times (z_t - z_0)} \tag{6}$$

If the resulting value of the parameter "*t*" is within the range [0,1], that means the ray intersects with one of the faces of

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

the object under consideration and the coordinates of that intersection point can be obtained from the parametric equation of the ray by substituting the resulting value of "*t*".

Step 3: Traverse on all the adjacent faces of the Intersection Plane. For every adjacent face, find out the perpendicular distance between the Target Point and that Adjacent Plane. Consider that adjacent face amongst all adjacent faces which is at the shortest distance from Target Point as the Adjacent Plane. It is noted that there are a few particular exceptions like base plane, planes at same distance, etc. The CMM Probe has to travel over this face to minimize the overall travel path length en route to the Target Point. If the perpendicular distane is zero, it implies that the Target Point is lying in that Adjacent Plane.

The distance between any arbitrary point (x_1, y_1, z_1) and a plane can be calculated as follows. Since the equation of plane is

$$a \times x + b \times y + c \times z + d = 0$$

the perpendicular distance between the plane and this point can be written as,

$$p = \frac{a \times x_1 + b \times y_1 + c \times z_1 + d}{(a^2 + b^2 + c^2)^{/2}} \qquad (7)$$

Step 4: Get the Common Edge of the Adjacent plane found in the last step and the Intersection Plane. This can be done using the face and edge traversers provided in the ARX API. The reason for this step is that the optimum path on solid objects in three dimensional modeling space lies on the edges of the object.

These steps are graphically shown in Fig. 2. The solid object is a simple prismatic part. This figure illustrates a Ray from P_0 to Pt and resulting in the intersection plane, adjacent plane, and common edge.

Step 5: The next step is to find a detour point on the Common Edge which fulfils the following criterion. An Intermediate Point is found in such a manner that the overall length of path from Start Point to Target Point is optimum

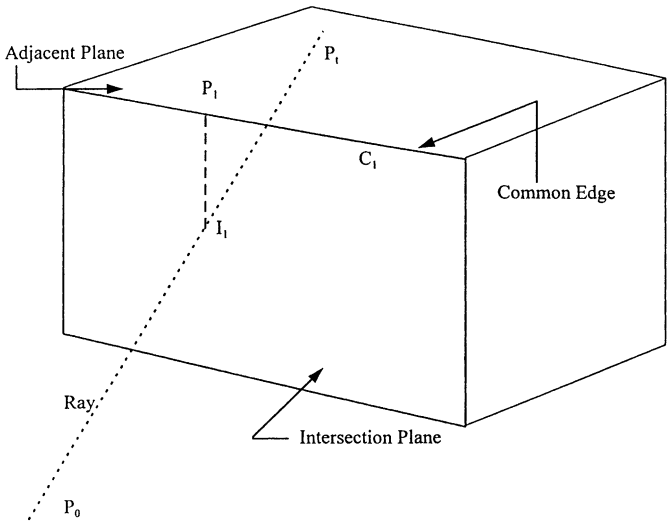


Fig. 2. An intersecting ray with object

and interference with the object is to be avoided. By detouring, the probe will travel to this Intermediate Point en route to Target Point. There could be more than one Intermediate Point depending on the position of the Target Point. The path between two Intermediate Points is always a straight line.

In this particular case the first Intermediate Point should lie somewhere on the Common Edge. To find the exact location of this point, this step is further justified in the next section by some mathematical treatments.

According to this technique, from Intersection Point I_1 move in a manner perpendicular to Common Edge C_1 till any edge is encountered. That Intersection Point I_1 with that edge is the Intermediate Point P_1. The probe can be directed from Start Point P_0 to the Intermediate Point P_1 without collision.

Step 6: From the Intermediate Point, establish one new Ray on that Adjacent Plane in the direction of Target Point P_t. This new Ray is shot in such a way that the orthographic projection of this Ray and that of the original Ray remain to be the same in one of the three views, viz. Top View or Side View or Front View.

Step 7: On the Adjacent Plane, a new Ray originating from the Intermediate Point will meet one of the following, generating three possible outcomes.

(a) Target Point P_t,
This situation will occur when the Adjacent Plane is the Target Plane itself. In that case, stop and finalize the path consisting of the Start Point, Intermediate Points and Target Point.

(b) An edge which is one of the Target Edges,
In some cases, the intersection point of the new Ray and that particular Target Edge is next Intermediate Point P_2. From that point, directly go to Target Point P_t, stop and finalize the path. As the Ray reaches one of the Target Edges, it can directly travel to Target Point without any further intersection.

(c) Any other edge,
Take dot product of this current ray R_1 and the outward normal of the other plane meeting at that edge. Check the sign of the dot product.

If it is positive, then the intersection point of the ray with that Target Edge is the next Intermediate Point. Repeat the process from steps 1 to 7 till the algorithm stops.

If it is negative, then establish the next Intermediate Point by dropping a perpendicular line from that intersection point across that plane till this perpendicular line meets the opposite edge. This intersection point with the opposite edge is the next Intermediate Point. From this, repeat steps 6 to 7 till the algorithm stops.

As a typical case to show the path generation using this algorithm, Fig. 3 is provided. The detoured probe path is $P_0 - P_1 - P_t$. P_0 is the Start Point and P_t is the target point for the probe. The collision between the probe and the part is avoided by forcing the probe to travel to point P_1. It should be noted that this point is on one of the object edges and the path in between two consecutive detour points is a straight

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

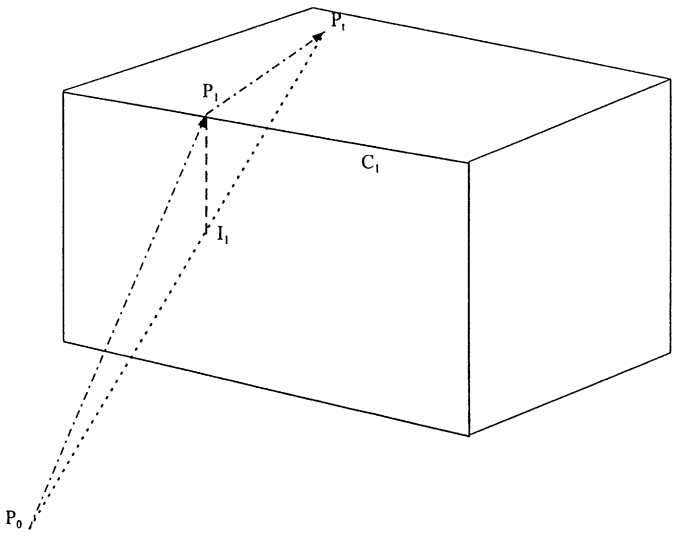


Fig. 3. Detour path using intermediate point

line. More cases are considered in the next section. Figure 4 illustrates the complete flow diagram of the algorithm.

3.5 Mathematical treatments

The actual probe path finding problem reduces to finding intermediate points "P_1, P_2, . . . , P_n" in space such that the probe is detoured to these points to avoid interference with the part from the start point to target point. It should be noted that these intermediate points for an optimum collision free path in an E^3 space lie on the edges of the object.[10] After step 4 of the algorithm, the Intersection Point and the Common Edge are known. The probe has to travel to such an Intermediate Point on the Common Edge that the distance traveled is a minimum. Hence, the Intermediate Point which is expected to be nearest to the Intersection Point, must lie on the Common Edge which is part of the Adjacent Plane boundary.

For the shortest distance between a fixed point A and a line EF, the foot of the perpendicular B from point A onto the line EF is unique. Also, the distance between the point and foot of perpendicular B is the shortest. So, this foot of the perpendicular A which is on the line EF becomes the Intermediate Point for the probe path. Fig. 4 is given to elaborate the concept.

The same fact can be looked upon as the Intermediate Point should be the nearest point of the Adjacent plane to the Intersection Point. Referring to Figure 5, A is an arbitrary point in space not on the plane, CD is the normal to the plane. It is proven that the point B on this plane should be such that the vector AB is collinear with CD and hence, orthogonal to the plane.[11]

Using the above two geometrical theorems, the Intermediate Points for the probe travel path can be obtained which always lie on the edges of the object. Figure 5 explains the overall concept about the optimal and collision free path generating procedure.

The concept of determining the shortest path gives rise to an idea that when the orthographic projections of the new path and the original ray are considered, they are the same in the top view. This idea could be extrapolated to find a new path whose orthographic projection is the same as that of the original ray. Depending on the orientation of the Common Edge, that particular orthographic view projection is to be considered for determining an optimal path.

This can be further extended to construct an imaginary section plane through the underlying object which is

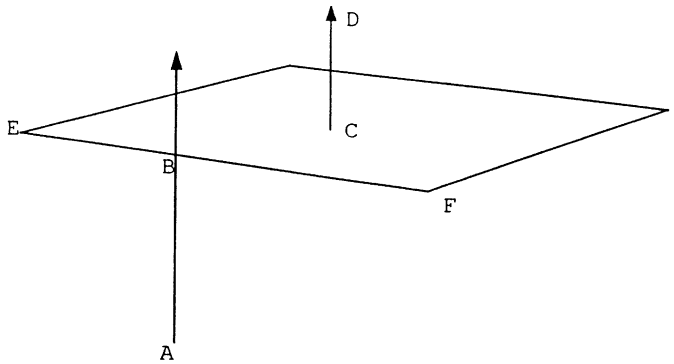


Fig. 4. Nearest point of a plane and an edge

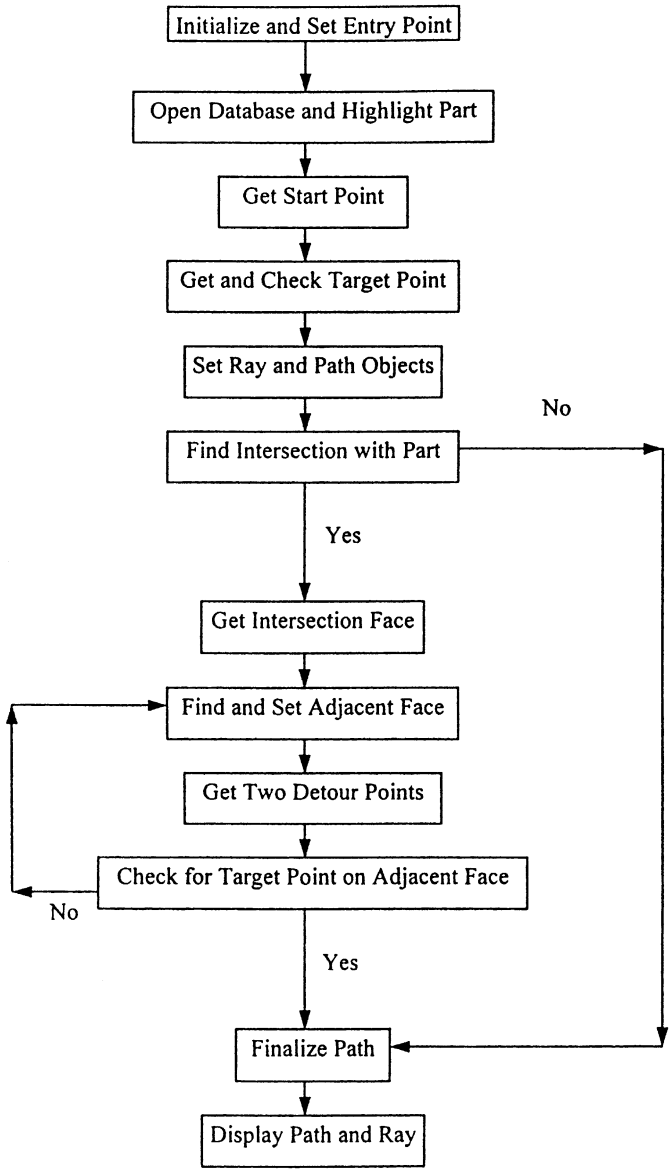


Fig. 5. The complete flow diagram of the algorithm

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

perpendicular to one of the three coordinate planes. It also contains the original intersecting ray. The contour of the object profile projected on this sectional plane can be used to determine the optimal path.

The optimal path should lie on the vertices of the resulting two dimensional profile on the sectional plane. Using this approach, the optimal and collision free path determining skill can be further implemented.

The above two approaches can also be used to solve the original path determining problem. The advantages, disadvantages and typical applications using these different approaches can be discussed. Based on those results, the better path generation system architecture can be built.

4. IMPLEMENTATION AND RESULTS

The work done on the implementation of the optimum and collision free path generation algorithm for a CMM probe path generation system is described in this section. A prototype system was developed using ARX application programming interface (API) on Mechanical Desktop V2. As described in the last section, the system needs a CAD database which can provide geometric information about B-Rep solid modeling database. AutoCAD graphic interface is also used to visualize the CMM probe path over the part in three dimensions.

4.1 Implementation

For the implementation of the above algorithm, the AutoCAD Runtime Extension (ARX) development environment supported by Mechanical Desktop V2 is being used. This environment is highly sophisticated, uses ACIS solid modeling API, which is developed by a third party solid modeler. The ARX environment is based on C++ programming language, CAD database libraries supported by AutoCAD and some of the ADS programming interface. Figure 6 shows the overall schematic diagram of the ARX structure.

An ARX application is a dynamic link library (DLL) that shares AutoCAD's address space and makes direct function calls to AutoCAD. The ARX libraries include macros to facilitate defining new classes and offer the ability to add functionality to existing classes in the library at run time. The ARX programming environment provides an object-oriented C++ application programming interface that enables developers to use, customize, and extend AutoCAD graphics capability through an open architecture, providing direct access to AutoCAD database structures, the graphics system, and native command definition. Therefore, the overall system is very flexible.[12]

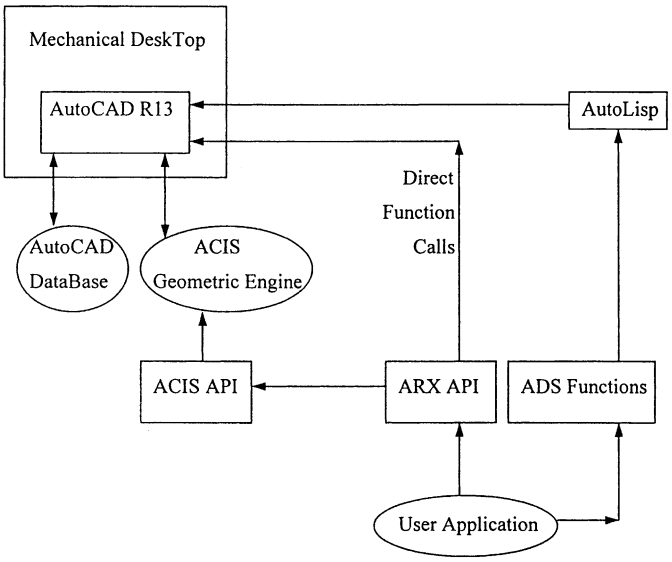


Fig. 6. ARX environment structure

The solid modeling scheme used in our ARX experiment is Radial Edge Data Structure, which is a non-manifold 3-D Boundary Representation (B-Rep) of solid objects. This explicit and evaluated form of representation allows the user to travel efficiently all over the basic underlying object entities, i.e. vertices, edges and faces. As the algorithm needs frequent traversals over part's surfaces, so it searches over geometrical and topological entity data of the object. The ARX library is the appropriate option for implementation using this scheme. It also provides built-in functions for mathematical manipulations such as vector manipulations which are most required for the proposed algorithm to work efficiently. The AcBr library of ARX has specific classes which not only holds the geometric data of the object by entities but also provide methods to traverse over the faces, loops, edges and vertices.

4.2 Prototype system

Following the above guidelines, the proposed algorithm is implemented which generates optimum and collision free CMM probe paths upon receiving the measurement points information from the CMM inspection process planning module. A prototype system is built which can run as any standard application in Mechanical Desktop V2. This application helps to check the feasibility of the proposed algorithm on the required type of parts. The functioning of the application can be described by the following steps. Figure 7 is provided to illustrate the application structure.

1. Open the particular three dimensional drawing of the underlying part in Mechanical Desktop V2.
2. Load the ARX "inspect.arx" application in AutoCAD R13.
3. To start the application, use command "inspectr".
4. Select the object to be inspected.
5. Input the CMM probe start point, anywhere in space.
6. Input the target point for the probe measurement point.
7. The selected object is highlighted.
8. Using the proposed algorithm, the optimum and collision free path is determined between the start point and target point.
9. The generated probe path and the original ray are displayed on screen.
10. A list of detour points on the probe path are displaced.
11. Repeat the steps from 5 to 10 for a new set of probe start and target points.

According to this prototype system interface, first the user has to enter the start and destination points information to start and to use the algorithm. Using this information about points, the application determines intermediate points which

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

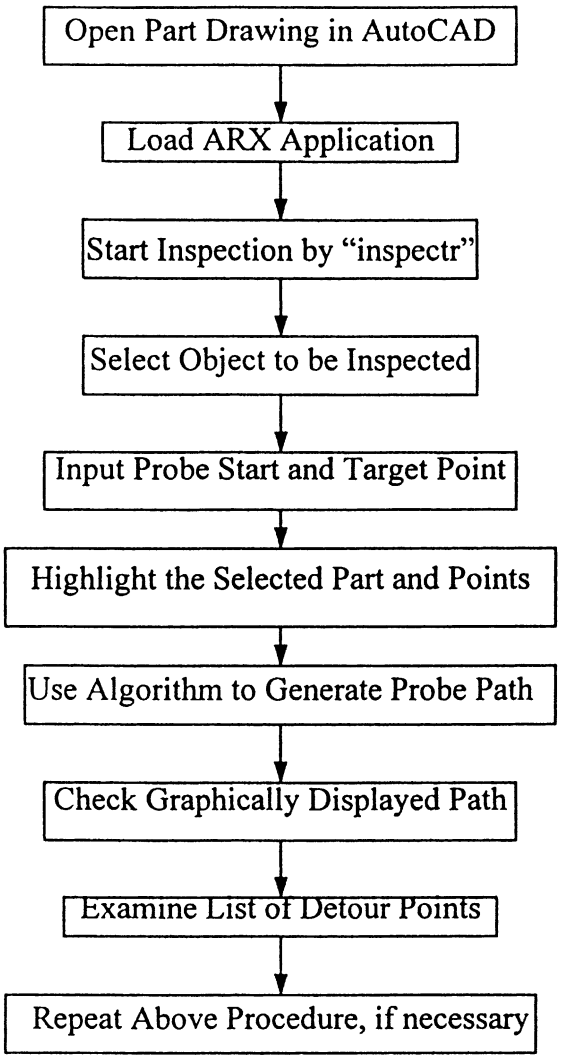


Fig. 7. Prototype system

lie on the required path. The CAD database is opened and object information is obtained by querying the database using API provided. This geometrical and topological data is processed to achieve the required results. Then, the generated path is displayed on the screen using AutoCAD drawing facility.

4.3 Test case suites

Using the above described application in AutoCAD R13, the proposed algorithm was tested for four different cases. These four objects models were selected so as to represent some real world solid parts. All these objects are prismatic models. For every test suite, the three dimensional solid models of these objects were drawn in AutoCAD R13. Specific destination points were selected on the different surfaces of the object to consider probable measurement points in the actual practice. For each object, such four target points are considered. For every pair of target point and start point, a list of intermediate points is generated.

4.4 Results of the tests

The results after running these four test suites are given in Tables I through IV. Figures 8 through 11 also show the

Table I. Results for test suite #1.

No.	Start Point	Target Point	Intermediate	Points
1	0,0,0	4,5,3		
2	0,0,0	7,7,3	4,7,2.36	
3	0,0,0	9,5,4	4,2.87,5	9,5,5

Table II. Results for test suite #2.

No.	Start Point	Target Point	Intermediate	Points
1	0,0,0	8,10,4	5,10,3.41	
2	0,0,0	15,7,3	15,5,2.85	
3	0,0,0	14,15,8	5,8.65,10	14,15,10

Table III. Results for test suite #3.

No.	Start Point	Target Point	Intermediate	Points
1	0,0,0	10,14,3	5,15,2.64	10,15,3
2	0,0,0	11,10,3.5	11,10,5	
3	0,0,0	19,15,4	7.49,5,5	19,15,5

Table IV. Results for test suite #4.

No.	Start Point	Target Point	Intermediate	Points
1	5,7,2	15,7,5	10,5,3.5	15,5,5
2	25,5,5	5,17,9	15,11.35,10	5,17,10
3	15,25,15	12,5,4	12.38,5,10	

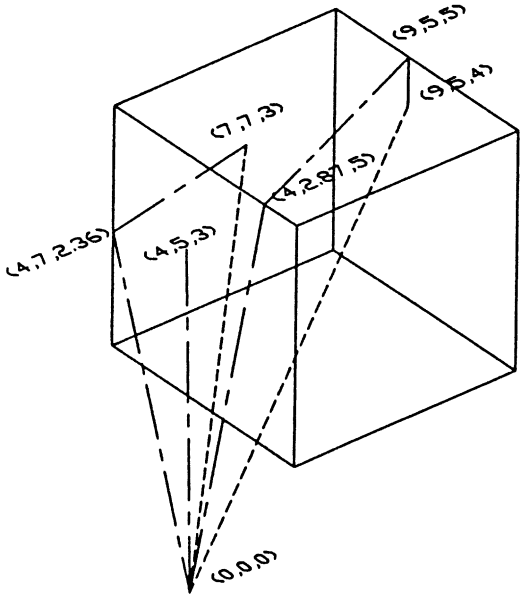


Fig. 8. Test suite #1

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

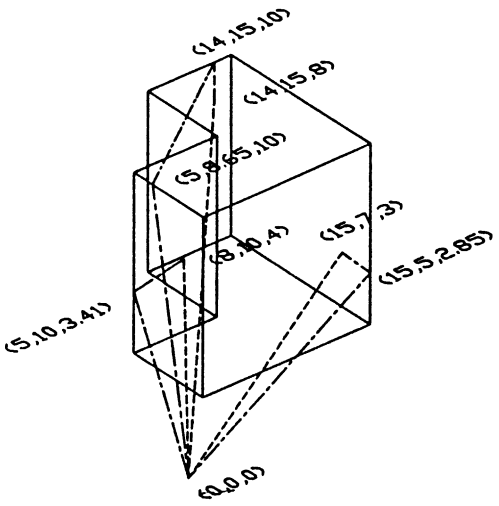


Fig. 9. Test suite #2

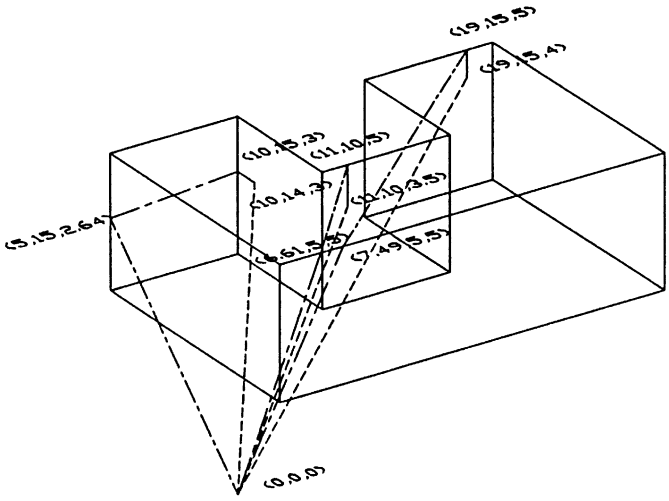


Fig. 10. Test suite #3

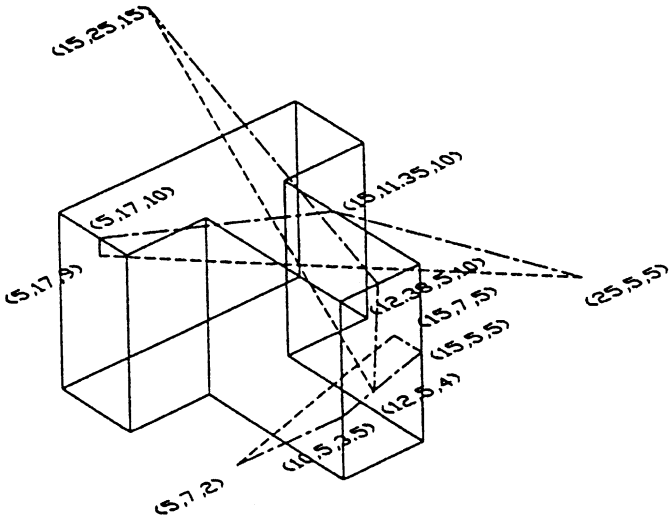


Fig. 11. Test suite #4

initial ray and final probe path for each test suite. The only constraint on selection of the target points is that they must be on one of the surfaces of the object. This constraint is required because a CMM probe always touches the measurement point which has to be on the surface. The other constraint on selection is that the target and start points should not be inside the object volume, because the CMM probe cannot penetrate the object. The algorithm checks these two prime conditions and prompts the user to modify a particular incorrect point selection.

The first object which is a cube, represents very basic element of any solid part. Five measurement points are selected so that they are on five faces of the cube. The results are shown in Fig. 8. Table I gives the list of intermediate detour points.

The second object is a step. Table II gives the list of results after running the application. Figure 9 shows the pictorial view.

The third object consists of a through slot. The target points are selected which are inside as well as outside the slot. Figure 10 and Table III describe more about this test suite.

The fourth object has a gusset or projection over itself. The selected target points are not only on the projection but also on the remaining portion of the object. The accompanying Figure 11 and Table IV throw more light on the results of this test suite.

It can be noted that as expected all intermediate points are located on the edges of the object. The final path is just connecting each consecutive pair intermediate points, starting at the start point and ending at the target point. In Figures 8 through 11, the original ray is shown in dashed line and the path generated is shown in center line.

4.5 programming architecture

The above described application is developed following Object Oriented Design (OOD) concept using *C++* as computer programming language.[3] The compiler used is Microsoft's Visual *C++* 4.0 running on a Windows 4.0 workstation. The whole computer program is divided into fifteen files comprising of some header files, some definition files, and remaining source files. All these fifteen files are needed to build the application executable file "inspectr.arx".

Figure 12 describes the classes used in the above application, viz. "Ray" and "Path" with their own functions and the global functions used. Instances of Ray and Path classes are created and these functions are used to manipulate the boundary representation database supported by ACIS and AutoCAD. The global functions and definitions of these classes are written generic, so that this source code can be further extended for future developments. Hence, this coding complies to the Object Oriented Programming standards. Next, Figure 13 shows the file structure used in this application. The "path.cpp" and "ray.cpp" files are class definition files for "Path" and "Ray" classes, respectively. Other "*.cpp" files are the source code files consisting of global functions. The declarations of "Ray" and "Path" classes are written in two files "ray.hpp" and "path.hpp", respectively. Remaining "*.hpp files" are

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

the header files for source files. The "*.lib" files are the ARX development library files necessary for the application to be executed. All these files are stored in a project workspace named "inspect" which on compilation and linking produces an ARX dynamic linked library application.

Two new classes are created with their declaration and definition files so as to represent the tracing ray and the probe path. Any number of instances of these classes can be created and used for manipulation. These classes hold

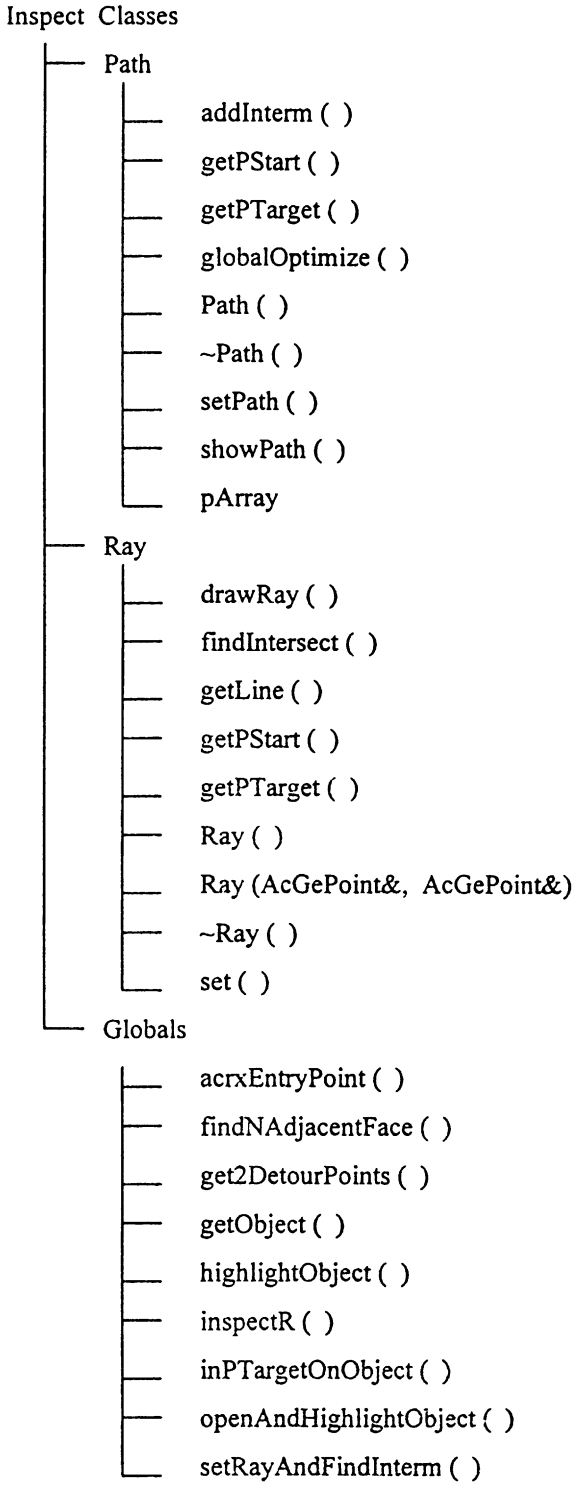


Fig. 12. Inspect classes and global functions

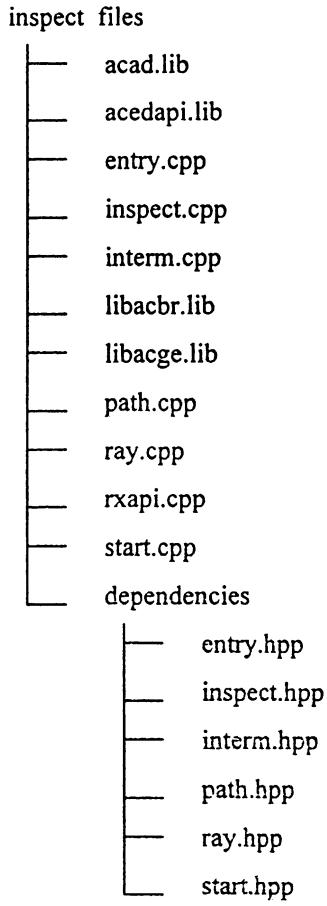


Fig. 13. Inspect files and dependencies

particular data for an instance such as the start and end points of a ray, intermediate points on the path, etc. Necessary functions are provided which interact with the class data and with the object information so as to produce desired results. One typical example of such function is, for a given ray find intersection with the object and get the intersection point with intersection face of the object. For the algorithm to perform various mathematical and geometrical tasks, the application programming interface (API) supported by ARX is used. This API helps in traversing over object geometry such as from one object entity to the other, finding distances between a line and a point, etc. Error checking is done at every possible place so that probability of crashing the program is minimized. In case of an error, the user is prompted with probable cause of error thus helping him to track the cause of such an error. The overall application is user friendly, provided that a user has decent background in AutoCAD R13.

5. CONCLUSIONS

Aimed at improving state-of-the-art metrology technology, an automated inspection path planning system has been designed. A generic algorithm was developed which should be incorporated in such an automated path planning system. The solid objects to which the proposed algorithm is applied and implemented are considered to be prismatic B-Rep polyhedral component models. An application has been developed to check the feasibility of implementing this

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press

algorithm for generating an optimum and collision free CMM probe path. This user friendly application runs on AutoCAd R13 and determines required path between user specified measurement points.

The interactive CAD environment provided by AutoCAD Runtime Extension (ARX) based on ACIS geometric engine was extensively used in implementing the algorithm based on object oriented design techniques. The major contribution focused on automating the inspection process for coordinate measuring machine functioning for both on line and off line processing. Although many challenges remain to be solved, the proposed work suggests a new way of automated CMM probe path generations. The algorithm also assures that the generated probe path is optimum and collision free. Optimality was considered as the key thrust area while this work was undertaken. The developed generic algorithm gives a simpler method for probe path generation. In addition, the generated path is smoother and thus involves less acceleration and deceleration of the probe mechanism. This improves the quality of measurement and the life of CMMs.

Moreover, the algorithm uses basic concepts of geometry and topology, making it more understandable and usable. It is generic in nature in a way that it is independent of the CAD system and platform used. Another important feature of this algorithm is that it can be extended for further improvements for dimensional inspection of parts having sculpted surfaces. As the algorithm can be applied to work in conjunction with any CAIP system, the performance of existing CAIP system can be improved using the generic capabilities of this algorithm.

Acknowledgement

This work was supported in part by OSGC Grant #02801 and SEM Grant #596–2250. They were gratefully acknowledged. The authors would also like to acknowledge the technical consultation provided by the research staff of the Intelligent Systems Division at The National Institute of Standards and Technology during the course of this work.

References

1. H.A. ElMaraghy and W.H. ElMaraghy, “Computer Aided Inspection Planning”, *Advances in Feature Based Manufacturing*, edited by J. Shah, M. Mantyla and D.S. Nau (John Wiley and Sons, New York, N.Y., 1994) pp. 363–396.
2. P. Gu and K. Chan, “Generative Inspection Probes and Probe Path Planning for CMM”, *Journal of Manufacturing Systems* **15**(4), 240–255 (1996).
3. C.H. Menq, H. Yau and C. Wong, “An Intelligent Planning Environment for Automated Dimensional Inspection Using Coordinate Measuring Machines”, *Journal of Engineering for Industry* **114**, 222–230 (May, 1992).
4. U. Roy, Y. Xu and L. Wang, “Development of an Intelligent Inspection Planning System in an Object Oriented Programming Environment”, *Computer Integrated Manufacturing Systems* **7**(4), 240–246 (1994).
5. S. Kim and S. Chang, “The development of the Off-line Measuring Planning System for Inspection Automation”, *Computers and Industrial Engineering* **30**(3), 531–542 (1996).
6. C.W. Brown and D.A. Gyorog, “Generative Inspection Process Planner for Integrated Production”, *1990 ASME Winter Annual Meeting, Symposium on Advances in Integrated Product Design and Manufacturing* (1990) pp. 151–162.
7. E. Lu, J. Ni and S.M. Wu, “An Algorithm for Generation of an Optimum CMM Inspection Path”, *Journal of Engineering for Industry,* **116**, 396–404 (September, 1994).
8. C.P. Lim and C.H. Menq, “CMM Feature Accessibility and Path Generation”, *International Journal of Production Research* **32**(3), 597–618 (1994).
9. Y.J. Lin and P. Murugappan, “A New Algorithm for CAD-Directed CMM Dimensional Inspection”, *Proceedings of the 1998 IEEE ICRA*, Leuven, Belgium (May, 1998) pp. 893–898.
10. A.S. Glassner, *An Introduction to Ray Tracing* (Academic Press, 1989).
11. W. Boehm and H. Prautzsch, *Geometric Concepts for Geometric Design* (A.K. Peters, Ltd., 1994).
12. “ARX Reference Guide”, *Autodesk Inc.* (San Rafal, CA., 1996).

https://doi.org/10.1017/S0263574700003076 Published online by Cambridge University Press