A 3D KNOWLEDGE-BASED ROUTER FOR WIRING IN AEROSPACE VEHICLES

Christian Van der Velden*, Cees Bil*, Xinghuo Yu and Adrian Smith*****
***School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia**
****School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia**
*****GKN Aerospace Engineering Services Pty Ltd, Brisbane, Australia**

The Sir Lawrence Wackett Centre for Aerospace Design Technology
GPO Box 2476V
Melbourne, Victoria 3001, Australia
Ph: +61 (0) 396 454 536
Fax: +61 (0) 396 454 534
Email: s2108846@student.rmit.edu.au

Keywords: Knowledge Based Engineering (KBE), intelligent algorithms, routing, Computer Aided Design (CAD), Finite Element Modelling (FEM), voxels

Abstract

The design of electrical wiring and hydraulic/pneumatic piping routes in aircraft is a long and repetitive process which is largely done by hand. Presented in this paper is progress in the development of a knowledge based routing system which automates much of the routing process using intelligent algorithms and interchangeable design rules. The system reads three dimensional CAD models to be navigated and source and target locations, and outputs wire/pipe geometry and other information required to describe the system path.

1 Introduction

The management and utilisation of data is of great importance within the engineering field. Knowledge Based Engineering (KBE) is a set of methodologies for enhancing engineering design processes through effective knowledge management. KBE systems are software applications which collect, store and utilise engineering rules and knowledge and are used for design automation, verification, and integration of design and production knowledge. Use of these practices facilitates a smooth transition between product lifecycle phases.

Electrical wiring looms in aircraft typically consist of thousands of cables and are usually routed by hand using Computer Aided Design (CAD) workstations with engineers using personal knowledge and experience of how to route cables through the structure. There are numerous regulatory and functional design rules which must be satisfied (such as bend radii, electromagnetic sensitivities, placement of support brackets, protection against corrosion and abrasion, cable bundling, intersections between cables, divergence of cable bundles, etc.). The routing process is highly repetitive and design outputs can vary significantly between engineers. Electrical wiring design often proceeds in parallel with principle structural design. The iterative nature of the total design process is such that structural changes are prone to occur requiring time consuming rework for any electrical cabling affected. In a similar way, hydraulic and pneumatic pipes in aircraft are manually routed and are governed by different set of design rules. The repetitive, rule-governed nature of the routing process makes it a prime candidate for application to a knowledge based system.

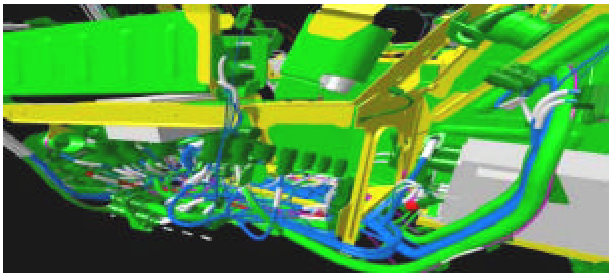


Fig. 1. Example of routed electrical loom

This paper describes current progress in the development of a knowledge based system for three-dimensional routing in aerospace vehicles with applications including electrical wiring and hydraulic/pneumatic piping design. The concept of the software is to read 3D CAD geometry with source and target terminals and output wire/pipe geometry and other information required to describe the system path. Section two of the paper discusses the general routing problem and the approach used for solving cases. The third section gives a description of the path finding algorithm used. Section four discusses input geometry issues including use of finite element and volume graphics for representing CAD models. Following this, a description of the process flow of the knowledge based system and concluding remarks are given.

2 Routing Problem

The routing problem is commonly encountered in numerous fields ranging from electronics, data flow in computer networks, navigation systems, and Artificial Intelligence (AI). Examples include design of Printed Circuit Boards (PCBs), Very Large Scale Ingetrated (VLSI) circuits, Global Positioning System (GPS) navigation systems, computer game and robot AI.

As mentioned previously, design of electrical wiring systems for aircraft is a complex tasks with hundreds of rules and best practices which must be adhered. Currently there is no dedicated section of the Federal Airworthiness Requirements for transport category aircraft (FAR-25) dedicated to wiring design practices. Instead, a number of sections touch on the subject including 25.1301/1309, 25.1529, 25.1353, 25.869, AC 43.13-1b, AC 25-16, AC 25-10, and policy memos [12]. Engineers must sift though a large amount of data to single out the rules applicable to wiring and monitor any updates to these rules made by governing bodies. For military aircraft, a different set of rules apply and are contained in MIL-W-5088L: Military Specification– Wiring, Aerospace Vehicle, last updated in 1991.

The Knowledge Based System (KBS) under development aims to collect these rules and implement them in a path finding algorithm which will return valid paths. For the system to have maximum flexibility, it will need to incorporate methods of updating existing rules and adding new rules as necessary, with minimal effort. The following, obtained from civil and military airworthiness requirements (FAR-25) and (MIL-W-5088L), lists just some of the areas of consideration when designing electrical wiring systems [10] and [12]:

- Electrical Loads
- Breaker/Wire Sizing
- Wire Routing
- Clamping
- Tie-wraps
- Bend radii
- Splicing
- Wire terminations
- Grounding & Bonding
- Wire Marking
- Connectors
- Conduits
- Wire Insulation
- Wire Separation
- Chafing
- Unused wiring
- Riding on structure
- Riding on other wires
- Passing through lightening holes
- Slack
- Corrosion / contamination

3 Routing Process

Practices in VLSI routing automation provide a good starting point for addressing the problem of electrical loom and pipe design in aerospace vehicles. Computer microprocessors consist of many million logic components which must be interconnected within a very small space using very fine wires. Numerous algorithms exist for connection of terminals while maximizing performance against several metrics. These include: area taken by the routed nets, routing completion rate (percentage of nets successfully routed), length of interconnection wires, number of vias used, and time taken to execute the routing run. Similar principles can be applied to aerospace applications by extending the domain of the problem from 2D to 3D and introducing additional performance criteria stemming from rules discussed above.

The VLSI routing problem is considered NP-complete commonly requiring the use of powerful heuristics which can generally find near optimal solutions. In general once the physical component placement for a system has been defined and the routing requirements given (usually in the form of a netlist), the routing process consists of four main steps [2]:

- Definition of regions to be routed: allows the problem to be split into a number of smaller routing problems.
- Global routing: planning phase which reads the position of terminals and fixed objects, and assesses and prioritises nets to be routed. Instructions for the detailed router are made such that completion rate is maximised and total path length is minimised, especially for critical nets, to reduce delay.
- Order in which regions are routed: determines ordering of globally routed regions such that congestion will be avoided.
- Detailed routing: determines the exact path taken by wires including layers used for each segment and vias connecting between layers.

There are numerous algorithms for a variety of routing applications in different stages of the routing process as well as for specialised applications. Fig. 2, modified from [14], shows a family of routing algorithm types and their use in different phases in the routing process. The list is by no means exhaustive, but covers several of the common routers: which are maze, channel/switchbox, and line search/probe algorithms.

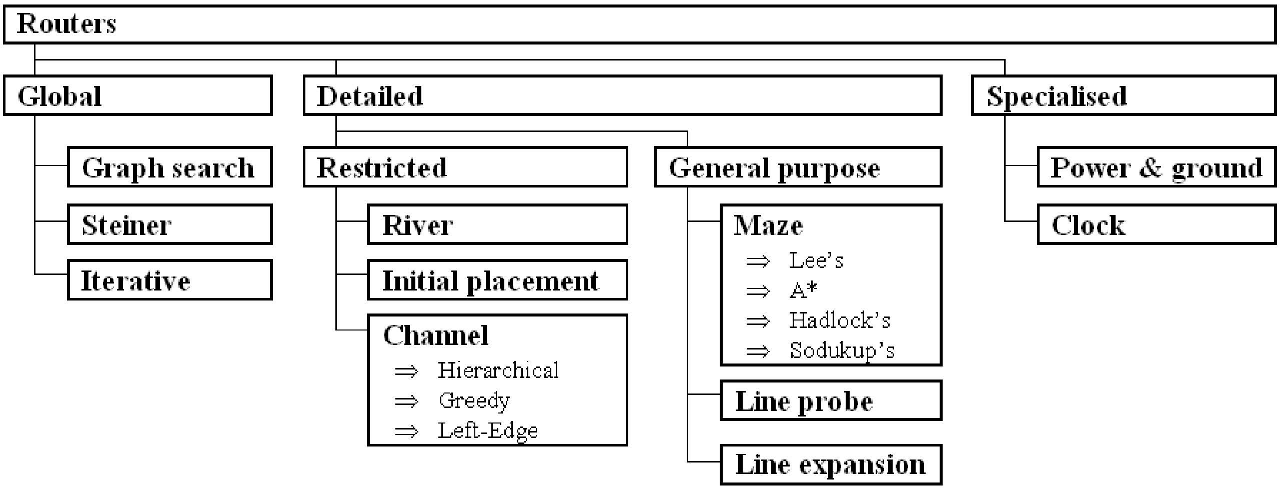
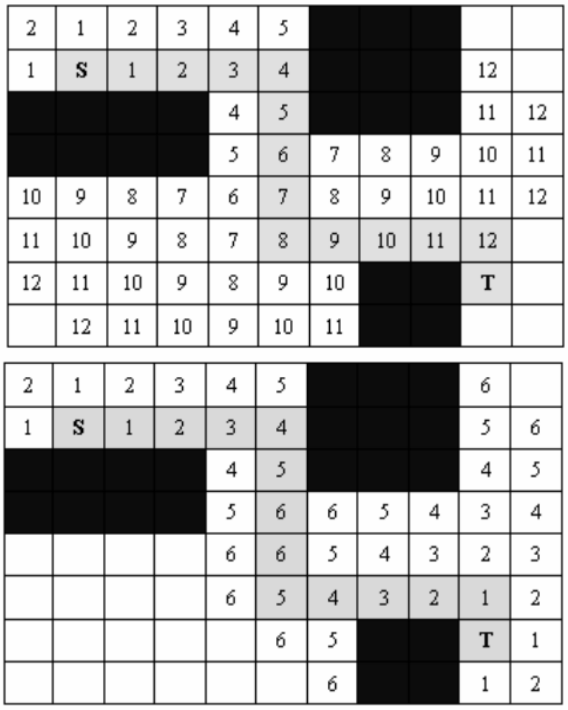


Fig. 2. Family of routing algorithms. Taken from [14].

4 Routing Algorithm

In its basic form, the algorithm used by the system to determine the path is based on the classic breadth-first, grid-based maze algorithm or Lee's algorithm [6]. Lee's maze router always returns the shortest path for a single net in a given search space with obstacles. The algorithm functions by propagating a wave from the source and/or target terminals over a grided search area and assigns a value to each node depending on its distance from the source or target. A backtracking phase then determines the shortest path between the two terminals (see Fig. 3).



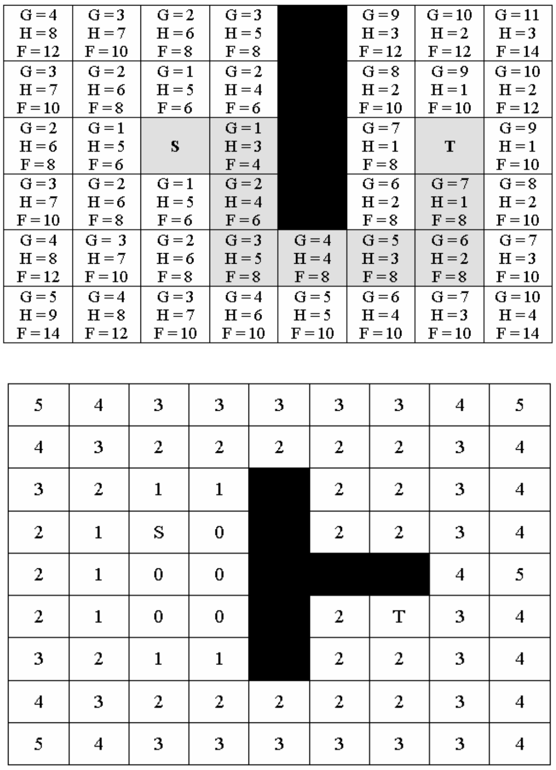
2	1	2	3	4	5					
1	S	1	2	3	4				12	
				4	5				11	12
				5	6	7	8	9	10	11
10	9	8	7	6	7	8	9	10	11	12
11	10	9	8	7	8	9	10	11	12	
12	11	10	9	8	9	10			T	
	12	11	10	9	10	11				

2	1	2	3	4	5				6	
1	S	1	2	3	4				5	6
				4	5				4	5
				5	6	6	5	4	3	4
				6	6	5	4	3	2	3
				6	5	4	3	2	1	2
					6	5			T	1
						6			1	2

Fig. 3. Example of maze routing. (Top) wave propagation from source only. (Bottom) wave propagation from source and target

The basic form of the maze algorithm has numerous limitations of which the main problem is sensitivity to the order in which nets are routed. Paths from routed nets form obstacles for subsequent nets and in some cases can prevent nets from being completed. In cases where this is encountered, a rip-up-and-reroute procedure can be used which removes routed nets and retries in a different order. In addition to this, the algorithm is inefficient when routing more than two terminals in a single net. In the case of multi-terminal nets, the connection between two terminals is found, then the partially routed net is treated as the source for remaining terminals. Also, the algorithm is inefficient when routing terminals in large empty spaces.

To address shortcomings of the basic maze algorithm, a large number of alternative path finders have been developed based on Lee's basic maze approach and used in numerous applications. One example is the A* (A star) algorithm which is used extensively in computer game navigation and uses a "best-first" search technique to determine path steps (Fig. 4 left) [8]. A score is given for each node based on minimum distance to target and distance from source, and the nodes with lower scores favoured. Another example is Hadlock's algorithm which uses a "greedy" search technique and adds penalties for every deviation away from the target (Fig. 4 right) [14].



G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 3 H = 5 F = 8		G = 9 H = 3 F = 12	G = 10 H = 2 F = 12	G = 11 H = 3 F = 14
G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	G = 2 H = 4 F = 6		G = 8 H = 2 F = 10	G = 9 H = 1 F = 10	G = 10 H = 2 F = 12
G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	S	G = 1 H = 3 F = 4		G = 7 H = 1 F = 8	T	G = 9 H = 1 F = 10
G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	G = 2 H = 4 F = 6		G = 6 H = 2 F = 8	G = 7 H = 1 F = 8	G = 8 H = 2 F = 10
G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 3 H = 5 F = 8	G = 4 H = 4 F = 8	G = 5 H = 3 F = 8	G = 6 H = 2 F = 8	G = 7 H = 3 F = 10
G = 5 H = 9 F = 14	G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 4 H = 6 F = 10	G = 5 H = 5 F = 10	G = 6 H = 4 F = 10	G = 7 H = 3 F = 10	G = 10 H = 4 F = 14

5	4	3	3	3	3	3	4	5
4	3	2	2	2	2	2	3	4
3	2	1	1		2	2	3	4
2	1	S	0		2	2	3	4
2	1	0	0				4	5
2	1	0	0		2	T	3	4
3	2	1	1		2	2	3	4
4	3	2	2	2	2	2	3	4
5	4	3	3	3	3	3	3	4

Fig. 4. Extensions to maze algorithm. (Top) A* algorithm. [8] (Bottom) Hadlock's algorithm [14]

The algorithm under development uses object-based programming principles and implements intelligent search techniques such as best-first and greedy searches. As far as possible, the algorithm uses intuitive, plain English terminology in its implementation of routing rules. The algorithm will access a knowledge base of electrical design rules collected from regulatory documents and best practices. This knowledge base will be interchangeable allowing rule sets for different routing applications to be plugged in. The algorithm used by the KBS can be applied to 3D path finding problems of any dimension, rather than a multilayered 2D approach as used by VLSI routing algorithms. The algorithm supports routing of multiple nets and in the same search space as well as multi-terminal nets.

Rules are implemented in the algorithm using a cost function approach similar to that of the A* algorithm discussed above. A score for each node is calculated based on distance from source, estimated distance to target, and a weighting based on a number routing rules determined using an inference engine. The algorithm favours nodes with lower scores. At this stage in the system development, a proof of concept has been demonstrated, implementing a path finding algorithm using cost functions to determine path steps. The currently capability of the path finder satisfying routing rules is quite basic, however, future work on the system will focus mainly on integration of an inference engine with the system to make more detailed decisions of path movements based on the rules described earlier.

A screen capture of a 3D demonstration applet is given in Fig. 5 below. 3D Visualisation of results is achieved by writing a Finite Element input file for NASTRAN which defines the path by nodes connected by one dimensional elements, with obstacles given as solid 3D elements. An example of the finite element visualization of the problem shown in Fig. 5, is given in Fig. 6. At this stage no optimisation of the algorithm has been attempted thus the efficiency is limited to $O(d_3)$.

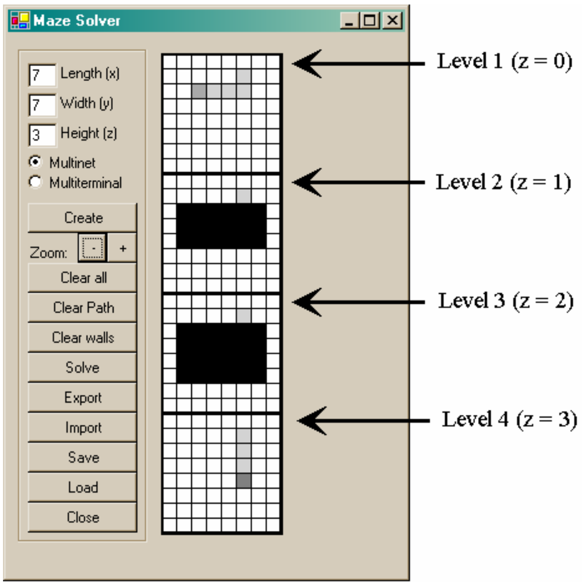

7 Length (x)
7 Width (y)
3 Height (z)
● Multinet
○ Multiterminal

Fig. 5. Three dimensional path finding algorithm demonstration applet. Blue: start location, Red: finish location, Black: obstacle, Yellow: path

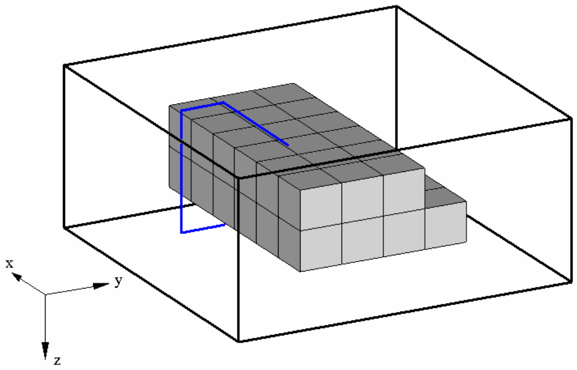


Fig. 6. Three dimensional representation of search space, path and obstacles using Finite Element software

5 Geometry Factors

To apply a grid-based algorithm to the routing problem, the 3D model of surrounding structure and obstacles with source and target positions is to be discretised. Currently two methods of geometry preparation are under consideration which use Finite Element (FE) and Voxel modelling techniques.

5.1 Finite Element

The first method uses a finite element modelling approach which is common procedure in the engineering process of analysing the response of structures to loading. For this method, empty space in the model will be meshed using solid elements (for example TETRA or CHEXA elements, Fig. 7), and this mesh searched for a valid path between given source and target. Property cards are used to identify between structural mesh and the search space mesh. The advantage of this method is that geometry importing and meshing is standard practice in the engineering design process, and existing knowledge and software can be utilised.

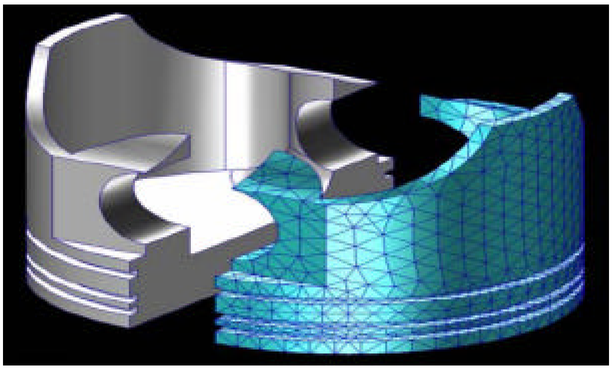


Fig. 7. Finite element solid mesh [11]

The searching process would proceed as shown in Fig. 8 and would begin by searching for and selecting the starting node and then querying the element list for attached elements. The algorithm would then select the "best" attached element based on a rule module and the direction of the target. From this element, attached nodes can be determined and the best node selected based on similar rules. This process of selecting nodes then attached elements would continue until the target is found and all conditions satisfied.

One of the disadvantages of this method is irregular mesh shapes and inconsistency of element and node ID labeling (i.e. element no. 1 may not necessarily be attached to element no. 2, etc.), and reflecting changes in the mesh.

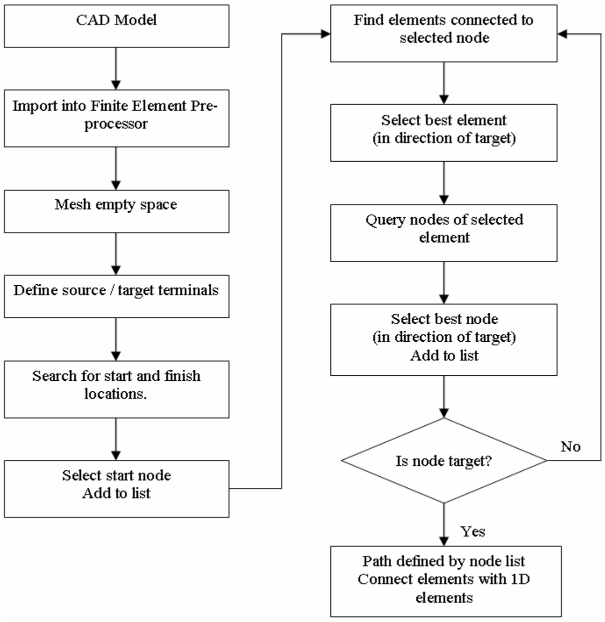


Fig. 8. Searching process using finite element methods for geometry discretisation

5.2 Voxels

3D models in virtual environments such as CAD/CAM applications and computer games are traditionally represented using surface graphics which are composed of polygon meshes (usually triangles) over hollow shells. For high quality visualisation of models, a large number of polygons is required. For each polygon complex computations are performed to determine shading requiring high end computer graphics hardware for display.

Volume graphics is an emerging technology in 3D model representation which uses stacks of 3D cube elements called voxels (or volume pixels) which are analogous to pixels in 2D images [13]. Voxels are defined by a number of characteristics including size, address (in x, y and z coordinates), state (on or off), colour, density, etc. (See Fig. 10 taken from [5] a provider of volume graphics software). A voxel rendering engine assumes all voxels are facing the "camera" at all times as the model is rotated, thus the memory requirement can be reduced to a single position and colour for each element and global voxel size. Whereas surface graphics require complex computations to determine shading, volume graphics can be displayed without use of high end 3D acceleration hardware.

The main advantage of this method lies in ease of implementation. Whereas surfaces modelled in traditional CAD software use complex relationships, volume graphic representation uses a fixed x, y, z integer-based address for each element allowing a grid based algorithm (such as those discussed in section 3) to be directly applied to the problem. The searching process would proceed as shown in the flowchart in Fig. 9.

The disadvantage of this method is the availability of voxel driven software. Some open source voxel engine software is available freely on the internet, however the power of these applications is limited, requiring expensive commercial software (such as that used in laser scanning of three dimensional models). Many commercial software packages include conversion utilities from common CAD model formats (eg. IGES) to voxelised representation.

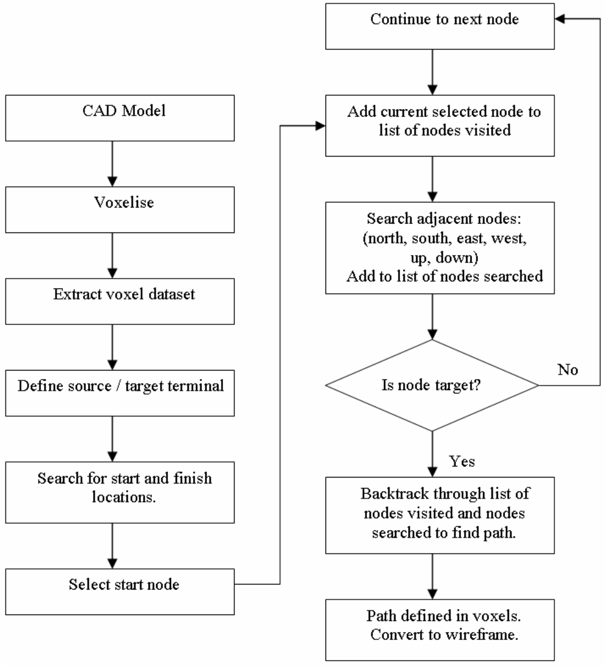


Fig. 9. Searching process using voxel methods for geometry discretisation.

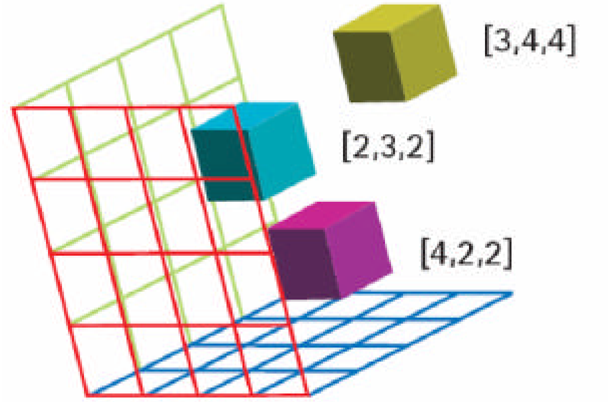


Fig. 10. Voxels represented in three dimensional space [5].

6 Knowledge Based Router

The knowledge based routing system comprises a number of steps which are as follows. Firstly, physical structure is designed and modelled using CAD software by structural engineers. Electrical or piping requirements are defined in terms of start and finish locations and other relevant characteristics such as category of load etc., and are given in the form of a netlist. The 3D model is then exported from the CAD package using a neutral file format such as IGES or STEP. The CAD model is converted to a discrete format suitable for applying a grid based search algorithm using either finite element or voxel techniques as discussed in section 4. The discrete data set is then extracted and fed into the maze algorithm which determines the paths for multiple nets given constraints stored in the knowledge module. This module can be interchanged for different routing applications, not necessarily limited to aircraft (for example a rule set for air conditioning ducts in buildings could be developed). After execution of the routing algorithm, the output path (defined in FE or voxel elements) is converted to wireframe geometry which is imported into a CAD package and detail added according to the knowledge base consulted in the process. These steps are summarised in the following flowchart.

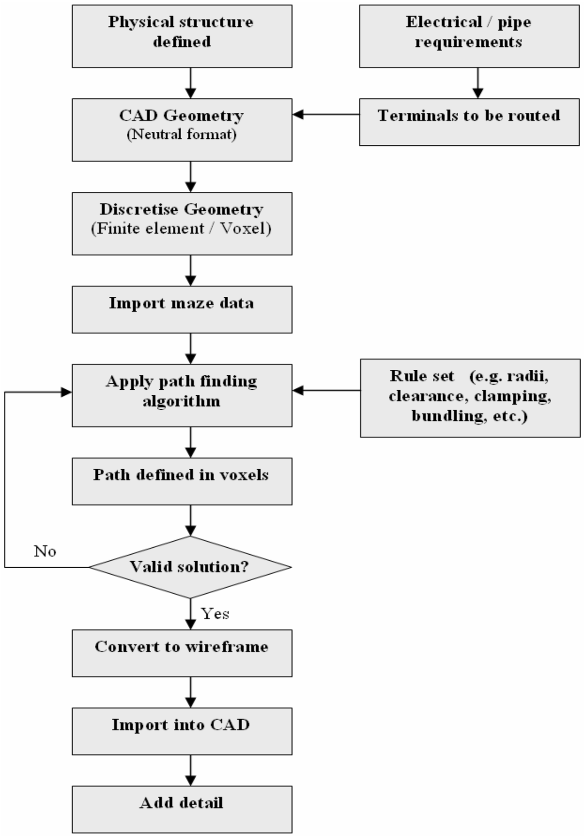


Fig. 11. Knowledge based router process flow.

7 Conclusion

This paper has briefly covered the concept of a knowledge based router currently in development which will have applications in electrical wiring and hydraulic/pneumatic pipe design. The system itself consists of a number of main elements including: a knowledge base for storing routing rules and constraints, a method of discretising geometry using either finite element or voxel modelling techniques, and an intelligent path finding algorithm to navigate the structure and return a valid path for the wire/pipe.

References

[1] Arnold, M. H., and Scott, W. S. *An Interactive Maze Router with Hints.* Lawrence Liver-more National Laboratory, University of California, 1988.

[2] Groeneveld, P. *Electronic Design Automation, Part 1.* Technische Universiteit Eindhoven, The Netherlands, 2005.

[3] Guruswamyl, M., and Wong, D. F. *A General Multi-layer Area Router.* The University of Texas at Austin, Department of Electrical and Computer Engineering, 1991.

[4] Joobbani, R., Siewiorek D. P. *WEAVER: A Knowledge-Based Routing Expert.* Department of Electrical and Computer Engineering, Carnegie-Mellon University, 1985.

[5] Kaas, E. *The NGRAIN Technology Difference Explained A Whitepaper For Technical Evaluators Of Visualization And Simulation Technologies.* NGRAIN Corporation. Vancouver, Canada.

[6] Kang, S.-S., Myung, S. and Han, S.-H. *A Design Expert System for Auto-Routing of Ship Pipes.* Journal of Ship Production, 1999.

[7] Lee, C. Y. *An Algorithm for Path Connections and its Applications.* IRE Transactions on Electronic Computers vol. EC-10, no. 2, 1961.

[8] Lester, P. *A* Pathfinding for Beginners.* Almanac of Policy Issues website, 2005. URL: http://www.policyalmanac.org/games/aStarTutorial.htm

[9] Lunow, R. E. *A Channelless, Multilayer Router.* Lawrence Livamore National Laboratory, California, 1988.

[10] "MIL-W-5088L: Military Specification – Wiring, Aerospace Vehicle". Department of Defense, United Stares. 10 May 1991.

[11] MSC Software. *MSC.Patran Data Sheet.* MSC Software Corporation. United States, 2004.

[12] Sadeghi, M. *Electrical Wiring Practices.* Federal Aviation Administration. Aircraft Electronics Association Convention, 2003.

[13] Stevenson, A. *Voxels And Volumetric Representation.* The University of British Columbia, Vancouver, Canada, 1996.

[14] Tehranipoor, M. *CAD Algorithms – Routing.* Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 2005.

[15] Vakil, D., Zargham, M. R. *An Expert System for Channel Routing.* Computer Science Department, Southern Illinois University, 1988.

[16] Van der Velden, C., Bil, C., Yu, X. Smith, A. *Mathematical Techniques Applied to Knowledge Based Engineering Design Systems.* Engineering Mathematics and Applications Conference, Melbourne, Australia, 2005.

[17] Yoshimura, T., and Kuh, E., S. *Efficient Algorithms for Channel Routing.* IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 1982.