

# A Web-Based Approach to Dynamically Assessing Space Conflicts by Integrating BIM and Graph Database

Wei-Ting Chien<sup>a</sup> and Shang-Hsien Hsieh<sup>a</sup>

<sup>a</sup>Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan  
E-mail: [r06521604@caece.net](mailto:r06521604@caece.net), [shhsieh@ntu.edu.tw](mailto:shhsieh@ntu.edu.tw)

## Abstract –

Construction activities require materials or resources to be arranged while planning the route of how these materials or resources can be transported is not a simple process. Traditionally, subcontractors will arrange a daily meeting and negotiate the desired space on site for the next day's resource transporting. However, many of them only use 2D drawings or aerial images while discussing the routes which lack an accurate calculation on space conflicts. In this paper, an approach combining Building Information Modelling (BIM) and Graph Database (GraphDB) is proposed. With the help from 3D BIM models, the coordinates of resource transporting routes can be first extracted and further analysed by the proposed approach. From the result, the collisions along the planning routes can be detected to assist in making a proper transporting plan. In addition, a web-based platform is developed for validation of this approach. Because the platform is a shared one, each user can arbitrarily plot their desired transporting routes in the model at any time. As long as the actual transporting schedule is known for the corresponding route, the platform can dynamically and instantly inspect whether the desired routes will collide with others. In this way, we can eliminate conflicts prior to the start of those activities and potential delay in the construction process can be reduced.

## Keywords –

Building Information Modelling; Construction Conflicts; Construction Management; Separating Axis Theorem; Graph Database

## 1 Introduction

Construction is a complex process which requires a proper plan prior to its commencement. A proper construction plan can assure the owner of the construction's quality and punctuality while a poor plan can not only cause time overrun but also induce a waste of investment. In an attempt to make a decent plan, several issues should be addressed first, including

resource shortage, labour shortage, workspace conflicts, and activity sequence conflicts. With the advent of Building Information Modelling (BIM) and other 3D digitalisation technologies, workspace conflicts and other geometric-related issues can be detected and mitigated. For example, Moon et al. [1] demonstrated that using an algorithm along with the model bounding box concept can assist in determining the adjacency distance between two conflict points. With the information as to where the conflict occurs and the percentage of conflict between two workspaces, the conflicts can be handled and reduced in advance. Similarly, Wu et al. [2] used 4D simulations and colour visualisation techniques on the BIM model to detect space conflict while the required workspace for activity was categorised into five types. In addition, some recent research concerned that the movement of workers could potentially raise workspace conflicts. Therefore, Mirzaei et al. [3] defined 16 execution alternatives for workers' movement patterns and developed a quantification system for calculating the workspace conflict impact level on the project through 4D BIM simulations.

As plenty of research is already concentrating on workspace management, another group of research is focusing on construction materials management. For example, Said and El-Rayes [4] presented an optimised model for making critical decisions on construction logistics while considering both material procurement and material storage on construction sites. Golkhoo and Moselhi [5] proposed a multi-layer perceptron approach to create an optimised material delivery schedule. This schedule enabled contractors to not only procure construction materials with minimum expenditure but also avoid shortage or surplus issues. Although the aforementioned research has helped materials be managed properly on site, little research is concentrating on how the materials can be transported safely from the staging area to the activity execution area in a construction site. Unlike the workspace for each activity which can be defined earlier within the construction plan, material transporting routes are normally assigned by the day before the corresponding activity executes. Furthermore, the transporting routes are not determined

by only one person such as the project manager. Instead, the routes are selected by different subcontractors who are in charge of those activities. In this case, a process is required for enabling users to dynamically assign their routes and automatically detect the route collisions. In addition, a shared platform for multiple users to coordinate their route collision detection is desired.

In this paper, an approach integrating BIM, Separating Axis Theorem (SAT) and Graph Database (GraphDB) is proposed for automatic route collision detection. SAT supports the detection as to whether two objects collide. GraphDB offers an efficient way of storing and extracting the assigned route data and the relationships among activities, materials and routes. BIM serves a visualisation platform and provides data related to the facilities under construction. With the development of a web-based BIM model viewer, all subcontractors are allowed to assign their desired transporting routes by drawing them directly in the BIM model. The viewer will automatically calculate and check whether the newly assigned route collides with the existing routes created by the previous users. In this way, the subcontractors will be able to plan their material transporting routes without requiring that all subcontractors be gathered in the same physical space. As long as they have access to the Internet, they can have discussions about route planning on the proposed platform with automatic route collision detection supported.

## 2 Methods for Route Conflict Detection

To develop an approach for assessing route conflict, the core algorithm for examining the collisions of objects needs to be proposed first. In this research, SAT is selected to achieve this. On the other hand, how to efficiently and accurately store and extract the route and collision data is another crucial part. Therefore, a graph database is adopted to enhance the efficiency of using the collected data.

### 2.1 Separating Axis Theorem

SAT is aiming at detecting collisions of two objects by examining if any separating line or plane exists between the objects [6]. The theorem can be applied to both 2D and 3D objects. It says if two 2D objects do not collide with each other, there must be at least one line which can separate the two objects without any intersections (Figure 1). That line is called a separating line. Equivalently, 3D objects can be deemed as not colliding by finding at least one separating plane to separate them from overlapping (Figure 1). As implied, there can be more than one separating line or plane for two separate objects. For example, those dashed lines in 2D object illustration shown in Figure 1 are also separating lines. However, finding all of them is not our

intention. Efficiently identifying one existing separating line or plane is sufficient to say that two objects are not colliding. Therefore, SAT allows for a rule-based algorithm to significantly narrow down the necessary lines or planes for checking. The algorithm proposes 4 and 15 equations, respectively, for checking whether the 2D and 3D objects are colliding with each other. Each equation represents an examination of one possible separating line or plane. If any of the equations is fulfilled, that means the corresponding line or plane succeeds in separating the two objects and the two objects are proved to be not colliding. In other words, if all of the equations cannot be fulfilled, the objects are proved to be colliding with each other. This strategy saves us a significant amount of time because only 4 lines or 15 planes need to be checked for every collision examination. There exists numerous possible lines or planes to separate two objects whereas this algorithm allows us to inspect just a few lines or planes to guarantee an accurate result as to whether the two objects are colliding or not. For details about the equations, please refer to the tutorial document by Johnny Huynh [6].

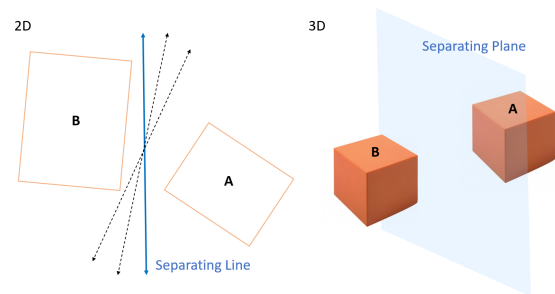


Figure 1. Separating axis theorem illustration

In this research, 3D object collision examination from SAT is applied. A material transporting route is approximately represented by a set of cuboids as shown in Figure 2. In other words, each route can be checked whether they are colliding with other routes simply by examining whether any cuboid in the set of targeting routes is colliding with another route's set of cuboids. For ease of generating a set of cuboids, a central curve of the route should be given first. After that, the curve is further segmented into a list of lines. This is a way of simplifying the curve representation and the point density value can be either specified by the computers automatically or assigned by the users based on their own preference. As the curve is already transformed into a list of segmented lines, determining the width and height of each segment is crucial since the ultimate goal is to transform them into a set of cuboids. Therefore, the width and height of the object using this route can be taken into consideration. For example, assume that the route is assigned for some heavy vehicles to drive through, then the width and

height of the vehicle should be taken into calculation in the collision examination. To achieve this, the segmented lines are horizontally expanded by half of the vehicle width on both sides to represent the total width needed while the vehicle is taking this route. Likewise, the curve is also expanded vertically by the vehicle height to provide the height for the vehicle to pass. In this case, with the segment length already determined in segmenting the curve, a cuboid for every segment can be generated as shown in Figure 2. While each route is transformed into this form of representation, SAT can be applied for checking whether the created routes are colliding with one another.

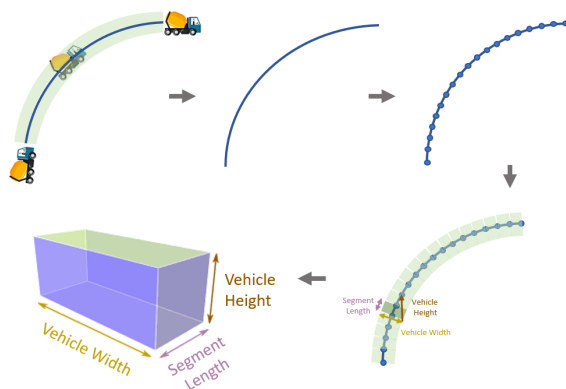


Figure 2. A demonstration of how a route is generated for applying SAT

## 2.2 Graph Database

Graph database is a database which uses graph as the main structure of storing and querying data [7]. While a graph is generally formed by nodes and edges, it has a greater ability of revealing and explaining complicated relationships among all data. This relatively new concept or technology can be applied to various fields. For example, Sadowski and Rathle [8] proposed using a graph database for finding common patterns from three of the most damaging types of fraud. Cattuto et al. [9] presented a time varying graph database structure for efficiently querying the social network formed with high-resolution records of human activities extracted from mobile devices and sensors. In the field of construction or BIM, Ismail et al. [10] suggested that the 3D IFC models should be automatically transformed into the graph database structure which enables users to manage and analyse the huge amount of building data and complicated relationships. The application of the graph database depends on how to define and manage the nodes and edges.

In this research, three types of nodes are specified,

including material, activity and route, which are assigned in green, yellow and blue colour, respectively (Figure 3). Prior to the collision examination, where the route starts and where it ends should be defined first. In this research, it is intended to detect whether a collision may happen while a material is transported from the staging area to the activity area. Therefore, the route's start node is assigned as the staging area and the route's end node is deemed as the activity area likewise. Both nodes encompass the coordinate information of the corresponding area. After the start and end nodes are determined, the route's intermediate nodes are specified by users' drawing of needed transporting routes in the BIM model. As mentioned before, the drawn curve is first interpreted by a list of points and segmented lines. Each point's coordinates are stored in each intermediate node. Furthermore, while the next step is to expand the segmented lines and transform them into a set of segmented cuboids, the information of each cuboid, including width, length, height, can also be stored in the edge (path relationship) between each pair of two adjacent nodes. In this way, each chain in the database can indicate a route for examination and it already contains sufficient information for SAT collision detection including the dimension of each cuboid and its coordinates. Two example chains are illustrated in Figure 3.

After the route data are stored in the database, having them efficiently extracted for SAT examination is another critical part. Taking advantage of the structure of graph database, we can read the cuboid data with just a few lines of code, for example:

```
MATCH (m{name: "<material name>"})-  
[rs*]-(a{name: "<activity name>"})  
RETURN rs
```

In the above code, m and a are representing the material and activity nodes respectively. Rs\* indicates all the relationships along the path from node m to node a. When "return rs" is called at the very end of the code, all the relationships are extracted and returned from the database. In addition, by taking the cuboid coordinates and dimension properties from the relationships (edges), 15 equations can be applied to each pair of cuboids to check whether they are colliding with each other. If any two cuboids are colliding; for example, the two relationships highlighted in pale red in Figure 3 are colliding, then the start nodes of the two relationships will be linked together by an extra relationship called conflict. This relationship is used as an indicator of collision between the two routes to help users identifying all the colliding routes by a few lines of code:

```
MATCH (a1:ACTIVITY) - [:PATH*] - () -
```

```
[rs:CONFLICT] - () - [:PATH*] -
(a2:ACTIVITY) RETURN a1,a2
```

In this graph structure, each pair of colliding routes can be detected. The code mentioned above is basically traversing the graph from one activity node to another and finding paired activities with a conflict relationship as a part of the path. For example in Figure 4, if activity 1 (a1) is able to reach activity 2 (a2) in the graph, the path must contain the conflict relationship because any route chain is independent of others in the very beginning until an extra conflict relationship is established. Therefore, by using this approach, users like subcontractors can make a better material transporting plan by detecting any conflicts in advance.

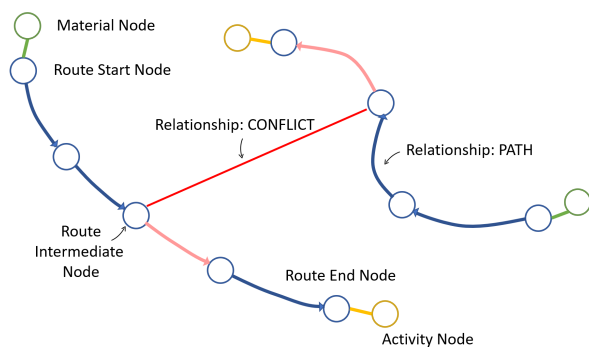


Figure 3. The structure of route data stored in graph database

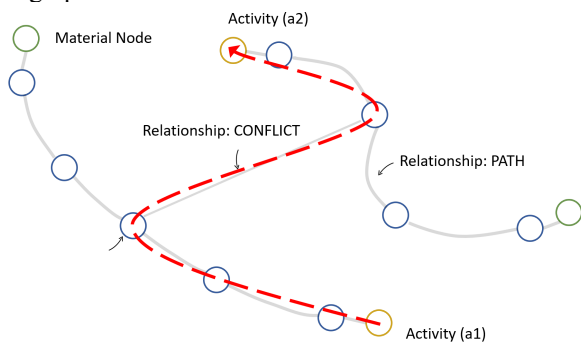


Figure 4. Example for finding colliding routes by traversing the graph

### 3 Web-based Collaborative Platform

In this research, a web-based collaborative platform is developed for detecting the occurrence of route collisions and making a better material transporting plan. Therefore, a Javascript library called React.js is selected to be the main language for building the whole website. For styling-related works, Bootstrap is adopted by virtue of its convenience. Furthermore, a BIM model viewer should also be provided for reference while users are

drawing their material transporting routes. Therefore, a web-based BIM model viewer called Autodesk Forge is employed. The whole viewer is based on another Javascript library called Three.js which is mainly responsible for 3D objects' rendering. For the users' drawn routes, Three.js also helps in displaying them on the 3D models. Regarding backend, Node.js is chosen as the main language handling server-side requests and responses. It is also used for connecting with the graph database management system called Neo4j which assists in creating a database instance and managing the behaviour of the database. The frontend and backend architecture of the platform from the language perspective is shown in Figure 5.

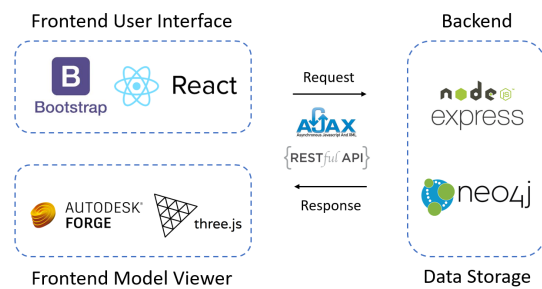


Figure 5. Frontend and backend architecture from the language perspective

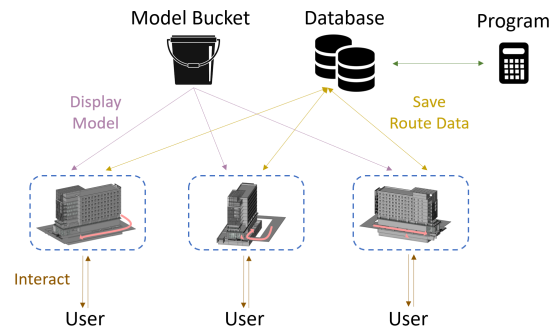


Figure 6. Data flow architecture for the web-based platform

The data flow for the platform (Figure 6) can be split into two parts. The first part concerns 3D BIM models and non-geometric data related to them. For the use of Autodesk Forge, models need to be uploaded to the buckets created in Autodesk server. When models are already in buckets, they can be displayed on any web pages as long as that page contains the configured Forge model viewer. The other part concerns the data related to material, activity and route. As mentioned before, material and activity nodes should be created first with their own coordinates. Those data are stored in the graph database instance as nodes. With that all done, users are

allowed to draw their own material transporting routes and those routes are then transformed into intermediate nodes and links stored in the database. In addition, the moment the route data are stored in the database, the collision examination function written in another javascript program is triggered to check whether there are collisions. If the result is positive, the program automatically creates new conflict relationships between each pair of colliding routes and notifies the frontend Three.js program to display the colliding routes in the 3D model using some conspicuous colours. In this way, users can have clear grasp of which routes are acceptable and which routes need modification.

#### 4 Demonstration

The proposed approach is demonstrated using a prefabricated construction building. In prefabricated construction, most structural components are preliminarily assembled at the manufacturing site. With all processes being completed, the components are further transported to the real construction site for final assembly. Without massive works for on-site grouting, lifting and installing the assembled structures to their designed location can occupy the primary workflow. In this research, the BIM model of the civil engineering research building at National Taiwan University is used. For the purpose of demonstration, the model is filtered and contains only the first floor and the basement. Along the both sides of the building, there are several beams requiring installation for resisting the loading from upper levels. In order to transport the beam from its staging area to the location it was designed, a transporting plan and collision examination should be made first to ensure a smooth process.

The proposed platform is developed to assist in making collision examinations to ensure proper material transporting plans. In the beginning, the platform offers a panel for inputting information including name and execution period of the targeting activity. In addition, the location of the activity should also be assigned by clicking on the 3D model. The program then automatically extracts the coordinates of the clicked point and wraps all the information of the activity into the activity node. Similarly, information such as name and location of a material should be inputted from the platform likewise. Most importantly, relationships between activities and material need to be specified to indicate which material will be used by which activity. With all these prerequisites defined, users can start making the collision examination. By selecting the activity name, the place the activity takes place, the material to be transported and the heavy equipment to use, the system will then be notified about the start point, end point, width and height of the creating route. For easy

specification of the equipment, the system has already embedded some crane trucks with its dimension (e.g. HD65 4x2 and HD120 4x2 in Figure 7) for users to select.

From the previous step, the system knows which material and activity the creating route is for. Since the coordinates of the two activities were also defined previously, the start point and destination of the route can be easily displayed in the digital model as the two dark green spheres shown in Figure 8. These actions are for helping users manage the route direction intuitively. After careful consideration, users can draw their own routes as the intermediate points between the start and end green spheres as shown in Figure 8. Once the users finish creating the route and click on the finish button, the data storage process and the collision examination are triggered. If the examination program detects any collision, the colliding routes are displayed with different colours in the BIM model as demonstrated in Figure 9. This collision examination not only detects the geometric overlapping but also takes into account the time period of the routes. Even if the two routes are detected to have geometrical collision, as long as they are not executed at the same time, they would not be considered as having collision by the system. In Figure 9, the system shows that the red and blue routes are colliding on the right side of the building to inform the user to handle the conflict prior to the start of the transporting process.

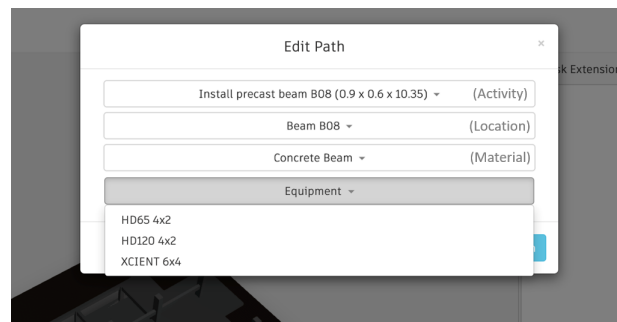


Figure 7. Selection panel for creating a route

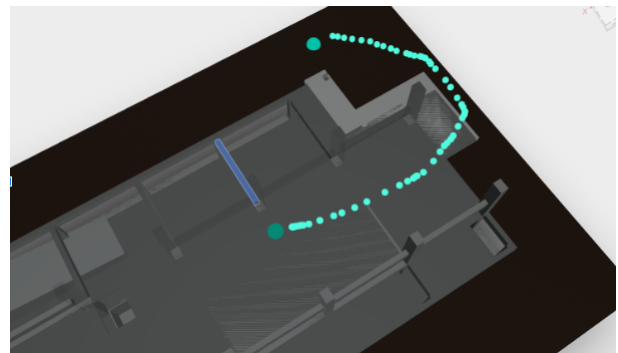


Figure 8. Route drawing demonstration

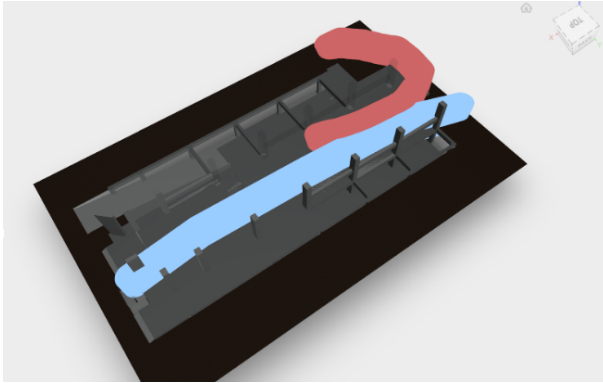


Figure 9. Result for collision examination

## 5 Conclusion

Construction is a complicated process which often involves different kinds of conflicts. In order to mitigate the negative impact the conflicts can make on a project, a construction plan is required for assuring every activity can be smoothly executed. As space conflict is one of the most common conflicts seen in many construction sites, plenty of research has already addressed it. However, research dealing with material transporting conflict is seldom seen although this kind of conflict is still frequently happening and may cause time overrun of a project. In view of this, this research proposes an approach that integrates BIM, SAT and GraphDB for automatically examining potential colliding routes of transporting construction materials. The result from the conflict examination can further assist the managers in modifying some improper material transporting routes. Furthermore, a web-based coordination platform is developed with the proposed approach to help authorised subcontractors view the BIM model and create their own transporting route plans. As long as they input the necessary information regarding the activity, material and equipment in advance, they are allowed to draw their own routes. The system then automatically checks whether a newly created route is colliding with the existing routes previously created by other users. In this way of coordination, subcontractors can make sure their material transporting routes are exempt from having conflicts with others and do not need to plan a meeting in which everyone is usually asked to use 2D drawings for resolving conflicts. This platform offers an accurate and convenient tool to detect the route conflicts and assist in making a better material transporting plan.

## References

- [1] Moon, H., Dawood, N., & Kang, L. (2014). Development of workspace conflict visualization system using 4D object of work schedule. *Advanced Engineering Informatics*, 28(1), 50-65.
- [2] Wu, I., & Chiu, Y. (2010). 4D Workspace conflict detection and analysis system. *Proceedings of the 10th International Conference on Construction Applications of Virtual Reality*, November 4-5, 2010, Sendai, Japan.
- [3] Mirzaei, A., Nasirzadeh, F., Parchami Jalal, M., & Zamani, Y. (2018). 4D-BIM dynamic time-space conflict detection and quantification system for building construction projects. *Journal of Construction Engineering and Management*, 144(7), 1-14.
- [4] Said, H., & El-Rayes, K. (2011). Optimizing material procurement and storage on construction sites. *Journal of Construction Engineering and Management*, 137(6), 421-431.
- [5] Golkhoo, F., & Moselhi, O. (2019). Optimized material management in construction using multi-layer perceptron. *Canadian Journal of Civil Engineering*, 46(10), 909-923.
- [6] Huynh, J. (2009). Separating axis theorem for oriented bounding boxes. Available from: [jkh.me/files/tutorials/Separating%20Axis%20Theorem%20for%20Oriented%20Bounding%20Boxes.pdf](http://jkh.me/files/tutorials/Separating%20Axis%20Theorem%20for%20Oriented%20Bounding%20Boxes.pdf).
- [7] Graph Database. (2020). In Wikipedia. Retrieved on June 10, 2020 from [https://en.wikipedia.org/wiki/Graph\\_database](https://en.wikipedia.org/wiki/Graph_database).
- [8] Sadowski, G., & Rathle, P. (2015). Fraud detection: Discovering connections with graph databases. *White Paper, Neo Technology*, CA, USA.
- [9] Cattuto, C., Quaghiotto, M., Panisson, A., & Averbuch, A. (2013, June). Time-varying social networks in a graph database: a Neo4j use case. *Proceedings of the First International Workshop on Graph Data Management Experiences and Systems*, Jun 22-27, 2013 New York, USA (pp. 1-6).
- [10] Ismail, A., Nahar, A., & Scherer, R. (2017). Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard. *Proceedings of the 24th International Workshop on Intelligent Computing in Engineering (EG-ICE 2017)*, July 10-12, 2017, Nottingham, UK.