

MODELLING OF INFRASTRUCTURE FOR WAREHOUSE SIMULATION

Michal KOCIFAJ

University of Žilina, Faculty of Management Science and Informatics,
Slovak Republic

e-mail: michal.kocifaj@fri.uniza.sk

Abstract

The construction of new large-scale logistic center or warehouse can be a very costly operation. Therefore, a decision about its future location and layout should be verified before the actual building begins. One of the usable verification methods is modelling and simulation. This paper is focused on modelling of movement of handling equipment, e. g. forklifts. Two types of infrastructure are considered. The first type, free-range area, is represented by a two-dimensional cellular structure and it is combined with a predefined fix layout mesh called "skip-links". A hybrid path finding algorithm utilizing cells and skip-links is presented as well.

Keywords: *cellular infrastructure, warehouse, simulation*

1 INTRODUCTION

Nowadays, thousands of tons of goods are transported from the manufacturers to the vendors and customers daily. After the freight is delivered to the vendor, it can be stored for a certain time in warehouses, where it waits for pick up. In this paper we will focus on logistic centers and warehouses, where the freight is unloaded from vehicles or loaded onto them using special equipment, like forklifts, for instance. These forklifts move on a defined infrastructure, which can be modelled by a predefined paths layout or by a free-range area.

The free-range area can be found useful in front of warehouse, where the vehicles come in order to unload or load. There are no marked paths, thus the handling equipment can move directly to the target vehicle or rack in warehouse respecting only existing static and dynamic obstacles. Two basic approaches can be utilized for the free-range infrastructure modelling. The first one is path finding on a graph. This graph may be constructed by splitting of the available space into cells. Centers of cells represent graph nodes. If the cells are adjacent, then the corresponding graph nodes are

interconnected by an edge. A complete and optimal path towards the goal can be found on this graph. The second usable approach is movement based on magnetic [7] or social forces [3], respectively.

In this paper we present an approach using free-range area, split into square cells, combined with a predefined skip-links mesh layout.

The paper is organized as follows. The second section introduces the cellular infrastructure model combined with skip-links. The connection between cells and warehouse racks is discussed as well. The path finding is briefly discussed in the third section. The last section presents the test infrastructure model.

2 MODEL OF INFRASTRUCTURE

2.1 Cellular infrastructure

The free-range area for vehicles movement can be split into cells of various shapes, centers of which form nodes of path finding graph. The graph edges are created by connecting centers of adjacent cells (see Fig. 1).

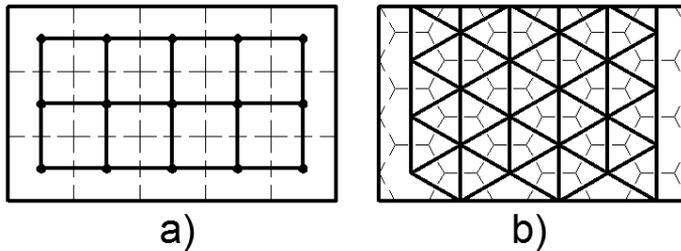


Figure 1 Graphs for path finding created from a) square and b) hexagonal cells

Most widely used are square and hexagonal cells. Brief comparison of squares and hexagons is presented in Table 1. The main differences between them, except their shape, lie in the different count of neighbors, the coordinate system, the searching for cell under a given point and the possible cells arrangements.

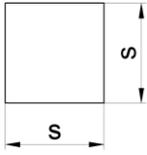
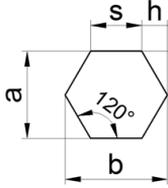
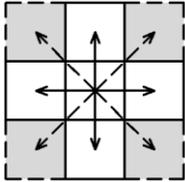
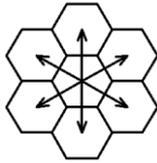
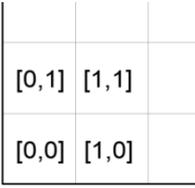
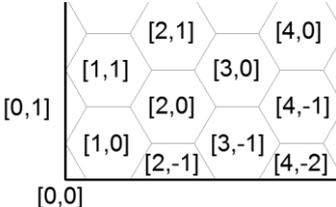
The cells can be stored in a two-dimensional array (zero based in our implementation). The coordinate system for squares is trivial. The first index represents column, the second defines row, as shown in Table 1.

The coordinates in a hexagonal world are a little more complicated. We have used a system involving also negative indexes. These have to be converted into a zero-based system. If we inspect coordinate system from Table 1, a simple relationship between rows and columns can be found:

$$row := row + (column \text{ div } 2) \quad (1)$$

where *div* is an integer division. For example, the simulation coordinates [4,-2] will be transformed to the real array indexes [4, 0]. For arrangement depicted on Fig. 2b) the row and column indexes have to be exchanged.

Table 1 Comparison of square and hexagonal cells

	Square cells	Hexagonal cells
Dimensions		 $h := s \cdot \sin(30^\circ)$ $r := s \cdot \cos(30^\circ)$ $b := s + (2 \cdot h)$ $a := 2 \cdot r$
Neighbourhood		
Coordinate system		

Searching for cell at particular world coordinates $[x,y]$ for squares is very simple, especially if the cells are aligned to the simulation world axis:

$$coord.x := Floor(x / s) \tag{2}$$

$$coord.y := Floor(y / s) \tag{3}$$

where the *Floor* function returns the nearest lower integer value.

In case of hexagons, more complex calculation has to be used. Solution for cells rotated after Fig. 2a) can be found in [6], calculation for hexagons arrangement depicted on Fig. 2b) is described in [4].

We have explored the square and hexagonal cells. The octagons are inapplicable, because there are undesired square gaps between them (depicted on Fig. 3).

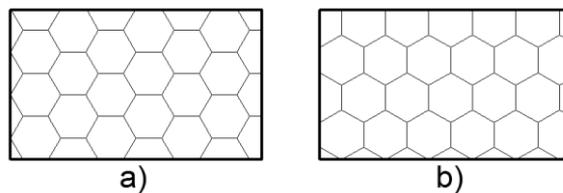


Figure 2 Possible arrangements of hexagonal cells

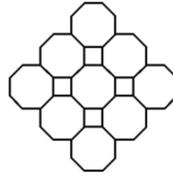


Figure 3 Octagonal cells with undesired square gaps

Additional detailed information about square, hexagonal and also triangular cells can be found in [8].

2.2 Mapping warehouse racks to cells

Before a path finding and movement can begin, the static obstacles have to be mapped to the infrastructure. In our warehouse simulation they are formed by pallet racks with regular rectangular shape and these rectangles have to be connected to our cellular world. There are three possible mapping modes:

- The warehouse racks are aligned to the infrastructure, their dimensions are changed in accordance with the cells dimensions, see Fig. 4a).
- The cells are adapted to the racks. Thus, more cellular subsystems have to be created and interfaces between them have to be defined (marked with dashed lines on Fig. 4b)).
- The racks and cells are placed independently and connection is done consequently. All cells under racks' shapes are considered blocked and impassable. Situation is depicted on Fig. 4c). The blocked cells are hatched.

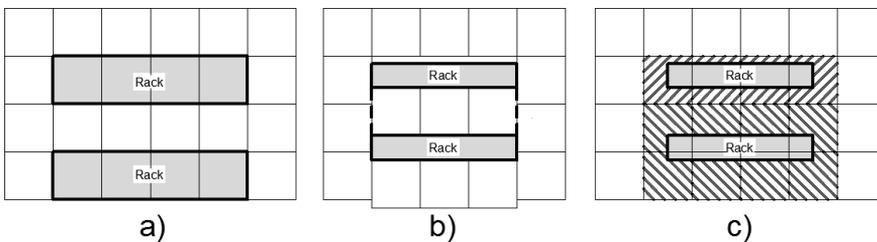


Figure 4 Warehouse racks to cells mapping. a) Racks are adapted to cells; b) cells are adapted to racks; c) independent cells and racks

In our work, a detailed simulation was required and the racks could not be moved or modified. Thus, the decision between options b) and c) had to be made. Option b) needs an interface definition; option c) fails in some situations, when neighbor racks block all cells between themselves (see Fig. 4c)). Both of them were applicable, but we have decided to explore the method c) combined with a predefined mesh layout called *skip-links* in critical blocked spaces. Option b) requires more cellular subsystems

in a single area and we have found it more complicated. However, it can be examined in the future and compared to the currently chosen method.

The warehouse infrastructure modeling using hexagonal cells is much more complicated in comparison with square cells, especially due to racks to cells mapping. Thus, only the square cells will be discussed in the next chapters.

2.3 Skip-links

The graph created from cells' centers is very handy on open area without any obstacles or with a few distant obstacles, but this approach fails in narrow parts of the modeled space (refer to the Fig. 5). In reality the forklift could move past the racks, but in the cellular world is it not possible because the cells under racks are considered blocked.

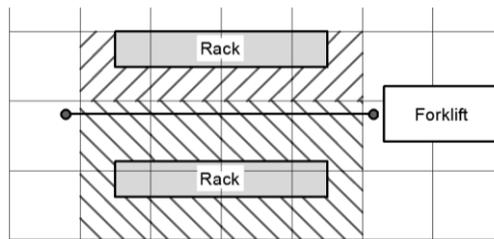


Figure 5 Cells under racks. The blocked cells are hatched. The angle of hatches changes according to the corresponding rack. A skip link with a start and end point is outlined between racks.

One of the possible solutions is to create additional connecting line, so called *skip-link*, into existing path finding graph. Then, a hybrid path is found using cells up to the racks region, combined with a skip-link between racks and then continuing on cells again.

The skip-link is registered into two cells – the cells under its start and end point. If the start or end cell is reached during path finding, the both skip-link nodes are processed like they are neighbors of the current cell. The coordinates of the opposite skip-link end are known, and the corresponding cell is determined easily using formulas (2) and (3) from section 2.1.

The skip link is assumed to be an atomic element, it has to be reserved, occupied or released all at once. If a forklift routes to (or from) a rack position, it uses skip-links for moving between racks. When it is positioned near the rack, it does the last necessary motion on cells, because the skip-link could be too far from the rack. Therefore, it temporary leaves the safety of skip-link, but remembers it. For movement into (or out of) the rack the information stored in cells is used to detect boundaries of racks. After the rack operation is done, the forklift uses cells to return back to the nearest point of the last used skip-link.

306 meters long (including racks and skip-links). The skip-links length varies from 5.2 to 8 meters.

Our solution was tested using generic simulation tool Villon [5]. The tests proved the usability of designed solution in different configurations with various racks blocks count (from 1 to 17) and forklifts number (from 1 to 10).

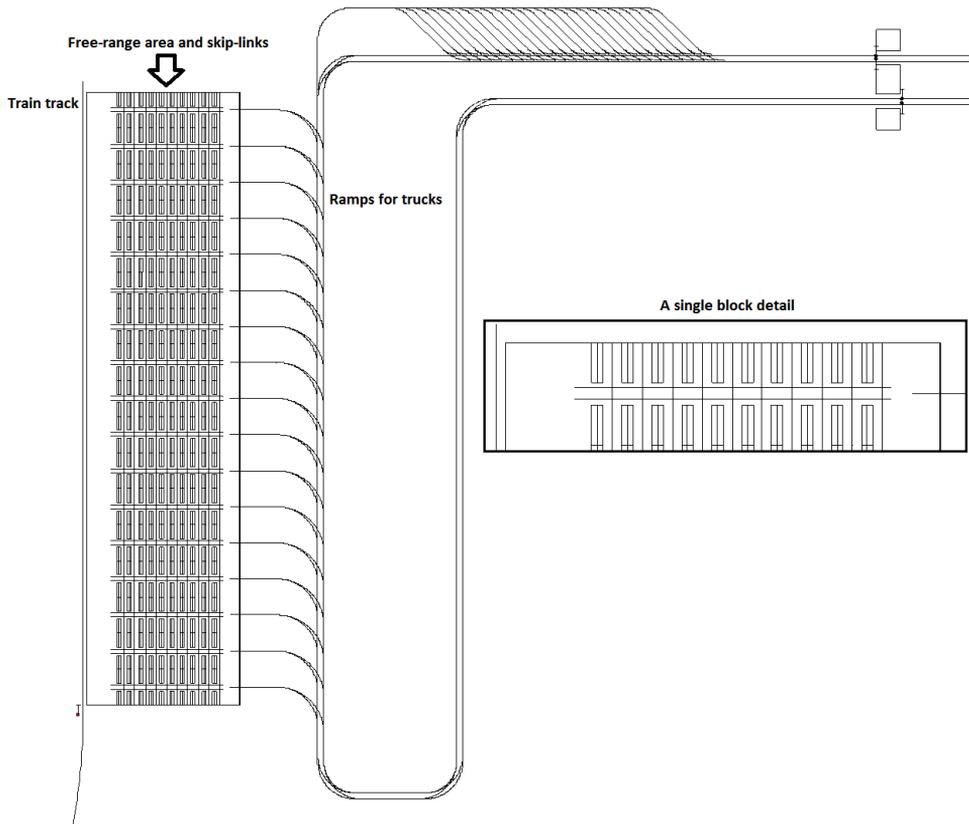


Figure 7 Test model of warehouse with 17 rack blocks

5 CONCLUSION

In this paper we have described one of the possible approaches to the free-range area infrastructure modeling for simulation purposes. The modeled area can be split into cells with square or hexagonal shape. All static obstacles, such as racks, have to be mapped to this cellular infrastructure.

Suggested cellular solution has been extended by predefined skip-links layout, because the initial method was not able to find path in tight corridors (e. g. between racks in the warehouse). The combined graph has been utilized for path finding and

movement of the handling equipment. The collision detection and avoidance system has been developed as well. However, its description is beyond the scope of this paper.

The usability of proposed solution has been proven on a test warehouse simulation model. Achieved results can be utilized in many fields like simulation of warehouses, container terminals, airports and other systems containing free-range area movement.

Further research will be focused on the automatic generation of skip-links layout and to the examination of cellular subsystems method (described in section 2.2 and depicted on Fig. 4b)).

REFERENCES

- [1] DIJKSTRA, E. W. A note on two problems in connexion with graphs. In *Numerische Mathematik*, Vol. 1, 1959, p. 269–271. ISSN 0029-599X.
- [2] HART, P. E., NILSSON, N. J., RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4 (2), 1968, p. 100–107. ISSN 0536-1567. Available at: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4082128>
- [3] HOOGENDOORN, S. P. Pedestrian Travel Behavior Modelling. In *Proceedings of the 10th international Conference on Travel Behavior Research*, Lucerne, August 10-15, 2003.
- [4] JAHN, T. *Coordinates in Hexagon-Based Tile Maps*, 2002. Available at: <http://www.gamedev.net/page/resources/_/reference/programming/sweet-snippets/coordinates-in-hexagon-based-tile-maps-r1800>
- [5] KAVIČKA, A., KLIMA, V., ADAMKO, N. Simulations of Transportation Logistic Systems utilizing Agent-Based Architecture. In *International Journal of Simulation Modeling*, 6 (1), 2007, p. 13-24. ISSN 1726-4529.
- [6] KOCIFAJ, M. *Container terminal module for agent based simulation model of logistic systems*. Diploma thesis. Žilina : Žilinská univerzita v Žiline, 2011, 78 p. (in Slovak)
- [7] OKAZAKI, S., MATSUSHITA, S. A study of simulation model for pedestrian movement with evacuation and queuing. In *Engineering for Crowd Safety*, 2000, p. 271 - 280. ISBN 978-0444899200.
- [8] PATEL, A. *Amit's Game Programming Information*, 2012. Available at: <<http://theory.stanford.edu/~amitp/GameProgramming/>>