Congestion-Aware Warehouse Flow Analysis and Optimization

Sawsan AlHalawani¹ and Niloy J. Mitra^{1,2}

¹ King Abdullah University of Science and Technology - KAUST

² University College London - UCL

Abstract.

Generating realistic configurations of urban models is a vital part of the modeling process, especially if these models are used for evaluation and analysis. In this work, we address the problem of assigning objects to their storage locations inside a warehouse which has a great impact on the quality of operations within a warehouse. Existing storage policies aim to improve the efficiency by minimizing travel time or by classifying the items based on some features. We go beyond existing methods as we analyze warehouse layout network in an attempt to understand the factors that affect traffic within the warehouse. We use simulated annealing based sampling to assign items to their storage locations while reducing traffic congestion and enhancing the speed of order picking processes. The proposed method enables a range of applications including efficient storage assignment, warehouse reliability evaluation and traffic congestion estimation.

1 Introduction

With the advancement of technologies, virtual environment applications have a growing need which poses many challenges to model and visualize man-made systems with high realism and accurate behaviour. In computer graphics, the aim of procedural modeling is to create high quality virtual models that closely mimic real world environments. In many cases, the layout of objects within a model plays a vital role in the functionality of the environment. In this paper, we study the layout and positioning of objects in a warehouse.

The placement of objects in a warehouse has a critical impact on the customer service levels and logistics costs. Managers usually aim to achieve the optimum layout which reduces material handling costs, minimizes space requirements, and lowers energy bills. In warehouse design, there are many techniques to approach the assignment of stock to storage locations. *Randomized storage* policy randomly assigns each item to an available location with equal probability. *Dedicated storage* is to assign a precise number of slots to each product which ensures easy tracking but results in wasted space with seasonal demand goods as the assigned slots can not be reused. *Class-based (ABC) storage* policy assigns the most frequently requested items closest to the input and output point where the loading and unloading happens.

In this research, we analyze objects layout within a warehouse with the aim of increasing its efficiency. There are many competing factors that we explore. Objects should be placed close to the input and output point so that requests can be quickly processed. However, this can lead to congestions as the density of moving people and carriers near the I/O point will be high and they will start blocking each other. Hence, we investigate to find a good balance between the processing speed and resulted congestion. We aim to harvest the advantages of two well known storage assignment policies. We start with a random assignment which prevents the wasted storage drawback of having seasonal items while providing fast processing of requests which is achieved in class-based storage assignment.

Fig. 1. Starting from a random storage assignment, we solve for congestion minimizing storage assignment while ensuring fast processing of orders. Shelves are depicted in random colors based on the assigned item type. Red, green and blue shelves represent the most, medium and lowest demanded items, while yellow represents empty shelves.

Figure 1 gives an overview of our proposed work. First, we analyze traffic flow within a warehouse and then optimize to find the best storage assignment that enhances the traffic flow. We are mainly interested in two questions: (i) what features are suitable to find a storage assignment policy that reduces traffic congestion; and (ii) what applications are possible using the proposed method.

2 Related Work

With the growing popularity of virtual worlds, a great amount of work has been conducted demonstrating various methods for modeling real worlds and generating their configurations. Smelik et al. [1] presented a survey for procedural modeling methods that are useful to generate features of virtual worlds. We discuss some of the most relevant works to our research.

Storage assignment. The goal of storage assignment policies is to determine which product is to be positioned at which location. This was initially addressed by Hausman et al. [2] who proposed a detailed taxonomy of storage location assignment policies such as randomized, dedicated and class-based storage assignment policies.

Similar to arranging items in a warehouse comes arranging furniture in rooms. Merrell et al. [3] proposed to arrange furniture based on interior design guidelines using a hardware-accelerated Monte Carlo sampler for the density function. Yu et al. [4] presented a system to automatically arrange a variety of furniture objects and generate realistic indoor scenes.

Layout structure. Among the most standard layouts are the ones with parallel storage aisles. Some of them add cross aisles in between the parallel aisles. Recently, Gue and Meller [5] introduced the Flying-V and Fishbone designs by relaxing the parallel and orthogonal aisles requirement. Figure 2 shows some examples of different layouts.

Other interesting layouts are building interior layouts. Designing layouts was initially treated as packing problem. Galle et al. [6] implemented an exhaustive algorithm to select the rectangular arrangements satisfying constraints among all possible generations. Wong et al. [7] searched feasible solutions by simulated annealing. Moreover, generating good building layouts was addressed by Bao et al. [8] who encode the spaces and transitions of good layouts in a portal graph and allow the user to explore the plausible layouts. Recently, Liu et al. [9] proposed an exploration method for the interior layouts of precast concrete-based buildings. Moreover, it is not only important to generate a model of the world but to control its details. Vanegas et al. [10] provided a mechanism to interactively edit an urban model using inverse modeling to vary and control the parameters during the modeling process. Additionally, good considerations of deformation analysis and detection should be addressed as investigated by AlHalawani et al. [11].

Fig. 2. Layout (a) shows an example of the standard layouts with parallel aisles [12]. Layouts (b) and (c) show the Flying-V and Fishbone designs [5].

Layout analysis. Recently, the analysis and optimization of warehouses design have been addressed as in Meller and Gau [13], Meller [14] and Tompkins et al [15]. Hall [16] analyzed different routing strategies and their impact on order picking efficiency. Similarly, Petersen [17] and [18] studied the impact of different routing policies on the layout by means of simulation.

Analyzing a layout requires a set of features for the evaluation as was addressed by Alhalawani et al. [19] who proposed a set of topological and geometric features to describe the functionality of a street network. Moreover, Gallager [20] proposed a routing algorithm that achieves the total minimum delay in the network. Aslam et al. [21] learn a congestion model based on real data and develop a congestion aware traffic planning system. When shortest paths are congested, passing through them can deteriorate the network efficiency considerably. In Ebrahimi [22], they present an adaptive routing algorithm for on-chip networks that selects a less congested path from a wide range of alternative paths between each pair of source and destination switches. We use a similar concept

to evaluate the reliability of traffic within a warehouse by considering possible redundant routes between the I/O point and the pick locations.

3 Warehouse Layout Structure

Typically, order pickers drive through warehouse aisles to retrieve products from their storage locations. Figure 3 (left) shows various aspects of the layout of an order picking area. There are several pick aisles that have racks on both sides in which to store products. Changing from one pick aisle to another is possible through the cross aisles, which are perpendicular to the pick aisles. These cross aisles do not contain pick locations. Adding more cross aisles increases the number of possible routes within a warehouse. The main advantage of having extra cross aisles in a warehouse is the increased number of routing options, which may result in lower travel distances [23].

Fig. 3. The left figure gives a top view of a typical order picking area in a warehouse. There are 5 pick aisles and 6 cross aisles. The right figure shows H possible redundant routes between the I/O point and a selected storage shelf s_i (in red). The blue route represents the shortest route.

Each warehouse layout is composed of a set of aisles and a set of storage shelves. To define the aisles, we obtain a set of nodes (i.e., intersections) $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ described by their 2D locations, together with the individual aisles connections $\{e_1, e_2, \dots, e_M\}$. Based on the node set and the connectivity of individual aisles, we construct a graph $G = \{V, E\}$ for the entire warehouse layout. The vertex set is defined as $V := \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ and the edge set as $E := \{e_{ij}\}$ where $e_{ij} = \overline{\mathbf{v}_i \mathbf{v}_j}$ denotes an aisle segment.

On each side of a picking aisle, there are storage shelves. We assume to have only one rack of shelves. The set of storage shelves is defined as $S := \{s_1, s_2, \dots, s_L\}$ where each node is assigned a type $t \in T$ where T is the set of item types to be stored in the warehouse. Moreover, based on the assigned type, it is associated with a demand $d \in D$ to approximate the importance of the item stored in the location. We assume to have three types; the most, medium and lowest demanded items. In order to access the storage shelves, each node is projected to the nearest aisle from the set E . We also consider a layout that has one input/output point (herein after I/O point) for the loading and unloading of items. We take this point as the bottom left corner in our examples.

4 Methodology

In this section, we describe the analysis of a warehouse in order to find the best items allocation that enhances the traffic flow within a warehouse. Since there are items that are more vital to the business than others, it is common and intuitive to place the most important items closer to the I/O point. However, this will lead to more congestion within the warehouse as more items are concentrated in one region. Therefore, we propose an efficient method that improves warehouse reliability by specifying storage locations and allocating their types.

4.1 Warehouse Flow Analysis

Given a warehouse layout with the empty shelves, we start by randomly assigning item types to some of the storage shelves based on a predefined desired quantity of each type. Then, we enumerate H possible redundant routes from the I/O point to each of the storage locations which are within twice the shortest distance. We also mark the shortest route from the I/O point to the storage location. Therefore, we define the set $R := \{R_1, R_2, \dots, R_L\}$ where R_i defines H possible redundant routes for each storage location $\mathbf{s}_i$ such that $R_i := \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_H\}$ where $\mathbf{r}_j$ is a single route composed of a sequence of edges that belong to the aisles edges E . Figure 3 (right) shows a set of possible redundant routes between the I/O point and a selected storage shelf.

Our goal is to place the items such that their flow has two features: (a) items can be accessed and delivered quickly relative to their importance, and to have (b) reliable traffic within the warehouse with the minimum congestion. In order to measure these features, we compute the following terms.

Processing speed term. This metric measures how fast it is to reach items with respect to their demand rate. It favors the most demanded items to be closer to the I/O point. We have the storage shelves, their assigned types t and the demand rate for each type d . We compute the processing speed energy term as the sum product of the demand value for type t_j and the average shortest distance to reach all storage shelves of type t_j which is expressed as follows:

$$E_{processing} = \sum_{j; t_j \in T} d_j * \frac{\sum_{i; t_i = t_j} \text{shortest distance}_i}{n \text{ elements of type } t_j} \quad (1)$$

where d_j is the demand rate for type t_j .

Dispersion term. In addition to aiming to have the most important items closer to the I/O point, we also want to keep items of the same type close to each other. Dispersion term ensures that the items of the same type are distributed in the same region which is more convenient and has the advantage of faster access for requests with large quantities. In order to achieve this feature, we compute dispersion term based on the distribution of the items of each type as follows:

$$E_{dispersion} = \frac{\sum_{j=1}^n \text{types } \frac{1}{n} \sum_{i=1}^n \text{elements of type } t_j ||z_i - c_j||}{n \text{ types}} \quad (2)$$

where c_j is the centroid of z_i points computed for each type t_j and $\|z_i - c_j\| = \sqrt{(x_i - c_1)^2 + (y_i - c_2)^2}$

Congestion term. We use this term to ensure having the minimum congestion within the warehouse. In order to evaluate the reliability of traffic, we consider the layout graph as a compound system of parallel and serial components as described in [24]. Each individual route along the aisles between the I/O point and the storage shelf s_i is composed of a series of edges while the multiple redundant routes are the parallel components. First, we start by computing edge probability for each edge $e_k \in E$ in the graph which is based on the number of times an edge is to be used by all redundant paths in R denoted as K such that:

$$p_edge_k = \frac{1}{K} \quad (3)$$

Then, using the series component probability of each aisle being composed of many edges connected in series, we find the probability of each redundant route:

$$p_route_j = \prod_{k \in r_j} p_k \quad (4)$$

By considering the redundant routes $R_i := \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_H\}$ for each storage shelf s_i as a parallel system, we can find congestion rate as follows:

$$E_{congestion} = 1 - \prod_{r_j \in R_i} (1 - p_route_j) \quad (5)$$

4.2 Storage Assignment Optimization

We aim to have an improved traffic flow within a warehouse by efficiently allocating the items in their storage places. We use the following energy to evaluate the traffic flow efficiency of a warehouse as follows:

$$E = \lambda(E_{processing} + E_{dispersion}) + (1 - \lambda)E_{congestion} \quad (6)$$

where λ determines the relative contributions among the terms.

In order to find the best items allocation, we minimize the energy given in equation 6 using a simulated annealing based sampling (SA) [25]. Initially, we start with random allocation and random type assignment. We also set $E \leftarrow \infty$ and $T = 500$. In each SA step, we randomly select two items and swap their types. We accept the new solution with energy $E(S)$ if $E_{new} \leq E(S)$; else we accept the new solution with probability of $\exp(-(E_{new} - E(S))/t)$, where t is the temperature; else we reject the new layout and retain the old one. If we accept a solution, we set $E \leftarrow E_{new}$. In the annealing schedule, we reduce temperature t and continue with the iterations. We stop if either the maximum number of steps (500 – 1000 in our tests) has been reached, or when $E < threshold$. An example of the results with the intermediate steps is shown in figure 1.

Table 1. A comparison between the distances to reach different types and the processing speed metric for various storage assignment policies (see Figure 4 for a visualization of the assignments). The average distance to reach the most demanded items of type $t = 1$ (shown in red in Figure 1) should be the least distance. Clearly, class-based and our assignment has the least distance to reach items of type $t = 1$. Our proposed assignment produces the lowest score which implies the fastest processing.

Assignment type	Random	Class-based	Opposite class-based	Our assignment
average distance for $t = 1$	56.25	40.9	78.6	40.6
average distance for $t = 2$	74.7	66.4	54.7	54.6
average distance for $t = 3$	71.4	104.6	29.4	86.5
processing speed metric	173.6	150.2	204.3	132.5

5 Evaluation and Applications

We evaluated our method on different sizes and layouts of warehouses. The results show that our framework is simple and yet effective for the analysis of warehouse layouts. It can be used to efficiently generate storage assignment which maximizes the reliability and productivity of a warehouse.

Processing speed analysis. Our processing speed term decreases as the most demanded items are nearer to the I/O point while the least demanded items are further which implies a faster order picking process. Therefore, it is useful to use this term to evaluate the processing speed for a given storage assignment policy. Figure 4 shows different storage assignments used in our evaluation and table 1 shows the values to compare the efficiency of these assignments.

Fig. 4. Different storage assignment policies used to allocate items in a 8x8 warehouse. Our assignment optimizes the allocation of items to achieve the fastest processing speed. As shown in table 1, our assignment has the fastest processing speed value.

Storage assignment analysis. We compare our storage assignment with some of the well known policies. The results in figure 5 show that our assignment reduces congestion level in a warehouse. The random and class-based storage have comparable congestion rates as they both depend on the randomness in their assignment without considering the traffic flow within a warehouse.

Traffic flow analysis. Our framework is able to analyze congestion levels based on the value computed using equation 3. In our simulator, we render the aisles with different colors based on the link probability at each aisle edge. Figure

Fig. 5. An example showing a comparison between the two standard storage methods and our proposed method. We use 7x13 warehouse structure.

6 shows that our storage assignment reduces the number of congested aisle edges as layout (c) has the least number of red edges. Moreover, the overall congestion score is lower than the two common storage assignments.

Fig. 6. Visualization of the traffic flow for different storage assignment policies. Red, green and yellow edges denote high, medium and low congestion levels, respectively. Our result (layout c) has the least number of red (most congested) edges.

Cross aisles evaluation. One factor affecting the efficiency of processes in a warehouse is the number of cross aisles in its layout structure. Adding more cross aisles increases the number of possible different routes and reduces the probability of congestion in the aisles which can be evaluated using equation 5. Figure 7 shows a demonstration of these results.

Fig. 7. Evaluating congestion score for a warehouse (a) with 5 cross aisles and a warehouse (b) with 9 cross aisles shows the decrease in congestion after adding more aisles.

Congestion-aware layout synthesis. We demonstrated earlier that congestion level decreases as we add more cross aisles. Therefore, we use our proposed congestion score to edit a layout and improve its reliability. We start from a layout that has many picking aisles and only cross aisles at the borders. The items are assigned to their storage positions previously using our method. Then,

we randomly add edges which represent parts of cross aisles. We evaluate congestion rate and minimize it to find the best layout which increases the warehouse reliability. Figure 8 shows the result of this synthesis.

Fig. 8. Starting from the (left) layout with a congestion score equals to 0.7092 and randomly adding cross aisles edges to reduce congestion score to 0.3139 (layout on the right). Original aisles are shown in orange while the added ones are shown in red.

6 Conclusion

We presented an algorithm for warehouse storage assignment policy that aims to assign the items to their storage locations while lowering congestion of moving vehicles within a warehouse. We evaluated our framework on many warehouse layouts to generate a storage assignment that minimizes congestion, while current storage assignment methods do not consider congestion in their policies.

The proposed method degenerates to a class-based storage assignment when we have a full warehouse without any empty storage locations since the reliability score will be the same in every optimization step. In the future, we would like to investigate coupling reliability estimate with the demand value to overcome this issue. Moreover, we do not consider picking multiple items in a single trip which has a great impact on the warehouse performance which we plan to investigate later. The proposed scores can be used to evaluate different warehouse layout structures as well as varying many warehouse parameters such as the number of cross aisles or the placement of the I/O point. This will lead to novel warehouse layout structure synthesis possibilities.

Acknowledgments.

We thank the anonymous reviewers for their useful suggestions and Dong-Ming Yan for his valuable assistance in preparing the simulation framework. This work was partly supported by an Anita Borg Google PhD scholarship award.

References

1. Smelik, R.M., Tutenel, T., Bidarra, R., Benes, B.: A survey on procedural modelling for virtual worlds. *Computer Graphics Forum* (2014)
2. Warren Hausman, L.B.S., Graves, S.C.: Optimal storage assignment in automatic warehouse systems. *Management Science* (1976)
3. Merrell, P., Schkufza, E., Li, Z., Agrawala, M., Koltun, V.: Interactive furniture layout using interior design guidelines. *ACM Trans. Graph.* **30** (2011) 87:1–87:10

4. Yu, L.F., Yeung, S.K., Tang, C.K., Terzopoulos, D., Chan, T.F., Osher, S.J.: Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph.* **30** (2011) 86:1–86:12
5. Gue, K.R., Meller, R.D.: Aisle configurations for unit-load warehouses. *IIE Transactions* **41** (2009) 171–182
6. Galle, P.: An algorithm for exhaustive generation of building floor plans. *Commun. ACM* **24** (1981) 813–825
7. Wong, D.F., Liu, C.L.: A new algorithm for floorplan design. In: *Proceedings of the 23rd ACM/IEEE Design Automation Conference. DAC '86*, Piscataway, NJ, USA, IEEE Press (1986) 101–107
8. Bao, F., Yan, D.M., Mitra, N.J., Wonka, P.: Generating and exploring good building layouts. *ACM Transactions on Graphics* **32** (2013) 1
9. Liu, H., Yang, Y.L., AlHalawani, S., Mitra, N.J.: Constraint-aware interior layout exploration for precast concrete-based buildings. *Visual Computer (CGI Special Issue)* (2013)
10. Vanegas, C.A., Garcia-Dorado, I., Aliaga, D.G., Benes, B., Waddell, P.: Inverse design of urban procedural models. *ACM Trans. Graph.* **31** (2012) 168:1–168:11
11. AlHalawani, S., Yang, Y.L., Liu, H., Mitra, N.J.: Interactive facades: Analysis and synthesis of semi-regular facades. *Computer Graphics Forum (Eurographics)* **32** (2013) 215–224
12. Tsige, M.T.: Improving order-picking efficiency via storage assignment strategies. (2013)
13. Meller, R.D., Gau, K.Y.: The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems* **15** (1996) 351 – 366
14. MELLER, R.: Optimal order-to-lane assignments in an order accumulation/sortation system. *IIE Transactions* **29** (1997) 293–301
15. Tanchoco, J.A.T.J.A.W.Y.A.B.J.M.A.: *Facilities Planning*. (2002)
16. HALL, R.W.: Distance approximations for routing manual pickers in a warehouse. *IIE Transactions* **25** (1993) 76–87
17. Petersen, C.G.: An evaluation of order picking routeing policies. *International Journal of Operations & Production Management* **17** (1997) 1098–1111
18. Petersen, C.G.: The impact of routing and storage policies on warehouse efficiency. *International Journal of Operations & Production Management* **19** (1999) 1053–1064
19. AlHalawani, S., Yang, Y.L., Wonka, P., Mitra, N.J.: What Makes London Work Like London? *Computer Graphics Forum* **33** (2014) 157–165
20. Gallager, R.: A minimum delay routing algorithm using distributed computation. *Communications, IEEE Transactions on* **25** (1977) 73–85
21. Aslam, J., Lim, S., Rus, D.: Congestion-aware traffic routing system using sensor data. 2012 15th International IEEE Conference on Intelligent Transportation Systems (2012) 1006–1013
22. Ebrahimi, M., Daneshtalab, M., Farahnakian, F., Plosila, J., Liljeberg, P., Palesi, M., Tenhunen, H.: Haraq: Congestion-aware learning model for highly adaptive routing algorithm in on-chip networks. *Proceedings of the 2012 6th IEEE/ACM International Symposium on Networks-on-Chip, NoCS 2012* (2012) 19–26
23. VAUGHAN, T.S., PETERSEN, G., C.: The effect of warehouse cross aisles on order picking efficiency. (1999)
24. Myers, A.: *Complex System Reliability*. 2 edn. 1614-7839. Springer-Verlag London (2010)
25. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* (1983) 617–680