A Multi-Agent-Based Approach for Furniture Arrangement

Satoshi Ono Takayuki Oshige Shigeru Nakayama

Department of Information Science and Biomedical Engineering,
Graduate School of Science and Engineering, Kagoshima University
1-21-40, Korimoto, Kagoshima, 890-0065 Japan
ono@ibe.kagoshima-u.ac.jp, k6180136@kadai.jp, shignaka@ibe.kagoshima-u.ac.jp

Abstract

This paper proposes a furniture arrangement method based on a multi-agent approach for interior coordination. In the proposed model, each furniture item acts as an agent, interacts with an environment and other agents, and moves to where it wants to go. Consequently, all furniture items reach well-coordinated placement. Agent movement of the proposed method is inspired by particle movement in particle swarm optimization algorithms, that is, agent's velocity is calculated from linear summation of vectors to avoid constraint violation, to harmonize with other agents, and so on. A simple example shows that the proposed method can make well-coordinated furniture arrangement from randomized positions.

1 Introduction

Interior coordination is a task involving the careful selection and placement of materials, fittings, furniture, and furnishings. Most of Japanese apartments and houses are sold or lent without any furniture and furnishings. Inhabitants must therefore buy furniture items such as a dining table, chairs, sofas, desks, beds, living boards, refrigerator, and so on, and arrange their placements whether they like it or not.

For those who wish to coordinate their privately-owned houses and apartments to the highest and most sophisticated standards it will most-likely involve the employment of an interior design specialist with whom a great deal of time is spent in communicating personal preferences, taste and style in order for the designer to be able to make and offer their best selections. With the rise in peoples purchasing wealth and expectations there is a greater demand for coordinated interior designing. To reduce the cost of design and enable more people to enjoy the benefits of personalized custom interior coordination, customers will consider not only the enlistment of an expensive interior designer but another reasonable independent way which can be done if the customers supply their own room requirements and user computer systems to provide the design coordination for them.

In this paper we focus on furniture placement of interior coordination, and propose a furniture arrangement method based on multi-agent system (MAS)[1, 2, 3]. An agent corresponds to a furniture item, sees the whole room, and decides which direction to go; consequently each furniture item thinks and moves autonomously. Agents negotiate with one another when a conflict occurs between them. Well-coordinated furniture arrangement is conducted by agents (furniture items) moving to their favorite positions.

We also develop a system for a non-design-professional resident to coordinated furniture arrangement. The system uses three dimensional graphics of furniture items which are sold on the actual market, so a user can recognize outputs of the proposed method at a glance and can judge whether the derived furniture arrangement is in harmony and functionable.

2 The proposed algorithm

2.1 Overview

The proposed model is a kind of homogeneous communicating multi-agent system[2, 3]. An agent corresponds to a furniture item, and an environment corresponds to a room involving floor, wall, ceiling, and fittings such as doors and windows. The agent knows its and other all agents' positions in the room as if using a camera mounted on the ceiling, investigates whether it violates constraints, and autonomously decides which direction to move. This model is therefore similar to Small Size robot League, one of the RoboCup Soccer League divisions [4].

Each agent moves to satisfy constraints concerning with itself only, and negotiates with other agent when they compete in a violation. As a consequence

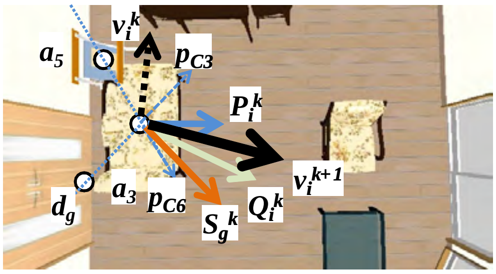

Figure 1: Example of velocity calculation.

of agents' independent-minded movements, the furniture items reached to well-coordinated arrangement.

At this stage, the proposed model allows furniture items to overlap during the process of arrangement, although it's impossible for real furniture to overlap each other in an the real world. Real furniture items cannot move along agents trail obtained from the proposed model's simulation, but the resulting furniture arrangement is valuable even in the real world.

2.2 Agent movement

Planning method in each agent is inspired from Particle Swarm Optimization[5, 6]. Furniture item's position and velocity are basically updated by simple equations involving linear summation of elemental vectors by the following equations:

$$\boldsymbol{v}_i^{k+1} = w\boldsymbol{v}_i^k + \boldsymbol{P}_i^k + \boldsymbol{Q}_i^k + \boldsymbol{S}_g^k \tag{1}$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1} \tag{2}$$

$\boldsymbol{v}_i^k$ and $\boldsymbol{x}_i^k$ are velocity and the center position of furniture (agent) i at step k. $\boldsymbol{P}_i^k$, $\boldsymbol{Q}_i^k$, and $\boldsymbol{S}_g^k$ are vectors calculated from violation resolution rules, negotiation results, and group coordination. w is an inertia weight.

The process flow is outlined as follows:

[Step 1] The agent checks all constraints as described in Section 2.3.

[Step 2] The agent looks for a vector $\boldsymbol{P}_i^k$ indicating a direction to resolve violations based on resolution rules corresponding to constraints.

$$\boldsymbol{P}_i^k = \sum_{m=1}^{N_i^V} \alpha_m c_m^P r_m^P \boldsymbol{p}_m^k \tag{3}$$

N_i^V is the number of violations agent i commits, $\boldsymbol{p}_m^k$ is a vector calculated from a rule to resolve a violation m, c_m^P is a weight parameter, and r_m^P is a random real number from 0 through 1. α_m is 1 when violation m is of a unary constraint, or when violation m is of a binary constraint and agent i loses precedence by negotiation. Otherwise α_m is 0. When violating a binary constraint, $\boldsymbol{p}_m^k$ is calculated after negotiation in **step 3**.

[Step 3] The agent negotiates with other agents when they are competing, i.e. violating a binary constraint. Negotiation is conducted by comparing agents' priority. The agent with lower priority level must break away from the agent with higher priority.

[Step 4] The agent calculates $\boldsymbol{Q}_i^k$ to coordinate other agents in the same group by the following equation:

$$\boldsymbol{Q}_i^k = \sum_{j \in N_g} \beta_{i,j} c_j^Q r_j^Q \boldsymbol{q}_{i,j}^k \tag{4}$$

$\boldsymbol{q}_{i,j}^k$ is calculated by two ways; if agent i and j are close and their distance is longer than the threshold T_l, then $\boldsymbol{q}_{i,j}^k = \boldsymbol{x}_j^k - \boldsymbol{x}_i^k$, or $\boldsymbol{q}_{i,j}^k$ turns into one of vectors to align themselves, side by side, face to face, L-shape, and so on. c_j^Q is a weight parameter, and r_j^Q is a random real number from 0 through 1. S_g is a set of agents in group g to which agent i belongs, and $\beta_{i,j}$ is 1 when the distance between agents i and j is less than threshold T_c, otherwise $\beta_{i,j} = 0$.

The agent also figures on a relationship between groups, and moves to keep adequate distance by calculating the following vector $\boldsymbol{S}_g^k$:

$$\boldsymbol{S}_g^k = \sum_h \gamma_h c_h^S r_h^S \boldsymbol{s}_{g,h}^k \tag{5}$$

γ_h is 1 when the distance between the center positions of group g (to which agent i belongs) and group h is less than the threshold T_g. $\boldsymbol{s}_{g,h}^k$ is calculated from center positions $\boldsymbol{g}_g^k$ and $\boldsymbol{g}_h^k$ of group g and h,

$$\boldsymbol{s}_{g,h}^k = \boldsymbol{g}_g^k - \boldsymbol{g}_h^k \tag{6}$$

c_h^S is a weight parameter, and r_h^S is a random real number from 0 through 1.

[Step 5] The agent calculates its velocity by equation (1). Inertia weight w is set to higher value when the simulation starts, and decreases at a fixed rate.

[Step 6] The agent moves to its new positions by equation (2).

Fig. 1 shows an example of velocity calculation of agent a_3. a_3 is in front of a door and overlap with

Table 1: Constraints and resolution rules.

Constraint no.	Constraint	Type	Rule no.	Rule	Priority weight o_m
C_1	A furniture item must be in a room.	Unary	R_1	Move toward the center of the room.	$2(n-1)$
C_2	At least 800mm width space must be free in front of functional faces of a furniture	Unary	R_2	Move toward the room's center.	—
C_3	No furniture item must be placed in front of a door.	Unary	R_3	leave the door's front.	$1.5(n-1)$
C_4	No furniture item must be placed in front of a window.	Unary	R_4	leave the window's front.	$1.5(n-1)$
C_5	A furniture item whose hight exceeds 1,000mm must be placed with its back against a wall.	Unary	R_5	Move toward the wall behind the item.	$n-1$
C_6	Furniture items must not overlap each other.	Binary	R_6	Negotiate with each other	—
C_7	The same furniture items next to each other must look to the same direction.	Binary	R_7	Negotiate with each other.	—
C_8	At least 800mm width space must be free in front of functional faces of a furniture.	Binary	R_8	Negotiate with each other.	—
C_9	A table must not stand behind a sofa.	Binary	R_9	Negotiate with each other.	—

Figure 2: Screenshot of the implemented system.

other agent a_5, and violates C_3 and C_6. Vector $\boldsymbol{p}^k_{C_3}$ to resolve the former violation is $(\boldsymbol{x}_{\boldsymbol{a_3}} - \boldsymbol{d}^g)$ where $\boldsymbol{d}^g$ indicates the center position of the door. Assuming that agents a_3 and a_5 have the priority level of 2 and 9, agent a_3 must depart from agent a_5, and $\boldsymbol{p}^k_{C_6}$ is calculated as $(\boldsymbol{x}_{\boldsymbol{a_3}} - \boldsymbol{x}_{\boldsymbol{a_5}})$. $\boldsymbol{P}^k_i$ is therefore calculated by adding $\boldsymbol{p}_{C_3}$ and $\boldsymbol{p}_{C_6}$. Because there are some furniture items belonging to the same group, vector $\boldsymbol{Q}^k_i$ is calculated to come close other sofa and low table. Dining table and chairs belongs to other group, Vector $\boldsymbol{S}^k_g$ therefore arises to step away. Finally, vector $\boldsymbol{V}^{k+1}_i$ is calculated by adding vectors $\boldsymbol{P}^k_i$, $\boldsymbol{Q}^k_i$, and $\boldsymbol{S}^k_g$.

2.3 Constraints

Constraints are provisions for a feasible furniture arrangement[7, 8]. Unary constraint is a condition between an agent and the environment, and binary one is a condition between agents. The proposed model involves nine constraints as shown in Table 1, and a simulation stops when all constraints are satisfied.

2.4 negotiation and priority

Negotiation is conducted by comparing furniture items' priority; the item having lower priority must deviate from the item having higher priority. Priority pri^k_i of agent i is calculated by the following equation.

$$pri^k_i = pri^0_i + \sum_m o_m + u_i + r^{pri}_i \tag{7}$$

pri^0_i is an initial priority level defined by its furniture items static attributes; taller and larger furniture items have higher priority level. o_m is a dynamic element calculated by constraint violation m as shown in Table 1, and u_i is another additional element by user operation, respectively. r^{pri}_i is a random number.

2.5 Furniture group

In the proposed method, furniture items comprise a group based on their functional height. Functional height means the height at which inhabitants use the furniture items; they use the item with standing or sitting on a floor, a low chair like a sofa, or a high chair like a dining chair.

H_{floor}: Furnitures such as floor cushions and low tables have this function height H_{floor}. Inhabitants use the furnitures with sitting on the floor.

H^{low}_{seat}: Furnitures such as sofas, low tables, and television boards have this function height H^{low}_{seat}. Inhabitants use them with sitting on sofas.

H^{high}_{seat}: Furnitures such as chairs, dining tables and desks have this function height H^{high}_{seat}. Inhabitants use them with sitting on chairs.

(a) Step 1. (b) Step 500. (c) Step 1,000. (d) Step 2,000. (e) Step 3,055.

Figure 3: Example search progress.

H_{stand}: Furnitures such as dining boards have this function height H_{stand}. Inhabitants use them with standing.

Furniture items with the same functional height gather in a group.

3 Output examples

Fig. 2 shows a screenshot of the resulting system consisting of three-dimensional viewer, two-dimensional floor plan viewer, and agent status window. Fig. 3 shows a sample transition to coordinate a furniture arrangement with a room involving an irregular rectangle floor shape and windows on South and East sides. An item circumscribed with a translucent rectangle violates one or more than one constraints.

At the earlier stage of the simulation, furniture items such as sofas and a center table belonging the same group roughly moves to cluster together. And all items drift from place to place till about step 2,000. As the simulation progresses, the group in which items have the functional height H_{seat}^{low} moves nearby the windows, and the dining table and chairs move to the back of the room.

4 Conclusions

Proposed in this paper is a multi-agent-based model for furniture arrangement, in which its agent movements are inspired by particle swarm optimization.

It is our future work to make a model in which furniture items' collision is prohibited. The improved model will allow to utilizing not only derived furniture positions but furniture trails, which is a plan to change the furniture arrangement[9].

Acknowledgements

We would like to thank TIS Inc. for supporting this research.

References

[1] E. H. Durfee and J. S. Rosenschein, "Distributed problem solving and multiagent systems: Comparisons and examples," *Proc. Int'l Workshop DAI*, 1994, pp. 94–104.

[2] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[3] P. Stone and M. M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[4] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," *Pro. IJCAI-95 Workshop on Entertainment and AI/ALife*, 1995.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int'l Conf. Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[6] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[7] S. Ono, T. Izumi, A. Fujiyama, C. J. Ashley, and S. Nakayama, "Interior coordination using case-based reasoning and constraint satisfaction paradigm," *Proc. IEEE Conf. Cybernetics and Intelligent Systems*, 2004, pp. 1066–1071.

[8] T. Izumi, S. Ono, and S. Nakayama, "Case- and constraint-based interior design system," *Systems, Control and Information*, vol. 19, no. 7, pp. 293–295, 2006.

[9] S. Ono, Y. Hamada, S. Yamamoto, K. Mizuno, and S. Nishihara, "Rearrangement of floor layouts based on constraint satisfaction," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics 2001*, 2001, pp. 2759–2764.