Pattern Recognition Letters 44 (2014) 88–97

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Crowd-Z: The user-friendly framework for crowd simulation on an architectural floor plan

Machi Zawidzki [a,*], Mohcine Chraibi [b], Katsuhiro Nishinari [c]

[a] Department of Architecture, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[b] Jülich Supercomputing Centre, Wilhelm-Johnen-Straße, 52428 Jülich, Germany
[c] Research Center for Advanced Science and Technology, The University of Tokyo, Komaba 4-6-1, Meguro-ku, Tokyo, 153-8904, Japan

A R T I C L E I N F O

Article history:
Available online 1 November 2013

Communicated by Rita Cucchiara

Keywords:
Pedestrian dynamics
Agent based modeling
Design support
Digitized floor plan

A B S T R A C T

This paper introduces Crowd-Z (CZ): a framework that provides a user-friendly platform where architects can perform simple crowd simulations on floor plans. A simple but robust and flexible agent-based system is used for modeling of the crowd dynamics. Such simulations can be performed at any stage of design – from rough sketches to the final blueprints. CZ allows acquiring the layouts for the simulations in a number of ways: freehand sketches, importing already prepared images and appropriating preprocessed images from commercially available Computer Aided Design programs. These three methods are illustrated with practical examples, followed by a number of simulations compared with the literature or other commercially available programs.

1. Introduction

In the past, several models for pedestrian dynamics were developed and used for various purposes, e.g. planning of buildings, organization of mass events, etc. In general, these models can be classified in two major categories: space continuous models, which are based on the Newtonian approach (Hirai and Tirui, 1977; Helbing and Molnár, 1995; Yu et al., 2005; Chraibi et al., 2011) and rule-based models discrete in space that describe the dynamics of crowd on a regular grid by means of transition probabilities (Gipps, 1986; Blue and Adler, 1999; Rajewsky et al., 1998; Kretz and Schreckenberg, 2006). Particularly relevant to the concept presented in this work are agent-based systems (Kerridge et al., 2001; Bandini and Vizzari, 2007; Shi et al., 2009).

Most of these models describe rather well pedestrian dynamics quantitatively and qualitatively. Based on these models, several commercial tools are available, for example: VISSIM (Kretz et al., 2008), ASERI, (Schneider and Könnecke, 2002), PedGo (Klüpfel et al., 2002). However, in the common architectural and urban planning practice they are rather rarely used. One of the reasons is that designers often do not feel necessity for such experimentation, but also they may get discouraged by the price, complexity or "unfriendliness" of the available software. The motivation for this work is to introduce an intuitive environment for designers, in which they can comfortably perform crowd simulations (CSs). The holistic approach presented here also contributes to the academia so the researchers can have an overview of what designers expect from such tools and models. For the same reason – it is also addressed to the software engineering community. Crowd-Z (CZ): the user-friendly framework brings the idea of CS to broader audience than before, and hopefully will encourage the architects and planners to use it for designing safer and more convenient spaces. The main objectives of CZ are:

1. To simulate the crowd dynamics (CD) on a given floor plan, in particular – to find the problematic areas where the traffic does not flow properly or conversely, the areas that are visited too sparsely.
2. To allow for experimentation by alterations to the geometry of the environment, in particular by introducing or erasing elements or their parts, changing the locations of the start terminals, exits, etc., and to immediately observe the influence of these changes on the overall CD.
3. To study the emergent behavior of agents determined by different behavioral scenarios.

The next section briefly describes the model used for CSs, followed by three practical applications of CZ in architectural design and a number of simulations validating the CZ model.

2. Crowd dynamics component

This work focuses on the geometrical processing rather than the mathematical modeling of pedestrian dynamics. Although modern models produce more realistic results in terms of CD, for testing

* Corresponding author. Tel./fax: +81 3 5452 5286.
E-mail address: zawidzki@gmail.com (M. Zawidzki).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.10.025

purposes a very simple and robust model is introduced. Since the crowd dynamics component (CDC) is an autonomous module of CZ, it can be easily equipped with more sophisticated models. The one presented here contains only a few parameters, which makes it intuitive and easy to understand, particularly for the users who are not familiar with crowd modeling.

Various strategies describing how pedestrians move towards exits are investigated in Kretz (2009) and Kretz et al. (2011). In the presented work a simple strategy based on the shortest path is used. It is defined at each cell of the environment by the smallest distance to the given exits. Hereby, the distance is defined by taxicab metric (Willard, 1970). The exits and walls are assigned the values of zero and infinity, respectively. A matrix containing this information for entire environment is called distance field (*DF*). In CZ, *DF* is static and does not depend on the dynamics of the agents. For a discussion on the influence of the definition of DF on CD see Kretz et al., 2008. The value of DF_{ij} is proportional to the distance of from a cell (i,j) to the given exit. The definition of the distance is determined by the type of the neighborhood (e.g.: Moore, von Neumann).

In cases with multiple exits, *DFs* are calculated for each exit k and combined into DF^{*}_{ij} in a way so that for each cell (i,j) the lowest value in the corresponding position in all *DFs* is selected:

$$DF^{*}_{ij} = \min_{k=1,\ldots,n} DF^{k}_{ij}, \tag{1}$$

with n the number of exits.

As a result, each agent proceeds to the closest exit. The movement of agents is determined entirely by the local neighborhood defined by *DF*, and can be expressed as the following rules:

1. If all of the neighboring cells are occupied, either by other agents or walls – wait.
2. Check the values of *DF* in the available neighboring cells and act according to the "perkiness".

The "perkiness" describes the willingness of an agent to move. In this model there are three possibilities: Lazy, Conservative and Perky. Lazy will only move to a neighboring cell if its value is lower than the current one, Conservative will move also to an equipotent cell and Perky will move to a neighboring cell of the lowest *DF*

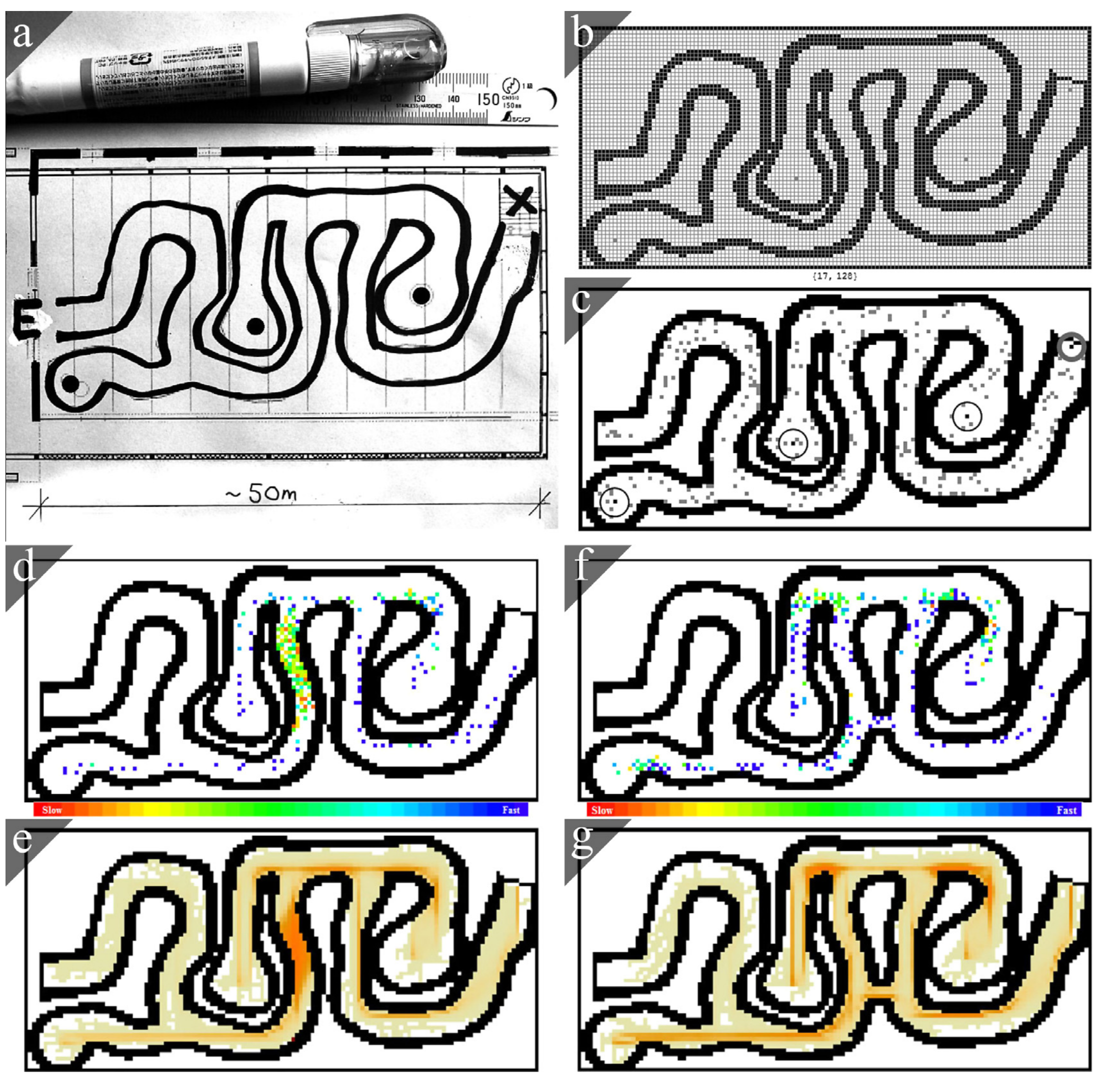

Fig. 1. (a) A free-form exhibition space with three exhibits marked with black dots. The entrance and exit are marked with E and X, respectively. (b) In the CZ interactive panel, clicking with a pointer on a cell reverses its value (wall to void and vice versa). The exhibits and exit are placed manually and shown in gray. The coordinates of the last clicked position are shown below (row from the top, column from the left). (c) 250 randomly distributed agents ready for the simulation are shown in gray. Three exhibits are shown in black and marked with black circles. The exit is marked with a gray circle. (d) A screen-cast of the initial simulation. (e) The corresponding heat map which confirms the congestion area. (f) A screen-cast showing that a small alteration to the layout can substantially improve the agents' flow. (g) The corresponding heat map.