ETHlogo

**Lecture with Computer Exercises:**

**Modelling and Simulating Social Systems with MATLAB**

Project Report

|  |
| --- |
| Crowd Simulation  … |

Samuel Oberholzer & Philipp Lütolf

Zürich

11.05.2014

**IMPORTANT**

**You MUST include the ETH declaration of originality here; it is available for download on the course website or at**

**http://www.ethz.ch/faculty/exams/plagiarism/index\_EN;**

**It can be printed as pdf and should be filled out in handwriting.**

**Agreement for free-download**

We hereby agree to make our source code of this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

|  |  |
| --- | --- |
| Samuel Oberholzer | Philipp Lütolf |

# Table of content

# Abstract

The goal of simulation is to model big crowds in different locations and identify the dangerous spots. Our model is programmed in Matlab, is agent – based and in continuous space. The physics are based on the ‘social force model’ from Helbing. To identify the dangerous spots, the density of the people is computed and illustrated in the simulation. In this report two different locations are considered.

We expect the most critical spots to be obstacles, corners, intersections and other bottlenecks where the crowd is disrupted. Therefore we chose to consider a crossroad setting and a curve. The locations are simulated with and without obstacles to identify the differences. To conclude our model we compared our results to projects from previous semesters which are mainly implemented as cellular models.

# Individual contributions

The work on this project was shared evenly between both authors because we always worked together.

# Introduction and Motivations

Big events with a lot of people bear a great risk of mass panic. At the love parade in Duisburg 2010, 21 people died and 541 people were injured due to a crowd disaster.

One critical factor of such tragedies is the arrangement of the location. We want to model big crowds and identify the dangerous spots in a specified environment.

# Description of the Model

## Social force model

The model is based on the ‘social force model’ developed by D. Helbing and P. Molnar [1] [2]. It describes the “social forces” acting on a pedestrian in a crowd. Each pedestrian in a crowd can be represented with a point in space, the speed of the pedestrians can be described by the equation and the acceleration by .

D. Helbing and P. Molnar added a fluctuation term which include random variations of the behaviour.

One term of describes the driving force which accelerates the pedestrian towards the desired velocity , another term consists of the repulsion force from other pedestrians and obstacles and the last force outlines attractive effects .

## Acceleration force

The driving force accelerates the pedestrian towards the destination which results in the desired direction

Deviations of the actual velocity from the desired velocity occurring from obstacles or other pedestrians are corrected within the “relaxation time” .

The desired speed is defined as

Where is the initial velocity and the maximum desired velocity.

The parameter

Characterizes the nervousness of the pedestrian to reach their destination, where describes the average speed of the pedestrian.

## Pedestrian Interactions

The term describes the repulsive force from another pedestrian . This force is defined as

Where denotes the respective interaction strength and the range of the repulsive interaction. The parameter is the sum of the radii of both pedestrians, is the distance between the centres of mass and is the normalised vector pointing from to .

The last parameter accounts for the directionally dependent behaviour of pedestrians. In the context of crowds the factor gives the pedestrians within sight greater influence than those out of sight.

## Boundary Interactions

# Implementation

The social force model is implemented in Matlab as an agent – based model in continuous space.

FOR every pedestrian in the system do

Calculate desired velocity according to (2.6.6)

Calculate desired destination

Calculate desired direction according to (2.6.5)

Calculate driving force (2.6.4)

FOR every other pedestrian do

Check distance between pedestrians

IF this distance < radius of Verlet-Sphere

Calculate Social Force (2.6.8)

Add Social Force to driving force

FOR each boundary element do

Check distance to element

Calculate the influence of the closest element from (2.6.12)

Add to the Social and driving force

END

Set first 2n vector components as (4.1.2)

Set second 2n vector components as (4.1.3)

## Waypoints

## Walls/Obstacles

## Plots

## Model constants

# Simulation Results and Discussion

# Summary and Outlook

# References

|  |  |
| --- | --- |
| [1] | P. M. D. Helbing, "Social force model for pedestrian dynamics," *Physical Review E,* vol. 51, no. 5, 1995. |