

## Technical Section

Using real life incidents for creating realistic virtual crowds with data-driven emotion contagion <sup>☆</sup>Ahmet Eren Başak<sup>a</sup>, Uğur Güdükbay<sup>a,\*</sup>, Funda Durupınar<sup>b</sup><sup>a</sup> Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey<sup>b</sup> Computational Biology Program, Oregon Health & Science University, Portland, OR 97239, USA

## ARTICLE INFO

## Article history:

Received 25 October 2017

Revised 2 February 2018

Accepted 5 February 2018

Available online 15 February 2018

## Keywords:

Crowd simulation

Emotion contagion

Parameter learning

Data-driven optimization

## ABSTRACT

We propose a data-driven approach for tuning, validating and optimizing crowd simulations by learning parameters from real-life videos. We discuss the common traits of incidents and their video footages suitable for the learning step. We then demonstrate the learning process in three real-life incidents: a bombing attack, a panic situation on the subway and a Black Friday rush. We reanimate the incidents using an existing emotion contagion and crowd simulation framework and optimize the parameters that characterize agent behavior with respect to the data extracted from the video footages of the incidents.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction

Crowd psychology has attracted the attention of scholars for more than a century. In his seminal work, “The Crowd: A Study of the Popular Mind”, Le Bon [1] describes the salient aspects of crowd psychology as impulsiveness, irrationality, emotionality and mental unity. This phenomenon is also known as *collective (mis)behavior*. Social psychology literature introduces various theories to explain the reasons for collective crowd behavior, including social contagion [1,2], predisposition [3–5] and emergent-norms [6] theories. Brown [7] describes an elaborate taxonomy of crowds and classifies crowds under two general categories as *audiences* and *mobs* depending on the existence of observable unified behavior, instead of the reasons bringing crowd members together. In both categories, crowd members share a common goal unlike pedestrians on a street who happen to be coincidentally at the same place at the same time. What distinguishes mobs from audiences is their active and emotional disposition, which leads to “mob”ility. This feature makes mobs more interesting to study (and simulate) as they display more diverse and interesting behaviors than audiences. Therefore, we focus on mob simulations in this work.

One of the most influential factors that causes collective mob behavior is emotion contagion. Emotion contagion is the

phenomenon of having the feelings and responses of one person influencing and manipulating the emotions of others in a group of individuals [8]. Within this continuous feedback mechanism, we generally observe that emotions and resulting behaviors converge to a single active response over time, thus converting audiences to mobs. Because of this feature, systems that model emotion contagion mostly focus on mob behaviors.

We need a universal, objective, quantitative and reusable method for validating crowd simulation models, not just in terms of the steering behaviors of individuals but the authenticity of the group behavior as a whole. We can then formally define future improvements to existing simulation systems and compare different systems under different scenario cases. Crowd simulation literature includes various techniques to evaluate the behavior of virtual agents such as learning parameters from crowd videos [9–11]; determining metrics to compare different simulations [12–14]; and referring to human expert opinions [15]. In this work, we propose a data-driven approach to mimic real crowd behaviors by learning the parameters that affect crowd behavior and to validate crowd simulation systems according to their fidelity to real life behaviors. We apply this approach to the epidemiological emotion contagion framework proposed by Durupınar et al. [16]. We explain how to learn the characteristics of emotion contagion from a real-life event video and how to improve and optimize the emotion contagion model by Durupınar et al. using the results of this analysis. To this end, we investigate the agent behavior before and after the incident and recreate the incident in a virtual environment.

<sup>☆</sup> This article was recommended for publication by Yiorgos L. Chrysanthou.

\* Corresponding author.

E-mail addresses: [eren.basak@bilkent.edu.tr](mailto:eren.basak@bilkent.edu.tr) (A.E. Başak), [gudukbay@cs.bilkent.edu.tr](mailto:gudukbay@cs.bilkent.edu.tr) (U. Güdükbay), [durupina@ohsu.edu](mailto:durupina@ohsu.edu) (F. Durupınar).

The contributions of this paper are as follows:

- We propose a data-driven, quantitative and reproducible pipeline for learning parameters from real crowd videos for synthesizing virtual crowds.
- We explain how real-life incidents can be utilized for evaluation and improvement of crowd simulations.
- We clarify the properties of suitable material for this process and demonstrate how to process videos of real-life incidents for virtual environment creation.
- We analyze three contemporary incidents and apply our proposed approach to an existing emotion contagion and crowd simulation system.

A preliminary version of this research has appeared as a conference paper [17]. Different from [17], which analyzes only one scenario, this extended version includes a comprehensive set of experimental results for three different scenarios. We introduce new error metrics to evaluate the proposed approach and include new figures illustrating our approach and its experimental results, as well as new sets of graphs about the experimental results. We also re-organize and extend the related work to fully cover the state-of-the-art on the subject.

The rest of the paper is organized as follows. In Section 2, we discuss the related work in emotion contagion, crowd simulation and empirical evaluation studies. In Section 3, we provide a brief overview of existing emotion contagion models and Durupinar Emotion Contagion Model that we base our studies on. In Section 4, we explain the proposed parameter learning framework and necessary steps to analyze crowd videos before using them for the optimization process. In Section 5, we explain the incidents that we studied, how we extracted data from them, how we recreated them in a virtual environment and how we simulated them using Durupinar model. In Section 6, we demonstrate and discuss the results of our parameter estimation mechanism on the studied incidents. Finally, we summarize our work in Section 7, draw conclusions and discuss future improvement ideas.

## 2. Related work

We provide a comprehensive review of related work on the simulation of virtual crowds including emotion contagion studies and on the comparison of virtual crowds with real crowds in our previous work [17]. The review refers to various crowd simulation studies that analyze interactions with the environment [18], the influence of architecture on crowd behavior [19], data-driven evaluation of crowds with trajectory extraction [9–11,14] and scoring metrics [12,13], emotion contagion models [16,20–23], the role of appraisal in emotion contagion [15] and how emotion contagion can be used for simulation of emergency situations [24,25]. In addition to these, there are other studies that cover the influence of the environment on the emotions and behavior of crowd members. For instance, Hoogendorn et al. study the information exchange and emotion contagion within crowds [26]. They model the change of information spread with respect to the emotional states of individuals and simulate an emergency situation to demonstrate their work. Borodin et al. [27] and Chen et al. [28] apply the concept of influence among the groups of people to social networks and show that the responses of key individuals steer the behavior of the whole group significantly.

Heterogeneity is an important aspect of realistic crowd simulation that has been studied by many groups. Pereira et al. [29] present a computational model for emotion contagion in virtual crowds, incorporating personality differences and interpersonal relationships. They take intimacy between virtual agents into account for the influence of emotions, where higher intimacy results in more homogeneous emotional behaviors

in crowds. Silverman et al. [30] propose an architecture that combines an existing pathfinding and cognitive navigation system (MACES) with PMFserv, which models the changing behaviors of individuals according to stress, emotions and motivations. Helbing and Molnar demonstrate the social forces model for explaining crowd behavior [31], where the characteristics of individuals in a crowd affect the motion of surrounding pedestrians. In a later study, they model the panic behavior in crowds mixing the individualistic behavior and collective instincts [32]. This study simulates a crowd of people escaping from a smoke-filled room and proposes an optimal strategy for escaping from such disasters.

Evaluation of simulated crowds in terms of their similarity to real world is another challenge that has been extensively studied. Fridman and Kaminka [33] demonstrate a crowd simulation model based on Social Comparison Theory and argue that their model is suitable for general usage. Furthermore, they propose a method for evaluating the imitation performance by showing people video clips of random crowds and as well as simulations, then asking questions to clarify whether they perceived the video as the behavior of unrelated individuals or more like a collective response. Lin et al. [34] model the crowd behavior evacuating an office building. In their case study, using the videos taken by the security cameras, they calibrate the parameters of their model. Similarly, Tan et al. [35] use an agent-based crowd model for simulating an evacuation incident and propose a method for representing indoor space for such simulations.

## 3. Emotion contagion approaches

### 3.1. ASCRIBE

Bosse et al. [24] present ASCRIBE, a computational model of neural mechanisms of social mutual adaptation for satisfactory common group decisions. ASCRIBE incorporates a basis for modeling the interaction between the beliefs and emotions of an agent while also providing mechanisms for the influence of emotions, intentions and beliefs among agents.

In its core, ASCRIBE has a model for agents that mirror the mental states of each other, representing the contagion phenomenon. In this model the amount of influence of a mental state of one agent on another depends on the *expressiveness* of the sender agent, *openness* of the receiver agent and *channel strength* between the subjects, which depends on physical conditions such as distance and field of view. The combination of the influence of all the other agents constitutes the overall contagion strength on an agent. The updated mental state of an agent is calculated as a combination of the overall contagion and the agent's previous state. The coefficient of the contagion component determines the speed of adjustment in an agent's mental state and the convergence of the crowd behavior.

The interaction among emotions, beliefs and intentions of an agent are also incorporated into the ASCRIBE model. In this model, fear starts affecting information retrieval and amplifies the influence of the beliefs on behavior if it is above a threshold. The value given to information by an agent will be affected by the fear and personality as well, e.g., a pessimistic person with high level of fear would be significantly affected by negative information; and positive information would have less influence on the agent's behavior. Similarly, information influences the emotional state. For example, negative information has a tendency to increase fear. Finally, beliefs and emotions together affect the intentions of an agent.

Bosse et al. test ASCRIBE with two scenarios, a synthetic office evacuation scenario which demonstrates the influence of information on agents' behavior, and a reanimation of a real-life incident for demonstrating the model's mimicking potential. The May 4th incident that happened in Dam Square, Amsterdam in 2010

involves a group of about 20000 people leaving the area in panic after a person starts screaming and demonstrates the contagious nature of fear.

### 3.2. ESCAPES

Tsai et al. [23] introduce Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison (ESCAPES), a multi agent evacuation system customized for the airport evacuation domain. ESCAPES includes four key components for the characterization of agent behavior:

- Various types of agents with different roles and priorities, such as travelers, families and security personnel.
- Emotional contagion, which causes an agent's emotion to affect others directly and indirectly. Specifically, passengers adopt the highest level of fear from their surrounding passengers and if there are security personnel nearby, passengers tend to calm down.
- Exchange of information related to the scene and the incident. ESCAPES is geared towards airport evacuations. Airports have people with little knowledge of the building unlike office spaces and houses. This makes the information exchange among passengers and the directives of security personnel play significant roles in the behavior of the crowd.
- Behavioral interaction utilizing Social Comparison Theory [36], which basically involves passengers comparing themselves to other passengers by evaluating their behavior. This evaluation is based on various features such as speed, distance, emotional reactions, and so on. According to this evaluation, agents tend to mimic the behavior of the agents that are similar to themselves, causing convergence in group behavior.

Tsai et al. run proof of concept tests of ESCAPES by modeling the Tom Bradley International Terminal at Los Angeles International Airport. They visualize the terminal in 3D based on their 2D model of the building. Later, they compare Durupinar Emotion Contagion model and ASCRIBE model by implementing both systems as the way emotional exchange takes place [25].

### 3.3. BioCrowds model

Neto et al. [37] introduce the emotion contagion concept into BioCrowds method [38]. Their emotional contagion model is similar to the model described by Bosse et al. [21], incorporating gradual contagion instead of threshold-based binary contagion. The model involves channel strength, which affects the inter-agent emotion influence and is inversely proportional to distance. Emotion contagion in BioCrowds supports multiple groups in a scenario, models inter-group emotion contagion as well as management of multiple emotions. The most intense emotion of an agent represents the emotional state of the agent at that moment. The goal of an agent is tied to the emotional state, which means that when the emotional state changes, the goal changes as well.

### 3.4. Durupinar model

We estimate the parameters of the emotion contagion model proposed by Durupinar et al. [16]. The crowd simulation system incorporating this model represents personality by the OCEAN (Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism) model [39], which describes five independent dimensions of human personality also known as the Big-Five Factor structure [40]. The system defines how each personality trait affects the development of emotions both for the individuals themselves and other people around. By specifying different trait values for each agent, one can generate heterogeneous crowds easily and observe

the change in convergence patterns of crowd behavior with respect to the given personalities.

Alongside the personality traits, agents' appraisal of their environment and surrounding individuals play an important role in the development of emotional reactions. The Durupinar crowd simulation system employs the Ortony, Clore, Collins (OCC) model [41] to simulate cognitive appraisal and emotions. In this model, individuals assess their environment in terms of their goals regarding others' and their own actions, their standards about other people where the actions of others are approved or disapproved and their attitudes towards objects. In order to determine the current emotional state and make decisions, the system utilizes the Pleasure-Arousal-Dominance model [42]. Finally, to model emotion contagion in a crowd, the system adopts the generalized contagion approach proposed by Dodds and Watts [8] with various augmentations.

In the Durupinar model, the emotion of an agent is represented as the combination of the agent's appraised emotional state with respect to the defined goals, standards and attitudes and the influence of other agents in the crowd, which corresponds to the empathy component representing the emotion contagion in the model. The calculation of empathy is straightforward while the appraisal part depends on the nature of the scenario and events.

The Durupinar model simulates emotion contagion by adopting an epidemiological threshold-based approach [8]. This way, the emotions do not show up until they reach a threshold level. Emotions add up when the agents are exposed to incidents and other agents' emotional responses and decay over time when they are not affected. The thresholds of emotional reactions are mainly determined by the *expressiveness* values of agents.

### 3.5. Discussion

Tsai et al. [23] and Bosse et al. [24] discuss the first steps and benefits of quantitative evaluation, comparison and optimization of emotion contagion and crowd simulation models using real world incidents. Yet, they do not discuss the properties of suitable incidents and materials; methods to track individuals on video footage and to project the tracking data to the 3D scene in detail. For instance, Bosse et al. track a relatively small group of 35 individuals in a massive crowd of size about 20000 people and use only this incident for parameter estimation.

Although these models seemingly serve the same purpose, they have differences in terms of their application targets. For example, ESCAPES has a very specific focus on airport evacuation scenarios; ASCRIBE and BioCrowd demonstrate other kinds of scenarios which involve fear, even though they are capable of being used in different kinds of scenarios. However, Durupinar model supports and demonstrates various scenarios including acquisitive mobs as in a Black Friday scene and angry mobs as in a protest event, alongside escape mobs with fear.

In the sense of supporting multiple groups and individual types in a scenario, the mentioned methods differ from each other. While ASCRIBE copes with only one group in a scenario, ESCAPES, BioCrowd and Durupinar models support multiple groups with different goals, intentions and responsibilities in a scenario.

Regarding the emotional contagion method, agents in ESCAPES adopt the maximum level of fear from their surrounding agents and their fear level decreases when they pass by an authority figure. In ASCRIBE and BioCrowd, contagion occurs gradually without any threshold, and distance is the major external factor in determining the strength of influence. In ASCRIBE, fear can increase and decrease, based on the incoming contagion. It can take a value between 0 and 1. The Durupinar model incorporates an epidemiological emotion concept where emotions show up after they exceed a threshold and decay over time. Overall, because of its capability

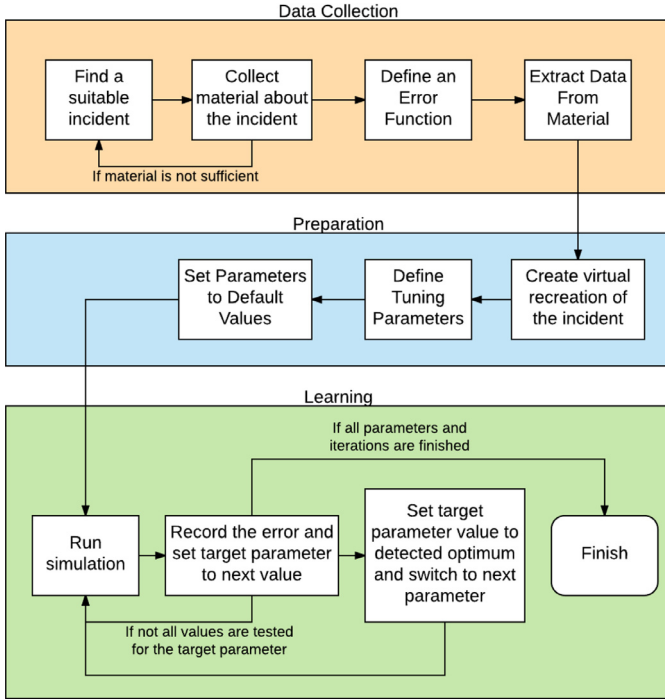


Fig. 1. General flow of optimization and parameter learning process.

of supporting various kinds of scenarios where crowds behave as a whole and react with various emotions, we base our studies on Durupinar Emotion Contagion Model and Crowd Simulation System.

#### 4. Optimization and parameter learning

Although our approach for data-driven parameter learning for crowds is not entirely new, we now provide a clear formulation of the schema, requirements and specific actions needed for performing this. For this purpose, we refer to previous work and our own experience. In our parameter learning framework, we first collect video footage for suitable incidents and preprocess them when necessary. Next, we track the people on the video and extract relevant data: either trajectories of individuals or occurrences of defined events, and define an error function that represents the accuracy of a simulation. Then, we recreate the scene in the virtual environment of the target crowd simulation model with its static obstacles, reference points and virtual agents. We project the extracted trajectories from video to the virtual scene. Finally, we define the parameters to be learned and run the parameter optimization algorithm (cf. Fig. 1).

For the first step of the data-driven parameter learning process, we collect data, which entails finding videos of real-life incidents including crowds with emotional responses. We describe the characteristics of videos that would be suitable for such work as being publicly available and without graphic content, having captured the emotional responses of crowds of suitable sizes and showing the incident on a flat surface [17]. In this work, we use three different videos, each belonging to a different incident. The videos are public and available online.

##### 4.1. Transfer from pixel coordinates to scenario coordinates

One of the main error functions we use for formulating the difference between real world and simulation data is trajectory matching. Trajectory matching is performed by summing the differences between the paths of individuals and their corresponding

virtual agents. Instead of camera parameter extraction which relies on careful calibration in a controlled environment [43], we use barycentric coordinates [44] to map real-world trajectories onto the virtual scene [17].

Projecting information from camera footage onto 3D space requires picking at least three points of interests with known coordinates both on the video and the virtual scene. So, we must create a virtual and properly scaled version of the area before the coordinate projection. In other words, the relative distances and sizes in the virtual environment should match the real world. When we know the camera space coordinates as well as the 2D coordinates in the scene, we can calculate the barycentric coordinates for any pixel in a video frame with respect to our three reference points. Then, we can calculate the coordinates of the point in the 3D scene using the barycentric coordinates.

Let our reference pixels be  $rp_i = [x_i, y_i]$ ,  $i = 1, 2, 3$  and query pixel be  $qp = [x_p, y_p]$ , for which we want to get the scene coordinates. We can find the corresponding virtual scene coordinates of pixel  $qp$  as follows:

1. Calculate the barycentric coordinates  $[b_1, b_2, b_3]$  of the query point ( $qp$ ) with respect to the pixel coordinates of the three reference points by solving

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$

2. Calculate the scene coordinates  $sc$  of the query point with the known scene coordinates of the three reference points ( $srp_{1, 2, 3}$ ) as:  

$$sc = srp_1 b_1 + srp_2 b_2 + srp_3 b_3.$$

This approach assumes that the event scene is flat and the camera image is a perfect linear projection of the scene without any lens distortion. For better results, the camera image can be pre-processed to disable lens distortion.

##### 4.2. Parameter optimization

Our goal is to find the best combination of parameters that control the behavior of virtual agents. For this purpose, we must formulate an error function that reflects the difference between the simulated scenario and the real event. The error function should be formulated per-scenario basis, considering the natures of events in the scenario. With this, the meaning of “best combination of parameters” becomes the vector of values, which minimizes the defined error function:

$$\underset{p}{\text{minimize}} \quad \text{error}(P)$$

$$\text{subject to} \quad p_{imin} \leq p_i \leq p_{imax}, p_i \in P$$

For searching the optimum parameter values in the search space, we run a simple independent parameter tuning algorithm (cf. Algorithm 1), similar to Bosse et al. [24]. This method instantiates parameters to their minimum values at the beginning, iterates through parameters optimizing one parameter at a time by calculating the error when the value of the parameter being optimized is updated step by step and the values of the rest of the parameters are fixed.

This optimization process allows us to scale the tuning ranges of individual parameters with minimal overhead to complete the whole process and gives information about the precision and the effect of individual parameters on the overall result. By taking advantage of these properties, we can automatically improve the testing efficiency with each iteration by reducing the step size of the sensitive parameters and increasing the possibility range of critical variables.



**Algorithm 1** Independent parameter tuning.

---

```

k: number of tuning iterations
P: set of parameters
si: step size of parameter pi
min(pi): minimum value of pi
max(pi): maximum value of pi
val[pi]: current value of pi
for c = 1 to k do
  for all pi ∈ P do
    val[pi] ← min(pi)
    bestError ← ∞
    bestVal ← val[pi]
    while val[pi] ≤ max(pi) do
      calculate currentError
      if currentError < bestError then
        bestError ← currentError
        bestVal ← val[pi]
      end if
      val[pi] ← val[pi] + si
    end while
  end for
end for

```

---

This technique scales quite well in terms of the number of parameters to be searched as well as the number of distinct values (steps) each parameter can take. If  $n$  is the number of parameters and  $m$  is the average number of steps of parameters, then the runtime of this algorithm is  $\mathcal{O}(n \times m)$ .

After all the values have been tested, the parameter is assigned to the optimal value and the calibration process continues with the next parameter. After all the parameters are tuned, the whole process restarts with the first parameter, using the previously found optimum values.

## 5. Scenarios

In order to demonstrate our system at work, we have recreated three different real-life incidents involving emotional responses of crowds using the Durupinar emotion contagion model and crowd simulation system. The first incident involves a suicide bomber detonating himself and the crowd's escape from the incident area. The second incident is a state of panic in a subway train where the crowd suspects one of the passengers being a suicide bomber. The third incident is a crowd storming the gate of a store opening on a Black Friday.

### 5.1. Ankara attack scenario

The first video is the footage of the terrorist attack in Ankara Train Station on October 10, 2015, which we are going to refer from now on as the Ankara Attack. During a gathering in an open space just outside the railway station, two bombs were detonated, resulting in a death toll of 103 civilians and the physical injury of more than 400. We chose this video because there is a stable footage of the incident taken from a surveillance camera overseeing the scene and the panicking crowd, the size of the crowd captured by the camera is between 50 and 200 people – suitable for the crowd simulation model, the footage does not contain graphic violence, thus can be used in public media, and the environment is flat.

#### 5.1.1. Surveillance camera footage

The video footage is taken from a city surveillance camera at the center of the gathering area, pointing to the west (see Fig. 2). On the video, there is a traffic light pole at the center, a street light pole and a white panel van car parked at the bottom right



Fig. 2. Surveillance camera footage of the Ankara Attack incident.

corner; and the scene is filled by the crowd slowly roaming the area or standing still. The explosion is seen on the left at the 11th second of the original video. After that, the crowd starts running away from the center of the explosion to the top and bottom right corners of the video.

The original video that we have access to is a mobile phone camera recording of a computer screen, playing the actual surveillance camera footage. Therefore, it contains unwanted panning and motion blur. In order to reduce these glitches, we preprocessed the video by stabilizing it with the traffic light pole as the reference point. After stabilizing the video, we cropped it, so the whole scene consists of the actual footage of the surveillance camera. Finally, we trimmed the video. The processed video has a resolution of  $446 \times 250$ , 16 seconds of length with 12 frames per seconds and 434 kbps of bit rate, which sums to 826 kilobytes in size.

#### 5.1.2. Tracking

In order to track the people in the crowd, we first tried automating the pedestrian trajectory extraction, as described in the literature [9]. However, because the quality of the video is low, the pedestrian detection methods perform poorly. So, after various attempts, we decided to track people manually, similar to Bosse et al. [21]. We used an open source software, called “Tracker” [45] for tracking people in the crowd. This was done in a per-agent basis, by clicking on the position of a person at each frame, doing this until the person leaves the area covered by the video. Because the video is blurry, it is difficult to track the positions of individuals in groups. We were able to track ten individuals. We started tracking just before the explosion and tracked these individuals for various durations– five seconds (or 62 frames) on the average. For the barycentric coordinate projection, we have identified three spots as the reference points: the traffic pole in the center, the street light on the right, and the corner of protection bars of the underpass for the projection process. The reference points are shown with red dots in Fig. 3.

#### 5.1.3. Virtual scene

We constructed the virtual scene with 180 agents by exporting a satellite image from Google Earth around the coordinates of 39.9366 latitude and 32.8442 longitude (cf. Fig. 4). We scaled the image as the ground plane in a Unity 3D scene with real world coordinates of one meter corresponding to one unit in Unity.

The scene is placed in such a way that the base of surveillance camera pole sits at the origin of the world coordinate system with the positive  $z$  axis pointing to the north and the positive  $x$  axis pointing to the east. With this setup,  $y$  axis points to the sky because Unity uses left-handed coordinate system. We placed static obstacles for the train station building, trees between the train station and incident scene, the traffic light pole, the street light pole,



**Fig. 3.** Reference points used for barycentric coordinate projection of tracked pixels in Ankara Attack incident. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



**Fig. 4.** Virtual simulation of the Ankara Attack incident.

the car under the street light and the underpass to populate the scene.

The Durupinar Emotion Contagion model allows us to define standards that individuals have for themselves as well as the others. Standards are about what people think of other people's actions: approving or disapproving them. In this scenario, we set approving standards of individuals towards themselves as well as towards other agents. Because people are gathered in the area for a common goal, we assume that they sympathize with each other. We also assigned them an unpleasant (displeased) goal of waiting in the area because the gathering was about a protest.

We defined 15 parameters that could be tuned easily and impact the outcomes of the simulation results (cf. Table 1). Ten of these parameters are the means and standard deviations of the five personality factors. The Durupinar Model allows us to assign the values of personality traits for each agent in the crowd from probability distributions whose means and standard deviations are specified. Hence, the parameters are not the values of the personality traits for each agent in the crowd, but rather the means and standard deviations of the personality traits for the crowd. This significantly reduces the number of parameters to be tuned for the crowd. We specify the ranges (minimum and maximum values) and optimum values for the means and standard deviations of the five personality traits. Alongside the personality variation, we tuned the parameters for the weights of the standards the agents have for themselves and for the other people, the initial goal of roaming around the gathering area, the goal of running from the explosion and the fear threshold for starting to panic.

The error function for this scenario is a trajectory matching error function, i.e., the sum of distances between each tracked person

**Table 1**  
Parameters used in the Ankara Attack scenario.

Parameter	Min.	Max.	Step size	Optimum value
mean(O)	-0.8	0.8	0.2	0.6
std(O)	0.0	1.0	0.2	0.4
mean(C)	-0.8	0.8	0.2	0.2
std(C)	0.0	1.0	0.2	0.2
mean(E)	-0.8	0.8	0.2	0.2
std(E)	0.0	1.0	0.2	0.8
mean(A)	-0.8	0.8	0.2	-0.2
std(A)	0.0	1.0	0.2	0.0
mean(N)	-0.8	0.8	0.2	0.6
std(N)	0.0	1.0	0.2	0.6
Wait goal	0.0	1.0	0.1	0.5
Escape goal	0.0	1.0	0.1	0.4
Std. for self	0.0	1.0	0.1	0.7
Std. for crowd	0.0	1.0	0.1	0.0
Panic threshold	0.0	1.0	0.1	0.2



**Fig. 5.** Security camera footage of the Subway Panic incident.

and the corresponding virtual agent. Minimizing this error function allows us to obtain a realistic escape pattern with accurate running speeds, escaping directions and obstacle avoidance behavior of agents.

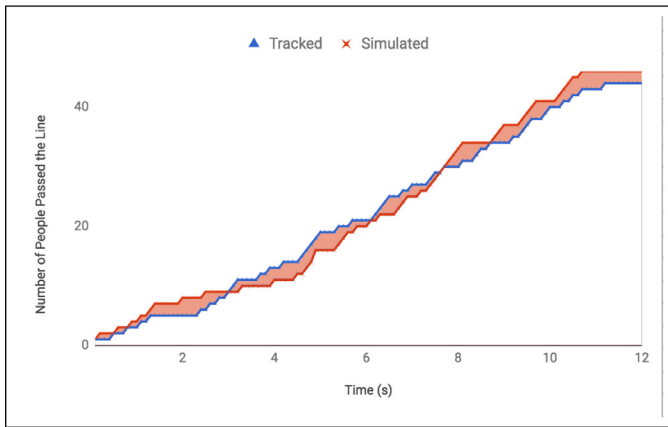
We set the tuning environment for the virtual Ankara Attack scenario with the described 15 parameters, running each test three times and taking the median for the error. The tests were run for four iterations, so the parameters took turns four times in the tuning process, with another set of tuned parameters each time. In addition, we ran the test scenario in  $0.5 \times$  slow motion in order to let Unity dedicate more processing time to each frame.

## 5.2. Subway panic scenario

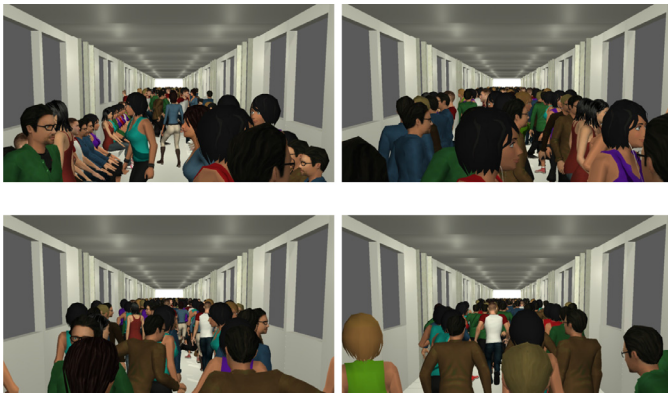
The second incident that we study is a state of panic on a subway car. Late 2015 was a period in Turkey when various terrorist bombing attacks had happened and people were expecting more attacks because of rumors on social media [46]. On October 16th, two days after a bombing attack, during rush hour in a subway car in Ankara, a passenger started yelling and pointing to another passenger stating that she suspected him to be a suicide bomber. At that point, two undercover police officers rushed to neutralize the suspect while other passengers ran away from him.

### 5.2.1. Security camera footage

The incident can be seen from video footage of a security camera in the subway car, published by the press. On the video, we see the passengers turning their heads to the screaming person and starting to run away from the incident point (see Fig. 5). The camera looks away from the screaming person, so she is not visible.



**Fig. 6.** Representation of the error function used for optimizing the model for Subway Panic incident. The area between the simulated and tracked people count lines corresponds to our error.



**Fig. 7.** Virtual simulation of the Subway Panic incident.

### 5.2.2. Tracking and error function

We tracked the movement of individuals to estimate the flow rate of the crowd in the car. For each frame, we counted how many people have entered the field of view of the camera so far. We aimed to capture the reaction times of individuals as the fear spreads, which can be used to calculate a flow rate for the crowd. We counted around 50 people as they ran from the suspect in fear and came up with a time series graph of the number of people who crossed the line with respect to the time passed. We derived the same graph for the simulation with the virtual agents and used the difference between the tracked and simulated graphs as an error function, as can be seen in Fig. 6.

### 5.2.3. Virtual scene

We created a scene with 300 virtual agents in three subway cars, with the middle car representing the one seen on the video (cf. Fig. 7). In the scene, seats are assigned to virtual agents by their proximities. When the fear level of an agent exceeds a fear threshold, which is a tuning parameter of the scene, that agent stands up and tries to run away.

Similar to the Ankara Attack scene, ten tuning parameters come from the OCEAN personality model: mean and standard deviation of each of the five personality traits. Additionally, we have defined a goal to freeze, i.e., stand still as nothing has happened, a goal for escaping from the danger, an approving standard for the agent itself, a disapproving standard for the surrounding agents and a fear threshold, after which the agents start running away (cf. Table 2).

**Table 2**

Parameters used in the Subway Panic scenario.

Parameter	Min.	Max.	Step size	Optimum value
mean(O)	-0.8	0.8	0.2	-0.4
std(O)	0.0	1.0	0.2	0.8
mean(C)	-0.8	0.8	0.2	0.4
std(C)	0.0	1.0	0.2	0.4
mean(E)	-0.8	0.8	0.2	0.4
std(E)	0.0	1.0	0.2	0.4
mean(A)	-0.8	0.8	0.2	-0.8
std(A)	0.0	1.0	0.2	0.2
mean(N)	-0.8	0.8	0.2	0.4
std(N)	0.0	1.0	0.2	0.2
Wait goal	0.0	1.0	0.1	0.2
Escape goal	0.0	1.0	0.1	0.6
Std. for self	0.0	1.0	0.1	0.4
Std. for crowd	0.0	1.0	0.1	0.6
Panic threshold	0.0	1.0	0.1	0.2



**Fig. 8.** Video footage of the Black Friday incident.

### 5.3. Black Friday scenario

The third incident is a Black Friday midnight opening of a store in a mall on November 25, 2011 where a crowd of people rushed to get inside the store. The incident involves about 400 people, getting inside in about 30 s.

We have a 30 s video of the incident, taken by a person with a cell phone camera from a higher ground. In the beginning, when the gate is closed, people are waiting outside the store. When the gates start being pulled up, people start yelling and pushing each other towards the store gate. After a few seconds, people start getting inside the store (see Fig. 8).

#### 5.3.1. Video tracking

The camera footage of the Black Friday incident has significant amount of movement and is not suitable for stabilization with post processing, therefore we couldn't use trajectory matching for this incident. For tracking the video we selected evenly-positioned 14 individuals in the crowd from the video and recorded their time to enter the store. Similar to the Ankara Attack scenario, we transformed the pixel coordinates of the initial positions of the selected individuals to the virtual scene positions. We created 14 clusters of virtual agents, each cluster representing a tracked individual. Then we assigned each virtual agent to one of the clusters according to its closest tracked individual. In the scenario, we used these clusters to record the average time-to-enter the store and as a metric for error computation.





Fig. 9. Virtual simulation of the Black Friday incident.

### 5.3.2. Virtual scene

We created a virtual scene with 400 agents waiting outside the store (cf. Fig. 9). At the beginning of the scenario, we assigned the agents to their representative tracked individual, forming a cluster around the tracked individual. We also selected a weighted random destination for each virtual agent inside the store. The weights used in selecting the random destination are based on whether the agent is on the left side of the crowd or the right. Depending on this, an agent's destination is more likely to be on the same side as the agent. In the scene, the virtual gate starts to open after two seconds from the scenario beginning at 0.25 m/s speed. When the gate starts opening, virtual agents try to push towards the gate, similar to what happens on the video. To simulate the people getting inside the store before the gate is fully open, we assigned a random threshold height between 1 m and 1.8 m for each virtual agent so that the agent can enter the store after the gate height is above the threshold.

The error function that has been used for optimizing the model for the Black Friday scenario is the store entrance duration difference between the tracked and the simulated individuals. For each virtual agent, we recorded the time to enter the store. After all the agents in the scene enter the store, the average duration of clusters to enter the store and the sum of the differences between the cluster averages and tracked agents were calculated. The error function can be formulated as

$$error = \sum_{i=1}^n \left| t(ta_i) - \frac{\sum_{j=1}^{c_i} t(aa_{i,j})}{c_i} \right|,$$

where  $ta_i$  is the  $i^{th}$  tracked agent,  $aa_{i,j}$  is the  $j^{th}$  virtual agent assigned to  $ta_i$ ,  $c_i$  is the number of virtual agents assigned to  $ta_i$ ,  $t(ta_i)$  and  $t(aa_{i,j})$  are the time to enter the store for  $ta_i$  and  $aa_{i,j}$ , respectively, and  $n$  is the number of tracked agents.

### 5.3.3. Tuning parameters

We have 16 parameters in this scenario (cf. Table 3). Similar to other scenarios, ten of these consist of means and standard deviations of the distributions of five OCEAN personality traits. We define a goal to pass the gate, disapproval towards the other shoppers, and some stimulating factor towards the sales event. If the neuroticism value of an agent is below a threshold, which is also parameter, the stimulating factor is used as “hope to get wanted items”. If the neuroticism level is above the threshold, the stimulating factor represents the “fear of not getting the items”.

## 6. Results and discussion

The total time to run all the simulations was about 50h with small differences among scenarios. For this work, we have used a

Table 3

Parameters used in the Black Friday scenario.

Parameter	Min.	Max.	Step size	Optimum value
mean(O)	-0.8	0.8	0.2	0.2
std(O)	0.0	1.0	0.2	0.8
mean(C)	-0.8	0.8	0.2	0.4
std(C)	0.0	1.0	0.2	0.6
mean(E)	-0.8	0.8	0.2	0.6
std(E)	0.0	1.0	0.2	0.4
mean(A)	-0.8	0.8	0.2	-0.4
std(A)	0.0	1.0	0.2	0.8
mean(N)	-0.8	0.8	0.2	0.6
std(N)	0.0	1.0	0.2	0.4
Std. for gate	0.0	1.0	0.1	0.7
Crowd displeasure	0.0	1.0	0.1	0.5
Sales fear	0.0	1.0	0.1	1.0
Sales hope	0.0	1.0	0.1	0.0
Wait goal	0.0	1.0	0.1	0.6

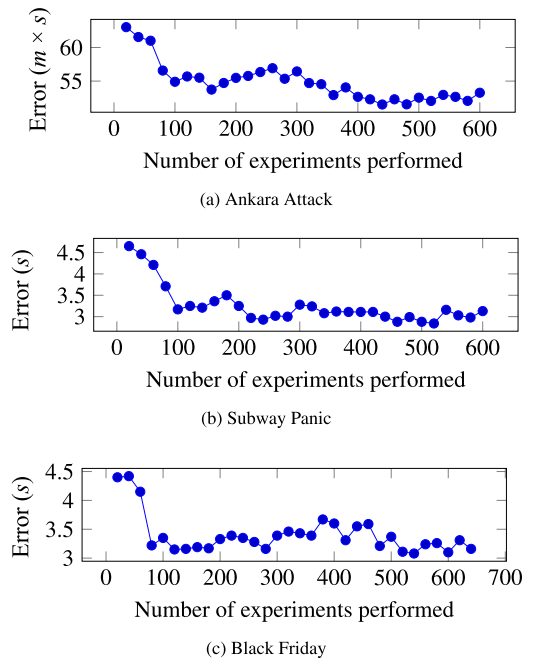


Fig. 10. Tendency of error versus the number of experiments performed.

computer with Intel® Core™ i7-4790 CPU running at 3.60 Ghz, 16GB RAM and two AMD® Radeon™ R9 290 GPUs with 4GB GDDR5 memory each connected with AMD® CrossFireX™ multi-GPU technology, running Windows® 10 64bit and Unity Engine v5.6.1.

In our experiments, we observe that the error tends to go down in general as we perform more experiments and tune the parameters accordingly (see Fig. 10). The improvement happens at a faster rate at the beginning because of the overwritten default values; but after the first pass, the improvement slows down. For example, it is very difficult to notice the improvement after about 150 experiments on the Black Friday scenario.

The fluctuations in the error graphs are caused by two reasons. The first one is that the parameter estimation algorithm tests different values for a single parameter over time and we can expect an actual improvement only when the complete range of values for a parameter is evaluated and a local optimum is chosen for that parameter. The second one is that the underlying crowd simulation system is not deterministic; even if the same simulation is performed with the same parameter configuration, the results vary significantly. The indeterminism is caused by the asynchronous



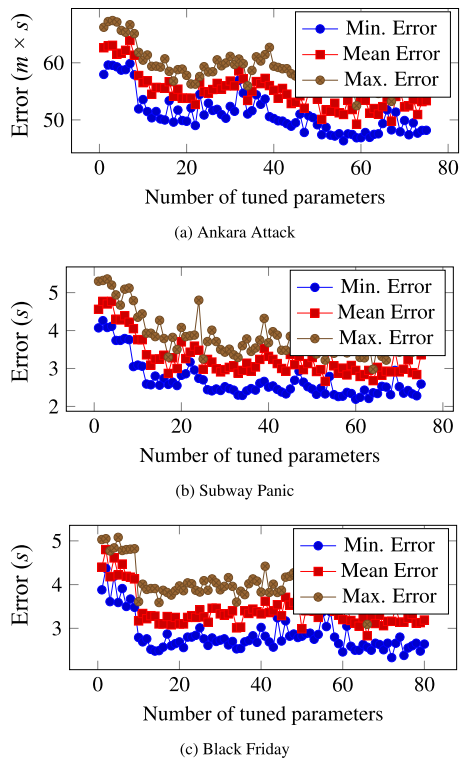


Fig. 11. Tendency of error as parameters are being tuned.

multi-agent pathfinding framework of the Unity3D engine. Because Unity does not provide control over the internal mechanisms and parameters of its algorithms, simulations produce different results even if we fix the values of our parameters. In order to achieve more stable simulation results, we repeated each experiment multiple times and recorded the average observed error within a repetition group as the error value of a parameter configuration.

As parameters are further calibrated, we expect the minimum, mean and maximum error values observed within the tuning groups to go down as well. In this context, a tuning group is a set of experiments where one parameter changes value and the other parameters are fixed. This is because parameters together affect the simulation results and as more parameters are in their 'correct' values, the effect of one 'wrong' parameter would be smaller. Our results show that this is the case, especially for the Ankara Attack and Subway Panic scenarios. As more parameters are tuned, the range of the observed error values go down (see Fig. 11). In the Black Friday scenario, blessing of indeterminism plays a role and the best error is reached in the earliest stages; but the results become more stable as more parameters are tuned. Ideally, this graph would be a non-increasing one, because every step represents one more tuned parameter, that would result with a new error value which is less than or equal to the previous one. Despite the fluctuations, observed errors tend to go down.

We expect that performing multiple passes of parameter tuning, i.e., setting parameters to their observed best values after an iteration and restarting the tuning process with these values would further decrease the error function. This is because the influence of the parameters on agent behavior and the overall simulation results are not independent of each other. Indeed, our experiments show that performing multiple tuning iterations would increase the simulation accuracy as can be seen in Fig. 12. Overall, the results show that our methodology allows simulations to resemble the real world events.

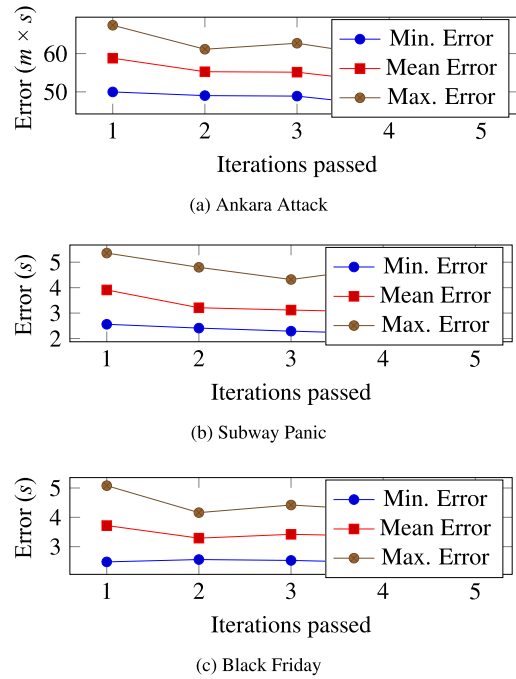


Fig. 12. Tendency of error with tuning iterations.

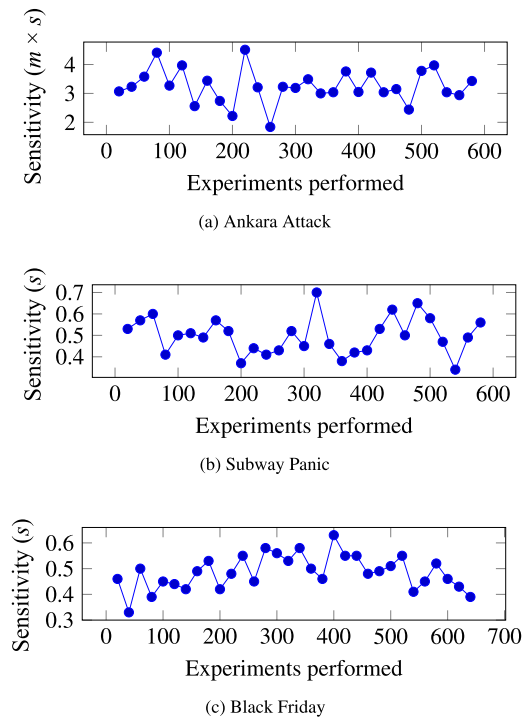


Fig. 13. Sensitivity of the parameters as the experiments performed.

Next, we observe that the sensitivities of parameters in each scenario are variable. Sensitivity is the variation in the resulting error for a given parameter. It allows determining the order of significance and granularity of tuning ranges of parameters. Initially, we expected parameter sensitivities to go down as parameters are being tweaked. As more parameters were optimized, the error would decrease and the effect of changing a single parameter while other parameters are fixed on their latest optima would get smaller over time. However, the experimental results (see Fig. 13) show that there is no correlation between the overall parameter

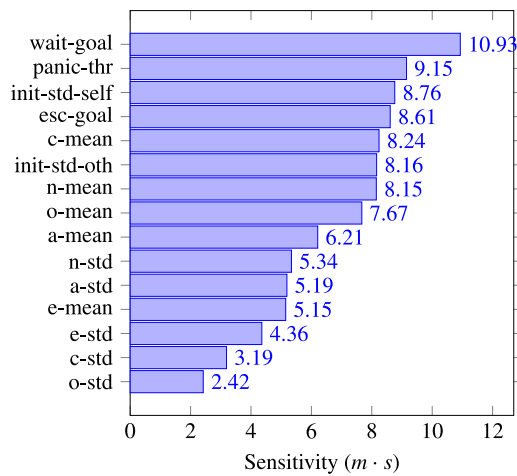


Fig. 14. Parameter sensitivities of the Ankara Attack scenario.

sensitivity and the number of experiments performed. We suspect that the indeterministic nature of the simulations and the amount of change in the error due to optimizations being smaller than the actual sensitivity of a parameter; i.e., the change in error when a parameter is changed, could cause such behavior.

In the Ankara Attack scenario, we expected the most sensitive and significant parameters to be the ones about survival instincts that are directly related to the steering behavior of agents: self standards of agents, their fear threshold to start panicking and the goal about running away from the explosion. In our results, as shown in Fig. 14, we confirm that individuals' standards for themselves and goals about the explosion have the highest sensitivity. Personality parameters have slightly less significance compared to appraisal elements because they usually affect the reactions indirectly. It is not surprising that neuroticism is one of the most significant traits because it is directly connected to the contagion of fear. Conscientiousness trait is also a significant parameter. One explanation of this observation is that because the conscientiousness of agents affects their collision avoidance behaviors, it can also affect their escape paths and thus the error function. The variations in the distribution of personality traits have less significant parameters, probably because individual differences do not matter when the crowd reaction converges quickly.

The nature of the Subway Panic scenario is similar to the Ankara Attack scenario: survival by escaping from a danger source. This led us to have similar expectations with the Ankara Attack scenario, i.e., that the most important factors would be about survival. As can be seen in Fig. 15, the experiments show that the most sensitive parameters of the Subway Panic scenario are similar to the ones in the Ankara Attack scenario. The survival reaction-related parameters (escape-goal, wait-goal, neuroticism, standard about self and others) are the most sensitive. However, we expected the wait-goal to have more significance because fear increases more slowly than the Ankara Attack scenario and the tendency to freeze would be a more determining factor. Similarly, we expected conscientiousness to be a more significant parameter as in the Ankara Attack scenario although the actual trajectories of agents are not considered in the error function here.

The Black Friday scenario has a different nature than the other scenarios where the agents are trying to escape from a danger source. The crowd is competing with each other to get their desired items, which is also about survival in primal sense [47]. Similar to the subway panic scenario, the error function is geared towards capturing a realistic human flow through a gateway. With these in mind, we expected the wait-goal, which is directly

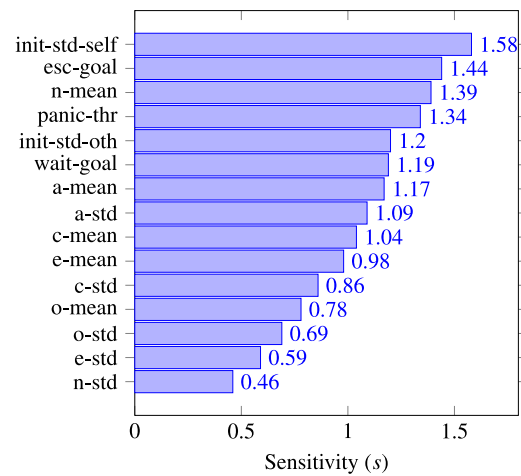


Fig. 15. Parameter sensitivities of the Subway Panic scenario.

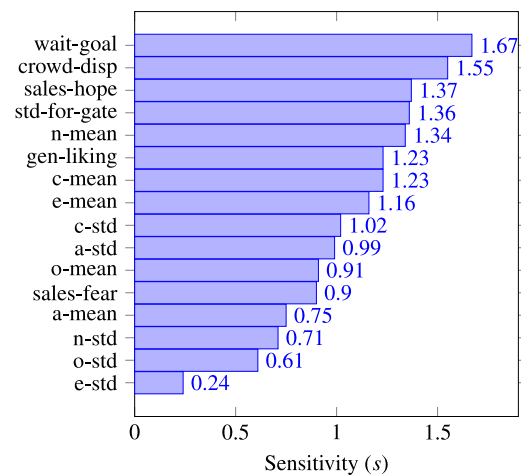


Fig. 16. Parameter sensitivities of the Black Friday scenario.

related to people rushing towards the store and passing under the gate before it is fully open; attitude towards the other members of the crowd (general-liking, crowd-displeasure), which affects the amount of pushing between agents and goals towards getting into the store (sales-hope, sales-fear) to be the most significant parameters. The experiments show that the results mostly meet our expectations (see Fig. 16). The wait-goal is the most significant factor because it directly affects the error function and crowd-displeasure is the second for directly affecting the crowd flow. However, fear of not getting the desired items is a less significant parameter. We observe that this parameter does not affect the agents' behavior until they enter the store. Because our error function only considers the motion of entering the store rather than what happens afterwards, the parameter becomes less significant.

In all three scenarios, we can observe that the scenario-specific parameters, such as the wait goal, the panic threshold, and the escape goal, are among the most significant ones. The reason for this pattern is that the scenario-specific tuning of parameters affects the agents' motions directly, causing changes in error. For example, when panic threshold is too low in the Ankara Attack scenario, we saw that simulated agents started running faster too soon, causing a larger difference between the simulated and tracked trajectories. On the opposite extreme, when the fear threshold was too low, the agents didn't panic at all and escaped from the scene by walking calmly, again resulting in a different trajectory.

Another common pattern among the sensitivity of parameters in different scenarios is that the least significant parameters were the OCEAN-related ones. This can be explained by only a few personality dimensions affecting the motions of the agents and their impact being lower than the scenario-specific parameters. In addition, the results indicate that standard deviations of OCEAN dimensions are much less significant than the mean values of the parameters. Our data extraction process mostly consists of tracking a small subset of individuals in the video and generalizing the results. We observe that even when the variation of personality values are high, the generalization of simulated results eliminates the heterogeneity of the data, which is the cause of the low sensitivities of parameters for standard deviations of OCEAN dimensions. For example, in the Black Friday scenario, when the standard deviation of the neuroticism is high, the difference between the motions of agents in the simulated group can be easily observed. However, this difference is not reflected in the error function because we take the average time-to-enter to the store of a cluster of agents while calculating the error.

The results of the optimization efforts is a way of validating the crowd simulation model that has been tuned. For each scenario, we defined error functions to represent dissimilarity between the real and the simulated world. Being able to achieve a small error by tuning a crowd simulation model can be interpreted as the suitability of the studied crowd simulation model to represent the specified scenario. In other words, if two different crowd simulation models are tuned with this approach, using the same scenarios and the error functions for both and keeping their own parameter sets, then their performances can be compared based on the resulting error values.

## 7. Conclusion, limitations and future work

We propose a framework for the optimization of crowd simulation parameters by using videos of real-life incidents. We describe the nature of suitable incidents and the steps involved in data extraction from them. We then optimize the Durupinar Emotion Contagion model considering a subset of individuals in such incidents by learning personality, emotion and contagion parameters from videos and tuning them one by one. We apply the process on three separate incidents and discuss the performance of the proposed method and our findings about the incidents.

In the future, we intend to improve our individual tracking method. To achieve this, multiple and clearer video tracks of similar events are required. This would be possible by gaining more access to press media, professional or surveillance camera footages. Applying the work done in this study to other kinds of incidents, such as protests, stampedes, riots, looting incidents, natural disasters, and so on, would be valuable for augmenting the learned personality distribution and improving the emotion contagion model.

By collecting media for real-life incidents and processing them, we could acquire solid evidence about the personalities of different groups or cultures. By optimizing the learned parameters from multiple incidents in a region, we could extract the actual distribution of personalities in the area and use these learned personalities for the simulation of possible incidents to understand how people would react in such events. This could be used to take precautions and design streets, public gathering areas and crowded buildings such as shopping malls and airports.

We use the barycentric coordinates for projecting positions from the camera to the scene thus ignoring the distortions caused by lenses. If we had more reference points or details about the characteristics of the cameras used, we could use intrinsic and/or extrinsic camera parameter extraction techniques to estimate the projection matrix of the pinhole model of the camera. With this camera matrix, we could reverse the projection from the camera

to the 3D world, which would result in a more accurate projection model.

For the parameter optimization problem, we try to optimize the personality parameters independently. Although the underlying OCEAN personality model depicts them as orthogonal traits, their mappings to behaviors and the outcomes of these behaviors would affect each other. Therefore, more general and stable parameter optimization methods such as genetic algorithms or support vector machines might produce faster and more accurate results. Moreover, the parameter estimation could be expanded to the amount of emotion doses according to the reactions to events, goals and response thresholds.

As the error function of the Ankara Attack incident, we use the sum of distances between the tracked agents and their corresponding virtual agents. Although this metric is beneficial for estimating the running speeds, directions and reaction times, using more accurate metrics could lead to better understanding on the decisions made by the individuals. One such metric could be the proportion of people doing action  $a_k$ ,  $1 \leq k \leq n$  in a set of actions  $\{a_i, i \in 1, \dots, n\}$ . For example, in the Ankara Attack scenario,  $a_1$  could be the action of “running north”,  $a_2$  could be “running east” and  $a_3$  could be “lying on the ground”. Moreover, optimizing the parameters for multiple metrics at the same time would help produce more robust models.

The immense computational requirements of the emotion contagion models and parameter estimation techniques limits us on the number of tuning iterations, granularity of parameter ranges, crowd size of the analyzed incidents, etc. Improvement in the results of this study can be achieved by dedicating more computational resources and time on the tuning process and working on areas of computational optimization in the crowd simulation model implementation. For example, we assign personality values to a group by assuming a Gaussian distribution of personality traits within the group; thus specifying only the mean and standard deviation values of the distribution. Optimizations on the model and more powerful computers may allow us to treat personalities of each individual agent as a separate parameter to be tweaked to have more fine-tuned results. In general, providing more data, preparing more scenarios and dedicating more resources result in a more stable and better tuned crowd simulation model.

Our parameter tuning approach is not limited to tuning the Durupinar model, but can be used with other crowd simulation models as well. The proposed pipeline can also be used to compare and validate different crowd simulation and emotion contagion models by comparing results of error functions after parameter tuning is done for each model. Of course, the parameters to be tweaked are specific to each model. However, we explicitly describe the actions to calibrate the parameters in order to find optimal values so that our schema can be generalized across different crowd simulation or emotional contagion models.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.cag.2018.02.004](https://doi.org/10.1016/j.cag.2018.02.004).

## References

- [1] Le Bon G. *The crowd: a study of the popular mind*. Dover Publications; 2002.
- [2] Blumer H. *Collective behavior*. In: Lee AM, editor. *Principles of sociology*. New York: Barnes & Noble; 1951. p. 67–121.
- [3] Allport G. *The historical background of modern social psychology*. In: Lindzey G, editor. *Handbook of social psychology*. Addison-Wesley; 1954. p. 3–56.
- [4] Milgram S, Toch H. *Collective behavior: crowds and social movements*. In: Lindzey G, Aronson E, editors. *The handbook of social psychology*. Reading, MA: Addison-Wesley; 1968.
- [5] Dollard J, and Neal Miller LD, Mowrer H, Sears R. *Frustration and aggression*. Yale University Press; 1939.



- [6] Turner R, Killian LM. *Collective behavior*. Prentice Hall; 1993.
- [7] Brown RW. *Mass phenomena*. In: Lindzey G, editor. *The handbook of social psychology*. Cambridge, MA: Addison-Wesley; 1954. p. 833–76.
- [8] Dodds PS, Watts DJ. A generalized model of social and biological contagion. *J Theor Biol* 2005;232(4):587–604.
- [9] Bera A, Kim S, Manocha D. Efficient trajectory extraction and parameter learning for data-driven crowd simulation. In: *Proceedings of the 41st Graphics Interface (GI '15)*. Canadian Information Processing Society; 2015. p. 65–72.
- [10] Lee KH, Choi MG, Hong Q, Lee J. Group behavior from video: a data-driven approach to crowd simulation. In: *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics; 2007. p. 109–18.
- [11] Lerner A, Chrysanthou Y, Lischinski D. Crowds by example. *Comput Graph Forum* 2007;26(3):655–64.
- [12] Singh S, Kapadia M, Faloutsos P, Reinman G. SteerBench: a benchmark suite for evaluating steering behaviors. *Comput Anim Virtual Worlds* 2009;20(5–6):533–48.
- [13] Musse SR, Cassol VJ, Jung CR. Towards a quantitative approach for comparing crowds. *Comput Anim Virtual Worlds* 2012;23(1):49–57.
- [14] Charalambous P, Karamouzas I, Guy SJ, Chrysanthou Y. A data-driven framework for visual crowd analysis. *Comput Graph Forum* 2014;33(7):41–50.
- [15] Carretero MR. *Expression of emotion in virtual crowds: Investigating emotion contagion and perception of emotional behaviour in crowd simulation*. Department of Computer Science and Communications, KTH Royal Institute of Technology; 2014. Master's thesis.
- [16] Durupinar F, Gdkbay U, Aman A, Badler NI. Psychological parameters for crowd simulation: from audiences to mobs. *IEEE Trans Vis Comput Graph* 2016;22(9):2145–59.
- [17] Bařak AE, Durupinar F, Gdkbay U. Learning from real-life experiences: a data-driven emotion contagion approach towards more realistic virtual crowds. In: *Proceedings of 40th international conference on computer animation and social agents (CASA 2017)*. Seoul, South Korea; 2017. p. 55–64.
- [18] Lemerrier S, Auberlet J-M. Towards more behaviours in crowd simulation. *Comput Anim Virtual Worlds* 2016;27(1):24–34.
- [19] Berseth G, Usman M, Haworth B, Kapadia M, Faloutsos P. Environment optimization for crowd evacuation. *Comput Anim Virtual Worlds* 2015;26(3–4):377–86.
- [20] Bosse T, Duell R, Memon ZA, Treur J, Van Der Wal CN. Multi-agent model for mutual absorption of emotions. In: *Proceedings of the 23rd European conference on modelling and simulation (ECMS '09)*. ECMS; 2009. p. 212–18.
- [21] Bosse T, Duell R, Memon ZA, Treur J, Van Der Wal CN. A multi-agent model for emotion contagion spirals integrated within a supporting ambient agent model. In: *Proceedings of the international conference on principles and practice of multi-agent systems (PRIMA 2016)*. Springer; 2009. p. 48–67.
- [22] Tsai J, Bowring E, Marsella S, Wood W, Tambe M. A study of emotional contagion with virtual characters. In: *Proceedings of the international conference on intelligent virtual agents (IVA '12)*. Springer; 2012. p. 81–8.
- [23] Tsai J, Fridman N, Bowring E, Brown M, Epstein S, Kaminka G, et al. ESCAPES: evacuation simulation with children, authorities, parents, emotions, and social comparison. In: *Proceedings of the 10th international conference on autonomous agents and multiagent systems (AAMAS '11)*, 2. IFAAMAS; 2011. p. 457–64.
- [24] Bosse T, Hoogendoorn M, Klein MC, Treur J, Van Der Wal CN, Van Wissen A. Modelling collective decision making in groups and crowds: integrating social contagion and interacting emotions, beliefs and intentions. *Auton Agents Multi-Agent Syst* 2013;27(1):52–84.
- [25] Tsai J, Bowring E, Marsella S, Tambe M. Empirical evaluation of computational fear contagion models in crowd dispersions. *Auton Agents Multi-Agent Syst* 2013;27(2):200–17.
- [26] Hoogendoorn M, Treur J, van der Wal CN, van Wissen A. An agent-based model for the interplay of information and emotion in social diffusion. In: *Proceedings of the IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology*, 2. IEEE Computer Society; 2010. p. 439–44.
- [27] Borodin A, Filmus Y, Oren J. Threshold models for competitive influence in social networks. In: *Proceedings of internet and network economics (WINE 2010)*. In: *Lecture notes in computer science*, 6484. Springer; 2010. p. 539–50.
- [28] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM; 2010. p. 1029–38.
- [29] Pereira G, Dimas J, Prada R, Santos PA, Paiva A. A generic emotional contagion computational model. In: *Proceedings of the international conference on affective computing and intelligent interaction*. Springer; 2011. p. 256–66.
- [30] Silverman BG, Badler NI, Pelechano N, O'Brien K. Crowd simulation incorporating agent psychological models, roles and communication. Technical Report. Center for Human Modeling and Simulation, University of Pennsylvania; 2005.
- [31] Helbing D, Molnar P. Social force model for pedestrian dynamics. *Phys Rev E* 1995;51(5):4282.
- [32] Helbing D, Farkas I, Vicsek T. Simulating dynamical features of escape panic. *Nature* 2000;407(6803):487–90.
- [33] Fridman N, Kaminka GA. Comparing human and synthetic group behaviors: a model based on social psychology. In: *Proceedings of the international conference on cognitive modeling (ICCM-09)*; 2009.
- [34] Lin Y, Fedchenia I, LaBarre B, Tomastik R. Agent-based simulation of evacuation: an office building case study. In: *Proceedings of the pedestrian and evacuation dynamics 2008*. Springer; 2010. p. 347–57.
- [35] Tan L, Lin H, Hu M, Che W. Agent-based simulation of building evacuation using a grid graph-based model. In: *Proceedings of the IOP conference series: earth and environmental science*, 18. IOP Publishing; 2014.
- [36] Festinger L. A theory of social comparison processes. *Human Relat* 1954;7(2):117–40.
- [37] Borges Fortes Neto A, Pelachaud C, Musse SR. Giving emotional contagion ability to virtual agents in crowds. In: *Proceedings of the 17th international conference on intelligent virtual agents (IVA 2017)*. Cham: Springer International Publishing; 2017. p. 63–72.
- [38] Bicho Ad L. *Da modelagem de plantas a dinmica de multides: um modelo de animao comportamental bio-inspirado*. Technical Report. University of Campinas - UNICAMP, Campinas; 2009.
- [39] Durupinar F, Pelechano N, Allbeck JM, Gdkbay U, Badler N. How the Ocean personality model affects the perception of crowds. *IEEE Comput Graph Appl* 2011;31(3):22–31.
- [40] Goldberg LR. An alternative "Description of Personality": the big-five factor structure. *J Personal Soc Psychol* 1990;59(6):1216–29.
- [41] Ortony A, Clore GL, Collins A. *The Cognitive Structure of Emotions*. Cambridge, MA: Cambridge University Press; 1988.
- [42] Mehrabian A. Pleasure-arousal-dominance: a general framework for describing and measuring individual differences in temperament. *Curr Psychol* 1996;14(4):261–92.
- [43] Zhang Z. A flexible new technique for camera calibration. *IEEE Trans Pattern Anal Mach Intell* 2000;22(11):1330–4.
- [44] Floater MS. Mean value coordinates. *Comput Aided Geom Des* 2003;20(1):19–27.
- [45] Brown D. Tracker video analysis and modeling tool for physics education. <http://physlets.org/tracker/>; 2017. [Online; accessed 10-September-2017].
- [46] Sabah D. CCTV footage shows false suicide bomber scare in Ankara subway. <https://www.dailysabah.com/ankara/2015/10/16/cctv-footage-shows-false-suicide-bomber-scare-in-ankara-subway>; 2016. [Online; accessed 10-September-2017].
- [47] Kruger D, Byker D. Evolved foraging psychology underlies sex differences in shopping experiences and behaviors. *J Soc Evol Cult Psychol* 2009;3(4):328.