Learning From Real-Life Experiences: A Data-Driven Emotion Contagion Approach Towards More Realistic Virtual Crowds

A. Eren Başak
Bilkent University
eren.basak@bilkent.edu.tr

Funda Durupınar
Oregon Health & Science University
durupina@ohsu.edu

Uğur Güdükbay
Bilkent University
gudukbay@cs.bilkent.edu.tr

Abstract

We propose a data-driven approach for measuring, validating and optimizing crowd simulation systems by learning parameters from real-life videos. We discuss the common traits of incidents and their video footage suitable for the learning step. We then demonstrate the learning process in a real-life incident that happened in 2015, Ankara, Turkey. We reanimate the incident with an existing emotion contagion and crowd simulation framework and optimize the parameters that take role in defining agent behavior with respect to the data extracted from the video footage of the incident.

Keywords: emotion contagion, crowd simulation, parameter learning, data-driven optimization

1 Introduction

When people congregate, they sometimes engage in spontaneous, homogeneous and irrational group behavior, losing their sense of identity. This phenomenon is known as collective (mis)behavior. Social psychology literature introduces various theories to explain collective crowd behavior. According to Brown [1], crowds can be classified under two categories as *audiences* and *mobs* depending on the existence of observable unified behavior. In both categories, the crowd members share a common goal, unlike pedestrians on a street who happen to be coincidentally at the same place at the same time. A crowd is called an audience if the group stays calm and relatively passive, such as students in a classroom or tourists visiting a historical building. On the other hand, mobs show active reactions, as in protests or hunger riots.

One of the most influential factors in the emergence of 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 [2]. 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.

Crowd simulation literature involves various techniques to validate the behaviors of virtual agents [3]. Although many studies exist about evaluating the quality of a crowd simulation system by considering human expert opinions [4] or some numeric metrics [5], we still need a universal, objective, quantitative and reusable method for validating crowd simulation models. Thus we can formally define future improvements to existing simulation systems and compare different systems under different scenario cases. In order to address this issue, we propose a data-driven approach for mimicking real crowd behaviors, learning parameters affecting crowd behavior and finally validating crowd simulation systems according to their fidelity to real behaviors. We apply this approach to the epidemiological emotion contagion framework proposed by Durupinar et al. [6]. 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 Durupinar 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.

The contributions of this paper are as follows:

- We propose a data-driven, quantitative and reproducible method for evaluating crowd simulation systems.
- We explain how real-life incidents can be utilized for evaluation and improvement of crowd simulation systems.
- 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 a contemporary incident and apply our proposed approach to an existing emotion contagion and crowd simulation system.

The rest of the paper is organized as follows. In Section 2, we discuss the related work in emotion contagion, crowd simulation and evaluation studies. In Section 3, we explain the proposed parameter learning framework and necessary steps to analyze crowd videos before using them for the optimization process. In Section 4, we explain the real-life incident in Ankara that we have used for learning the parameters of the Durupinar emotion contagion model. In Section 5, we demonstrate our framework on the Ankara attack and the implementation steps for virtually recreating the incident and optimizing the underlying emotion contagion model. In Section 6, we summarize our work, draw conclusions and discuss future improvement ideas.

2 Related Work

Ramos et al. [4] present a computational model for crowds showing emotional responses via body movements and emotion contagion and discuss their analysis on the role of background perception in emotion contagion. The model uses emotions to simulate body movements. Different from our data-driven approach, they use human expert opinions for evaluation.

Lemercier et al. [7] propose a crowd simulation model in which agents adapt to their environments and show different behaviors in different situations. They provide heterogeneity by providing virtual agents with different interaction possibilities with their environments. They use previously-tuned values for the parameters in their model for better results; however they do not discuss the metrics and the tuning process.

Charalambous et al. [5] describe a novel approach for data-driven evaluation of crowd simulation models. Similarly, Bera et al. [3], Lee et al. [8] and Lerner et al. [9] propose data-driven approach to learn how pedestrians behave, using trajectories extracted from real-life crowd videos. They optimize the underlying multi-agent simulation parameters and make simulated agents decide on actions according to the a database of real-life behaviors. Similar to our work, they analyze videos to extract trajectories of pedestrians and use this data for evaluation and improvement of their system. However, their material consists of high-quality videos of passive pedestrian audiences or synthetic movements of people in contrast to our focus on active and emotionally-driven mobs.

Singh et al. [10] present a numerical framework for evaluating crowd simulation systems, including metrics, scoring methods and test cases. Similar to our work, they score crowd simulation systems bases on various error functions. However, relative performances of crowd simulation systems based on these error metrics do not necessarily indicate realism performance since they are not based on real-life data. Likewise, Musse et al. [11] propose a model to quantitatively compare flow characteristics of two crowds by calculating 4D histogram distances.

Berseth et al. [12] analyze the effects of architectural decisions to the pedestrian flow and demonstrate a framework for optimal architectural element placement. The methodology of optimizing variables by tuning them according to a quantitative metric of simulation results is similar to our strategy. However, this architectural work fixes the crowd simulation system and modifies the environment, whereas we adapt the crowd simulation system parameters to the environment and events.

Bosse et al. [13] use a multi-agent based approach to simulate the emotion contagion phenomenon within crowds and propose the concept of negative emotion spirals in teams demonstrating the effects of ambient agents on the emotional responses of a team [14]. Later, they apply their emotion contagion model to simulate a real world incident that took place in Amsterdam and optimize some decision making parameters accordingly [15]. Tsai et al. [16] show their investigation results of emotion contagion phenomenon with various experiments. They introduce ESCAPES, a simulation tool designed for reproducing evacuation scenarios [17]. They compare two emotion contagion models proposed by Bosse et al. and Durupinar et al. and evaluate their impacts using the ESCAPES simulation tool. With their studies, Tsai et al. and Bosse et al. explain the first steps and benefits of quantitative evaluation, comparison and optimization of emotion contagion and crowd simulation models utilizing real world incidents. Yet, they do not discuss the properties of suitable incidents and materials; methods to track individuals in video footage and to project the tracking data to the real scene. For instance, in their reference videos, they track a relatively small group of 35 individuals in a massive crowd of size 20000. However, the Durupinar emotion contagion model is geared towards crowds with less than 200 agents. This improvement idea inspired us to apply data-driven optimization to the Durupinar model, focusing on the reproducibility of the process with suitable data.

Durupinar et al. [6] investigate the differences between audiences, which are passive crowds, and mobs, which are active crowds with emotional and seemingly homogeneous behavior. Their system facilitates the simulation of virtual environments and the specification of different groups of agents with varying personality characteristics and roles within a scenario. This allows easy manipulation of the impact of the events, personality traits, goals, and emotions on their behavior.

Durupinar emotion contagion model represents personality by the OCEAN (Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism) model [18], which describes five independent dimensions of human personality. The framework defines how each personality trait affects the development of emotions both for the individual itself and other people around. By specifying different 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 role in development of emotional reactions. The framework employs Ortony, Clore, Collins (OCC) model [19] 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 individuals and their attitudes towards objects. The framework utilizes the Pleasure-Arousal-Dominance model [20] to determine the current emotional state and make decisions. For emotion contagion in a crowd, they employ the social contagion model proposed by Dodds and Watts [2] with various augmentations.

3 Optimization and Parameter Learning

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 their trajectories. Then, we recreate the scene in the virtual environment of the target crowd simulation system 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.

3.1 Collecting Video Material

Our framework learns parameters of crowd simulation systems using videos of real-life incidents. In order to perform efficient parameter learning, the incident and its video should have the following properties:

- There should be available video footage of the incident, taken from a high ground, containing the moment the incident happens. The video should be taken from a static camera, like a security or surveillance camera, and at least some of the individuals in the incident should be trackable for some period of time.
- The incident should involve emotional responses of people in the scene.
- The size of the crowd should be suitable for the underlying crowd simulation model. For example in order to evaluate the Durupinar model, we aim for crowd sizes between 50 and 200 individuals.
- For easier projection, the scene should not have variations in the vertical space; i.e., the surface that the pedestrians move should be flat. Otherwise, it becomes difficult to project the camera image to the virtual scene. For a non-flat environment, a high resolution height map of the terrain is required and the barycentric coordinate transformation is not applicable.

3.2 Transfer from Video Pixel Coordinates to Scenario Coordinates

In order to synthesize the environment of the incident, we need to transform the pixel coordinates that constitute the output of the tracking process to the virtual scene. There are two approaches to this end: barycentric coordinate translation and camera parameter extraction. The camera parameter extraction relies heavily on a high amount of known reference points (around 60) [21], which are generally captured by placing a simple and easy to detect pattern, such as a checkerboard, in the scene. However, most of the videos of real-life incidents suitable for our framework are obtained by surveillance cameras; thus camera parameter extraction is not possible. Therefore, we focus on barycentric coordinate translation, which can be performed with only three reference points.

Barycentric coordinates allow us to describe a point in space with respect to other known reference points [22]. They represent a point as a weighted average of other known (reference) points. For convenience, the weights are usually normalized, i.e., they sum up to 1. One important property of barycentric coordinates is that they do not change with linear projections; i.e, when the space is scaled, translated or rotated, barycentric coordinates stay the same.

In our case, we can benefit from the barycentric coordinates within the two dimensional space. For our three reference points, we know the camera space coordinates as well as the 2D coordinates in the scene. We calculate the barycentric coordinates for all the pixels in the video frames with respect to our three reference points. Then, we calculate the coordinates of the point in the three dimensional scene using the barycentric coordinates.

Let our reference points be $rp_i = [x_i, y_i], i = 1, 2, 3$ and query point be $qp = [x_p, y_p]$ with the barycentric coordinates $[b_1, b_2, b_3]$. We can find the corresponding pixel coordinates 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 the following linear equation:

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

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 preprocessed to disable lens distortion.

3.3 Parameter Optimization

Our goal is to find the best combination of various parameters that play role in 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 events. 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:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & error(P) \\ \text{subject to} \quad & p_{imin} \leq p_i \leq p_{imax}, p_i \in P \end{aligned}$$

For searching the optimum parameter values in the search space, we run a simple independent parameter tuning algorithm (cf. Algorithm 1), similar to the work of Bosse et al. [15]. This method instantiates parameters to their minimum values at the beginning and iterates through the parameters optimizing one parameter at a time by calculating the error when the parameter being optimized is changed step by step while 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 effect of individual parameters on the overall results. By taking advantage of these properties, we can automatically improve the testing efficiency with each iteration by reducing the step size of sensitive parameters and increasing the possibility range of critical variables.

This technique also scales quite well in terms of number of parameters to be searched as well as 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 values have been tested, the parameter is assigned to the optimal value and the tuning process continues with the next parameter. After all parameters are tuned, the whole process restarts with the first parameter using the previously found optimum values.

Algorithm 1 Independent Parameter Tuning

```
$k$: number of tuning iterations
$P$: set of parameters
$s_i$: step size of parameter $p_i$
$min(p_i)$: minimum value of $p_i$
$max(p_i)$: maximum value of $p_i$
$val[p_i]$: current value of $p_i$
for $c = 1$ to $k$ do
    for all $p_i \in P$ do
        $val[p_i] \Leftarrow min(p_i)$
        $bestError \Leftarrow \infty$
        $bestVal \Leftarrow val[p_i]$
        while $val[p_i] \leq max(p_i)$ do
            calculate $currentError$
            $val[p_i] \Leftarrow val[p_i] + s_i$
            if $currentError < bestError$ then
                $bestError \Leftarrow currentError$
                $bestVal = val[p_i]$
            end if
        end while
    end for
end for
```

4 Ankara Attack Scenario

In order to demonstrate our proposed parameter learning method, we have used the video of the terrorist attack in Ankara Train Station on October 10, 2015, which we are going to refer from now on as 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 video 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, which is 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.

In the scene, 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 Figure 1.

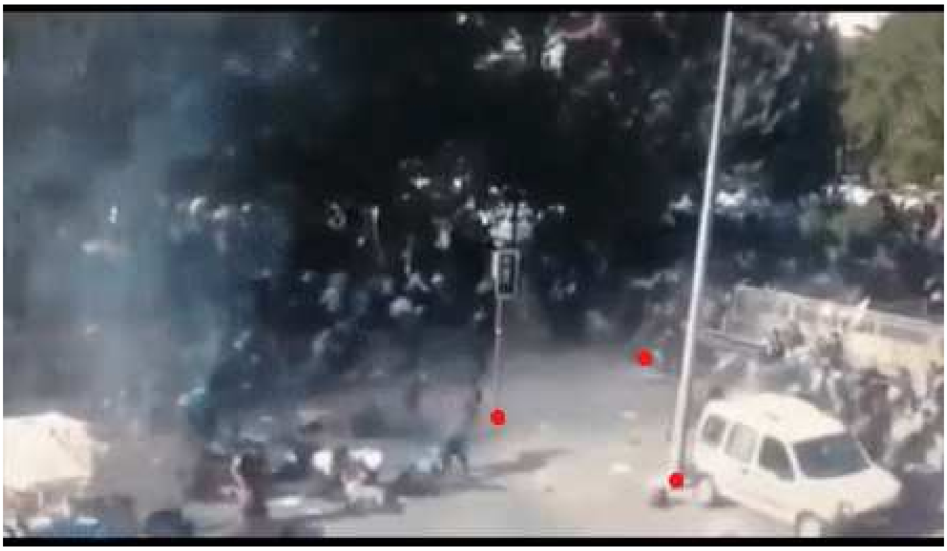

Figure 1: The surveillance camera footage of Ankara Attack with red dots indicating the reference points used for barycentric coordinate projection.

4.1 Surveillance Camera Footage

The video footage is taken from a city surveillance camera at the center of the gathering area, pointing to the west. In 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 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 11'th second of the original video. After that, the crowd starts running away from the center of 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 that the whole scene consists of the actual footage of the surveillance camera. Finally, we trimmed the video. The processed video has 446×250 resolution, 16 seconds of length with 12 frames per seconds and 434 kbps of bit rate, which sums to 826 kilobytes in size.

4.2 Tracking

In order to track people in the crowd, we automated the pedestrian trajectory extraction, as described in [3]. Because the quality of the video is low, the pedestrian detection methods perform poorly. After various attempts, we decided to track people manually, as it is done in [14]. We used an open source software, called “Tracker” [23] 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.

4.3 Virtual Scene

We created 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. 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

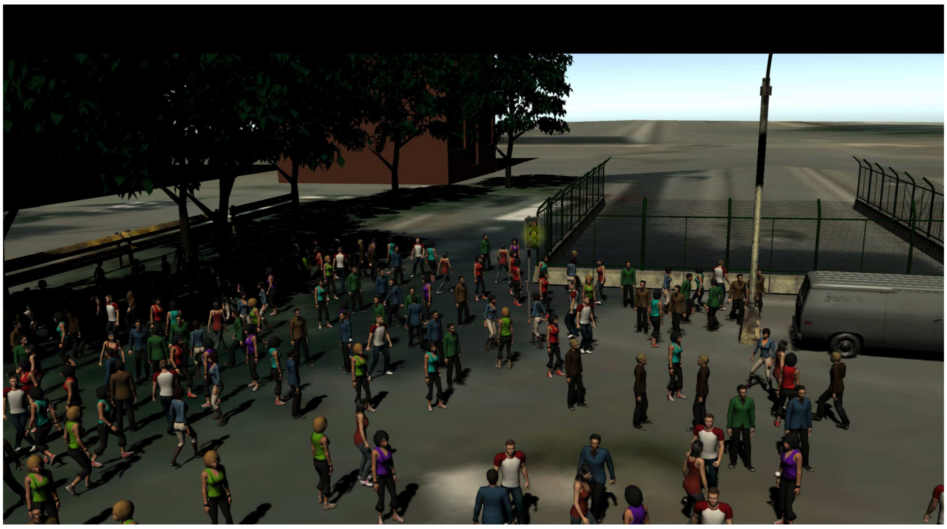

Figure 2: The virtual recreation of Ankara Attack.

for the train station building, trees between the train station and incident scene, the traffic light pole, the street light pole, 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. In this scenario, we set approving standards of individuals towards themselves as well as towards other agents. This is based on the fact that the people gathered in the area for a common goal, therefore they sympathize with each other. We also gave a displeased goal of waiting in the area since the gathering was about a protest.

We defined 15 parameters that can be tuned easily and impact the outcomes of the simulation results (cf. Table 1). Ten of these parameters are the mean and standard deviations of the five personality factors. Alongside the personality variation, we tuned the parameters for the weights of 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.

We also defined the error function for this scenario to be a trajectory matching error function, which is the sum of distances between of each tracked agent and its corresponding virtual agent. Optimizing this error function allows us to obtain a more realistic escaping pattern with more accurate running speeds, escaping directions and obstacle avoidance behavior of agents.

Table 1: Parameters used for emotion contagion.

Parameter	Min	Max	Step size
mean(O)	-0.8	0.8	0.2
std(O)	0.0	1.0	0.2
mean(C)	-0.8	0.8	0.2
std(C)	0.0	1.0	0.2
mean(E)	-0.8	0.8	0.2
std(E)	0.0	1.0	0.2
mean(A)	-0.8	0.8	0.2
std(A)	0.0	1.0	0.2
mean(N)	-0.8	0.8	0.2
std(N)	0.0	1.0	0.2
Wait goal	0.1	0.8	0.1
Explosion goal	0.3	0.8	0.1
Std. for self	0.05	0.8	0.1
Std. for others	0.05	0.8	0.1
Panic threshold	0.1	0.8	0.1

4.4 Results

We set the tuning environment for the virtual Ankara Attack scenario (see Figure 2) with the described 15 parameters, running each test three times and taking the median for the error. The tests run for four iterations so the parameters took turns four times in the tuning process, with another set of tuned parameters each time. Moreover, we run the testing scenario in $0.5\times$ slow motion in order to let Unity dedicate more processing time at each frame. With these numbers, the tuning process took about 10 hours.

In this scenario, we expected the most sensitive parameters to be the ones about survival in the scene, which are the standards that agents have for themselves, their fear threshold to start panicking and displeased goals about the explosion. In our results, as shown in Figure 3, we confirm that individuals' standards for themselves and goal about the explosion have the highest sensitivity but the panic threshold, contrary to our expectations, is one of the least sensitive parameters. When we investigate the scenario, we see that no matter how high it is, the panic threshold is exceeded in a very short time; therefore its change does not show a significant difference, making it a nonsensitive parameter.

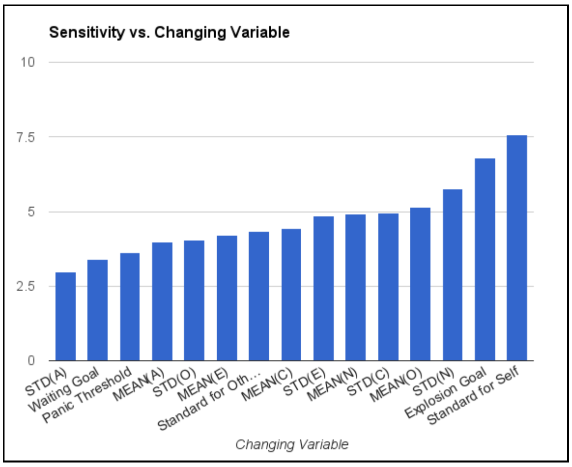

Figure 3: Sensitivity values of the tuned parameters.

We expected to see a reduction in the average sensitivity, i.e., total effect of changing a single parameter with each iteration because as other parameters are more and more optimized, the affect of changing a single parameter would decrease. The experimental results indeed show such a trend, as can be seen in Figure 4.

Finally, we expected to see a decrease in the observed error, as the parameters get optimized more and more. Figure 5 shows the decrease in the error with 60 tuning attempts, which are the searches performed in the value space of a parameter. Also, we expected to see the reduction in error fluctuation to decrease with each iteration because of a decrease in sensitivity of parameters, which is shown in Figure 6.

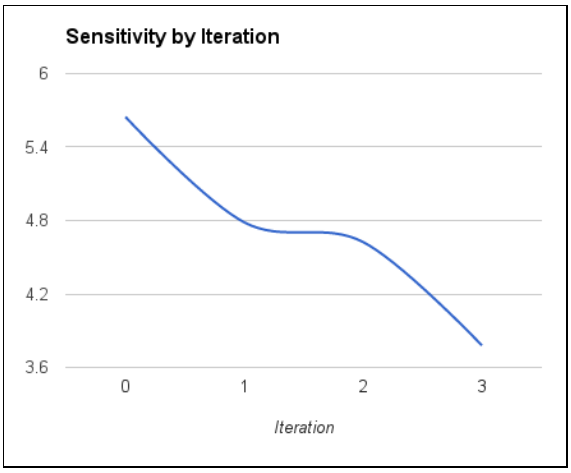

Figure 4: Sensitivity change with iterations.

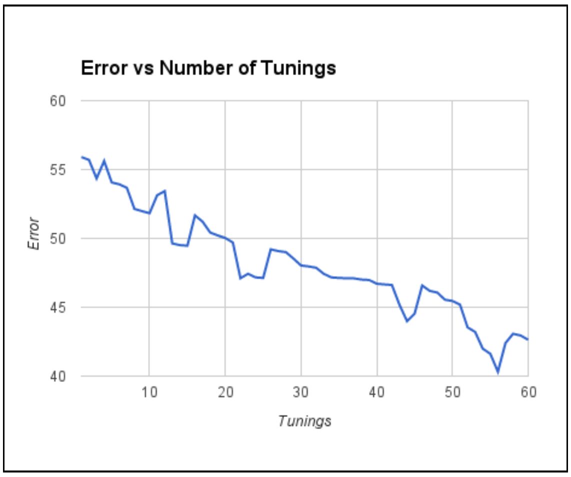

Figure 5: Change in error with each optimization step.

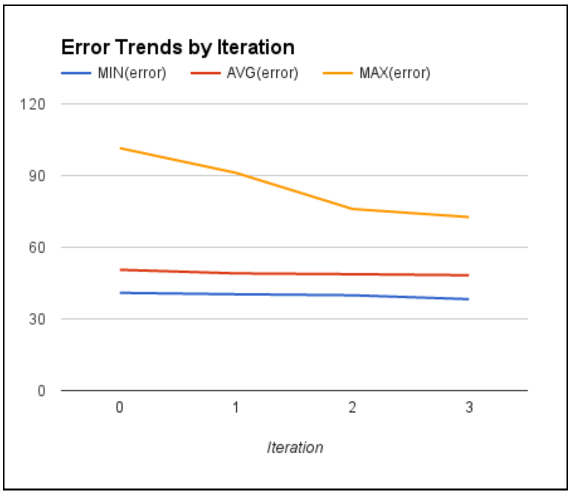

Figure 6: Change in minimum, maximum and average error with each iteration.

5 Conclusion, Limitations and Future Work

We propose a method for validating crowd simulation systems and learning emotion contagion parameters from real-life incidents. We explain the steps involved in data extraction from suitable videos of real-life incidents. We then optimize an existing emotion contagion model with respect to a subset of individuals in such an incident by learning personality parameters from real videos and tuning them one by one.

As a future work, one can repeat this work with more agents tracked for a longer time span. For this, multiple and clearer video tracks for such events are required. This can be possible by more access to press media, professional camera and surveillance footage.

Repeating the work done in this study for different kinds of incidents, such as Black Friday crowds, protests, incidents in crowded public transportation stops may show value for augmenting the learned personality distribution and improving the emotion contagion model.

By collecting media for real-life incidents and processing them, we can acquire solid evidence about the personalities of different cultures. By optimizing the learned parameters from multiple incidents in a region, we can extract the actual distribution of personalities in the area and use these learned personalities for the simulation of possible incidents in order 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 used 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 may 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 may affect each other. Therefore, more general and stable parameter optimization methods, like genetic algorithms or support vector machines, may produce faster and more accurate results. Moreover, the parameter estimation can also be expanded to the amount of emotion doses according to the reactions to events, goals and response thresholds.

We used the sum of distances between the tracked agents and their corresponding virtual agent as the metric to optimize. Although this metric is beneficial for estimating the running speeds, directions and reaction times, using more accurate metrics can lead to better understanding on the decisions made by the agents. One such metric could be the proportion of people doing action $a_k, 1 \leq k \leq n$ in a set of actions $\{a_i, i \in 1, \ldots, n\}$. 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 in the ground". Moreover, optimizing the parameters for multiple metrics at the same time may help produce more robust models.

References

[1] R. W. Brown. *The Handbook of Social Psychology, G. Lindzey (ed.),* chapter Mass Phenomena, pages 833–876. Addison-Wesley, Cambridge, MA, 1954.

[2] P. S. Dodds and D. J. Watts. A generalized model of social and biological contagion. *J. Theo. Bio.,* 232(4):587–604, 2005.

[3] A. Bera, S. Kim, and D. Manocha. Efficient trajectory extraction and parameter learning for data-driven crowd simulation. In *Proc. 41st Graph. Int. (GI '15),* pages 65–72. Canadian Inf. Proc. Soc., 2015.

[4] M. R. Carretero. Expression of emotion in virtual crowds: Investigating emotion contagion and perception of emotional behaviour in crowd simulation. Master's thesis, Dept. Comp. Sci. Comm., KTH Royal Inst. Tech., 2014.

[5] P. Charalambous, I. Karamouzas, S. J. Guy, and Y. Chrysanthou. A data-driven framework for visual crowd analysis. *Comp. Graph. Forum*, 33(7):41–50, 2014.

[6] F. Durupınar, U. Güdükbay, A. Aman, and N. I. Badler. Psychological parameters for crowd simulation: From audiences to mobs. *IEEE Trans. Vis. Comp. Graph.*, 22(9):2145–2159, 2016.

[7] S. Lemercier and J.-M. Auberlet. Towards more behaviours in crowd simulation. *Comp. Anim. Virt. Worlds*, 27(1):24–34, 2016.

[8] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. Group behavior from video: a data-driven approach to crowd simulation. In *Proc. ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, pages 109–118. Eurographics, 2007.

[9] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Comp. Graph. Forum*, volume 26, pages 655–664, 2007.

[10] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. Steerbench: a benchmark suite for evaluating steering behaviors. *Comp. Anim. Virt. Worlds*, 20(5-6):533–548, 2009.

[11] S. R. Musse, V. J. Cassol, and C. R. Jung. Towards a quantitative approach for comparing crowds. *Comp. Anim. Virt. Worlds*, 23(1):49–57, 2012.

[12] G. Berseth, M. Usman, B. Haworth, M. Kapadia, and P. Faloutsos. Environment optimization for crowd evacuation. *Comp. Anim. Virt. Worlds*, 26(3-4):377–386, 2015.

[13] T. Bosse, R. Duell, Z. A. Memon, J. Treur, and C. N. van der Wal. Multi-agent model for mutual absorption of emotions. In *Proc. 23rd Euro. Conf. Mod. Sim. (ECMS '09)*, pages 212–218. ECMS, 2009.

[14] T. Bosse, R. Duell, Z. A. Memon, J. Treur, and C. N. van der Wal. A multi-agent model for emotion contagion spirals integrated within a supporting ambient agent model. In *Proc. Int. Conf. Prin. Prac. Multi-Agent Sys. (PRIMA '16)*, pages 48–67. Springer, 2009.

[15] T. Bosse, M. Hoogendoorn, M. C. A. Klein, J. Treur, C. N. van der Wal, and A. van Wissen. Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. *Auto. Agents Multi-Agent Sys.*, 27(1):52–84, 2013.

[16] J. Tsai, E. Bowring, S. Marsella, W. Wood, and M. Tambe. A study of emotional contagion with virtual characters. In *Proc. Int. Conf. Int. Virt. Agnts. (IVA '12)*, pages 81–88. Springer, 2012.

[17] J. Tsai et al. ESCAPES: Evacuation Simulation with Children, Authorities, Parents, Emotions, and social comparison. In *Proc. 10th Int. Conf. Auto. Agents Multiagent Sys. (AAMAS '11)*, volume 2, pages 457–464. IFAAMAS, 2011.

[18] F. Durupinar, N. Pelechano, J. M. Allbeck, U. Güdükbay, and N. I. Badler. How the Ocean personality model affects the perception of crowds. *IEEE Comp. Graph. App.*, 31(3):22–31, 2011.

[19] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge Univ. Press, Cambridge, MA, 1988.

[20] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Curr. Psych.*, 14(4):261–292, 1996.

[21] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Patt. Anal. Mach. Int.*, 22(11):1330–1334, 2000.

[22] M. S. Floater. Mean value coordinates. *Comp.-Aid. Geom. Des.*, 20(1):19–27, 2003.

[23] D. Brown. Tracker video analysis and modeling tool for physics education, 2017. [Online; accessed 15-March-2017].