A Report of Trajectory Prediction based on Deep Learning

Chenxiao Tian
tian01@ads.uni-passau.de
University of Passau

Rita Akhmetova
akhmet04@ads.uni-passau.de
University of Passau

Yashu Wang
wang52@ads.uni-passau.de
University of Passau

Zubair Ahmed
ahmed08@ads.uni-passau.de
University of Passau

ABSTRACT

Pedestrians trajectories prediction has become a popular topic in recent years. Many approaches can be used to accomplish the prediction, including deterministic and non-deterministic ways. In this report we conduct research and discuss how to use a non-deterministic method, i.e. deep learning methods to solve the prediction problem. The algorithm we are going to use is a combination of SocialGAN and InfoGAN. Firstly, we train a model by the adopted algorithms. Secondly, during the research, we propose our research questions and try to answer them in the report.

PHASES ASSIGNMENT

- **Phase1** *Chenxiao Tian*
- **Phase2** *Zubair Ahmed*
- **Phase3** *Yashu Wang*
- **Phase4** *Rita Akhmetova*

INTRODUCTION

The innate abilities that human beings possess to process complex things effortlessly in daily life are impressive. To translate even the fractional part of one of these abilities of the human being into a machine is a challenging task in itself. One such ability of human beings is to navigate in a social environment. For example, when we walk in a crowded public space we follow a large number of common-sense rules and social etiquette. Which includes respecting the personal space of others, yielding right-of-way, avoiding walking through the people belonging to the same group, taking the shortest or safer path to the destination, and much more.

This ability of ours in the field of technology is commonly known as Human/Pedestrian trajectory prediction. The task of predicting human trajectories is crucial for current and future technological advancements. There are already many applications that make intensive use of modeling the pedestrians motion data and social interactions, e.g. infrastructure design (urban safety, city planning) [4, 9, 13], traffic operations [11], autonomous driving, are just a few to name. Typically, this modeling is performed in an offline manner by gathering the pedestrian motion data beforehand and completing the analysis to make decisions for the improvements in the environment. In modern world applications, however, one needs to perform this in real-time, allowing one to predict the pedestrian moves and infer their short or mid-term intentions in the environment. That allows to take preventive actions and trigger the alarm for the monitoring system in critical real-time decision-making applications. As in the case of autonomous driving, finding the intention of the pedestrians surrounding the car is of paramount importance in avoiding collisions. The task of modeling social interactions is extremely challenging as there exists no fixed set of rules which govern human motion. Here we refer to this task of predicting the human motion as Pedestrian trajectory prediction.

Many approaches have been proposed and developed to solve this complex task. In general, they can be classified into two categories: deterministic and statistical. The deterministic methods use hand-crafted functions based on certain observable conditions, such as Newton's laws of motion (which use velocity and acceleration to calculate position) and shortest paths (with the assumption that humans prefer the shortest path to the target position), etc., to generate human motion trajectories. A far-reaching example is social forces, a model proposed by Helbing and Molnar [10] based on equations describing the relationship between main effects (including attractive forces from goal and repulsive forces from other agents and obstacles) and human motion. Yi [17] built a model to calculate the optimal path for humans based on the formulated energy map. On the other hand, statistical ways rely on learning patterns from data through various methods, such as neural networks, Hidden Markov Models, etc. In 2015, Zhou et al. [18] build a linear dynamic system, applying the Expectation-Maximization (EM) algorithm to estimate parameters, to learn motion patterns in crowded scenes. Altché [2] proposes a method that predicts the trajectory on the highway using Long Short-Term Memory (LSTM). Alahi et al. [1] give a sequence model based on LSTM as well as a social pooling that aggregates the human-human interaction in a scene. With the vast amount of data available today, these methods can model complex situations that are difficult for humans to observe, which is valuable information for predicting the behavior of pedestrians. Thus, this approach is gaining more and more popularity in the research field.

Over the past few years, the statistical approaches have really helped us make better predictions about human trajectories. However, these methods mentioned previously learn only the pattern of human motion from data. Predicting human trajectory is a complex task. This is because both internal and external stimuli, such as intentions and other directly or indirectly observable influences, can affect human motion, as mentioned in the survey [16]. In addition to the location, which is usually recorded in the dataset, many factors that are not explicitly recorded in the dataset, such as speed, direction, or even not recorded, such as route and human intent. Recent researches have shown that Generative Adversarial Network (GAN) can better capture these uncertainties with latent space and thus naturally preserve multimodality. Gupta et al. [8] used GAN

Data Science Lab, 2021, Uni Passau
.

and a Pooling Module to predict socially acceptable trajectories and found that certain directions in the latent space are related to direction and velocity. What is more, the study of Amirian et al. [3] has shown that InfoGAN, an information-theoretic extension to the Generative Adversarial Network [5], partly improves the performance on commonly used datasets that have the largest variance in the prediction distribution, while still leaving some room for improvement.

Even though these researches give various effective models that fulfill the prediction task and attempt to encompass hidden aspects that influence the trajectory, they have not disentangled these factors in the latent space. Suppose we know the factors that affect pedestrians' trajectories and apply these factors in specific scenarios. We can obtain better performance of prediction on various distributed datasets and mitigate the limitations of the observed data. Therefore, we decide to consider the hidden factors behind different datasets.

In this study, our aim is not to improve the prediction of human motion but to bridge the gap of not disentangling the factors that influence human motion trajectories in the GAN model. We are interested in the relationship between human trajectories and those hidden factors that are not described by the data. We want to disentangle these factors from the latent space, and, ideally, with these factors, we can have a general model that works on different datasets. We assume that different datasets have different static environments, and so the data in a dataset share some specific common features. We consider three factors that may exist between different datasets: (that can be different between different datasets): obstacles (obstacles information such as the presence of static obstacles and the coordinates), maps (geometry and topology), and semantics (environment semantics such as no-go-zones, crosswalks, sidewalks, or traffic lights), which are the cues (in a static environment) that influence human trajectories, denoted by the survey [16].

We summarize our research questions as follows:

- What factors we can obtain that influence human trajectories?
- Can the factors we consider describe the variances between datasets? That is, with these factors input to our model, can our model gain better performance for different datasets?

Based on the problems, we propose to develop a controllable generative model to predict human motion. It can be controlled by factor c to have different static environments. We demonstrate that human movement is influenced by these three factors that we consider in a static environment. Also, by inputting different factors in static environments, our model can achieve better performance on different datasets.

1 PROBLEM STATEMENT

In this paper, our goal is to develop a controllable generative model to predict pedestrian trajectories. Consider the problem of predicting the future trajectory of each pedestrian. Let (x_i^t, y_i^t) denote the position of the i pedestrian at time t, and a sequence of coordinates $[(x_i^t, y_i^t), (x_i^{t+1}, y_i^{t+1}), ..., (x_i^{t+n}, y_i^{t+n})]$ denote the trajectory of pedestrians from time t to $t+n$.

Given the observed trajectory of n_{obs} steps $X_i^t = [(x_i^t, y_i^t), (x_i^{t+1}, y_i^{t+1}), ..., (x_i^{t+n_{obs}}, y_i^{t+n_{obs}})]$, with certain controllable factor c and random variable z, we want to fit a function to generate the prediction of trajectory for the next n_{pred} steps $Y_i^t = [(x_i^{t+n_{obs}+1}, y_i^{t+n_{obs}+1}), (x_i^{t+n_{obs}+2}, y_i^{t+n_{obs}+2}), ..., (x_i^{t+n_{obs}+n_{pred}}, y_i^{t+n_{obs}+n_{pred}})]$. That is

$$Y_i^t = f(X_i^t | c, z)$$

The prediction Y_i^t is controllable by the vector c, where consist of (c_1, c_2, c_3). So we can control the factors of obstacles, maps, and semantics respectively. These factors are independent of each other. They may vary depending on the data set and time.

2 DATA ACQUISITION & PRE-PROCESSING

This section discusses the dataset used for pedestrian trajectory prediction. Since the trajectory prediction is a data-driven task, and the data-driven task requires its data to be available in quantity with sufficient quality. Data for pedestrian trajectory prediction can be obtained in two different formats: either in image coordinates or in real-world coordinates. Image coordinates mean that each pedestrian is represented with the pixels it occupies in an image from the camera, whereas real-world coordinates mean that each pedestrian is represented by its position in meters (basic metric unit of length) with origin in an arbitrary point of the world.
The choice of the coordinate format selection depends on the type of the application: image coordinates are mostly used in video surveillance applications, whereas real-world coordinates are used in autonomous driving, robotic, or other similar applications. As we focus on the prediction of pedestrian trajectory in real-world scenarios therefore our datasets are using real-world coordinates. Since acquiring new data for the research is a very difficult and expensive task. Often ETH and UCY are the most commonly used datasets in different research works, due to their public availability and use of real-world coordinates. Hence we are also using the ETH and UCY datasets in this research.

2.1 Data acquisition

The most commonly used datasets for pedestrian trajectory prediction are ETH and UCY datasets. The BIWI Walking Pedestrians (also named ETH Walking Pedestrians [EWAP]) dataset referred to as ETH [15] is the research work of Pellegrini et al. from ETH Zurich University, this dataset is comprised of two scenes (namely eth and hotel) taken from a bird's eye view. In total the dataset has 785 different pedestrians 365 and 420 for eth and hotel respectively. The pedestrian position is annotated at 2.5fps in both datasets that is every 0.4 seconds of a trajectory. Figure 1 represents each scences from the ETH dataset.
Whereas the UCY dataset [14] is the research work of Lerner et al. from the University of Cyprus. The dataset is comprised of three scenes (univ, zara1, and zara2), also taken from a bird's eye view. In total it contains trajectories of more than 1100 pedestrians containing 850, 148, and 204 for univ, zara1, and zara2 respectively. Just like ETH, the UCY dataset is also annotated with pedestrian positions every 0.4 seconds. Figure 2 represents each scences from the UCY dataset.
These two datasets are often used in combination: in total, they contain five scenes (eth, hotel, univ, zara1, and zara2), with more than 1800 pedestrian trajectories. For training and testing purposes we use leave-one-out approach: basically, the model is first trained

on four scenes and tested on the fifth, and the procedure is repeated five times, once for each scene.
The original datasets can be downloaded directly from the official download links (ETH - BIWI Walking Pedestrians dataset) [1], and (UCY - Crowded Data) [2].

Figure 1: EWAP[1] datasets scenes - left image represents scene from eth, while right from hotel dataset [14]

Figure 2: UCY[2] datasets scenes - left image represents scene from univ, while right image from zara1 and zara2 dataset [15]

2.2 Data preprocessing

In this research, we use ETH and UCY datasets for the implementation of InfoGAN based trajectory prediction model. The dataset that is available from ETH is provided in a text file where each line represents frame number in the scene, pedestrian id, pedestrian positions, and velocities in x y z coordinates in the respective frame, the data is formatted as:

[frame_number pedestrian_ID pos_x pos_z pos_y v_x v_z v_y]

The metric used is 'meters' for the positions and velocities. The dataset also includes the homography matrix used to calculate the metric in real-world coordinates. However, pos_z and v_z (direction perpendicular to the ground) are not used. The data from the ETH is enough for the modeling and does not need preprocessing for our research, other then rescaling (min-max normalization).
The dataset availabe from UCY is however uses different formating, which require preprocessing to format it to ETH dataset. However we consider it to be out of the scope of this research and hence we directly use the UCY data provided by the SocialWays[6] availabe via a publicaly shared Link [3] preprocessed to match the ETH dataset format.

[1]https://data.vision.ee.ethz.ch/cvl/aem/ewap_dataset_full.tgz
[2]https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data
[3]http://www.dropbox.com/sh/lh1s41d1pqp8cbx/AAD4sB1JAiZIkCL7LHht-S4Ca

3 MODELING

3.1 Methodology

Our experiments are based on Social GAN by Agrim Gupta et al [8]. It utilizes generative models to overcome the limitations of previous work. They build an LSTM-based encoder-decoder GAN capable of capturing the multimodality of trajectory prediction. In addition, they propose a novel pooling mechanism that enables the network to learn social norms among humans, and a k variety loss that motivates the network to generate diverse samples.

In the following subsections, we will describe the key methods of our experiments.

3.1.1 Generative Adversarial Networks.

According to the research of Generative Adversarial Nets [7]: a Generative Adversarial Network (GAN) consists of two network components, a discriminator D and a generator G, which compete with each other. G takes the input noise variable z and generates the sample $G(z)$, D takes the generated sample or training data as input x and predicts the probability $D(x)$ that x comes from the data and not generated by G. D is trained to maximize the probability of assigning correct labels to training samples and generated samples, while G is trained to minimize the correctness of D. In other words, D and G play a min-max game with the value function $V(G, D)$.

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{\text{data}(x)}}[\log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

In our case, for pedestrian trajectory data, the generator is trained to generate possible future trajectories that have a distribution similar to the training data, given certain previously observed trajectories, while the discriminator learns to distinguish the rationality of the generated paths. These two networks are trained simultaneously. As the discriminators are learned, the generators are improved.

3.1.2 InfoGAN.

Information Maximizing Generative Adversarial Networks (InfoGAN) enable GANs to unsupervised learn disentangled representations by including information-theoretic regularization. It claims that the mutual information between a subset of the generator input (i.e., latent code c decomposed from input z) and the generator output should be high. The lower bound of the mutual information $I(c; G(z, c))$ can be obtained as

$$I(c; G(z,c)) \geq \mathbb{E}_{c\sim P(c), x\sim G(z,c)}[\log Q(c|x)] + H(c)$$

, where only $Q(c|x)$ is an active variable and x come from $G(z, c)$.

According to InfoGAN [5], the additional work compared to GAN includes: G takes the noise variable z and the latent code c as input to generate the sample $G(z, c)$. The additional auxiliary network Q accepts the input $G(z, c)$ and outputs the parameters for $Q(c|G(z, c))$. G and Q are trained to maximize $Q(c|G(z, c))$. So, the min-max game is updated with a regularization:

$$\min_{G,Q} \max_D \mathbf{V}_{\mathbf{InfoGAN}}(D, G, Q) = \mathbf{V}(D, G) - \lambda I(c; G(z, c))$$

In our work, we added parameterizable categorical latent codes and continuous latent codes to the generator, where the number of latent codes and the dimensionality of each categorical latent code are parameterizable. We implement the auxiliary network Q in a way that it shares all network layers with D except the last layer, and the final network of Q is a multilayer perceptron that accepts the features extracted by the shared network D_s, and whose output depends on the distribution of the latent code. The implemented model is shown in Figure3.

For the categorical latent code, we simply make it conform to the categorical distribution, so the output can be a softmax representation.

$$Q_{\text{head}}(D_s(G(z,c))) = MLP(D_s(G(z,c)); W_Q) = \hat{c}$$

For a continuous latent code, we make it conform to a Gaussian distribution, so the output can be Gaussian parameters, i.e., the expected value μ and the variance σ^2. For convenience, we implement the output as a logarithmic variance, so that the variance can be forced to be positive using an exponential transformation when calculating the loss.

$$Q_{\text{head}}(D_s(G(z,c))) = MLP(D_s(G(z,c)); W_Q) = [\ \mu\ \log(\sigma^2)\ ]$$

where $[\ \mu\ \log(\sigma^2)\ ]$ is the concatenation of μ and $\log(\sigma^2)$

3.1.3 Loss.

Our experiments attempt to disentangle the latent code so that the latent code can correspond to the semantic features of the data. To enrich our expression, we allow two types of latent codes, categorical latent codes and continuous latent codes. On the basis of that, we can introduce a series of semantic factors. For example, velocity and direction can be represented as continuous latent codes and scenes can be expressed as categorical latent codes. For map/obstacle information, the image embedding of the background image (the background image of the video recording trajectory data) can be used as a sequence of continuous latent codes.

For categorical latent code c (which conforms to the categorical distribution), we can use cross entropy as a loss function to maximize the information between the code c and the trajectory generated by code c.

$$\mathcal{L}_{\text{disc}}(c, Q(G(z,c))) = CE(c, Q(G(z,c)))$$

For a continuous latent code c (which conforms to a Gaussian distribution), we can use the negative log likelihood loss to maximize the information between the code c and the trajectory generated by code c.

$$\mathcal{L}_{\text{cont}}(c, Q(G(z,c))) = NLL(c, Q(G(z,c)))$$

The categorical latent code loss and the continuous latent code loss can be scaled by the hyperparameters λ_{disc} and λ_{cont}, respectively.

3.2 Experimental Setup

The first goal of our experiment is not to deteriorate the original SocialGAN network. Because we just added some latent codes to the original network, we did not reduce the dimensionality of the noise vectors or make any other changes. The second goal is to disentangle some factors with our implementation of InfoGAN, which is also the main goal of our experiments. To achieve these goals, we introduced the following metrics, baselines, and measures of the disentanglement.

3.2.1 Evaluation Metrics.

We evaluate our work with the following metrics that have been used extensively in previous work [1, 8, 12].

Average Displacement Error (ADE): Average Euclidean distance between ground truth and the prediction over all predicted time steps.

$$\text{ADE}_T = \frac{1}{T}\sum_{t=0}^{T} \|\langle x^t, y^t\rangle - \langle \hat{x}^t, \hat{y}^t\rangle\|_2$$

Final Displacement Error (FDE): Euclidean distance between ground truth and the prediction at the final time step.

$$\text{FDE}_T = \|\langle x^T, y^T\rangle - \langle \hat{x}^T, \hat{y}^T\rangle\|_2$$

Consistent with Social GAN [8], we performed experiments with T equal to 8 or 12, which is also a common choice.

3.2.2 Baselines.

Since our experiments are based on Social GAN, we directly choose Social GAN as our baseline for comparison. We followed the same strategy as Social GAN, using the leave-on-out method for training (training on 4 sets and testing on the remaining sets). In particular, we can set some parameters regarding Social GAN. We can train a model containing social pooling net, using k variety loss. Then in testing, we generate N samples for the next T steps trajectory and report the minimum ADE and FDE in these N samples. In our experiments, we chose two specific settings of Social GAN as the baseline:

- SGAN-20VP-20 (training Social GAN with social pooling module using 20 variety loss, and generating 20 samples in testing)
- SGAN-20V-20 (training Social GAN using 20 variety loss, and generating 20 samples in testing)

3.2.3 Disentanglement Measuring.

To measure the disentanglement of our implementation of InfoGAN, we utilized latent traversal. We generate traversals in this way: we first randomly select N trajectories. For each of these trajectories, we traverse over a single latent code while keep other units fixed, and then we generate trajectory for each traversal value of this latent code with the corresponding inputs (noise, other codes and this traversed code) and plot them. Then we obtain the traversals of a single latent code over N samples.

- For categorical latent codes with d categories, the traversal is over d categorical latent variables representing each category.
- For a continuous latent code, the traversal is over range (-2,2). We perform linear interpolation to get 10 points between (-2, 2), which can be the common values of the latent code that conforms to distribution $\mathcal{N}(0,1)$.

We inspect the latent traversal for N samples. We expect to find some common patterns of variations in a single latent codes, which means that the code is disentangled and encode with semantic meaning.

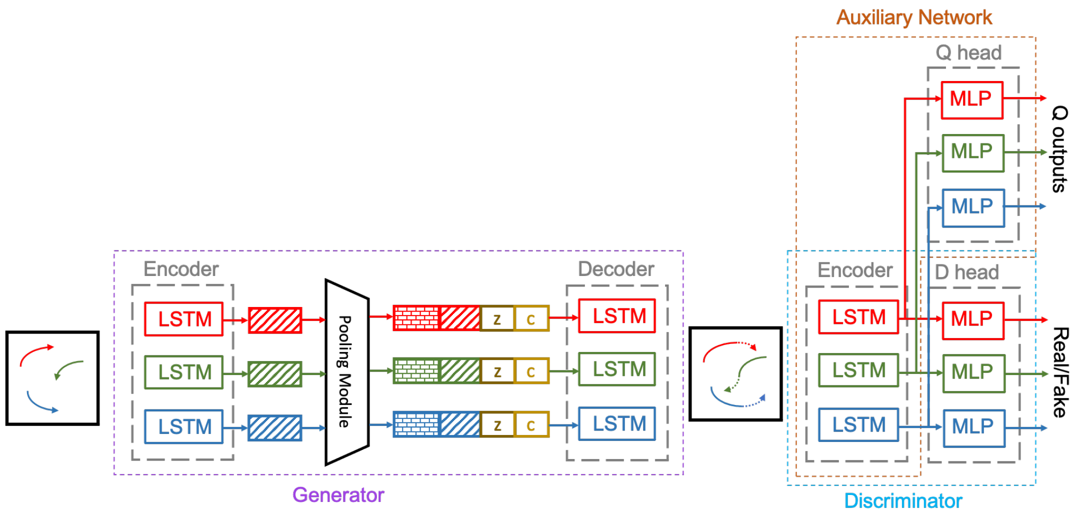

Figure 3: Model Structure
The latent code c is concated to the input of the generator decoder. The auxiliary network Q is implemented in the way that Q shared networks with D and the head of Q takes the extracted features from shared network as input.

EVALUATION

3.3 4.1 Results

Table 1: SemEval-2019 task 6: Subtask A

	Dataset	SGAN + InfoGAN	SGAN
ADE	NULI	BERT	0.829
2	vradivchev_anikolov	BERT-Large	0.815
3	UM-IU@LING	BERT-Base, Uncased	0.814
4	Embeddia	BERT	0.808
5	MIDAS	ensemble (CNN, Bi-LSTM)	0.807
6	BNU-HKBU	BERT-Base, Uncased	0.806
7	SentiBERT	-	0.804
8	NLPR@SRPOL	Transformer	0.803
9	YNUWB	CNN	0.802
10	LTL-UDE	BERT	0.790

Table 2: SemEval-2020 task 12 data distribution

	Subtask A			Subtask B			Subtask C			
	OFF	NOT	total	TIN	UNT	total	IND	GRP	OTH	total
training	1448861	7640279	9089140	149550	39424	188974	120330	22176	7043	149549
test	1080	2807	3897	850	1072	1922	580	190	80	850

REFERENCES

[1] Alexandre, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. (2016), 961–971. https://doi.org/10.1109/CVPR.2016.110

[2] Florent Altché and Arnaud de La Fortelle. 2017. An LSTM network for highway trajectory prediction. (2017), 353–359. https://doi.org/10.1109/ITSC.2017.8317913

[3] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. 2019. Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs. *CoRR* abs/1904.09507 (2019). arXiv:1904.09507 http://arxiv.org/abs/1904.09507

[4] Stephen C. Bitgood. 2006. An Analysis of Visitor Circulation: Movement Patterns and the General Value Principle. *Curator: The Museum Journal* 49 (2006), 463–475.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 2180–2188.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *CoRR* abs/1606.03657 (2016). arXiv:1606.03657 http://arxiv.org/abs/1606.03657

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc., 2672–2680. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[8] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social GAN: Socially Acceptable Trajectories With Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] André Hanisch, Juri Tolujew, Klaus Richter, and Thomas Schulze. 2003. Online simulation of pedestrian flow in public buildings. *Proceedings of the 2003 Winter Simulation Conference, 2003.* 2 (2003), 1635–1641 vol.2.

[10] D. Helbing and P. Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* (1995), 4282–4286. https://doi.org/10.1103/PhysRevE.51.4282

[11] Andreas Horni, Kai Nagel, and Kay W. Axhausen. 2016. The Multi-Agent Transport Simulation MATSim.

[12] Namhoon Lee, Wongun Choi, Paul Vernaza, Chris Choy, Philip Torr, and Manmohan Chandraker. 2017. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. 2165–2174. https://doi.org/10.1109/CVPR.2017.233

[13] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by Example. *Computer Graphics Forum* 26 (2007).

[14] A. Lerner, Yiorgos L. Chrysanthou, and D. Lischinski. [n.d.]. Crowds by example. , 655–664 pages. Issue 3. https://doi.org/10.1111/j.1467-8659.2007.01089.x

[15] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*. 261–268. https://doi.org/10.1109/ICCV.2009.5459260

[16] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. 2020. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research* 39, 8 (2020), 895–935.

[17] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2015. Understanding pedestrian behaviors from stationary crowd groups. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3488–3496. https://doi.org/10.1109/CVPR.2015.7298971

[18] Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. 2015. Learning Collective Crowd Behaviors with Dynamic Pedestrian-Agents. 111, 1 (2015). https://doi.org/10.1007/s11263-014-0735-3