# Destination Flow for Crowd Simulation

Stefano Pellegrini[1,⋆], Juergen Gall[2], Leonid Sigal[3], and Luc Van Gool[1]

[1]ETH Zurich, [2]MPI for Intelligent Systems, [3]Disney Research Pittsburgh

**Abstract.** We present a crowd simulation that captures some of the semantics of a specific scene by partly reproducing its motion behaviors, both at a lower level using a steering model and at the higher level of goal selection. To this end, we use and generalize a steering model based on linear velocity prediction, termed LTA. From a goal selection perspective, we reproduce many of the motion behaviors of the scene without explicitly specifying them. Behaviors like "wait at the tram stop" or "stroll-around" are not explicitly modeled, but learned from real examples. To this end, we process real data to extract information that we use in our simulation. As a consequence, we can easily integrate real and virtual agents in a mixed reality simulation. We propose two strategies to achieve this goal and validate the results by a user study.

## 1 Introduction

The modeling of human crowds is of enormous interest for a multitude of applications, ranging from gaming, over movie effects, to evacuation simulators and urban planning. Humans exhibit a huge variety of motion behaviors that is not trivial to reproduce in virtual characters. Following the terminology of [17], we distinguish among three levels of motion behavior: goal setting, steering and locomotion. In this paper we will focus on goal setting and, partly, on steering.
Our goal is to simulate crowds within a specific context. Depending on the environment there will be different motion patterns. Whereas people may leisurely walk in a park, they probably are quite hasty in a business district. Different environments, which could also be indoor, will also contain different sources, attractors, and sinks where people resp. appear, often go to, or disappear. All these factors must be accounted for when animating a virtual crowd. Manual specification (e.g., scripting) is common, but also very tedious and time consuming. As alternative, data-driven approaches [8, 9] extract real trajectories from an actual crowd and use these. Data-driven approaches, however, do not generalize well and are difficult to adapt to scene changes.
In this paper, we therefore propose to combine rule-based steering models with data-driven goal selection to minimize the manual work for animators while still being able to generalize to scene changes. Human steering models have been studied for several decades in different disciplines. When building a rule-based steering model, rather than a data-driven approach, one can conveniently integrate this knowledge into simple rules that capture general properties of human

⋆ This project has been funded by TANGO European Project.

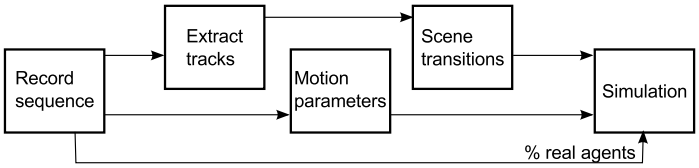


**Fig. 1.** Method workflow.

low-level motion. These properties are not expected to be valid only in a specific scene. The same model can therefore be employed in different situations by simply adapting the goal selection layer. In this layer, behaviors like "wait at the tram stop", "stroll-around" or "go and grab a coffee" are not explicitly scripted but learned from real examples. This said, user interaction is easy to integrate in the process, and arbitrary behaviors can be chosen.

Our approach only requires a few minutes (5 to 10) of video recording with real people moving in the scene. Their trajectories and group memberships are extracted by means of a semi-automatic tracker similar to the one in [15]. This video data is processed both to analyze lower level properties such as typical personal area [4] and to extract the regions in the scene that are relevant for crowd motion. Based on the extracted trajectories, the *destination flow* between the regions is learned, which is a probabilistic goal selection layer that extends the steering model. To carry out the actual agent motion, we choose to generalize the LTA steering model [14]. In particular, we generalize the avoidance component in that model, add a temporal horizon to the collision anticipation and add the realistic possibility of people walking in groups within the crowd. Finally, in order to maximally benefit from the real trajectories that one might have at one's disposal, we propose the use of mixed crowds, consisting of a blend of real and virtual trajectories.[1] At simulation time, the mix of real vs. virtual people is a free parameter, allowing the user to adapt crowd density (see Fig. 1).

**Contributions**: The first contribution of the paper is the introduction of the *destination flow*, a probabilistic goal selection layer that extends the steering model. In another contribution this paper proposes two different, possibly combinable, strategies to mix real and individual trajectories in the crowd simulation. Finally, we generalize the LTA steering model to fit our purposes.

## 2 Related Work

Autonomous agents are an active area of research since a long time[18, 5, 19, 22]. A detailed review can be found in [20].

Steering models have mostly been used for computer graphics animations and to simulate evacuation scenarios. However, other fields, such as robotics [21, 24] and computer vision [1, 14] recently developed interest for such models and came out with their own solutions.

In this work we will build upon LTA [14], an energy minimization based steering

[1] We will call real agents those crowd members the trajectory of whom is derived directly from captured data, and virtual agents those that are generated by simulation.

model. As other models [17, 16, 13, 24], LTA uses a linear velocity prediction, but in different ways. In particular LTA defines a smooth energy function that depends on the subject velocity. The smooth formulation of the energy function leads to a straightforward integration of multiple interactions and additional components (*e.g.* grouping), like for [5]. However, the choice of the next agent velocity does not depend on the absolute value of the function, as in the Social Force model [5], rather on the function shape (*i.e.* the location of the minima), making it easy to design and check the model properties. Also geometric models [24] share this advantage. Contrary to geometric models, however, LTA's energy function is differentiable and efficient gradient minimization techniques can be readily applied. LTA has never been tested before for crowd simulation. In this paper we improve and generalize the model to make it suitable for such purpose.

While the majority of the steering models have focused on individual motion, recently more and more authors propose to include the grouping feature into the simulation [11, 12, 7]. Using groups, but data-driven, is the approach of [8]. Here, state-velocity pairs are extracted from real data and at simulation time a combination of them is employed in order to simulate group behaviors. Also data-driven, a different kind of microscopic model has been proposed by [9]. This is an example-based model, that uses a database build with real world trajectories. [6] build a model of the crowd structure and motion, learning these features from real data. The crowd animation is carried out by selecting collision free trajectories that are consistent with the formation model. The authors show how models learned from different types of crowd can be combined and result in a blending of the original models.

Closely related to our work, is the study presented in Chapter 6 of [20]. The authors here first describe which scene information could be useful to reproduce a real scene in a simulation. However, the information extraction at this point is manual and requires a customization for each scene. Furthermore, they use real tracked trajectories to extract scene specific velocity fields. The velocity fields are then clustered and virtual agents at simulation time are driven by these fields. Our approach is different in that we compute region of interest and transitions among them, rather than extract velocity fields. This allows us to simulate in a stochastic manner complex scenes, where repetitive behaviors are possible. Last, we study the mixing of real and virtual agents [25] in the reproduced scene.

Another important aspect of an autonomous virtual agent is the goal selection. In [23] the intentions of different types of agents are represented by different flow-charts. In [12], when in the *autonomous mode*, the crowd responds to events following the rules specified by the animator. A Finite State Machine is used to model the goal selection layer both in [2] and in [3]. In these works, the region structure and the transition rules are manually specified. In our paper we reduce the concept of intention, or goal, to that of a destination in space and we let the transitions be learned automatically from the scene.

## 3 Steering Model

To simulate crowds we use the steering model presented in [14], termed LTA. In this model, the next velocity for an agent is computed by minimizing an energy function that reproduces avoidance and goal-seeking aspects of human walking. The key aspect of the model is that the decision of the next velocity is based on the distance of maximum approach $\mathbf{d}_{ij}$ between pedestrians $i$ and $j$. In this paper we generalize the model, with parameters $\lambda$, in three directions:

- We substitute the isotropic avoidance component $I_{ij}$ of the original formulation with an anisotropic one:

$$I_{ij} = \exp\left(-{\mathbf{d}_{ij}}^T \begin{bmatrix} \lambda_{I,1} & 0 \\ 0 & \lambda_{I,2} \end{bmatrix} \mathbf{d}_{ij}\right) . \tag{1}$$

- We add to the energy function a grouping term to allow representing groups of pedestrians walking together.

$$G_{ij} = {\mathbf{d}_{ij}}^T \begin{bmatrix} \lambda_{G,1} & 0 \\ 0 & \lambda_{G,2} \end{bmatrix} \mathbf{d}_{ij} . \tag{2}$$

- We introduce a temporal horizon $T$, to limit the computation of the $\mathbf{d}_{ij}$ to a realistic time interval.

The function that we minimize at each time step[2] for each agent $i$ becomes:

$$E_i = \sum_{j \in \mathcal{V}_i} w_{ij} I_{ij} + \sum_{j \in \mathcal{G}_{ij} \cap \mathcal{V}_i} w_{ij} G_{ij} + S_i \lambda_S + D_i \lambda_D , \tag{3}$$

where $w_{ij}$, $S_i$ and $D_i$ are the same weights, speed and destination components as defined in the original paper, $\mathcal{V}_i$ is the set of pedestrians that are visible to subject $i$ and $\mathcal{G}_i$ is the set of subjects that are in the same group of subject $i$.

## 4 Scene Transitions

The original LTA assumes that the destination $\mathbf{z}_i^t$ and the desired speed $u_i$ are known for each subject $i$. In order to be able to mimic the motion patterns in a specific scene, we propose to learn a probabilistic model for these quantities. To this end, we extract the trajectories of real people and the groups from a video with an interactive tracker [15] and label the static objects $\mathcal{O}^t$ manually. We use cameras from the top (Fig. 3) to capture the entire scene and to facilitate the tracker job. We also define a set of scene events, $\mathcal{E}$, and the subset $\mathcal{A}^t \subseteq \mathcal{E}$ of active events at time $t$. In the following, we assume that $\mathcal{A}^t$ and $\mathcal{E}$ are given.
Based on the trajectories, we model the desired speed $u_i$ by a normal distribution, where mean $\mu$ and variance $\Sigma$ are estimated from the observed velocities. As people do not walk with very low speed, we set an additional threshold of

[2] Note that we dropped the time dependency to reduce notational clutter.

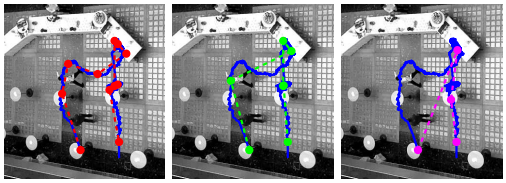

**Fig. 2.** The approximation of a real trajectory (in blue) for $\kappa = 1$ (red), $\kappa = 10$ (green) and $\kappa = 100$ (magenta). Note how a too small value of $\kappa$ will introduce too many corners while the large value will miss some.

$0.5ms^{-1}$ to avoid slow walkers being sampled from the tail of the speed distribution. In our model, the subjects in the same group share the same desired speed. To learn the sequence of destinations for each group of subjects, we need to know the *intentional* destination of the recorded pedestrians. To this end, we extract interesting points from the scene by segmenting the trajectories.

### 4.1 Segmenting the Trajectories

Given a uniformly sampled point trajectory $\mathcal{Q}_i = [\mathbf{p}_i^0 \dots \mathbf{p}_i^T]$, we are interested in a sequence of points, or *corners*, $\mathbf{p}_i^0 \dots \mathbf{p}_i^{t_c} \dots \mathbf{p}_i^T$ that split the trajectory in a sequence of sub-tracks. Each of these sub-tracks specifies a unit of motion that a subject should be able to undertake without any complex path planning operation. Since thresholding the velocity or the curvature of the trajectories turned out not to be robust, the problem is solved by a shortest path search in a graph, as in [10]. The graph is obtained by connecting each $\mathbf{p}_i^t \in \mathcal{Q}_i$ with all subsequent points. The cost of the transition from $\mathbf{p}_i^{t_a}$ to $\mathbf{p}_i^{t_b}$ is defined by

$$\gamma(t_a, t_b) = \kappa + \sum_{t=t_a}^{t_b} ||\mathbf{p}_i^t - (\mathbf{p}_i^{t_a} + \frac{t - t_a}{t_b - t_a}(\mathbf{p}_i^{t_b} - \mathbf{p}_i^{t_a}))||^2 \,, \tag{4}$$

where $\kappa$ is a fixed cost associated to each split to regularize the number of splits. The summation in Eq. 4 is the cost of approximating the portion of the trajectory from $t_a$ to $t_b$ with a straight line. The impact of the regularizer $\kappa$ is shown in Fig. 2. We set $\kappa = 10$ in this paper. The corners of a trajectory make the sequence of desired destinations and the movement from one corner to the next is a *transition*. Rather than assigning the corner directly to the scene, we extract $R$ interesting spatial *regions* from the scene and assign a corner to each of them. We distinguish between entrance, exit and transition corners and regions depending on whether they are at the beginning, in the middle or at the end of the trajectories. The corners are distributed within the region according to a distribution $\rho_r$, with $r = 1 \dots R$.
In our experiments, we use square cells to represent each region as shown in Fig. 3, left. The size of each region cell affects the destination flow. We found a good compromise at $2m$ edge size. We also define two special regions, *create* and *destroy* to handle agents initialization and termination, respectively. Note that the same solution can be used for different kinds of region shapes, not necessarily square cells and user interaction with shape and cell position is straightforward.

The distribution $\rho_r$ is estimated using a kernel density estimate over the region corners, with a normal kernel of $0.2m$ standard deviation.
In this paper, a group of subjects always share the same intention, and therefore the same destination region $r$. The splitting described in Sec. 4.1 can be generalized to a group of trajectories by simply averaging the transition costs in Eq. 4 of the subjects within the same group. In practice, due to the problems produced by groups that are at the edge of two cells, we count the transitions at the individual level. A region transition $r_a \rightarrow r_b$ is therefore defined by a pair of consecutive corners in a subject trajectory. The transition probability $p(r_b|r_a)$ is estimated by the normalized count of the transitions $r_a \rightarrow r_b$.
The probabilities, however, can change based on the events that are active. We therefore model the transition probabilities for each active event set $\mathcal{A}^t$ by accumulating in each $p^{\mathcal{A}^t}(r_a|r_b)$ only the transitions that are observed when the corresponding event set is active. We call the set of regions and transitions the *destination flow*.

### 4.2 Simulation

The simulation of a group $g$ of people starts with a transition from the *create* region to a entrance region $r_g$, according to the transition probability $p^{\mathcal{A}^t}(r_g|create)$. Every time a new transition $r_g$ is sampled for a group, for each subject $i_g$ a new destination point $\mathbf{z}_{i_g}^t$ is sampled from the destination region $r_g$, according to the distribution $\rho_r$. The agent motion to the destination $\mathbf{z}_{i_g}^t$ is demanded to the steering model of Sec. 3. Note that the agents are not performing any navigation task. The destination flow, made mostly of straight paths, partially replaces the navigation task.
A group $g$ reaches the destination when each subject $i_g$ in the group reaches its individual destination point $\mathbf{z}_{i_g}$, e.g. when the distance $d(\mathbf{p}_{i_g}^t, \mathbf{z}_{i_g}^t)$ is small. We define an indicator binary variable $n_g^t$, that is 1 when the group reached destination, otherwise it is 0. A group selects a new region $r_i$ (possibly the same) at time $t$ only when $n_g^{t-1} = 1$. More formally, the region transitions that determine the motion pattern of simulated pedestrians are modeled as

$$p(r_g^t|r_g^{t-1}, \mathcal{A}^t, n_g^{t-1}) = \begin{cases} \delta_{r_g^t, r_g^{t-1}} & n_g^{t-1} = 0 \\ p^{\mathcal{A}^t}(r_g^t|r_g^{t-1}) & \text{otherwise} \end{cases} \tag{5}$$

where $\delta_{a,b}$ is the Kronecker function that is 1 if $a = b$, and 0 otherwise. The dependencies of the variables described in this section are shown in the graphical model in Fig. 3. Note how the groups share the same quantities.

## 5 Mixed Reality

Adding real agents to a simulated scene, or the other way around, can be beneficial to enrich the simulation with specific, possibly actor-played, actions. The main problem is that the real agents are not aware of the simulated ones. Simply

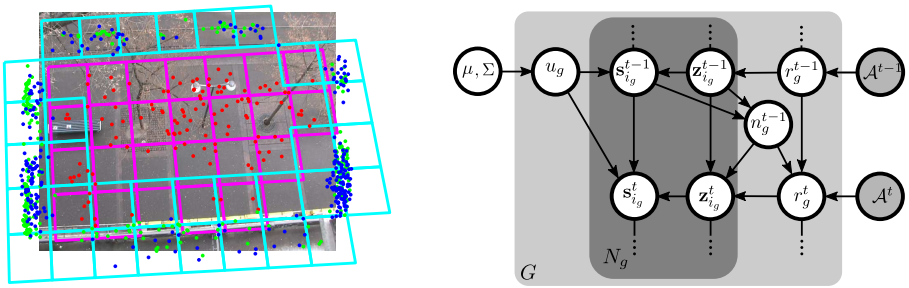


**Fig. 3. Left**: The grids used for the destination flow. We show the 3 kind of corners: entrance (green), exit (blue) and turn (red). We show the grid for turning points (magenta) and the two overlapping entrance and exit grid (cyan). Empty cells are not shown. **Right**: The dependencies in the goal selection layer. $\mathbf{s}_i^t$ is the state of subject $i$ at time $t$, given by her position and velocity. Shaded nodes are observed variables.

adding them together will result in general in unlikely configurations, especially when the density of the scene increases. We propose two possible solutions:
**Sampling:** Sampling from the destination flow is straightforward, as the model dependencies contain no directed cycle; see Fig. 3. Each sample produces a different scene simulation. Therefore we can sample repeatedly until a certain criterion, *e.g.* a minimum number of collisions, is met.
**Path-following:** Another solution is to allow real agents for small deviations from their real trajectory. We implement this solution by using a path following strategy for each subject. In particular, each real subject at time $t$ becomes a simulated one with her future position at $t+2s$ as destination. The agent desired speed is the average speed in the next 2 seconds. The limited freedom granted by this path following strategy will allow to reduce the number collisions. Note that this strategy can be combined with the previous one.

## 6 Experiments

Our goal is to reduce to a minimum the amount of manual work for the animator in order to build a realistic simulation of a specific scene. In order to test our algorithm, we need therefore to use real data. We will use three real sequences for our experiments: the *Students* sequence, a $\sim$ 3.5 minutes outdoor sequence has been provided by a third party [9]; the *Meeting* sequence, 10 minutes long, recorded in an indoor during the coffee break of a project meeting; the *Street* sequence [14] contains people walking in the proximity of a tram stop (the event "tram" is active when the tram is at the stop with open doors).
Where not available, the ground truth was extracted with the help of the tracker.

### 6.1 User Study

To validate the quality of the reproduced scene, we set up a user study. We used 2 different video sequences: Meeting and Street. Three 45s long mixtures of real

and virtual agents were generated for these sequences: 0% (purely simulated), 50% and 100% (real sequence without the path-following strategy of Sec. 5). A purely simulated sequence was generated also using a random transition matrix[3]. We call this last simulation the random simulation. For each simulation, the entrance time and the composition of the group was decided by the real sequence. We used a simplified reconstruction of the environment.

Each agent $\mathcal{V}_i$ is populated with all the pedestrians within a 10m radius that are connected to $i$ after a Delaunay triangulation and that are in the 180 degrees field of view of $i$. The interaction parameters were set to match those of the simulated scene. In Fig. 4-center, we report the 2D histograms that show the frequency of the displacement of two subjects, one of which sits in the center of the histogram $\mathbf{p} = \mathbf{0}$ with velocity positive only along the horizontal axis. Left-right symmetry (with respect to the subject in the middle of the histogram) is enforced. These histograms were used to fit the interaction parameters $\lambda_{I,1}, \lambda_{I,2}, \lambda_{G,1}, \lambda_{G,2}$, while we manually set for all the sequences $\lambda_S = 0.3$ and $\lambda_D = 0.03$. To fit the interaction parameters, we use a Gibbs measure interpretation of the energy terms, and we minimize the sum of squared residual between the histogram and

$$(1/Z) \exp\left(-\omega(I(\mathbf{d}) + G(\mathbf{d}))\right) \tag{6}$$

where $I$ and $G$ are from Eq. 1 and Eq. 2, respectively, and Z is a normalizing constant to ensure that Eq. 6 sums to 1. $\omega$ is a parameter that is used only for the conversion from energy to probability. Fig. 4-left shows the result of the fitted energy for the Street sequence. The user study was made available to volunteers on the web. The users were given a sample of the original recorded sequence and then they were asked: "The videos below refer to the scenario shown in the video of the previous page. How realistic does it look to you?". The users had to answer with a score from 1 (unrealistic) to 10 (realistic) for each video. 58 people gave an answer for the Meeting sequence, 51 for the Street one.

The results are shown in Fig. 4, left. In the Street sequence, there is no significant difference between the results of the three mixtures ($p > 0.05$), while there is a difference between the score of each of the mixtures and the random simulation ($p < 0.05$). This suggests that the simulated and mixed simulations achieve a realism similar to the one of the real sequence, with integration of real and virtual agents providing opportunity for adding specific, real, possibly unusual, actions. For the Meeting sequence the real sequence is scored significantly better ($p < 0.05$) than all the others. The random sequence has the lowest average score, but the difference with the other simulated sequences is not significant. In this case, the user does not find the simulated agents as realistic as the real ones, in terms of motion and interactions. We believe that this happens for two reasons. The first is that in the real sequence the transitions are time-dependent, meaning the people in the beginning of the sequence move to the main table and later move to the other part of the scene. Instead, within the event occurrence, our transitions model a stationary process. The second reason is that the environment is dense with static obstacles and standing people. This makes the navigation difficult.

[3] We did not allow arbitrary transitions from any region to the *exit* regions.

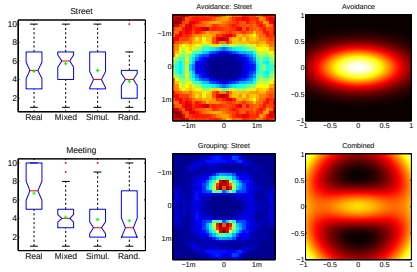


**Fig. 4. Left**: The results of the user study. The central red bar is the median score and the box extends from the 25-th to the 75-th percentile. The dashed lines go to the data points not considered as outliers (red cross). The green cross is the mean score. **Center**: The occupancy histogram for the Street sequence. **Right**: The energy terms after fitting to the histogram.

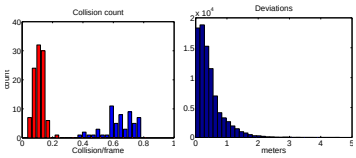


**Fig. 5. Left**: Collisions with path-following strategy (red) and without (blue). **Right**: The real agents' deviation (median $0.33m$) from the original trajectory for one sample, with path-following strategy.

### 6.2 Mixing Real and Simulated People

As discussed in Sec. 5, an interesting possibility is that of using real and simulated agents in the same simulation. There we proposed 2 possible strategies to achieve such goal. Fig. 5, left, shows the results of the comparison. In detail, we extract 100 sample simulations of the Students sequence, with a 50% mixture, once with and once without the path-following strategy, and for each sample we collect the number of collisions (distance $< 0.4m$) between real and virtual agents. The effect of the path-following strategy in reducing the collisions is evident. This, however, comes at the cost of slight deviations from the real agent original trajectory. This is shown in Fig. 5 right. The path-following strategy should then be preferred when the fidelity of the real agent trajectories is not crucial.

## 7 Conclusions

Our goal was to produce a model that could adapt to the semantics of a particular scene, and reproduce it with small effort. Once the virtual agents behave similarly to real people, it is easier to integrate them in the same environment. We validated this possibility by conducting a user study.
Even if the goal selection layer is reduced to a destination selection, a series of interesting behaviors emerge. For example, people gather around tables thus forming new groups, even with no notion of group merging in our model. The action of getting in a tram when it comes, is also the pure result of a learned "go-to" behavior. Although probably less visible, the customization of the steering model to the particular scene has been effective in reproducing its features.
**Timings**. For the Meeting sequence, it took $\sim$ 4 hours to extract tracks and group memberships. All the other processing is carried out in the order of seconds. With our unoptimized code, the simulation of the Students sequence, $215s$ long with an average of 40 agents, requires $117s$ on a single core @2.67GHz.

## References


1. G. Antonini, S. V. Martinez, M. Bierlaire, and J.P. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *IJCV*, 69:159–180, 2006.
2. L. M. Barros, A. T. da Silva, and S. R. Musse. Petrosim: An architecture to manage virtual crowds in panic situations. In *CASA*, pages 111–120, 2004.
3. Adriana Braun, Bardo Bodman, and Soraia Raupp Musse. Simulating virtual crowds in emergency situations. In *VRST*, 2005.
4. M. Gérin-Lajoie, C. Richards, and B. J. McFadyen. The negotiation of stationary and moving obstructions during walking: anticipatory locomotor adaptations and preservation of personal space. *Motor control*, 2005.
5. D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 1995.
6. E. Ju, M. Choi, M. Park, J. Lee, K. Lee, and S. Takahashi. Morphable crowds. In *SIGGRAPH Asia*, 2010.
7. Ioannis Karamouzas and Mark Overmars. Simulating the local behaviour of small pedestrian groups. In *VRST*, 2010.
8. Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *SCA*, 2007.
9. A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *EUROGRAPHICS*, 2007.
10. R. Mann, A.D. Jepson, and T. El-Maraghi. Trajectory segmentation using dynamic programming. In *ICPR*, 2002.
11. M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 2010.
12. Soraia Raupp Musse and Daniel Thalmann. Hierarchical model for real time simulation of virtual human crowds. *TVCG*, 7, 2001.
13. J. Ondřej, J. Pettré, A. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. In *SIGGRAPH*, 2010.
14. S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.
15. Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010.
16. J. Pettré, J. Ondřej, A. Olivier, A. Cretual, and S. Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *SCA*, 2009.
17. Craig Reynolds. Steering Behaviors for Autonomous Characters. In *GDC*, 1999.
18. Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH*, 1987.
19. Wei Shao and Demetri Terzopoulos. Autonomous pedestrians. In *SCA*, 2005.
20. Daniel Thalmann and Soraia Raupp Musse. *Crowd Simulation*. Springer, 2007.
21. Peter Trautman and Andreas Krause. Unfreezing the Robot: Navigation in Dense, Interacting Crowds. In *IROS*, 2010.
22. A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM TOG*, 2006.
23. Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *SIGGRAPH*, 1994.
24. Jur P. van den Berg, Ming C. Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *ICRA*, pages 1928–1935, 2008.
25. Zhang Y, J. Pettré, J. Ondřej, X. Qin, Q. Peng, and S. Donikian. Online inserting virtual characters into dynamic video scenes. *CAVW*, 22(6), 2011.