DOI: 10.1111/cgf.14506
EUROGRAPHICS 2022
D. Meneveaux and G. Patanè
(Guest Editors)

Volume 41 (2022), Number 2
STAR – State of The Art Report

Authoring Virtual Crowds: A Survey

Marilena Lemonari[1], Rafael Blanco[2], Panayiotis Charalambous[3], Nuria Pelechano[2]
Marios Avraamides[5], Julien Pettré[4] and Yiorgos Chrysanthou[1,3]
[1]Department of Computer Science, University of Cyprus, Cyprus [2]Universitat Politècnica de Catalunya, Spain [3]CYENS - Centre of Excellence, Cyprus
[4]Inria Rennes - Bretagne Atlantique, France [5]Department of Psychology, University of Cyprus, Cyprus

Abstract

Recent advancements in crowd simulation unravel a wide range of functionalities for virtual agents, delivering highly-realistic, natural virtual crowds. Such systems are of particular importance to a variety of applications in fields such as: entertainment (e.g., movies, computer games); architectural and urban planning; and simulations for sports and training. However, providing their capabilities to untrained users necessitates the development of authoring frameworks. Authoring virtual crowds is a complex and multi-level task, varying from assuming control and assisting users to realise their creative intents, to delivering intuitive and easy to use interfaces, facilitating such control. In this paper, we present a categorisation of the authorable crowd simulation components, ranging from high-level behaviours and path-planning to local movements, as well as animation and visualisation. We provide a review of the most relevant methods in each area, emphasising the amount and nature of influence that the users have over the final result. Moreover, we discuss the currently available authoring tools (e.g., graphical user interfaces, drag-and-drop), identifying the trends of early and recent work. Finally, we suggest promising directions for future research that mainly stem from the rise of learning-based methods, and the need for a unified authoring framework.

CCS Concepts

• ***Computing methodologies*** → ***Interactive simulation; Motion path planning; Collision detection; Intelligent agents;***

1. Introduction

Crowds constitute an important part of our daily lives; we interact with them in streets, workplaces, concerts, etc. Thus, the computer graphics and animation community has been exploring various ways to simulate and animate virtual crowds, reflecting the characteristics of real humans; behaviours, group dynamics, interpersonal and environment interactions, etc. Achieving believable virtual crowds is an essential component of numerous significant applications e.g., computer games, movies, architectural visualisations. Crowd behaviour typically emerges from the interaction between many individual autonomous agents. These agents can be simulated through a large variety of methods. However, achieving a specific final behaviour out of the emergent interaction between autonomous entities, can be extremely challenging and is a common problem amongst these methods. Thus, authoring tools enabling users to describe the desired final behaviour of a crowd, or even tune certain parameters to further polish the resulting simulation, are key for populating lively environments.

1.1. Crowd simulation and authoring

Depending on the nature of the application, authors focus on satisfying certain application-specific characteristics. Therefore, there are many levels to authoring including considerations of *interactivity*, *scale* and *variety*. For instance, when looking to author a crowd in terms of editing a finished product (at run-time and according to additional requirements), as for example in video games, then the authoring process revolves around interactive manipulation. On the other hand, in cases where we care about the global behaviour of crowds, as in architectural visualisations and urban cities, the attention shifts back to authoring how crowds move at a macroscopic scale, or controlling aspects of the environment. At a smaller scale, there is a need to consider the level of intelligence of the virtual agents, with interest being directed towards producing characters who exhibit various behaviours and autonomy. These are particularly useful in more creative applications such as movies, while still being relevant for urban planning, allowing for pedestrians to have meaningful and consistent whereabouts. Variety in character appearance is also invaluable to the realism of the virtual crowd.

The underlying concept that unites all of these is *crowd simulation*. Authoring provides ways to manipulate crowd simulators and their parameters, or the results of crowd simulations (i.e. editing). In that sense, as previously stated, there are many levels to authoring, highlighting its complexity. It ranges from generating and running the simulation, to altering its aspects at a later stage. An authoring system aims to give control to the users to realise their creative intents, through intuitive and natural interfaces, thus alleviating the need for user expertise. Additionally, providing tools for interactive modification of the content at every stage of the simulation, leads to flexible systems that enable instant feedback and visualisation of the results, helping authors to achieve the desired outcome. Authoring tools range from coding and scripting to GUIs (Graphical User Interfaces), reflecting the amount of user experience required which varies in each case e.g., GUIs are intuitive tools, usable by non-experts.

In this survey, we turn our attention to methods for authoring virtual crowds. Authoring may apply to several different aspects of crowd simulation, making this a complex and multi-level task. Aiming to help the reader understand the full extent of the task of authoring crowds, we propose a categorisation of authorable aspects, by identifying six main components of crowd simulation that allow for content modification: high-level behaviours; path-planning; local movements; 3D body animation; visualisation; and post-processing. Each component affects certain crowd characteristics, all influencing the realism and believability of virtual crowds.

There has been a large number of crowd simulation models that have emerged over the last few decades [vTP21,MCT21]. The combination of having so many models and also many aspects that can be simulated in a crowd (see bubbles in Figure 1) suggests a lack of consistency in previous literature that hinders the attempts for a unified approach for authoring. Each researcher provides specific authoring tools that can be exclusively applied to the set of parameters available in their model. We believe the timeliness of this survey is important due to the need to evaluate all different authoring approaches that have been proposed so far, and to discuss their advantages and limitations. This will allow us to start leading the discussion towards a unified authoring pipeline, so that in the near future we could have a framework built of interchangeable models for each of the identified bubbles in Figure 1, sharing a common authoring interface.

1.2. The bubble space of crowd authoring

For further clarification of the proposed structure, we build a diagram with the relevant crowd simulation components, depicted in Figure 1. The individual components extend the diagram according to what aspects can be edited. These represent what can be authored for each component, giving a general idea of the levels at which the user can influence the simulation. The green bubbles in the diagram can each be further extended to demonstrate popular interfaces through which authoring of these aspects can be achieved.

More specifically, regarding **high-level behaviours**, we can define agents' agendas or describe their desires. The former includes specification of spawn and exit location, goal destinations, and schedules corresponding to pre-planned sequence of tasks. Authoring desires aims to reflect agent intentions and motivations. This component is the most prominent, influencing the rest, with past research tackling this kind of authoring mostly via *scripting* and *drag-and-drop GUIs of behavioral units*.

Additionally, defining **path-planning** aspects corresponds to authoring global paths, trajectories, or the planning strategy. This is done by indicating the desired group flow or character paths, mainly achieved through *sketch-based* interfaces (e.g., sketching arrows). For global paths, authoring group formations usually by

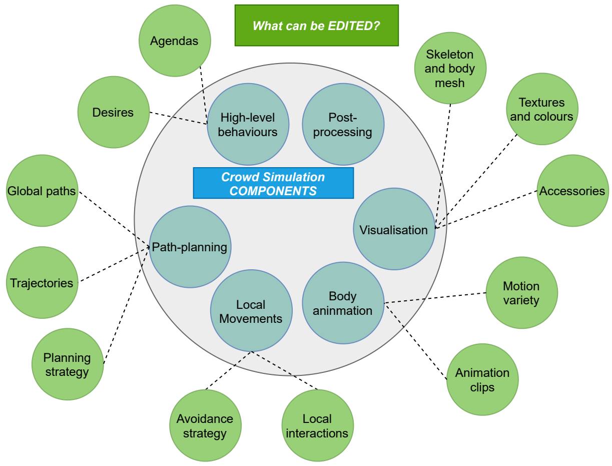

Figure 1: *Diagram reflecting the crowd simulation components and each component's authorable aspects.*

drawing them, is also relevant, whereas editing the planning strategies concerns the environmental and other factors taken into consideration before implementing the path-planning algorithm.

In a similar manner, **local movements** deal with: avoidance strategies, and local interactions with other agents (local group behaviours) or with the environment. Mainly, the authoring of local movements and behaviours is addressed by previous literature with low-level parameter specification using *coding* or basic *GUIs* that allow the user to specify values for a rather small set of parameters.

Meanwhile, editing **body animation** includes modifying the animation clips or influencing the perceived motion variety. Past work has explored ways of editing the data, blending and interpolating, as well as manipulating the animation clips subject to authors' wishes. This is often done *manually*, while motion variety can also be obtained by some more expressive interfaces (e.g., *gestures*). Commercial software has focused on this, delivering the so-called motion templates that facilitate authoring crowd diversity.

Appearance considerations comprise the **visualisation** component, which deals with body mesh deformations (e.g., shape, gender, scale), and other character attributes such as textures, colours, and accessories. For this component, we talk about characteristics that can be authored for static agents and scenes, without considering movements. In practice, the employment of *assets and templates* assists such visualisation descriptions.

The final component of crowd simulation refers to **post-processing** and is concerned with editing the finished product. Authoring then, comes down to interactive and direct manipulation of the simulated and animated crowd. This affects all previous components, since editing operations can be applied to path-planning, local movements, animation etc. Various types of intuitive interfaces have been developed so far, some requiring more user experience, and others being more suited to novice users, with *deformation gestures* being highly convenient.

Figure 1 presents an overview of an authoring system's functionalities in the form of bubbles instead of the traditional pyramid format. This intentional layout attempts to emphasise the task's peculiarity, in the sense that authoring can deal with one of these components, or all of them together. Ideally, a perfect system would allow authoring all the elements, but the majority of past research focuses on one or two, and the rest are taken as default (e.g., from the AI and animation systems of a game engine). It is also important to note that often, several of these bubbles could be overlapping to some extent, because the components' behaviour may be entangled. For example, a mental state aspect such as experiencing stress, could affect the path finding in terms of route selection, but also the local movement in terms of velocity and personal space preferences.

1.3. Survey overview

This paper attempts to clarify and provide structure to the task of authoring, by dismantling the authorable components into some suggested categories. We believe that this standardisation will make it easier to establish which aspects lack the desired efficiency, as well as unveil the underlying workings needed to create successful authoring systems. Moreover, we aim to provide the reader with an overview of what has been already done in the field, and how current state-of-the-art is achieved. Thus, this survey appeals to both inexperienced audiences, looking to understand the basics of authoring virtual crowds, and more experienced readers, invoking the need for some standardised structure.

However, it is important to note that such structuring is difficult, since there are several grey areas between each of the categories, which are highly interconnected and integrated within each other. Our attempt is to shed light on the underlying purpose and common characteristics of these aspects in order to reach a rational categorisation. For the purposes of this survey, we: identify the authorable components of crowds simulation (Figure 1-blue bubbles); propose a categorisation of these components according to what aspects can be authored (Figure 1-green bubbles); discuss the main methods used by previous research to tackle this task for each component; and present the available interfaces through which users can author the simulation content.

All the above are considered for each crowd simulation component separately. Hence, the rest of the paper is structured as follows: Section 2 deals with high-level behaviours; Section 3 is concerned with path-planning; Section 4 refers to local movements; Section 5 represents body animation; Section 6 turns attention to visualisations; and Section 7 looks into post-processing operations. We conclude with Section 8, by discussing the content of this survey, and highlighting key points such as what are the current limitations and potential interesting directions for future work.

2. High-level behaviours

In simulated crowds, agents often align their actions with the general movement either guided towards certain environment locations, or follow indicated trajectories. However, in these cases, the agents do not exhibit intelligent behaviours, and thus the resulting crowds suffer in terms of realism. Therefore, an important component of virtual crowds is the agents' high-level behaviours. In this context, they refer to the behaviours of individual agents who function as part of a crowd but also have their own purpose, such as having a target position or following an agenda. High-level behaviours are mainly looked at from an agent-centric point of view, which adds to the realism of the virtual crowd by means of emergent global behaviours and behaviour diversity.

Hence, controlling such high-level behaviours is of particular interest in the authoring domain. For most authoring applications such as computer games and movies, the focus of the author is not limited to guiding the crowd as a whole or controlling purpose-less characters, but also includes the control of the crowd using high-level descriptions. To this end, past work has touched on editing several useful aspects of the high-level agent behaviours.

This level of authoring is perhaps the most important as it could influence all the other components e.g., path-planning, local movements. Therefore, we dedicate this section to the review of methods for authoring behaviours at a high level. We discuss two main aspects which can be edited by the user to achieve the expected behaviours: the agents' agendas, and desires.

2.1. Agendas

One way to control agents at a high-level, is through defining their agendas. For realistic movie and computer game scenes for example, where protagonists find themselves in a crowded area, the background characters in the crowd need to behave naturally by performing a series of tasks that are logically related. Therefore, the presence of crowds with intelligent agents, having their own agendas, is crucial for the analysis and the perceived plausibility of the crowds. Past work on controlling agendas focuses mostly on specifying the spawn and exit points of the agents, as well as describing their schedule more explicitly by incorporating more information leading to the presence of intermediate tasks.

2.1.1. Spawns and exits

The most straightforward purpose that agents can have is their goal destinations. Most of the literature dealing with crowd simulation aspects like path-planning and global path-finding, has user-specified start and exit points of agents as prerequisites for their systems. Defining these points comprises a high-level authoring task and is tackled in a number of ways. For methods incorporating navigation graphs in the path-planning stage, control over the goal destination is often achieved via updating the relevant graph nodes. Early work on this by Yersin *et al.* [YMC*05] allows for description of goal destinations through a GUI, which updates a semantically augmented navigation graph instead of inputting these goals in vector form.

Also, some research exploring macroscopic approaches utilises dynamic potential fields. Treuille *et al.* [TCP06] employ continuum crowds, a model based on dynamic potential fields, which handles global planning in a collision-free manner both for static and dynamics objects (Figure 2). This facilitates the authoring of goal-oriented crowds, which is achieved via manual parameter selection.

Another way of incorporating user-specified spawn/exit points is to add them as another layer of a multi-layered map representation. Particularly, the system by McIlveen *et al.* [MMHR16] passes control to the user in terms of spawn/exit specification through an

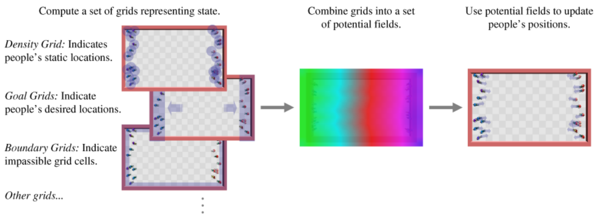

Figure 2: *Example of grid-based destination specification via a dynamic potential field approach, taken from [TCP06].*

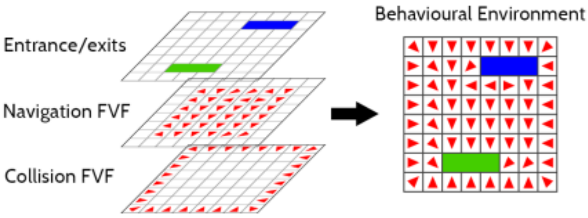

Figure 3: *Example of map-based spawn/exit specification via layers, taken from [MMHR16].*

independent layer in a multi-layer FVF (Force Vector Field). For authoring purposes, they use a painting tool and offer an intuitive graphical interface by means of a map (Figure 3). Following a similar concept, Gonzalez and Maddock [GM17] employ a sketching-based interface for authoring simulation aspects including defining spawn and exit points, which results in an altered navigation representation. Agents then base their path-planning on this representation. Both research works are successful in reflecting the movements of people in complex environments (e.g., train stations), in a controlled and realistic manner.

2.1.2. Schedules

Agendas usually imply more complex schedules than merely having a goal destination, since they determine a sequence of tasks that need to be followed by the agent, or places they need to visit. Naturally, destination points are incorporated into the agents' agendas, since they are part of the tasks to be completed. One main advantage of being able to author such schedules is their effectiveness in creating believable virtual cities.

Allbeck's CAROSA system, presented in [All10], is seminal in this area. This system handles instructions in the form of agent responsibilities, and descriptions of agent characteristics. Such aspects are authored via common Microsoft Office tools, thus accommodating non-experts. In another work, De Paiva *et al.* [dPVM05] propose the UEM (Urban Environment Model) composed of agents with profiles that define the movements and stays during the day. These profiles are defined through coding and consist of a list of time intervals, followed by a fixed place or place category to allow random place choice. Alternatively to ontology-based methods, Jorgensen and Lamarche [JL14] propose a model to endow virtual agents with scheduling capabilities of long-term tasks, defining paths and sequence of tasks under spatial, temporal and personal

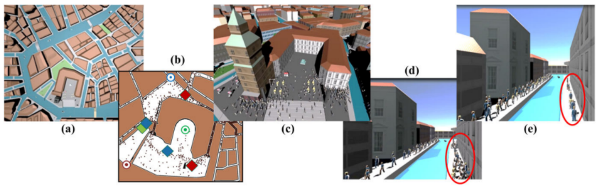

Figure 4: *Urban walkability design example from [MKMA19]. Inputs of the framework are: urban layout (a), agent goals and intentions from which 2D (b) and 3D (c) layout modifications can be inferred, and editable attributes such as densities (d), (e).*

characteristic constraints. The activities are modelled using hierarchical descriptions based on the combinations of tasks and constructors, which indicates how the tasks can be combined. Then, the scheduler takes these descriptions for building a sequence of tasks to be performed, minimising the effort measured by the distance that agents have to move to complete the activities. Mathew *et al.* [MKMA19] also deal with user-specified goals and intentions, modifying the urban layout according to those specifications. A visual example of their framework can be seen in Figure 4. Rogla *et al.* [RPP21] demonstrate the capabilities of their system in an urban environment with park, work, and school buildings. They integrate their PCG (Procedural Crowd Generation) model to a game engine and enable the specification of agendas using a rule-based grammar. Their system allows for environment manipulations as well, such as copy and paste of objects in different locations, and adding landmarks.

2.2. Desires

An additional authoring aspect is the specification of agents' desires and motivations. To some degree, this is related to Section 2.1 in that, the series of actions to be satisfied by the agents reflect their desires. Despite the conceptual similarities, we claim that an agent's desire relates to temporary, spontaneous intentions as opposed to pre-planned sets of tasks. These correspond to emotional state, ambitions, or individual traits e.g., personality and attitude. These can be selected to affect individual agents or a portion of the population, thus allowing users to decide the level of detail they want to author.

Early work by Badler *et al.* [BBA*98] turns the attention to describing a scenario according to high-level instructions given by the user in natural language, translated to low-level actions by parameterised rules. Alternatively, Musse and Thalmann [MT01] use scripting to describe reactions and behaviours on an agent-level as well as group-level. Funge *et al.* [FTT99] worked on the cognitive aspect, incorporating knowledge and learning into their framework. Shao and Terzopoulos [ST07] allow authoring of agent needs by incorporating a memory component to their system and representing the mental state of agents with individual parameters such as tiredness and curiosity. Being able to specify personality traits of agents is highly valuable to authoring systems e.g., Yu and Terzopoulos [YT07] via parameter manipulations yield attributes like courage. These traits affect various levels of crowd simulation, in addition to the high-level behaviours component; the HiDAC model

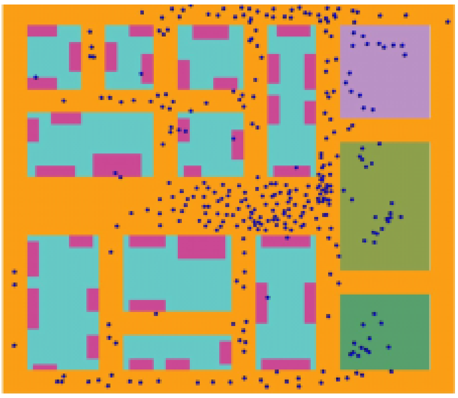

Figure 5: *Example of user-inputted environment layout with area annotations eventually assigning agents to specific behaviours from [LCHL07].*

[PAB07] directly affects path-planning. Desires of being close to other agents are also incorporated in some path-planning systems like Crowd art [JCC*15].

Lee *et al.* [LCHL07] allow for control of behaviour models learned from video data. Authoring corresponds to creating the environment layout and annotating it according to the desired behaviour models, eventually assigning each agent to behaviours, such as chatting (Figure 5). Li and Allbeck [LA11] implement a system based on roles and role switching. Each agent has a set of roles, employed in accordance with simulation conditions such as time, location, needs or reactions to other agents. The roles are defined by activities that indicate the places where the agent should be according to the simulation time. They also define the behaviour of agents within the places they moved to. The definition of roles and activities is through code, using well-defined data tuples.

Later work by Krontiris *et al.* [KBK16] presents a framework revolving around user-specified agent desires, translated into behaviours via influence maps. The environment plays a significant role, as it can be used to define environmental attractors. This enables a range of applications including the emergence of heterogeneous crowds, as well as desired group dynamics by declaring other agents as attractors. Similarly, Aschwanden *et al.* [AHH*09] populate virtual cities where agents are endowed with sensors to see and hear the attractors on the environment, and dynamically change the locations they planned to visit. Zhang *et al.* [ZSH*19] provide a framework that defines the movement and behaviour of agents inside a building, with rooms satisfying space, behaviour and motivation constraints. The authoring process consists of defining numerical parameters that indicate the desired building occupancy over time, the proportion of agents performing the behaviours defined per room, and the motivation level the agents have for these behaviours. The schedulers aim to resolve the occupancy and behaviour distribution constraints, maximising the agent´s motivation. Normoyle *et al.* [NLS14] allow for specification of duration, ordering, and preferences and create a model satisfying these constraints according to a stochastic graph.

Furthermore, Kraayenbrink *et al.* [KKT*14] focus their research

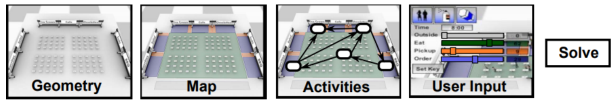

Figure 6: *Author uses sliders and brushes to set the agents' distribution, which then define the geometry, map, and activities that [NLS14]'s method depends on. Figure taken from [NLS14].*

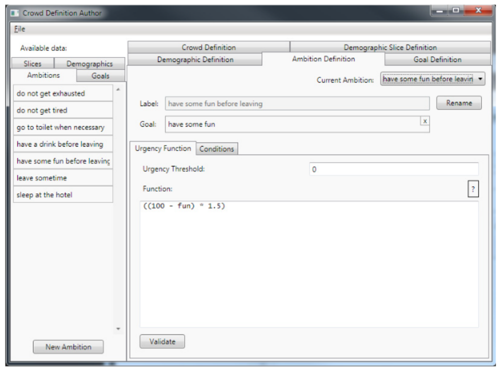

Figure 7: *Snapshot of the crowd editor, allowing for profile specifications, taken from [KKT*14].*

on re-usable crowd templates to author scenes based on agent desires and ambitions selected via an interactive editor, illustrated in Figure 7. They make use of a BDI (Belief, Desire, Intention) agent model variation and achieve such re-usability by minimising the environment information that the agents are required to possess, and storing it into the environment objects instead. Recent work by Mathew *et al.* [MBA20] use indicators for behaviour specification and employ a swarm optimisation model.

Extending the range of possible applications, Kapadia *et al.* [KSRF11] focus on developing an authoring framework which facilitates the generation of user-specified storylines, while attempting to minimise the authoring burden of specifying low-level parameters. Their behaviour generation framework allows authoring of the state and action space, refined by the use of modifiers through scripting, as seen in Figure 8. This work is useful for developing narratives and scenarios like authoring a robbery scene.

The introduction of logical constructs and cognitive models such as smart events, planners and especially PBTs (Parameterised Behaviour Trees) has opened the door to a whole new range of capabilities. Kapadia *et al.* [KFS*16] turn attention to partial-order planners to automatically complete partial user specifications. This is of particular importance in the context of authoring, since it further alleviates the burden of having to specify all agent actions. Their system allows authors to describe a story containing the main events with intermediate, secondary ones being automatically generated, permitting the production of more sophisticated scenarios, easily. Related work by Kapadia *et al.* [KSS*16] demonstrates the authoring of a market scene utilising smart events, affordances and PBTs. This system, again allows the user to define the Story Arcs, through an easy-to-use graphical interface (Figure 9). The inter-

```
Action Move(Velocity : v, TStep: dt) {
  Precondition:
   CheckCollisions(self.position + vdt) == false;
  Effect:
   self.position = self.position + vdt;
  Cost Effect:
   self.energyCost = ½(self.mass)|v|²;
   self.distanceCost = |v|dt;
}
```

(a)

```
Behavior GoalBehavior {
  Precondition: self.goalPosition ≠ 0;
  Goal: self.goalPosition;
  Objective Function:
   min(self.distanceCost + self.energyCost);
}
```

(b)

```
Constraint PedSignalConstraint {
  Precondition: true;
  Constraint:
   if ((signal.signalState == 0
   ∧CrossingRoad(self.position,C))
   ∨(signal.signalState == 1
   ∧CrossingRoad(self.position,A))
   ∨(trafficSignal.signalState == 2
   ∧CrossingRoad(self.position,B)))
    true;
   else false;
}
```

(c)

Figure 8: *Scripting of the "move" action, moving to a goal position (pedestrian behaviour), obeying traffic lights (constraint), shown in (a), (b), and (c) respectively, taken from [KSRF11].*

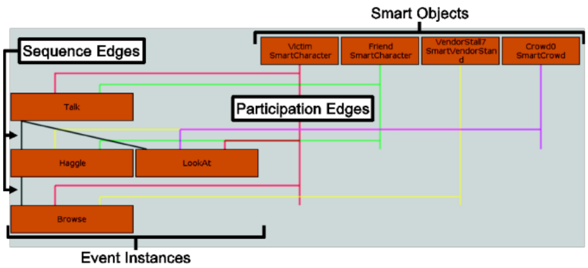

Figure 9: *Drag-and-drop interface for creating a Story Arc, taken from [KSS*16].*

faces of [KFS*16, KSS*16] are suited to non-experts and trade-off ease of specification and autonomy of behaviour generation.

Another piece of work by Durupinar et. al. [DPA*11] attempts to create different trajectories which correspond to certain personality traits. Their mapping of low-level parameters to such personality traits aids the authoring process, allowing untrained authors to use the system and create motion variety in a controlled manner. The control they provide has an apparent effect on additional crowd simulation levels apart from high-level behaviours, such as the local movements. Figure 10 illustrates the personality factors which reflect agent behaviours and act as inputs to the used Ocean personality model. Best *et al.* [BNCM14] consider the psychological aspect of personal space as a way to affect the distance an agent keeps from others.

Later work by Durupinar *et al.* [DGAB15] incorporates personality and emotion to reflect the different crowd types, from audi-

Agent behavior	Personality factor	Ocean factor*
Leadership	Assertive, social, unsocial, calm, fearful	E, N
Trained or untrained	Informed, ignorant	O
Communication	Social, unsocial	E
Panic	Oversensitive, fearful, calm, orderly, predictable	N, C+
Impatience	Rude, assertive, patient, stubborn, tolerant, orderly	E+, C, A
Pushing	Rude, kind, harsh, assertive, shy	A, E
Right preference	Cooperative, predictable, negative, contrary, changeable	A, C
Avoidance or personal space	Social, distant	E
Waiting radius	Tolerant, patient, negative	A
Waiting timer	Kind, patient, negative	A
Exploring environment	Curious, narrow	O
Walking speed	Energetic, lethargic, vigorless	E
Gesturing	Social, unsocial, shy, energetic, lethargic	E

**The letters in this column stand for openness, conscientiousness, extroversion, agreeableness, and neuroticism.*

Figure 10: *Personality factors reflecting agents' behaviours vs Ocean's low-level parameters, taken from [DPA*11].*

ences to mobs. They aim to give users the ability to author such types, creating several scenarios (e.g., protests, and sales) by utilising the Ocean personality model; this work on high-level behaviours influences other levels as well such as local movements and path-planning. Sinclair *et al.* [SL15] attempt to incorporate both personality and emotional traits into their framework. The framework takes as inputs these traits and maps them to model parameters, affecting both the local movements and path-finding.

2.3. Industry software

Various tools have been created for authoring high-level behaviours, especially in the real-time domain (e.g., computer games). For example, several plugins have been developed for most Game Engines that allow game designers to graphically author such high-level behaviours, using standard industry methods such as Behaviour Trees and FSMs (Finite State Machines). Examples of that include the nodeCanvas tool by Paradox Notion [Par] and Behavior Designer by Opsive [Ops] for the Unity Game Engine, and the Behaviour Trees system that is integrated into the Unreal Game Engine [Enga]. Several studios also develop their own custom tools and pipelines to author high-level behaviours for characters, such as the procedural Play-as-Anyone system developed by Ubisoft for the Watch Dogs Legion video game [Ubi].

2.4. Summary

In conclusion, we have seen two ways of describing the high-level behaviours of virtual agents: specifying agendas, and agent desires. Specifying agendas contains the simpler task of defining spawn and exit locations. Past literature has achieved authoring of such locations by employing grid-based approaches (e.g., maps), which benefit both crowd diversity (agents come and go from and to different locations) and environment manipulation. We also reviewed the methods enabling description and editing of agents' agendas, which normally consist of a series of predefined actions or tasks. For this, developing agent profiles or roles is quite common. When wishing to describe such schedules, authoring via scripting is widely used.

Contrary to describing some predefined, task-oriented agenda, some researched ways to make agents more human-like by incorporating the concept of desires. Definition and manipulation of such desires by the user, reflects the more psychological side of agents' intelligence. Such desires might correspond to their ambitions or intentions (e.g., rob a bank). Scripting is used for authoring such desires, but also recent approaches have incorporated some more user-friendly interfaces (e.g., drag-and-drop) to facilitate the creation of complete narratives. It is common that frameworks tackling the authoring of high-level behaviours infiltrate other simulation components; this demonstrates the significance of high-level parameters.

3. Path-planning

Another authorable component of crowd simulation is path-planning. This section aims to describe an intermediate level of control between high-level behaviours and local movements. For the purposes of this survey, we limit the high-level behaviours category to include the actions and movements stemming from intelligent virtual agents that have a sense of purpose. On the other hand, local movements correspond to agent decisions made on a frame-by-frame basis. Therefore, this category refers to high-level user control affecting the agents on a global scale, not requiring, however, purposeful agents and merely sufficing to agents with desires of following certain paths. Perhaps this is the first aspect we seek to control e.g., the paths followed by the characters, which is important for the development of meaningful and controllable crowds.

Describing such paths is particularly important in cases of complex virtual environments, endless crowds (not limited to a certain time-window), and for achieving heterogeneity. The task of authoring the aspects of path-planning is fundamental to the majority of applications, from entertainment (e.g., movies and games) to simulation systems and training exercises.

In this section, we present the state-of-the-art in this area and we consider: authoring global paths (i.e. flows and formations); authoring trajectories; user-control given by patch-based methods; and controlling the path-planning strategy.

3.1. Global Paths

The first aspect of the editing and specification of global paths includes describing flows, and formations. For both, such user descriptions imply a global, collective desire to either follow a path or move in a coordinated manner. Since the inputs come from the author in the form of more high-level specifications, the agents still need to plan their paths explicitly.

3.1.1. Flows

When dealing with virtual crowds, we typically seek to control the movement of a large number of crowd members, even the whole crowd, without explicitly having to specify individual paths. Such approaches have been explored through the years, to allow user specification of crowd flows, or group flows within crowds. In most techniques, the motion of agents is subject to the existence of some type of velocity field (e.g., flow and navigation field). Therefore, flow control stems from the ability to define and manipulate such fields, either through direct interaction or through comprehensive and practical interfaces. In the literature, there have been several methods attempting to provide this level of control to users, delivering a range of suitable interfaces such as sketching, mapping and graph-based.

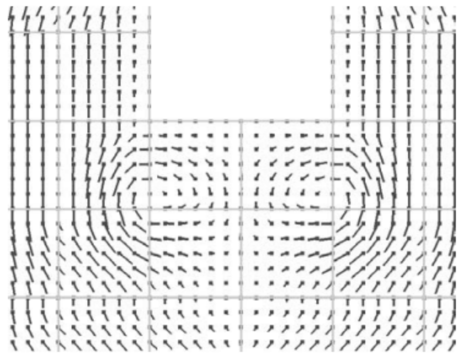

Figure 11: *Example of flow tiles specification from [Che04].*

Early work in this area by Chenney *et al.* [Che04] formulates the idea of flow tiles, and provide a natural interface to define such flow fields. Placing crowd members in the environment, the agents are influenced by the current flow field and guided toward the desired path. Figure 11 illustrates an example of such flow tiles, which achieve divergence-free velocity fields, and are successful in populating large environments in an easy and relatively quick manner. Later, Jin *et al.* [JXW*08] again work with the concept of vector fields, allowing user specification via combinations of RBFs (Radial Basis Functions). They employ a sketch-based approach for inputting arrows to drive the crowd motion, by assigning velocities at certain anchor points. The resulting velocity field is an interpolation of the impact of anchor points. The method facilitates instant feedback coming from the interactive manipulation of these points, and proves useful for handling large crowds at interactive rates.

In the 2010s, Park *et al.* [Par10] reinvent the role of control particles for guiding crowd flow. Specifically, they establish a potential field controlled by the trajectories of the control particles, successfully driving massive crowds while retaining the advantages of both key-framing and dynamic potential fields. The trajectories of the control points are defined through an intuitive interface, shown in Figure 12, along with the derived gradient fields. By construction, this method also enables more sophisticated crowd manipulation, such as group separation. Additionally, Patil *et al.* [PBC*11] utilise navigation fields allowing authoring in two ways. Firstly, through a sketch-based interface, and secondly, by inputting flow fields corresponding to real-world crowds. In other work, Kang and Kim [KK14] adapt an alternative method for generating flow fields, which makes use of vector paintings, while Barnett [Bar14] use a Reeb graph to extract topology and create moving crowds in environments, at maximum capacity.

Data-driven methods such as in [JCP*10] manage to extract information from source data, learning both a trajectory and a formation model that interact with each other. Users can define these trajectories, forcing groups of agents to follow the predefined paths (Figure 13).

More recently, Gonzalez and Maddock [GM17] employ navi-

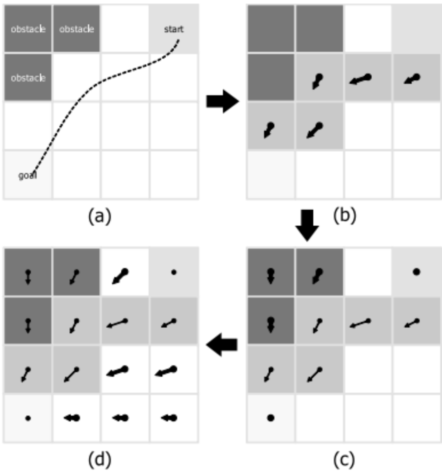

Figure 12: *Example of control particle trajectories from [Par10] and the corresponding control gradients at the explored, obstacle, and remaining cells (b, c & d respectively).*

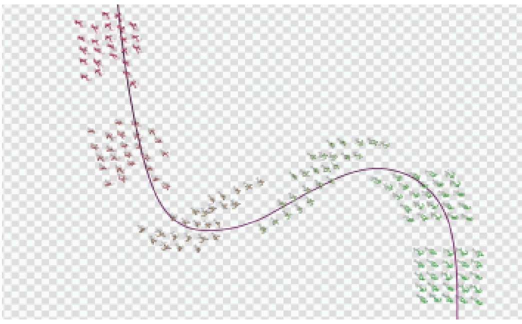

Figure 13: *Example of path-following according to a user-drawn trajectory, taken from [JCP*10].*

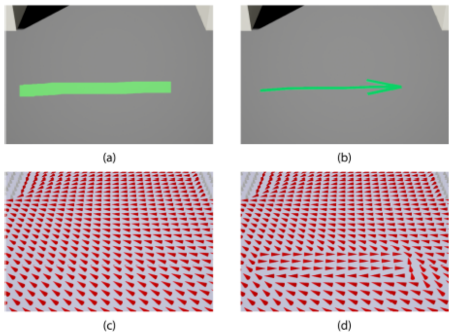

Figure 14: *Example of how a navigation map changes after the addition of user-specified inputs, taken from [GM17] (navigation map before (c) and after (d)).*

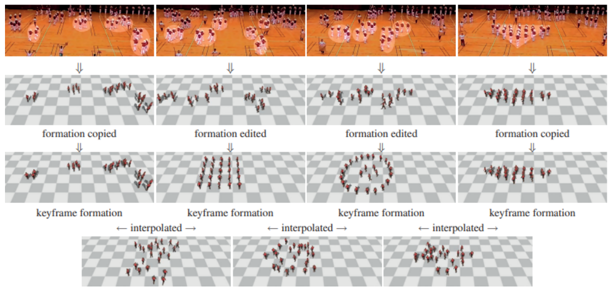

Figure 15: *An example of the process taken from [TYK*09]. Extracted keyframe formations are edited according to the author, and then interpolated.*

gation maps to guide agents, and develop a framework useful for applications in training systems and city planning. In particular, by sketching arrows in the environment, the user can define and change agent trajectories as in Figure 14, which further facilitates the emergence of behaviours such as lane formation. A data-driven approach implemented by Bulbul and Dahyot [BD17] produces flows of pedestrians for environments based on real cities. Their system uses geo-located information from social media that reflects the spatial and temporal distribution of people over the city. Note that, in the majority of flow control techniques, the resulting systems have an integrated avoidance strategy (or local movement technique), which is either dealt with in a separate component of the system, or implemented concurrently via the developed interfaces ([KRR10, GM17]). For instance, [GM17] allow for addition of barriers, which correspond to collision maps integrated with the other navigation maps.

3.1.2. Formations

Group formation control is an additional aspect of authoring, investigated in literature. The goal is to intuitively and quickly describe the desired formation, then letting the system translate these descriptions to global paths while preserving the neighbouring relationships of agents within the selected group.

In early works on authoring formations, Takahashi *et al.* [TYK*09] take keyframe formations and interpolate between them using spectral analysis. The input formations are extracted from video data, and then edited by the user to reflect their desired shape. The process concludes with the interpolation of these keyframes, as illustrated in Figure 15.

An alternative way to describe the desired group formation is the specification of formation boundaries ([GD11]). This is supported by intuitive and expressive interfaces such as the sketch-based one in [GD11], allowing to draw the target formation shape. Their method uses the novel concept of formation coordinates to preserve the adjacent naturalness of agents, when they change their position to match the target formations. Again using formation coordinates, later work by Gu and Deng [GD13] allow for the sketching of formation boundaries, or authoring target formations via a brush metaphor. In contrast, Henry *et al.* [HSK14] employ a multi-touch device to obtain a deformable mesh in an attempt to provide formation control, while taking environment into considera-

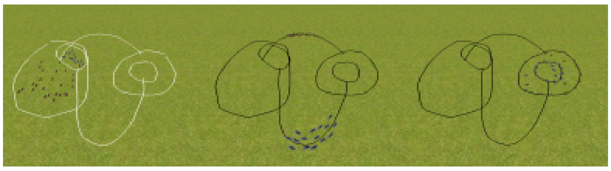

Figure 16: *Example of six input strokes of a group and subgroup with different user-defined paths, taken from [APKM15].*

tion. This produces a single-pass algorithm which enables authored crowds in complex environments. Conversely, Xu *et al.* [XWY*14] do not use an intuitive interface, but present a system controllable by the user at several stages, such as target formation, overall direction and agent movement control.

Additionally, Allen *et al.* [APKM15] highlight the usefulness of sketch-based tools for authoring as it achieves formation control for heterogeneous crowds and more complex goals via sub-group sketching and increased number of input strokes. Their system supports both formation and flow control (Figure 16).

3.2. Trajectories

The methods used for guiding crowds and controlling group flows often result in homogeneous crowd movement, without enough motion differentiation among agents to provide the feel of a realistic natural-moving crowd. By design, velocity fields occupying the moving space do not facilitate movement diversity. Inspired by the need to produce such variations, there is some literature dedicated to allowing manipulation of the trajectories of individual agents.

Some research adjusted the concept of grid-based vector fields to enable the specification of individual paths. For instance, Metoyer and Hodgins [MH03] create potential fields according to user input in the form of desired paths, sketched via an interface shown in Figure 17. The path-following is then performed by exerting forces for the individual trajectories. To alleviate the authoring burden, this method stores user examples and implements a learning-based method to provide suggestions for future situations. Another example of reactive path-following for individual trajectories is the method in [OO09]. Similarly to Metoyer and Hodgins, Oshita and Ogiwara [OO09] deliver a sketch-based interface, translating example paths to a set of parameters which correspond to the desired animation.

Recently, interaction fields by [CvTZ*22] use an intuitive sketch-based tool for authoring the paths as well as the orientation of agents. This gives users artistic freedom since it provides a set of basic tools and tools for complicating the results according to the artists' knowledge and desires.

3.3. Patch-based methods

Another family of methods dealing with manipulation of paths are patch-based methods. These approaches are data-driven and grant authoring control to users in terms of manipulating the trajectories and local interactions temporally and spatially, as well as defining

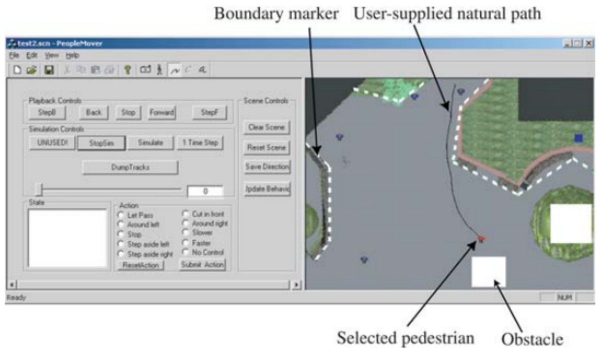

Figure 17: *Example of path specification in a sketch-based interface, taken from [MH03].*

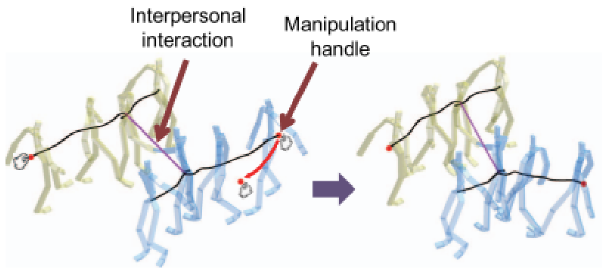

Figure 18: *Example of interactive manipulation of motion patches, taken from [KHHL12].*

the scale of the resulting virtual crowd (e.g., stitching patches together), hence comprising a separate concept from Section 3.2.

The pioneer work has been the Motion Patches method, introduced by Lee *et al.* [LCL06]. They construct environmental building blocks from annotated data, achieving highly-realistic results and interactive performance. Since this is a data-driven method, authoring-wise there is the option of altering the data, for example changing a user-defined parameter affecting data clustering. Still, the main authoring tool emerging from this method is the ability to interactively design the virtual crowd according to user preference, by creating, stitching and editing the motion patches. Expanding on this idea, Kim *et al.* [KHHL12] further study the tiling of motion patches, achieving dense interaction between multiple characters with interactive user control (Figure 18). By a simple drag-and-drop of the manipulation handles, the user can alter the trajectories accordingly. This concept has inspired further research ([SKSY08, YMPT09]). Focusing on solving multi-character interactions, Interaction Patches by Shum *et al.* [SKSY08] expand a game tree and store the resulting precomputations to use at runtime. The high-level user control offered by this method concerns the selection of action patterns, affecting both the agent's path and the interactions with other agents.

However, the most used patch-based method, incorporated for the development of sophisticated authoring tools, is Crowd Patches by Yersin *et al.* [YMPT09]. They can edit trajectories e.g., by mov-

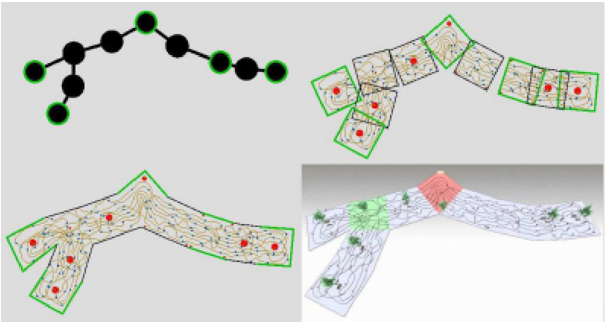

Figure 19: *Deformation gestures for the design of populated environments, taken from [JPCC14].*

ing the crowd patches, making them longer or cutting them. Crowd Patches also enable temporally controlling the agents, in the sense of defining when the agents enter and exit a certain patch. The concept of crowd patches has been utilised in further research ([JPCC14, JCC*15]). Crowd Sculpting by [JPCC14] introduces and integrates space-time mutable elastic models, yielding an authoring tool for creating large, endless, moving crowds. This is done via intuitive deformation gestures such as bending and stretching, as demonstrated in Figure 19. Jordao *et al.* [JCC*15] utilise the crowd patches approach to guide the crowd through the environment. They enable the specification of such flows via a painting interface, providing an artistic tool suited to untrained users.

3.4. Planning strategy

Path-planning algorithms can be split into families of techniques such as: state-space search, potential fields, and sampling-based ([PKL08]). Changing the parameters of the path-planner to incorporate considerations of congestion and stress, for example, will have a significant impact on how the agents will navigate, thus comprising an additional component of authoring paths.

Navigating through virtual environments requires: knowledge of the desired location; some representation of the environment; and certain factors used to evaluate how good a computed path is. Initially, path-planning algorithms were considering factors such as shortest distance to desired location. This, however, is not representative of the real-world as other factors also affect such decision-making process. Earlier work by Pelechano *et al.* [POSB05] use a psychological model to account for factors such as stress, emotion and motivation. These factors affect the path-planning capabilities of individual agents, for instance in terms of orientation.

Environmental factors could also influence agent paths such as agent density ([JCC*15, BNCM14, PAB07, BP15, LCM*17, HOD15]) or temperature comfort requirements ([CJM*18]). These are included under the comfort criterion, instigating motions towards greater comfort, instead of predefined locations. Pelechano *et al.* [PAB07] present their HiDAC (High-Density Autonomous Crowds) system in which mental state parameters, such as impatience, affects the path-planning strategy in the sense that it would cause them to avoid routes with high densities. Additionally, by breaking the task down to smaller sub-tasks and employing multiple domains of control, Kapadia *et al.* [KBG*13] managed to incorporate density and apply their method in dynamic environments. Barnett [Bar14] uses a Reeb graph to infer topology so as to consider congestion and control the flow of crowds.

Another factor which affects navigation is the interests of the agents. In many cases, these are reflected via authoring regions of interest in the environment. For example, Morini *et al.* [MYMT07] define areas of various levels of interest, which are indicated by the user through parameter specification. Following a similar concept, McIlveen *et al.* [MMHR16] support the specification of areas of interest, forcing the agents to change their path to visit such areas while still moving towards their goals.

Expanding on this, altering the environment also influences the planning strategies of agents. These changes do not have an impact on the factors used in the cost function of the planner, but still give authors the necessary control over the environment, either by allowing changes off-line or in real-time. Paris *et al.* [PDB06] construct a hierarchical abstract graph to create agents capable of responding to changes in the environment setup. Virtual environments might also include forbidden or hidden areas. Ninomiyia *et al.* [NKS*14] deal with authoring these environmental aspects focusing on agents avoiding being seen or passing through certain designated regions by employing a graph-based approach to find such constraint-aware paths. Another direction attempts to bridge the gap between path-planning and path following (local) by allowing control over the time component ([NLS14, KSHF09]). In particular, Kapadia *et al.* [KSHF09] address this issue using affordance maps in an egocentric manner.

3.5. Industry software

Most industry-developed tools, provide their own implementations of an A* path planner on top of a Navigation Mesh representation of the movable space. Authoring-wise, the majority of such systems do not allow for the modification of the path planning cost function, which is the fundamental way of modifying the planning strategy. Most of these however, allow for limited *implicit* control of the strategy, by modifying agent-related properties such as the agent radius, maximum step-size or inclination the agent can move on, etc ([Unic, Engc, Hav]). These parameters affect the generation of the Navigation Mesh (walkable area) which, in turn, affects how agents can move in the environment. An illustration of the effect of modifying three of such parameters on the resulting Navigation Mesh is shown in Figure 20. These systems also allow for minor editing of the Navigation Mesh such as linking its vertices in order to allow for agents to jump (for example) between separate meshes.

Several commercial softwares that are used in the movie industry [Goa, MAS], allow for direct control of the flow/formation of groups of characters by sketch-based interfaces (i.e. artists draw paths/flows). In these applications, authors typically care for full control of crowds in specific shots, and therefore the generated crowds are simulated for a short duration of time and satisfy artists' requirements precisely.

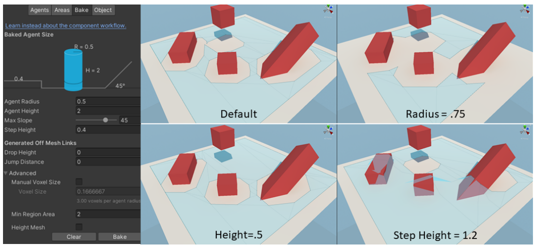

Figure 20: *The effect of changing radius, height, and step height parameters on the corresponding navigation mesh in Unity.*

3.6. Summary

Summarising, guiding a group of agents by authoring the desired flow has been successfully used in practice, producing large crowds moving with other crowd members in a coordinated manner. The authoring tools for this are significantly advanced, facilitating the use by non-experts. Commonly used interfaces include sketching of arrows, which is both efficient and easy. Some methods for flow control offer additional functionalities either in the form of automatic collision avoidance or authorable components such as lane formations. Explicit control of group formations has also been explored, with the state-of-the-art managing to reflect the desired formations with sufficient detail. Authoring such formations is widely addressed via sketch-based interfaces, mainly necessitating the sketching of formation boundaries.

Designing global paths mostly leads to uniform behaviours. In an attempt to introduce some autonomy and variety, other research focused on controlling the paths of individual agents independently. Drawing arrows or dragging gestures are the common authoring tools for this task. Patch-based approaches are a special category allowing multiple types of high-level user control, from entry/exit points to environment/crowd design. Commonly-used interfaces for authoring in the context of patch-based approaches are drag-and-drop interfaces, but there are some crowd-patch inspired approaches also employing gesture-based and painting-based interfaces.

The last aspect of path-planning that we discussed is the planning strategy and the degree to which it can be authored. Integrating factors into the path-planning framework enables authors to guide the decision-making process of the agents by controlling these factors either manually (i.e. parameters) or via an API. Such control is useful when agents need to navigate by considering their surroundings while reaching a pre-defined destination, also having some intuition (criteria/factors) as to how to do it. High-level behaviours impact this level in terms of the criteria considered for path-planning; emotional states and intelligent behaviours change the way the agents move to reflect these attributes.

4. Local Movements

In real life, apart from accounting for our long-term goals and needs, we need to perform some lower-level, short-term tasks to satisfy those needs. As with real life, virtual agents must prepare to deal with local movements like collision avoidance, local behaviours, and interactions with the environment and other crowd members. Adopting strategies to deal with these aspects is a secondary but necessary step to realise their goals and wishes. Past literature has extensively examined such methods, developing sophisticated collision avoidance and local interaction frameworks.

Several aspects of local movements can be edited by the user, either controlled to achieve an intended outcome, or described to reflect authors' wishes. Having influence over how agents perform local movements is a desirable attribute as it allows us to turn our attention to the details of crowd generation. In applications like computer games, we need agents to be able to react and adapt to changes in their immediate environment, such as the break-out of a fight in the middle of a street, while also having a global purpose. Additionally, in architectural studies and training systems, such local movements are important to be studied, understood and later on modified in a manner that represents human behaviour.

Since the scope of this survey concerns the task authoring crowd simulation components, in this section we discuss two aspects of local movements which can be edited: avoidance strategy, and local interactions.

4.1. Avoidance strategy

Collision avoidance is a crucial component of all moving virtual crowds, incorporated in the overwhelming majority of crowd simulation systems. Therefore, past research has explored several ways to improve collision avoidance strategies between agents and static objects, dynamic objects, and other moving agents. However, most of them are fully automated, not giving authoring privileges to the user. A lot of the current systems usually use the functionalities of game engines for this task. For further information on collision avoidance algorithms and the recent advancements, the reader is referred to [vTP21]. For the scope of this survey, we focus on work allowing some degree of user intervention.

Force-based. Early works on collision avoidance, develop, employ and build upon force-based approaches which are conceptually straightforward. In principle, this family of approaches deals with how to best define the forces subject to which each agent moves, so that the resulting movements are collision-free. Helbing and Molnar [HM98] initiated the research in this area by assigning attractive and repulsive forces to the agents, creating a SFM (Social Force Model). Later on, predictive methods gained popularity with [KHvBO09]. Authoring such avoidance strategies mainly regards controlling the parameters of the prediction algorithm, such as time and distance to collision. In particular, Karamouzas *et al.* [KHvBO09] employ a collision prediction algorithm on a SFM, given the time-to-collision, thus allowing users to control the way agents avoid each other by manipulating this prediction parameter. Zanlungo *et al.* [ZIK11] also make use of the time-to-collision parameter, while Karamouzas *et al.* [KSG14] employ a universal power law that is based on energy minimisation affected by the projected time-to-collision.

Velocity-based. In the late 2000s velocity-based approaches made an appearance with [PPD07]. Velocity-based approaches involve looking at several options for future velocities, evaluating them based on a set of criteria, and allowing the agents to move according to the best option. This evaluation stems from a defined cost function. Authoring-wise, having influence on this function affects how agents decide to move. The two mostly used methods of velocity-based approaches for collision avoidance are RVOs (Reciprocal Velocity Obstacles) by [vdBLM08], and ORCA (Optimal Reciprocal Collision Avoidance) by [vdBGLM11], the former representing sampling of the velocity space, and the latter corresponding to identifying a specific optimal velocity. The cost function used in [vdBLM08] includes a parameter via which the apparent aggressiveness or sluggishness of individual agents can be reflected. Based on RVO, Karamouzas and Overmars [KO10] set a cost function of three main factors including time-to-collision, along with the difference of sampled from current velocity and desired velocity. With the use of three corresponding weighting parameters, users can adjust the contribution of each factor, achieving a variety of avoidance behaviours. ORCA ([vdBGLM11]) and its derivative approaches like in [GCC*10], follow the same concept only allowing for limited, low-level control via programming.

More recently, Ji and Han [JH15] present a novel velocity-based method inspired by real-life humans for local collision avoidance, based on the least-effort principle. Karamouzas *et al.* [KSNG17] show that other techniques do not respond well to changes in the time horizon, whereas their proposed optimisation-based method is robust. Additionally, Van Toll and Pettré [vTP19] employ a novel topology-driven method to detect potential collisions and trigger a re-planning process. Through adjusting the re-planning time parameter, the user gains implicit control over the avoidance strategy.

Vision-based. In contrast to approaches we reviewed so far, when humans perform collision avoidance they do not assume a global knowledge of their surroundings, but they get their strategies based on the visual information they receive. This gives rise to vision-based approaches ([War06, PRY13]). The novel vision-based method introduced by Ondrej *et al.* [OPOD10] reflects this phenomenon by predicting collisions from visual stimuli. Their method is reactive and depends on certain user-controlled attributes, such as anticipation time and security distance from obstacles. Users adapt these parameters by setting the corresponding weights, directly affecting the avoidance behaviour of the agents. Extending the work on vision-based steering, Dutra *et al.* [DMCN*17] introduce a gradient-based solution and give users authoring rights via the movement and obstacle cost functions. The changeable parameters, again affect anticipation time and minimum distance to obstacles, influencing the avoidance strategy directly (Figure 21). In addition, methods using optical flow and light, are governed by a control function to enable agent movement ([LCMP19a], [LCMP19b]). In that, authors can assume control of the functions' parameters to impact the avoidance strategy. For instance, Lopez *et al.* [LCMP19b] enable parameter tweaking resulting in varied adaptation times and freedom as to detour exploration.

Data-driven. More recently, data-driven and learning-based methods have received increasing interest from numerous domains. For the purposes of collision avoidance, literature has taken advantage of such methods to alleviate the need for "handcrafting" rules or functions. The most intuitive way to use data for learning purposes is via searching datasets. In particular, Charalambous and

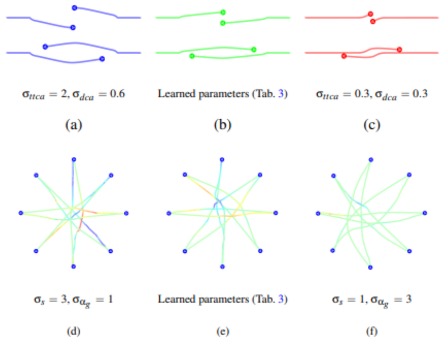

Figure 21: *Impact of these user-adjusted parameters on the agents' motion regarding collision avoidance (Taken from [DMCN*17]).*

Chrysanthou [CC14] build a perception-action graph to represent a clustering of the data, which facilities searching the database. On the authoring aspect, except for changing the input data, any change in the resulting graph such as the neighbours considered and the temporal representation, leads to different simulation outcomes. Following the concept of clustering, Zhao *et al.* [ZTC13] use simulated data and employ an ANN (Artificial Neural Network). For this family of methods, altering the dataset of examples, gives authors the power to significantly affect the outcome ([BKSB15, RXX*21]).

Additionally, real-life data can be utilised by deep learning networks. Most work focuses on character animation, but research like [AGR*16] and [GJF*18] use deep learning for trajectory prediction in the context of multiple characters (LSTM and GAN respectively). Most deep learning models aim to fully automate the learned behaviours, with user intervention being at the selection of the criteria based on which the network learns, which are usually fixed and defined before training. In the case of [AGR*16] and [GJF*18] the network is trained to minimise trajectory deviation from ground truths, and a variety loss to produce diverse, socially-accepted behaviours, respectively. Reinforcement learning (RL) comprises another set of learning-based methods, for which users define the reward function. In recent research on RL, Casadiego and Pelechano [CP15] address the collision avoidance aspect by penalising when agents get too close to obstacles through the rewards function. Recently, Haworth et. al. [HBM*20] use deep RL attempting to provide fully integrated physical character control. However, this is the lowest level of user control and for authoring purposes, this implicit control does not suffice for intuitive manipulation and description according to the users.

4.1.1. Mental state

Sometimes, to ease the process of specifically understanding how each of the parameters of the local movements affects the look and appearance of the simulation, researchers created some additional layer of psychological aspects that is mapped to these low-level parameters, enabling the user to then simply define a smaller set of parameters, which are easier to understand e.g., aggressiveness. This is a typical attempt to reduce the number of parameters that the user has to specify to get a desired local movement result.

When psychological models were first introduced to crowd simulation, they were mostly reflected in the local movements, with some high-level aspects e.g., path-finding. For the personality feature of being patient or impatient for example, at the lower level, agents would be pushier and try to move faster, while at the higher level, they would try to get shortcuts or avoid paths with bottlenecks. So far, personality aspects have been mostly included in the local movements because it is easy to exhibit these personality differences e.g., speed of movements and distances for avoidance are easily perceived. Previous research by Pelechano *et al.* [POSB05] employ a psychological model affecting, apart from path-planning, local motion in terms of speed modification subject to the agents' mental state. Additionally, Guy *et al.* [GKLM11] present a method to map personality attributes to low-level parameters, by employing a perceptually-driven method.

4.2. Local interactions

Apart from the avoidance strategy, an additional aspect of local movements, which can be edited to conform to the user's specifications, is local interactions. Here, we turn our attention to methods which provide some control over how agents interact with each other or react to the environment, still preserving their global goals. Therefore, we look at interactions locally, having short-term effects. Mainly, we study group behaviours and environment interactions, in the context of authoring.

4.2.1. Group dynamics

Although agents may share common global goals, locally there is still the possibility to model agents' social behaviours. This mainly involves examining group formation, following behaviour and flow. A pioneer in the domain of such group formations is [Rey87, Rey99] with later work exploring force-based methods as well as other innovative approaches ([MPG*10]). Although literature often focuses on finding ways to automatically model such formations to capture the underlying social implications, our focus is on the controllable components of such systems giving authoring capabilities to users.

In particular, Qiu *et al.* [QH10] offer explicit control over the resulting group formations via user-defined matrices relating to the incentive of agents to follow and influence others. Other model parameters provide authoring capabilities for formation details such as distances between agents within a group. Following the concept of providing control through user-specified matrices, Ren *et al.* [RCB*17] base their method on velocity obstacles and enable the authoring of local interactions within groups via relation matrices. An important attribute of the method is the dynamic manipulation of the controllable parameters, achieving authoring at run-time. Figure 22 demonstrates the effect of one such parameter (group weight) and how authors can use it to force groups not to break into clusters when densely interacting with other groups. This allows for more realistic, socially-plausible virtual crowds.

4.2.2. Following behaviours

More recently, more automatic ways for group emergence have been studied ([LA15, hPN*16]). User intervention for these methods is limited to parameter manipulation. In particular, He *et al.* [hPN*16] concentrate on group emergence and pass authoring control to users, who can dynamically change number, shape, and size of the groups. Their method adapts to changes in the environment to handle large, dense crowds. Another category of literature, emerged in the 2010s, studies the speed adaptation of agents in scenarios necessitating following behaviours. In particular, Lemercier et. al. [LJK*12] experimentally explore such interactions and build a model simulating them. Their model consists of changeable parameters corresponding to agents' reaction times, density, and acceleration amplitude. This research deals with following behaviours in the context of circular queues. Additionally, van Goethem *et al.* [vGJCIG15] focus on local flows in order to allow agents both independent and coordinated behaviours. They achieve this by defining incentives and streams, with incentives reflecting the agents' motivation to be a part of or leave a coordinated group.

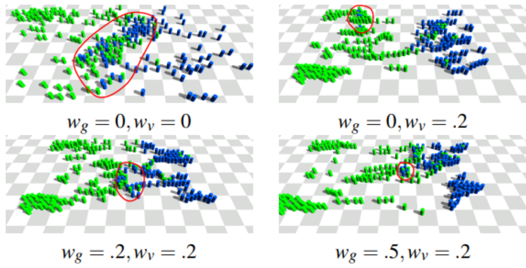

Figure 22: *Impact of user-specified "group weight" parameter on the group dynamics, taken from [RCB*17].*

4.2.3. Other local interactions

Apart from group dynamics, Sung *et al.* [SGC04] turn their attention to simulation-based method (not agent-based) to author the reactions of agents associated with their immediate environment and implement them by sampling a probability distribution. Figure 23 shows how a painting interface could be used to specify situations. Using such interface facilitates graphical control of the environment which, in turn, affects the way agents react, locally. Kapadia et al [KSHF09] address space-time planning to account for complex interactions of agents with their immediate environment, other agents and objects. Using affordances and employing an egocentric method, they resolve these local interactions in complex situations e.g., deadlocks. Ju *et al.* [JCP*10] change the distribution of local interactions, and define those local behaviours so as to resemble some example provided beforehand. McIlveen *et al.* [MMHR16] create an authorable environment and provide a graphical interface which, among others, allows for specification of attraction and avoidance areas, encouraging agents to pass through these areas on the way to their goal through individual layers of FVFs.

4.3. Debugging and evaluation tools

In terms of real-time interfaces for evaluating and debugging simulation outputs, past work focused on local movements' evaluation ([KSA*09, SKFR09, GvdBL*12, BKHF15, KSHG18]). For instance, Charalambous *et al.* [CKGC14] have a visual interface where you can see the simulation but also identify outlier behaviours with

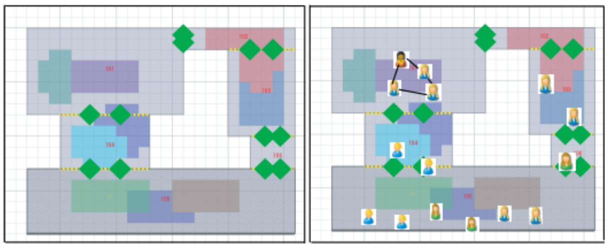

Figure 23: *Example of specifying spatial (left) and non-spatial (right) situations via a painting interface, taken from [SGC04].*

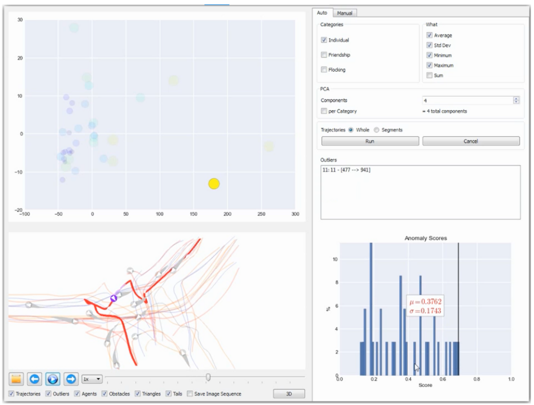

Figure 24: *Outlier identification, from the video of [CKGC14].*

regards to certain crowd attributes (Figure 24). Wolinski *et al.* [WGO*14] provide an optimization framework to find optimal parameters of simulators to match reference data for fair comparisons between simulators. More recently, Daniel *et al.* [DMH*21] combines feedback from experts (user evaluation) with metrics to form a quality function. However, we identify the need for evaluation of more high-level aspects of crowd simulations.

4.4. Industry software

Nowadays, the tendency to develop authoring tools for avoidance strategies in practice, is limited. The majority of frameworks and commercial software rely primarily on game engines' functionalities for high performing automatic collision avoidance algorithms. Authoring abilities are limited to defining a small set of parameters reflecting (for example) the agent size, collision radius, distance to other agents, acceleration and velocities. For experienced users, some aspects of the collision strategy (i.e. time horizon) can be programmed explicitly. Typically, designers resort to using high-level behaviour tools (e.g., Behaviour Trees; see Section 2.3) to control local interactions such as group dynamics (e.g., following behaviours).

4.5. Summary

In summary, the avoidance strategies employed by the majority of literature are controllable by means of parameter manipulation. Avoidance techniques are advanced enough to be exploited in several applications which include complex environments with dynamic obstacles and agents moving in close proximity to each other. There are still some disruptors of realism linked to avoidance algorithms such as reversals. Allowing agents to stop via UIs or parameter manipulation, prevents them from turning unnecessarily and thus mitigates this issue to an extent. Incorporating mental state aspects is typically an attempt to reduce the number of parameters to be modified by the user, and to make the internal parameters of each model more transparent to the user.

Local movements can also refer to local interactions, the ways in which agents adapt their movements in accordance to environment changes or their intentions for participating in local, group behaviours. The authorable components for this, range from low-level parameters, to matrix specifications, to graphical interfaces (especially for reactions to environment changes). Current state-of-the-art is successful in accommodating local interactions and behaviours in meaningful environments like city streets, supermarkets, malls, and theatres.

5. Body animation

Yet another component for producing lively virtual crowds is animating the characters, thus creating 3D moving agents. To this end, motion capturing systems have been developed to aid the transformation from specified positions and velocities to seamless 3D movements. Recently, extracting information from data of real humans is used as the basis of 3D animation, since such information is essential both for creating animated avatars exhibiting simple behaviours (locomotion: walking, running), and adding variety to these avatars in terms of tasks and movement style. However, we note that the range and parameters of body motion variability to be taken into consideration, is not intuitive and thus is task-dependent.

Authoring animation refers to the ability to control aspects of it, which will lead to 3D animations conforming to the author's wishes. In practical applications, several game engines offer motion or character templates to allow authoring at the animation level. Golaem's commercial software revolves around composing assets representing categories of characters with similar characteristics. Since 3D animation is a visual criterion determining the believability and naturalness of the crowds, being able to author its aspects is of particular importance.

For the scope of this survey, we present two aspects of 3D animation which can be edited by users: the animation clips; and the motion variations. This section provides an explanation of the most popular methods used to author each aspect and discusses how such manipulations affect the animation.

5.1. Animation clips

Data-centric approaches include methods utilising motion capture or synthetic data. The main result is, in both cases, some type of database of animation clips. There has been an overwhelming amount of work on animation, but for the scope of this survey we focus on authoring. Users can edit the animation clips by blending or interpolating between them, as well as perform other operations such as rearranging and cutting them. It is, however, important to

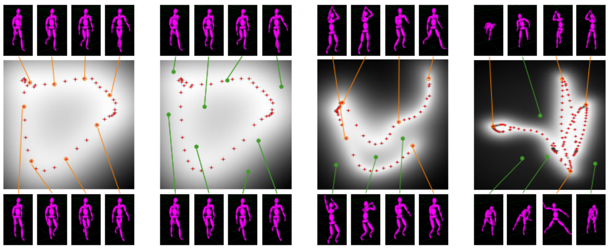

Figure 25: *Effect of using different input data on the learned latent space, taken from [GMHP04]*

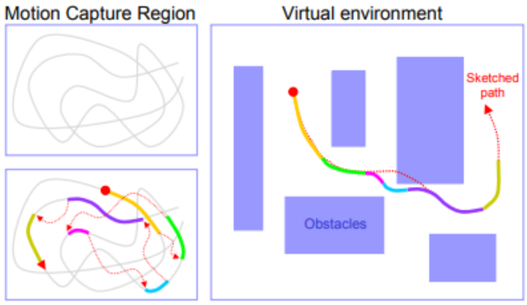

Figure 26: *User interface, taken from [LCR*03]. Through search of the captured data (top left), a series of suitable motion clips are selected (bottom left) to match the sketched path (right).*

note that finding the best blending method does not benefit authoring, since it affects the animation quality rather than the content. Additionally, the animation clips can be recomputed by changing the input data or editing the motion capture data e.g., via layering or splicing. Animation clips can also be used to construct some sort of data structure (e.g., motion graphs) to facilitate authoring.

5.1.1. Blending

Having animation clips, one common operation needed to produce a seamless animation according to the desired story-line, is blending multiple clips (e.g., blending walking and running motions according to the desired speed). Typically, several animation clips are represented as points in a 2D space, and then blending weights are computed based on the distance between the desired outcome and the points representing each animation clip. Game engines use this type of blending, which is known as Blend Trees Unity ([Unia]), and Blend Spaces in Unreal [Engb]. Users are normally allowed to manipulate the blending parameter, the weighing of the interpolation, or the interpolation function.

In research, various motion blending and interpolation techniques have been explored for the purposes of using current animation clips to produce novel movements according to some desired outcome. For instance, Rose *et al.* [RISC01] address the inverse kinematics problem by employing and interpolating RBFs according to artist-generated examples, while Huang *et al.* [HK10] present an optimisation-based method to find blending weights, subject to user-defined spatial constraints. Additionally, Grochow *et al.* [GMHP04] synthesise new motion using a GPLVM (Gaussian Process Latent Variable Model) approach. Authoring-wise, they achieve different motion styles, by training on different input datasets (Figure 25), as well as developing a method for interpolating between styles, according to some interpolation parameter.

5.1.2. Transitions

The other set of solutions, used for animating a crowd, are the motion graph approaches, introduced in [KGP02]. Instead of blending, these approaches do not directly modify the motion capture data, but set-up animation graphs and look into ways to choose transitions from one piece of motion to another. The analysis of the methods yielding the best transitions is outside of the scope of this survey, since they affect the quality rather than the content of animation. For authoring, users can control animation clips by rearranging and concatenating them ([RGBC04,KGP02,LCR*03]), or manipulating them according to certain constraints ([LS00]). Lee *et al.* [LCR*03] provide an interface through which the user can sketch the desired path, which constitutes the reference for finding the suitable animation clips, as seen in Figure 26.

More recently, an alternative way of selecting transitions between motion clips is Motion Matching, presented in [Cla16]. Instead of selecting transitions prior to the implementation, Motion Matching allows control over which animation clips by modifying the predicted trajectory, adjusting the influence of past paths, as well as defining the set of joints to be considered by the matching algorithm. This indirectly affects motion diversity; when only considering feet joints, the system searches a wider space of possible clips. As with other methods in this section, changing the database also influences the associated animation.

The field of animation is very large, with successful work on authoring animations being done by research labs like Disney (e.g., with [Res]), allowing for drawings and gestures to exaggerate movement or several refinements and corrections. However, these mostly focus on one character at a time and have not been easily extended to crowds so far. Some work has been done for small groups of two-three characters (e.g., [SKY07, KCPS08]). Recent work Starke *et al.* [SZZK21] develops a learning-based method for a two-person martial arts simulation, achieving high-quality results and allowing control over the affected body parts.

Although literature on animation is vast, in research there is not a lot of work attempting to provide authoring privileges over the animation of crowds. The most common way to author animations for crowds, revolves around explicitly triggering animations. Ulicny *et al.* [UCT04] use a brush metaphor to trigger animations according to brush operators (Figure 27). The users can apply these brush operations on a selected group of characters, forcing them (for example) to transition from walking to running. Additionally, Beacco *et al.* [BPKB15] utilise animation clips to allow characters to take different footsteps, following a footstep trajectory. They determine the best interpolation of animations to get feet correctly located over the terrain. This applies to groups and crowds, and authoring relies on inputting animation clips to provide as much variety of footsteps as possible, to get better interpolating positions. Environ-

Figure 27: *Visualisation of brush metaphor, taken from [UCT04].*

ment triggers also affect the animation, for example when the animation has physically-based components (e.g., uphill and downhill motion) and users change the outcome of the animation by changing the environment, without affecting the animation algorithm.

5.2. Motion variations

Most blending trees and motion graph approaches lead to uniform animations. Therefore, literature has dealt with a second perspective of 3D animation, adding variety, to enrich realism. This could pave the way for several interesting applications, such as creation-support systems for choreographers ([SBT10]). Motion variety and appearance variety are both important in the perceived realism of the crowd, however, we leave the discussion about visualisation aspects in Section 6. Several perceptual studies have investigated the perception of variety in virtual crowds, with the intention of revealing the key components affecting the perceived variety ([MLD*08, PO11]). The findings aim to provide the fundamentals for building motion templates useful to create believable, diverse crowds with minimal effort.

In a crowd setting, the need for motion variety among its members is even more prominent due to the presence of character interactions, which makes achieving believability even more challenging. Fitting behaviours by Lerner *et al.* [LFCCO09] add secondary motions to the characters to make them more human-like by means of an action graph, storing examples obtained from videos. Their system facilitates authoring by providing control over the frequency of action occurrence and the frequency of specific actions. Additionally, Lacroix *et al.* [LMK09] utilise the concepts of norms to facilitate behaviour descriptions, and generate varied behaviours. The system enables the creation of behaviour violations, leading to unspecified behaviours, further enriching diversity.

In the 2010s, Gu and Deng [GD10] dealt with this issue, while allowing specification of parameters impacting local diversity, consistency, and global style utilisation. Maïm *et al.* [MYT09] vary the motion by having the characters holding a phone to their ear, using an IK-based system to adapt the characters' postures. In more recent work, Hoyet *et al.* [HOKP16] address the perceived realism in dense, interacting crowds via incorporating shoulder motion for agents passing each other in close proximity.

Another beneficial characteristic is the ability to produce diverse animations, without the need to recapture data, but modify existing motions instead. Previous literature has studied this on a single-character level due to computational considerations, and addressed the topic either via PCA (Principle Component Analysis) or statistical methods. For the former, Min et al [MLC10] present a simple interface allowing for authoring identity and style by dragging points. For the latter, Ma *et al.* [MXH*10] utilise a Bayesian network for obtaining multiple variations of user-specified style parameters.

5.3. Industry software

There are many tools dedicated to character animation; these typically integrate some motion databases and allow for authoring and editing character animations. However, an extensive discussion of these tools is out of the scope of this survey. Some of the most popular ones include Autodesk's Maya ([Autb]), 3D Studio Max ([Auta]) and MotionBuilder ([Autc]), Reallusion's iClone ([Reab]), Mixamo ([Ado]), etc. Several game engines also allow for the manual design of motion graphs between motion clips (e.g., Unity's Mecanim system [Unib]), the creation of Blend Trees and more recently Motion Matching. In general, commercial software for editing animation clips usually allow a mixture of drag-and-drop interfaces with coding, enabling authors to access the scripting editor. There are plenty of tools for authoring animations going into much detail such as modifying the curves for each specific body joint. However, this requires some familiarisation with the system and so is limited to experts. In terms of motion variety, some commercial software (Golaem [Goa]) allows artists to define motion templates.

5.4. Summary

We have seen two main authorable aspects of body animation. The first has to do with the types of editing of animation clips, whereas the second is concerned with adding variety for increased realism, in a manner controllable by the user. Most common ways of authoring for this category include a combination of programming interfaces, gesturing, drag-and-drop, and coding conditions for transitions. Such interfaces could be adjusted to abide by high-level behaviours as they can be used to inform the animation and vice versa; the animation and available data could give rise to certain behavioural characteristics.

6. Visualisation

An additional component, critical to how virtual crowds are perceived, is the visualisation of characters. The visual aspect can significantly enhance the apparent variety among the crowd, give characters a sense of identity, and increase believability; authoring this component helps as to how the scenes are presented and perceived.

Many application domains utilise visualisation authoring as an easy way to increase realism. Computer games often turn their attention to appearance variations, changing the clothes, gender, and colours. In entertainment, especially in movies, the lack of control over these visualisations can heavily taint the essence of a particular scene. Having a realistic-looking crowd, directs the attention of the viewer to the intended plot (e.g, actions, emotions). Additionally, authoring the visual aspect is important when it comes to audience perception. Perceptual evaluations highlight the relevance of the kind of realism in crowds; character-level, or group-level. Viewers' attention is directed towards the content, rather than the appearance; if the crowd is poorly done, that will likely detract from the audience's attention, whereas if the characters seem realistic, the audience will focus on the content of the simulation.

Here, we relate authoring such visualisations with the character and crowd characteristics which can be changed, prior to incorporating motion. Therefore, in this section, we talk about: skeleton and body mesh; textures and colours; and accessories.

6.1. Skeleton and body mesh

Humans 3D models are composed of two main components for representing the skeleton and the skin. The skeleton is a union of vertices that define important aspects of the character such as its height and length of extremities. The skin is represented by a 3D mesh that covers the skeleton. These components can be modified to generate a variety of characters from a single human model.

For instance, Maïm *et al.* [MYTP09] allow designers to specify how to deform the character skeleton and skin for generating variations from a single body template. They use FatMaps which are extra grayscale UV textures for defining the areas that can be deformed in the shape variation process. Additionally, they can define the minimum and maximum scale parameters for tuning the skeleton's joints. Similarly, Galvao *et al.* [GLD08] provide authoring tools for interactively generating displacement maps that define body mesh regions and skeleton vertices to be scaled, and their respective scaling values. These tools are employed directly over the body mesh, rendered using a brush metaphor. Furthermore, Kasap *et al.* [KMT09] derive variations of skinned human templates by identifying, deforming and sizing the different body regions using anthropometric body measurement standards. Then, the skeleton is adapted to the new body size. Moreover, they use Free Form deformation method to constrain the regions where the deformations will be performed. The next year, Kasap and Magnenat-Thalmann [KMT10] improved the previous method by employing a different segmentation body technique that uses the skeletal joints for identifying the body regions. Niswar *et al.* [NKF12] present a customisation algorithm based on 12 human body measurements to modify the avatar's shape by scaling its mesh's vertices globally and locally.

In addition, Shi *et al.* [SOWO17] implement a data-driven approach to generate a variety of body shapes using the body measurements and demographic data available in the CAESAR database. They found, by conducting perceptual studies, that the height and girth (waist circumference) measurements were the most salient features that differentiate people from each other. Then, they used these attributes to distribute and select the set of data entries to be modeled as a 3D body shape. Moreover, Silva *et al.* [SdSdSH19] generate variants of body meshes with different levels of obesity, using the body mass index (BMI) as parameter. Their model considers how each different body part is affected when the BMI parameter varies.

6.2. Textures and colours

Tinting the t-shirt, pants, skin and hair with different colours is a popular technique used to provide variety in a crowd simulation composed of a few unique character representations. The general approach consists of identifying and encoding the different body parts of the body mesh in base textures for applying random colours, considering the valid values for each body part.

Tecchia *et al.* [TLC02] combined texture compression with multipass rendering techniques to modify textures on the fly and display a diverse crowd of virtual humans. Their approach takes advantage of the alpha-channel for manually identifying, storing and colouring the different regions of the body. The body-parts identification and encoding process was performed using Autodesk 3DS Max. This technique has been widely accepted and employed for giving variety to crowds with few unique body meshes ([DHOO05], [MLD*08]). Similarly, De Heras *et al.* [dHCSMT05] make a few grayscale base textures to create a variety of avatars differentiated by the colour of their clothing and accessories; this approach is based on alpha map channels encoding and deals with situations where triangles in the mesh template correspond to more than one region with different colours. The same year, De Heras *et al.* [dHCSM*05] extended their contribution by implementing the fragment shader; a hardware approach used to assign the colours to pixels, based again on the texture alpha-channel encoding. Moreover, to avoid unsuitable colours for specific body parts, they use the HSB colour model to control the variety of colours adequate for each body zone. Finally, they implemented a GUI providing facilities to author several aspects of the crowd, such as the human template, texture and define colour ranges for a body part.

Later work by Maïm *et al.* [MYT09] develop a different technique for encoding the body parts in the texture of human templates using a set of segmentation maps. This encoding method allows to associate a single texel with several body parts at the same time, improving visual aspect of the transitions between body parts. Gosselin *et al.* [GSM05] use per-pixel lighting technique to improve the details of the characters they render. Furthermore, they give variety to their crowd by combining a mask texture that indicates the different zones of the body with a small 2D texture that contains combinations for darkest and lightest tinting. Finally, they add random decal textures for uniforms to provide more unique characteristics to the crowd. Additionally, Ulicny *et al.* [UCT04] use their brush metaphor to give new scaled colours for the current texture to the selected characters in real time. An example of such operation is illustrated in Figure 28. Moreover, Golaem has a tool which allow for changing the appearance of a character using a statistics-based method using variations provided by the user.

6.3. Accessories

Accessories such as jewellery, watches, shopping bags, cellphones and wigs, can be modelled as small meshes. Adding accessories to humans templates is a method used to vary the shape of the body mesh and can significantly improve the variety of the crowd.

Maïm *et al.* [MYT09] attach accessories to skeletons in human templates (Figure 29). They model their accessories providing them with details for correct placing and orientation, according to each

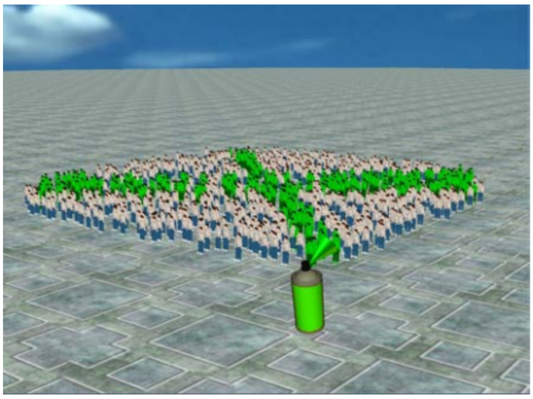

Figure 28: *Illustration of colour brush, taken from [UCT04].*

Figure 29: *Illustration of utilisation of accessories and segmentation maps of five human templates, taken from [MYT09].*

available human template's mesh. Firstly, they define the skeleton joint for attaching the element, that can vary in some cases because of their height or any other physical aspect of the human template's mesh. Secondly, they perform some transformations relatively to the attached joint, to make them perfectly coincide with the mesh. They also include an accessories categorisation of type and theme that ease the designer's selection task when generating a population with specific characteristics. The same year, Maïm *et al.* [MYTP09] extended their method including solutions for complex accessories (cellphones, bags, flowers, etc.) that require changes in the human's animation by constraining the skeleton's joints.

6.4. Industry Software

The practices for authoring the visualisation aspect of virtual crowds are pretty consistent throughout most of the current systems. More specifically, for systems like MakeHuman, Mixamo, and Character Creator ([Hum, Ado, Reaa]), even for setting up your own avatar in game platforms, the process is typical: go through body parts and properties; for each of them go through drag-and-drop interfaces with some kind of GUI/sliders for specification and fine-tuning. Software used in the industry enables authoring the distributions of certain characteristics like accessories, giving control to the artist as to what proportions of the population should have them. Also, there are tools (e.g., from Golaem) which facilitate the specification of body shapes, easily and quickly.

6.5. Summary

Briefly, the authorable aspects of visualisation of crowds can be split into: skeleton and body mesh; textures and colours; and accessories. Generating variations of a small set of human templates enhances the believability and diversity of virtual crowds without increasing the high design and memory requirements. Defining the variations of such aspects, usually is achieved through interfaces which allow authors to identify and encode the different body areas in the skeleton, body mesh and textures and select the range of values over which these components can vary. Finally, the system follows the editing parameters to modify the original human template in render time and generate crowds with a variety of characters. Generally, visualisation is essential in communicating emotional state or personality characteristics to the audience, once again highlighting the impact of Section 2 on the remaining components.

7. Post-processing

The last component of crowd simulation is post-processing (i.e. editing the finished product). This refers to editing various aspects of the virtual crowd, without necessarily being familiar with the methods and processes followed to obtain that simulation. Instead, users can easily and effortlessly bypass the technical details and just deal with manipulating the end-result as they wish.

Post-processing is important to simulation systems, with the majority of applications benefiting from the ability to refine the produced crowd interactively. From movie scene visualisations to urban studies, editing allows for spotting and correcting errors easily, and enhances the authoring process by assisting the users realise their vision. In contrary to other sections, this is not limited to a single aspect, but can apply to all the previously discussed simulation components. In terms of editing high-level behaviours, Mathew *et al.* [MKMA19] support the adjustment of crowd behaviours after the development of the simulation. The types of adjustments that their system facilitates are in terms of density, goal destinations, and environment topology (Figure 4).

Perhaps one of the most common aspects of post-processing is the editing of agent paths. Post-processing also affects local movements, as several systems provide the ability to edit the movements parameters. Such parameters may correspond to temporal synchronisation and spatial deformations as well as other supplementary attributes such as velocity changes. It is also common for most systems which offer post-processing tools for local movements, to also support editing the agents' paths.

A primary example is the method presented in [KHKL09], which enforces agent trajectories in the form of user-defined constraints in an optimisation framework. Laplacian motion editing is popular among literature to account for modifications of agent trajectories. Kim *et al.* [KHKL09] also turn their attention to synchronising characters, thus facilitating local interactions with agents and the environment. Through pinning positions and directions, their methods allow for further detailed authoring of such interactions as shown in Figure 30. Similarly, the method which is introduced in [KSKL14], offers a tool for controlling crowds interactively. The tool is straightforward and enables the manipulation of the temporal as well as the spatial aspect of the movement of the virtual

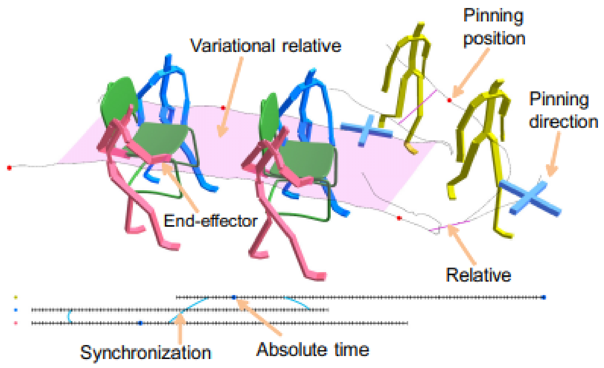

Figure 30: *Example of post-processing manipulation of local interactions and velocities, taken from [KHKL09].*

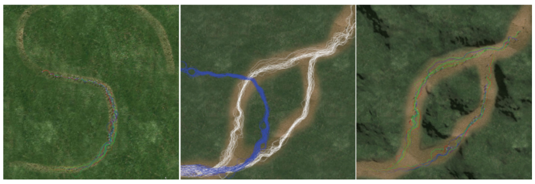

Figure 31: *Example of post-processing manipulation of local interactions and velocities, taken from [KLLT08].*

crowd. To allow for this level of authoring, Kim *et al.* [KSKL14] utilise as-rigid-as-possible deformation integrated with cage-based deformation, achieving detailed authoring of agents in a small town scenario, for instance. Additionally, they enable gesture-based deformation of a set of selected paths.

Another approach is to manipulate movements by controlling the velocities of agents either individually or as a whole, at certain times or locations. The benefits of tweaking movement parameters depend on the scope of the intended application of the authoring system. Such capabilities provide low-level control as the author specifies more details, which is desirable to achieve greater realism and naturalness. Allain *et al.* [ACC14] propose a method to control the simulation by editing the crowds regardless of the underlying model used. They allow users to specify a range of constraints including velocities, densities, or target locations, thus affecting both the local movements and high-level behaviours.

In the context of editing a set of paths affecting a group of agents, Kwon *et al.* [KLLT08] present a group motion editing technique where, via intuitive deformation gestures, users can deform selected paths or stitch them together (for example) to produce longer animations (Figure 31). To this end, they use a graph-based method, making use of Laplacian deformation. Their method automatically deals with temporal and spatial errors, further alleviating the authoring burden (i.e. the need to specify unimportant details). Several works dealing with formations, also allow the option of reforming these formations at run-time ([XWY*14], [BSK15]).

Another set of systems provides refining options. For instance, data-driven approaches as in [LCHL07] allow user intervention to modify and correct potential unrealistic results derived from their method. There are also some systems which, even though their primary goal is not the editing of the complete simulation, their interface allows for such post-processing. For example, McIlveen et al [MMHR16] utilise Block, Brush, and Eraser tools for refinement purposes through their user interface.

7.1. Summary

In summary, this section consists of approaches allowing for editing or refining the produced simulation. Post-processing operations impact several aspects of the simulation, with the most common being path refinement. There have been developed a wide variety of authoring tools for these kinds of operations, from coding to intuitive GUIs, with deformation gestures being among the most popular.

8. Discussion

Simulating crowds has been the focal point of plenty of research, exploring different components and going in-depth for each of them throughout the years. In this survey, we have seen that authoring takes many different forms, uses various metaphors, and focuses on several aspects of the crowd simulation pipeline. However, there exists no *common framework* for authoring the desired, final behaviour of crowds. This is mainly due to: *(1)* the complex pipeline involving multiple components, each with independent ways of being controlled, *(2)* the problem's complexity; there exists a massive set of parameters to be controlled for large crowds, and *(3)* the assumptions as to how crowd simulation is achieved, making it harder to integrate into simulation pipelines. For the future of authoring virtual crowds, we direct the focus towards a more unified framework, built to facilitate all levels simultaneously; all components are interconnected: changes in one affect the others. Perhaps the most prominent section able to influence the rest, is high-level behaviours. As shown in Table 1, high-level behaviour parameters infiltrate other levels. For example, in the work by Durupinar *et al.* [DPA*11], defining behaviours like impatience or leadership (Figure 10) affects how agents select paths and how they interact with other agents locally. This further emphasises the need for a standardised pipeline. It is also seen that intuitiveness of authoring depends on the tool used, from coding which requires expertise, to GUIs and sketch-based interfaces which are more user-friendly; the many natures of the authoring tools indicate the need for an abstract interface.Also note that, in research, the visualisation aspect is commonly affected by changes in high-level parameters, even though industrial tools provide this sort of connection between appearance and behaviour, underlying the lack of effective communication between research and industry.

In this survey, we have discussed the different crowd simulation components individually, and also seen the interaction between them. For each component, there are some potential directions which will yield a more desirable outcome. We claim that for the ***high-level behaviours*** (Section 2) the use of drag-and-drop and map-based interfaces is good practice, avoiding the time-consuming process of tuning system parameters, which usually requires knowledge of a system-specific parameter set. However, there is still room for the development of scripting-free interfaces able to handle more sparse and high-level descriptions of agent

Table 1: *Chart of influential high-level systems and the underlying components they affect. *PP is Path-planning, LM is Local movements, and A is Animation*

Paper	Authoring tool	Affected Components*		
		PP	LM	A
[TCP06]	coding	✓	✓	
[PAB07]	coding	✓	✓	
[FTT99]	scripting	✓	✓	✓
[ST07]	scripting	✓	✓	✓
[YT07]	scripting	✓	✓	✓
[DPA*11]	scripting	✓	✓	
[All10]	GUI	✓	✓	
[NLS14]	GUI	✓	✓	✓
[DGAB15]	GUI	✓	✓	✓
[JCC*15]	sketching	✓	✓	✓

profiles, making them usable by non-experts and expanding the expressiveness of the crowd e.g., finding a mapping between natural language and agent behaviours giving rise to intelligent and emotional agents, exhibiting subtle behaviours. For authoring the ***path-planning*** strategy (Section 3), so far, densities and environment are the main factors considered, which are controlled manually. This control is limited and requires effort, and so it would be interesting to explore how to incorporate other factors such as mental state, and provide more easy-to-use interfaces. This will lead to more intelligent, diverse crowds, easily controlled by the authors. Regarding ***local movements*** (Section 4), mental state parameters have been incorporated in some systems as an additional layer to further shrink the user-defined parameter set, and give clarity as to how parameter manipulation affects the end-product. To this end, past work usually allows user-control manually, while the industry resorts in automated methods (i.e. limited or no control). Psychological status has been included in some local movement algorithms, since these are easily perceived, but it could also be incorporated in path-finding and even in animations. For local interactions, involving agents being influenced by their immediate neighbours and adapting to the environment, users assume control of the implementation by programming, and less often via GUIs (for environment changes). Regarding the addition of motion variety in ***animation*** (Section 5), which is usually obtained by having motion templates or directly influencing models' parameters, authoring is normally achieved again via drag-and-drop or programming. To further facilitate creativity and expressiveness when authoring, it would be highly beneficial to create interfaces with intuitive parameters, and sliders to define those parameters in addition to the gesture-based interfaces currently employed. Additionally, as discussed in Section 6, the ***visualisation*** component can be authored in terms of controlling templates, reflecting differences in: skeleton and body mesh; textures and colours; and accessories. A common practice is the employment of graphical interfaces allowing for selection of template variations and their distribution among the crowd. Lastly, several methods have been developed for ***post-processing*** operations, which might affect a single or multiple simulation aspects. We have seen that some systems exclusively focus on post-processing, whereas others integrate this aspect into their framework. Authoring this is dealt with a number of tools, often involving intuitive GUIs. In the future, it would be useful to have tools that allow specifying the details of selected aspects of some agents, while preserving the global aspect of the simulation.

Apart from potential improvements in authoring the individual components, we make some more general suggestions applicable to multiple components. We claim that *data-driven* methods still have unexplored potential in this field. For example, having agent profiles as inputs to data-driven algorithms to drive the simulation on top of the data. Currently, making changes implies the need for gathering data specific to the intended behaviours or tasks, which is hard to replicate. Blending behaviours as well as assuming control over the blending process and criteria will allow for authoring specified behaviours seamlessly. Additionally, even though complex simulations emerge from the interaction of many individual agents, it is difficult to have control at all levels. Defining high-level behaviour or appearance sometimes implies losing detail of the movement of a specific agent. Therefore, allowing to choose the level of control relevant to the author's wishes is essential, and hence we suggest that the main direction for future research is *allowing for different granularity of control*. Last but not least, improving the *transparency of the authoring parameters* (i.e. being clear how changes in parameters affect the result) is also important. A primary example is the mental state or personality attributes of agents. We have seen how these can authored in terms of affecting the local movements, mainly avoidance. Even though few attempts have been currently made to incorporate mental state parameters into the planning strategy ([PAB07, GKLM11]), due to the conceptual difficulties of mapping these inputs to easily perceived behaviours, finding a way to integrate them into both path-planning and animation will highly benefit the authoring process.

Overall, we find the urgent need for a clear pipeline and evaluation of these systems. The overlapping and interconnecting nature of the aforementioned levels fuels the need for such standardised framework. This is more prominent when considering high-level behaviours and how they impact multiple components, as well as influence actions both on an agent-based and crowd-based level. So far, the several types of methods developed (e.g., scripting or visual interfaces) to control certain parameters do not support the clear distinction of which and how each level responds to such control. A means to mitigate that would be creating abstract interfaces capable of linking the different levels. Preliminary examples could be UMANS in [vTGG*20], where everybody has an action which is essentially a velocity, even Menge by [CBM16], which is however mostly for navigation purposes. On a related note, an attempt to formalise the standard animation pipeline is welcome. Some have attempted to provide explanations of this e.g., Van Toll and Pettré [vTP21]; high-level navigation waypoints and local interactions distinction. Of course, attempting any of this is a challenging undertaking, even more so when it comes to the evaluation of these proposition.

As an end note, we do emphasise the complexity of authoring virtual crowds. Since it comprises a tool to aid creativity and ease the process of generating crowd motion in various ways, it has a lot of artistic dimensions. Making this observation, we conclude that there is a lack of a proper evaluation framework. Our suggestion for a starting point regarding validating these propositions is communication between research and industry.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 860768 (CLIPE project).

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578 and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

References

[ACC14] ALLAIN P., COURTY N., CORPETTI T.: Optimal crowd editing. *Graph. Model. 76* (2014), 1–16.

[Ado] ADOBE: Mixamo. https://www.mixamo.com. Accessed: 2021-10-27.

[AGR*16] ALAHI A., GOEL K., RAMANATHAN V., ROBICQUET A., FEI-FEI L., SAVARESE S.: Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 961–971. doi:10.1109/CVPR.2016.110.

[AHH*09] ASCHWANDEN G., HAEGLER S., HALATSCH J., JEKER R., SCHMITT G., VAN GOOL L.: Evaluation of 3D city models using automatic placed urban agents. In *9th International Conference on Construction Applications of Virtual Reality* (2009), pp. 165–176.

[All10] ALLBECK J.: CAROSA: A Tool for Authoring NPCs. In *International Conference on Motion in Games* (Nov. 2010), pp. 182–193. doi:10.1007/978-3-642-16958-8_18.

[APKM15] ALLEN T., PARVANOV A., KNIGHT S., MADDOCK S.: Using Sketching to Control Heterogeneous Groups. In *Computer Graphics and Visual Computing (CGVC)* (2015), Borgo R., Turkay C., (Eds.), The Eurographics Association. doi:10.2312/cgvc.20151249.

[Auta] AUTODESK: AUTODESK 3DS Max. https://www.autodesk.com/products/3ds-max/overview?term=1-YEAR&tab=subscription. Accessed: 2021-10-27.

[Autb] AUTODESK: AUTODESK Maya. https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription. Accessed: 2021-10-27.

[Autc] AUTODESK: AUTODESK Motion Builder. https://www.autodesk.com/products/motionbuilder/overview?term=1-YEAR&tab=subscription. Accessed: 2021-10-27.

[Bar14] BARNETT A.: Topology based global crowd control. In *CISA 2013 Proceedings for the 26th International Conference on Computer Animation and Social Agents* (2014).

[BBA*98] BADLER N. I., BINDIGANAVALE R., ALLBECK J. M., SCHULER W., ZHAO L., PALMER M.: A Parameterized Action Representation for Virtual Human Agents. In *EMBODIED CONVERSATIONAL AGENTS* (1998).

[BD17] BULBUL A., DAHYOT R.: Populating virtual cities using social media. *Computer Animation and Virtual Worlds 28*, 5 (2017), e1742.

[BKHF15] BERSETH G., KAPADIA M., HAWORTH B., FALOUTSOS P.: SteerFit: Automated Parameter Fitting for Steering Algorithms. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2015), SCA '14, Eurographics Association, pp. 113–122.

[BKSB15] BOATRIGHT C. D., KAPADIA M., SHAPIRA J. M., BADLER N. I.: Generating a multiplicity of policies for agent steering in crowd simulation. *Computer Animation and Virtual Worlds 26* (2015), 483–494.

[BNCM14] BEST A., NARANG S., CURTIS S., MANOCHA D.: DenseSense: Interactive Crowd Simulation using Density-Dependent Filters. In *Symposium on Computer Animation* (2014), Citeseer, pp. 97–102.

[BP15] BRUNEAU J., PETTRÉ J.: Energy-Efficient Mid-Term Strategies for Collision Avoidance in Crowd Simulation. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (New York, NY, USA, 2015), SCA '15, Association for Computing Machinery, pp. 119–127. doi:10.1145/2786784.2786804.

[BPKB15] BEACCO A., PELECHANO N., KAPADIA M., BADLER N. I.: Footstep parameterized motion blending using barycentric coordinates. *Comput. Graph. 47* (2015), 105–112.

[BSK15] BARNETT A., SHUM H., KOMURA T.: Coordinated Crowd Simulation With Topological Scene Analysis. *Computer Graphics Forum 35* (Oct. 2015). doi:10.1111/cgf.12735.

[CBM16] CURTIS S., BEST A., MANOCHA D.: Menge: A Modular Framework for Simulating Crowd Movement. *Collective Dynamics 1*, 0 (2016), 1–40. URL: https://collective-dynamics.eu/index.php/cod/article/view/A1, doi:10.17815/CD.2016.1.

[CC14] CHARALAMBOUS P., CHRYSANTHOU Y.: The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation. *Computer Graphics Forum 33* (2014).

[Che04] CHENNEY S.: Flow Tiles. In *Symposium on Computer Animation* (2004), Boulic R., Pai D. K., (Eds.), The Eurographics Association. doi:10.2312/SCA/SCA04/233-242.

[CJM*18] CHEN L., JUNG C. R., MUSSE S. R., MONEIMNE M., WANG C., FRUCHTER R., BAZJANAC V., CHEN G., BADLER N. I.: Crowd Simulation Incorporating Thermal Environments and Responsive Behaviors. *Presence 26*, 4 (2018), 436–452. doi:10.1162/PRES_a_00308.

[CKGC14] CHARALAMBOUS P., KARAMOUZAS I., GUY S., CHRYSANTHOU Y.: A Data-Driven Framework for Visual Crowd Analysis. *Computer Graphics Forum 33* (Oct. 2014). doi:10.1111/cgf.12472.

[Cla16] CLAVET S.: Motion matching and the road to next-gen animation. In *Proc. of GDC* (2016).

[CP15] CASADIEGO L., PELECHANO N.: *From one to many: Simulating groups of agents with reinforcement learning controllers*. International Conference on Intelligent Virtual Agents, 2015, pp. 119–123. URL: http://hdl.handle.net/2117/78495, doi:10.1007/978-3-319-21996-712.

[CvTZ*22] COLAS A., VAN TOLL W., ZIBREK K., HOYET L., OLIVIER A.-H., PETTRÉ J.: Interaction Fields: Intuitive Sketch-based Steering Behaviors for Crowd Simulation. In *The Eurographics Association* (2022).

[DGAB15] DURUPINAR F., GÜDÜKBAY U., AMAN A., BADLER N. I.: Psychological parameters for crowd simulation: From audiences to mobs. *IEEE transactions on visualization and computer graphics 22*, 9 (2015), 2145–2159.

[dHCSM*05] DE HERAS CIECHOMSKI P., SCHERTENLEIB S., MAÏM J., MAUPU D., THALMANN D.: Real-time Shader Rendering for Crowds in Virtual Heritage. In *VAST* (2005), vol. 5, pp. 1–8.

[dHCSMT05] DE HERAS CIECHOMSKI P., SCHERTENLEIB S., MAÏM J., THALMANN D.: Reviving the Roman Odeon of Aphrodisias: Dynamic Animation and Variety Control of Crowds in Virtual Heritage. *VSMM* (Jan. 2005).

[DHOO05] DOBBYN S., HAMILL J., O'CONOR K., O'SULLIVAN C.: Geopostors: a real-time geometry/impostor crowd rendering system. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games* (2005), pp. 95–102.

[DMCN*17] DUTRA T. B., MARQUES R., CAVALCANTE-NETO J. B., VIDAL C. A., PETTRÉ J.: Gradient-Based Steering for Vision-Based Crowd Simulation Algorithms. *Comput. Graph. Forum 36*, 2 (May 2017), 337–348.

[DMH*21] DANIEL B. C., MARQUES R., HOYET L., PETTRÉ J., BLAT J.: A Perceptually-Validated Metric for Crowd Trajectory Quality Evaluation. *Proc. ACM Comput. Graph. Interact. Tech. 4*, 3 (Sept. 2021). doi:10.1145/3480136.

[DPA*11] DURUPINAR F., PELECHANO N., ALLBECK J., GÜDÜKBAY U., BADLER N. I.: How the Ocean Personality Model Affects the Perception of Crowds. *IEEE Computer Graphics and Applications 31*, 3 (2011), 22–31. doi:10.1109/MCG.2009.105.

[dPVM05] DE PAIVA D. C., VIEIRA R., MUSSE S. R.: Ontology-based crowd simulation for normal life situations. In *International 2005 Computer Graphics* (2005), IEEE, pp. 221–226.

[Enga] ENGINE U.: Behavior Trees. https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTrees/. Accessed: 2021-10-27.

[Engb] ENGINE U.: Unreal Documentation Blend Spaces. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/Blendspaces/. Accessed: 2021-10-28.

[Engc] ENGINE U.: Unreal Engine Navigation Components. https://docs.unrealengine.com/4.27/en-US/Basics/Components/Navigation/. Accessed: 2021-10-27.

[FTT99] FUNGE J. D., TU X., TERZOPOULOS D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99* (1999).

[GCC*10] GUY S. J., CHHUGANI J., CURTIS S., DUBEY P. K., LIN M. C., MANOCHA D.: PLEdestrians: a least-effort approach to crowd simulation. In *SCA '10* (2010).

[GD10] GU Q., DENG Z.: Context-aware motion diversification for crowd simulation. *IEEE Computer Graphics and Applications 31*, 5 (2010), 54–65.

[GD11] GU Q., DENG Z.: Formation sketching: an approach to stylize groups in crowd simulation. In *Graphics Interface* (2011).

[GD13] GU Q., DENG Z.: Generating Freestyle Group Formations in Agent-Based Crowd Simulations. *IEEE Computer Graphics and Applications 33*, 01 (Jan. 2013), 20–31. doi:10.1109/MCG.2011.87.

[GJF*18] GUPTA A., JOHNSON J., FEI-FEI L., SAVARESE S., ALAHI A.: Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. *CoRR abs/1803.10892* (2018). URL: http://arxiv.org/abs/1803.10892.

[GKLM11] GUY S., KIM S., LIN M., MANOCHA D.: Simulating Heterogeneous Crowd BehaviorsUsing Personality Trait Theory. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2011)* (Aug. 2011), pp. 43–52. doi:10.1145/2019406.2019413.

[GLD08] GALVAO R., LAYCOCK R. G., DAY A. M.: Gpu techniques for creating visually diverse crowds in real-time. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology* (2008), pp. 79–86.

[GM17] GONZALEZ L. R. M., MADDOCK S. C.: Sketching for Real-time Control of Crowd Simulations. In *CGVC* (2017).

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-Based Inverse Kinematics. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 522–531. doi:10.1145/1015706.1015755.

[Goa] GOALEM: Golaem. https://golaem.com/. Accessed: 2021-10-27.

[GSM05] GOSSELIN D., SANDER P. V., MITCHELL J. L.: Drawing a crowd. *ShaderX3 (CHARLES RIVER MEDIA)* (2005), 505–517.

[GvdBL*12] GUY S. J., VAN DEN BERG J., LIU W., LAU R., LIN M. C., MANOCHA D.: A Statistical Similarity Measure for Aggregate Crowd Dynamics. *ACM Trans. Graph. 31*, 6 (Nov. 2012). doi:10.1145/2366145.2366209.

[Hav] HAVOC: Havoc AI. https://www.havok.com/havok-ai/. Accessed: 2021-10-27.

[HBM*20] HAWORTH M. B., BERSETH G., MOON S., FALOUTSOS P., KAPADIA M.: Deep Integration of Physical Humanoid Control and Crowd Navigation. *Motion, Interaction and Games* (2020).

[HK10] HUANG Y., KALLMANN M.: Motion Parameterization with Inverse Blending. In *Motion in Games* (Berlin, Heidelberg, 2010), Boulic R., Chrysanthou Y., Komura T., (Eds.), Springer Berlin Heidelberg, pp. 242–253.

[HM98] HELBING D., MOLNAR P.: Social Force Model for Pedestrian Dynamics. *Physical Review E 51* (May 1998). doi:10.1103/PhysRevE.51.4282.

[HOD15] HUGHES R., ONDŘEJ J., DINGLIANA J.: DAVIS: Density-Adaptive Synthetic-Vision Based Steering for Virtual Crowds. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games* (New York, NY, USA, 2015), MIG '15, Association for Computing Machinery, pp. 79–84. doi:10.1145/2822013.2822030.

[HOKP16] HOYET L., OLIVIER A.-H., KULPA R., PETTRÉ J.: Perceptual Effect of Shoulder Motions on Crowd Animations. *ACM Trans. Graph. 35*, 4 (July 2016). doi:10.1145/2897824.2925931.

[hPN*16] HE L., PAN J., NARANG S., WANG W., MANOCHA D.: Dynamic Group Behaviors for Interactive Crowd Simulation. *Symposium on Computer Animation* (Feb. 2016).

[HSK14] HENRY J., SHUM H. P. H., KOMURA T.: Interactive Formation Control in Complex Environments. *IEEE Transactions on Visualization and Computer Graphics 20* (2014), 211–222.

[Hum] HUMAN M.: Make Human Community. http://www.makehumancommunity.org/. Accessed: 2021-10-27.

[JCC*15] JORDAO K., CHARALAMBOUS P., CHRISTIE M., PETTRÉ J., CANI M.: Crowd art: density and flow based crowd motion design. *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games* (2015).

[JCP*10] JU E., CHOI M. G., PARK M., LEE J., LEE K. H., TAKAHASHI S.: Morphable Crowds. In *ACM SIGGRAPH Asia 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH ASIA '10, Association for Computing Machinery. doi:10.1145/1866158.1866162.

[JH15] JI Q., HAN Y.: Novel Integrated Framework for Crowd Simulation. In *IWOST-1* (2015).

[JL14] JORGENSEN C., LAMARCHE F.: Space and Time Constrained Task Scheduling for Crowd Simulation. *Collection des Publications Internes de l'Irisa* (Jan. 2014).

[JPCC14] JORDAO K., PETTRÉ J., CHRISTIE M., CANI M.-P.: Crowd sculpting: A space-time sculpting method for populating virtual environments. *Computer Graphics Forum 33* (2014).

[JXW*08] JIN X., XU J., WANG C. C. L., HUANG S., ZHANG J.: Interactive Control of Large-Crowd Navigation in Virtual Environments Using Vector Fields. *IEEE Computer Graphics and Applications 28* (2008).

[KBG*13] KAPADIA M., BEACCO A., GARCIA F. M., REDDY V., PELECHANO N., BADLER N.: Multi-domain real-time planning in dynamic environments. In *SCA '13* (2013).

[KBK16] KRONTIRIS A., BEKRIS K. E., KAPADIA M.: ACUMEN: Activity-centric crowd authoring using influence maps. In *Proceedings of the 29th International Conference on Computer Animation and Social Agents* (2016), pp. 61–69.

[KCPS08] KWON T., CHO Y.-S., PARK S., SHIN S.: Two-Character Motion Analysis and Synthesis. *IEEE transactions on visualization and computer graphics 14* (May 2008), 707–20. doi:10.1109/TVCG.2008.22.

[KFS*16] KAPADIA M., FREY S., SHOULSON A., SUMNER R., GROSS M.: CANVAS: computer-assisted narrative animation synthesis. In *Symposium on Computer Animation* (2016).

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion Graphs. *ACM Trans. Graph. 21*, 3 (July 2002), 473–482. doi:10.1145/566654.566605.

[KHHL12] KIM M., HWANG Y., HYUN K., LEE J.: Tiling motion patches. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), pp. 117–126.

[KHKL09] KIM M., HYUN K., KIM J., LEE J.: Synchronized Multi-Character Motion Editing. *ACM Trans. Graph. 28*, 3 (July 2009). doi:10.1145/1531326.1531385.

[KHvBO09] KARAMOUZAS I., HEIL P., VAN BEEK P., OVERMARS M. H.: A Predictive Collision Avoidance Model for Pedestrian Simulation. In *MIG* (2009).

[KK14] KANG S.-J., KIM S.: Crowd Control with Vector Painting. *J. Res. Pract. Inf. Technol. 46* (2014).

[KKT*14] KRAAYENBRINK N., KESSING J., TUTENEL T., HAAN G. D., BIDARRA R.: Semantic crowds. *Entertain. Comput. 5* (2014), 297–312.

[KLLT08] KWON T.-J., LEE K., LEE J., TAKAHASHI S.: Group motion editing. In *SIGGRAPH 2008* (2008).

[KMT09] KASAP M., MAGNENAT-THALMANN N.: Sizing avatars from skin weights. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* (2009), pp. 123–126.

[KMT10] KASAP M., MAGNENAT-THALMANN N.: Customizing and populating animated digital mannequins for real-time application. In *2010 International Conference on Cyberworlds* (2010), IEEE, pp. 368–374.

[KO10] KARAMOUZAS I., OVERMARS M.: A Velocity-Based Approach for Simulating Human Collision Avoidance. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents* (Berlin, Heidelberg, 2010), IVA'10, Springer-Verlag, pp. 180–186.

[KRR10] KARMAKHARM T., RICHMOND P., ROMANO D. M.: Agent-based Large Scale Simulation of Pedestrians With Adaptive Realistic Navigation Vector Fields. In *TPCG* (2010).

[KSA*09] KAPADIA M., SINGH S., ALLEN B., REINMAN G., FALOUTSOS P.: SteerBug: an interactive framework for specifying and detecting steering behaviors. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)* (Jan. 2009), pp. 209–216. doi:10.1145/1599470.1599497.

[KSG14] KARAMOUZAS I., SKINNER B., GUY S. J.: Universal power law governing pedestrian interactions. *Physical review letters 113 23* (2014), 238701.

[KSHF09] KAPADIA M., SINGH S., HEWLETT W., FALOUTSOS P.: Egocentric affordance fields in pedestrian steering. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics, SI3D 2009, February 27 - March 1, 2009, Boston, Massachusetts, USA* (Jan. 2009), pp. 215–223. doi:10.1145/1507149.1507185.

[KSHG18] KARAMOUZAS I., SOHRE N., HU R., GUY S. J.: Crowd Space: A Predictive Crowd Analysis Technique. *ACM Trans. Graph. 37*, 6 (Dec. 2018). doi:10.1145/3272127.3275079.

[KSKL14] KIM J., SEOL Y., KWON T., LEE J.: Interactive manipulation of large-scale crowd animation. *ACM Transactions on Graphics (TOG) 33* (2014), 1–10.

[KSNG17] KARAMOUZAS I., SOHRE N., NARAIN R., GUY S. J.: Implicit Crowds: Optimization Integrator for Robust Crowd Simulation. *ACM Trans. Graph. 36*, 4 (July 2017). doi:10.1145/3072959.3073705.

[KSRF11] KAPADIA M., SINGH S., REINMAN G., FALOUTSOS P.: A Behavior Authoring Framework for Multi-Actor Simulations. *IEEE Computer Graphics and Applications 31* (Nov. 2011). doi:10.1109/MCG.2011.68.

[KSS*16] KAPADIA M., SHOULSON A., STEIMER C., OBERHOLZER S., SUMNER R. W., GROSS M.: An Event-Centric Approach to Authoring Stories in Crowds. In *Proceedings of the 9th International Conference on Motion in Games* (New York, NY, USA, 2016), MIG '16, Association for Computing Machinery, pp. 15–24. doi:10.1145/2994258.2994265.

[LA11] LI W., ALLBECK J. M.: Populations with purpose. In *International Conference on Motion in Games* (2011), Springer, pp. 132–143.

[LA15] LEMERCIER S., AUBERLET J.-M.: Towards more behaviours in crowd simulation. *Computer Animation and Virtual Worlds 27* (Feb. 2015). doi:10.1002/cav.1629.

[LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: a data-driven approach to crowd simulation. In *SCA '07* (2007).

[LCL06] LEE K. H., CHOI M. G., LEE J.: Motion Patches: Building Blocks for Virtual Environments Annotated with Motion Data. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, Association for Computing Machinery, pp. 898–906. doi:10.1145/1179352.1141972.

[LCM*17] LUO L., CHAI C., MA J., ZHOU S., CAI W.: ProactiveCrowd: Modelling Proactive Steering Behaviours for Agent-Based Crowd Simulation: Modelling Proactive Steering for Agent-Based Crowd Simulation. *Computer Graphics Forum 37* (Sept. 2017). doi:10.1111/cgf.13303.

[LCMP19a] LÓPEZ A., CHAUMETTE F., MARCHAND E., PETTRÉ J.: Attracted by Light: Vision-Based Steering Virtual Characters among Dark and Light Obstacles. In *Motion, Interaction and Games* (New York, NY, USA, 2019), MIG '19, Association for Computing Machinery. doi:10.1145/3359566.3360085.

[LCMP19b] LÓPEZ A., CHAUMETTE F., MARCHAND E., PETTRÉ J.: Character navigation in dynamic environments based on optical flow. *Computer Graphics Forum 38* (May 2019). doi:10.1111/cgf.13629.

[LCR*03] LEE J., CHAI J., REITSMA P., HODGINS J., POLLARD N.: Interactive Control of Avatars Animated with Human Motion Data. *ACM Transactions on Graphics (SIGGRAPH 2002) 21* (July 2003). doi:10.1145/566570.566607.

[LFCCO09] LERNER A., FITUSI E., CHRYSANTHOU Y., COHEN-OR D.: Fitting Behaviors to Pedestrian Simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, Association for Computing Machinery, pp. 199–208. doi:10.1145/1599470.1599496.

[LJK*12] LEMERCIER S., JELIC A., KULPA R., HUA J., FEHRENBACH J., DEGOND P., APPERT-ROLLAND C., DONIKIAN S., PETTRÉ J.: Realistic following behaviors for crowd simulation. *Computer Graphics Forum 31*, 2 (2012), 489–498. URL: https://hal.archives-ouvertes.fr/hal-00817688, doi:10.1111/j.1467-8659.2012.03028.x.

[LMK09] LACROIX B., MATHIEU P., KEMENY A.: Generating various and consistent behaviors in simulations. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)* (2009), Springer, pp. 110–119.

[LS00] LEE J., SHIN S.: A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. *Computer Graphics (Proceedings of SIGGRAPH'99) 33* (Aug. 2000). doi:10.1145/311535.311539.

[MAS] MASSIVE: Massive Software. https://www.massivesoftware.com/. Accessed: 2021-10-27.

[MBA20] MATHEW C. D. T., BENES B., ALIAGA D. G.: An output-driven approach to design a swarming model for architectural indoor environments. *Comput. Graph. 87* (2020), 103–110.

[MCT21] MUSSE S. R., CASSOL V. J., THALMANN D.: A history of crowd simulation: the past, evolution, and new perspectives. *The Visual Computer* (2021), 1–16.

[MH03] METOYER R. A., HODGINS J. K.: Reactive pedestrian path following from examples. In *Proceedings 11th IEEE International Workshop on Program Comprehension* (2003), pp. 149–156. doi:10.1109/CASA.2003.1199318.

[MKMA19] MATHEW C. D. T., KNOB P., MUSSE S. R., ALIAGA D. G.: Urban Walkability Design Using Virtual Population Simulation. *Computer Graphics Forum 38* (2019), 455–469.

[MLC10] MIN J., LIU H., CHAI J.: Synthesis and Editing of Personalized Stylistic Human Motion. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, Association for Computing Machinery, pp. 39–46. doi:10.1145/1730804.1730811.

[MLD*08] MCDONNELL R., LARKIN M., DOBBYN S., COLLINS S., O'SULLIVAN C.: Clone Attack! Perception of Crowd Variety. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 1–8. doi:10.1145/1360612.1360625.

[MMHR16] MCILVEEN J., MADDOCK S. C., HEYWOOD P., RICHMOND P.: PED: Pedestrian Environment Designer. In *CGVC* (2016).

[MPG*10] MOUSSAÏD M., PEROZO N., GARNIER S., HELBING D., THERAULAZ G.: The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. *PloS one 5* (Apr. 2010), e10047. doi:10.1371/journal.pone.0010047.

[MT01] MUSSE S. R., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (Apr. 2001), 152–164. doi:10.1109/2945.928167.

[MXH*10] MA W., XIA S., HODGINS J. K., YANG X., LI C., WANG Z.: Modeling Style and Variation in Human Motion. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2010), SCA '10, Eurographics Association, pp. 21–30.

[MYMT07] MORINI F., YERSIN B., MAÏM J., THALMANN D.: Real-Time Scalable Motion Planning for Crowds. *2007 International Conference on Cyberworlds (CW'07)* (2007), 144–151.

[MYT09] MAÏM J., YERSIN B., THALMANN D.: Unique Character Instances for Crowds. *IEEE Computer Graphics and Applications 29* (2009), 82–90.

[MYTP09] MAÏM J., YERSIN B., THALMANN D., PETTRÉ J.: YaQ: An Architecture for Real-Time Navigation and Rendering of Varied Crowds. *IEEE Comput. Graph. Appl. 29*, 4 (July 2009), 44–53.

[NKF12] NISWAR A., KHAN I. R., FARBIZ F.: Avatar customization based on human body measurements. In *SIGGRAPH Asia 2012 Posters*. SIGGRAPH Asia 2012, 2012, pp. 1–1.

[NKS*14] NINOMIYA K., KAPADIA M., SHOULSON A., GARCIA F., BADLER N.: Planning approaches to constraint-aware navigation in dynamic environments: Constraint-aware navigation in dynamic environments. *Computer Animation and Virtual Worlds 26* (Oct. 2014). doi:10.1002/cav.1622.

[NLS14] NORMOYLE A., LIKHACHEV M., SAFONOVA A.: Stochastic activity authoring with direct user control. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2014), pp. 31–38.

[OO09] OSHITA M., OGIWARA Y.: Sketch-Based Interface for Crowd Animation. In *Smart Graphics* (Berlin, Heidelberg, 2009), Butz A., Fisher B., Christie M., Krüger A., Olivier P., Therón R., (Eds.), Springer Berlin Heidelberg, pp. 253–262.

[OPOD10] ONDŘEJ J., PETTRÉ J., OLIVIER A.-H., DONIKIAN S.: A Synthetic-Vision Based Steering Approach for Crowd Simulation. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH '10, Association for Computing Machinery. doi:10.1145/1833349.1778860.

[Ops] OPSIVE: Behavior Designer. https://opsive.com/solutions/ai-solution/. Accessed: 2021-10-27.

[PAB07] PELECHANO N., ALLBECK J., BADLER N.: Controlling Individual Agents in High-Density Crowd Simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2007, San Diego, California, USA, August 2-4, 2007* (Aug. 2007), vol. 2007, pp. 99–108. doi:10.1145/1272690.1272705.

[Par] PARADOXNOTION: nodeCanvas. https://nodecanvas.paradoxnotion.com/. Accessed: 2021-10-27.

[Par10] PARK M. J.: Guiding Flows for Controlling Crowds. *Vis. Comput. 26*, 11 (Nov. 2010), 1383–1391. doi:10.1007/s00371-009-0415-4.

[PBC*11] PATIL S., BERG J. P. V. D., CURTIS S., LIN M. C., MANOCHA D.: Directing Crowd Simulations Using Navigation Fields. *IEEE Transactions on Visualization and Computer Graphics 17* (2011), 244–254.

[PDB06] PARIS S., DONIKIAN S., BONVALET N.: Environmental abstraction and path planning techniques for realistic crowd simulation. *Computer Animation and Virtual Worlds 17* (2006).

[PKL08] PETTRÉ J., KALLMANN M., LIN M. C.: Motion Planning and Autonomy for Virtual Humans. In *ACM SIGGRAPH 2008 Classes* (New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. doi:10.1145/1401132.1401193.

[PO11] PRAZAK M., O'SULLIVAN C.: Perceiving human motion variety. In *Proceedings - APGV 2011: ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization* (Aug. 2011), pp. 87–92. doi:10.1145/2077451.2077468.

[POSB05] PELECHANO N., O'BRIEN K., SILVERMAN B., BADLER N.: Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation* (2005).

[PPD07] PARIS S., PETTRÉ J., DONIKIAN S.: Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. *Computer Graphics Forum 26* (2007).

[PRY13] PARK J. H., ROJAS F. A., YANG H. S.: A collision avoidance behavior model for crowd simulation based on psychological findings. *Computer Animation and Virtual Worlds 24*, 3-4 (2013), 173–183. doi:https://doi.org/10.1002/cav.1504.

[QH10] QIU F., HU X.: Modeling group structures in pedestrian crowd simulation. *Simulation Modelling Practice and Theory 18*, 2 (2010), 190–205. doi:https://doi.org/10.1016/j.simpat.2009.10.005.

[RCB*17] REN Z., CHARALAMBOUS P., BRUNEAU J., PENG Q., PETTRÉ J.: Group Modeling: A Unified Velocity-Based Approach. *Computer Graphics Forum 36*, 8 (Dec. 2017), 45–56. URL: https://hal.inria.fr/hal-01372766, doi:10.1111/cgf.12993.

[Reaa] REALLUSION: Character Creator. https://www.reallusion.com/character-creator/. Accessed: 2021-10-29.

[Reab] REALLUSION: iClone 7: Real-time 3D Animation Software. https://www.reallusion.com/iclone/. Accessed: 2021-10-27.

[Res] RESEARCH D.: Authoring Motion Cycles. https://la.disneyresearch.com/publication/authoring-motion-cycles/. Accessed: 2021-10-29.

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87* (1987).

[Rey99] REYNOLDS C. W.: Steering Behaviors For Autonomous Characters. In *Game developers conference, 1999* (1999).

[RGBC04] ROSE C., GUENTER B., BODENHEIMER B., COHEN M.: Efficient Generation of Motion Transitions using Spacetime Constraints. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics 30* (Apr. 2004). doi:10.1145/237170.237229.

[RISC01] ROSE III C., SLOAN P.-P., COHEN M.: Artist-Directed Inverse-Kinematics Using Radial Basis Function Interpolation. *Comput. Graph. Forum 20* (Sept. 2001). doi:10.1111/1467-8659.00516.

[RPP21] ROGLA O., PELECHANO N., PATOW G. A.: Procedural crowd generation for semantically augmented virtual cities. *Computers & Graphics 99* (2021), 83–99.

[RXX*21] REN J., XIANG W., XIAO Y., YANG R., MANOCHA D., JIN

X.: Heter-Sim: Heterogeneous Multi-Agent Systems Simulation by Interactive Data-Driven Optimization. *IEEE Transactions on Visualization and Computer Graphics 27* (2021), 1953–1966.

[SBT10] SOGA A., BOULIC R., THALMANN D.: Motion Planning and Animation Variety using Dance Motion Clips. In *2010 International Conference on Cyberworlds* (2010), IEEE, pp. 421–424.

[SdSdSH19] SILVA G. C., DA SILVA A. T., DA SILVA HOUNSELL M.: Crowd generation using morphological obesity criteria. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2019), IEEE, pp. 81–90.

[SGC04] SUNG M., GLEICHER M., CHENNEY S.: Scalable behaviors for crowd simulation. *Computer Graphics Forum 23* (2004).

[SKFR09] SINGH S., KAPADIA M., FALOUTSOS P., REINMAN G. D.: SteerBench: a benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds 20* (2009).

[SKSY08] SHUM H. P. H., KOMURA T., SHIRAISHI M., YAMAZAKI S.: Interaction patches for multi-character animation. *ACM Trans. Graph. 27* (2008), 114.

[SKY07] SHUM H. P. H., KOMURA T., YAMAZAKI S.: Simulating competitive interactions using singly captured motions. In *VRST '07* (2007).

[SL15] SINCLAIR J., LUI C. S. M.: Integrating personality and emotion for human crowd simulation. In *Fifteenth International Conference on Electronic Business* (2015).

[SOWO17] SHI Y., ONDŘEJ J., WANG H., O'SULLIVAN C.: Shape Up! Perception based body shape variation for data-driven crowds. In *2017 IEEE Virtual Humans and Crowds for Immersive Environments (VHCIE)* (2017), IEEE, pp. 1–7.

[ST07] SHAO W., TERZOPOULOS D.: Autonomous Pedestrians. *Graphical Models 69* (Sept. 2007), 246–274. doi:10.1016/j.gmod.2007.09.001.

[SZZK21] STARKE S., ZHAO Y., ZINNO F., KOMURA T.: Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG) 40* (2021), 1–16.

[TCP06] TREUILLE A., COOPER S., POPOVIC Z.: Continuum crowds. *ACM Trans. Graph. 25* (2006), 1160–1168.

[TLC02] TECCHIA F., LOSCOS C., CHRYSANTHOU Y.: Image-based crowd rendering. *IEEE computer graphics and applications 22*, 2 (2002), 36–43.

[TYK*09] TAKAHASHI S., YOSHIDA K., KWON T., LEE K. H., LEE J., SHIN S. Y.: Spectral-Based Group Formation Control. *Computer Graphics Forum* (2009). doi:10.1111/j.1467-8659.2009.01404.x.

[Ubi] UBISOFT: PLAY-AS-ANYONE: THE AI OF WATCH DOGS LEGION. https://toronto.ubisoft.com/watch-dog-legions-play-as-anyone/. Accessed: 2021-10-27.

[UCT04] ULICNY B., CIECHOMSKI P. D. H., THALMANN D.: Crowdbrush: interactive authoring of real-time crowd scenes. In *SCA '04* (2004).

[Unia] UNITY: Unity Documentation Blend Trees. https://docs.unity3d.com/Manual/class-BlendTree.html. Accessed: 2021-10-27.

[Unib] UNITY: Unity Documentation Mecanim Animation System. https://docs.unity3d.com/460/Documentation/Manual/MecanimAnimationSystem.html. Accessed: 2021-10-27.

[Unic] UNITY: Unity Documentayion Navigation and Pathfinding. https://docs.unity3d.com/Manual/Navigation.html. Accessed: 2021-10-27.

[vdBGLM11] VAN DEN BERG J., GUY S., LIN M., MANOCHA D.: *Reciprocal n-Body Collision Avoidance*, vol. 70. Robotics Research, Apr. 2011, pp. 3–19. doi:10.1007/978-3-642-19457-3_1.

[vdBLM08] VAN DEN BERG J., LIN M., MANOCHA D.: Reciprocal Velocity Obstacles for Real-Time Multi-agent Navigation. In *ICRA* (May 2008), pp. 1928–1935. doi:10.1109/ROBOT.2008.4543489.

[vGJCIG15] VAN GOETHEM A., JAKLIN N., COOK IV A., GERAERTS R.: On Streams and Incentives: A Synthesis of Individual and Collective Crowd Motion. In *Computer Animation and Social Agents (CASA) 2015* (May 2015).

[vTGG*20] VAN TOLL W., GRZESKOWIAK F., GANDÍA A., AMIRIAN J., BERTON F., BRUNEAU J., CABRERO DANIEL B., JOVANE A., PETTRÉ J.: Generalized Microscropic Crowd Simulation using Costs in Velocity Space. In *Symposium on Interactive 3D Graphics and Games* (May 2020), pp. 1–9. doi:10.1145/3384382.3384532.

[vTP19] VAN TOLL W., PETTRÉ J.: Connecting global and local agent navigation via topology. In *Motion, Interaction and Games*. Association for Computing Machinery, 2019, pp. 1–10.

[vTP21] VAN TOLL W., PETTRÉ J.: Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s. *Computer Graphics Forum 40*, 2 (2021), 731–754. doi:https://doi.org/10.1111/cgf.142664.

[War06] WARREN W. H.: The dynamics of perception and action. *Psychological review 113 2* (2006), 358–89.

[WGO*14] WOLINSKI D., GUY S., OLIVIER A.-H., LIN M., MANOCHA D., PETTRÉ J.: Parameter Estimation and Comparative Evaluation of Crowd Simulations. *Computer Graphics Forum 33* (May 2014). doi:10.1111/cgf.12328.

[XWY*14] XU M., WU Y., YE Y., FARKAS I., JIANG H.: Collective Crowd Formation Transform with Mutual Information-Based Runtime Feedback. *Computer Graphics Forum 34* (Aug. 2014). doi:10.1111/cgf.12459.

[YMC*05] YERSIN B., MAÏM J., CIECHOMSKI P. D. H., SCHERTENLEIB S., THALMANN D.: Steering a virtual crowd based on a semantically augmented navigation graph. In *In Proceedings of the First International Workshop on Crowd Simulation* (2005), pp. 169–178.

[YMPT09] YERSIN B., MAÏM J., PETTRÉ J., THALMANN D.: Crowd Patches: Populating Large-Scale Virtual Environments for Real-Time Applications. *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (Feb. 2009). doi:10.1145/1507149.1507184.

[YT07] YU Q., TERZOPOULOS D.: A decision network framework for the behavioral animation of virtual humans. In *SCA '07* (2007).

[ZIK11] ZANLUNGO F., IKEDA T., KANDA T.: Social force model with explicit collision prediction. *EPL 93* (2011), 68005.

[ZSH*19] ZHANG X., SCHAUMANN D., HAWORTH B., FALOUTSOS P., KAPADIA M.: Coupling agent motivations and spatial behaviors for authoring multiagent narratives. *Computer Animation and Virtual Worlds 30*, 3-4 (2019), e1898.

[ZTC13] ZHAO M., TURNER S. J., CAI W.: A Data-Driven Crowd Simulation Model Based on Clustering and Classification. *2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications* (2013), 125–134.