Immersive Sketching to Author Crowd Movements in Real-time

Andrea Bönsch*
Visual Computing Institute,
RWTH Aachen University, Germany

Sebastian J. Barton
Department of Computer Science,
RWTH Aachen University, Germany

Jonathan Ehret
Visual Computing Institute,
RWTH Aachen University, Germany

Torsten W. Kuhlen
Visual Computing Institute,
RWTH Aachen University, Germany

Figure 1: Guiding a virtual crowd via sketch-based *barriers* and connected *sections* in virtual reality.

ABSTRACT

Sketch-based interfaces in 2D screen space allow to efficiently author the flow of virtual crowds in a direct and interactive manner. Here, options to redirect a flow by sketching barriers, or guiding entities based on a sketched network of connected sections are provided.

As virtual crowds are increasingly often embedded into virtual reality (VR) applications, 3D authoring is of interest. In this preliminary work, we thus present a sketch-based approach for VR. First promising results of a proof-of-concept are summarized and improvement suggestions, extensions, and future steps are discussed.

CCS CONCEPTS

• **Human-centered computing** → **Virtual Reality**; **Social navigation**.

KEYWORDS

sketch-based interface, authoring, virtual crowds, virtual reality

ACM Reference Format:
Andrea Bönsch, Sebastian J. Barton, Jonathan Ehret, and Torsten W. Kuhlen. 2020. Immersive Sketching to Author Crowd Movements in Real-time. In *IVA '20: Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents (IVA '20), October 19–23, 2020, Virtual Event, Scotland Uk*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3383652.3423883

*boensch@vr.rwth-aachen.de

IVA '20, October 19–23, 2020, Virtual Event, Scotland Uk

1 INTRODUCTION

Crowd simulations are defined as autonomous navigation of a multitude of virtual entities. Modeling realistic pedestrian crowds is thereby of particular interest. Here, a variety of customizable parameters is typically given, allowing experts to define the pedestrians' behavior in the crowd. While various crowd simulations and support tools for behavior fine-tuning exist (e.g., [1, 10]), authoring the resulting crowds is relatively little researched.

Authoring allows users to direct the pedestrians in a given environment. Influence maps can be used to drive a single entity, combining the entity's desire with predefined attraction areas in a scene [5]. To guide a pedestrian flow, *sketching* is a preferable technique [2], allowing a direct and often interactive crowd control for experts and novice users alike. Based on intuitive sketches, areas of attraction and avoidance [6], example trajectories [9], or networks through specific locations [2, 11] are added to a scene, directing the pedestrians.

To the best of our knowledge, the sketching-based approaches are currently limited to a 2D screen space. However, one growing field of application for virtual crowds is virtual reality (VR). Here, crowds, e.g., enliven the environments allowing users a more intense and realistic experience. To this end, we pursue two objectives: (*O*1) We aim to investigate whether authoring crowds in VR results in faster guiding processes or more plausible pedestrian behavior compared to the 2D approaches. (*O*2) We are interested in evaluating whether real-time authoring is a useful tool during scene exploration. In case of, e.g., a careful inspection of a certain location, using temporary avoidance areas to prevent pedestrians from entering, or rerouting them gives users space while the scene remains lively.

As work-in-progress, we thus present a first step towards our objectives: We contribute an immersive sketch-based interface (Sec. 2), a feasibility check as proof-of-concept (Sec. 3), and a discussion of the insights gained and future steps to take (Sec. 4).

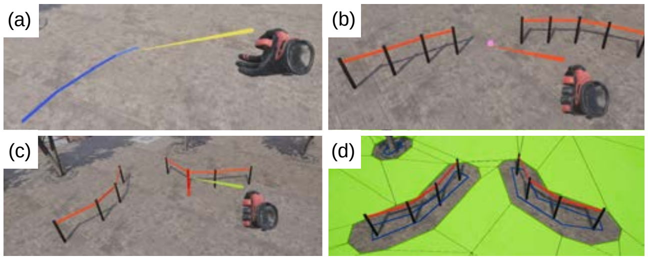

Figure 2: (a) Barrier sketch operation with preview; (b) Deletion and (c) translation of interim poles; (d) NavMesh update.

2 VR-BASED CROWD AUTHORING

Shifting the authoring to VR allows users to sketch with a hand-held controller directly in the 3D environment without any constraints to the 2D world. Thus, we require an intuitive, interactive, and immersive user interface for directing the pedestrians, while the flow adaptions have to be executed in real-time. Considering *O*2, crucial sketches for understanding the flow should furthermore be blended into the scene. Lastly, the 3rd person sketching perspective can be easily enhanced by an additional 1st person perspective ([3]): Possessing a random pedestrian allows users to experience the scenario in life-size (Fig. 1), supporting the authoring of realistic walking scenarios.

A recent, desktop-based approach of Gonzalez et al. detailing *barriers* and *sections* for the *flow control* [2] serves as our model, allowing a fair comparison between VR-based and 2D sketching (cp. *O*1) in a future step. Furthermore, two crowd algorithms ([4], [7]) and the Unreal Engine 4[1] (UE) provide the means for our approach. The utilized UE's navigation system provides an automatic navMesh generation, allowing us to obtain an accurate representations of the walkable spaces, while real-time, asynchronous navMesh updates are supported.

Barriers, shown in Figure 2, are defined as invious obstacles blocking the entities' paths. To embed them, users sketch a line, which is directly projected onto the *navMesh*. During the sketching operation, a real-time preview of the line is shown as visual feedback. As barriers often close off free areas, their existence and positions are crucial to understand a pedestrian flow. To intuitively blend in the boundaries of the walkable space, the preview is replaced by actual scene elements in form of barrier tapes once the sketch is finished.

After embedding a *barrier*, it can be altered, e.g., by deleting poles or by relocating them. On all operations, the navMesh and thus the walkable space is automatically updated by UE's navigation system.

Sections are defined as waypoints on the pedestrians' paths. To embed them, users sketch a convex area, which is projected onto the scene's ground. During the sketching operation, a real-time preview of the area is shown as visible feedback. Once finished, the preview is kept and the *section* is configured either (a) as entry allowing pedestrians to spawn periodically during runtime (Fig. 4), (b) as exit allowing pedestrians, who enter the *section*, to vanish (Fig. 4), or (c) as interstation. The latter serves as interim goal for the pedestrians' movement, before sending them to their next location either based on a *section*-hierarchy defined during the succeeding *flow control* configuration or to a random position in the scene. Additional parameters allow to define the group size and spawn rate for entries, the time pedestrians

[1] https://www.unrealengine.com/en-US/; last-visited: 2020-09-13

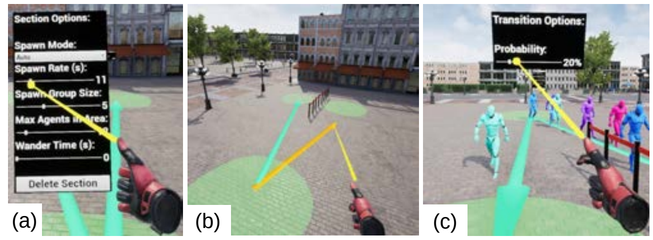

Figure 3: (a) Configuring a *section*. (b) Sketching a transition btw. two *sections*. (c) Weighting it (end *section's* perspective).

wander in a *section*, and the amount of pedestrians simultaneously allowed in a *section*, to control the flow density (Fig. 3(a)).

Instead of embedding the *sections* into the navMesh, random points are sampled in a section and projected onto the navMesh to generate specific waypoints on a pedestrians' paths. By this, we avoid an excessive encroachment into the UE's navigation system. Another positive side-effect relates to the future work: While in our current approach *sections* cannot touch each other, overlapping *sections* might be of interest for versatile, however authored, movements. Here, configuring different walking speeds might be also of interest, e.g., reducing speeds in a park, or increasing them while crossing a street.

Flow Control is done by interconnecting the *sections* with transitions. Therefore, users can sketch a line between two *sections*, defining a flow from the line's start to its end. During the sketching operation, a line preview is shown as visible feedback (Fig 3(b)). Once finished, the line is replaced by an arrow, indicating the flow direction. This way, entries and exits can be connected either directly (Fig. 4) or via interstations. If vanishing pedestrians are unwanted, interstations can be chosen as last *section* in the *flow control* hierarchy.

In case several transitions start in one *section*, users can also weight the individual transitions to further guide the flow. Therefore, a percentage is assigned to each transition indicating to which probability the respective path is chosen by a pedestrian (Fig. 3(c)).

3 PROOF-OF-CONCEPT

We conducted a first feasibility check as a proof-of-concept. Therefore, we used an HTC Vive Pro tracked at 90 Hz by means of two tripod-mounted SteamVR Base Stations 2.0 in an area of $3.5m \times 3.5m$ ($w \times d$) and two VIVE controllers for steering-based navigation and interaction.

We prepared a feasibility check using the tool ourselves to design various guided flows in several areas of an urban scene (Fig. 5(a)). Then we conducted the check in form of a pilot study with seven subjects (6 males, M=27.1, SD=5.4). After getting familiar with the

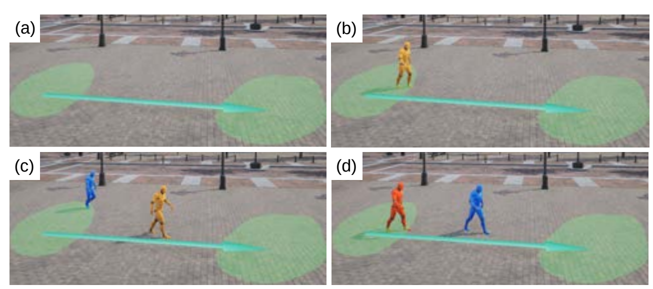

Figure 4: Progressing guided flow from an entry to an exit.

Figure 5: (a) Urban scene. (b, c) Two solutions ensuring VAs enter the building through the back and leave through the front door.

tool by re-creating five pre-defined guiding scenarios, the participants had to solve a complex guiding task: In the initial state, two pedestrian flows entered a building through two narrow, opposite doorways. Sticking to several requirements regarding the flows, subjects had to re-configure the environment ensuring that, i.a., one doorway is configured as entry, the other one as exit. Two exemplary solutions are shown in Figures 5(b) and (c). Afterwards, subjects rated the ease of use of our approach via the System Usability Scale (SUS)[2]. The proposed 5-point scales were used (1 = strongly agree; 5 = strongly disagree). The SUS calculator yielded an average score of 78.21 (*SD*= 7.03).

4 DISCUSSION & CONCLUSION

Although our approach is in an early stage, it has already been proven effective in practice. After creating a json-based extension to store and reload configurations, setting up the pilot study using the VR-based flow control was easy and engaging to do. Although limited, the elements supported us in editing the scene appropriately to match our study needs. We also often used the option to possess a random VA in order to obtain a first-hand impression of the configured flow. Finally, being able to freely change the viewpoint while still being immersed was helpful during the flow configuration, especially in cases of guiding pedestrians from outdoor to indoor areas, or vice versa.

Taking the subjects' feedback into account revealed, that the interface of our sketch-based tool as well as the interaction with the different elements are indeed promising. All subjects were able to use the tool and some even intuitively used options in the familiarization phase which had not been introduced by then. In addition, the average SUS-score indicated an adjective rating of "good". However, different suggestions were made, which will be taken into account in future work. Examples are the request for a better interface to adjust transition probabilities. Especially for end-*sections* in a larger (Euclidean) distance, the interface hovering over the transition's end was hard to read and to interact with. Furthermore, subjects would like to have an option to delete a *barrier* at once instead of pole-wise, or to extend a *section* instead of deleting and re-creating it.

Although a systematic analysis of our approach, e.g., in comparison to common 2D sketching approaches, still needs to be conducted, our proof-of-concept was essential to gain first knowledge on VR-based crowd authoring to determine the course of action. It turned out, that the elements and techniques used for 2D sketching are well applicable in VR, allowing us to build on previously gained knowledge. Furthermore, the option to possess an entity and experience the flow firsthand proves itself as a meaningful option to evaluate the flow guidance. Lastly, adapting the 3rd person sketching perspective in VR was supportive and actively used. While most 2D approaches use a top-down view onto the scene during sketching, users in VR quickly adapted their height to have different perspectives onto the scene. For high-level guiding, a high viewpoint was chosen as it provides a good overview, while fine-tuning was mostly done in a lower height for a more detailed view.

In conclusion, we presented a, preliminary VR-based sketching approach to directly and interactively author pedestrian flows in real-time by means of *barriers* and configurable *sections*. A first proof-of-concept was promising, indicating that a VR-based approach is a suitable research object. In the next steps the current design will be improved and more authoring elements will be added, e.g., storyboard-options [2] or additional techniques to configure the high-level flow distribution [8]. Afterwards, a detailed comparison between our approach and [2] shall provide insight into the differences of VR-based and 2D authoring in terms of usability, time to author, and resulting plausibility of the pedestrian flows. Lastly, using the approach as add-on during scene explorations (cp. *O*2) will be examined.

[2]https://www.usabilitest.com/system-usability-scale; last-visited: 2020-09-13

REFERENCES

[1] Andrea Bönsch, Marcel Jonda, Jonathan Ehret, and Torsten W. Kuhlen. 2020. Towards a Graphical User Interface for Exploring and Fine-Tuning Crowd Simulations. In *IEEE Virtual Humans and Crowds for Immersive Environments*. https://doi.org/10.1109/VRW50115.2020.00033
[2] Luis R.M. Gonzalez and Steve Maddock. 2019. A Sketch-based Interface for Real-time Control of Crowd Simulations that Use Navigation Meshes. In *Proc. 14h Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.* https://doi.org/10.5220/0007344200410052
[3] Wolfgang Hurst and Roland Geraerts. 2019. Augmented and Virtual Reality Interfaces for Crowd Simulation Software-A Position Statement for Research on Use-Case-Dependent Interaction. In *IEEE Virtual Humans and Crowds for Immersive Environment.* https://doi.org/10.1109/VHCIE.2019.8714733
[4] Ioannis Karamouzas, Peter Heil, Pascal Van Beek, and Mark H Overmars. 2009. A Predictive Collision Avoidance Model for Pedestrian Simulation. In *Motion in Games.* https://doi.org/10.1007/978-3-642-10347-6_4
[5] Athanasios Krontiris, Kostas E. Bekris, and Mubbasir Kapadia. 2016. ACUMEN: Activity-centric crowd authoring using influence maps. *ACM Int. Conf. Proceeding Ser.* (2016). https://doi.org/10.1145/2915926.2915935
[6] James McIlveen, Steve Maddock, Peter Heywood, and Paul Richmond. 2016. PED: Pedestrian Environment Designer. *Comput. Graph. Vis. Comput.* (2016). https://doi.org/10.2312/cgvc.20161304
[7] Sahil Narang, Andrew Best, and Dinesh Manocha. [n.d.]. Simulating Movement Interactions Between Avatars & Agents in Virtual Worlds Using Human Motion Constraints. In *IEEE Virtual Real. Conf.* https://doi.org/10.1109/VR.2018.8446152
[8] Aline Normoyle, Maxim Likhachev, and Alla Safonova. 2014. Stochastic Activity Authoring with Direct User Control. In *Proc. Symp. Interact. 3D Graph.* https://doi.org/10.1145/2556700.2556714
[9] Masaki Oshita and Yusuke Ogiwara. 2009. Sketch-Based Interface for Crowd Animation. In *Int. Symp. on Smart Graphics.* https://doi.org/10.1007/978-3-642-02115-2_22
[10] Shawn Singh, Mubbasir Kapadia, Petros Faloutsos, and Glenn Reinman. 2009. An Open Framework for Developing, Evaluating, and Sharing Steering Algorithms. *Motion in Games* (2009). https://doi.org/10.1007/978-3-642-10347-6_15
[11] Branislav Ulicny, Pablo De Heras Ciechomski, and Daniel Thalmann. 2004. Crowdbrush: Interactive Authoring of Real-time Crowd Scenes. In *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animat.* https://doi.org/10.1145/1028523.1028555