

Sketching Vocabulary for Crowd Motion




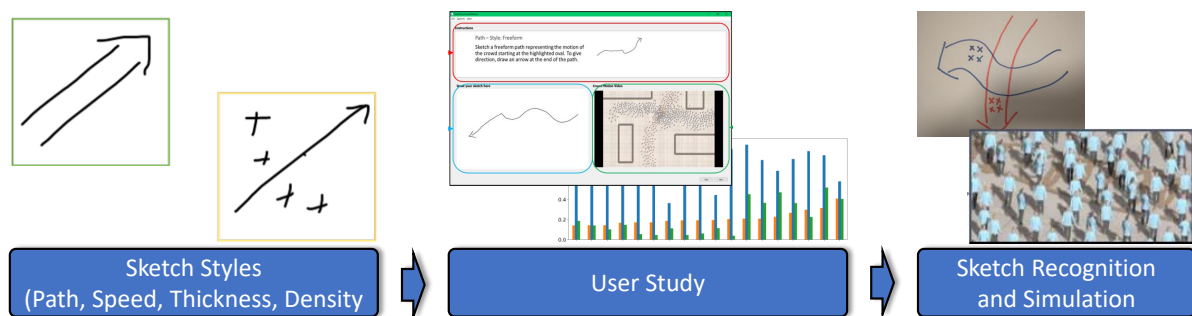
C. D. T. Mathew¹  B. Benes¹  D. G. Aliaga¹ ¹Purdue University, USA

Figure 1: *Sketching Vocabulary for Crowd Motion:* This paper proposes various intuitive sketch styles for specifying crowd motion parameters defining motion path, walking speed, path thickness, and crowd density ("Sketch Styles"+. A prototype system pipeline converts sketches into parameters for a crowd simulation and animation ("Sketch Recognition to Crowd Simulation") and it reports the best styles, and relevant observations, as per a user study ("User Study and Results").

Abstract

This paper proposes and evaluates a sketching language to author crowd motion. It focuses on the path, speed, thickness, and density parameters of crowd motion. A sketch-based vocabulary is proposed for each parameter and evaluated in a user study against complex crowd scenes. A sketch recognition pipeline converts the sketches into a crowd simulation. The user study results show that 1) participants at various skill levels can draw accurate crowd motion through sketching, 2) certain sketch styles lead to a more accurate representation of crowd parameters, and 3) sketching allows to produce complex crowd motions in a few seconds. The results show that some styles although accurate actually are less preferred over less accurate ones.

CCS Concepts

• *Computing methodologies* → *Procedural animation*; • *Human-centered computing* → *User studies*; *Gestural input*;

1. Introduction

Sketching shows human movement in sports playbooks (e.g., soccer, American football, etc.), evacuation diagrams, military tactic diagrams, and urban applications such as evacuation planning. In computer graphics, sketching has found its way into many practical applications, such as 3D modeling for CAD and industrial design [DAI*18; LPBM20; GHL*20]. Sketching works with the assumption that human-understandable simplifications depict the key elements of the sketched phenomenon. For example, in sketching humans [MQW06], silhouettes and facial features are the prominent elements that depict the expression and the motion. Similarly,

sketching urban models [NGA*16; CKX*08] uses street layout to convey the overall style of a city.

Sketching input can be provided to a computer via digital surfaces (e.g., digital pen, trackpad) or traditional surfaces (e.g., paper, existing diagrams). Our work targets traditional or non-digital, sketching which omits dependency on digital pen technologies and enables processing previously performed sketches. This is the more challenging case since it does not provide the implicit digital cleanup and potentially additional information channels. Recent work, such as Simo-Serra et al. [SISI16] or Liu et al. [LRS18], attempt to clean up such raw sketching input for subsequent digital use. Once in a cleaned digital form, concept sketches and other tasks are en-

abled (e.g., Gryaditskaya et al. [GHL*20]). We use a scanner to generate the input required for the pipeline. In our work, we set out to directly work with raw sketches and evaluate the expressiveness of visual sketching gestures for crowd simulation design.

Sketching has been briefly explored in crowd simulation works (e.g., MASSIVE [Mas], Patil et al. [PvdBC*11], Gu et al. [GD13], and Mathew et al. [MBA20]), and these works use sketching to enhance other authoring methods. To the best of our knowledge, no dedicated study has been conducted to establish a sketching vocabulary for crowd motion specification. Thus, it is unclear which are good drawing primitives for sketching crowd motion in raw, traditional, sketches. The state of the art does not explore sketching motions that would enable untrained users to quickly and intuitively express a desired synthetic crowd motion.

The key idea of our work is to represent crowd motion as a base set of crowd motion parameters and then determine, through a user study, which are a preferred set of sketching styles for expressing motion parameter values (Fig. 1). We focus on four parameters of crowd motion: (1) the *path* P which is an ordered set of way points on which crowd agents move, (2) the walking *speed* S of the agents, (3) the *thickness* or width T of the crowd when following a path, and (4) the *density* D of agents defined as the inter-person spacing as they follow a path. We define in total ten sketching styles that stem from their usage in prior published works [BG95; MQW06; SGC04] and experiments (Sec. 3). We have also implemented a testing system where each sketch input is converted to a crowd animation (Sec. 4).

Given the sketching styles and the testing system, we performed a user study to analyze the performance and preference of the sketching styles via both quantitative and qualitative assessments (Sec. 5). Twenty-five participants who self-identified as having low to medium experience in computer animation participated in the user study. Our results show, with strong statistical significance (e.g., at $\alpha = 0.05$), the accuracy, time, and preferences amongst our multiple sketch styles. Some observations of our user-study were: 1) The most accurate style is not necessarily the most preferred style 2) The most accurate style is not an indication of the ease of crowd sketching tasks for low-experience users.

The main contributions of our paper include: (1) an analysis of the trade-offs between accuracy, time and preference for multiple sketching styles, and (2) a proposed sketching language for crowd motion design.

2. Related Work

Depending on its primary focus, sketching can be divided into i) 2D sketching of static 3D objects using artistic and industry-inspired techniques and conventions [ES07; RB13], and ii) sketching of dynamic objects (e.g., Popovic et al. [PSE03] explores control of physics-based simulations on user input sketches) and in particular human motions [Har97; Cam06; Wil12]. We explore sketching as a way of prescribing the desired crowd motion.

Crowd simulation methods focus on interactions among the agents and the environment, centering on collision avoidance and natural-looking motion. These methods may be character-

ized as microscopic or macroscopic. In a microscopic agent-based model, the reaction to the virtual world is based on local rules [GCK*09; WJO*14], velocity obstacles [vdBGLM11; KGM13]), vision-based steering [DMC*17], and other agent-based methods. Macroscopic methods aim to govern the overall behavior of the crowd using space colonization [dLRM*12], continuum fluid-like flows [TCP06], or global optimization methods [KSNG17]. In our work, we seek the best sketching language for representing crowd behavior. While our results are not strictly tied to one crowd simulation engine, we use Bicho et al. [dLRM*12] to create the crowd animations shown to the user study participants. We used this engine as it provides a straightforward way to control crowd thickness and density using the space colonization markers of the engine. Our method can be used with macroscopic simulations if the sketch parameters can be translated into the simulation parameters meaningfully. However, some system modifications to the parameters could also be applied to smaller groups of simulated humans.

Crowd motion design enables specifying high-level goals for crowd movement and focuses on rapid design instead of scripting individual agents. In prior work, Sung et al. [SGC04] defines individual character situation-based behavior when an overall crowd scenario is given. Kapadia et al. [KSA*09] introduces time-varying metrics but does not focus on the overall crowd motion. Wolinski et al. [WJO*14] describes an optimization-based method to tune several simulation algorithms to match a reference motion. Another set of methods focuses on combining patches of crowd motion to populate virtual environments, such as Li et al. [LCS*12] who look for periodic patterns and symmetric arrangements to enable connecting motion patches, extending the work of Yersin et al. [YMPT09]. Lerner et al. [LCL07] combines example real-world trajectories. Jordao et al. [JPCC14] also uses example patches but deforms them to stitch them together. Direct manipulation of agents or trajectories is also explored as a method of crowd authoring. Ulciny et al. [UCT04] uses brush-based tools to author and manipulate crowds. Kim et al. [KSKL14] explores editing trajectories using a deformable cage that can change spatial and temporal arrangements. Kwon et al. [KLLT08] uses a graph structure on top of the trajectories to preserve formation constraints during manipulation. Mathew et al. [MBA20] uses brush-based tools to create crowds and optimization methods to copy crowd behaviors. Although rapid crowd authoring is the focus of most of these works, they do not directly address the notion of creating a sketching language for crowd motion and thus can be considered complementary to this article.

Close to our work are works that bring a limited form of sketching (GUI-based or otherwise) when authoring crowd motion. Jordao et al. [JCC*15] support GUI-based controls' use to specify desired density and flow values over the crowd. Group formation work (e.g., [TYK*09; JCP*10]) allows to draw formations but does not address any expressive sketching for a complex crowd motion. Commercial crowd simulation and authoring software systems, such as Massive [Mas] and as per their feature list, provides a key-frame based high-level crowd control method (i.e., manually specifying agent variables at different instants). Patil et al. [PvdBC*11] supports user-specified line strokes for drawing a vector-based guidance field and enables deforming the field to control high-level agent navigation. Lemonari et al. [LBC*22] catego-

rized modern crowd simulation components, animation and visualization techniques. Collar et al. [CvTZ*22] introduces a sketch-based method consisting of freeform directional arrows for modeling what's known as an interaction field. This interaction field described velocities' or orientations' virtual agents should use around others agents or obstacles. None of these methods attempts to provide a freeform sketching language to specify crowd motion that allows even novice users (see Fig. 6) to prescribe complex crowd motion. A digital canvas [JCC*15] gives us access to additional data such as stroke speed. A GUI-based tool allows to give sliders and controls [MBA20] explicitly, and it allows to records strokes and set desired values. However, a simple static sketch can convey the information about a human motion which is one of the primary motivations of this work (Sec. 1). Hence, our work exploits a digital sketching tool to evaluate the vocabulary but focuses on static sketching and evaluates a sketching vocabulary to represent human motion that would work in static sketches. Fig. 17 shows a result where a paper sketch is converted to an animated crowd motion.

While some crowd sketching systems employ the user sketch to lookup similar content in a database (e.g., [YVN*14]) or within the same sketch (e.g., [WKC*18]), we are more interested in evaluating and formulating expressive yet accurate 2D sketching gestures to generate novel crowd motions by providing high-level control to an underlying crowd simulation. In particular, we focus on sketching gestures to specify the four aspects mentioned above of crowd motion: path, speed, thickness, and density. Our underlying crowd simulation engine [dLRM*12] then takes the sketched input and, together with collision avoidance, produces the crowd motion animation.

3. Sketch Styles

Inspired by prior work [BG95; MQW06; SGC04], we define ten sketch styles that consist of three styles for the path P , two styles for speed S , three styles for thickness T , and two styles for density D . Each style is discussed in detail in this section.

3.1. Path

The **path** of a crowd is the set of ordered waypoints the crowd will move along. It is commonly denoted using some sort of line and arrowhead. We define three styles of sketching: freeform, smooth, and crosses. While a curve is the most general case, our ordered waypoints support the straight motion and approximate curved paths (via a sequence of linear segments). As a note, the Shibuya crossing (Fig. 15) is mainly comprised of straight-like motions of crowds.

Freeform (Fig. 2a): Freeform strokes have been used in several previous works to denote a motion path (e.g., [KGC*17; OO09; MM19]), and thus we adopt this as one of our styles for testing.

Smooth (Fig. 2b): Another option is to provide the user with a continuously smooth path during the interactive sketching process. Balaguer et al. [BG95] perform knot removal, Guay et al. [GCR13] uses cubic Hermite curves, and Arora et al. [AHG*18] uses the path smoothing method from [TSB11]. In this style, we perform a simplified version of Thiel et al. [TSB11] by using a 1D smoothing kernel based on the drawn sketch's length line.

Crosses (Fig. 2c): are inspired by [GCR13]. Here we provide an option of drawing a sequence of crosses (or marks) representing the line of motion. Guay et al. [GCR13] used this as a line of action method to control character poses.

3.2. Speed

The **speed** of the crowd can vary from a very slow speed (e.g., $0.5ms^{-1}$) to a very fast walking speed (e.g., $3ms^{-1}$). In some previous work, the speed of drawing has been used as an indication of the desired speed of an object or character (e.g., Mao et al. [MQW06], and Oshita et al. [OO09]). However, this method prohibits using traditional sketches (e.g., photographs of sketches on paper, whiteboards, or previously-made). Hence, we propose two styles inspired by the modern user interface (UI) design and experimentation.

Arrowhead Count (Fig. 2d): style is based on the icons used for rewind and fast-forward in modern video UI's where the number of arrows denotes the speed of rewinding or fast-forwarding. In our prototype, one to five arrows represent the speed of the crowd, where one arrowhead is the slowest, and five arrowheads are the fastest.

Arrowhead Size (Fig. 2e): Instead of using the speed of sketching, we propose using the size of the arrowhead (similar to how [TSB11] uses inertia) as an indicator for speed, bounded by predefined minimum and maximum speeds.

3.3. Thickness

Thickness controls the space that the crowd will occupy along the normal direction of motion. By allowing to sketch the thickness, a user can spread a crowd among a larger space with a few sketch strokes. Previous work has proposed many ways of sketching thickness. Using these as inspiration, we propose three options.

Arrow Width (Fig. 3a): Sung et al. [SGC04] and Gu et al. [GD13] enable sketching thick 2D lines to specify the area that the crowd should occupy. Inspired by these works, we propose using the width of the arrow as an indicator of thickness. In particular, an arrow is comprised of two parallel lines ending with a large arrowhead.

Parallel Lines (Fig. 3b): Oshita et al. [OO09] asks a user to draw multiple instances of the path to follow and use the repeated sketches to infer a path width value. In our work, this style is transformed into a more straightforward sketch style by using two parallel lines beside a portion of the main path to denote the thickness of the crowd (i.e., the distance between the approximately parallel lines represents thickness). This simplifies the sketch for a longer path, as the parallel lines need not be drawn along the whole path.

Crosses (Fig. 3c): Using crosses to denote the positions of humans is a popular notation in sports playbooks [Dum]. Our version of this style uses two crosses beside the main path (one on each side), where the distance between the crosses denotes the crowd's thickness.

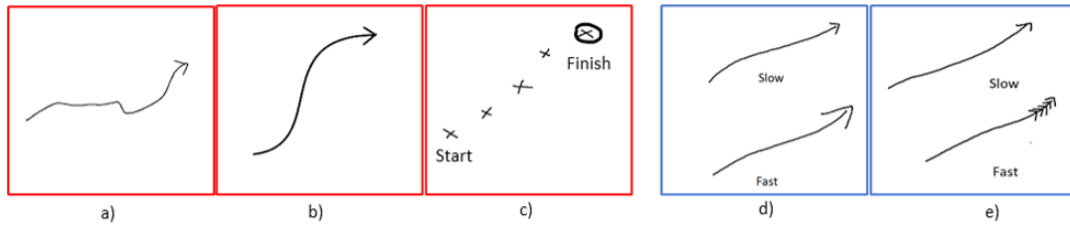


Figure 2: The path styles (red border) and speed sketch styles (blue border). Path Styles: a) Freeform b) Smooth c) Crosses, Speed Styles: d) Arrowhead Size e) Arrowhead Count

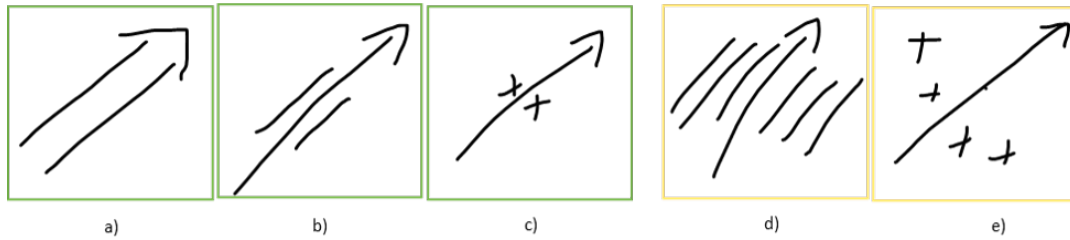


Figure 3: The thickness styles (green border) and density sketch styles (yellow border). Thickness Styles: a) Arrow Width b) Parallel Lines c) Crosses, Density Styles: d) Multiple Lines e) Multiple Crosses

3.4. Density

The crowd **density** corresponds to the average distance between people. Zhang et al. [ZLD15], and others specify density via a GUI. Karamouzas et al. [KSHG18] use a light-to-dark shading of paths to represent density. Density can be described as an extension to thickness. We define two options to represent the average density.

Lines: The user can draw 1-3 (parallel) lines on each side of the main path (thus, 2-6 in total) to represent density. Density is calculated as the average distance between the lines.

Crosses: The users can draw 1-3 crosses on each side of the main path (hence, 2-6 total) to represent density. Density is calculated as the average distance between the intersections of the crosses.

4. System Pipeline

This section describes the system pipeline used for our work. First, we describe a set of virtual scenes provided to the user for sketching during our user study. Then, a sketch recognition engine classifies the sketch and quantifies the corresponding parameter values. The parameter values are given to a crowd simulation engine and a crowd animation is produced.

4.1. Crowd Scene Setup

We define three base configurations to subsequently generate many virtual scenes capturing a variety of virtual layouts.

- A: Two large groups of people crossing – similar setups were used in previous crowd simulation works.
- B: A busy street crossing with people moving in at least 6 directions at a given time.
- C: An open plaza where the crowd moves in various directions in a coordinated manner.

C: An open plaza where the crowd moves in various directions in a coordinated manner.

For each of the above three base configurations, we generate three variations for each of the four crowd motion parameters. This leads to $3 \times 3 \times 4 = 36$ virtual scenes available for the user study.

To create a virtual scene for the user study, we extend the system of [MBA20] which makes use of the crowd simulation system of Bicho et al. [dLRM*12]. First, we create a base scene based on one of the above three configurations. Second, we randomly perturb the four crowd motion parameters of the scene (*i.e.*, alter the path P , speed S , thickness T and density D of the crowd motion). The perturbations are performed in the following manner:

1. The **path** is randomly divided into four to eight sections. Each section is randomly displaced in the direction of its normal by -20% to $+20\%$ of its length. Then, Catmull-Rom interpolation is used to smoothly join the segments into a continuous path.
2. The crowd walking **speed** is randomly selected to be a value between 0.5 to 4 ms^{-1} .
3. The crowd **thickness** (or width) along the path is randomly selected to be between 1 and 20 meters.
4. The crowd **density** has a random value between 0.2 - 1 m^2 .

4.2. Sketch Recognition Algorithms

We developed a set of prototype image-based recognition algorithms to automatically parse the drawn sketches. The algorithms convert the input pixels to strokes and then convert the strokes to simulation parameters in order to drive the crowd simulation engine. Our current algorithms all run at interactive rates (12 fps for a 640×480 image) and at 91% accuracy (out of a dataset of 412 images created for validation purposes).

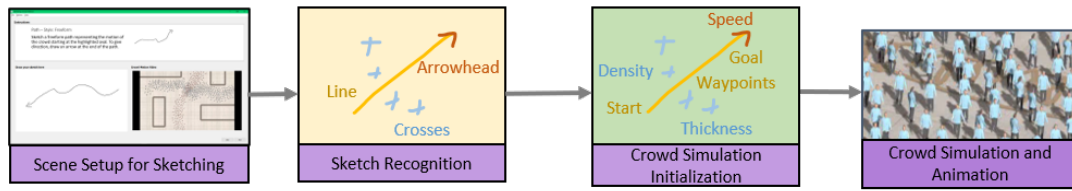


Figure 4: System Pipeline: A black-and-white sketch is recognized and broken down into individual components of the sketch vocabulary. These are then mapped to simulation initialization parameters that drive a later crowd simulation.

4.2.1. Pixels to Strokes

The user sketch is a grayscale image that is binarized to zero (black; sketch pixel) and one (white; background pixel). A sequence of black pixels is grouped together as a stroke by using contour following [Sob78]. At any intersection of strokes the pixels that form a low variance in angle are considered to belong to the same stroke.

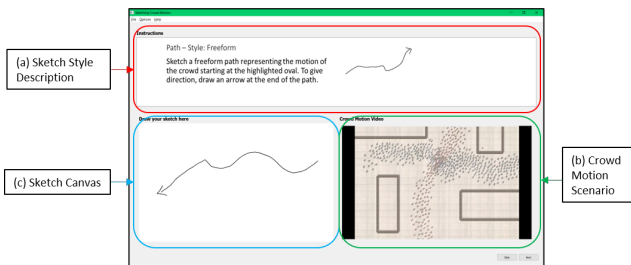


Figure 5: GUI of the application used for the user study was designed to show (a) the sketch style and parameter to be used, (b) an example crowd video for the user to observe and replicate as a sketch, (c) a canvas area for sketching.

4.2.2. Classification

After identifying individual strokes, we classify them into sketch elements that are main path, arrowheads, crosses, lines, and a circle (which is used to indicate the final cross in style Path:Crosses). We disregard any stroke that is less than 5% of the canvas diagonal. We start with the classification of the main path which is the longest stroke (except for style Path:Crosses Sec. 3.1). The arrowhead is assumed to be at one end of the path and is defined by two strokes at an angle between 120° and 150° . For style Speed:Multiple Arrowheads, we expect one to five arrow heads to be identified. Crosses (as seen in styles Path:Crosses, Thickness:Crosses, Density:Crosses) are identified by two strokes that intersect near the middle 40%-60% of each other. To identify sketched lines approximately parallel to the main path stroke (e.g., as in Thickness:Parallel Lines, and Density: Parallel Lines) we compare the average angle between the drawn lines to the main path, and select ones that are $\pm 10^\circ$. For the Thickness:Arrow Width style, two main strokes are identified and the average path between them is taken to identify the main path. In all cases the arrow head gives the direction of the main path.

Sketch Param.	Control mechanism for crowd simulation
Path	Waypoints for navigation offset by normal distance to sketch center line
Speed	Desired speed at each timestep
Thickness	Width of the Marker placement from the main center line
Density	Density of Marker placement

Table 1: Process of converting sketch parameters to crowd simulation parameters

4.2.3. Crowd Simulation Parameters

The aforementioned crowd motion parameters are then used as input to a crowd simulation engine (Tab. 1). The crowd simulation algorithm [dLRM*12] relies on the placement of a set of markers that are spaced out 2D points on walkable areas. Virtual humans need to acquire them during one time step in order to move forward. By changing the marker placement to the desired thickness, we control the thickness of the moving crowd. By changing the density of the marker placement, we control the crowd's density while in motion. The desired speed of the crowd is set as input at every time step. The main path indicated by the sketch is considered the center-line of the path for the set of agents. To derive the path of each virtual human, the normal vector from the center-line is calculated relative to the starting position, and then this normal vector is used at equidistant points along the center-line to derive the waypoints of an agent. This assures that the virtual humans move as a natural group instead of forming a line and traveling just along the center-line. We add a random variable of $X \sim \mathcal{N}(0, 0.25m^2)$ to x and y directions to randomize the motion and reduce coordination among the crowd. Default values (e.g., a default walking speed of $1.2ms^{-1}$ and a crowd density of 1m between each person) are used when the sketch does not give information about these parameters. The simulation is initialized with the values from the sketches and the crowd is simulated until all virtual humans cease moving, resulting in the completion of the crowd animation.

5. User Testing Methodology

A desktop application was designed to evaluate each crowd motion parameter (i.e., P , S , T , and D). The GUI of the application (shown in Fig. 5) has three main areas. The top section indicates the sketch style that should be used. The bottom right section displays a crowd

motion scenario. The bottom left section contains the digital canvas where the participants should sketch the crowd motion scenario using the indicated style. Further, there is a "Clear" button to clear the digital canvas of any strokes at the bottom of the application and a "Next" button to proceed to the next trial.

Each user used the system remotely via Zoom to accommodate the pandemic issues at the time of this study. Each user controlled the application on their screen and performed their sketches, but the application ran on the remote computer of the study supervisor. All strokes and the time taken per stroke were recorded.

Each anonymous study participant:

1. Filled out a form for basic demographics. The form asked the participant's age, handedness, device used (mouse or touch device), and experience level with computer games, computer illustration, and computer animation.
2. The participant was shown a five-minute presentation that explained the user study process.
3. The participant was given full remote control of a full-screen application.
4. The participant then performed two trial sketches to get familiar with the functionality of the application. The performance during these two trial sketches was not recorded.
5. The participant was then asked to perform nine trials (three scenes and three path styles each) to evaluate the path sketch styles. In each trial, a crowd motion video was randomly selected out of a pre-recorded set and one of the corresponding path sketch styles was randomly selected. The participant was requested to sketch the observed crowd motion using the shown path sketch style.
6. Following the same procedure as the path styles, the participant was asked to sketch 6 trials for speed evaluation (three scenes and two speed styles), 9 trials for thickness evaluation (three scenes and three thickness styles), and 6 trials for density evaluation (three scenes and two density styles). For each user, out of the total 36 scenes, a subset of 30 (randomly chosen) was used.
7. The user study concluded with a questionnaire (see below).

5.1. Qualitative Questionnaire

The questionnaire requested feedback on each sketch style by asking for the preference measured by a five-level Likert scale, showing options of (5) Strongly Agree, (4) Agree, Neutral (3), Disagree (2), and Strongly Disagree (1). For each parameter (P, S, T, D), the preference of each style was ranked by each participant using the scale from 1-5. Feedback and comments related to why the preference options were picked were also asked from the participants.

5.2. Error Metrics

We use P_E , S_E , T_E , D_E to denote path error, speed error, thickness error and density error. To compute the path error we also make use of discrete Frechet distance F of a trajectory of a virtual human between the trial scene and the ground truth scene. The subscript GT is used to denote ground truth, and the lack of a subscript denotes the parameter values comes from a trial. Hence, the path's

error metric and the error metric for speed, thickness, and density is as follows:

$$P_E = \text{norm}[\sum F] \quad (1)$$

$$X_E = \text{norm}[\sum (X - X_{GT})^2], \quad (2)$$

where $X \in \{D, S, T\}$.

The participant's sketches were analyzed. Each error metric was normalized by the maximum error bound using the function *norm*, and the accuracy was plotted as 1 – Error in the result figures. The following process was used:

1. The methods of Sec. 4.2 were used to identify the presence of sketch elements (e.g., main path, arrowheads, crosses, parallel lines), map them to crowd simulation parameters, and drive a crowd simulation for the respective scene (e.g., A, B or C) which, in turn, produced a set of trajectories.
2. The trajectories were then compared against the ground truth trajectories of the respective crowd motion content, using the aforementioned error metrics. The averaged accuracy grouped by sketch style provided a value for the quantitative analysis.
3. The preference score was obtained by giving an integer value of 1 to 5 to the Likert's scale values, averaging these values, and grouping by sketch style.

6. Results and Analysis

This section validates our approach by analyzing the quantitative and qualitative data gathered from the conducted user study. In particular, we show the statistical significance of the results in the context of establishing a sketching language of crowd motion. We used the two-tailed t-test to analyze the statistical significance of a pair of results.

6.1. User Study Participants

The user study had a total of twenty-five participants between the ages of 20 and 52. The study was pre-approved by our university's IRB, and all participants were volunteers, subject to minimal risk, and participation time was limited by design to avoid fatigue. The participants were recruited through an email campaign targeted toward computer graphics students and researchers. The participants had varying degrees of experience in Computer Animation (Fig. 6). The majority of them can be considered novices as they classified themselves as having low experience.



Figure 6: Computer Animation Experience of User Study Participants. The user study requested all participants to rate their experience in Computer Animation on a scale of 1(low)-5(high).

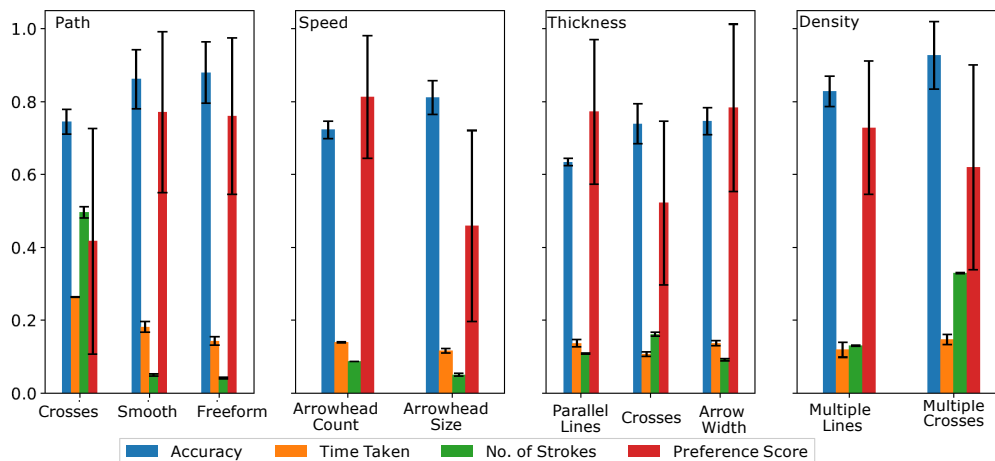


Figure 7: Performance of each parameter and style combination; sorted by accuracy within each style. Each parameter is normalized within $[0,1]$ using the method described in Sec. 5.2.

The sketches resulting from each trial were analyzed, and the accuracy was computed making use of our error metrics (Eqn. 1, 2). The user-study application also gathered the *time-taken* and *number-of-strokes* for each trial. The qualitative questionnaire at the end of each trial (Sec. 5.1) allows us to compute preference scores for the sketch styles by averaging the user-provided ratings.

To show the variety of our virtual scenes, we show the accuracy, time-taken, and number-of-strokes for each of the 36 virtual scenes (Sec. 4.1) presented to the users. Although the scenes are shown in a randomized order to the users, we sort the 36 scenes by the average time taken across all trials and all users. This sorting helps to show that our scenes vary considerably in terms of accuracy and number-of-strokes for all time-taken values.

Fig. 9 shows example sketches drawn to indicate the desired crowd thickness for a specific scene. This particular user, on average, obtained high-accuracy results. In our accompanying video, more example sketches are provided.

6.2. First Order Analysis

As a first-order analysis, Fig. 7 shows the accuracy, time-taken, number-of-strokes, and also *preference score* for each sketch style, and the standard deviation of these values amongst all users and trials. The values mentioned above are grouped per parameter and within each parameter sorted by increasing accuracy.

Path: Users show notably better accuracy using freeform and smooth path styles as compared to crosses path style at a statistical significance value $\alpha = 0.05$ (*i.e.*, freeform and smooth path styles were not statistically different ($p = .2502$). Still, the accuracy advantage of both was significant at $p < 1 \times 10^{-3}$ and $p < 1 \times 10^{-3}$ for freeform and smooth path styles compared to the crosses style, respectively). Moreover, freeform and smooth path styles also exhibit better time-taken for path sketching. As per the preference score, these two styles are significantly and almost equally preferred over crosses path style (*i.e.*, $p < 1 \times 10^{-3}$ and $p < 1 \times 10^{-3}$). Nonetheless, the freeform path style is consistently

better in all three accuracy measures, time taken, and the number of strokes. Some commentary feedback from the users includes "freeform gives fine control" and "smooth curve would fix mistakes", thus further supporting the preference of those two styles.

Speed: Users were more accurate using the arrowhead size style over the number of arrowheads also at $\alpha = 0.05$ (*i.e.*, $p = 0.0318$). However, users preferred the arrowhead count style (also statistically significant, $p = 7.76 \times 10^{-17}$). Users did refer to the familiarity of the arrowhead count style as one explanation for their preference. At least one user who indicated they preferred the arrowhead size style explained that "the continuous nature of the style allowed for more fine-grained control" and "the single stroke made it easier to draw".

Thickness: The arrow width style is more accurate than the parallel lines style. But, the difference of these is not statistically significant at ($p = 0.01$) but of similar accuracy to the crosses style ($p = 0.826$) (since $p > 0.05$, they are not statistically significant). Interestingly, the crosses style was least preferred by the users by a large statistically significant margin. (*i.e.*, crosses style preference score having $p < 1 \times 10^{-3}$ and $p < 1 \times 10^{-3}$ against arrow width and parallel lines styles). One user wrote, "the crosses style seemed to raise an ambiguity as to whether the thickness started at the bottom vertices of the cross or the intersection or the top vertices". However, another user wrote, "it was the most intuitive to illustrate thickness", thus clearly showing conflicting preferences.

Density: Concerning the density styles, the multiple crosses style shows higher accuracy at significance level $p = 1 \times 10^{-3}$ as compared to the multiple lines style. However, the crosses style was also the least preferred style ($p < 1 \times 10^{-3}$). The users who did prefer the crosses styles indicated "it was the most intuitive". In contrast, those who did not prefer it wrote: "the number of strokes was more than with parallel lines."

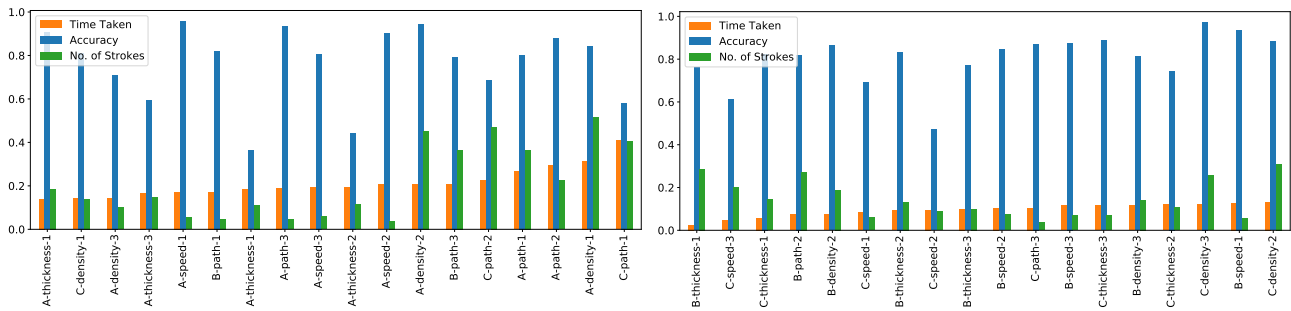


Figure 8: Virtual scenes used by the user study for participants to observe and sketch the crowd motion; ordered by increasing time-taken – shortest time is for top-row left and longest time is for bottom-row right. Each parameter is normalized within $[0,1]$ using the method described in Sec. 5.2. Scene A, B, C is shown in the paper video.

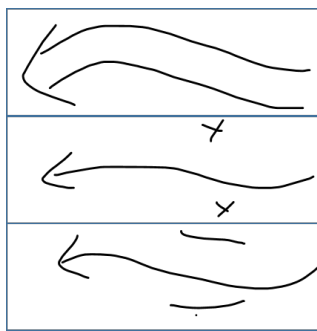


Figure 9: Example sketches for thickness parameter by a user exhibiting high accuracy. The styles from top to bottom: Arrow Width, Crosses, Parallel Lines.

6.3. Insights

By plotting accuracy against the time taken, we show which styles are easier to sketch with high accuracy (Fig. 10). The styles with points that cluster near the upper left of each graph would seem to be better choices for sketching crowd motion. On the one hand, we see that the density style crosses and path style freeform are best in terms of accuracy and time. On the other hand, thickness style parallel lines and path style crosses appear to have the most diverse and lower performance. In Fig. 11, we sort all ten styles by accuracy over time-taken to provide an approximate sorting of the overall relative performance of the styles. Statistical analysis between Density (Multiple Crosses) and other styles for this metric show that the p-values fall between 0.01 and $p=6.46 \times 10^{-13}$ showing a significant difference.

By inspecting the error histograms for each parameter (where the error is the normalized inverse accuracy values – *i.e.*, smaller is better), we observe that users draw some parameters better than others (Fig. 12). In particular, users perform best sketching density and path parameters and perform worst sketching speed and thickness parameters. One potential reason for the inferior performance of speed and thickness is that these parameters are more difficult to quantify for an average user visually.

Correlating the computed metrics leads to some interesting in-

sights (Fig. 13). First, following our intuition that more strokes will take more time, the numbers of strokes are positively correlated. Second, the preference of a style is negatively correlated to the time taken. This confirms that the more time a style takes, the less it is preferred regardless of the accuracy.

We observe that both low-skilled and medium-skilled users (in animation) obtain similar sketching accuracy values by considering user expertise. As a simple analysis, Fig. 14 we group all users with an experience level in computer animation less than 3 into "Low-skill" and those with an experience level greater than 3 into "Medium-Skill". While there are multiple ways the responses could be analyzed we opted to ignore users who rated "3" to focus on the extreme skill levels. Then, we show the average accuracy values for each style. In general, the accuracy values amongst the different parameters appear similar between low and medium-skilled users. A straightforward t-test tells us the accuracy of low-skilled and medium-skilled users are statistically similar even for a p-value of 0.1 except for the Thickness (Parallel Lines) style. The lowest value out of the statistically similar accuracy observed is 0.18.

6.4. Recommended Styles

Based on the previous analysis and considering the accuracy, preference score, and time taken, we make the following recommendations for each crowd motion parameter (Tab. 2).

Param.	Style
Path	Freeform
Speed	Arrowhead Count
Thickness	Arrow width
Density	Crosses

Table 2: Recommendations of choices for sketching based on our user study.

We used more complex sketches to create examples using our visual sketching language. Examples in Fig. 15 and Fig. 16 shows the usage of the recommended style. Corresponding animations are present in the accompanying paper video.

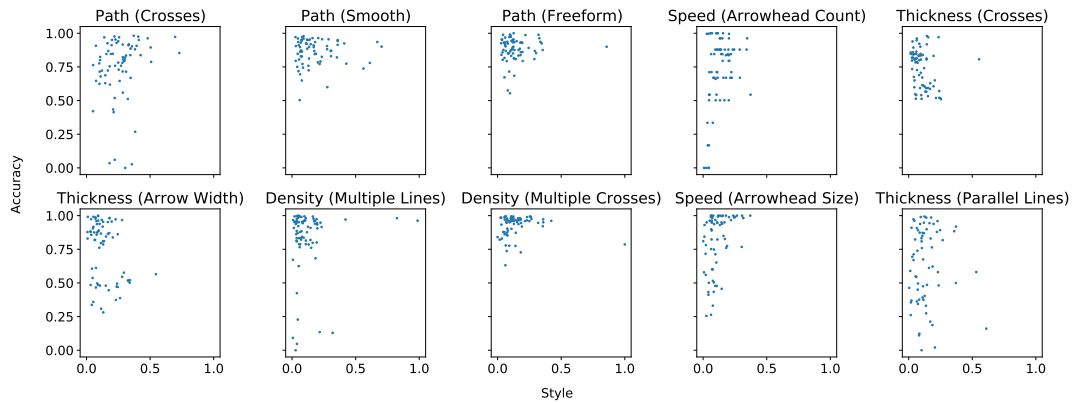


Figure 10: Accuracy (vertical axis) vs. time taken (horizontal axis) for all styles. Styles with clustering in the upper left imply high accuracy and low time taken, which we presume to mean the method works well. In contrast, styles with a wide spread of points imply a wide variety of user behavior. Thus, density-crosses and path-freeform appear to be the best-performing styles.

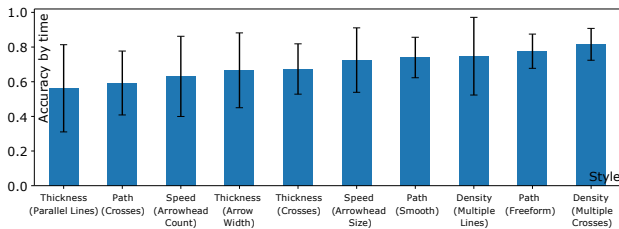


Figure 11: The accuracy divided by time is plotted against time and sorted from the worst to the best.

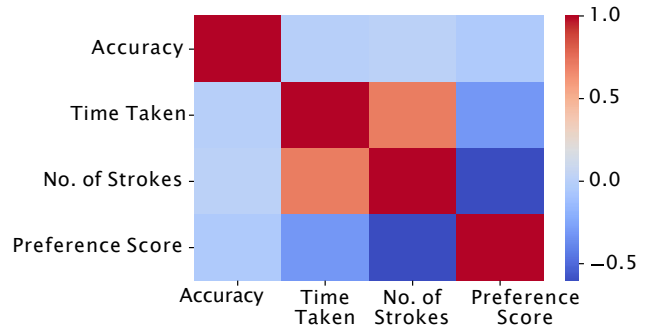


Figure 13: Positive and negative correlations of each metric; The warmer (red) colors indicate a positive correlation and the cooler (blue) colors indicate a negative correlation.

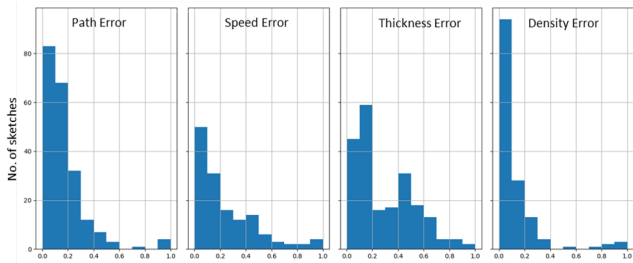


Figure 12: Error histogram for each parameter. From left to right: Path, Speed, Thickness, and Density.

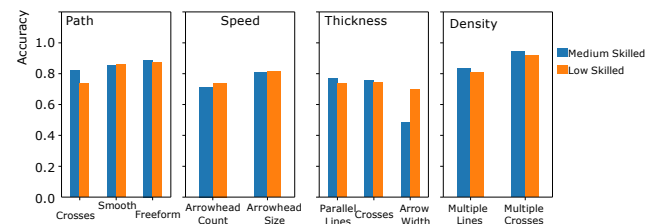


Figure 14: The comparison of accuracy between low-skilled and medium-skilled participants.

7. Conclusion, Limitations and Future Work

We have presented a summary of evaluating the expressiveness of various traditional sketching gestures for crowd simulation design. Instead of assuming digital sketching input and requirements, we address the more difficult case of traditional non-digital sketching (e.g., pen and paper), which provides more freedom and ease to the human but more challenges to the computer interface. Our sketch-based method opens up the realm of crowd motion authoring to less trained users allowing them to design complex scenes. We anticipate the study and proposed sketching language will help foment better crowd simulation design systems.

We have reported the results of our user study and have tested preliminary results using the BioCrowds crowd simulation method [dLRM*12]. However, our analysis can be generalized to other well-known crowd simulation methods such as ORCA (a velocity-driven, collision-avoidance method) or Continuum Crowds [TCP06] (a fluid-based method). The parameters obtained by our gestures can be mapped to the appropriate fluid flow parameters.

While we did explore the effect of different gesture styles on the

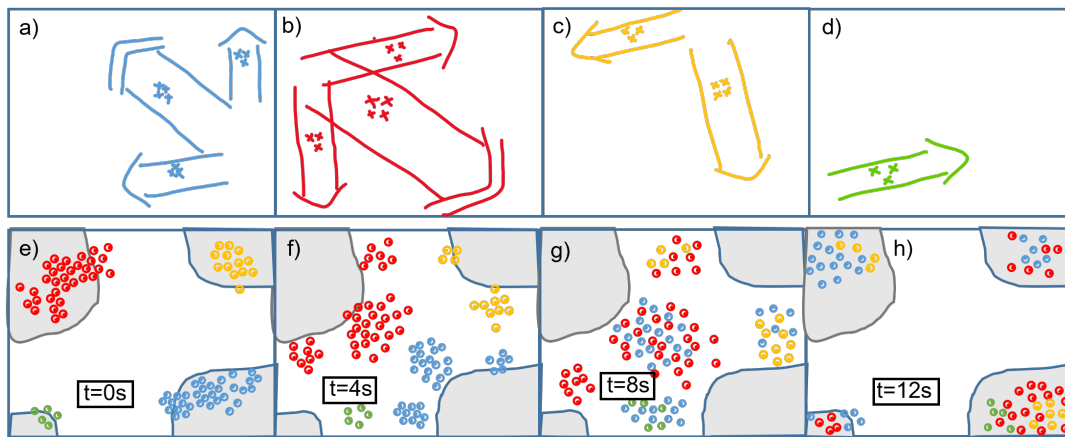


Figure 15: Shibuya crossing (in Japan) created using the proposed sketching language in Table 2. a) Three sketched paths for the crowd in the lower-right (blue) b) Three paths to take for the crowd in the top-left (red) c) Two paths to take for the crowd in top-right (yellow) d) The path to take for the crowd in the lower-left (green). e-h) Shows the crowds moving at simulation time $t=0s, 4s, 8s, 12s$.

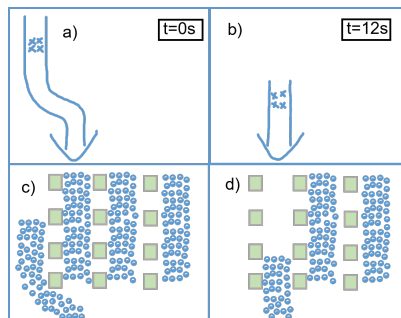


Figure 16: Crowds in a Tokyo market square scene created using the proposed sketching language with keyframes. The keyframes are an extension on top of the proposed language to accommodate time-varying motions. a) The sketch created for moving the leftmost crowd at $t=0s$. b) The sketch created for moving part of the second to the left crowd at $t=12s$. c-d) The crowds moving based on the keyframed sketches.

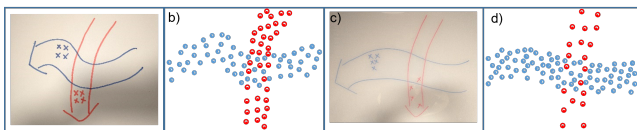


Figure 17: Photographs of sketches on static surfaces being turned to crowd motion. a) Photograph of crowd motion sketch drawn on paper. b) Crowds moving corresponding to part a). c) Photograph of crowd motion sketch drawn on a whiteboard. d) Crowds moving corresponding to part d)

accuracy, time taken, and the number of strokes, more aspects can be explored requiring further user studies. For example, i) our work has not addressed the ability to change the parameters within a stroke itself, *i.e.*, to change speed, density, thickness. This is a current limitation of the system but not of the vocabulary. For example,

the thickness of a crowd can be started with an arrow the starting wide and then taper down to a thin line. Such a change can be depicted, and the system can be extended beyond the proof of concept in this paper to a more functional aspect as future work. ii) We have not explored developing sketching curves that produce interactions between them (*e.g.*, forcing vortices to occur). While the underlying crowd simulation displaces agents to avoid collisions, the sketch could indicate more specific inter-agent behaviors. iii) Also, introducing an event sequence through the sketching itself (*e.g.*, possibly by associating a number with each main path) would allow a traditional sketch to create a complex key-framed crowd motion sequence. iv) Furthermore, extending the language to include intra-crowd formations, such as social groups and leader-follower, would provide further sophistication v) The statistical procedures used to analyze the data could further benefit from the results of alternative analysis methods such as the Mann-Whitney test, vi) Finally, different groupings of the skill of users (*e.g.*, low-medium-high) could further enhance the understanding of the results.

References

- [AHC*18] ARORA, RAHUL, HABIB KAZI, RUBAIAT, GROSSMAN, TOVI, et al. "Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ". *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, 1–15 3.
- [BG95] BALAGUER, JEAN-FRANCIS and GOBBETTI, ENRICO. "Sketching 3D animations". *Computer Graphics Forum*. Vol. 14. 3. Wiley Online Library. 1995, 241–258 2, 3.
- [Cam06] CAMARA, SERGI. *All about techniques in drawing for animation production*. Barron's Educational Series, 2006 2.
- [CKX*08] CHEN, XUEJIN, KANG, SING BING, XU, YING-QING, et al. "Sketching reality: Realistic interpretation of architectural designs". *ACM Transactions on Graphics (TOG)* 27.2 (2008), 1–15 1.
- [CVTZ*22] COLAS, ADÈLE, van TOLL, WOUTER, ZIBREK, KATJA, et al. "Interaction Fields: Intuitive Sketch-based Steering Behaviors for Crowd Simulation". *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, 521–534 3.

- [DAI*18] DELANOY, JOHANNA, AUBRY, MATHIEU, ISOLA, PHILLIP, et al. "3d sketching using multi-view deep volumetric prediction". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.1 (2018), 1–22 1.
- [dLRM*12] De LIMA BICHO, ALESSANDRO, RODRIGUES, RAFAEL ARAÚJO, MUSSE, SORAIA RAUPP, et al. "Simulating crowds based on a space colonization algorithm". *Comp. & Graph.* 36.2 (2012), 70–79 2–5, 9.
- [DMC*17] DUTRA, T. B., MARQUES, R., CAVALCANTE-NETO, J.B., et al. "Gradient-based Steering for Vision-based Crowd Simulation Algorithms". *Comput. Graph. Forum* 36.2 (May 2017), 337–348. ISSN: 0167-7055 2.
- [Dum] DUMMIESCOM. *Running Plays in Football*. <https://www.dummies.com/sports/football/offense/running-plays-in-football/>. (Accessed on 12/16/2020) 3.
- [ES07] EISSEN, KOOS and STEUR, ROSELIEN. *Sketching drawing techniques for product designers Koos Eissen and Roselien Steur*. Page One, 2007 2.
- [GCK*09] GUY, S. J., CHHUGANI, J., KIM, C., et al. "ClearPath: Highly parallel collision avoidance for multi-agent simulation". *Proceedings of the SCA*. Aug. 2009, 177–187 2.
- [GCR13] GUAY, MARTIN, CANI, MARIE-PAULE, and RONFARD, RÉMI. "The line of action: an intuitive interface for expressive character posing". *ACM Transactions on Graphics (TOG)* 32.6 (2013), 1–8 3.
- [GD13] GU, QIN and DENG, ZHIGANG. "Generating freestyle group formations in agent-based crowd simulations". *IEEE Computer Graphics and Applications* 33.1 (2013), 20–31 2, 3.
- [GHL*20] GRYADITSKAYA, YULIA, HAHNLEIN, FELIX, LIU, CHENXI, et al. "Lifting freehand concept sketches into 3D". *ACM Transactions on Graphics (TOG)* 39.6 (2020), 1–16 1, 2.
- [Har97] HART, CHRISTOPHER. *How to draw animation*. Watson-Guptill Publications, 1997 2.
- [JCC*15] JORDAO, KEVIN, CHARALAMBOUS, PANAYIOTIS, CHRISTIE, MARC, et al. "Crowd art: density and flow based crowd motion design". *MIG*. 2015 2, 3.
- [JCP*10] JU, EUNJUNG, CHOI, MYUNG GEOL, PARK, MINJI, et al. "Morphable Crowds". *ACM SIGGRAPH Asia*. ACM, 2010, 140:1–140:10. ISBN: 978-1-4503-0439-9 2.
- [JPCC14] JORDAO, KEVIN, PETTRÉ, JULIEN, CHRISTIE, MARC, and CANI, MARIE-PAULE. "Crowd sculpting: A space-time sculpting method for populating virtual environments". *Comput. Graph. Forum* 33 (2014), 351–360 2.
- [KGC*17] KAZI, RUBAIAT HABIB, GROSSMAN, TOVI, CHEONG, HYUNMIN, et al. "DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design." *UIST*. Vol. 14. 2017, 401–414 3.
- [KGM13] KIM, SUJEONG, GUY, STEPHEN J., and MANOCHA, DINESH. "Velocity-Based Modeling of Physical Interactions in Multi-Agent Simulations". *SCA '13*. Anaheim, California: Association for Computing Machinery, 2013, 125–133. ISBN: 9781450321327 2.
- [KLLT08] KWON, TAESOO, LEE, KANG HOON, LEE, JEHEE, and TAKAHASHI, SHIGEO. "Group motion editing". *ACM Transactions on Graphics (TOG)* 27.3 (2008), 1–8 2.
- [KSA*09] KAPADIA, MUBBASIR, SINGH, SHAWN, ALLEN, BRIAN, et al. "SteerBug: An Interactive Framework for Specifying and Detecting Steering Behaviors". *Proc. of SCA*. SCA '09. New Orleans, Louisiana: ACM, 2009, 209–216. ISBN: 978-1-60558-610-6 2.
- [KSHG18] KARAMOUZAS, IOANNIS, SOHRE, NICK, HU, RAN, and GUY, STEPHEN J. "Crowd Space: A Predictive Crowd Analysis Technique". *ACM Trans. Graph.* 37.6 (Dec. 2018), 186:1–186:14. ISSN: 0730-0301 4.
- [KSKL14] KIM, JONGMIN, SEOL, YEONGHO, KWON, TAESOO, and LEE, JEHEE. "Interactive Manipulation of Large-scale Crowd Animation". *ACM Trans. Graph.* 33.4 (July 2014), 83:1–83:10. ISSN: 0730-0301 2.
- [KSNG17] KARAMOUZAS, IOANNIS, SOHRE, NICK, NARAIN, RAHUL, and GUY, STEPHEN J. "Implicit Crowds: Optimization Integrator for Robust Crowd Simulation". *ACM Trans. Graph.* 36.4 (2017), 136:1–136:13. ISSN: 0730-0301 2.
- [LBC*22] LEMONARI, MARILENA, BLANCO, RAFAEL, CHARALAMBOUS, PANAYIOTIS, et al. "Authoring Virtual Crowds: A Survey". *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, 677–701 2.
- [LCL07] LERNER, ALON, CHRYSANTHOU, YIORGOS, and LISCHINSKI, DANI. "Crowds by Example". *Comput. Graph. Forum* 26 (2007), 655–664 2.
- [LCS*12] LI, YI, CHRISTIE, MARC, SIRET, ORIANNE, et al. "Cloning Crowd Motions". *Proc. of SCA*. SCA '12. Lausanne, Switzerland: Eurographics Association, 2012, 201–210. ISBN: 978-3-905674-37-8 2.
- [LPBM20] LI, CHANGJIAN, PAN, HAO, BOUSSEAU, ADRIEN, and MITRA, NILOY J. "Sketch2CAD: sequential CAD modeling by sketching in context". *ACM Transactions on Graphics (TOG)* 39.6 (2020), 1–14 1.
- [LRS18] LIU, CHENXI, ROSALES, ENRIQUE, and SHEFFER, ALLA. "StrokeAggregator: Consolidating Raw Sketches into Artist-Intended Curve Drawings". *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301 1.
- [Mas] MASSIVESOFTWARE. *Massive*. URL: <http://www.massivesoftware.com/>.
- [MBA20] MATHEW, THARINDU, BENES, BEDRICH, and ALIAGA, DANIEL. "Interactive Inverse Spatio-Temporal Crowd Motion Design". *Symposium on Interactive 3D Graphics and Games*. I3D '20. San Francisco, CA, USA: Association for Computing Machinery, 2020. ISBN: 9781450375894 2–4.
- [MM19] MONTANA GONZALEZ, LR and MADDOCK, SC. "A sketch-based interface for real-time control of crowd simulations that use navigation meshes". *Proceedings of the 14th International Conference on Computer Graphics Theory and Applications*. Vol. 1. SciTePress. 2019, 41–52 3.
- [MQW06] MAO, CHEN, QIN, SHENG FENG, and WRIGHT, DAVID K. "Sketching-out virtual humans: from 2D storyboarding to immediate 3D character animation". *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. 2006, 76–es 1–3.
- [NGA*16] NISHIDA, GEN, GARCIA-DORADO, IGNACIO, ALIAGA, DANIEL G., et al. "Interactive Sketching of Urban Procedural Models". *ACM Trans. Graph.* 35.4 (July 2016), 130:1–130:11. ISSN: 0730-0301 1.
- [OO09] OSHITA, MASAKI and OGIWARA, YUSUKE. "Sketch-based interface for crowd animation". *International Symposium on Smart Graphics*. Springer. 2009, 253–262 3.
- [PSE03] POPOVIĆ, JOVAN, SEITZ, STEVEN M, and ERDMANN, MICHAEL. "Motion sketching for control of rigid-body simulations". *ACM Transactions on Graphics (TOG)* 22.4 (2003), 1034–1054 2.
- [PvdBC*11] PATIL, SACHIN, van den BERG, JUR, CURTIS, SEAN, et al. "Directing Crowd Simulations Using Navigation Fields". *IEEE Transactions on Visualization and Computer Graphics* 17.2 (Feb. 2011), 244–254. ISSN: 1077-2626 2.
- [RB13] ROBERTSON, SCOTT and BERTLING, THOMAS. *How to Draw: drawing and sketching objects and environments from your imagination*. Design Studio Press, 2013 2.
- [SGC04] SUNG, MANKYU, GLEICHER, MICHAEL, and CHENNEY, STEPHEN. "Scalable behaviors for crowd simulation". *Comp. Graph. Forum* (2004). ISSN: 1467-8659 2, 3.
- [SISI16] SIMO-SERRA, EDGAR, IZUKA, SATOSHI, SASAKI, KAZUMA, and ISHIKAWA, HIROSHI. "Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup". *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301 1.
- [Sob78] SOBEL, IRVIN. "Neighborhood coding of binary images for fast contour following and general binary array processing". *Computer graphics and image processing* 8.1 (1978), 127–135 5.

- [TCP06] TREUILLE, ADRIEN, COOPER, SETH, and POPOVIĆ, ZORAN. “Continuum crowds”. *ACM Trans. Graph.* 25.3 (2006), 1160–1168 2, 9.
- [TSB11] THIEL, YANNICK, SINGH, KARAN, and BALAKRISHNAN, RAVIN. “Elasticurves: exploiting stroke dynamics and inertia for the real-time neatening of sketched 2D curves”. *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 2011, 383–392 3.
- [TYK*09] TAKAHASHI, SHIGEO, YOSHIDA, KENICHI, KWON, TAESOO, et al. “Spectral-Based Group Formation Control”. *Comput. Graph. Forum* 28 (2009), 639–648 2.
- [UCT04] ULICNY, BRANISLAV, CIECHOMSKI, PABLO DE HERAS, and THALMANN, DANIEL. “Crowdbrush: Interactive Authoring of Real-time Crowd Scenes”. *Proc. of SCA*. SCA '04. Grenoble, France: Eurographics Association, 2004, 243–252. ISBN: 3-905673-14-2 2.
- [vdBGLM11] Van den BERG, JUR, GUY, STEPHEN J., LIN, MING, and MANOCHA, DINESH. “Reciprocal n-Body Collision Avoidance”. *Robotics Research*. Ed. by PRADALIER, CÉDRIC, SIEGWART, ROLAND, and HIRZINGER, GERHARD. Springer Berlin Heidelberg, 2011, 3–19 2.
- [Wil12] WILLIAMS, RICHARD. *The animator's survival kit: a manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators*. Macmillan, 2012 2.
- [WJO*14] WOLINSKI, D., J. GUY, S., OLIVIER, A.-H., et al. “Parameter Estimation and Comparative Evaluation of Crowd Simulations”. *Comput. Graph. Forum* 33.2 (May 2014), 303–312. ISSN: 0167-7055 2.
- [WKC*18] WILLETT, NORA S, KAZI, RUBAIAT HABIB, CHEN, MICHAEL, et al. “A mixed-initiative interface for animating static pictures”. *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 2018, 649–661 3.
- [YMPT09] YERSIN, BARBARA, MAIM, JONATHAN, PETTRÉ, JULIEN, and THALMANN, DANIEL. “Crowd Patches: Populating Large-scale Virtual Environments for Real-time Applications”. *Proc. of I3D*. I3D '09. Boston, Massachusetts: ACM, 2009, 207–214. ISBN: 978-1-60558-429-4 2.
- [YVN*14] YOO, INNFARN, VANEK, JURAJ, NIZOVTSOVA, MARIA, et al. “Sketching human character animations by composing sequences from large motion database”. *The Visual Computer* 30.2 (2014), 213–227 3.
- [ZLD15] ZHANG, PENG, LIU, HONG, and DING, YAN-HUI. “Crowd simulation based on constrained and controlled group formation”. *The Visual Computer* 31.1 (2015), 5–18 4.