# Realism of Character Representations in Crowd Simulation

**A user study on different representations of simulated moving crowds**

Simon Rosman simon.rosman@gmail.com
Utrecht University

June 26, 2014

In the subject of Path Planning and Crowd Simulation, a wide variety of algorithms and methods are available to simulate virtual characters. The results of these planning algorithms are visualized in a lot of different ways, so called representations. We present an analysis of the different parameters that are in effect in each different representation. An analysis of commonly used representations reveals that there are three main distinctive levels of shapes used to represent a virtual character, i.e. *simple*, *intermediate*, and *complex*. As these representations are to be used by humans, they should allow the human observer to focus on the character paths they are visualizing. We conduct a user study to determine how the realism, observed by humans, is affected by each representation. We test the three levels of representations by using a 2D top-down overview using discs, and 3D perspective views using pawns and articulated bodies. The user study shows that none of the representations is evidently better than others.

## 1. Introduction

For most persons crowds are a part of everyday life, and therefore crowds are a heavily researched subject. In computer science we want to imitate the behavior of real crowds using virtual crowds. For this a plethora of methods are available, such as the well known model of Social Forces for pedestrian dynamics [HM95], the Continuum Crowds method [TCP06], and many others. The purpose of these methods is to plan motion, simulate group behavior, and avoid collisions for a set of virtual characters such that they can move around the environment towards their goals. In this work, we focus on the realism of the paths resulting from such a simulation, i.e. the trajectory of each character. All methods present their results in some way, e.g. the visual style, and since there is no standard for presenting results these can be quite different. Since all method's results are to be shown to humans, we consider it important that the results should be presented in

such way that human observers can focus on the character paths, and can spot potential errors in the paths.
In this work we analyze the parameters that affect the representation of a character. We take a look at crowd simulation-related methods to observe what are commonly used representation. We conducted a user study to determine how the representation affects a human's ability to judge the realism of the character paths. The main question we want to answer is as follows:

*Do different representations of moving crowds affect the level of perceived realism of the character paths for human observers?*

We will provide a definition for the term "realism" in the next section.

### 1.1. What is realism?

As Mori [Mor70] points out, the realism of physical or virtual characters is strongly affected by "the Uncanny Valley". This phenomenon mostly considers the physical properties of characters [Han06], i.e. how much their appearance and limb movement resemble those of real human beings. However, these aspects are mostly influenced by graphics rendering techniques (e.g. lighting and texturing), and 3D models combined with muscle movements [Mac+09] (controlled by kinematics and physics). Since we focus only on the path quality and group behaviour, we do not consider lighting and animation when referring to "realism". The model used to represent characters is part of our subject, albeit only the shape with varying levels of complexity, which we will explain later on.

### 1.2. Subquestions

To provide more insight in the exact parameters we are testing, the main question is split into three parts.

1. *What are the most commonly used representations for moving crowds, and in which case are they being used?*
2. *Do human observers find moving crowds to be more realistic when more sophisticated representations are used?*
3. *Which representation allows for the best judgement of realism? The representation consists of:*
   - *The shape and complexity of a character*
   - *The view and projection of the scenario.*

### 1.3. Hypotheses

1. *We expect that more sophisticated, yet non-photorealistic, representations tend to have the effect of distracting the observer from judging the motion of a crowd.*

*Meanwhile, we expect that overly simplistic representations will not provide enough sense of involvement [Jen+08] and will make the observer more critical.*

2. *We expect that an abstract top-down view will reveal certain phenomenon better, for example lane forming, oscillating waves, or congestion. These phenomenon might be more difficult to spot using perspective view because of occlusion.*

# 2. Related works

The goal of crowd simulation is to simulate movement for a large set of virtual characters as realistically as possible. It requires modeling of pedestrian dynamics, group behaviour motion synthesis, and rendering such that a character's behavior closely resembles that of humans [Guy+10].
Since this is a very broad topic, most methods focus on one or a few aspects, e.g. finding a shortest path, avoiding collisions, etcetera. Some methods do not even consider realism as their goal is just to find any viable solution for a path-finding problem, but these methods mainly focus on the field of robotics. Our goal is to find the best way to present the results of any method and allowing the observer to focus on the quality of the path. All methods show their results in some way, and the more recent methods also provide movies for a more vivid comparison. None of the methods however provide much reason for choosing particular views and representations for these results.
According to Hanson [Han06], if the aesthetics of a representation are right, any level of realism or abstraction can be appealing. Therefore, it is hard to objectively measure the results of a method. The Steerbench framework [Sin+09] uses a set of metrics to measure the quality of a method, such as efficiency and the number of collisions. Wolinski et al. [Wol+14] provides another framework in which methods are compared against real data and a method's parameters are optimized towards the reference data. Again, this method does not take into account the level of perceived realism for observers.
If we want to look at the effect of different representations on the perceived level of realism, there are some links to the thoroughly researched Uncanny Valley [Mor70]. For example, White et al. [WMP07] investigates the role of motions in the Uncanny Valley, but only considers the realism of a character's limb movement instead of the realism of character paths. In our experiment we performed a user study comparing the effect of different representations on the perceived level of realism. We take the important parameters into account, which we will explain in the next sections. Our experiment shows which representations and scenarios are prone to being perceived less realistic and which ones appear more realistically.

## 2.1. Commonly used representations

When we look at the existing crowd simulation methods, we can make a few observations about how the results are usually presented. There are a couple of parameters in effect. First of all there is the **view** that defines the angle and projection type at which a scene is shown. The most common views are Top-down (orthographic projection), and perspective (i.e. a birds-eye view). Next, an important parameter is the **shape** of the characters. There are roughly three levels of abstraction, as shown in Figure 1.

### 2.1.1. Simple

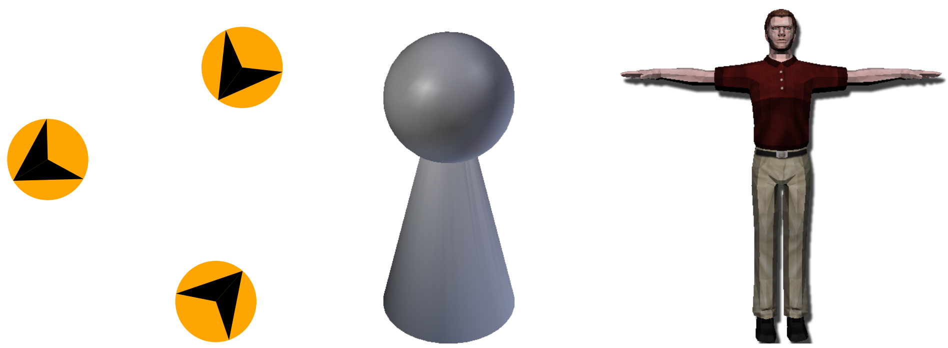
A common way of representing a character is using a *Simple* shape. This especially occurs in the older methods because the available hardware was not capable of creating more complex visualization. Typical examples of *Simple* shapes include Triangles, squares, discs (such as depicted in Figure 1a), arrows, and cylinders [Rey99].
To indicate the direction of movement an arrow might be used. In the case of triangles the triangle might be isosceles, with the smallest angle pointing in the direction of movement.

### 2.1.2. Intermediate

Since *Simple* shapes are hard to be identified or recognized as representing humans, *Intermediate* shapes are also used. Typical *Intermediate* shapes include Pawns (such as shown in Figure 1b), Humanoids, Stick-figures, and Templates. Templates, a.k.a. impostors [ABT00], are pre-rendered images of virtual characters in various positions and viewed from various angles. These shapes are more complex than *Simple* shapes and are less abstract. Their shape might show more resemblance to humans.

### 2.1.3. Complex

The most advanced way of representing virtual characters is using 3D models of humans (such as shown in Figure 1c). The models can either be static (without limb movement), or fully animated using kinematics. Methods may use multiple models to allow for variation in a crowd. Since dynamic templates [ABT98] can also be animated, they are considered to be a *Complex* shape.



(a) *Disk shapes* (b) *Intermediate shape* (c) *3D human models, controlled using kinematics*

Figure 1: *An overview of the different levels of complexity of character representations.*

The realism of a representation is also subject to the scenario which is being displayed. Since the scenario can depict completely different social situations, a human observer might expect different behavior. Some methods display scenarios of special cases, such as evacuations or panicking crowds. However, since we focus on normal behavior for daily situations, we only consider more simple scenarios. In subsection 5.1, we will elaborate on the specific scenarios included in this experiment.
Furthermore, there are many additional representation parameters. Some methods show extra information, such as the active forces per character [Rey99], character velocity, or waypoints. Some methods also show a curve behind each character to indicate the path it has traversed. This is useful for printing as only a single image is required instead of a series of consecutive frames.

| View | Perspective | Shape | Usage | Common focus |
|---|---|---|---|---|
| 2D | Top-Down | Simple | 51% | Path planning |
| 2D | Perspective | - | 0% | - |
| 2D | any | Intermediate | 0% | - |
| 3D | Top-Down | Complex | 15% | Crowd dynamics |
| 3D | Perspective | Complex | 53% | * |
| 3D | any | Intermediate | 12% | * |
| any | Top-Down | any+trajectory curve | 32% | * |

Table 1: *Usage of parameters in existing methods.*
** There is no clear common focus for the methods using this combination of parameters.*

To determine which parameters are important to test, we have analyzed the results from existing methods and created an overview of the most commonly used combinations. This overview is based about 40 papers dating from 1995 till 2014. On average the papers are cited over 250 times. The papers have been selected to focus on Collision avoidance, Crowd Dynamics / Group behavior, Design/ Flow, Evaluation / Benchmarking, Pathplanning, Social behavior, Steering behavior, and Visualization / Rendering. The overview is displayed in Table 1. Please refer to Appendix A for a complete overview of all the analyzed methods.

Finally, the results are also affected by the display of the environment, i.e. whether or not it is textured and the lighting and coloring techniques which are used. As explained in subsection 1.1, we do not focus on those parameters because they do not affect the realism of character paths.

## 3. Method

Our goal is to perform a user study in order to answer the research question. This requires the users to look at, and judge motions of moving crowds, and thus we require

these motions to begin with. Therefore the first step in our approach is creating a *Crowd motion dataset*. In this step, we collected video footage from locations the users might be familiar with, and extracted the motions of the crowds contained in them. Additionally, we extracted the most important features of each location, i.e. the outline of buildings and permanent obstacles. The process of gathering the dataset will be further explained in section 4.
In the next step we created visualizations of scenarios selected from the dataset. In each visualization the paths of each selected character in the scenario is played back using the same representation. Since we want to test different representations (as explained in subsection 2.1, we implemented these representations in two frameworks. The 2D visualizations are creating using the ECM framework [KGO09] and the 3D visualizations are created using Unity3D [Uni]. The selected scenarios and the visualizations are further explained in section 5.
Finally, we conducted a user study in which we showed the chosen combination of different representations to the participants. In section 6 we elaborate on the details of the study.

# 4. Crowd motion dataset

A dataset for realistic crowd movements is established. It consists of real data and is constructed by extracting movements from recorded footage of moving crowds. The character positions extracted from video footage are back-projected into 3D to obtain 2D positions on the ground plane. Since we do not have a detailed knowledge of the ground plane in the video we will assume that the ground plane is flat and level. Additionally, we assume that the camera remains at a fixed position and that the camera intrinsics do not change over time, e.g. the camera will not change its lens focus during the video. We also assume that the camera's lens distortion is neglectable. Furthermore, we assume that the persons in the video footage do not have any disabilities affecting movement, and that they are not carrying or moving large objects that might hinder the surrounding pedestrians. If the video footage contains such cases, they are be excluded from the dataset, i.e. we select only the characters that are not influenced.

The dataset contains 2D trajectories of the path traversed by each individual along with its orientation (i.e. the viewing direction). Each trajectory is described by a series of points of key positions along with a timestamp and an orientation vector (i.e. the direction in which a person is looking).

## 4.1. Path extraction tool

A tool was written to simplify the extraction process. The tool enables the user to play a previously recorded video of crowd footage, and identy the path of each person in the video. Each path consists of a series of locations along with a timestamp, and an orientation.
Since the input consists of 2D-frames, we must perform backprojection into 3D to obtain the original coordinates for each location. We will however first explain the process of

forward projection since this principle is used by the camera to create the video. Having a basic understanding of this principle will help in understanding the reverse process.
The equation for converting a 3D point $\vec{P_w} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ into a 2D pixel $\vec{P_i}$ is [BK08] [Gla89] [Zha00]

$$\vec{P_i} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}(\vec{P_w}\mathbf{R} + \vec{t}), \tag{1}$$

where $\vec{P_i}$ denotes a pixel in a video frame, $\vec{P_w}$ denotes the corresponding 3D point, $\mathbf{K}$ denotes the camera-intrinsics matrix, $\mathbf{R}$ the rotation matrix, and $\vec{t}$ the translation vector of the camera from the world origin. Note that because we are using homogeneous coordinates, the 3D point $\vec{P_w}$ has a fourth component, and the 2D point $\vec{P_i}$ has a third component, which are set to 1. These components are dropped once the calculation is finished. The subscripts $i$, $w$, and $c$ are used to indicate in which space a point is located, where $i$ denotes the 2D image, $c$ the camera plane in 3D, and $w$ the world.
The combination of $\mathbf{R}$ and $\vec{t}$ results in a rotation and translation matrix:

$$\begin{bmatrix} \mathbf{R} & \vec{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The camera intrinsics are set up as follows:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $f_x$ and $f_y$ describe the camera's focal length in pixel units, and $c_x$ and $c_y$ determine the principal point of origin in the 2D image.
Since we do not know any of these matrices, we must start by determining the **camera calibration** for the chosen video, i.e. solving $\mathbf{K}$ and $\mathbf{R}$ and $\vec{t}$. The QtCalib tool [ZL12] allows doing this in an easy way by loading in a single frame of the video and clicking on 4 points that form a rectangle on the ground plane, shown in Figure 2. The camera configuration is then solved using Tsai's [Tsa87] method.
Since we want to do the reverse operation, and project the 2D screen pixel onto a 3D point on the ground plane, the equation must be inversed. Ideally, this would result in the following equation:

$$\vec{P_w} = \vec{P_i}\mathbf{K}^{-1}\mathbf{R}^{-1} - \vec{t}. \tag{4}$$

However, since we don't know the exact depth of the point at pixel $\vec{P_i}$, this will produce a result where the resulting point $\vec{P_w}$ will not lie on the ground plane, but on the camera plane. Therefore, we will use ray-tracing to find the intended point. This is similar to shooting a ray from the camera origin $\vec{t}$ through the pixel $\vec{P_i}$, and intersect it with the plane $z = 0$.

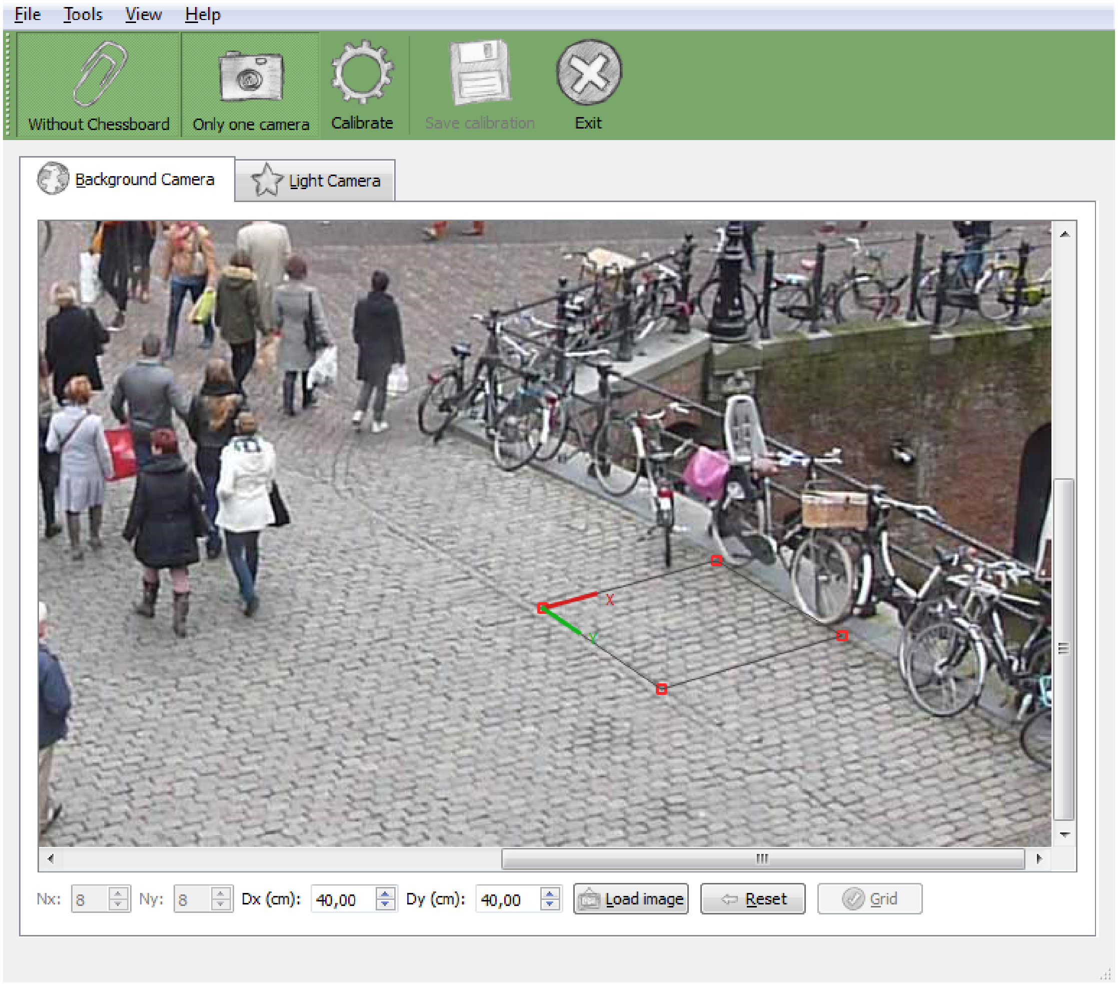



Figure 2: *QTCalib - Tsai [Tsa87] calibration tool*

First we require the inverse of both matrices **K** and **R**.
The inverse of matrices **K** and **R** are easy to obtain by inserting an inversed $f_x$, $f_y$, $c_x$, and $c_y$, such that

$$\mathbf{K}^{-1} = \begin{bmatrix} -f_x & 0 & -c_x \\ 0 & -f_y & -c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

and

$$\mathbf{R}^{-1} = \mathbf{R}^T \tag{6}$$

since an inverse of a rotation matrix is equal to its transpose.

Now we can compute the back-projection of 2D image point $\vec{P_i}$ into the 3D world point $\vec{P_w}$. First, the 2D image coordinates $\vec{P_i}$, in pixel units, are converted into camera plane coordinates $\vec{P_c}$ as follows:

$$\vec{P_c} = \vec{P_i}\mathbf{K}^{-1} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \mathbf{K}^{-1}. \tag{7}$$

Next, the camera plane coordinates $\vec{P_c}$ are rotated into world coordinates. This results in a 3D vector $\vec{r}$ describing the ray we are tracing, i.e.

$$\vec{r} = \vec{P_c}\mathbf{R}^{-1}. \tag{8}$$

We then have to determine the depth $\alpha$, which represents the length of the ray from the camera location to the ground plane:

$$\alpha = \frac{t_z}{r_z}. \tag{9}$$

Finally, we can compute $\vec{P_w}$'s $x$ and $y$ world coordinates of the pixel on the ground plane.

$$\vec{P_w} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = \begin{bmatrix} t_x - \alpha \times r_x \\ t_y - \alpha \times r_y \\ 0 \end{bmatrix}. \tag{10}$$

We use this process of backprojection to convert mouse positions in the 2D video into 3D positions in the original environment on the ground plane. We can then play the video and indicate key positions (and orientations) for a character, and store the resulting 3D positions, along with the timestamp of the position. This results in a series of points describing the path the character has traversed, and a series of vectors indicating the direction in which the character is looking at that position.
Additionally, we can use this process to indicate the position and outline of buildings, obstacles, and the type of terrain. These are stored as sets of polygons.

# 5. Experiment setup

## 5.1. Scenarios

To test our hypothesis for a broad range of crowd movements, a set of common scenarios has been chosen. The SteerBench framework [Sin+09] describes a large set of possible benchmark scenarios, and we have selected a subset of scenarios, or similar scenarios, occurring frequently in everyday situations in public. Since most crowd simulation problems consider deadlocks and collisions [Sin+09], the scenarios must also contain such problems. These problems occur in situations where there are e.g. intersecting streets, in an area where the direction of movement is not restricted, meaning that there are no directional lanes or rules restricting movement and thus characters should handle dynamic obstacle-avoidance themselves.
The scenarios are chosen such that they contain these parameters. The scenarios included in the experiment are shown in Figure 3. We will now describe each individual scenario and explain their significance and properties. In the next subsections, we will elaborate on the different shapes and views that are used in the experiment.

### 5.1.1. Oncoming (2 agents)

The most common basic principle in everyday situations is to avoid another person from an opposing direction. Therefore in this first scenario, two individual characters walk past each other in opposing directions. This scenario is set in an open environment where the characters will have enough room to avoid one another without interfering with surrounding characters. Another property of this scenario is that there is a clear line of sight between the characters, i.e. that they can see the other character approaching, and that is does not appear suddenly.

### 5.1.2. Crossing (2 agents)

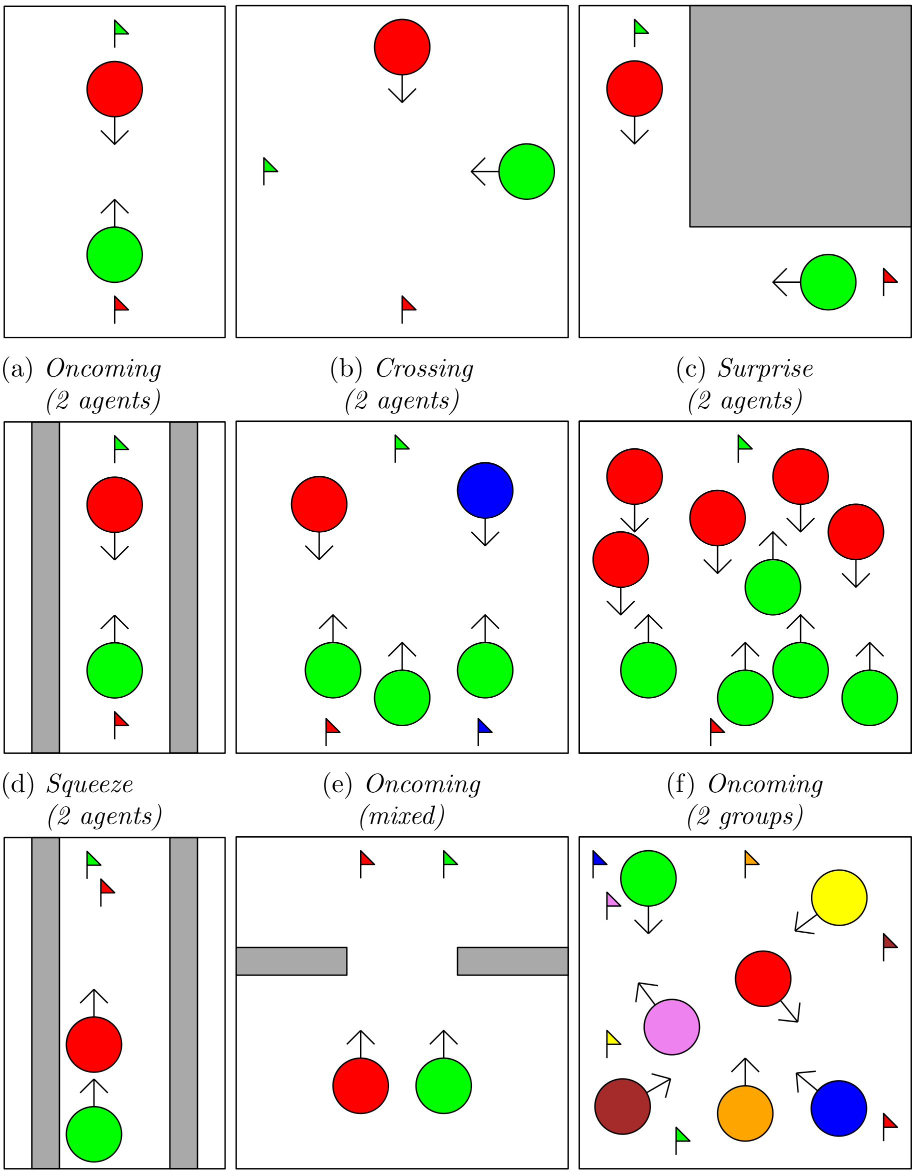
In this second scenario, the paths of two characters intersect. Therefore, the characters will have to anticipate the other character's movement and plan its own path such that collisions are avoided. This scenario is also set in an open environment.

### 5.1.3. Surprise (2 agents)

In some situations a person cannot see another oncoming person approaching because this person is around a corner, or behind an obstacle. Since this reduces the amount of time available to avoid the other person, the resulting avoiding behavior is different from a regular oncoming scenario. In some cases the two persons might even briefly collide.

### 5.1.4. Squeeze (2 agents)

This scenario is important as it forces two persons to walk closer two each other than they might prefer. In this squeeze scenario, this is behavior is caused by an environment



(a) *Oncoming (2 agents)*
(b) *Crossing (2 agents)*
(c) *Surprise (2 agents)*
(d) *Squeeze (2 agents)*
(e) *Oncoming (mixed)*
(f) *Oncoming (2 groups)*
(g) *Overtake (2 agents)*
(h) *Doorway (2 agents)*
(i) *Random (multiple agents)*

Figure 3: *An overview of the different scenarios*

having walls or obstacles that form a corridor. Additional to the persons keeping less distance from each other, they might also keep closer to the walls of the corridor.

### 5.1.5. Oncoming (mixed)

It is important to include scenarios containing groups since group behavior strongly affects the movement of individual persons. In this scenario, we start with a basic group scenario in which a group encounters one or more individual persons from an opposing direction. We define a group as 2 or more people having a close social connection, e.g. a family, some friends, or a couple. The group members will show coherent movement behavior and might favor staying close together.

### 5.1.6. Oncoming (2 groups)

This second group scenario is somewhat more complex because two groups are passing each other in opposing direction. Since groups tend to occupy a wide space, it will be difficult for both groups to pass each other without adjusting their formation.

### 5.1.7. Overtake (2 agents)

In this scenario the preferred walking speed of one character (the green character in Figure 3d) is much higher than the other character. Therefore, the first character will try to overtake the second one. This might not be possible instantly when there is not enough room.

### 5.1.8. Doorway (2 agents)

Doors are common obstacles in daily situations. In public situations it also occurs that these are simultaneously used by multiple persons. In case of most doors, e.g. sliding doors, or shop-doorways, people have to squeeze to pass through simultaneously, or they will have to wait briefly. Since this strongly affects the planning of movement and anticipation, this scenario is included in the test.

### 5.1.9. Random (multiple agents)

Finally, we have included a scenario which covers movement behavior of a large set of characters. The purpose of this scenario is to provide a broader sense of crowd interactions as the behavior is less specified, and the number of characters is larger. For this scenario, we will use data from an open environment, i.e. that there are no walls or large obstacles limiting movements. This scenario can also include occurrences of the other scenarios.

## 5.2. Visualizing using different representations

The scenario(s) from the dataset are visualized using the ECM/IRM framework [KGO09] and Unity3D [Uni]. Figure 1 (subsection 2.1) shows visualizations of the shapes that will be used.

### 5.2.1. Simple

For the visualizations that use a simple shape, the ECM framework [KGO09] provides a suitable implementation. It consists of a circle with an arrow contained inside (as shown in Figure 1a). In our implementation the arrow is used to indicate the direction of movement, i.e. the direction in which a character is walking. The shape is assigned a color to indicate whether a character belongs to a certain group, e.g. if a couple is present in a scenario they are assigned the same color. Characters that do not belong to a specific group are assigned a neutral color, for which we use grey.

### 5.2.2. Intermediate

This shape is implemented using a 3D model of a pawn (as shown in Figure 1b) to represent a character. The pawn is modeled using a 3D editing tool, and imported into Unity3D [Uni]. The pawn is assigned a color corresponding to the group to which a character belongs.

### 5.2.3. Complex

This shape is implemented using various 3D models of humans (such as the example shown in Figure 1c). A set of motion-capture animations was used to animate the 3D models. The set contained an animation clip for walking, a clip for standing idle, and clips to blend between walking and standing idle. For each character, the velocity is computed at each frame, this is converted into an animation speed such that the velocity matches the displayed walking animation. This way, artifacts such as foot-sliding are kept to a minimum.

## 5.3. Views

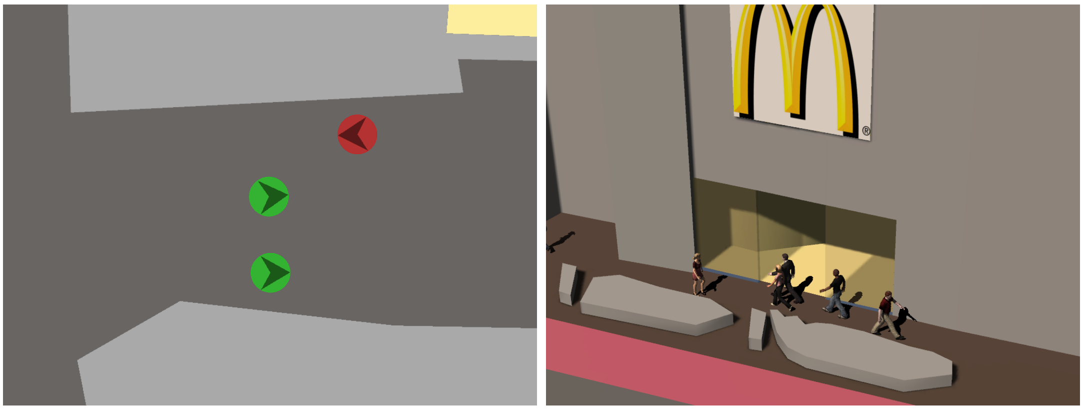
The third parameter of the test will be the view at which scenes will be displayed to the user. We will use the most commonly used views, as shown in Chapter 2: A top-down view, and a perspective view and projection. In either case, the camera will keep still during the test.

### 5.3.1. Top-down view

This view is used to provide the viewer with an abstract view of a scenario. The scenario is rendered using orthographic projection, i.e. the ground plane is viewed from a right angle, and projected using affine transformations (as depicted in Figure 4a). The camera is positioned and zoomed such that all characters of the active scenario will be visible during the entire scenario. The top down-view orthographic view is only used for visualizations that use *Simple* shapes.

#### 5.3.2. Perspective view

With this view we try to provide the user with a view that is similar to a real-life view of the same environment, i.e. as if the user was standing in a position similar to the position of the camera used to record the dataset of the current scenario. To enable this we use a perspective view and projection (as depicted in Figure 4b) for rendering the scenario, which is done in Unity3D [Uni]. In the perspective view, the model of the environment, i.e. the outline of buildings and obstacles (as explained in section 4), is edited using a 3D editor. The outline of buildings are extruded and assigned a height similar to the original buildings. Additionally, doorways are modeled in the buildings used for the Doorway scenario (see subsubsection 5.1.8). Obstacles are also assigned a height. The Perspective view is not used for visualizations using *Simple* shapes since those are only 2D.



(a) *Top-down view* (b) *Perspective view*

Figure 4: *An overview of the different views used in the used to display the scenarios.*

## 6. User study setup

Each participant is shown a series of short clips sampled from the visualized dataset. Since we use 9 scenarios, and 3 shapes with 2 views, this results in a set of $9 \times 3 = 27$ combinations. Since the shape used for a scenario also controls the view, the view parameter is not tested separately. To reduce the possibility of introducing bias from using a single data sample for each scenario, we extracted multiple copies of the same scenario from the dataset whenever available. This results in 51 sample clips in total.

Each combination is repeated over 3 trials. This allows us to take the average of these three measures and it helps reducing bias. Additionally, all clips are shown in a random order to further reduce bias. This leads to a set of 153 clips. Since each clip has a length of approximately 5-10 seconds, each session took approximately 30 minutes.

Each participant is carefully instructed to try to focus on the paths (i.e. the trajectories) the characters in the sample clip choose. After watching each sample clip, the participant grades the clip on a 5-point Likert scale based on how realistically he or she thought the paths in the clip are.

The participants are not given any clues to what the origin of the data is, i.e. whether the clips are based on algorithms or on real data. They do receive information beforehand about the three representations (as explained in subsection 5.2) that will be shown consisting of a brief description of the Simple, Intermediate, and Complex shape along with one example figure per shape, to allow them to prepare themselves for the kind of images they will be viewing.

Per participant, the following additional data is gathered: *Age*, *Sex*, *Familiarity* with virtual characters (none, medium, high), and the *Top three* of their favorite games. The latter question is included enable distinguishing gamers and non-gamers. Additionally, the participants are asked to comment on things they noticed in the sample clips and comments in general (both optional). Twenty persons participated in the user study.

# 7. Analysis

We are analyzing the data using method commonly used in psychological, social, medical, and other user studies. This is because the realism we are testing is subjective and susceptible to human bias. Therefore we must perform a series of checks on our data before we can search for significant differences regarding our research questions. First we will determine whether the data we gathered from our participants is reliable, i.e. whether all three measures of the same combination of factors are graded consistently. Next, we will test the sphericity of the data, which checks whether the variances of the differences between conditions are roughly equal. After accounting for these two issues, we will test for significance in the differences between the conditions. We will use an ANOVA repeated-measures analysis design for this, as described in *Discovering statistics using SPSS* [Fie09, p. 482-502].

## 7.1. Data reliability

We are using three measures for each combination of a *Representation* (subsection 5.2) and a *Scenario* (subsection 5.1) so that we use the averages of all participants to compare the averages of other combinations. However, the averages will only provide a realistic representation of all the measures if each question measures the same concept. We use Cronbach's $\alpha$ [Cro51] to test whether this is the case.

### 7.1.1. Testing reliability

Cronbach's $\alpha$ calculates the internal consistency of a set of data, resulting in a single number in the range $[-\infty, 1]$. If the Cronbach's $\alpha$ of a set of data is high, it is more likely that a participant will get the same results when he or she retakes the test. A commonly accepted threshold for a reliable dataset is established at $\alpha \geq 0.7$ [NBB67].

| Scenario | Simple | Intermediate | Complex |
|---|---|---|---|
| 1 | .822 | .807 | .719 |
| 2 | .900 | .755 | .405 |
| 3 | .519 | .880 | -.046 |
| 4 | .870 | .812 | .809 |
| 5 | .672 | .366 | .841 |
| 6 | .580 | .748 | .467 |
| 7 | .485 | .790 | .675 |
| 8 | .633 | .713 | .840 |
| 9 | .649 | .439 | .661 |
| 10 | .615 | .666 | .863 |
| 11 | .807 | .640 | .723 |
| 12 | .857 | .659 | .652 |
| 13 | .757 | .520 | .678 |
| 14 | .577 | .865 | .878 |
| 15 | .774 | .450 | .745 |
| 16 | .689 | .520 | .752 |
| 17 | .718 | .669 | .544 |
| Average | .713 | .651 | .703 |
| Standard deviation | .122 | .152 | .219 |

Table 2: *Cronbach's $\alpha$ values of all the 51 combinations of a Representation and a Scenario. Values below* 0.7 *are highlighted.*

We compute the $\alpha$ value for each combination of factors, i.e. for the three measures of a combination of factors for all participants. The results are displayed in Table 2. Unfortunately, more than half of all the $\alpha$'s are below the 0.7 threshold, thus making it appear as if our data is completely unreliable. Therefore we cannot continue using this data straightforward. We will now explain the actions we performed to improve the reliability of our data.

#### 7.1.2. Improving reliability

The easiest way to move forward is to lower the threshold for the reliability. However, since our data has some quite low $\alpha$ values, this is insufficient. A common method is to exclude measures from data set, and check whether this improves the consistency. We checked this for our data, and when the original $\alpha$ is below 0.8, and increases when a measure is deleted, we remove it from our dataset. This results in the Cronbach's $\alpha$ values as displayed in Table 3.

#### 7.1.3. Accepting reliability

As we can observe from the $\alpha$ values in Table 3, the ideal Cronbach's $\alpha$ is still not met for all combinations. However, we can still accept the results as being reliable ,for a couple

| Scenario | Simple | Intermediate | Complex |
|---|---|---|---|
| 1 | .822 | .807 | .799 |
| 2 | .900 | .866 | .537 |
| 3 | .717 | .880 | .623 |
| 4 | .870 | .812 | .809 |
| 5 | .672 | .463 | .841 |
| 6 | .580 | .849 | .648 |
| 7 | .485 | .818 | .675 |
| 8 | .633 | .753 | .840 |
| 9 | .649 | .681 | .754 |
| 10 | .615 | .666 | .863 |
| 11 | .807 | .640 | .788 |
| 12 | .857 | .694 | .809 |
| 13 | .860 | .551 | .719 |
| 14 | .711 | .865 | .878 |
| 15 | .804 | .519 | .745 |
| 16 | .689 | .520 | .752 |
| 17 | .718 | .669 | .567 |
| Average | .729 | .709 | .744 |
| Standard deviation | .113 | .132 | .099 |

Table 3: *Improved Cronbach's $\alpha$ values of all the 51 combinations of a Representation and a Scenario. Values below* 0.5 *are highlighted.*

of reasons. Most importantly, Cronbach's $\alpha$ is dependent on the number of items in the scale ($N$) [Cor93]. The numerator of the Cronbach's $\alpha$-equation contains $N^2$. Since we have three measures per combination our $N = 3$, which makes it more prone to scoring low. Secondly, after the improvements we made in subsubsection 7.1.2, not only did our average reliability increase, but the spread is reduced (as shown in Table 4. Finally, Kline [Kli13] argues that "$\alpha$ values below 0.7 can still be considered to be reliable when dealing with psychological constructs because of the diversity of the constructs being measured" [Fie09, p.675]. Table 5 provides an overview of different thresholds an the number of $\alpha$'s failing to meet the threshold and the percentage of $\alpha$'s that does meet the criterion. The two combinations failing to meet the lowest threshold do not share the same factor, i.e. they have a different *Scenario* and *Representation*. Now that we have proved the reliability of our data, we can continue using our data by taking averages of the three measures for each combination.

## 7.2. Testing sphericity

As a final check before we can analyze our data is that we have to check the Sphericity of our data. This is required because we are using a repeated-measures design. The Sphericity is tested using Mauchly's Test of Sphericity [Fie09, p. 460], which checks

| | | Simple | Intermediate | Complex |
|---|---|---|---|---|
| Original | AVG | .713 | .651 | .703 |
| Improved | AVG | .729 | .709 | .744 |
| Original | STDEV | .122 | .152 | .219 |
| Improved | STDEV | .113 | .132 | .099 |

Table 4: *Averages and Standard deviations of the Cronbach's $\alpha$ values compared before and after improvements.*

| | *threshold* $\alpha$ | $\alpha's <$ *threshold* | $\alpha's \geq$ *threshold* |
|---|---|---|---|
| Original | .7 | 26 | 49% |
| Improved | .7 | 21 | 59% |
| Original | .6 | 13 | 75% |
| Improved | .6 | 8 | 84% |
| Original | .5 | 7 | 86% |
| Improved | .5 | 2 | 96% |

Table 5: *Overview of Cronbach's $\alpha$ values tested against several thresholds. The total number of $N$ is* 51*.*

whether the variances between the different measures of each factor are roughly equal. Since each participant grades all combinations his or her grading of a combination that is shown later can be influenced by a combination shown earlier. If the assumption of Sphericity doesn't hold, we must be more strict when testing for significant differences between our conditions.
Mauchly's test indicated that the assumption of sphericity had been violated for the main effects of *Scenario*, $\varepsilon = .031$, $p < .5$, and *Representation*, $\varepsilon = .016$, $p < .5$. Therefore the degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity [Fie09, p. 461].

### 7.3. Testing significant differences

We test for significance in the effect using ANOVA [Fie09, p. 482-502]. In this test, the different *Scenarios* (subsection 5.1) and *Representations* (subsection 5.2) are compared to determine whether a significant difference in *Realism* (subsection 1.1) exists. We set the significance level to $p < .05$.

There was a significant main effect of the type of *Scenario*, $p = .000$, but no significant effect of the type of *Representation*, $p = .265$. Additionally, a significant interaction effect between *Scenario* and *Representation* was found, $p = .000$. The latter indicates that for some scenarios there is a significant difference in *Realism* between some of the *Representations*. An overview of the *Realism* per *Scenario* and *Representation* is plotted in Figure 5. Please refer to Table 6 for the names of the *Scenarios*.

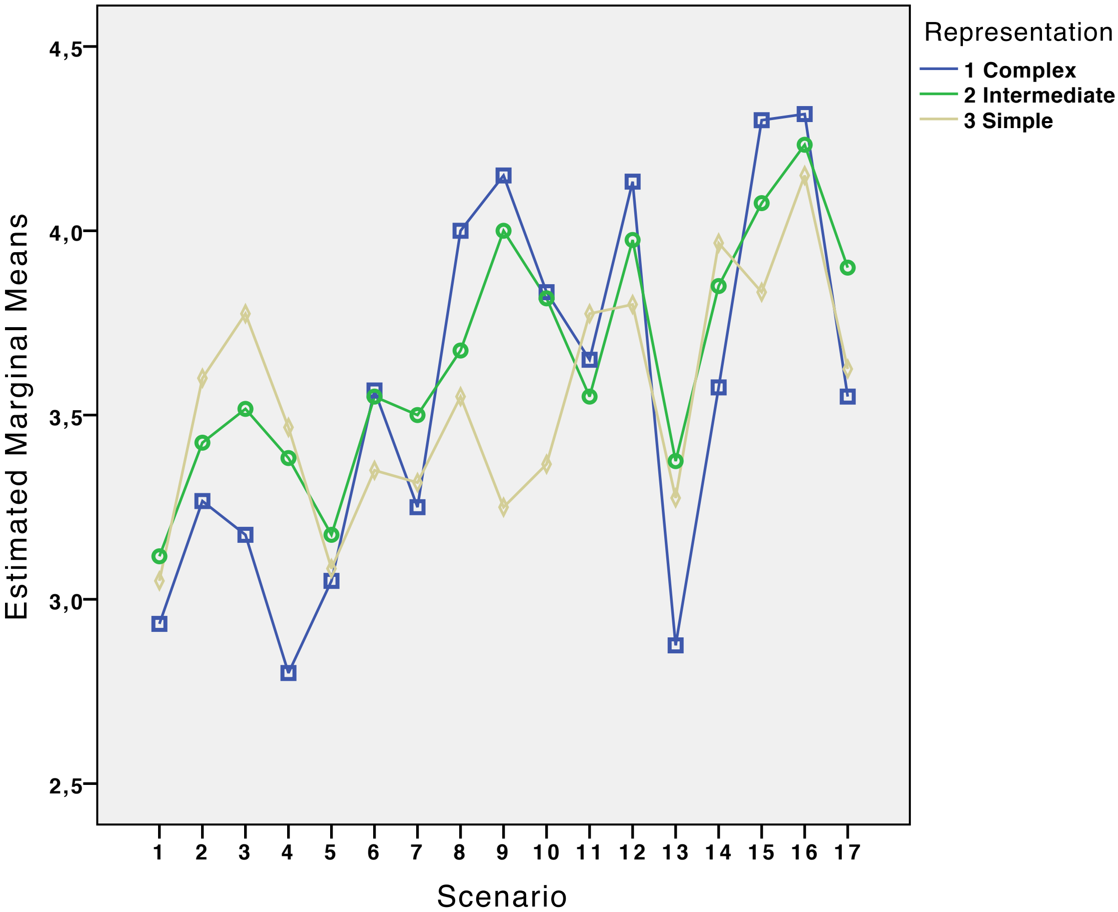



Figure 5: *Plot of the average Realism per combination of Scenario and Representation*

| Scenario | Name |
|---|---|
| 1 | crossing_2agents1 |
| 2 | crossing_2agents2 |
| 3 | doorway1 |
| 4 | doorway2 |
| 5 | oncoming_2agents1 |
| 6 | oncoming_2agents2 |
| 7 | oncoming_2agents3 |
| 8 | oncoming_2groups1 |
| 9 | oncoming_mixed1 |
| 10 | oncoming_mixed2 |
| 11 | overtake_2agents1 |
| 12 | overtake_2agents2 |
| 13 | random1 |
| 14 | random2 |
| 15 | squeeze1 |
| 16 | squeeze2 |
| 17 | surprise1 |

Table 6: *A list of the scenario numbers and their corresponding names. The scenarios are sorted alphabetically.*

### 7.3.1. Problem cases

While processing the statistics we noticed a few issues with some scenarios. The *doorway1* and *doorway2* scenarios (subsubsection 5.1.8) score very different in comparison, especially when the *Complex* representation is used. The same issue also occurs with the *random1* and *random2* scenarios (subsubsection 5.1.9), where *random1* scores much lower. The questionnaire that the participants filled in revealed the source of the problem. It turns out that the implementation of the *doorway* scenarios is incomplete. The original environment contains an automated sliding door, causing people to wait for the door to open, and causing them to enter the door through the opening in the middle. The sliding door was not implemented in visualizations. Our participants noticed this, e.g. “Entering buildings seemed to be hard”, and “People always entered through the door exactly in the middle and did not overtake each other, even though the doorway was wide enough”. In the *random1* scenario the problem had a different cause; a character in front of the camera rotated very quickly, causing the animation to look weird. Figure 6 shows screenshots of the involved scenarios. To make sure these problematic cases do not falsely affect the results of our analysis, we have also performed the analysis without these cases.

(a) *Doorway1* (b) *Random1*, $t = 1.9$ *seconds* (c) *Random1*, $t = 2.0$ *seconds*

Figure 6: *Screenshots of the problematic scenarios. The red character in (b) and (c) has a weird animation* ($180°$ *rotation in very short time).*

#### 7.3.2. Analysis excluding problem cases

We have performed the same ANOVA test as in subsection 7.3, but with *Scenarios* 3, 4, and 13 excluded, i.e. *doorway1*, *doorway2*, and *random1*. The *Scenario* factor still shows a significant main effect, $p = .000$, as does the interaction effect between *Scenario* and *Representation*, $p = .002$. The effect of the type of *Representation* is still insignificant, $p = .157$, although it shows a slight increase (Originally: $p = .265$; Problems excluded: $p = .157$). Since the problematic cases do not significantly affect our results, we can keep on using all the tested scenarios.

### 7.4. Observations

Based on the ANOVA tests, we can observe that there is no significant difference in *Realism* for the different *Representations*. We can also observe that a difference in *Realism* does exist for the different *Scenarios*. This becomes clear when we look at Figure 7, where we can see that the *Scenarios* 15 and 16 (subsubsection 5.1.4 score quite higher in *Realism* than almost all other scenarios. Although there is no significant difference for the three *Representations*, we can still make some cautious observations. Table 7 shows the *Realism* averaged per *Representation*. This data suggests that the Intermediate representation scores slightly higher than the others, and that the Simple representation scores the lowest. Additionally, we can observe that it appears as though subsubsection 5.2.3 representations reveal navigation issues the quickest. This observation is also backup up by the data in Figure 5 and by the comments of the participants of the user study. Again, we must point out that because of lacking significance we can only provide cautious observations and no hard conclusions for this

| | Simple | Intermediate | Complex |
|---|---|---|---|
| Average realism | 3.543 | 3.654 | 3.554 |

Table 7: *Average Realism of the different Representations.*

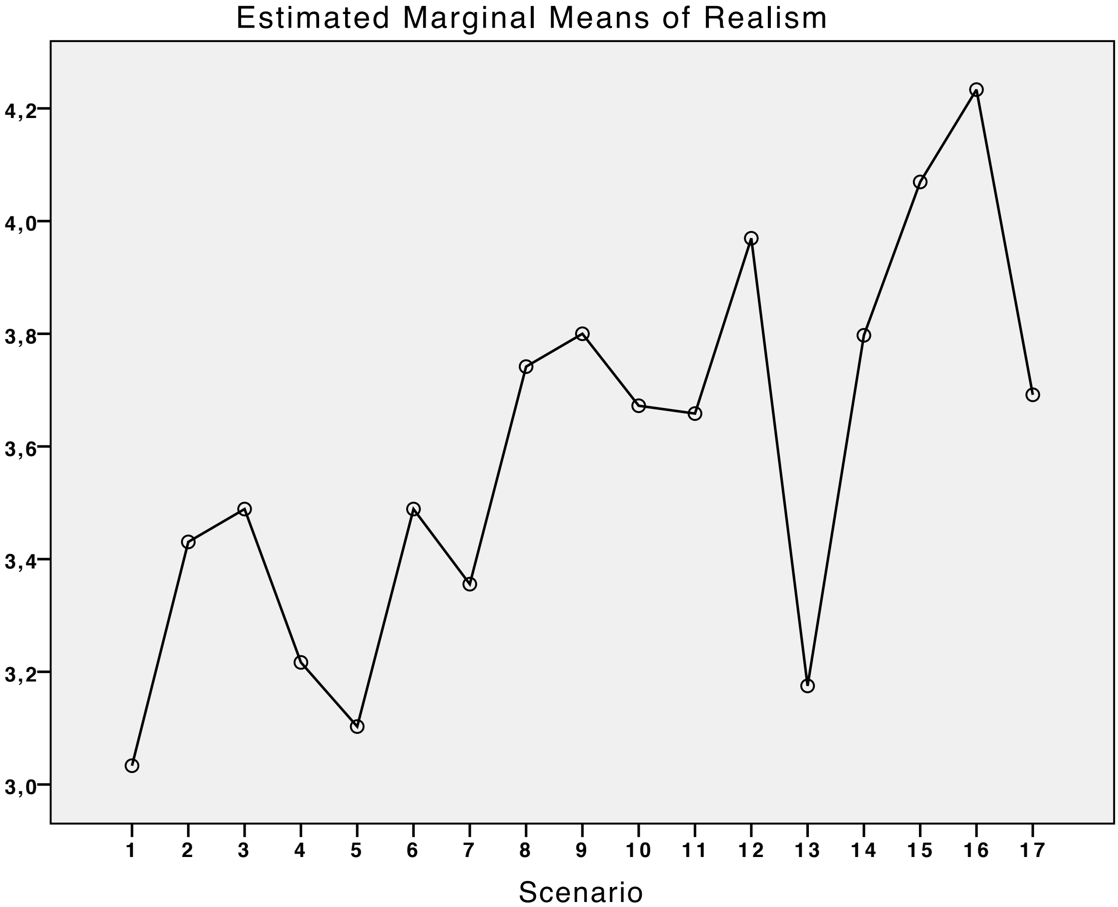



Figure 7: *Plot of the average Realism per Scenario. Please refer to Table 6 for the names of the Scenarios.*

# 8. Discussion & Future work

We have performed a user study to determine whether or not there is a difference in *Realism* when different types of *Representations* are used. We have used different *Scenarios* to eliminate bias possibly introduced by a specific scenario. The study showed that there is no significant difference between the *Realism* for different *Representations*, although it suggests that an Intermediate representation (Section 5.2.2) scores slightly higher than the others.

Most importantly, the study shows that there is a significant difference in *Realism* between the *Scenarios* that are being used. From this we can conclude that some scenarios satisfy human observers more easily than others. Furthermore, it appears as though errors in navigation are most easily spotted when a Complex representation (Section 5.2.3) is used.

These results have some implications for the field of Path Planning and Crowd Simulation, i.e. whenever a new method is being researched, it must be tested against lots of different scenarios, and optionally using different representations.

Our research also has some limitations. First of all, the set of Crowd motion data was extracted by hand (section 4), and is therefore prone to human error. Furthermore we had to limit our scope to a small set of representations. There are many additional parameters (subsection 2.1) that can be used, e.g. showing velocity information per character, or showing a curve of the traversed path. However, they exponentially increase the combinations that must be tested. Therefore we chose the most commonly-used parameters for our implementation.

As future work we would like to research whether it is useful to test more parameters. This would involve a more complex statistical scheme as different groups of participants will have to be formed and compared. When more parameters are tested, a session would become too long for a participant if all combinations were to be tested. Furthermore we would like to fix the flaws in our implementation (subsubsection 7.3.1), and have more participants participate in the study. Perhaps we will find significance for the differences in the *Representations* after all.

Additionally, it would be interesting to benchmark the Crowd motion dataset using a synthetic benchmark, and to compare it to other Path planning methods. The Benchmarking could be done e.g. using Steerbench [Sin+09], or using Wolinski's evaluation framework. [Wol+14].

## References


[ABT98] A. Aubel, R. Boulic, and D. Thalmann. "Animated impostors for real-time display of numerous virtual humans". In: *Virtual Worlds*. Springer. 1998, pp. 14–28.

[ABT00] A. Aubel, R. Boulic, and D. Thalmann. "Real-time display of virtual humans: levels of details and impostors". In: *Circuits and Systems for Video Technology, IEEE Transactions on* 10.2 (2000), pp. 207–217.

[BK08] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly, 2008.

[Cor93] J. M. Cortina. "What is coefficient alpha? An examination of theory and applications." In: *Journal of applied psychology* 78.1 (1993), p. 98.

[Cro51] L. J. Cronbach. "Coefficient alpha and the internal structure of tests". In: *psychometrika* 16.3 (1951), pp. 297–334.

[Fie09] A. Field. *Discovering statistics using SPSS*. Sage publications, 2009.

[Gla89] A. Glassner. *An introduction to ray tracing*. Morgan Kaufmann, 1989.

[Guy+10] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha. "Pledestrians: a least-effort approach to crowd simulation". In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association. 2010, pp. 119–128.

[Han06] D. Hanson. "Exploring the aesthetic range for humanoid robots". In: *Proceedings of the ICCS/CogSci-2006 long symposium: Toward social mechanisms of android science*. 2006, pp. 39–42.

[HM95] D. Helbing and P. Molnar. "Social force model for pedestrian dynamics". In: *Physical review E* 51.5 (1995), p. 4282.

[Jen+08] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton. "Measuring and defining the experience of immersion in games". In: *International journaltitle of human-computer studies* 66.9 (2008), pp. 641–661.

[KGO09] I. Karamouzas, R. Geraerts, and M. Overmars. "Indicative routes for path planning and crowd simulation". In: *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM. 2009, pp. 113–120. URL: http://www.staff.science.uu.nl/~gerae101/motion_planning/irm.html.

[Kli13] P. Kline. *Handbook of psychological testing*. Routledge, 2013.

[Mac+09] K. F. MacDorman, R. D. Green, C.-C. Ho, and C. T. Koch. "Too real for comfort? Uncanny responses to computer generated faces". In: *Computers in Human Behavior* 25.3 (2009), pp. 695–710.

[Mor70] M. Mori. "The uncanny valley". In: *Energy* 7.4 (1970), pp. 33–35.

[NBB67] J. C. Nunnally, I. H. Bernstein, and J. M. t. Berge. *Psychometric theory*. Vol. 226. McGraw-Hill New York, 1967.

[Rey99] C. W. Reynolds. "Steering behaviors for autonomous characters". In: *Game developers conference*. Vol. 1999. 1999, pp. 763–782.

[Sin+09] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. "SteerBench: a benchmark suite for evaluating steering behaviors". In: *Computer Animation and Virtual Worlds* 20.5-6 (2009), pp. 533–548.

[TCP06] A. Treuille, S. Cooper, and Z. Popović. "Continuum crowds". In: *ACM Transactions on Graphics (TOG)*. Vol. 25. 3. ACM. 2006, pp. 1160–1168.

[Tsa87] R. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". In: *Robotics and Automation, IEEE journaltitle of* 3.4 (1987), pp. 323–344.

[Uni] *Unity - Game Engine, tools and multiplatform*. June 2005. URL: http://unity3d.com/unity.

[WMP07] G. White, L. McKay, and F. Pollick. "Motion and the uncanny valley". In: *journaltitle of Vision* 7.9 (2007), pp. 477–477.

[Wol+14] D. Wolinski, S J Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettré. "Parameter estimation and comparative evaluation of crowd simulations". In: *Computer Graphics Forum*. Vol. 33. 2. Wiley Online Library. 2014, pp. 303–312. URL: http://gamma.cs.unc.edu/CParameter/.

[ZL12] A. Zang and D. Lucio. *QtCalib Tsai Calibration Tool*. [Online; accessed 13-January-2014]. Aug. 2012. URL: http://w3.impa.br/~zang/qtcalib/.

[Zha00] Z. Zhang. "A flexible new technique for camera calibration". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.11 (2000), pp. 1330–1334.

# A. Common Representations

Table 8 provides an overview of the methods that were analyzed for compiling a set of commonly used parameters.

| Method | Representation | View | Projection | Shape | Method's focus |
|---|---|---|---|---|---|
| Social force model for pedestrian dynamics [HM95] | Simple | 2D | Top-down | Disc | Steering behavior |
| Probabilistic roadmaps for path planning in high-dimensional configuration spaces [Kav+96] | Simple | 2D | Top-down | Disc; Curve | Pathplanning |
| Steering behaviors for autonomous characters [Rey99] | Simple | 2D | Top-down | Arrow | Steering behavior |
| RRT-Connect [KL00] | Simple | 2D | Top-down | Square; Curve | Pathplanning |
| Hierachical model for Real time simulation of virtual human crowds [RMT01] | Complex | 3D | Top-down; Eye-level; Perspective | Articulated body | Social behavior |
| Crowd simulation for interactive virtual environments and VR training systems [UT01] | Complex | 3D | Perspective | Articulated body | Social behavior; Emergency |
| Visualizing crowds in real time [TLC02] | Intermediate | 3D | Perspective | Impostor | Visualization; Rendering |
| Modeling individual behaviors in crowd simulation [Bra+03] | Simple | 2D; 3D | Top-down; Perspective | Colored Disc; Cube | Behavior; Efficiency; Evacuation |
| The flow of human crowds [Hug03] | Simple | 2D | Top-down | Curve | Crowd flow dynamics |
| A Comparative Study of Probabilistic Roadmap Planners [GO04] | Simplistic | 3D | Perspective | Curve | Pathplanning |
| A physically-based particle model for emergent crowd behaviors [Hei+03] | Simple; Intermediate | 2D; 3D | Top-down; Eye-level; Perspective | Disc; Impostor | Steering behavior; Crowd dynamics |
| Flow tiles [Che04] | Complex | 3D | Perspective | Articulated body | Design; Flow |
| Finding paths for coherent groups using clearance [KO04] | Simple | 2D; 3D | Top-down; Perspective | Disc; Cylinder | Pathplanning |
| Scalable behaviors for crowd simulation [SGC04] | Simple; Complex | 2D; 3D | Top-down; Perspective | Icon; Articulated body | Social behavior |
| Survey of real-time rendering techniques for crowds [RD05] | Complex | 3D | Perspective | Impostor | Visualization; Rendering |
| Geopostors: a real-time geometry/impostor crowd rendering system [Dob+05] | Complex | 3D | Perspective | Impostor | Visualization; Rendering |
| Continuum Crowds [TCP06] | Complex | 3D | Top-down; Perspective | Curve; Articulated body | Steeringbehavior; Crowd dynamics |
| Pedestrian reactive navigation for crowd simulation: a predictive approach [PPD07] | Simple; Complex | 2D; 3D | Top-down; Perspective | Curve; Articulated body | Collision avoidance |
| Controlling individual agents in high-density crowd simulation [PAB07] | Intermediate | 3D | Top-down; Perspective | Simple body | Crowd dynamics |
| Group behavior from video: a data-driven approach to crowd simulation [Lee+07] | Complex | 3D | Top-down; Perspective | Articulated body | Group behavior; Crowd dynamics; Learning |
| The corridor map method [GO07] | Simple | 2D; 3D | Top-down; Perspective | Disc | Pathplanning |
| Real-time navigation of independent agents using adaptive roadmaps [Sud+07] | Complex | 3D | Top-down; Perspective | Articulated body | Pathplanning; Crowd dynamics |
| Motion planning [KL08] | Simple | 2D | Top-down | Disc; Curve | Pathplanning |
| Realtime crowd motion planning [Yer+08] | Complex | 3D | Perspective; Eye-level | Articulated body | Pathplanning; Crowd dynamics |
| Composite Agents [Yeh+08] | Simple; Complex | 2D; 3D | Top-down; Perspective | Disc; Curves; Articulated body | Social behavior; Crowd dynamics |
| Multi agent navigation on the GPU [Ble09] | Simple | 2D; 3D | Top-down; Perspective | Disc; Cylinder | Collision avoidance |
| Clearpath [Guy+09] | Intermediate | 3D | Perspective | Articulated body | Collision avoidance |
| Aggregate dynamics for dense crowd simulation [Nar+09] | Complex | 3D | Perspective | Articulated body | Crowd dynamics |
| Indicative routes for path planning and crowd simulation [KGO09] | Simple | 2D | Top-down | Disc; Curve | Pathplanning |
| Crowd patches [Yer+09] | Complex | 3D | Perspective | Articulated body | Procedural generation |
| GPU accelerated path planning for multi-agents in virtual environments [FSN09] | Complex | 3D | Perspective | Articulated body | Steering behavior; Crowd dynamics |
| The virtual marathon: parallel computing supports crowd simulation [YIÇ09] | Complex | 3D | Perspective | Impostor | Visualization; Rendering |
| Simulating human collision avoidance using a velocity-based approach [KO10] | Intermediate | 3D | Perspective | Articulated body | Collision avoidance |
| Planning shorts paths with clearance using Explicit corridors [Ger10] | Simple | 2D | Top-down | Disc; Curve | Pathplanning |
| Pledestrians [Guy+10] | Complex | 3D | Perspective | Articulated body | Pathplanning; Crowd dynamics |
| Reciprocal n-body collision avoidance [VDB+11] | Simple; Complex | 2D; 3D | Top-down; Perspective | Disc; Articulated body | Collision avoidance |
| Directing Crowd Simulations using Navigation Fields [Pat+11] | Complex | 3D | Perspective | Articulated body | Design; Flow |
| Least-effort trajectories lead to emergent crowd behaviors [Guy+12] | Simple; Complex | 2D; 3D | Top-down; Perspective | Disc; Articulated body | Collision avoidance; Energy optimization |
| Realtime densitybased crowd simulation [VTCG12] | Simple; Complex | 2D; 3D | Top-down; Perspective | Curves; Articulated body | Congestion avoidance |
| Realtime path planning in heterogeneous environments [JCG13] | Simple | 2D | Top-down | Curve | Terrain characteristics; Agent preferences |
| Parameter Estimation and Comparative Evaluation of Crowd Simulations [Wol+14] | Simple; Complex | 2D; 3D | Top-down; Perspective | Discs; Curves; Articulated body | Evaluation; Benchmarking |

Table 8: *An overview of the Representations as used in different methods. A semicolon indicates that multiple parameters are used in a method.*

## References


[Ble09] A. Bleiweiss. "Multi agent navigation on the gpu". In: *White paper, GDC* 9 (2009).

[Bra+03] A. Braun, S. R. Musse, L. P. L. de Oliveira, and B. E. Bodmann. "Modeling individual behaviors in crowd simulation". In: *Computer Animation and Social Agents, 2003. 16th International Conference on.* IEEE. 2003, pp. 143–148.

[Che04] S. Chenney. "Flow tiles". In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Eurographics Association. 2004, pp. 233–242.

[Dob+05] S. Dobbyn, J. Hamill, K. O'Conor, and C. O'Sullivan. "Geopostors: a real-time geometry/impostor crowd rendering system". In: *Proceedings of the 2005 symposium on Interactive 3D graphics and games.* ACM. 2005, pp. 95–102.

[FSN09] L. G. Fischer, R. Silveira, and L. Nedel. "Gpu accelerated path-planning for multi-agents in virtual environments". In: *Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on.* IEEE. 2009, pp. 101–110.

[Ger10] R. Geraerts. "Planning short paths with clearance using explicit corridors". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE. 2010, pp. 1997–2004.

[GO04] R. Geraerts and M. H. Overmars. "A comparative study of probabilistic roadmap planners". In: *Algorithmic Foundations of Robotics V.* Springer, 2004, pp. 43–58.

[GO07] R. Geraerts and M. H. Overmars. "The Corridor Map Method: Real-Time High-Quality Path Planning." In: *ICRA.* 2007, pp. 1023–1028.

[Guy+09] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey. "Clearpath: highly parallel collision avoidance for multi-agent simulation". In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* ACM. 2009, pp. 177–187.

[Guy+10] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha. "Pledestrians: a least-effort approach to crowd simulation". In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association. 2010, pp. 119–128.

[Guy+12] S. J. Guy, S. Curtis, M. C. Lin, and D. Manocha. "Least-effort trajectories lead to emergent crowd behaviors". In: *Physical Review E* 85.1 (2012), p. 016110.

[Hei+03] L. Heigeas, A. Luciani, J. Thollot, N. Castagné, et al. "A Physically-Based Particle Model of Emergent Crowd Behaviors". In: *Graphicon 2003-13th International Conference on Computer Graphics.* 2003, pp. 1–9.

[HM95] D. Helbing and P. Molnar. "Social force model for pedestrian dynamics". In: *Physical review E* 51.5 (1995), p. 4282.

[Hug03] R. L. Hughes. "The flow of human crowds". In: *Annual review of fluid mechanics* 35.1 (2003), pp. 169–182.

[JCG13] N. Jaklin, A. Cook, and R. Geraerts. "Real-time path planning in heterogeneous environments". In: *Computer Animation and Virtual Worlds* 24.3-4 (2013), pp. 285–295.

[KO04] A. Kamphuis and M. H. Overmars. "Finding paths for coherent groups using clearance". In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association. 2004, pp. 19–28.

[KGO09] I. Karamouzas, R. Geraerts, and M. Overmars. "Indicative routes for path planning and crowd simulation". In: *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM. 2009, pp. 113–120. URL: http://www.staff.science.uu.nl/~gerae101/motion_planning/irm.html.

[KO10] I. Karamouzas and M. Overmars. "A velocity-based approach for simulating human collision avoidance". In: *Intelligent Virtual Agents*. Springer. 2010, pp. 180–186.

[KL08] L. E. Kavraki and S. M. LaValle. "Motion planning". In: *Springer Handbook of Robotics* (2008), pp. 109–131.

[Kav+96] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *Robotics and Automation, IEEE Transactions on* 12.4 (1996), pp. 566–580.

[KL00] J. J. Kuffner and S. M. LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*. Vol. 2. IEEE. 2000, pp. 995–1001.

[Lee+07] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. "Group behavior from video: a data-driven approach to crowd simulation". In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association. 2007, pp. 109–118.

[Nar+09] R. Narain, A. Golas, S. Curtis, and M. C. Lin. "Aggregate dynamics for dense crowd simulation". In: *ACM Transactions on Graphics (TOG)*. Vol. 28. 5. ACM. 2009, p. 122.

[PPD07] S. Paris, J. Pettré, and S. Donikian. "Pedestrian reactive navigation for crowd simulation: a predictive approach". In: *Computer Graphics Forum*. Vol. 26. 3. Wiley Online Library. 2007, pp. 665–674.

[Pat+11] S. Patil, J. Van Den Berg, S. Curtis, M. C. Lin, and D. Manocha. "Directing crowd simulations using navigation fields". In: *Visualization and Computer Graphics, IEEE Transactions on* 17.2 (2011), pp. 244–254.

[PAB07] N. Pelechano, J. M. Allbeck, and N. I. Badler. "Controlling individual agents in high-density crowd simulation". In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association. 2007, pp. 99–108.

[RMT01] S. Raupp Musse and D. Thalmann. "Hierarchical model for real time simulation of virtual human crowds". In: *Visualization and Computer Graphics, IEEE Transactions on* 7.2 (2001), pp. 152–164.

[Rey99] C. W. Reynolds. "Steering behaviors for autonomous characters". In: *Game developers conference*. Vol. 1999. 1999, pp. 763–782.

[RD05] G Ryder and A. Day. "Survey of Real-Time Rendering Techniques for Crowds". In: *Computer Graphics Forum*. Vol. 24. 2. Wiley Online Library. 2005, pp. 203–215.

[Sud+07] A. Sud, R. Gayle, E. Andersen, S. Guy, M. Lin, and D. Manocha. "Real-time navigation of independent agents using adaptive roadmaps". In: *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*. ACM. 2007, pp. 99–106.

[SGC04] M. Sung, M. Gleicher, and S. Chenney. "Scalable behaviors for crowd simulation". In: *Computer Graphics Forum*. Vol. 23. 3. Wiley Online Library. 2004, pp. 519–528.

[TLC02] F. Tecchia, C. Loscos, and Y. Chrysanthou. "Visualizing Crowds in Real-Time". In: *Computer Graphics Forum*. Vol. 21. 4. Wiley Online Library. 2002, pp. 753–765.

[TCP06] A. Treuille, S. Cooper, and Z. Popović. "Continuum crowds". In: *ACM Transactions on Graphics (TOG)*. Vol. 25. 3. ACM. 2006, pp. 1160–1168.

[UT01] B. Ulicny and D. Thalmann. *Crowd simulation for interactive virtual environments and VR training systems*. Springer, 2001.

[VDB+11] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha. "Reciprocal n-body collision avoidance". In: *Robotics research*. Springer, 2011, pp. 3–19.

[VTCG12] W. G. Van Toll, A. F. Cook, and R. Geraerts. "Real-time density-based crowd simulation". In: *Computer Animation and Virtual Worlds* 23.1 (2012), pp. 59–69.

[Wol+14] D. Wolinski, S J Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettré. "Parameter estimation and comparative evaluation of crowd simulations". In: *Computer Graphics Forum*. Vol. 33. 2. Wiley Online Library. 2014, pp. 303–312. URL: http://gamma.cs.unc.edu/CParameter/.

[Yeh+08] H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha, and M. Lin. "Composite agents". In: *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association. 2008, pp. 39–47.

[Yer+08] B. Yersin, J. Maïm, F. Morini, and D. Thalmann. "Real-time crowd motion planning". In: *The Visual Computer* 24.10 (2008), pp. 859–870.

[Yer+09] B. Yersin, J. Maïm, J. Pettré, and D. Thalmann. "Crowd patches: populating large-scale virtual environments for real-time applications". In: *Proceedings of the 2009 symposium on Interactive 3D graphics and games*. ACM. 2009, pp. 207–214.

[YIÇ09] E. Yilmaz, V. Isler, and Y. Y. Çetin. "The virtual marathon: parallel computing supports crowd simulations." In: *IEEE computer graphics and applications* 29.4 (2009), pp. 26–33.