D4.7 Reactive Agent and Touch Enabling

Grant Agreement nr	856879
Project acronym	PRESENT
Project start date (duration)	September 1st 2019 (36 months)
Document due:	28/02/2022
Actual delivery date	28/02/2022
Leader	Inria
Reply to	julien.pettre@inria.fr
Document status	Submission Version

Project funded by H2020 from the European Commission

Project ref. no.	856879
Project acronym	PRESENT
Project full title	**P**hotoreal **RE**altime **S**entient **ENT**ity
Document name	Reactive Agent and Touch Enabling
Security (distribution level)	Public
Contractual date of delivery	28/02/2022
Actual date of delivery	28/02/2022
Deliverable name	D4.7 - Reactive Agent and Touch Enabling
Type	Report
Status & version	Submission Version
Number of pages	52
WP / Task responsible	Inria
Other contributors	-
Author(s)	Alberto Jovane, Adele Colas, Julien Pettre
EC Project Officer	Ms. Diana MJASCHKOVA-PASCUAL - Diana.MJASCHKOVA-PASCUAL@ec.europa.eu
Abstract	This deliverable presents a set of techniques that aim at providing virtual humans with the ability to react and interact with a user, in a non-verbal communication mode that involves, proxemics, gestures or gaze behaviours for characters. We also explore the role of touch sensation on user behaviour.
Keywords	Virtual agent, reaction, interaction with user, touch sensation on user behaviour
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	No
Read by partners	No
Mgt. Board approval	No

Document History

Version and date	Reason for Change
1.0 15-02-2022	Document created by Julien Pettré, Alberto Jovane, Adèle Colas
1.1 23-02-2022	Version for internal review
1.2 DD-MM-YYYY	Revisions in response to review: final version submitted to Commission

Table of Contents

1 EXECUTIVE SUMMARY

The deliverable D4.7 presents a set of techniques mainly developed by the partner Inria in the context of the PRESENT project which aims at providing virtual humans with the ability to react and interact with a user, in a non-verbal communication mode that involves, proxemics, gestures or gaze behaviours for characters. We also explore the role of touch sensation on user behaviour.

We therefore hypothesise here that both parties, the user and the virtual agent, interact through virtual reality. The user is immersed in a virtual environment populated by virtual humans. The problem addressed in the project, and in particular via the activity of the partner Inria in work package 4, is to enable a form of non-verbal communication between the user. We consider 3 aspects in this communication:

- The case of an interaction between a user and a single character (1-to-1 interactions): in this case, we are interested in the adaptation of a character animation (produced by an external method) to the relative position of a user, considering then the user as an observer of this movement.
- The case of an interaction between a user and several characters (1-to-n interactions): in this case, we are interested in issues of proxemics and in particular relative trajectories between groups of characters and the user. We explore methods of controlling the configuration of this group, to allow a naive user to produce new interactions with a very relative coding effort.
- Still in the case of 1-to-n interactions with characters, we explore characters' gaze behaviours, and how this relates to visual saliency, which can turn into an important question when initiating interactions with users.
- Finally, in the case of very close user-character interactions, we explore the possible role of a tactile communication channel, and question the influence that the sensation of contact can have on a user's behaviour.

For each of these points, we provide technical solutions described in detail in this document, as well as an evaluation of these solutions to verify their applicability in the context of the PRESENT project and more generally of interaction with virtual characters. Our work opens new avenues for the design and control of virtual character behaviour.

2 BACKGROUND

This deliverable is completing the previous deliverable D4.3. In this previous document, we reported a number of activities that were essential in the development of the animation techniques presented in the current deliverable. In D4.7, we will not repeat the content of D4.3, but include an updated version of the description of animation techniques in the state they reached by the end of the project.

Involved in interactions with users, we want the PRESENT agents to be capable of reacting in natural ways to users motions and behaviours. Reaction capabilities involve for example being attentive to users, show emotional reactions to their movements, adjust their behaviours to the changes in user position. It covers both cases of interaction where a user is interacting with a single agent (1-to-1 interaction), or a group of agents (1-to-n interactions). As the reaction capabilities of agents can be interpreted in many different meanings, PRESENT is focusing efforts on some main aspects : (i) provide designers friendly tool to extend in large amounts the believability, expressivity and richness of collective behaviours for agents; (ii) create new animation techniques to extend the expressivity of characters animation with possibility to perform on-line adjustments of the conveyed information through body gestures; (iii) explore new modalities for non-verbal communication with users with haptic techniques.

This deliverable reports work performed in WP4 to develop reaction capabilities of agents, as well as usage of haptic techniques to communicate through other sensory channels with users. These aspects are mainly explored by the Inria partner. We also report on studies that we have performed to better fix the requirements of both animation and haptic techniques.

3 INTRODUCTION

3.1 Main objectives and goals

When involved in an interaction with a virtual agent, any action of the users out of a set of predefined actions will demonstrate the lack of reaction capabilities of agents, starting from the most basic ones, such as when trying to startle them.

For this reason, PRESENT is exploring solutions to improve agents reaction capabilities in three two man ways:

1 In cases where users interact with a **group** of agents, by showing agents collective behaviours in complex scenarios, and how these collective behaviours adjust to users actions.
2 In cases where users interact with a **single** agent, by showing how the agent's motion is adjusted to the changes in positions of the user, or scenario-guided indications to express, through body movements, specific intentions.

Since these two goals are aligned with the more general goal of improving non-verbal communication capabilities for agents, we finally add to:

3 Explore touch as a sensory communication channel for agents to convey information to users.

3.2 Methodology

The methodology applied to achieve these 3 goals will structure the report at hand.

Section **4.1** describes our technical approach to the problem of adjusting agents' animations to make them more expressive in reaction to users' actions or given scenarios. We elaborate an

"expressive animation filter" that can process motion capture data, considered to be recorded in neutral conditions, and taint it to be adjusted to the position of the user, acting as an observer of the performed action.

Section **4.2** describes our techniques to improve collective reaction capabilities for PRESENT agents. In this part of the work, we have identified the difficulty in designing new and potentially collective behaviours as a major bottleneck. To overcome these difficulties, we propose an animation system based on the concept of *Interaction Fields* to elaborate new types of local interactions between agents, as well as with users.

Section **4.3** explores more precisely, in the context of interaction with multiple characters, how their gaze animation can be a potential mean to initiate interactions with users, or more precisely, to make characters wanting to interact with users more salient to him.

Section **4.4** explores the importance of haptic rendering of contacts between users and virtual agents, and evaluates the usage of vibrotactile wearable devices that meet the requirement of immersion with mobility and light equipment.

3.3 Terminology

expressive filter: the proposed system that automatically edit the motion of a virtual agent to adapt it to a desired visual appearance,
visual motion features: a set of measurements of a human motion computed on the image space from an observer point of view,
warping units: coordinated and weighted alteration of the kinematic structure of the agent skeleton,
interaction fields(IF): vector fields that prescribe a velocity or an orientation to agents regarding their relative position to the source of the field.
source of an IF: An agent or an object that impacts their neighbouring agents with IFs and are not impacted.

4 Methods for reacting agent and touch enabling

4.1 EXPRESSIVE FILTERS: 1-to-1 interactions

In order to enhance the expressivity of the PRESENT virtual agent and enrich the interactive communication with the user, we introduce in this section a novel system that we have called *expressive filter*. We start from the assumption that human's movements are often performed in relation to an interactant. This is particularly true when motion is considered as a nonverbal communication channel **[Hinde 1972]**. Indeed in such communication tasks, humans control their movements by fundamentally taking into account how it can be visually perceived by an observer. For instance, when one waves at someone – a typical voluntary nonverbal communication gesture to attract attention – one makes sure his/her hand is visible to this person, e.g. adjusting body orientation, waving amplitude and speed to make the motion salient enough, as well as moving his/her face and eyes to enable gaze contact and ensure that attention is successfully attracted. This example clearly highlights the links between the kinematics of a motion and the *visual features* perceived by an observer.

The *expressive filter* aims to simulate these interactions, editing the motion according to evaluated *visual motion features*. This notion of *visual motion features* are defined in relation with an observer's point of view and Field of Vision (FoV), and covers features such as visibility and centrality of limbs, coverage of limbs and of their motion, generated optical flow, motion amplitude in the view plane, etc.

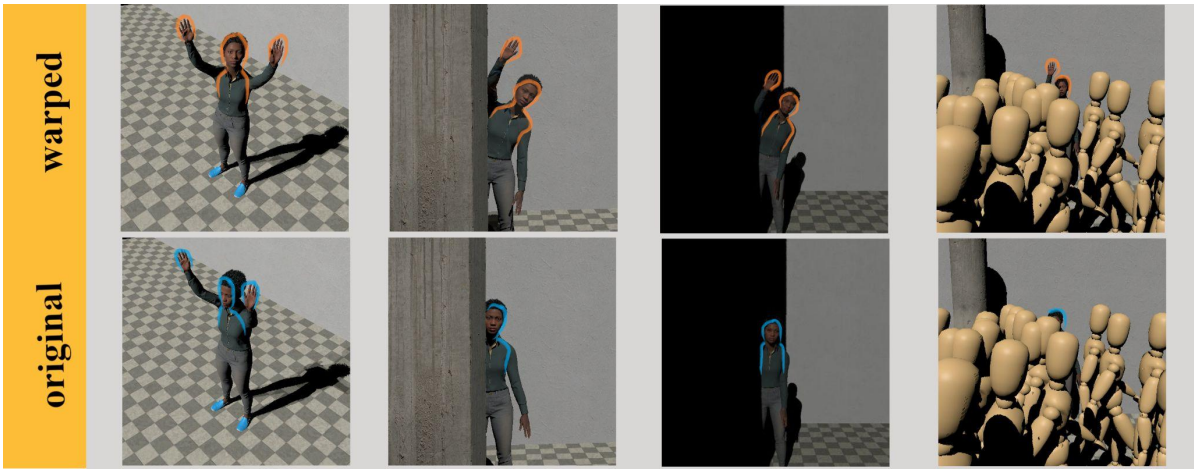


Figure 4.1.1: We propose a novel real-time motion editing technique that performs a view-dependent environment-ware warping of character animations, driven by user-specified visual motion features. The bottom row displays examples of original animations and the top row displays the warped versions of the same animations. The images show how a desired increment in visual coverage (one of our visual features) impacts the kinematic chains of the character to draw more attention. The *visual features* are aware of visibility and lighting conditions.

4.1.1 Context

Nonverbal communication refers to all modes of communication that do not use speech. This includes facial expression, para-linguistic, proxemics and touch, as well as body gestures and postures. During social interactions, a continuous exchange of such signals is possible, mainly because people are able to see each other **[Cañigueral and Hamilton 2019]**. This means that the interpretation of social signals involving a mutual interaction between the observer and the other(s) person(s) should be expressed in the reference frame of the observer, namely his/her FoV. This visual perception for social interaction allows then to infer on a person's intention [Blakemore and Decety 2001; Knoblich and Sebanz 2008], personality **[Neff et al. 2010]**, as well as emotions **[de Gelder et al. 2015; Roether et al. 2009]**.

We have designed the PRESENT expressive filter to animate agents in social interactions, or in immersive virtual reality applications where agents are meant to adjust themselves to the users' state in the virtual scene.

Yet only a few approaches have considered the influence of the camera angle, and the resulting *visual features* it yields, as a mean to control and warp a character animation. This work proposes the design of viewpoint-dependent *motion warping units* that perform subtle updates on animations through the specification of *visual motion features* such as visibility, or spatial extent.

Our approach is inspired by robotics visual servoing **[Chaumette and Hutchinson 2006]**, a technique to control a robot's motion based on visual sensor feedback, e.g. a camera. In analogy with visual servoing, we want to control a character (a robot) motion with respect to an observer (a camera) position. We express the problem as a specific case of visual servoing, where the warping of a given character motion is regulated by a number of *visual features* to enforce.

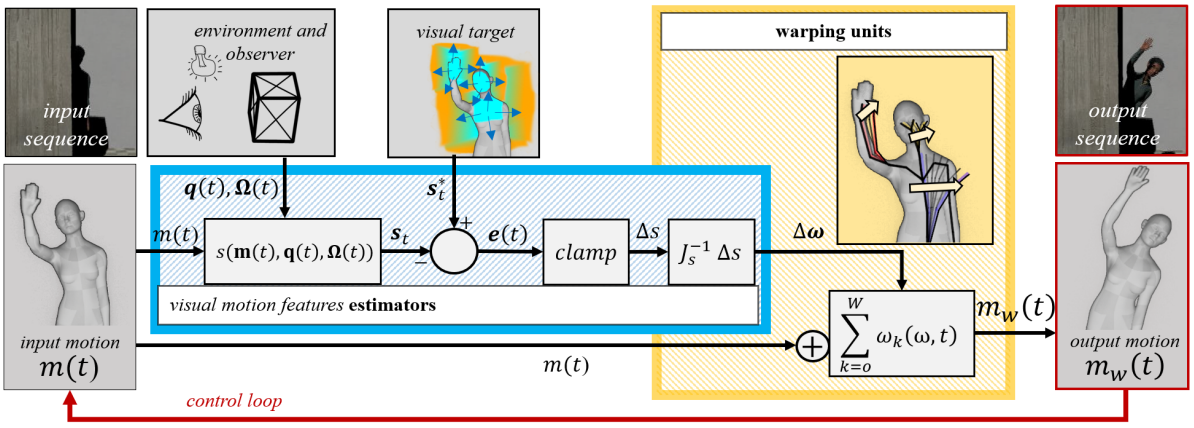


Figure 4.1.2: Overview of our approach. From an input sequence of a character animation, we first estimate different *visual motion features* on the current pose, considering the environment, the observer's viewpoint and a visual target (blue). Then, multiple plausible motion modifications are computed manipulating *warping units* (yellow), and the ones that minimise the visual error between the current state and the target are applied to the output motion. This process is repeated for the whole motion over a control loop (red).

4.1.2 Method

The aim of our approach is to provide designers or interactive applications with a warping controller that adjusts a character animation using view-dependent *visual motion features*. The input of our system is (i) an animated character, (ii) a set of user-specified *visual motion features* to fulfil (e.g. visual coverage, vertical extension), and (iii) a camera angle or trajectory that views the character in its environment. Our system adjusts the character animation on a per-frame basis by satisfying user-defined *visual motion features* in an inverse design approach, from features to parameters.

Technically, we propose to express the problem as a specific instance of the eye-to-hand visual servoing principle **[Dombre and Khalil 2013]** in which a camera, fixed or animated in the world, observes the motion of one or multiple kinematic chains. Rather than driving the velocity of kinematic chains from on-screen velocities – as implemented in traditional visual servoing tasks **[Espiau et al. 1992]** or with through-the-lens control **[Gleicher and Witkin 1992]**, our objective is to develop a control law that updates velocities in the kinematic chain by regulating the difference between globally measured *visual motion features* and expected ones in the 2D camera space. This requires the design of (i) *estimators* that are able to measure the values of the expected *visual motion features* from the given camera, (ii) *warping units* that alter the parameters of the kinematic chains, and (iii) a control loop that exploits the difference between estimated and expected features to drive the warping operators. Our overall approach is described in Figure **4.1.2**.

Section **4.1.2.1** presents the different *visual motion features* we consider when warping the animations, then Section **4.1.2.2** details the design of our warping operators, and Section **4.1.2.3** explains the regulation of the *visual motion features* using our warping operators.

4.1.2.1 Estimators of visual motion features

The perceptual mechanisms by which humans look at, read, and understand images are well studied nowadays. Typical features such as chrominance, contrast and motion in the spectator field of view are well-known bottom-up key factors that influence audience attention. Attention is also driven by a number of top-down factors such as object semantics (faces draw strong attention), cultural background, and tasks to perform **[Kimura et al. 2013]**.

With this in mind, our purpose is to propose, *estimators* to measure *visual motion features* are computational characteristics designed to measure how well the motion of a character is perceived from an observer's viewpoint. Moreover, these features represent here a proxy for visual attention. As such, they provide a view-dependent metric influenced by the character motion, the lighting in the scene and by potential occluders, and therefore provide a mean to control the amount of perceived motion in a screen space.
For a given time frame t , we express the *visual motion features* as a time dependent vector s_t:

$$s_t = [s_1(t), ..., s_V(t)]^T = s(\boldsymbol{m}(t), \boldsymbol{q}(t), \Omega(t))$$ (Equation 4.1.1)

where $s_i(t)$ is a function of a) the character pose specified by $\boldsymbol{m}(t)$, b) the observer pose specified as a camera pose $\boldsymbol{q}(t)$, and c) the state of the environment $\boldsymbol{\Omega}(t)$ that accounts for the rest of the scene, notably any lights and geometries that may affect the visibility of the virtual agent from the observer viewpoint. The function $s(\boldsymbol{m}(t), \boldsymbol{q}(t), \boldsymbol{\Omega}(t))$ performs the scene and character rendering from the given camera, and computes the *visual motion features*.

In the present implementation of the method we consider and evaluate the *visual motion features* listed here:

- *Apparent static coverage* measures how much of a character's projected image is perceived in frame (accounting for visibility and lighting).
- *Apparent static extension* measures the horizontal occupancy (resp. vertical occupancy) as a ratio between the left-most and right-most pixels (resp. bottom-most and top-most) of the character on the screen width (resp. height), also accounting for visibility and lighting.
- *Apparent motion coverage* measures how much of a character's motion from one frame to another is perceived in the image.
- *Apparent motion extension* measures the horizontal occupancy (resp. vertical occupancy) as a ratio between the left-most and right-most pixels of a character's motion (resp. bottom-most and top-most) on the screen width (resp. height).

In practice, *visual motion features* are computed through hardware rendering and straightforward image analysis. At each time step, a frame is rendered from the observer's viewpoint and only the perceived pixels of the virtual agent are kept. A pixel is considered as perceived if it is not occluded, or if its luminance is under a given threshold (e.g. in a shaded or dark area). *Visual motion features* are estimated through pixel operations such as counting or comparing coordinates and distances in the image space.

Semantic layers on body representations. Sub-meshes of virtual characters are tagged so as to identify specific parts (face, arms, chest, legs, inside of hands). This enables us to arbitrarily activate or deactivate body parts according to the performed motion, e.g. to focus and render only the waving hand for a waving character case. In addition, rigid objects can be attached to the skeletal joints and *visual motion features* can be computed on them (e.g. holding a sheet of paper).

4.1.2.2 Motion warping units

We first rely on a classical skeletal structure with joints using a tree of kinematic chains. An animation of the skeletal structure is defined as a set of keyframes along with an interpolation technique. We can thus define a function $\boldsymbol{m}(t)$ that computes the current pose of the character at time t , expressed as a vector of the degrees of freedoms of the characters joints denoted $\boldsymbol{\theta}_t$:

$$\boldsymbol{\theta}_t = [\theta_0, .., \theta_K]^T = \boldsymbol{m}(t)$$ (Equation 4.1.2)

where K represents the number of degrees of freedom of the skeletal structure.

Rather than regulating *visual features* by controlling simultaneously the whole vector $\boldsymbol{\theta}$ of joints of a character (which may create unexpected or unrealistic changes in the body poses), we propose to define specific groups of joints in the skeletal representation and define these as *motion warping units*. These groups are defined to provide a localised control on a character (e.g. only the spine, only the arms, only the head plus shoulders), which is a classical approach when designers need to locally warp motions without impacting the whole body (see Table **4.1.1**). Furthermore, we design our *motion warping units* in a parameterised way that maintains the coupling between parameters of the kinematic chain by using a linear combination given a warping factor. This enables small warpings on the animations without losing the nature of the motion.

Joint	Movement
Spine	Bend forward-backward | Bend left-right | Rot. vertical axis
Neck-head	Bend forward-backward | Bend left-right | Rot. vertical axis
Shoulders	Flexion-extension | Abduction-adduction | Medial-lateral rot.
Elbow	Flexion-extension | Rotation on its axis
Wrist	Flexion-extension | Ulnar-radial deviation | Supination-pronation

Table 4.1.1: Defined pose *warping units* for upper-body motions.

Our *motion warping unit* is therefore defined as a function $\boldsymbol{\omega}_k(\omega,t)$ that, given a scalar displacement value w, computes a pose offset vector at time t for the joint angles in the *warping unit*. For a given *warping unit*, the pose offset vector is added to the current pose $\boldsymbol{m}(t)$ to generate the warped animation. The index k represents the k-th *warping unit*, and $\boldsymbol{\omega}_k(\omega,t)$ yields a vector of size M_k (depending on the number of degrees of freedom of the *warping unit*). Our *motion warping unit* defines a pose offset vector in which each offset is a weighted linear combination of ω:

$$\Delta\theta = \boldsymbol{\omega}_k(\omega,t) = [w_0^k\omega, .., w_M^k\omega]^T$$ (Equation 4.1.3)

The scalar value w_i^k is a weighting constant specific to a given *warping unit* k and degree of freedom i of the kinematic chain and can be viewed as a stiffness coefficient, traditionally used when manipulating inverse kinematic chains. The linear combination with ω ensures a coupling in the offset computation of animation parameters. The corresponding value $\Delta\theta i$ represents the i-th kinematic angle offset computed by the warping operator.

Each *motion warping unit* therefore computes a pose offset vector $\Delta\theta i$, and all offset vectors are aggregated on the current character pose to create the warped motion (see next equation). The magnitude and direction of the computed offset vectors are driven by a vector $[\omega 1, .., \omega k]^T$ of *warping unit* parameters, the value of which is computed by the visual servoing task (see Section **4.1.2.3**).

The overall warped parameters $\boldsymbol{mw}$ (t) of the animation at each time t are given by:

$$\boldsymbol{m}_{\boldsymbol{w}}(t) = \boldsymbol{m}(t) + \sum_{k=0}^{W} \boldsymbol{\omega}_k(\omega,t)$$ (Equation 4.1.4)

which is a simplified notation since vectors $\boldsymbol{\omega}_k(\omega,t)$ are of different sizes.

4.1.2.3 Driving warping units through visual motion features

Our objective is to compute the optimal set of *warping unit* parameters $\boldsymbol{\omega}$ at each time t of the animation from a given set of desired *visual motion features* s_t , through a **control loop**. We first express the relation $s_t = f_t(\boldsymbol{\omega})$ where f_t computes the estimation of *visual motion features*. As a direct relationship between s_t and $\boldsymbol{\omega}$ exists, our goal is to solve this equation to obtain $\boldsymbol{\omega}$ from s_t . Due to the strong non-linearity of the relation between *visual motion features* and kinematic parameters, a classical approach is to study the problem in the velocity space.

Given this set of *visual motion features* s_t that depend both on a camera viewpoint at time t and a set of *warping unit* parameters $\boldsymbol{\omega}$, the differential s_t expresses how the variations in the visual features are related to the camera and the character animations.

As defined, this problem is a specific case of an eye-to-hand visual servoing problem where a specified velocity in the image space of a fixed camera looking at kinematic chain is used to drive its degrees of freedom **[Espiau et al. 1992]**. This visual servoing relation is generally defined as:

$$\dot{s}_t = L_s V_n J_n(\theta)\dot{\theta} + \frac{\delta s}{\delta t}$$ (Equation 4.1.5)

where $\boldsymbol{J}_n(\theta)$ is the Jacobian of the kinematic chain, $\boldsymbol{V}_n$ the kinematic tensor transformation from the camera to the character, $\boldsymbol{L}_s$ the interaction matrix, and $\frac{\delta s}{\delta t}$ describes the variations of s caused by a

movement of the camera (n represents the number of degrees of freedom of the kinematic chains). This relation defines the correlation between the variation in *visual motion features* and the variations in degrees of freedom of the kinematic chain. We rely on this formulation to express our problem in terms of the *warping unit* parameters $\dot{\omega}$.

$$\dot{s}_t = J_s\dot{\omega} + \frac{\delta s}{\delta t}$$ (Equation 4.1.6)

Jacobian computation using finite differences. Each element of the Jacobian matrix encodes a partial derivative of *visual motion features* values (s) over each *warping unit* feature (ω):

$$J_s = \left(\frac{\delta s_q}{\delta \omega_k}\right)_{q,k}, q \in [0..V], k \in [0..W]$$ (Equation 4.1.7)

where V is the total number of *visual motion features* and W is the total number of *warping units*. A forward evaluation enables us to compute a variation in *visual motion features* Δs from a variation in *visual motion features* $\Delta\theta$ (for small enough variations):

$$\Delta s = J_s \Delta\theta$$ (Equation 4.1.8)

To solve the problem, we therefore reverse the previous equation (Equation 4.1.8) by approximating J^{-1}using a damped least square method.

$$\Delta\omega = J^{-1}\Delta s$$ (Equation 4.1.9)

The input vector Δs is classically computed as the difference between the expected features and the measured features $e = s^* * s_t$ and capped with a maximum threshold:

$$\Delta s = \begin{cases} e & if \|e\| \leq DS_{max} \\ DS_{max}\frac{e}{\|e\|} & otherwise \end{cases}$$ (Equation 4.1.10)

In practice the computation of the Jacobian J_s at any time t requires to evaluate each *visual motion feature* for each $2W$ variation of *warping unit* parameters. This is performed using finite differences (scenes need to be rendered to assess *visual features*).
We finally build the Jacobian matrix as in:

$$J_s = \left(\frac{s_q^{\omega_k^+} - s_q^{\omega_k^-}}{2\omega_k}\right)_{q,k,t}$$ (Equation 4.1.11)

We then estimate the inverted Jacobian, and use it to extract the warping direction $\Delta\omega$ (see Equation (Equation 4.1.9) as $\Delta s = J_s\Delta\theta$). An additional clamping is applied to the obtained vector to smooth the final motion modification. Finally, the new warped motion is computed with Equation (Equation 4.1.4). Implementation details for different case studies are detailed in the following section.

4.1.3 Case studies

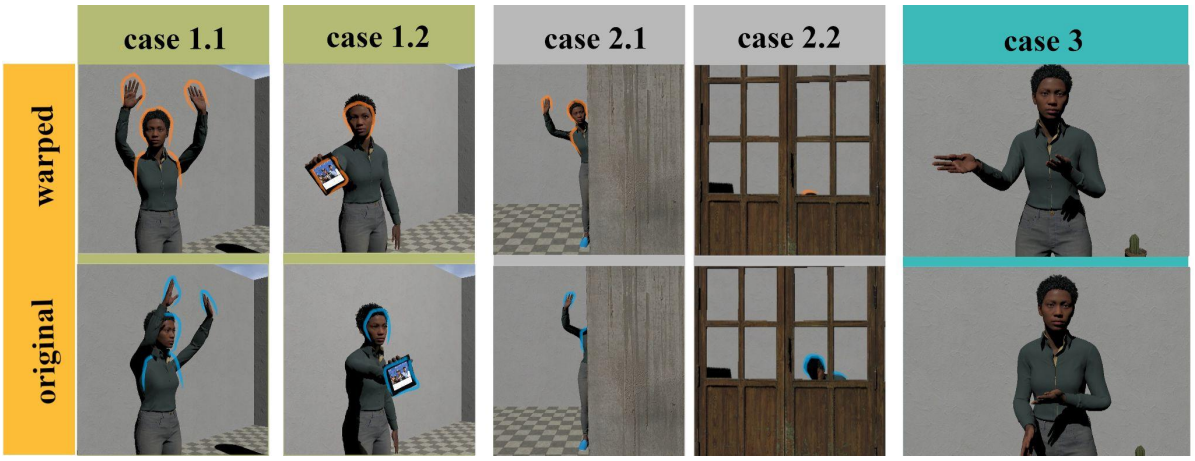


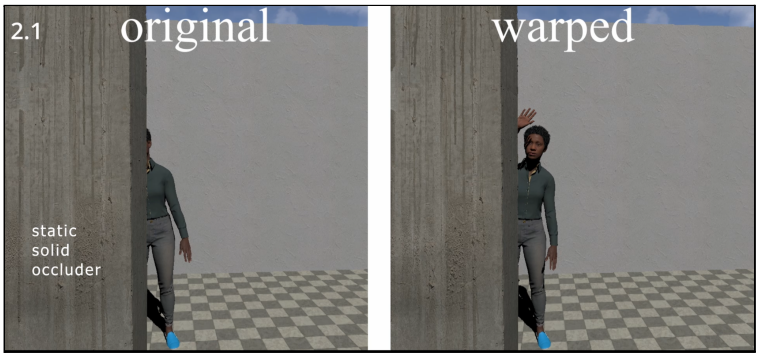
Figure 4.1.3: Results from our three use-cases. Character parts/objects on which the estimation of visual features is performed are highlighted. On the left (1.2 and 1.2), we study the influence of viewpoint changes for a character with and without an object. The desired visual feature consisted in increasing the visual coverage. In the middle (2.1 and 2.2), we study the influence of solid and sparse occluders. Visual coverage was also used as the regulating visual motion feature. On the right (3) we explore how extraverted effects can be created by applying upper body *warping units*, and regulating their apparent vertical and horizontal extension.

We demonstrate our method over three case studies that all consider one virtual agent acting in front of one observer. We focus on upper-body nonverbal communication mainly using head, torso, arm and hand body parts. About environment conditions and related visual targets, we explore: i) the influence of changes in the observer's viewpoint, ii) the influence of occlusion or lighting conditions, and iii) the potential exaggeration of extraverted traits of a motion. Our companion (Video **4.1.1)** provides additional examples. Results are discussed in this section, and the method is more generally discussed in Section **4.1.4**.

4.1.3.1 Implementation

We implemented our technique using Unreal Engine 5. The approach runs at interactive frame-rates (>30fps) and can be used in interactive and non-interactive contexts. *Estimators* (see Section **4.1.2.1**) were implemented using shaders, while the *visual motion feature* vector was computed through multiple rendering passes. The warping operators were built above Unreal control rigs (see Section **4.1.2.2**) which provide a direct access to the degrees of freedom of the skeletal structure of animated characters. Our virtual agent is based on a Meta-humans model. The baseline motions (unwarped) were either recorded using a motion capture system (Xsens suit) or taken from a public database (Mixamo). Finally, we execute at run time the control loop (see Section **4.1.2.3**) by generating $(2*W)+1$ copies at each frame of the virtual agent from the observer point of view: 1 copy is used as a reference (unwarped motion) for *visual motion features* evaluation, whilst the $2W$ others are used to compute $\boldsymbol{J}_s$ by rendering warped motions, for each of the W *warping units*, in both warping directions.

To encode the different use-cases, the designer first needs to decide the set of active *warping units*, i.e. parts of the body which animation will be warped (see Table **4.1.3**). Then, he/she needs to select which *visual motion features* to control, along with the related body parts on which these features are computed (see Section **4.1.2.1**). For each *visual motion feature* the designer can specify the magnitude and direction of its change over time. The magnitude affects mostly the timing on which the modifications are applied, then, for each scenario we select the most appropriate level, generally lower intensity for cyclic motions that has time to adapt, and higher for single motion and fast adaptive.



Video 4.1.1: All the obtained results in a video showcase here.

4.1.3.2 Case study 1: viewpoint changes

Objective: The purpose of this first study is to experiment the influence of viewpoint change on the animations to edit.

Scenario 1.1: In this scenario, the virtual agent performs a two-hand waving motion. We recorded the baseline motion assuming that the agent was facing the observer. The objective is to modify this motion for a different camera angle, in a way that the agent better captures the observer's attention according to his/her position. The *warping units* related to the agent's spine, neck and arms were selected since they affect the waving motion of both hands. We performed the visual evaluation with the face and the waving hands as body parts linked to the visual target. We selected visual coverage as the *visual motion feature* and specified a value to maximise visual appearance. Also, when positioning the observer at different distances from the camera, we used the control of the apparent horizontal extension to adapt the waving amplitude.

Scenario 1.2: Here, the virtual agent shows to the observer an object placed in its right hand. Again, our baseline animation was recorded with the observer facing the agent. The challenge is to adapt arm and hand motions to the observer's point of view in such a way that the object shown appears in the centre of the observer's FoV. The selected *warping units* were the ones related to the spine, neck and right arm of the agent since the motion was performed with this arm. Regarding the relevant limbs related to the visual target evaluation, we selected the head and the additional external object (a tablet) held by the hand. Similarly to the previous scenario, we aimed for an increment of the visual coverage of the face and the tablet screen to maximise their visual appearance.

Informal analysis: Scenarios 1.1 and 1.2 are illustrated in Figure **4.1.3** left. These scenarios were tested with different viewpoint parameters, namely, position, orientation, static/dynamic motions (see also our companion Video **4.1.1**). Results show that our approach successfully adjusts motions to changing viewpoints. Indeed, the first scenario demonstrates how the waving amplitude adjusts according to the observer's distance. In the second scenario we show how a unitary motion could be rapidly warped, to fit the duration of the action, with the same target of maintaining the visibility of a relevant object shown in the observer's FoV. We also show that our approach allows for the control of multiple limbs by generating subtle variations on a motion without affecting its original purpose. One could argue that similar results, especially Scenario 2, could be replicated using an inverse kinematics (IK) approach. This is only partly true, as our approach based on *visual motion features* also integrates environmental conditions such as lighting or scene layout (e.g. bring an object in front of someone, in light) with the exact same setting.

4.1.3.3 Case study 2: occlusion - visibility

Objective: In this case study we aim at exploring through two scenarios the warping of agent limb motions to ensure proper visibility from the observer's viewpoint.

Scenario 2.1: In the first scenario, we recorded a one-hand waving motion, fully visible to a facing observer. The objective is to adapt this motion to improve its visibility by accounting for environment effects – occluding or partially occluding the animation. The selected *warping units* influenced the

spine, the right arm and hand. We considered as relevant limbs the ones that usually help capture the observer's attention at a distance i.e. the waving hand, the face and the upper torso. For the same reason, the visual coverage was selected as the active *visual motion feature*. In this case, an increment of this feature for the hand, face and upper torso would improve the perceived visibility of the motion.

Scenario 2.2: Here, our agent tries to hide from the observer; to achieve this, we captured a crouching motion as if someone was hiding behind an object, and used it for the original animation. The objective here was to adapt the motion to make the agent less visible from the observer's viewpoint according to occluders' size. We selected as *warping units* those related to the spine and the neck, and as relevant limbs all the upper body ones. The visual coverage was selected as the *visual motion feature*, with the visual target of an overall decrement in this coverage. Finally, we also tried to simultaneously increment the perceived visual coverage of a specific limb while hiding the rest, e.g. maintaining eyes visible.

Informal analysis: Scenarios 2.1 and 2.2 are illustrated in Figure **4.1.3** centre. These scenarios test different visibility parameters: static/dynamic occluders, static/dynamic lighting, different kinds of obstacles (see also our companion Video **4.1.1**). Results show that our method successfully adapts motions to visibility conditions and targets. Indeed, scenario 2.1 demonstrates how a waving motion initially recorded in a clear view situation facing a frontal observer can be adapted in other visibility conditions, by bending and adjusting the configuration of the arm towards a space where it was more visible. Scenario 2.2 shows how a crouching motion can be adapted to increase its hiding purpose, and we also show that our method enables combining this purpose with additional minor behaviours such as maintaining the top of the head visible. Such results demonstrate the advantage of our visual approach for these kinds of situations, by enhancing and adapting motions in different visibility conditions.

4.1.3.4 Case study 3: expressivity

Objective: Here, our aim is to experiment how *visual features* could be exploited to control the expressivity of a motion with our approach. For the current example, we aim at influencing extraversion, one of the Big Five traits of personality **[Goldberg 1990]**. This trait describes someone who typically captures the attention of an observer, is enthusiastic, energetic and sociable.
In this scenario, the virtual agent is performing communication gestures as if it was talking. For the original animation we motion captured an actor conversing with the experimenter in a neutral way. The objective here is to adapt the arm and the hand motions to increase the trait extraversion of the agent, taking into account the observer's viewpoint.

We selected *warping units* related to the elbow, shoulders, arms and hands, and the selected limbs were arms and hands. Visual coverage, apparent vertical and horizontal extensions were selected for the *visual motion features*. The visual target was then to increase both extensions to make the agent appear more extraverted. Indeed, **[Neff et al. 2010]** described several modifications to the character's gestures with a perceptual effect of creating an extraverted personality – by increasing spatial scale of the strokes, elbow rotations outwards (arm swivel) and increasing the shoulder raise.
Informal analysis. Scenario 3 is illustrated in Figure **4.1.3** right (see also our companion Video **4.1.1**). In this case study, we show that our method can modify gestures in similar ways than previous studies **[Neff et al. 2010]**, which should also increase the perceived extraversion of the character.

4.1.4 Discussion and Limitations

The presented system (*expressive filter)* presents a motion warping technique that enables linking low-level motion variables with FoV-dependent *visual motion features*. Our results show that our approach is effective and applies to various cases, such as adjusting motions to changes in an observer's position, environment or lighting conditions. We also believe that this technique can be exploited to influence the expressions conveyed by animations (e.g. intro vs. extroversion) thereby helping designers to fine-tune the personalities of their characters or having virtual characters adapt their non-verbal communication towards observers (or avatars). Our proof of concept validates the visual servoing scheme and yields a general and promising solution. While the method presents some analogies with inverse kinematics methods, through-the-lens techniques, or line of action

control, they offer higher levels of control than just positions and velocities of joints in the image space.

Currently, our method is limited in multiple ways. First, it requires a prior selection of *visual motion features* to be controlled, *warping units* to activate, or magnitude and direction of visual features. Methods to select relevant combinations of parameters would improve the practical usability of the approach. In addition we only explored a limited set of spatial visual features while operators could also perform time warping, and measure features over a sliding window rather than on a per-frame basis. Directly integrating computational saliency techniques **[Bruce et al. 2015]** closer to visual attention mechanisms could also help to guide the warping of animations. Additional saliency biases could be added to account for top-down attention mechanisms specific to characters (eg. focus of attention on head, eyes and hand movements) to build an attention-driven approach.

Second, the design of our *warping units* remains empirical. Existing work to automatically define rigging functions **[Holden et al. 2016]** could improve over our solution. In our case, we could explore means to automatically correlate low-level motion variables with the variations of *visual motion features*, with the difficulty that these relations depend on the motion performed, the desired goal of editing, and the observer's position.
At this stage, we also left apart the question of setting the appropriate levels of *visual motion features* editing. This question is particularly interesting in the case of controlling motion expressivity. By which level a feature should be adapted to change the expression of motion? Of the same importance, do kinematic limitations of motion editing (e.g. enforcing joints limitations, remaining in the human motion manifold, not provoking self collisions) enable reaching the desired editing levels? A data-driven approach would be relevant to address these questions. The difficulty is twofold: one is to gather the required amount of data to capture feature level variations with corresponding semantics, the other one is to deal with human variability in such behaviours.
An important direction for future work is related to the evaluation of our results. At this stage, we presented what we consider to be a proof of concept of our approach, and displayed its effectiveness on various use-cases. Yet, in-depth perceptual evaluations of the edited motions would need to be conducted. Thanks to the real-time capacity of our approach, evaluations could be performed in VR by evaluating the behaviour, presence and immersion of observers facing an attention-aware virtual character.

4.1.5 Conclusions

In conclusion, the proposed *expressive filter* is a viewpoint dependent motion editing approach that exploits a number of visual features from an external observer's point of view to drive and warp animations. As a result, this can empower creative artists, but also autonomous characters with means to control the information they convey to an observer.

4.2 INTERACTION FIELDS: 1-to-n interactions

In this section, we describe our system to control agents' position to react to users' presence in the virtual environment, as well as their behaviours. Our choice is to control these positions following a paradigm in crowd simulation techniques, where the behaviour of each crowd agent is controlled through models of local interactions, that dictate how one's motion is influencing others neighbour's motion. However, existing techniques do not allow for rich interactions scenarios. Our effort in PRESENT is thus dedicated to create an animation system with designers' friendly techniques to achieve complex models of interactions in an intuitive way.

Many applications of computer graphics, such as cinema, video games, virtual reality, training scenarios, therapy, or rehabilitation, involve the design of situations where several virtual humans are engaged. In applications where a user is immersed in the virtual environment, the (collective) behaviour of these virtual humans must be realistic to improve the user's sense of presence. As part of realism, expressive behaviour appears to be a crucial aspect. For example, **[Slater 2006]** showed that the expressive behaviour of an audience in VR had a direct impact on the speaker's performances and perception of themselves. This work concerns the motion through an environment

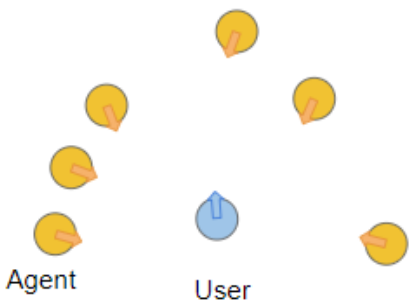
of virtual humans. In the area of crowd simulation, collective behaviours are typically simulated using models based on forces **[Helbing 1995]**, potential fields **[Treuille 2006]** velocity selection **[van der Berg 2011]**, or vision **[Lopez 2019]**. However, those techniques lack expressiveness and do not allow to capture more subtle scenarios (e.g., a group of agents hiding from the user or blocking his/her way), which require the ability to simulate complex interactions. As subtle and adaptable collective behaviours are not easily modelled, there is therefore a need for more intuitive ways to design such complex scenarios.

4.2.1 Context

The category of 1-to-n non-verbal communication scenario considers the interaction between a user and several virtual humans. We are interested in the case where virtual humans are part of the same group and interact with a user, who may feel as part of the group or not, and whose decisions and reactions may be influenced by the group. Several theoretical works regarding social interactions have motivated the design of such scenarios.

According to **[Wilder 1986]**, "persons organise their social environment by categorising themselves and others into groups". Three categories have then be described (1) there is no relation between the perceiver and the group, (2) the perceiver is a member of the group (in group), and (3) the perceiver does not belong to the group and compares with his/her own group (in group/out group). This categorization implies notions such as similarity, homogeneity and differences and would allow individuals to simplify their social environment and predict future social behaviour. In addition, previous works on conversational groups **[Kendon 1990]** have shown that the relative position of the members of the group, which can be considered as an in group situation, are set in a way that each member of the group has a similar shared space with direct and exclusive access. Kendon refers to the F-formation system which describes this "spatial-orientational behaviour" and that can be dynamically adapted so as to include another person in the group. Recently, **[Cafaro 2016]** have used closely related concepts to design believable virtual agents in small conversational groups (static condition) exhibiting nonverbal behaviour. Authors manipulated the relative position of each individual (group formation) as well as interpersonal attitudes (friendly vs. unfriendly). They defined the "in-group attitude" which was directed towards the member of the group and observable by the user as not being a member of the group, and the "out-group" attitude where an overall attitude was expressed towards the users' avatar approaching the group. Results showed that out-group attitude has a main impact on social presence and that proxemics, i.e., interpersonal distance, was affected by the in-group, out-group attitude.

In line with these studies, we would like to extend the design of interactive and expressive virtual humans to dynamic situations. Especially, we aim at modulating the non-verbal expressivity conveyed by virtual humans through their collective motion. Collective motion emerges when individuals achieving the same goal interact with each other. In this context, simulating an expressive and collective motion consists in defining the respective position of each virtual human in time so as to convey a certain amount of unity between the members of the group. The modulation of the group expressivity will be achieved through their motion and final configuration relative to the one of the user. Based on this in and out group concepts as well as group formation principles, we designed in collaboration with CREW, a surrounding scenario that will allow us to manipulate the valence of the situation from the user point of view.



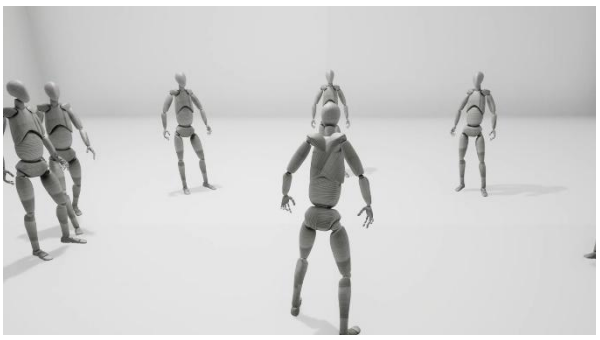


Figure 4.2.1: Illustration of 1-to-n scenarios, where a user is surrounded by a group of virtual agents.

Example of such a scenario is illustrated in Figure **4.2.1** (co-designed with CREW): a user is approaching a group of virtual agents. We elaborate on the reactions and the expressions this group of agents could exhibit. In terms of reactions a large palette of body motions may convey to the user the fact that its presence among virtual humans has triggered events. This could be about making eye-contact, turning bodies toward the user, moving toward the user, moving away from the user, leaving his room to join, etc. Each can be categorised as being neutral, positive or negative (valence). The intensity of such a reaction is adaptable. Synchrony and propagation of reactions will convey the reaction collectiveness. The **4.2** sections is a follow-up of the previous D.3 deliverable where Interaction Fields was already introduced. In the following sections, Interaction Field will be described with more precisions and new resulting scenarios will be detaileds as well as a user study to evaluate Interaction Fields technique.

4.2.2 Overview of the PRESENT system for local interaction design and simulation

Our crowd simulation takes place in a bounded 2D environment $E \subset R^2$ with $m \geq 0$ *obstacles* $\{O_i\}_{i=0}^{m-1}$ and $n \geq 0$ agents $\{A_i\}_{t=0}^{n-1}$. It is common to implement obstacles as simple polygons and to model each agent A_i as a disk with radius r_i. However, our method does not explicitly rely on these implementation choices.

The simulation uses discrete time steps (*frames*). In each frame, every agent A_i computes a new value for its acceleration $\mathbf{a}_i$, which will induce a change in its velocity $\mathbf{v}_i$ and position $\mathbf{p}_i$. As explained in Section **4.3.1**, the process of computing $\mathbf{a}_i$ can be based on algorithms for e.g. path following, collision avoidance, and group behaviour. Each agent A_i also has an *orientation* $\mathbf{o}_i \in S^1$ (a 2D unit vector) that represents the direction that A_i is facing. This work will present *interaction fields* (IFs) as an additional way to control the velocities and orientations of agents. IFs can be used together with other navigation algorithms, as well as independently.

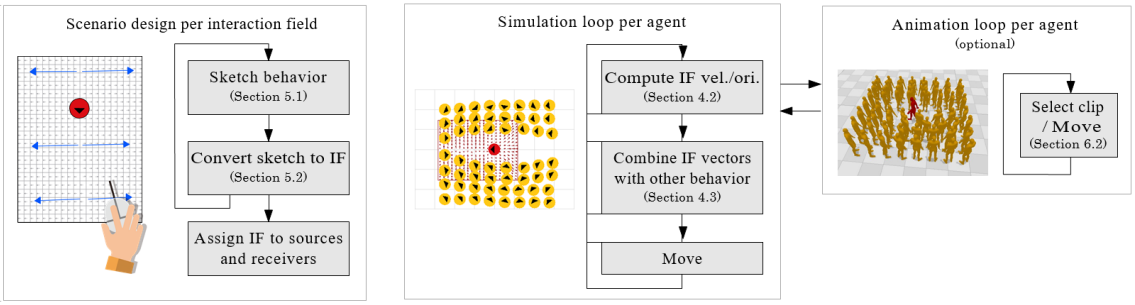


Figure 4.2.2: Outline of a complete simulation system with interaction fields.

Figure **4.2.2** shows an overview of the proposed system that combines IF design, crowd simulation, and character animation. The details per component will be provided throughout this deliverable.

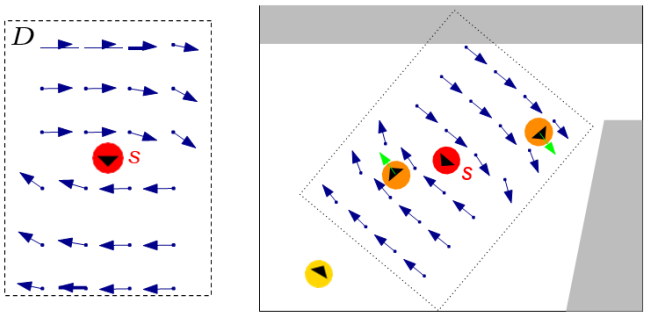
First, a user sketches an IF in an *editing tool*, which we will describe in Section **4.2.4**. The user can draw elements onto a canvas, and this sketch is automatically converted to an IF. The user can then inspect the resulting agent behaviour in the simulation (to be described below) and return to the sketching phase if they wish, until they obtain the desired agent behaviour. To set up a complete simulation scenario, the user should also specify which objects *emit* the IF (as *sources*) and which agents *respond* to it (as *receivers*). Next, the sketched IFs are applied to the *simulation* in the way presented in Section **4.2.3**. (For ease of comprehension, this work will discuss the simulation first and the IF editor second.) In each simulation frame, every agent A_i performs a sequence of tasks. First, A_i should respond to the IFs emitted by nearby sources, which results in an *IF velocity* and *IF orientation* proposed by these IFs. Next, A_i can combine this result with other behaviour such as collision avoidance, resulting in a new velocity and orientation to use. Finally, A_i moves and rotates according to the computed vectors. It is also possible to combine the 2D simulation output with animated 3D characters. Although this is not the focus of our work, it is an

important and non-trivial component for many applications. We will discuss the options and our implementation in Section **4.2.5.2**.

4.2.3 Interaction Field

This section defines the concept of an *interaction field* (IF) and explains how to integrate IFs into a crowd simulation loop. First it is important to define an IF. Overall, a single IF describes either the *velocities* or the *orientations* that agents should use in the vicinity of a particular *source*, which we denote by *s*. We also say that the source *emits* the IF. A source can be an agent, an obstacle, or any other aspect of the environment that should induce a certain kind of behaviour. Because an IF prescribes behaviour around a source *s*, we define it in a Euclidean coordinate system relative to *s*, with *s* located at the origin (0, 0) and oriented towards the negative *y*-axis. Using this, a *velocity IF* with source *s* and domain $D \subset R_2$ is a vector field $VIF_{s,D} : D \rightarrow R^2$ that maps any position $\mathbf{p} \in D$ to a 2D vector $VIF_{s,D}(\mathbf{p})$, indicating the velocity that any agent should use at this position. Likewise, an *orientation IF* is a function: $OIF_{s,D} : D \rightarrow S_1$ that maps any $\mathbf{p} \in D$ to a 2D unit vector $OIF_{s,D}(\mathbf{p})$ that agents should use as their orientation.

Figure **4.2.3** shows an abstract example of an IF. Whenever it does not matter whether an IF concerns velocities or orientations, we will use the notation $IF_{s,D}$. Note that an IF prescribes a vector for *all* points in the domain *D*. Our figures will only show sample velocities for the sake of illustration.



(a) IF definition (b) Mapped onto the environment

Figure 4.2.3: (a) An interaction field is a vector field (shown here in blue) that prescribes velocity or orientation vectors in a domain D around a source object s (here: the red agent). (b) During the simulation, the IF is mapped onto the environment to match the current position and orientation of s. Other agents (in orange) use this mapped IF to compute a velocity or orientation (in green), which they can apply directly or combine with other navigation algorithms. Agents outside the domain (in yellow) are not affected.

4.2.3.1 Applying IFs during the simulation

An IF is defined relatively to a source *s*. During the crowd simulation, the position $\mathbf{p}_s$ and orientation $\mathbf{o}_s$ of *s* may change over time, especially if *s* is an agent. To apply the function $IF_{s,D}$ at runtime, the IF should be translated and rotated to match the current values of $\mathbf{p}_s$ and $\mathbf{o}_s$. Informally, if we see $IF_{s,D}$ as a predefined 'picture' around *s*, we should always line up this picture with how *s* is currently positioned and oriented. We call the result the *mapped IF*, and we denote it by $IF'_{s,D}$. Figure **4.2.3** shows an example.It is important to note that this mapping can remain implicit during the simulation. There is no need to translate and rotate complete IFs at run-time. For any position $\mathbf{q} \in E$, we can easily compute the relevant IF vector $IF'_{s,D}(\mathbf{q})$ by applying the inverse mapping to $\mathbf{q}$. One special case is worth mentioning: if the source *s* is the entire environment E, then $D = E$ as well, and there is no mapping to apply during the simulation ($IF'_{s,D} = IF_{s,D}$). Such an IF is similar to a navigation field **[Patil, 2010]**: it prescribes vectors for the whole environment, and not for the neighbourhood of one specific object.

The purpose of an IF is to model a single type of behaviour around a source, so most simulations will feature multiple IFs at the same time. As part of the scenario's design, the user should specify for each IF which objects *emit* it and which agents *respond* to it. Consequently, it is possible for agents to respond to only *some* IFs and to ignore others, i.e. to model different behaviour for different agents. At any moment in the simulation, each agent A_i should respond to the relevant

interaction fields emitted by nearby sources. To this end, let I = $\{VIF_{s_j,D_j}\}_{k-1}$ be the set of all *velocity* IFs to which A_i can respond *and* that currently have $\mathbf{p}_i$ in their mapped domain. The *IF velocity* $\mathbf{v}_i^{IF}$ for A_i is defined as a weighted average of the vectors that these IFs propose:

$$\mathbf{v}_i^{IF} = \frac{\sum_j VIF'_{s_j,D_j}(\mathbf{p}_i) \cdot w_j}{\sum_j w_j}$$

where the w_j are weights for prioritising between IFs, e.g. to increase the influence of an IF as an agent moves closer to the source. It will often be sufficient to use $w_j = 1$ for all j. For orientation IFs, we define the *IF orientation* $\mathbf{o}_i^{IF}$ for A_i analogously, the only difference being that we explicitly normalise the result.

4.2.3.2 Combining IFs with other simulation components

There are several ways to combine the IF velocity and orientation with other simulation aspects (such as collision avoidance). In most traditional crowd simulations, the behaviour of each agent A_i per simulation frame is already subdivided into multiple steps:

1. Compute a *preferred velocity* $\mathbf{v}_i^{pref}$ that would send the agent towards its goal, possibly with the help of a global path.
2. Compute a *new velocity* $\mathbf{v}_i^{new}$ that stays close to $\mathbf{v}_i^{pref}$ while following local rules for collision avoidance, group behaviour, etc. This yields an acceleration $\mathbf{a}_i := (\mathbf{v}_i^{new} - \mathbf{v}_i)/\Delta t$, where Δt is the length of this simulation frame in seconds.
3. If the agent is currently colliding with other agents or obstacles, compute contact forces **fi** and update the acceleration: **ai** := **ai** + **fi** /m, where m is the agent's mass (usually 1).
4. Update the agent's velocity and position via Euler integration: $\mathbf{v}_i := \mathbf{v}_i + \mathbf{a}_i \cdot \Delta t$, $\mathbf{p}_i := \mathbf{p}_i + \mathbf{v}_i \cdot \Delta t$. Both $\mathbf{v}_i^{new}$ and $\mathbf{v}_i$ are typically clamped to a maximum walking speed v_{max} to prevent unrealistically large velocities.
5. Compute the IF orientation $\mathbf{o}_i^{IF}$. If $\mathbf{o}_i^{IF} \neq 0$, update the agent's orientation as $\mathbf{o}_i := \mathbf{o}_i^{IF}$. Otherwise, keep $\mathbf{o}_i$ unchanged, or update it in a 'traditional' way, e.g. as an average of $\mathbf{v}_i^{pref}$ et $\mathbf{v}_i^{new}$.

To add *velocity* IFs to the system, we have the choice between letting the IF velocity $\mathbf{v}_i^{IF}$ influence an agent's *preferred* velocity (in step 1) or its *new* velocity (in step 2). We will use the first option in our implementation. This allows for an intuitive combination of IFs and collision avoidance, where IFs play an 'advising' role and collision avoidance has the final say. Thus, we use IFs as an alternative way to compute a preferred velocity $\mathbf{v}_i^{pref}$. It is also possible to let $\mathbf{v}_i^{pref}$ depend on IFs *and* on other factors (such as goal reaching) at the same time. We will use this in some of our example scenarios (Section **4.2.6**).

4.2.3.3 Parametric interaction fields

Next, we extend IFs so that they can change during the simulation according to parameters. These parameters may affect both the vectors and the domain of the IF. In other words, a parametric IF encapsulates different 'ordinary' IFs for different parameter values.

A *parametric velocity IF* with l scalar parameters can be described as a function: $PVIF_s : R^l \rightarrow (D^* \rightarrow R^2)$,
where the resulting velocity vectors and the domain D^* now also depend on the l parameter values. A parametric *orientation* IF can be defined analogously. Theoretically, there is no limit on the number of parameters. In this work, though, we create IFs based on user sketches, and we will use at most 1 parameter to keep the design process intuitive. We will now discuss two specific types of parametric IFs that are supported by our sketching tool.

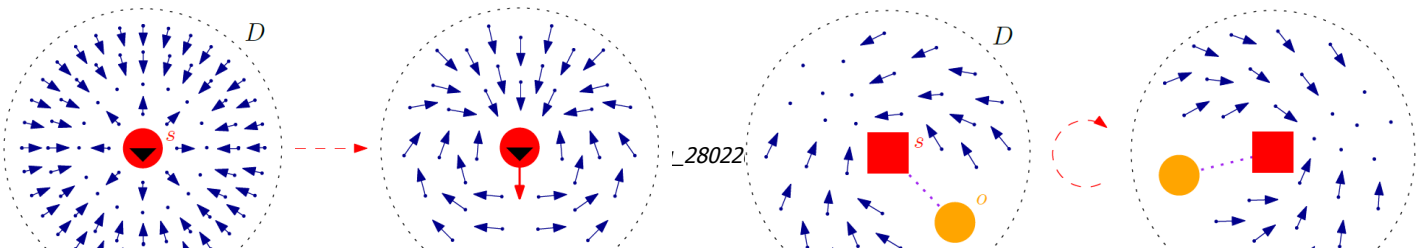


Figure 4.2.4: Parametric interaction fields. (a) Example of an IF that depends on the speed of its source agent s. (b) Example of an IF that depends on the angular relation between the source s and another object o.

One way to specify a parametric IF is to define IFs for a few specific values of a single parameter. These IFs then act as *keyframe IFs* at runtime, and the IF for any other parameter value is defined via linear interpolation between the two nearest keyframe IFs.

For example, Figure **4.2.4(a)** shows a parametric velocity IF with two keyframes, where the parameter is the speed of the source agent *s*. When *s* is standing still, agents will gather around *s* in a circle. When *s* is moving at a certain predefined speed, agents will attempt to follow *s* from behind. There are infinitely many vector fields for the source speeds in-between. During the sim **(a) Based on the source speed** nterpolated field that matches the **(b) Based on an angular relation**

Next to the speed of the source agent, other examples of parameters could be the width or height of a source obstacle (to apply the IF to obstacles of various sizes), the current simulation time (to model behaviour that changes over time), or the local crowd density around an agent (to model density-dependent behaviour). A parameter could also represent an agent's state of mind, such as its hastiness or the amount of panic it experiences.

The simulation never needs to fully compute an interpolated IF. In any simulation frame, an agent only needs to compute a single output vector for each parametric IF in range. Formally, let there be *k* keyframe IFs associated to *k* parameter values: $\{(q_j, KIF_j)\}_{j=0}^{IF}$, ordered by increasing q_j values. Assume for now that all keyframe IFs have the same domain *D*. Given a parameter value *q*, the parametric IF is defined as follows for any point $\mathbf{p} \in D$:

- If $q < q_0$, then $PIF_s(\mathbf{p}) = KIF_0(\mathbf{p})$.
- If $q \geq q_{k-1}$, then $PIF_s(\mathbf{p}) = KIF_{k-1}(\mathbf{p})$.
- Otherwise, $q_j \leq q < q_{j+1}$ for some *j*, and $PIF_s(\mathbf{p}) = (1 - \lambda) \cdot KIF_j(\mathbf{p}) + \lambda \cdot KIF_{j+1}(\mathbf{p})$, where $\lambda = (q - q_j)/(q_{j+1} - q_j)$.

If two subsequent keyframe IFs have *different domains*, we require that any domain in-between can be obtained via linear interpolation as well. For example, this is the case if the domains are both axis-aligned rectangles or both disks. The IF vector $PIF_s(\mathbf{p})$ is then only defined if **p** lies inside the interpolated domain.

The concept of keyframe IFs can be extended to more than one parameter. In that case, each keyframe will be associated to a point in a higher-dimensional parameter space. As mentioned earlier, though, we will focus on single-parameter examples because these are still relatively intuitive for non-expert users to design.

A parameter of an IF could also be a relation between two objects *a* and *b*. Possible examples are the distance between $\mathbf{p}_a$ and $\mathbf{p}_b$, or the angle between the vector $\mathbf{p}_b - \mathbf{p}_a$ and the *x*-axis. As a concrete example, Figure **4.2.4(b)** shows a velocity IF that lets agents move behind a source *s* (typically an obstacle) to hide from another object *o* (typically a specific agent *Ak*). In this specific example, the parameter of the IF is the angle α between $\mathbf{p}_o\,\mathbf{p}_s$ and the *x*-axis. The effect of α is that it simply rotates the IF: it does not affect the IF vectors themselves, but it only changes how the IF is mapped onto the environment. In contrast to regular IFs, this mapping now no longer depends on the orientation of the source *s*. Note that this example can theoretically be combined with keyframe IFs, where the keyframes determine the IF vectors and the angular relation determines the mapping onto E. The result would be a parametric IF with two parameters. In our IF editor, for simplicity, an angular relation between *s* and another object *o* is currently the only object relation that users can draw. A *distance-based* relation between two objects could be implemented with the help of keyframes again: the user specifies which two objects determine the distance parameter, and then they draw keyframes with different distances between these objects.

4.2.4 Sketch-based construction of interaction fields

We have developed a graphical interface in which users can intuitively sketch IFs. This section describes the components of this 'IF editor' and their mathematical meaning for the IF being drawn.

In the IF editor, the user starts by defining a bounding shape D^b, which will serve as the IF domain D. The IF editor then creates a rectangular canvas on which the user can draw. Next, the user can draw elements onto the IF canvas to specify parts of the IF. Section **4.2.4.1** will describe these elements in more detail.

Finally, the program can convert a drawing to a discretized IF: a rectangular grid of vectors, with a user-specified level of precision. This conversion process, which we will describe in Section **4.2.4.2**, uses an interpolation scheme to fill in any regions where the user has not drawn. Of course, the user can adapt this result if desired, by drawing additional elements and then rebuilding the grid.

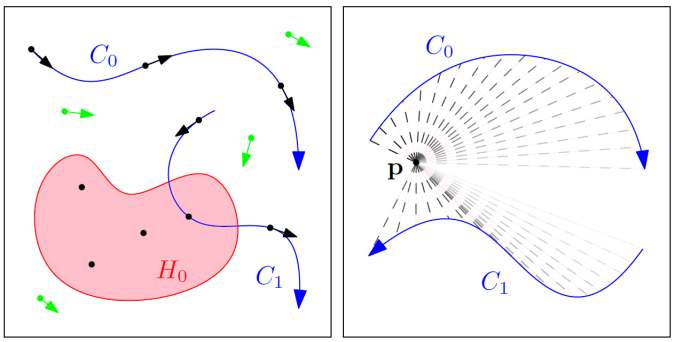
4.2.4.1 Main elements of the IF editor

The user can draw three main types of elements in the IF editor, and a sketch can contain multiple elements of each type.

An **object** is anything that can serve as the *source* of an IF. In our IF editor, it can be an agent (visualised as a disk) or a polygon (which can represent an obstacle or something more abstract). One of the objects on the canvas can be marked as the source object s. Other (non-source) objects can be drawn as a visual aid, or to help define a *parametric* IF. We will explain this further in Section **4.2.4.3**.

A **guide curve** is a curve $C_i : [0, 1] \rightarrow R^2$, with an associated magnitude v_i, that exactly specifies the IF vectors along that curve. For any point **p** that lies on C_i (i.e. if **p** = $C_i(t)$ for some value t), the curve prescribes a vector $\mathbf{c}_i$ with magnitude v_i and direction d/dt $C_i(t)$. Figure **4.2.5(a)** contains two examples of a guide curve. In the final IF, the vector $IF_{s,D}(\mathbf{p})$ at any point **p** will be an interpolation of the vectors proposed by all guide curves. Section **4.2.4.2** will describe this interpolation. In the IF editor, users can draw a guide curve as a piecewise-linear curve or as a freehand curve. For velocity IFs, the default value for v_i is the maximum walking speed of our agents (1.8 m/s), but the user can change this value per curve. For orientation IFs, v_i is fixed to 1 so that C_i proposes unit vectors.

A **zero area** is a region Hj R2 where the IF is 'empty'. For velocity IFs, Hj prescribes the zero vector, meaning that an agent will stand still when it is located inside Hj. For orientation IFs, Hj acts as a hole in the domain D, i.e. as a region where the IF does not propose any specific orientation. Figure **4.2.5(a)** contains one example of a zero area. Note that zero areas always have priority over guide curves, as will be further clarified in Section **4.2.4.2**. In the IF editor, users can draw zero areas with a paintbrush tool, or they can erase IF vectors after converting their sketch to a grid.



(a) IF editor elements (b) Interpolation

Figure 4.2.5: Concepts of the IF editor. (a) The user can draw guide curves (blue) and zero areas (red) to specify IF vectors; example vectors are shown in black. IF vectors for points in-between will be interpolated (green). (b) For any point p outside all zero areas, the IF vector is a weighted average of all vectors along all guide curves, where weights depend on the distance to p.

4.2.4.2 Converting a sketch to an IF

The user draws the elements listed in the previous section onto the canvas. We now describe how to convert this sketch to an IF.

An important aspect of the conversion is to 'fill in' the IF for areas where nothing has been drawn. To infer a meaningful IF vector for any point p in the domain D, we interpolate between all vectors proposed along all guide curves. This interpolation is based on inverse distance weighting **[Shepard 1968]**, a commonly used method for estimating values among scattered data points.
Given a set of c guide curves C = $\{C_i\}_{i=0}^{c-1}$, the estimated IF vector for a point p ∈ D is the following:

$$\mathbf{u}(\mathbf{p},\mathcal{C}) = \frac{\sum_{i=0}^{c-1}\left(\int_0^1 w(\mathbf{p},C_i(t))\cdot\mathbf{v}_i(t)\,dt\right)}{\sum_{i=0}^{c-1}\left(\int_0^1 w(\mathbf{p},C_i(t))\,dt\right)}$$

Here, $w(\mathbf{p}, \mathbf{q}) = \frac{1}{||p-q||^{\kappa}}$, and κ ∈ R^+ is a power parameter that determines how strongly the influence of a curve point decays along with the distance to **p.** Preliminary experiments have led to a use of κ = 1.9 in our implementation. This yields IFs where all vectors are meaningful even with a small number of guide curves. We remind the reader that users can still edit their drawing after the conversion, in case the resulting IF does not match their expectations. In practice, the integrals are approximated by sums, using regularly spaced sample points on each curve. Figure **4.2.5** gives a visual impression of this interpolation scheme. Note that the number of samples does not affect the curve's importance; it only determines the precision by which *C* is approximated.

This type of interpolation has several useful properties. First, if a point **p** lies exactly on a curve point $C_i(t)$, then $\mathbf{u}(\mathbf{p}) = \mathbf{v}_i(t)$, and other curves do not matter (unless **p** is visited multiple times due to curve intersections). Second, if there *are* intersections between or within curves, they do not need to be handled explicitly: the interpolation scheme will simply produce an average vector at an intersection point.

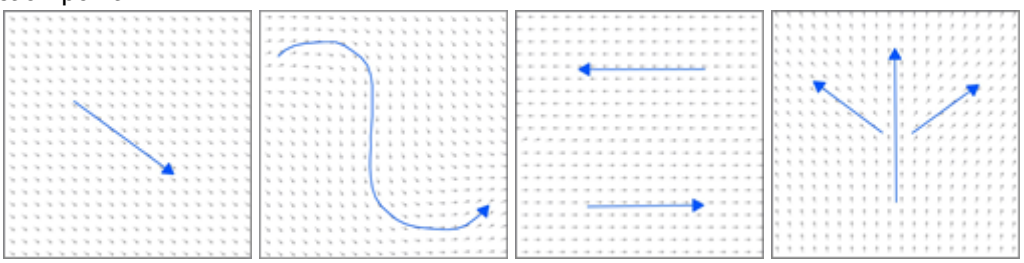


Figure 4.2.6: Examples of guide curves (shown in blue) and their resulting IFs. The gray arrows are the IF vectors (following from the interpolation scheme).

Third, the distance-based decay of a curve's influence is only *relative* and not *absolute*. Moving away from a curve point $C_i(t)$ does not 'shrink' the vector that it proposes; it only reduces the relative weight by which it is taken into account. Figure **4.2.6** shows a number of examples of IFs for different guide curves. Note that the simplest example contains only one straight guide curve, and its IF contains a uniform vector everywhere.

4.2.4.3 Computing the final IF

We now define the overall interaction field that can be obtained from a source *s*, a bounding shape D_b, a set of guide curves C = $\{C_i\}_{i=0}^{c-1}$,and a set of zero areas H = $\{H_j\}_{j=0}^{h-1}$. For a *velocity* IF, the domain *D* is equal to D^b, and the velocity function $VIF_{s,D}$ works as follows for any point **p** ∈ *D*:

- If **p** is inside any zero area H_j ∈ H, then $VIF_{s,D}(\mathbf{p}) = \mathbf{0}$.
- Otherwise, $VIF_{s,D}(\mathbf{p}) = \mathbf{u}(\mathbf{p}, C)$.

For an *orientation IF*, recall that zero areas are treated as holes in the domain. In other words, the domain D is equal to $D^b - \bigcup_{i=0}^{h-1} H_i$, i.e. the set of points that is not covered by any zero area. For any point **p** in the remaining domain D, the final orientation function normalises the interpolated vector to unit length:

$$OIF_{s,D}(\mathbf{p}) = \frac{\mathbf{u}(\mathbf{p},\mathcal{C})}{\|\mathbf{u}(\mathbf{p},\mathcal{C})\|}.$$

The IF editor finally converts a drawing to a grid by computing $IF_{s,D}(\mathbf{p}_i)$ for a set of regularly sampled grid points $\mathbf{p}_i$. The resulting grid of vectors can be used in the crowd simulation.

Recall from Section **4.2.3.3** that a *parametric IF* is an IF that depends on additional scalar parameters. To draw a parametric IF based on *keyframes*, the user can simply draw separate IFs and specify the corresponding parameter values. To draw a parametric IF based on an *object relation*, the user can draw a line-segment connection (a *link*) between the two relevant objects. As mentioned in Section **4.2.3.3**, the IF editor currently only supports a link between the source *s* and another object *o*, and this link implies an angle-based relation between *s* and *o*. We leave other types of links for future work.

4.2.5 IF Implementation

This section defines the concept of an *interaction field* (IF) and explains how to integrate IFs into a crowd simulation loop.

4.2.5.1 Crowd Simulation framework and settings

We have extended *UMANS,* an existing real-time agent-based crowd simulation framework **[van Toll 2020]**, to support interaction fields. The simulation represents each IF by a grid. We compute an IF vector using bilinear interpolation between the nearest grid cells. For parametric IFs based on keyframes, recall from Section **4.2.4.2** that any interpolated IFs are not explicitly computed. However, we sometimes visualise an interpolated IF for the sake of illustration.

In line with other research, our simulations use Euler integration and a fixed frame length $\Delta t = 0.1$ s. Each agent has a disk radius of 0.3 m, unit mass, a preferred speed of 1.3 m/s, and a maximum speed of 1.8 m/s. For contact forces in case of collisions, we use the model by Helbing et al. **[Helbing 2000]** with coefficients $K_{ag} = 5000/80$ for agent forces and $K_{obs} = 2500/80$ for obstacle forces. These values are commonly used in literature when the agents have unit mass.

Next to these overall simulation settings, each agent A_i will use one of the following *behaviour profiles*:

- *IFs-Only*: A_i uses the IF velocity $\mathbf{v}_i^{IF}$ directly as the preferred velocity $\mathbf{v}_i^{pref}$ and as the new velocity $\mathbf{v}_i^{new}$. There is no additional goal reaching or collision avoidance.
- *IFs+GoalReaching*: A_i computes $\mathbf{v}_i^{pref}$ as the average of $\mathbf{v}_i^{IF}$ and a velocity that sends A_i to a pre-defined *goal* at the preferred speed. There is no collision avoidance, so $\mathbf{v}_i^{new} := \mathbf{v}_i^{pref}$.
- *IFs+RVO*: A_i computes $\mathbf{v}_i^{pref}$ using IFs. It then computes $\mathbf{v}_i^{new}$ using the RVO algorithm for collision avoidance **[van den BERG, 2000]**, using the default settings suggested by its authors. Overall, RVO looks for a velocity close to $\mathbf{v}_i^{pref}$ that has a low collision risk.
- *UserControl*: A_i receives $\mathbf{v}_i^{pref}$ and $\mathbf{v}_i^{new}$ directly from a user (e.g. via keyboard or controller input). The agent still receives contact forces in case of a collision. In our figures and videos, user-controlled agents will always be visualised in red.

Of course, and most importantly, each scenario will use its own specific *interaction fields* to model specific types of behaviour, and different agents can emit and receive different IFs.

4.2.5.2 Coupling with character animation.

To visualise our results using animated 3D characters for our supplementary video, we have connected the crowd simulation to the Unity game engine. Synchronising a 2D simulation (of 10

FPS) with an animated 3D scene (of a higher framerate) is not a trivial task. There are at least two options to choose between:

Simulation priority: Let the 3D characters move exactly to the positions produced by the crowd simulation, and use interpolation to fill in the additional animation frames. For body animation, apply a suitable motion clip to each character, accepting possible artefacts such as footsliding.

Animation priority: Use the *output* of the simulation as *input* for an animation system that chooses an appropriate motion clip per character. The chosen animation determines where a character actually moves, and this overrides the simulation results.

The first option is often used in crowd simulation papers, whenever a perfect correspondence to the simulation is more important than animation accuracy. The second option is popular for e.g. controllable characters in games, where the animation should be smooth and natural. It can also help filter out motion for which no animation clips exist, such as fast backward motion or sudden rotations.

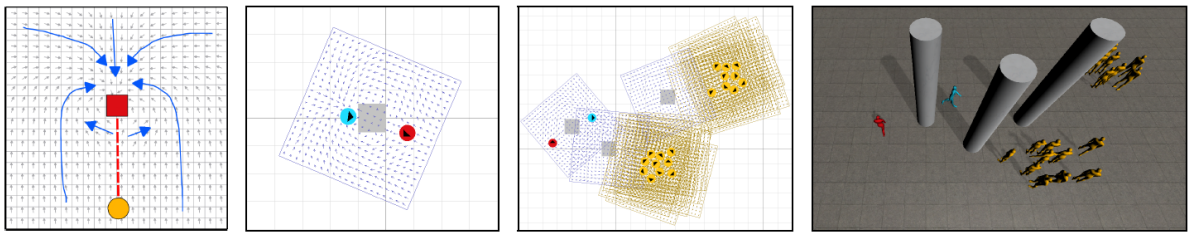
For crowd simulations with IFs, we see use cases for both options. In our supplementary video, we consistently use the second option, based on a Unity plugin for *motion matching* **[Animation 2020]**.

4.2.6 Resulting Scenarios

This section shows the capabilities of interaction fields in a number of example scenarios. Our main purpose is to demonstrate specific features of IFs (such as the use of parameters), and to show that these can easily be combined into more complex scenarios. For each scenario, we will show the input IFs created in our editor, as well as screenshots of the resulting simulation. All simulation screenshots include a grid with cells of 1x1 m, to illustrate the scale of the environment. For visualisation purposes, we also show several IFs mapped onto the environment. Recall from Section **4.2.3.1** that the simulation itself does not need to compute any mapped IFs.

4.2.6.1 Scenario 1: Hide and Seek

Our first scenario uses an angle-dependent parametric velocity IF to let an agent hide behind an object. This IF, shown in Figure **4.2.7(a)**, was drawn using 7 guide curves and a rotation link. In the simplest version of the scenario, one obstacle O emits this IF, with a user-controlled agent A_0 as the linked object. An agent A_1 with the *IFs-Only* profile responds to the IF. As the user moves A_0 around, A_1 automatically hides behind O depending on where A_0 is located. Figure **4.2.7(b)** shows a screenshot of the simulation. In the extended scenario shown in Figure **4.2.7(c)**, we have added several obstacles and agents (with the *IFs+RVO* profile) that all emit the same IF. Consequently, the agent A_1 hides behind whichever object is nearby, treating obstacles and agents in the same way. The extra agents do not respond to any IFs, but they use collision avoidance to make way for the user if necessary. Figure **4.2.7(d)** and the supplementary video visualise the scenario in 3D.



(a) Velocity IF (5×5 m) (b) Simulation (simple) (c) Simulation (extended) (d) 3D visualization

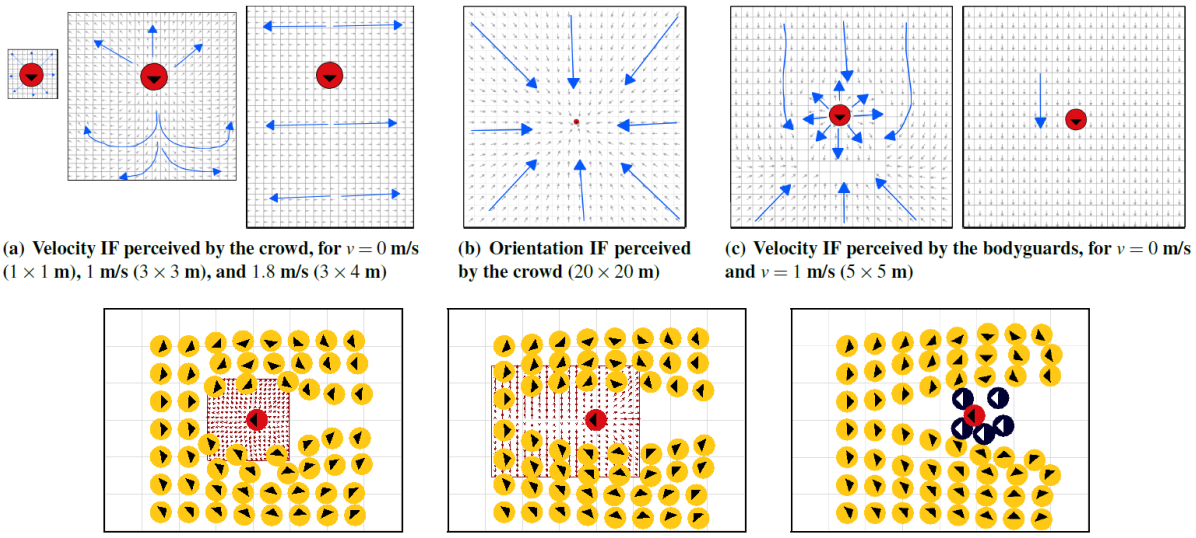
Figure 4.2.7: Results for the Hide and Seek scenario. (a) A velocity IF with a rotation link (red dashed segment) between the source (red) and a second object (orange). Guide curves are shown in blue. (b) A simulation where the blue agent uses this IF to hide from the user-controlled red agent. (c) A simulation where the blue agent can hide behind all obstacles and orange agents, each emitting the same IF. (d) A 3D impression with the two main agents on the left.

4.2.6.2 Scenario 2: VIP in a crowd

Next, we show an example where a crowd makes room for a user-controlled 'VIP' agent. To model this, we make the VIP agent emit two IFs: a parametric velocity IF that depends on the source speed (Figure **4.2.8(a)**) and an orientation IF that makes agents look at the source (Figure **4.2.8(b)**). For the velocity IF, the domain grows and the pushing effect becomes stronger as the speed increases.

The simulation features a small crowd of agents with the *IFs+GoalReaching* profile. The goal of each agent is set to its starting position, so that the agents move back to their old position after the VIP has passed. Figures **4.2.8(d)** and **4.2.8(e)** show how the crowd responds differently depending on the speed of the VIP agent.

Finally, we extend the scenario to include five 'bodyguard' agents with the *IFs-Only* profile. We make the VIP agent emit another velocity IF to which only the bodyguards respond. This IF (shown in Figure **4.2.8(c)**) is parametric again: it lets the bodyguards align with the VIP when it is moving, and (re-)group around the VIP when it stands still. The latter keyframe IF uses zero areas to let the bodyguards stop in a circle around the VIP. Furthermore, the bodyguards themselves also emit the same pushing IF as the VIP. Figure **4.2.8(f)** shows an example of the simulation with bodyguards.



(a) Velocity IF perceived by the crowd, for $v = 0$ m/s (1×1 m), 1 m/s (3×3 m), and 1.8 m/s (3×4 m)

(b) Orientation IF perceived by the crowd (20×20 m)

(c) Velocity IF perceived by the bodyguards, for $v = 0$ m/s and $v = 1$ m/s (5×5 m)

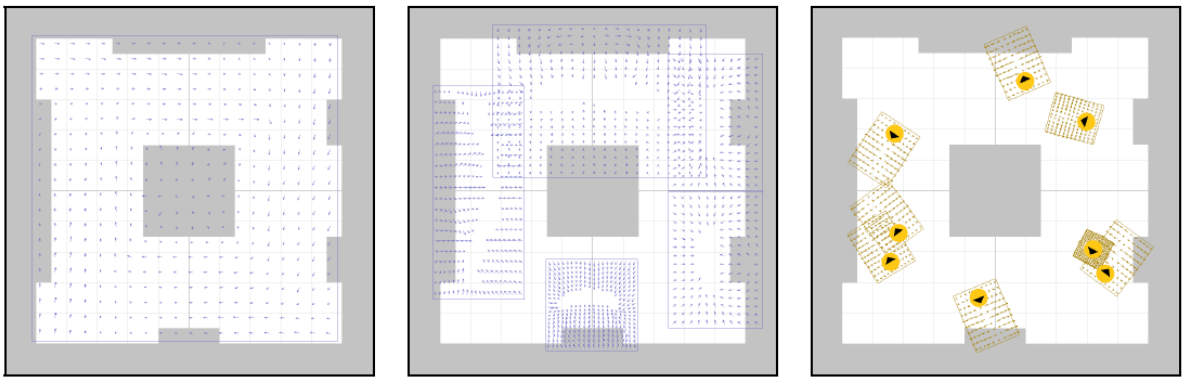
(d) Simulation (VIP speed: 0.6 m/s)

(e) Simulation (VIP speed: 1.8 m/s)

(f) Simulation with bodyguards

Figure 4.2.8: Results for the VIP in a Crowd scenario. (a) Keyframes of the velocity IF used by the crowd. (b) The orientation IF used by the crowd. (c) Keyframes of the velocity IF used by the bodyguards. (d–e) Simulation examples with different speeds for the VIP (in red). The interpolated IF is shown as well. (f) Simulation example with bodyguards (in dark blue). Here, all IFs are omitted for clarity.

4.2.6.3 Scenario 3: Museum



(a) Velocity IF (corridor)

(b) Velocity IFs (paintings)

(c) Simulation

Figure 4.2.9: Results for the Museum scenario. (a) One of the velocities IF for walking around the central pillar. (b) The velocity IFs for all five paintings. (c) Screenshot of the simulation, also showing the parametric IFs around standing and moving agents.

Our final example is a museum scenario where 8 *IFs+RVO* agents move through a corridor and look at paintings. The central pillar emits two velocity IFs for walking around it in a clockwise or counterclockwise way; each agent uses one of these two IFs. Figure **4.2.9(a)** shows the clockwise IF. Next, each painting emits a velocity IF with a zero area that lets agents stand still at a certain distance from that painting; these IFs are shown in Figure **4.2.9(b)**. Each painting also emits an orientation IF that lets agents face the painting; we have omitted these IFs from our figures for clarity reasons.

Also, each agent A_i emits a parametric velocity IF that prevents others from entering A_i's line of sight when it is standing still. This way, others will avoid A_i politely when looking at a painting. Figure **4.2.9(c)** shows a screenshot of the simulation and the agents' IFs.

To let agents switch between walking around and studying a painting, we have added the ability to (de)activate IFs using timers. Whenever an agent enters the domain of a painting velocity IF for the first time, the agent will ignore the corridor IF for a number of seconds. When this timer has passed, the agent ignores the painting IF and uses the corridor IF again, so it continues exploring the museum. However, the *orientation* IFs stay active all the time, so that agents always face paintings that are in range. The timer system is not part of the IF technique itself, and it required some additional modelling/programming effort specifically for this scenario. Note that this is the only example with such an extra system.

4.2.7 User study

To evaluate the efficacy of IFs and the IF editor for non-expert users, we conducted a user study with 22 users who were familiar with computer animation but not with IFs. Our goal was to evaluate how easily they could learn to independently sketch IFs to design specific agent interactions. Please note that the scenarios in this study are different from those in Section **4.2.6**: thus, the user study shows even more examples of IF use cases. We will describe the study only briefly in this section. For in-depth experimental details and results, we refer to the appendix of IF paper **[Colas 2022]**.

All participants completed the study at the institution with the experimenter present, using two 24-inch screens with 1 GUI window to draw the fields and 1 simulation window to see the resulting behaviour. They could always update their IF sketch interactively until they were satisfied with the simulation results. All participants started with a short video-guided training session, where they could freely explore our IF tool and interact with the experimenter. After the introduction phase, participants were asked to sketch IFs for scenarios of increasing complexity.

Each scenario started with a video example and training tasks covering a specific concept, such as controlling the velocity or creating parametric IFs. Each training was followed by one or more evaluation tasks, where participants were asked to create a given scenario based on a number of high-level instructions. There were seven of these evaluation tasks in total. The tasks were designed to require a small number of IFs each (e.g. one orientation IF and one velocity IF), and ordered so as to gradually introduce the users to all IF features (e.g. by saving parametric IFs for last). After each evaluation task, participants reported their satisfaction with their result on a 7-point Likert scale using an online form. They also filled out a usability questionnaire based on SUS **[Brooke 1996]** at the end. The time to complete the study varied between participants but never exceeded 2 hours.

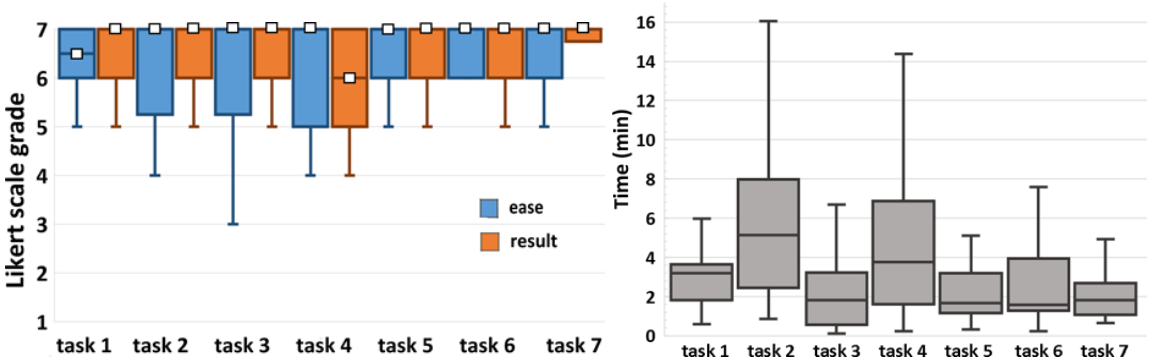


Figure 4.2.10: Results of the user study, per task. Box plots show medians, interquartile ranges and maximum/minimum values (excluding outliers). Top: user ratings for the ease of design (in blue) and satisfaction with the result (in orange). Bottom: completion time for each task (in minutes).

Figure **4.2.10** shows that the participants found the tool easy to use and were very satisfied with the behaviours they designed. The average completion time per task was between 2 minutes 24 (for the fastest task) and 5 minutes 43 (for the slowest task). The final usability ques- tionnaire showed a high average score of 80.6 percentile, which gives our IF editor a A- rating on the Sauro-Lewis Grading scale **[Lewis 2018]**.

Knowing that the IF editor is a simple interface not yet designed for commercial use, this grade shows a very high usability performance. Overall, our study suggests that novice users can easily use the IF editor to sketch agent interactions.

4.3 GAZE BEHAVIOURS: 1-to-n interactions

In this section, we focus on the gaze behaviours of users immersed in VR in relation with the gaze behaviours of virtual agents present in this environment. By doing this, we explore how users react in VR to the gaze of a crowd of virtual agents (1 vs N).

4.3.1 Context

In the current research, we are interested in the initiation of an interaction between virtual humans and a user, and we ask whether the virtual humans' gaze behaviour can be useful in initiating it. Can the gaze trigger a mutual gaze between the user and the virtual human? Can it focus the user's attention on it? Can this constitute the starting of an interaction through nonverbal communication?

Indeed, since nonverbal cues are paramount for humans in their daily social interactions **[Burgoon 2003]**, previous studies have investigated (i) how to reproduce these cues in virtual environments, to make users interact with virtual agents **[Bailenson 2005]**, and (ii) to what extent effects induced by nonverbal communication cues could be reproduced in VR [**Buhler 2018, Narang 2016]**. Regarding gaze and posture cues, Bailenson et al. **[Bailenson 2001]** showed the preservation of the equilibrium between mutual gaze and personal space distance in VR. Additionally, Garau et al. **[Garau 2003]** showed the effect of an inferred-gaze model on perceived quality of communication in VR, compared to a random-gaze model. In line with this, Nummenmaa et al. **[Nummenmaa 2009]** showed the importance of VR users' interpretation of virtual agents' gaze cues in order to avoid collisions when navigating towards another them.

Moreover, regarding gaze cues for nonverbal communication, research outside the field of Virtual Reality (VR) has revealed an effect of gaze in 1 vs N situations, called the "stare-in-the-crowd effect" **[Von Grunau 1995]**. It demonstrated that when multiple faces are exposed to a subject during a visual search task, the detection of faces whose gaze is directed towards the subject is faster (vs. averted ones). It has also been shown that in free visual tasks, visual attention is affected by the presence of directed gaze among averted ones **[Crehan 2015]**.

While these observations were made primarily on the basis of photographic stimuli, in our work we question whether such effects are maintained when a user is immersed in VR. To answer that, we replicate the experiment of Crehan et al. **[Crehan 2015]**, adapting it to VR. We aim at analysing whether the presence of the stare-in-the-crowd effect is retained when observing a virtual crowd, as well as how it is affected by the self-assessed social anxiety levels of the users.

4.3.2 Objective and hypotheses

In this study, we aim at investigating the perception of nonverbal cues when a user is immersed in a virtual environment populated with virtual agents. Our main objective is to study the reaction of users, through their gaze behaviour, when facing a virtual crowd where agents can either look at them or look away.

Previous studies using eye-tracking investigated users' gaze when observing photographs depicting a seated audience. They showed users' preference for gazing at individual subjects in these photographs, whose gaze was directed towards them rather than averted from them, also called the stare-in-the-crowd effect **[Von Grunau 1995].**

According to the literature, VR can be used to depict social interactions with user's behaviours that are close to real-life ones. We are thus interested in the presence of this effect in VR – an environment more adapted to natural human interactions than photographs.

Towards this objective, we propose two hypotheses, H1 and H2. First, we expect that we will observe the same effect as reported in Crehan et al. **[Crehan 2015]** using a series of photographs, but in VR.

• H1: The stare-in-the-crowd effect is preserved with virtual agents in VR.

This means that eye-tracking data will show more saliency characteristics (number of fixations, gaze duration) towards the agent who is directing its gaze towards the user as opposed to when the agent is not looking at the user. Moreover, we also expect the same effect comparing the static averted condition to each dynamic one, i.e., during the phenomena being caught staring and catching someone else staring. However, for these gazing conditions we expect a lower magnitude of effect than for the static directed gaze one, since the time when the agent is looking at the user is shorter. Finally, we are also interested in the comparison between the behaviour of the user in the dynamic conditions as opposed to static directed one.

Moreover, it has been shown previously that social anxiety influences VR users' gaze behaviours towards a virtual crowd, in a similar way to when interacting with humans in physical reality **[Lange 2019, Wieser 2010].** Indeed, a higher social anxiety is typically correlated with a lower rate of mutual eye contact towards directed gazes than in the case of socially non-anxious individuals **[Baker 2002**, **Schulze 2013]**. Therefore, we expect that:

• H2: There will be a negative correlation between the time spent gazing towards the agents who are staring at the user and the user's level of social anxiety.

This suggests a possibility that the stare-in-the-crowd effect will depend on the amount of socially anxious individuals in our test sample. With many users scoring high on social anxiety this effect could disappear completely, thus, it is relevant to explore this relationship. It is also important to note that in some cases lack of gaze towards a socially anxious individual can be more frightening, as it can signal disinterest. However, we created the experimental conditions where the context of the averted gaze would not be interpreted like this.

4.3.3 Experiment

4.3.3.1 Overview

To study the stare-in-the crowd effect in VR, we designed an experiment inspired by Crehan et al. **[Crehan 2015]**, which demonstrated the presence of this effect using photographs. In our

experiment, the user is asked to observe a virtual crowd where the gazes of the virtual agents are manipulated according to a series of target conditions/behaviours, similarly to Crehan et al. **[Crehan 2015]**. These crowd gaze conditions are:

• Averted - A: no virtual agent looks towards the human user during the observation task (see Figure **4.3.1(1)**);

• Directed - D: one virtual agent, referred to as the "active agent", stares at the user at the beginning of the observation task and will keep staring at him or her until the end of the task, while no other virtual agent stares at the user (see Figure **4.3.1(2)**);

• Averted-then-Directed - AD: no virtual agent looks towards the user at the beginning of the observation task, but the active agent will start staring at the user once looked at and will continue to stare until the end of the task (see Figure **4.3.1(3)**);

• Directed-then-Averted - DA: the active agent stares at the user at the beginning of the observation task, but will stop once looked at, while no other virtual agent staring (see Figure **4.3.1(4)**);

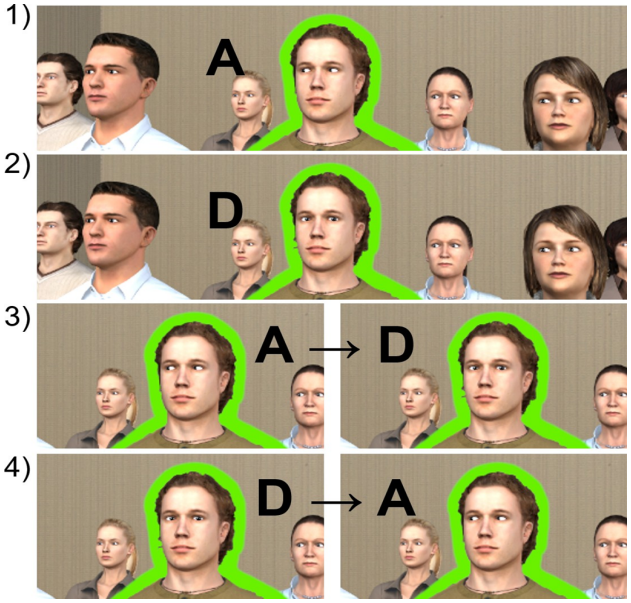


Figure 4.3.1: Our four crowd gaze conditions (active agent in green): 1) averted gaze - A, 2) directed gaze - D, 3) averted-then-directed gaze - AD, and 4) directed-then-averted gaze - DA.

We asked users to observe the virtual crowd, without telling them to actively search for directed or averted gazes. Such indications are different with respect to some previous studies **[Colombatto 2020**, **Doi 2007**, **Ramamoorthy 2019**], but consistent with Crehan et al. **[Crehan 2015**, **Crehan 2021**]. In line with Crehan et al. **[Crehan 2015]**, we also propose to use an eye-tracking system to evaluate the users' gaze behaviours instead of using a search task, which would be less natural. However, opposite to previous studies **[Colombatto 2020**, **Cooper 2013, Crehan 2015**, **Crehan 2021, Doi 2007**, **Framorando 2016, Ramamoorthy 2019**], we use a crowd of virtual agents in VR as visual stimuli (see Figure **4.3.2**).

4.3.3.2 Virtual environment and stimuli creation

The virtual environment we used here, shown in Figure **4.3.2**, was created using Unity 2021.2.0b9. It is composed of a room, resembling a classroom or a conference room, equipped with standard pieces of furniture as well as individual chairs placed on a wooden stage. Virtual agents (our virtual crowd) are seated on these chairs, like an audience, 1m away from the user at the minimum. All virtual agents are clearly visible to the user, without any occlusion between their heads. The wooden stage hides part of the virtual agents' bodies, so as to make the user focus on their faces. Similarly to the photographic stimuli used in Crehan et al. **[Crehan 2015]**, the virtual audience was slightly (10°) oriented to the right, as well as the user (20°). Moreover, the user was placed slightly on the

right of the virtual crowd. Such position/orientation choice was chosen for two main reasons: (i) to have all the virtual characters in the user's initial field of view, since they appear at real scale (1:1)); and (ii) to allow virtual agents to look towards the user's position without needing to rotate their head, but only their eyes, while maintaining a natural gaze behaviour (e.g., horizontally rotating the eyes a maximum of 30° with respect to the head). These two aspects ensured that all virtual agents could be easily viewed, and that gaze orientation would be the main difference between them, with different gaze behaviours but similar head orientation, thus avoiding such kind of bias **[Marschner 2015]**.

We considered eleven virtual agents' models from the Microsoft RocketBox adult avatars collection **[Gonzalez-Franco 2020]**, including six females and five males. Figure **4.3.2** shows this virtual audience from top and from the user's point of view. Additionally, we placed another male model in front of the crowd, as if he was giving a presentation to them. However, no speech could be heard by the user, it was only to provide a social setting, and to justify why the crowd was looking towards a common point away from the user. To increase the naturalness of agents' behaviours, we applied simple blinking animation on their eyes. Then, a specific gaze behaviour was chosen according to the condition at hand, A, D, AD, or DA, as described in Section **4.3.3.1**.

The virtual agent staring at the users, referred to as the "active agent", is chosen randomly among nine of the eleven agents of the crowd. These nine agents are highlighted with red dots in Figure **4.3.2.** This choice was driven by the need to have a balanced distribution of active gazing agents across the user's field of view, as suggested in **[Doi 2007**, **Palanica 2011a, Palanica 2011b]** to test any potential position effects on the results.

Figure 4.3.2: Virtual scene; the user faces eleven agents listening to a speaker standing behind the user. The inset shows the user's viewpoint during the observation task. Active agents are noted by red dots

It should be noted that for coherence with the other conditions and to enable a consistent comparison of our metrics (see Section **4.3.3.6**), an active agent is also chosen in condition A (no agent looks at the user), although it does not behave differently to the rest of the crowd.

For agents' gaze behaviours we built a gaze mechanism, favouring eye rotations over head and torso rotations, while providing realistic results - e.g., the maximum angle of eye rotation was 30°. This way we could create realistic eye gaze for all positions and conditions.

Finally, in conditions AD and DA, where the agent's gaze dynamic behaviour (from averted to directed or vice-versa) is triggered by the user, we introduced a time limit as suggested by Crehan

et al. **[Crehan 2015]**. If the user has not looked at the target agent within half of the total trial time, the agent's gaze changes anyway, without waiting for the user's gaze. Following Crehan et al. **[Crehan 2015]**, each trial repetition (i.e., the user looking at the crowd) lasted 16 seconds. After this time, the environment fades out and fades in again to the same scene but featuring a new gazing behaviour and active agent (see Section **4.3.3.5**).

4.3.3.3 Participants and Apparatus

30 participants (8 females, 22 males; age: aver. 30, SD: 9.5; VR experience from 1 to 5: aver. 3.4, SD: 1.4; computer games experience from 1 to 5: aver. 3.5, SD: 1.5) took part in our experiment, all with normal or corrected-to-normal vision. They voluntarily participated in the experiment and received no compensation for it. The study complied with the Declaration of Helsinki and was approved by the local ethical committee (COERLE). Participants were asked to sit on a standard chair throughout the whole experiment, and to wear the VR head-mounted display FOVE, which has an embedded eye-tracking system. Its field of view for a user is 100°, as well as the one of eye-tracking. Its advertised spatial tracking accuracy is less than 1°, and its maximum eye-tracking sampling rate is 120 Hz.

4.3.3.4 Data collection

We collected two types of data: (i) continuous user's gaze behaviour during the VR experience, and (ii) social personality data after it.

For (i), gaze behaviour was collected using the embedded eye tracking system of the VR headset. At each frame, the user's gaze information was logged along with the timestamp and the current gaze condition of the virtual crowd (A, D, AD, or DA). This gaze information was indicating the presence or the absence of a hit on the head of the "active agent", computed using the 2D screen position of the VR user's gaze and the current 2D scene viewed by the user.

For (ii), information about users' social anxiety was collected after the experiment through a questionnaire. We used the standardised questionnaire based on the Liebowitz Social Anxiety Scale **[Liebowitz, 1987].** This one allows for the evaluation of social anxiety through self estimation of the levels of fear and avoidance of a person in determined social situations. A score can be computed from the answers, ranging from 0 (not socially anxious) to 144 (very socially anxious).

4.3.3.5 Experimental procedure

First, an informative document about the study was given to the users, along with the informed consent form and oral explanations to answer any questions. Once ready, users were seated on a chair and equipped with the FOVE headset. A calibration of the eye-tracking system was performed to ensure the quality of gaze data collection.

Then, the users were immersed in our virtual environment for a brief training phase, where they had time to familiarise with the environment and setup. During this phase, all agents of the virtual crowd were looking at the virtual speaker, were not changing their gazing behaviour over time, and random agents would be blinking in the crowd. Users were free to look both at the crowd and behind them to see the virtual speaker – which was not talking, to understand the context of the scene. It was explained to them that their task would be to face and observe the virtual audience, and to not look at the virtual speaker after the training phase. No information about gazing behaviours or any other specific tasks to complete were provided.

After this training phase, users were asked to perform 72 trials of this observation task, each lasting 16 seconds. All users were exposed to the same trials i.e., all the tested conditions described in Section 4.3.3.1. Each combination of "gaze condition/behaviour" per "active virtual agent positioning" was shown twice to each user, leading to: 4 gaze behaviours 9 possible active agents 2 repetitions = 72 trials in total. In order to make it possible for the user to rest during the experiment, the trials were ordered in 3 blocks, with equal number of gaze conditions presented in each block of 24 trials, as well as the distribution of the active virtual agent. Order of active agents was randomised inside each block. In averted conditions, an agent was chosen randomly and the position of these agents was balanced with the agents in the other conditions, which all include a directed gaze. Additionally, virtual agents' models were randomly switched between all eleven

positions, so that the appearance of the models would not influence the results. A 3-seconds black screen was displayed to the users between each trial. During this pause, users were asked to re-position their head and gaze orientation towards the top-centre of the screen, by looking at a small geometric shape. This was done to ensure the same initial point for the user's gaze at each trial. Users were notified that the trials would be divided into three blocks of 24, so as to allow them to rest and remove the headset between each block to minimise fatigue. In addition, such breaks were also used to re-calibrate the eye-tracking system to ensure data quality. If needed, users could also stop within a block.

Finally, users were asked to fill a post-experiment questionnaire with the social anxiety questions, along with demographic ones (age, gender, experience with VR and games) and a free comment section.

4.3.3.6 Metrics

From the eye-tracking collected data, we computed different metrics related to the users' gaze towards the active agent of the crowd. Gaze activity was split between saccades when such activity was shorter than 150 ms, and fixations when it was longer **[Manor 2003, Westheimer 1954]**. For each trial, we considered the following metrics in line with Crehan et al. **[Crehan 2015]:**

• Dwell time: the total time spent looking at the active virtual agent;

• Fixations count: the total number of fixations on the active virtual agent;

• First fixation time: the time of the first fixation on the active virtual agent, counted from the beginning of the trial;

• First fixation duration: the length of the first fixation;

• Second fixation time: the time of the second fixation on the active virtual agent, counted from the beginning of the trial;

• Second fixation duration: the length of the second fixation.

All the above metrics are used to identify the stare-in-the-crowd effect, particularly the dwell time and fixation count metrics that are computed even in absence of multiple fixations on the active agent. The analysis of first and second fixations are also important to better understand user's gaze behaviours, even though they are not always present in stare-in-the-crowd related studies. But they are particularly relevant for the dynamic conditions that we included here, where the user's first fixation on the active agent triggers the change in its gaze behaviour (from averted to directed or vice-versa).

4.3.4 Results and discussion

4.3.4.1 Gaze behaviours

According to our objectives and hypotheses, we focused on five comparisons, related to three cases: (1) the stare-in-the-crowd effect in static conditions, (2) catching someone else staring and (3) being caught staring phenomena, in line with Crehan et al. **[Crehan 2015]**. For (1), we compared the averted to the directed gaze conditions – A vs. D. Then, we compared each static condition with each dynamic. For (2), averted versus averted-then-directed – A vs. AD, and directed versus averted-then-directed – D vs. AD. For (3), the averted versus directed-then-averted – A vs DA, and directed versus directed-then-averted – D vs. DA. For pairwise comparisons, we ran dependent paired samples t-tests on the six metrics we described in Section **4.3.3.6** as continuous variables. Such tests guarantee conservative results in the comparison between different gaze conditions. The normal distribution assumption was verified for 25 of our 30 dependent paired samples when running a Shapiro-Wilk test: we ran Student's t-tests for these samples, and Wilcoxon signed rank tests for the remaining ones. Due to our multiple comparison design, we conducted a Bonferroni correction which changed our target significance level from $\alpha=0.05$ to $\alpha=0.00166$.

Results are shown in Tables **4.3.1** to **4.3.5**. For each metric, they contain the means and standard deviations, along with significance level, plus statistics and effect size (both when doing Student's t-test). They are shown by comparison of pairs, in Table **4.3.1** for conditions A vs. D, Table **4.3.4** for A vs. AD, Table **4.3.5** for D vs. AD, Table **4.3.2** for A vs. DA, and Table **4.3.3** for D vs. DA. These results are based on the averages obtained by each user across all trials that share the same gazing conditions regardless of position, i.e., 18 in total for each condition. In these tables, a symbol * indicates a p-value <0.00166, ** a p-value <0.00033, and *** a p-value <0.00003.

Comparison A vs D interpretation.

As shown in Table **4.3.1**, p-values from the metrics dwell time, fixation count, first and second fixation durations were all significant, with higher values on the directed condition, which are all indicators of the presence of a stare-in-the-crowd effect. We also expected users to spot the active agent in the directed gaze condition sooner, which should be reflected through significantly earlier first fixation time. Such results have been reported and used to confirm the presence of a stare-in-the-crowd effect in previous studies with drawing or photographic stimuli **[Ramamoorthy 2019**, **Von Grunau 1995]**. However, in our experiment, first fixation time results do not reveal such a significant difference. For this metric, we discuss our results later in this section (see Active agent's position effect and Table **4.3.7** with its analysis). Based on the expectations of the stare-in-the-crowd effect, our results nonetheless show a significantly earlier second fixation time on the directed condition compared to the averted one, following the trend expected for the first fixation time. Figure **4.3.4(1)** summarises the comparison between the results on averted and directed conditions and its interpretation for the stare-in-the-crowd effect.

	Averted	Directed			
Metric	Mean (SD)	Mean (SD)	p-value	t	η_p^2
Dwell time	504 (175)	1570 (864)	<0.00001 ***	-6.75	0.61
Fixation count	1.15 (0.29)	2.35 (1.03)	<0.00001 ***	-6.45	0.59
1st fix. duration	332 (77)	552 (185)	<0.00001 ***	-5.61	0.52
1st fix. time	5173 (1213)	4969 (1402)	0.53119	0.63	0.014
2nd fix. duration	407 (282)	554 (214)	0.00158 *	wilc.	wilc.
2nd fix. time	8602 (1395)	6861 (1785)	0.00031 **	4.09	0.37

Time and duration in ms.

Table 4.3.1: Gaze metrics results - comparison of A vs. D conditions

When comparing with Crehan et al. **[Crehan 2015]**, we found the same results on all our metrics, except for the significantly longer duration for the first fixation in the directed condition in our experiment. Nonetheless, this result is in line with other previous studies **[Ramamoorthy 2019**, **Von Grunau 1995]** and the stare-in-the-crowd effect by definition. In addition, it could be explained by a stronger effect of VR to capture attention with directed gazes, as suggested by our larger effect size results for the other metrics, compared to Crehan et al.'s ones **[Crehan 2015]**.

Comparison A vs DA interpretation.

	A	DA			
Metric	Mean (SD)	Mean (SD)	p-value	t	η_p^2
Dwell time	504 (175)	808 (363)	0.00005 **	-4.78	0.44
Fixation count	1.15 (0.29)	1.56 (0.56)	0.00015 **	-4.37	0.40
1st fix. duration	332 (77)	483 (165)	0.00003 ***	-4.92	0.45
1st fix. time	5173 (1213)	4847 (1307)	0.32902	0.99	0.03
2nd fix. duration	407 (282)	374 (95)	0.34921	wilc.	wilc.
2nd fix. time	8602 (1395)	7773 (1724)	0.05175	2.03	0.12

Time and duration in ms.

Table 4.3.2: Gaze metrics results - comparison of A vs. DA conditions

As shown in Table **4.3.2**, first fixation duration, dwell time and fixation count were significantly different between averted and directed-then-averted conditions, with higher values in the latter. In contrast, second fixation duration and second fixation time were not significantly different between these conditions. First fixation time metric did not show significant differences either; for this result, see the discussion point Active agent's position effect later in this section. The results for all the other five metrics might be understood and explained according to the procedure of the directed-then-averted gaze trial. Indeed, in this condition, once the first fixation had started on the active agent, users could observe a dynamic gaze change. This might have captured their attention and could explain the fact that they stared significantly longer towards the active agent during the first fixation. After, the active agent entered the averted gaze condition: this could explain why the directed-then-averted condition results of second fixation duration and second fixation time were not significantly different compared to the averted condition ones. Finally, dwell time and fixation count were nevertheless significantly higher in the dynamic condition, which could be explained by the multiple rechecks by users towards the active agent during the remaining time of a trial, to see if the agent would look at them again Figure **4.3.4(2)** summarises the comparison between the results on the averted and directed-then-averted conditions and its interpretation in relation with the stare-in-the-crowd effect and the effect of dynamic gaze changes.

In addition, when comparing our results to the ones of Crehan et al. **[Crehan 2015]**, both studies found similar effects, except that in their case instead of finding a significant difference for the first fixation duration, they found it for the second fixation one.

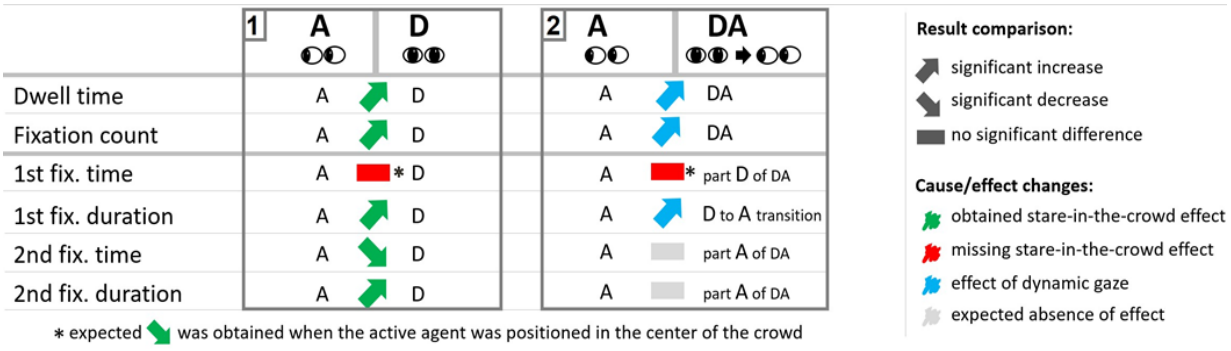


Figure 4.3.4: Summary of results for two representative cases of comparison: 1) A vs D reveals the presence of the stare-in-the-crowd effect in VR, and 2) A vs DA reveals effects of dynamic gazes

<u>Comparison D vs DA interpretation.</u>

	Directed	Directed-then-Averted			
Metric	Mean (SD)	Mean (SD)	p-value	t	η_p^2
Dwell time	1570 (864)	808 (363)	<0.00001 ***	6.83	0.62
Fixation count	2.35 (1.03)	1.56 (0.56)	<0.00001 ***	6.17	0.57
1st fix. duration	552 (185)	483 (165)	0.07638	1.84	0.10
1st fix. time	4969 (1402)	4847 (1307)	0.69058	0.40	0.01
2nd fix. duration	554 (214)	374 (95)	0.00016 **	4.32	0.39
2nd fix. time	6861 (1785)	7773 (1724)	0.05246	-2.03	0.12

Time and duration in ms.

Table 4.3.3: Gaze metrics results - comparison of D vs. DA conditions

As shown in Table **4.3.3**, dwell time, fixation count and second fixation duration were significantly different between directed and directed-then-averted conditions, with lower values in the latter. These results confirm the stare-in-the-crowd effect: indeed, in a directed-then-averted condition, once the first fixation on the active agent had started, its gaze remained averted, thus significant differences are consistent with the ones observed for these metrics on the averted vs. directed comparison. In a similar way, in both conditions, the agent's gaze was directed before the first fixation started, which can explain the absence of the significant difference for the first fixation time. After that, for the first fixation duration and the second fixation time, the absence of significant difference between these two conditions is consistent with the interpretation given for averted vs. directed-then-averted conditions and could thus be explained the following: the gaze change of the active agent that occurred at the beginning of the first fixation could have captured the VR users' attention at a level not significantly different to the one caused by a directed gaze for the first fixation in terms of duration, and could have nonetheless made them check back towards this agent as soon as in the directed gaze condition, therefore through an early second fixation on it.

In addition, compared to our results, Crehan et al. **[Crehan 2015]** did not observe the stare-in-the-crowd effect in all the metrics, since they found no effect of dwell time or fixation count. However, as we did, they found a significant difference for the second fixation duration. Our differences may come from the specifics of our setup, e.g. using VR that adds depth and space information, unlike photographs.

<u>Comparison A vs AD interpretation.</u>

	Averted	Averted-then-Directed			
Metric	Mean (SD)	Mean (SD)	p-value	t	η_p^2
Dwell time	504 (175)	1503 (789)	<0.00001 ***	wilc.	wilc.
Fixation count	1.15 (0.29)	2.18 (0.79)	<0.00001 ***	-7.24	0.64
1st fix. duration	332 (77)	544 (176)	<0.00001 ***	-6.22	0.57
1st fix. time	5173 (1213)	5371 (1229)	0.87121	wilc.	wilc.
2nd fix. duration	407 (282)	644 (226)	0.00011 **	wilc.	wilc.
2nd fix. time	8602 (1395)	6978 (1544)	0.00032 **	4.08	0.36

Time and duration in ms.

Table 4.3.4: Gaze metrics results - comparison of A vs. AD conditions

As shown in Table **4.3.4**, dwell time, fixation count, first fixation duration, second fixation duration and second fixation time were significantly different between conditions averted and averted-then-directed, with lower value for the second fixation time and higher values for the other

metrics in the averted-then-directed condition. These results confirm the stare-in-the-crowd effect in VR: indeed in an averted-then-directed condition, once started the first fixation on the active agent, its gaze remains a directed gaze, therefore significant differences are consistent with the ones observed for these metrics on averted vs directed comparison. Moreover, first fixation time was not significantly different between the two conditions, which is coherent since in both conditions the active virtual agent starts with an averted gaze.

In addition, in comparison with Crehan et al.'s results **[Crehan 2015]**, we found similar results, except the fact that they did not observe a higher level of first fixation duration in the dynamic condition. Similarly, our differences may come from the specifics of our setup, e.g. using VR that adds depth and space information, unlike photographs.

Comparison D vs AD interpretation.

	Directed	Averted-then-Directed			
Metric	Mean (SD)	Mean (SD)	p-value	t	η_p^2
Dwell time	1570 (864)	1503 (789)	0.45729	0.75	0.02
Fixation count	2.35 (1.03)	2.18 (0.79)	0.15961	1.44	0.07
1st fix. duration	552 (185)	544 (176)	0.85695	0.18	0.00
1st fix. time	4969 (1402)	5371 (1229)	0.16703	-1.41	0.06
2nd fix. duration	554 (214)	644 (226)	0.13021	-1.56	0.08
2nd fix. time	6861 (1785)	6978 (1544)	0.75169	0.32	0.00

Time and duration in ms.

Table 4.3.5: Gaze metrics results - comparison of D vs. AD conditions

As shown in Table **4.3.5**, dwell time, fixation count, first fixation duration, second fixation duration and second fixation time were not significantly different between the two conditions. These results are coherent since in the averted-then-directed condition, once the first fixation on the active agent had started, its gaze remained directed, i.e., with a gaze similar to the directed condition. Finally, for the first fixation time metric, there was no significant difference; see the paragraph Active agent's position effect for the discussion of this result. In addition, all our results are coherent with Crehan et al.'s ones **[Crehan 2015].**

Active agent's position effect.

In addition to these results that average all the data by gaze condition, our metrics can also be computed based on the averages obtained by each user across the trials that share both the same viewing conditions and the same position of the "active agent" in the crowd and therefore in the user's field of view – 2 repetitions in total for each condition. Due to the variability of the number of fixations across conditions and users, dwell time and fixation count metrics were preferred here over fixation time and duration metrics, since the former ones can always be computed even when no fixations occurred on the expected agent during the trials – in that case, missing values would be reported for the other metrics when computing averages. For these nine position conditions, we only compared the averted and directed gaze conditions here, as they were the most representative ones for the evaluation of our hypothesis H1. For our two metrics, the normality assumption could not be verified for all our dependent paired samples, thus Student's t-tests or Wilcoxon signed rank tests were run depending on the case. Due to our multiple comparisons, we conducted a Bonferroni correction that changed our target significance level from $\alpha=0.05$ to $\alpha=0.00555$. Table **4.3.6** shows the results of these comparisons for the dwell time on the left, and for the fixation count on the right. In the tables, a symbol * indicates a p-value <0.00555, ** a p-value <0.00111, and *** a p-value <0.000111.

	Dwell time metric	Fixation count metric
Paired samples for t-test	p-value	p-value
Left-close A - Left-close D	0.00203 *	0.00078 **
Left-middle A - Left-middle D	0.71318	0.43556
Left-far A - Left-far D	0.00902	0.05810
Centre-close A - Centre-close D	<0.00001 ***	<0.00001 ***
Centre-middle A - Centre-middle D	<0.00001 ***	<0.00001 ***
Centre-far A - Centre-far D	0.00137 *	0.00399 *
Right-close A - Right-close D	0.00001 ***	<0.00001 ***
Right-middle A - Right-middle D	0.00006 ***	0.00074 **
Right-far A - Right-far D	0.00008 ***	0.00021 **

Table 4.3.6: Metrics comparison for each position A vs. D - dwell time and fixation count

These results show an effect of the active agent's position on the dwell time and fixation count results when comparing averted and directed conditions. For seven out nine positions a significant difference was found between these two conditions for this metric, revealing the presence of a stare-in-the-crowd effect; in contrast, for the middle and far left positions, no significant difference was found. Nonetheless, this result is in line with previous studies that discussed the real existence of a stare-in-the-crowd effect across any stimuli positions **[Cooper 2013]** and any position in the user's field of view **[Palanica 2011a, Palanica 2011b]**. In addition, we found that this absence of significant difference between averted and directed condition was due to a larger time spent on the middle/far left field of view on the averted gaze conditions rather than to a lower one on the directed condition, compared to the results obtained on other positions. This could be explained by a leftward bias of humans during a visual exploration of a scene, as described in the literature [8, 18, 36]. Finally, this difference on the left may also have been caused by our experimental stimuli. Indeed, in our experiment the averted gazes of the virtual crowd were always towards a distractor – our virtual speaker – positioned at the left of the user, meaning that the majority of the virtual crowd was looking in that direction. Yet, in their study about the stare-in-the-crowd effect, Palanica et Itier **[Palanica 2011b]** found a congruency effect of the averted gazes on the user's gaze behaviour, in the sense that active agents whose positions were in the direction signalled by averted gazes were detected faster. Similarly, Sun et al. **[Sun 2017]** also found an effect of the perceived direction of the gaze of the virtual crowd on users' gaze behaviour, where users tend to look towards the same direction that they perceive when the majority of the crowd is looking towards one particular direction – in our case to the left.

We also wanted to test if the active agent's position could have affected other metrics than dwell time and fixation count. We found an effect for the first fixation time on the trials where the active agent was in the centre – without distinction of depth i.e. 6 trials in total for each gaze condition (3 positions by left/central/right zone * 2 repetitions for each user). For these data samples, the normality assumption could not be verified for all our dependent paired samples, thus Student's t-tests or Wilcoxon signed rank tests were run depending on the case. Due to our multiple comparisons, we conducted a Bonferroni correction that changed our target significance level from $\alpha=0.05$ to $\alpha=0.016$. Table **4.3.7** shows the results of these first fixation time comparisons, with one column for each gaze comparison studied, one line for each position zone – left/central/right, and one final line with the p-value previously obtained with the global data without position distinction. In this table, a symbol * indicates a p-value <0.0160, ** a p-value <0.0033, and *** a p-value <0.0003.

Gaze comp.	A vs D	A vs DA	D vs DA	A vs AD	D vs AD
Position	p-value	p-value	p-value	p-value	p-value
Left	0.0745	0.0792	0.8816	0.0293	0.4587
Central	0.0027 **	0.0027 **	0.8340	0.3765	0.012 *
Right	0.9018	0.2103	0.2421	0.7535	0.6181
All	0.5312	0.3290	0.6906	0.8712	0.1670

Table 4.3.7: First fixation time metric - comparisons by pair of gaze conditions and across position zones

These comparison results give new insights on the first fixation time metric, and allow for new interpretations about the effect of gaze conditions on it. First the data where all positions are gathered show no significant differences between any gaze conditions, as well as the results considering only left or right positions. However, data related to central positions reveal different results with: 1) the presence of significant differences for the comparisons between averted and directed gaze conditions (A vs. D), averted and directed-then-averted ones (A vs. DA), and directed and averted-then-directed ones (D vs. AD), and 2) the absence of significant differences for the other comparisons. Such results are interesting because they are the ones that were expected according to the stare-in-the-crowd effect: indeed, before the first fixation, the three comparisons present in 1) are equivalent to an averted vs. directed gaze comparison, whereas for the two comparisons of 2), gazes are the same ones in both conditions for these two comparisons (two averted, or two directed). These results confirmed the presence of a stare-in-the-crowd effect in VR, here regarding the results for the first fixation time metric for active agents in central positions.

We may have found this effect only in the central position because of visual differences between VR and photographs. Photographs resolution allows for high-quality display of a crowd in a narrow field of view, about 30° for a user looking at a computer screen. In contrast, in our VR setup the total field of view was larger for the user (the 100° of the FOVE headset), but, because of resolution issues and the scale 1:1 for the agents used to provide immersion in VR, more space was required for each agent. Therefore, it could explain why previous results are equivalent to our central part results.

4.3.4.2 Gaze behaviours and social anxiety

To investigate whether users with a higher level of social anxiety were less likely to gaze towards agents who are gazing at them, we computed correlations between the final score on the social anxiety questionnaire, i.e., the Liebowitz Social Anxiety Scale, and our gaze metric data. This final social anxiety score can range from 0 to 144, with low scores depicting absence of social anxiety and high scores depicting a significant presence of social anxiety. We conducted a Shapiro-Wilk test to determine if our variables were normally distributed or not. As some of them were not normally distributed and to be able to compare the correlation coefficients between themselves, we conducted Spearman's rank-order correlation on our data, between the final social anxiety scores and the gaze metrics results.

As expected, we found some negative correlations between social anxiety and metrics of the eye-tracking data. In particular, dwell time for directed (D) and dynamic conditions (DA, AD) showed significant negative correlations (D : rs = -0.42; p = 0.022; AD : rs = -0.57; p = 0.001; DA : rs = -0.37; p = 0.047), indicating that the more socially anxious the user was, less time he or she spent observing the agent whose gaze was directed towards them. The correlation was particularly high in the AD condition (getting caught staring). Other metrics were not correlated with social anxiety, except for the averted condition first fixation duration (A : rs = -0.40; p = 0.028) and the averted-then-directed condition fixation count (AD : rs = -0:49; p = 0.006).

4.3.5 General discussion

Our study evaluated VR users' gaze behaviours depending on different gaze conditions that were applied to a virtual crowd, and therefore aimed to test the stare-in-the-crowd effect in VR. Our H1 hypothesis was that the stare-in-the-crowd effect would be preserved in VR, and H2 hypothesis

that we would observe a negative correlation between the time spent towards the agents who are staring at the user and the user's level of social anxiety.

In terms of verifying H1, we compared our results with the one obtained by Crehan et al. **[Crehan 2015]** using similar metrics, and found similar effects, confirming the stare-in-the-crowd effect in VR. Some differences with the previous study were found also, but we were able to find explanations for this (see Section **4.3.4.1**). One major difference was that we used a VR environment that could have affected the gaze behaviour simply due to the field of view being different to the view of the people looking at photographs. It appears to be important how the user is positioned in VR as well, since our results showed that for the middle and far left field of view, some aspects of the stare-in-the-crowd effect were not present (related to dwell time metric) and in the left and right field of view for other aspects (here through first fixation metric, by finding expected results only for active agents in central positions.) An important difference was also between dynamic conditions of both studies. In our study, we found less gaze fixation behaviour in the dynamic conditions than in the directed static one, opposite to the findings of the previous study. This could be explained by users expecting changes in the behaviour of virtual agents in VR, since agents were slightly animated (blinking), whereas photographic stimuli may not have had the same anticipation effect. Our results are potentially more accurately transferable to physical reality than previous results that were collected by using photographs as stimuli.

Regarding H2, our results show that social anxiety is negatively correlated with dwell time for all conditions that include directed gaze. Therefore, on average, the higher the social anxiety, the less time users spent looking at the agents when their gaze was directed towards them, which is in line with the gaze behaviour of socially anxious individuals **[Baker 2002]**. Particularly interesting is the result that the averted-then-directed condition ("being caught staring") had the strongest correlation compared to other conditions, meaning that socially anxious individuals were particularly sensitive to agents who looked at them after the user saw them. Other metrics (fixation time, etc.) were not correlated, meaning that perhaps the additive effect of dwell time metric was stronger. However, we did get the negative correlation with fixation count for the averted-then-directed condition again, but also for the averted condition, with the first fixation duration. The latter could indicate that users with higher social anxiety may avoid to look at characters at the very beginning of the trial for fear of meeting their gaze. Some users reported their fear of the virtual agents in our post-experiment questionnaire and reported avoiding agents who were staring at them: "actually, older people are super scary", "embarrassed by the stare of the avatars towards me, I run away from them rather quickly", "some avatars felt creepier than others, their gaze felt heavier when they were looking for afar, and more normal or natural when they were actually just in front of me". Importantly, we were able to demonstrate a stare-in-the-crowd effect in our study, indicating that the amount of socially anxious individuals in our sample of users was not high.

There are limitations to our study. Firstly, our sample of participants was not balanced in terms of gender, which may have affected our data. While our sample was not balanced in gender, we made sure that we had a balanced representation of both genders in the stimuli sample. We also cannot generalise our results to more natural social situations. While we designed the agents to be as realistic in appearance as possible, better models and animations could be used to make the results more transferable to interactions in the physical world. In addition, other scenarios than the one where the virtual audience is listening to a speaker, could be considered. Moreover, in this study we took behavioural measures using an eye-tracking system and an indirect measure with the social anxiety questionnaire, however we could also have used some subjective measures such as presence and social presence **[Bailenson 2001, Slater 1994]**. Another limitation is that we did not check specifically for cybersickness. Nonetheless we ensured a sufficient framerate in the FOVE headset and our VR users were seated and had limited movements, therefore adverse effects of cybersickness were limited. We also found the importance of where the user is positioned in VR as this affects the stare-in-the-crowd effect. Future studies are needed to better understand the stare-in-the-crowd effect at different observing positions and also in times when the user is allowed to move through the environment.

4.3.6 Conclusions and future work

This work addressed the well-known stare-in-the-crowd effect, which predicates the existence of a search asymmetry between directed and averted gaze towards the observer, with faster detection and longer fixation towards directed gaze. In other words, it represents the tendency of humans in noticing and observing, more frequently and for longer time, gazes oriented toward them (directed gaze) than gazes directed elsewhere (averted gaze). The existence of the stare-in-the-crowd effect has been already proven using photographic stimuli, but never in VR.

Our results confirmed the stare-in-the-crowd effect in VR alongside the evidence that this effect is milder with people reporting higher social anxiety levels. With this, we showed that gaze can indeed change the focus of attention of a user, and potentially trigger the interaction with an agent. Such results are very encouraging, since they can improve our understanding of social interactions in VR applications and help design more engaging experiences with agents. For example, our gaze conditions could be used to initiate the interaction with the user in a virtual crowd. We also demonstrated a simple dynamic gaze condition that signals complex social behaviour, e.g., directed-then-averted gaze could potentially be interpreted as a sign of embarrassment of the agent. These subtle gaze conditions could be explored further to create more believable social interactions in VR.

In the future, we plan to explore the stare-in-the-crowd and other related effects in more complex scenarios, e.g., including more dynamic and heterogeneous virtual agents, changing their number, giving the user different tasks. Moreover, we will expand our analysis to also consider further social and behavioural aspects of our human users, so as to see how they relate to the gazing times. Finally, we want to expand the subject pool for achieving a better balance in terms of gender and age, which will also enable us to analyse the effects of such characteristics on the results.

4.4 HAPTICS TECHNIQUES

In this section, we describe how we put in practice the haptic rendering techniques we have set as part of immersive technologies as shown in the previous deliverable D4.3. More specifically, we explore their possible effect on users behaviours in 1-to-n scenarios.

4.4.1 Preliminaries

In order to explore multimodal 1-to-n interactions, we also investigated the use of wearable haptics as a modality to communicate novel types of information from the virtual agents to the user. Wearable haptic interfaces are an easy and unobtrusive way to convey contact sensations, as they can provide rich information without impairing the user's motion. As haptics is a prominent sense in our lives, we expect its use to be paramount to convey realistic and compelling interactions. As previous works have mostly been limited to distant interactions between groups of characters, due to the difficulty of rendering realistic sensations of collisions in VR, our first goal was to evaluate whether rendering physical contacts through the use of wearable haptics could influence user behaviours.

To this aim, we conducted a VR experiment (see Figure **4.4.1**) where participants navigated in a crowded virtual train station, while being equipped with a motion capture suit (to capture and display their movements on their avatar), wearable haptics armbands (to render physical contacts), and a wide-field-of-view head-mounted display (HMD). Participants either experienced haptic feedback when they collided with virtual characters (i.e., when they virtually entered in contact with them), or did not receive any feedback. The purpose of the study was to investigate the effect of haptic rendering of collisions on participants' behaviour during navigation through a static crowd in VR. To explore this question, we immersed participants in a virtual trainstation and asked them to perform a navigation task which involved moving through a crowd of virtual characters. In some conditions, collisions with the virtual characters were rendered to participants using 4 wearable vibrotactile haptic devices (actuated armbands). Our general hypothesis is that haptic rendering changes the participants' behaviour by giving them feedback about the virtual collisions.

Moreover,we also expect that even after removing haptic rendering, an after-effect still persists on the participants' behaviour.
Our results show that providing haptic feedback improved the overall realism of the interaction between the user and the virtual characters. In particular, participants more actively avoided collisions with the virtual characters when they experienced haptic rendering of contacts, therefore demonstrating changes in their localised interactions but no changes in their global trajectories. We also noticed a significant after-effect in the user's behaviour, where they continued to show more careful interactions with the virtual characters even though they were not experiencing haptic rendering of collision in the last part of the experiment. This experiment was recently submitted to the journal IEEE Transactions on Visualization and Computer Graphics, a renowned high-impact publication in the field. These results confirmed that haptic rendering can therefore be relevant to communicate novel types of information from the virtual agents to the user, which we want to further explore to create more expressive virtual characters.

Figure 4.4.1: We investigated the influence of rendering physical contacts while navigating in a virtual crowd. Participants wore a Xsens motion capture suit, wearable haptics armbands, and a wide-field-of-view HMD.

4.4.2 Materials & Methods

4.4.2.1 Apparatus

For the purpose of immersing participants in the virtual environment and investigating the potential effects of haptic rendering while navigating in groups of characters, we used the setup described in the previous deliverable, that we briefly remind here:

- Motion Capture: to record participants' body motions, as well as to render their animated avatar in the scene, we used an IMU-based (Inertial Measurement Unit) motion capture system (Xsens 2).
- HMD: to immerse participants in the virtual environment, we chose to use a Pimax 3 virtual reality headset, in particular because of the wide field of view provided in these situations of close proximity with other characters
- Haptic Rendering: to render haptic collisions between participants and the virtual characters, we equipped participants with four armbands (one on each arm and forearm). Each armband is composed of four vibrotactile motors with vibration frequency range between 80 and 280 Hz and controlled independently. Motors are positioned evenly onto an elastic fabric strap. An electronics board controls the hardware. It comprises a 3.3 V Arduino Mini Pro, a 3.7 V Li-on battery, and a Bluetooth 2.1 antenna for wireless communication with the external control station.

4.4.2.2 Environment and Task

Participants were immersed in a digital reproduction of the metro station "Mayakovskaya" in Moscow, amongst a virtual static crowd (see Figure **4.4.1**). A total of 8 different configurations of the scene were prepared in advance and used in the experiment. A configuration is defined by the exact position of each crowd character in the virtual station. In each configuration, the crowd formed a squared shape, and character positions followed a Poisson distribution resulting in a density of

1.47 ± 0.06 character/m^2 . Such a distribution combined with such a level of density ensures that a gap of 0.60 m on average exists between each character. The crowd is composed of standing virtual characters animated with various idle animations (only small movement but standing in place). In each configuration, characters were animated according to two types of behaviour, either waiting (oriented to face the board displaying train schedules, moving slightly the upper body) or phone-calling (with a random orientation). We used several animation clips for each of the two behaviours, in order to prevent the exact same animation clip to be used for two different virtual characters. At the beginning of each trial, participants were initially standing at one corner of the square crowd, embodied in a gender-matched avatar. They were instructed to traverse the crowd so as to reach the board displaying train schedules, and to read aloud the track number of the next train displayed on the board before coming back to their initial position. They were physically walking in the real room, while their position and movements were used to animate their avatar. This task required participants to reach the opposite corner of the space in order to read information on the board, while forcing them to move through the virtual crowd. Also, the screen displayed the train information only when participants were at less than 2 m from it. Furthermore, we provided the following instruction to participants prior to the experiment: "*Walk through the virtual train station as if you were walking in a real train station*".

4.4.2.3 Protocol and Participants

Participants were equipped with the Xsens suit, the four armbands for haptic rendering, a wearable backpack computer, the head-mounted display and headphones for sound immersion. Calibration of the Xsens motion capture system was then performed to ensure motion capture quality, as well as to resize the avatar to participants dimensions. Once ready, participants performed a training trial in which they could explore the virtual environment and get familiar with the task. The experiment then consisted of 3 blocks of 8 trials, where the blocks were presented for all participants in the following order: NoHaptic1, Haptic, and NoHaptic2. The Haptic block corresponded to performing the task with haptic rendering of contacts, while the NoHaptic blocks did not involve any haptic rendering of contacts. The experiment therefore consisted in performing first a block without haptic rendering, in order to measure a baseline of participants' reactions. The purpose of the second block was then to investigate whether introducing haptic rendering influenced their behaviour while navigating in a crowd, while the purpose of the last block (without haptic) was to measure potential after-effects.

Twenty-three unpaid participants, recruited via internal mailing lists amongst students and staff, volunteered for the experiment.

4.4.2.4 Hypotheses

H1: Haptic rendering will not change the path followed by participants through the crowd. Indeed, pedestrians mainly rely on vision to control their locomotion, and we replicated each crowd configuration across the 3 blocks, resulting in identical visual information for participants to navigate. Therefore the followed path will be similar in the tree blocks of the experiment (NoHaptic1, Haptic and NoHaptic2).

H2: Haptic rendering of collisions will make participants aware of collisions and influence their body motion during the navigation through the crowd. Therefore, concerning the NoHaptic1 and Haptic blocks of the experiment, we expect that:
H2_1: Participants will navigate in the crowd more carefully in the Haptic block in order to avoid collisions. There will be more local avoidance movements (e.g., increased shoulder rotations) and a difference in participants' speed.
H2_2: With these changes on participants' local body motions, there will be both less collisions, and smaller volumes of interpenetration when a collision occurs.

H3: We expect some after-effect due to haptic rendering, i.e., we expect that participants will remain more aware and careful about collisions even after we disabled haptic rendering. Therefore we expect H2_1 and H2_2 to remain true in the NoHaptic2 block.

H4: Haptic rendering will improve the sense of presence and the sense of embodiment of participants in VR, as they will become more aware of their virtual body dimensions in space with respect to neighbour virtual characters.

4.4.3 Results

This section presents the results of our experiment, starting with the study of H1 on the trajectories formed by participants through the virtual crowd. We then explore H2_1 and H2_2 with respect to the analysis of body movements. Finally, we report the results on collision metrics so as to evaluate H3, to finish with the answers to the Presence and Embodiment questionnaires related to H4.

4.4.3.1 Trajectory Analysis

To study H1, we compared participants' trajectories through the virtual crowd. To this end, we decomposed the environment into cells based on a Delaunay triangulation, the vertices of which were the crowd characters. A trajectory is then represented as a sequence of traversed cells.

Table **4.4.1** shows the results of the Dice similarity measure between all possible pairs of blocks. Similarity ranges from 84.7% (Nohaptic1 vs. Haptic blocks) to 88.5% (Haptic vs. NoHaptic2 blocks). The score is higher for Haptic vs. Nohaptic2 blocks (88.6 ± 4.1%) and for Nohaptic1 vs. Nohaptic2 (85.9 ± 4.0%). Because it is difficult to identify from this data only whether the obtained level of similarity is due to natural variety in human behaviours, or to the difference in conditions explored in each block, we propose to measure similarity between paths belonging to the same block as follows. For each block and each configuration, we randomly divided the trajectories into two subsets and computed the Dice similarity score between them. We repeated this process 30 times (which changes the way trajectories are divided into 2 subsets). Performing this process and computing similarity over the 3 blocks resulted in 90 measures of "intra-block similarity". The obtained average value is 81.2 ± 3.3%, which can be compared with the "inter-block similarity" scores presented in Table **4.4.1**. Our results show that there is no statistical difference between intra-block and Nohaptic1 vs. Haptic blocks similarity measure ($p > 0.05$. There is however a significant difference between intra-block and Haptic vs. Nohaptic2 blocks ($p < 0.01$), as well as intra-block and Nohaptic1 vs. Nohaptic2 ($p < 0.05$), where intra-block similarity measures are always lower. Given that similarity measures between pairs of blocks were either as similar or more similar than intra-block similarities, we can conclude that participants chose their path through the crowd similarly, irrespective of the block condition, which supports H1.

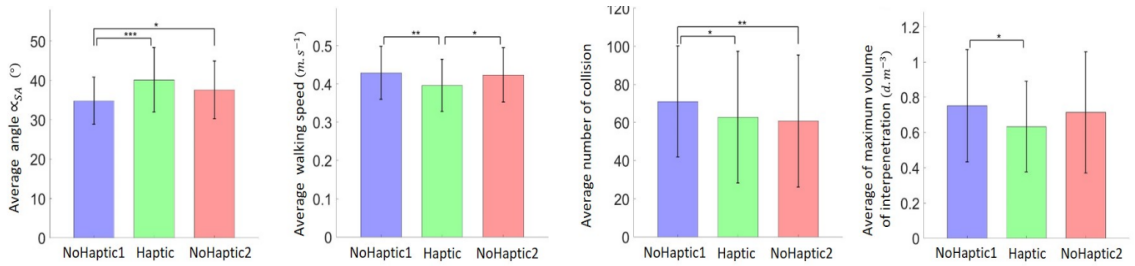
Trials	Blocks		
	NoHaptic1 vs. *Haptic*	*Haptic* vs. *NoHaptic2*	*NoHaptic1* vs. *NoHaptic2*
T_1	84.0%	88.6%	85.0%
T_2	88.4%	93.8%	88.3%
T_3	78.1%	93.2%	79.4%
T_4	91.9%	88.7%	90.7%
T_5	88.4%	90.2%	85.3%
T_6	82.8%	85.8%	91.0%
T_7	78.8%	81.6%	82.0%
T_8	85.0%	85.9%	85.3%
$\boldsymbol{T_{all}}$	**84.7 ± 4.8%**	**88.6 ± 4.1%**	**85.9 ± 4.0%**

Table 4.4.1: Similarity measure (Dice) of participant trajectories between all blocks (NoHaptic1, Haptic, NoHaptic2) for all the trials.

4.4.3.2 Body Motion

*Shoulder Rotation (Figure **4.4.2(a)**).* The average amplitude of shoulder rotations was significantly different in each block (F (2, 44) = 13.0, p < 0.001). In particular, it was significantly higher in the block with haptic rendering (40.1 ± 8.2 deg), than in the first block without haptic rendering (34.3 ± 6.0 deg). A higher shoulder rotation angle means that participants made a larger rotation to squeeze between virtual characters, therefore validating the hypotheses H2_1. Furthermore, it was also significantly higher in block NoHaptic2 (38.7 ± 3.7 deg) than in block NoHaptic1, suggesting that participants continued to turn more their shoulders even after haptic rendering was disabled, therefore supporting H3.

*Walking Speed (Figure **4.4.2(b)**).* We found an effect of haptic rendering (F (1.56, 34.2) = 7.14, p = 0.005) on participant's average walking speed, where participants' walking speed was on average significantly lower in theHaptic block (0.40±0.07 m/s) than in the NoHaptic1 (0.43±0.07 m/s) and NoHaptic2 (0.42±0.07 m/s) blocks. This result therefore supports hypothesis H2_1 .



a) shoulder rotation b) walking speed c) number of collisions d) volume of interpenetration

Figure 4.4.2: Effect of the experimental condition (no haptic, haptic rendering of collision, no haptic 2) on shoulder rotation, walking speed, number of collisions and volume of interpenetration with virtual agents (mean ±SD).

4.4.3.3 Number of collisions are volume of interpenetration

We analysed the number of collisions as well as the volume of interpenetration between the user's avatar and the virtual agents in the scene, shown in Figures. **4.4.2(c)** and **4.4.2(d)**. The average number of collisions per trial was influenced by haptic rendering with a large effect (F (2, 44) = 7.13, p = 0.002). Post-hoc analysis showed that the number of collisions was higher during the NoHaptic1 block (71 ± 29.2) than during the Haptic (62.8 ± 34.6 , p = 0.018) and NoHaptic2 blocks (60.7 ± 34.6, p = 0.002), which shows that participants made on average more collisions before they experienced haptic rendering. The average volume of interpenetration was also influenced by the block (F (2, 44) = 4.35, p = 0.019), where post-hoc analysis showed that this volume was smaller (p = 0.016) in the Haptic block (0.6 ± 0.3 dm^{-3}) than during the NoHaptic1 (0.8 ± 0.3 dm^{-3}). These results validate our hypothesis H2_2, which states that haptic rendering reduces the severity of collisions between participants and virtual characters. Furthermore, as the number of collisions is higher during block NoHaptic1 than during block NoHaptic2, this also supports H3 on potential after-effects of haptic rendering.

4.4.3.4 Presence and Embodiment

Presence and Embodiment. Another important aspect of our analysis is its perceptual relevance. In accordance with H4, we looked for any difference in the users' feelings of presence and embodiment, comparing the registered subjective perception with and without haptic rendering. Participants answered both questionnaires at the end of each block (Embodiment then Presence), answering each question on a 7-point Likert scale.

The average participant ratings and all the questions for embodiment are shown in Tables **4.4.2**, **4.4.3**, and **4.4.4**. We did not find any significant effect of the blocks for Agency (p = 0.438), Change (p = 0.085) and Ownership (p = 0.753). Furthermore, Table **4.4.5** shows the questions and the average participant ratings for presence, for which we also did not find a significant effect of the blocks (p = 0.222). These results therefore do not support hypothesis H4, suggesting that haptic rendering does not improve the sense of presence or the sense of embodiment of participants in VR.

Questions	**blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
The movements of the virtual body felt like they were my movements. I felt like I was controlling the movements of the virtual body I felt like I was causing the movements of the virtual body. The movements of the virtual body were in sync with my own movements.	6.1 ± 0.9	6.0 ± 0.8	5.9 ± 0.7

Table 4.4.2: Agency questionnaire: average participant ratings for the three blocks.

Questions	**Blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
I felt like the form or appearance of my own body had changed. It felt like the weight of my own body had changed. I felt like the size (height) of my own body had changed. I felt like the width of my own body had changed.	3.6 ± 1.3	3.8 ± 1.5	3.3 ± 1.5

Table 4.4.3: Ownership questionnaire: average participant ratings for the three blocks.

Questions	**Blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
It felt like the virtual body was my body. It felt like the virtual body parts were my body parts. The virtual body felt like a human body. It felt like the virtual body belonged to me.	4.9 ± 1.4	5.1 ± 1.2	5.0 ± 1.2

Table 4.4.4: Slater-Usoh-Steed (SUS) questionnaire and average participant ratings for the three blocks.

Questions	**Blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
I had a sense of being there in the train station. There were times during the experience when... the train station was the reality for me... The train station seems to me to be more like... I had a stronger sense of... I think of the train station as a place in a way similar to other places that I've been today. During the experience I often thought that I was really standing in the train station.	5.2 ± 0.9	5.2 ± 1.2	5.0 ± 1.1

Table 4.4.5: Change questionnaire: average participant ratings for the three blocks.

4.4.4 Discussion

The main objective of this study was to evaluate the effect of haptic rendering of collisions on participants' behaviour while navigating in a dense virtual crowd. These results confirmed that haptic rendering can therefore be relevant to communicate novel types of information from the virtual agents to the user, which we want to further explore to create more expressive virtual characters.

Trajectories. The analysis of the Dice similarity measure showed that haptic rendering did not change the way participants selected their path through the crowd, as stated in hypothesis H1 . We even found that paths across blocks were "more similar" than within the same block. One possible explanation is given by the way we compose the sets we compare the similarity of, where we assume that paths are independent from participants. Indeed, the intra-block similarity measure required us to split a set of trajectories belonging to the same block and crowd configuration, which resulted in comparing paths performed by different participants. In contrast, the inter-block analysis considered sets that were split according to haptic rendering conditions, thus comparing paths performed by the same group of 23 participants. In spite of this limitation in our analysis, we consider that paths are similar across blocks. One can describe human motion as a trajectory resulting from a perception-action loop. Depending on the tasks, the loop is a multimodal one, meaning that different senses are used to control motion. However in the context of walking, vision is the most used perceptual input to navigate to the goal. Such statements hold in our case, where a major difference with previous work is the higher density of obstacles. Nevertheless, assuming that tactile feedback may affect path selection, it would have been probable that some participants reversed their course after a collision has been rendered, which was not observed.

Avoidance Behaviour. In this experiment, we demonstrated that haptic rendering had an effect on shoulder rotations, which supports hypothesis H2_1. In particular, participants rotated more their shoulders when traversing the gaps between virtual characters during the Haptic block than during the NoHaptic1 block. Let us remind that the human trunk is most often larger along the transverse axis than along the antero-posterior axis. Thus, the more the participants turn their shoulders the smaller the volume swept by their body motion. Our results therefore suggest that participants might have tried to minimise the risk of collision with virtual characters more in the condition where they experienced haptic rendering than in the first block of the experiment. The slower speed observed in the Haptic block also reveals that participants moved more cautiously. Being more cautious effectively resulted in less collisions as expected in hypothesis H2_2. Results show that the average number of collisions as well as the average volume of interpenetration were significantly lower in the Haptic block than in the NoHaptic1 block.

Haptic Rendering After-effects. While there were less collisions and more shoulder rotations observed in the Haptic block in comparison with the NoHaptic1 block, there was no difference between the Haptic and the NoHaptic2 blocks. This supports hypothesis H3 on potential after-effects of haptic rendering. However, such an after-effect did not equally influence all measurements, such as walking speed that increased again in the NoHaptic2 block. One possible explanation might be a perceptual calibration of the participants. During the experiment, participants became more familiar with the environment, the task to be performed, but also the virtual representation of their body and the virtual environment, enabling them to move faster and better avoid collisions with the virtual characters in the last block (NoHaptic2). Another point to highlight is that participants, at the beginning of the Haptic block, did not know that contacts would now trigger a vibrotactile haptic sensation. For this reason, we might expect to see a short learning phase at the beginning of the block, where participants learn to deal with the newly-rendered haptic collisions. Considering this point, we can expect the effect of providing haptic sensations of collisions even stronger than registered. However, to provide a more definitive conclusion on the role of the haptic after-effect would require to add a control group with no haptic rendering throughout the 3 blocks of the experiment, which could be explored in future work. These results can also open perspectives regarding the design of new experiments including haptic priming tasks.

Embodiment and Presence. In contrast with our hypothesis H4, we did not find any significant change in terms of the user's perceived senses of embodiment and presence when experiencing haptic feedback. This result is quite surprising, as we did find significant effects in other

measurements, suggesting that participants took different actions when provided with haptic sensations of contact. An explanation for this result could lie in the fact that users already registered high embodiment and presence levels without experiencing haptic feedback in the first condition (NoHaptic1), leaving little room for improvement in the Haptic condition. Finally, a last explanation could be the location and number of our haptic devices. Employing a higher number of bracelets spread throughout the body might better render the target contact sensations. All these considerations will drive our future work.

5 CONCLUSION

With this work around reaction capabilities of agents, Inria has explored various promising research directions for the future.

In the context of character body animation, we have been interested in a mechanism for adapting movement based on the idea that it must first and foremost be adapted to the position of an observer in order to preserve its meaning in the sense of non-verbal communication. This is an original idea, which shows in particular that the method is capable of adapting the movement in a robust way to changes of viewpoint, or to satisfy visibility constraints. In the future, we would like to allow the control of parameters related to personality or emotions, which time has not allowed us to address.

For the case of collective movements, we have addressed a critical point: the difficulty of modelling new behaviours to enrich the simulation and address new scenarios. We propose the "interaction fields" solution, which seems to us to be promising for several reasons. It has allowed us to realise scenarios never seen before in simulation (e.g., hide and seek scenario), and we are in the process of using this method for the automatic learning of new behaviours by using the interaction fields as a projection base for real data.

Finally, we show through two studies the essential role of the gaze behaviour of people, as well as making users aware of the contacts made with humans. While these last two studies do not strictly speaking propose appropriate new techniques, they do provide guidance on the proper design of immersive platforms for interaction with characters and requirements in terms of animation techniques. Ongoing work focuses on the automatic control of character gaze in scenes.

6 REFERENCES

[Argyle 1969] Argyle, M. (1969). Social interaction. Tavistock publications.

[Aristidou 2018] Aristidou, A., Lasenby, J., Chrysanthou, Y., & Shamir, A. (2018, September). Inverse kinematics techniques in computer graphics: A survey. In Computer Graphics Forum (Vol. 37, No. 6, pp. 35-58).

[Ash 1951] Asch, S. E. (1951). Effects of group pressure upon the modification and distortion of judgments. Organizational influence processes, 295-303.

[Bailenson 2001] J. N. Bailenson, J. Blascovich, A. C. Beall, and J. M. Loomis. Equilibrium theory revisited: Mutual gaze and personal space in virtual environments. Presence: Teleoperators & Virtual Environments, 10(6):583– 598, 2001.

[Bailenson 2003] Bailenson, J.N., Blascovich, J., Beall, A. C., and Loomis, J. M. (2003). Interpersonal distance in immersive virtual environments. Personality and Social Psychology Bulletin, 29, 7, 819–833.

[Bailenson 2005] J. N. Bailenson and N. Yee. Digital chameleons: Automatic assimilation of nonverbal gestures in immersive virtual environments. Psychological science, 16(10):814–819, 2005.

[Baker 2002] S. R. Baker and R. J. Edelmann. Is social phobia related to lack of social skills? duration of skill-related behaviours and ratings of behavioural adequacy. British Journal of Clinical Psychology, 41(3):243–257, 2002.

[Brady 1978] Brady, A. and Walker, M.B. (1978). Interpersonal distance as a function of situationally induced anxiety. British Journal of Social and Clinical Psychology, 17, 2, 127–133.

[Buhler 2018] M. A. Buhler and A. Lamontagne. Circumvention of pedestrians while walking in virtual and physical environments. IEEE transactions on neural systems and rehabilitation engineering, 26(9):1813–1822, 2018.

[Burgoon 2003] J. K. Burgoon and A. E. Bacue. Nonverbal communication skills. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey, USA, 2003.

[Cafaro 2016] Cafaro, A., Ravenet, B., Ochs, M., Vilhjálmsson, H. H., & Pelachaud, C. (2016). The effects of interpersonal attitude of a group of agents on user's presence and proxemics behavior. ACM Transactions on Interactive Intelligent Systems (TiiS), 6(2), 1-33.

[Cañigueral and Hamilton 2019] Cañigueral, Roser, and Antonia F. de C. Hamilton. "The role of eye gaze during natural social interactions in typical and autistic people." *Frontiers in Psychology* 10 (2019): 560.

[Chaumette and Hutchinson 2006] François Chaumette and Seth Hutchinson. 2006. Visual servo control. I. Basic approaches. IEEE Robotics & Automation Magazine 13, 4 (2006), 82–90.

[Christian 1961] Christian, J.J., Vagh F., and Davis, D. E. (1961). Phenomena associated with population density. Proceedings of the National Academy of Science 47:428-49.

[Colombatto 2020] C. Colombatto, B. van Buren, and B. J. Scholl. Gazing without eyes: A "stare-in-the-crowd" effect induced by simple geometric shapes. Perception, 49(7):782–792, 2020.

[Cooper 2013] R. M. Cooper, A. S. Law, and S. R. Langton. Looking back at the stare-in-the-crowd effect: Staring eyes do not capture attention in visual search. Journal of vision, 13(6):10–10, 2013.

[Crehan 2015] E. T. Crehan and R. R. Althoff. Measuring the stare-in-the-crowd effect: a new paradigm to study social perception. Behavior research methods, 47(4):994–1003, 2015.

[Crehan 2021] E. T. Crehan and R. R. Althoff. Me looking at you, looking at me: The stare-in-the-crowd effect and autism spectrum disorder. Journal of Psychiatric Research, 2021.

[Dael 2012] Dael, N., Mortillaro, M., & Scherer, K. R. (2012). The body action and posture coding system (BAP): Development and reliability. Journal of Nonverbal Behavior, 36(2), 97-121.

[de Gelder et al. 2015] Beatrice de Gelder, AW De Borst, and R Watson. 2015. The perception of emotion in body expressions. Wiley Interdisciplinary Reviews: Cognitive Science 6, 2 (2015), 149–158.

[Dombre and Khalil 2013] Etienne Dombre and Wisama Khalil. 2013. Robot manipulators: modeling, performance analysis and control. John Wiley & Sons, Hoboken, New Jersey, USA.

[Ekman 1978] Ekman, P., & Friesen, W. V. (1978). Facial action coding systems. Consulting Psychologists Press.

[Espiau et al. 1992] Bernard Espiau, François Chaumette, and Patrick Rives. 1992. A new approach to visual servoing in robotics. ieee Transactions on Robotics and Automation 8, 3 (1992), 313–326.

[Framorando 2016] D. Framorando, N. George, D. Kerzel, and N. Burra. Straight gaze facilitates face processing but does not cause involuntary attentional capture. Visual cognition, 24(7-8):381–391, 2016.

[Garau 2003] M. Garau, M. Slater, V. Vinayagamoorthy, A. Brogni, A. Steed, and M. A. Sasse. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 529–536, 2003.

[Gleicher and Witkin 1992] Michael Gleicher. 1998. Retargetting Motion to New Characters. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98). Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/280814.280820

[Gonzalez-Franco 2020] M. Gonzalez-Franco, E. Ofek, Y. Pan, A. Antley, A. Steed, B. Spanlang, A. Maselli, D. Banakou, N. Pelechano Gomez, S. Orts-Escolano, et al. The rocketbox library and the utility of freely available rigged avatars. Frontiers in virtual reality, 1(561558):1–23, 2020.

[Hall 1963] Hall, E. T. (1963). A system for the notation of proxemic behavior. American anthropologist, 65(5), 1003-1026.

[Hall 1968] Hall, E. T., Birdwhistell, R. L., Bock, B., Bohannan, P., Diebold Jr, A. R., Durbin, M., and La Barre, W. (1968). Proxemics [and comments and replies]. Current anthropology, 9(2/3), 83-108.

[Helbing 1995] Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. Physical review E, 51(5), 4282.

[Hinde 1972] Robert A Hinde. 1972. Non-verbal communication. Cambridge University Press, Cambridge, England.

[Holden et al. 2016] Daniel Holden, Jun Saito, and Taku Komura. 2016. Learning Inverse Rig Mappings by Nonlinear Regression. IEEE transactions on visualization and computer graphics 23, 3 (2016), 1167–1178.

[**Doi 2007] H. Doi and K. Ueda.** Searching for a perceived stare in the crowd. Perception, 36(5):773–780, 2007.

[Iachini 2014] Iachini, T., Coello, Y., Frassinetti, F., and Ruggiero, R. (2014). Body space in social interactions: a comparison of reaching and comfort distance in immersive virtual reality. PloS one, 9, 11, e111511.

[Juslin 2005] Juslin, P. N., Scherer, K. R., Harrigan, J. A., & Rosenthal, R. (2005). The new handbook of methods in nonverbal behavior research.

[Kendin 1990] Kendon, A. (1990). Conducting interaction: Patterns of behavior in focused encounters (Vol. 7). CUP Archive.

[Kimura et al. 2013] Akisato Kimura, Ryo Yonetani, and Takatsugu Hirayama. 2013. Computational models of human visual attention and their implementations: A survey. IEICE TRANSACTIONS on
Information and Systems 96, 3 (2013), 562–578.

[Kmiecik 1979] Kmiecik, C., Mausar, P., & Banziger, G. (1979). Attractiveness and interpersonal space. The Journal of Social Psychology, 108(2), 277-278.
Remland, M. S., Jones, T. S., & Brinkman, H. (1995). Interpersonal distance, body orientation, and touch: Effects of culture, gender, and age. The Journal of social psychology, 135(3), 281-297.

[Lange 2019] B. Lange and P. Pauli. Social anxiety changes the way we move—a social approach-avoidance task in a virtual reality cave system. PloS One, 14(12):e0226805, 2019.

[Liebowitz 1987] M. R. Liebowitz. Social phobia. Modern problems of pharmacopsychiatry, 1987.

[Lopez 2019] López, A., Chaumette, F., Marchand, E., & Pettré, J. (2019, May). Character navigation in dynamic environments based on optical flow. In Computer Graphics Forum (Vol. 38, No. 2, pp. 181-192).

[Manor 2003] B. R. Manor and E. Gordon. Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. Journal of Neuroscience Methods, 128(1):85–93, 2003.

[Marschner 2015] L. Marschner, S. Pannasch, J. Schulz, and S.-T. Graupner. Social communication with virtual agents: The effects of body and gaze direction on attention and emotional responding in human observers. International Journal of Psychophysiology, 97(2):85–92, 2015.

[Narang 2016] S. Narang, A. Best, T. Randhavane, A. Shapiro, and D. Manocha. Pedvr: Simulating gaze-based interactions between a real user and virtual crowds. In Proceedings of the 22nd ACM conference on virtual reality software and technology, pp. 91–100, 2016.

[Neff et al. 2010] Michael Neff, Yingying Wang, Rob Abbott, and Marilyn Walker. 2010. Evaluating the Effect of Gesture and Language on Personality Perception in Conversational Agents. In Intelligent Virtual Agents, Jan Allbeck, Norman Badler, Timothy Bickmore, Catherine Pelachaud, and Alla Safonova (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 222–235.

[Nummenmaa 2009] L. Nummenmaa, J. Hyona, and J. K. Hietanen. I'll walk this way: Eyes reveal the direction of locomotion and make passersby look and go the other way. Psychological science, 20(12):1454–1458, 2009.

[Palanica 2011a] A. Palanica and R. Itier. Measuring the stare-in-the-crowd effect using eye-tracking: Effects of task demands. Journal of Vision, 11(11):1327–1327, 2011.

[Palanica 2011b] A. Palanica and R. J. Itier. Searching for a perceived gaze direction using eye tracking. Journal of Vision, 11(2):19–19, 2011.

[Patil 2010] Patil, S., Van Den Berg, J., Curtis, S., Lin, M. C., & Manocha, D. (2010). Directing crowd simulations using navigation fields. *IEEE transactions on visualization and computer graphics*, *17*(2), 244-254.

[Randhavane 2019] Randhavane, T., Bhattacharya, U., Kapsaskis, K., Gray, K., Bera, A., & Manocha, D. (2019). Identifying emotions from walking using affective and deep features. arXiv preprint arXiv:1906.11884.

[Ramamoorthy 2019] N. Ramamoorthy, K. Plaisted-Grant, and G. Davis. Fractionating the stare-in-the-crowd effect: Two distinct, obligatory biases in search for gaze. Journal of Experimental Psychology: Human Perception and Performance, 45(8):1015, 2019.

[Roether et al. 2009] Claire L Roether, Lars Omlor, Andrea Christensen, and Martin A Giese. 2009. Critical features for the perception of emotion from gait. Journal of vision 9, 6 (2009), 15–15.

[Santello 2016] Santello, M., Bianchi, M., Gabiccini, M., Ricciardi, E., Salvietti, G., Prattichizzo, D., ... & Kyriakopoulos, K. (2016). Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands. Physics of life reviews, 17, 1-23.

[Schulze 2013] L. Schulze, J. S. Lobmaier, M. Arnold, and B. Renneberg. All eyes on me?! social anxiety and self-directed perception of eye gaze. Cognition and Emotion, 27(7):1305–1313, 2013. PMID: 23438447. doi: 10.1080/02699931.2013.773881

[Slater 1994] M. Slater, M. Usoh, and A. Steed. Depth of presence in virtual environments. Presence: Teleoperators & Virtual Environments, 3(2):130–144, 1994.

[Slater 2006] Slater, Mel & Pertaub, David-Paul & Barker, Chris & Clark, David. (2006). An Experimental Study on Fear of Public Speaking Using a Virtual Environment. Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society. 9. 627-33. 10.1089/cpb.2006.9.627.

[Sun 2017] Z. Sun, W. Yu, J. Zhou, and M. Shen. Perceiving crowd attention: Gaze following in human crowds with conflicting cues. Attention, Perception, & Psychophysics, 79(4):1039–1049, 2017

[Sweeny 2014] Sweeny, T. D., & Whitney, D. (2014). Perceiving Crowd Attention: Ensemble Perception of a Crowd's Gaze. Psychological Science, 25(10), 1903–1913. https://doi.org/10.1177/0956797614544510

[Treuille 2006] Treuille, A., Cooper, S. & Popović, Z. (2006). Continuum crowds. ACM Trans. Graph. 25, 3 (July 2006), 1160–1168.

[ven den Berg 2011] van den Berg, Jur & Guy, Stephen & Lin, Ming & Manocha, Dinesh. (2011). Reciprocal n-Body Collision Avoidance. 10.1007/978-3-642-19457-3_1.

[Von Grunau 1995] Von Grunau and C. Anston. The detection of gaze direction: A stare-in-the-crowd effect. Perception, 24(11):1297–1313, 1995.

[Westheimer 1954] G. Westheimer. Eye movement responses to a horizontally moving visual stimulus. AMA Archives of Ophthalmology, 52(6):932–941, 12 1954

[Wieser 2010] M. J. Wieser, P. Pauli, M. Grosseibl, I. Molzow, and A. Muhlberger. Virtual social interactions in social anxiety—the impact of sex, gaze, and interpersonal distance. Cyberpsychology, Behavior, and Social Networking, 13(5):547–554, 2010.

[Wilder 1986] Wilder, D. A. (1986). Social categorization: Implications for creation and reduction of intergroup bias. In Advances in experimental social psychology (Vol. 19, pp. 291-355). Academic Press.

[Witkin 1995] Witkin, A., & Popovic, Z. (1995, September). Motion warping. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 105-108).

[Shepard 1968] Shepard, A., & Popovic, Z. (1968).A two-dimensional interpolation function for irregularly-spaced data(pp. 517–52).

[van Toll 2020] van Toll, Wouter, & Grzeskowiak, Fabien & López Gandía, Axel & Amirian, Javad & Berton, Florian & Bruneau, Julien & Cabrero Daniel, Beatriz & Jovane, Alberto &Pettré, Julien. 2020. Generalized Microscropic Crowd Simulation using Costs in Velocity

Space. In *Symposium on Interactive 3D Graphics and Games* (*I3D '20*). Association for Computing Machinery, New York, NY, USA, Article 6, 1–9. DOI:https://doi.org/10.1145/3384382.3384532

[Helbing 2000] Helbing, Dirk, Farkas, Illés, and Vicsek,Tamas. "Simulating dynamical features of escape panic". *Nature* 407 (2000), 487–490 11. [HM95] H_{ELBING}, D_{IRK} and $M_{OLNÁR}$, $P_{ÉTER}$. "Social force model for pedestrian dynamics". *Physical Review E* 51.5 (1995), 4282–4286 3.

[van den Berg 2008] Van den BERG, JUR, LIN, MING, and MANOCHA, DINESH. "Reciprocal velocity obstacles for real-time multi-agent navigation". Proc. IEEE International Conference on Robotics and Automation. IEEE. 2008, 1928–1935 3, 11.

[Animation 2020] ANIMATION UPRISING. Motion Matching for Unity. 2020. URL: https://assetstore.unity.com/packages/tools/animation/motion-matching-for-unity-145624 11.

[Lewis 2008] LEWIS, JAMES R and SAURO, JEFF. "Item benchmarks for the system usability scale". Journal of Usability Studies 13.3 (2018) 14.

[Brooke 1996] BROOKE, JOHN. "SUS: a quick and dirty usability scale". Usability Evaluation in Industry 189 (1996) 14.

[Colas 2022] COLAS A, van Toll W, Zibrek K, Hoyet L, Oliver A-H and Pettré J. "Interaction Fields: Intuitive Sketch-based Steering Behaviors for Crowd Simulation", 2022 Eurographics.