Robot Navigation in Dense Human Crowds: the Case for Cooperation

Pete Trautman, Jeremy Ma, Richard M. Murray, and Andreas Krause
Caltech, ETH Zurich, and NASA/JPL
peter.trautman@gmail.com

Abstract

We consider mobile robot navigation in dense human crowds. In particular, two questions are explored. Can we design a navigation algorithm that encourages humans to cooperate with a robot? Would such cooperation improve navigation performance? We address the first question by developing a probabilistic predictive model of cooperative collision avoidance and goal-oriented behavior. Specifically, this model extends the recently introduced interacting Gaussian processes approach to the case of multiple goals and stochastic movement duration. We answer the second question by empirically validating our model in a natural environment (a university cafeteria), and in the process, carry out the first extensive quantitative study of robot navigation in dense human crowds (completing 488 runs). The “multiple goal” interacting Gaussian processes algorithm performs comparably with human teleoperators in crowd densities near 1 person/m², while a state of the art noncooperative planner exhibits unsafe behavior more than 3 times as often as our planner. Furthermore, a reactive planner based on the “dynamic window” approach—widely used for robotic tour guide experiments—fails for crowd densities above 0.55 people/m². We conclude that a cooperation model is critical for safe and efficient robot navigation in dense human crowds.

1 Introduction

One of the first substantive and well publicized deployments of an autonomous robot in an unscripted human environment occurred in the late 1990s at the Deutsches Museum in Bonn, Germany [1]. This RHINO experiment was quickly followed by another robotic tour guide experiment; the robot in the follow-on study, named MINERVA [2], was exhibited at the Smithsonian and at the National Museum of American History in Washington D.C. Despite the many successes of the RHINO and MINERVA work and the studies they inspired, fundamental questions about navigation in dense crowds remain. In particular, prevailing algorithms ([3]) and opinion ([4]) on navigation in dynamic environments emphasize deterministic and decoupled approaches. Critically, an experimental study of robotic navigation in dense human crowds is unavailable.

In this paper, we focus on these two major deficiencies: a dearth of human-robot cooperative navigation models and the complete absence of a systematic study of robot navigation in dense human crowds. We thus develop a novel cooperative navigation methodology and conduct the first extensive (n runs ≈ 500) field trials of robot navigation in the natural human crowds¹ of a university cafeteria. In these experiments, we quantify the degree to which our cooperation model improves navigation performance. We conclude that a cooperation model is required for safe and efficient navigation in crowds.

¹We point out the critical importance of an experiment with unscripted human subjects. Simulated humans do not capture the range of human behavior and response. Scripting humans to follow a routine introduces bias that is impossible to quantify.
Related Work  Naively modeling the uncertainty in dynamic environments (e.g., with independent agent constant velocity Kalman filters) leads to an uncertainty explosion that makes safe and efficient navigation impossible ([5]). Some research has thus focused on controlling predictive uncertainty: in [6], high fidelity independent human motion models were developed, in the hope that reducing the uncertainty would lead to improved navigation performance. Similarly, [7] holds the individual agent predictive covariance constant at a low value as a surrogate for near perfect prediction (in the hope that as the robot gets close to the dynamic agents, prediction and replanning will be good enough for safe navigation to occur).

The work of [8] and [9] shares insight with the approach of [7], although more sophisticated individual dynamic models are developed: motion patterns are modeled as a mixture of Gaussian processes with a Dirichlet process prior over mixture weights. The Dirichlet process prior allows for representation of an unknown number of motion patterns, while the Gaussian process allows for variability within a particular motion pattern. Rapidly exploring random trees (RRTs) are used to find feasible paths. However, all the above approaches ignore human-robot interaction. As was argued in [5], unless the dependencies between agents is modeled, navigation will fail in dense crowds.

In [10], the authors develop a model of how the robot’s actions influence both the uncertainty in the environment and the behavior of other actors in the environment. They successfully test their algorithm in large Brownian simulations with repulsion to the simulated robot. In [11], pedestrian decision making is first learned from a large trajectory example database using inverse reinforcement learning (IRL), and then the robot navigates such that the human’s predicted path is minimally disrupted. In [12], the authors extend IRL to work in dynamic environments, and the planner is trained using simulated trajectories. The method successfully recovers the behavior of the simulator.

In [13], a theory of learning interactions is developed using game theory and the principle of maximum entropy; only 2 agent simulations are tested. Similarly, the work of [14] leverages IRL to learn an interaction model from human trajectory data. This research pioneers IRL from human data (and explicitly models cooperation), but the experiments are limited in scale—one scripted human crosses paths with a single robot in a laboratory environment.

2 Methods: Interacting Gaussian Processes

We first review the interacting Gaussian processes (IGP) approach of [5] for cooperative navigation and describe its shortcomings. In Sections 2.1 and 2.1 we address these deficiencies.

The IGP approach is motivated by the following: dynamic navigation algorithms typically assume agent motion to be independent of robot motion. As is shown in [5], this independence assumption leads to highly suboptimal behavior in dense crowds; coupling agent action and robot action via a joint trajectory probability density \( p(f^{(R)}, f^{(1)}, f^{(2)}, \ldots, f^{(n_t)}) \mid z_{1:t}) \) can dramatically improve navigation performance. In this density, \( t \) is the present time, \( f^{(i)} \) is a random function representing agent \( i \)'s trajectory in \( \mathbb{R}^2 \) from time \( 1:T \) where \( T > t \), \( i \) ranges over \( (R, 1, \ldots, n_t) \), \( R \) is the robot superscript in \( f^{(R)} \), \( n_t \) is the number of agents at time \( t \), and \( z_{1:t} \) are the measurements of all the agents from time \( 1:t \). The coupling in IGP is achieved with a multiplicative potential function that models cooperative collision avoidance. On the one hand, IGP is a forecast of the crowd’s evolution in time. On the other, the density is interpretable as a navigation protocol: choose robot actions according to the maximum a-posteriori (MAP) value. This navigation interpretation is an instance of planning reducing to inference, a concept formalized in [15].

However, for each agent, IGP assumes a single, deterministic goal. Furthermore, the goal arrival time is assumed known in advance. Although these assumptions are suitable in simulation, real world agents (such as cafeteria patrons) have multiple probabilistic goals as well as stochastic goal arrival times. In Section 2.1, we address the “single, deterministic goal with known arrival time” assumption with multiple goal interacting Gaussian processes (mgIGP). Section 2.1 extends the planning and inference methods used for IGP in [5] to the case of mgIGP.
Gaussian Processes for Single Goal Trajectory Modeling  
IGP models each agent’s trajectory as a random function distributed as a Gaussian Process ([16]), \( f^{(i)} \sim GP(f^{(i)}; 0, k) \). New measurements \( z_{t+1}^{(i)} \) update the GP to \( p(f^{(i)} | z_{t+1}^{(i)}) = GP(f^{(i)}; m_{t+1}^{(i)}, k_{t+1}^{(i)}) \), where \( m_{t+1}^{(i)}(t') = \Sigma_{t+1:t,t'}^{T}(\Sigma_{1:t,1:t} + \sigma^2 I)^{-1}z_{t+1}^{(i)} \) and \( k_{t+1}^{(i)}(t_1, t_2) = k(t_1, t_2) - \Sigma_{t+1:t,t'}^{T}(\Sigma_{1:t,1:t} + \sigma^2 I)^{-1}I_{t+1:t,t'} \). Hereby, \( \Sigma_{1:t,t'} = \begin{bmatrix} [k(1, t'), k(2, t'), \ldots, k(t, t')] \\ \end{bmatrix} \) and \( \Sigma_{1:t,1:t} \) is the matrix such that the \((i,j)\) entry is \( \Sigma_{i,j} = k(i,j) \) and the indices \((i,j)\) take values from \( 1 : t \).

In practice there may be uncertainty between multiple, discrete goals that an agent could pursue; for this reason, we develop a novel probabilistic model over waypoints and the transition potential \( R \). That is, online learning of the kinematic models of the agents occurs.

Although the hyperparameters are fixed once they are learned, the actual mean and kernel functions change at each time step; with GPs then, the kinematic model for both the humans and the robot is adaptive. That is, online learning of the kinematic models of the agents occurs.

Modeling Robot Human Cooperation  
IGP couples each individual agent model via an interaction potential \( \psi(f^{(R)}, f) = \psi(f^{(R)}, f^{(1)}, f^{(2)}, \ldots, f^{(n)}) \) that results in a joint model over the \( n_t + 1 \) agent function space:

\[
p(f^{(R)}, f | z_{1:t}) = \frac{1}{Z} \psi(f^{(R)}, f) \prod_{i=R}^{n} p(f^{(i)} | z_{1:t}^{(i)}),
\]

using the notation \( f = (f^{(1)}, \ldots, f^{(n)}) \). We choose \( \psi \) to be

\[
\psi(f^{(R)}, f) = \prod_{i=R}^{n} \prod_{j=i+1}^{T} \prod_{\tau=t}^{\infty} \left( 1 - \frac{\alpha}{\exp\left( \frac{1}{2\pi} |f^{(i)(\tau)} - f^{(j)(\tau)}| \right)} \right)
\]

where \( |f^{(i)}(\tau) - f^{(j)}(\tau)| \) is the Euclidean distance at time \( \tau \) between agent \( i \) and agent \( j \). The rationale behind our choice is that any set of paths \( f^{(R)}, \ldots, f^{(n)} \) becomes very unlikely if, at time \( \tau \), two agents \( i \) and \( j \) are too close. The parameter \( \alpha \) controls the “safety margin”, and \( \alpha \in [0, 1] \) its strength.

Reducing Planning to Inference  
Our model \( p(f^{(R)}, f | z_{1:t}) \) immediately suggests a natural way to perform navigation: at time \( t \), find the MAP value \( (f^{(R)*}, f^*) = \arg \max_{f^{(R)}, f} p(f^{(R)}, f | z_{1:t}) \), and take \( f^{(R)*}(t+1) \) as the next action in the path (where \( t+1 \) means the next step of the estimation). At \( t+1 \), we receive observations, update the distribution to \( p(f^{(R)}, f | z_{1:t+1}) \), find the MAP, and choose \( f^{(R)*}(t + 2) \) as the next step. This process repeats until the robot arrives at its destination.

2.1 Multiple Goal Interacting Gaussian Processes

In practice there may be uncertainty between multiple, discrete goals that an agent could pursue; similarly, it is exceedingly rare to know in advance the time it takes to travel between these waypoints. For this reason, we develop a novel probabilistic model over waypoints and the transition time between these waypoints by generalizing GPs to a mixture of GPs interpolating between waypoints. Figure 1(c) illustrates our motivation. We begin with the assumption that the environment in which we will be doing trajectory prediction has a fixed number of goals \( G \) (corresponding roughly to the number of eating stations in the cafeteria): \( g = (g_1, g_2, \ldots, g_G) \). For the purposes of this
analysis, we restrict the distributions governing each goal random variable to be Gaussian. We also restrict our goals $g_a$ to lie in the plane $\mathbb{R}^2$.

In order to learn the distribution of the goals $g$, we gridded the cafeteria floor, collected frequency data on pedestrian linger time within each cell, and then used Gaussian Mixture Model clustering ([17]) to segment the pedestrian track data into “hot spots”. In particular, we learned $p(g) = \sum_{k=1}^{G} \beta_k N(g; \mu_{g_k}, \Sigma_{g_k})$ where $\beta_k$ is the weight of each component learned, $\mu_{g_k}$ is the mean of the goal location, and $\Sigma_{g_k}$ is the uncertainty around the goal. The ovals in Figure 1(c) illustrate this idea. Given $p(g)$, we derive, from experimental data, the transition probability $p(g_a \rightarrow g_b)$ for all $a, b \in \{1, 2, \ldots, G\}$. For transitions between two goals $g_a \rightarrow g_b$, we learn $p(T_{a \rightarrow b})$, the density governing the duration random variable $T_{a \rightarrow b}$. Finally, we introduce a waypoint sequence $g_m = (g_{m_1} \rightarrow g_{m_2} \rightarrow \cdots \rightarrow g_{m_F})$, composed of waypoints $g_{m_k}$ with $m_k \in \{1, 2, \ldots, G\}$, for locations indexed by $m_1, m_2, \ldots, m_F$ where $F \in \mathbb{N}$, with associated waypoint durations $T_m = \{T_{m_0 \rightarrow m_1}, T_{m_1 \rightarrow m_2}, \cdots, T_{m_{F-1} \rightarrow m_F}\}$ where $T_{m_0 \rightarrow m_1}$ is the time to the first goal.

**Generative process for a sequence of waypoints** We now describe a generative process for a sequence of waypoints that we will use as a prior in our Bayesian model. Beginning with the set of goals $g$, we draw indices from the set $\{1, 2, \ldots, G\}$. The first index is drawn uniformly at random, with the following indices drawn according to $p(g_a \rightarrow g_b)$. Simultaneously, we draw transition times $T_{a \rightarrow b} \sim p(T_{a \rightarrow b})$. Thereby, a possibly infinite series of waypoints and transition times is generated.

We formulate agent $i$‘s prediction model by marginalizing over waypoint sequences $g_m$ and durations $T_m$: $p(f^{(i)} | z^{(i)}_{1:t}) = \int_{g_m, T_m} p(f^{(i)} | g_m, T_m, z^{(i)}_{1:t})$. Using the chain rule, we have

$$p(f^{(i)} | z^{(i)}_{1:t}) = \int_{g_m, T_m} p(f^{(i)} | z^{(i)}_{1:t}, g_m, T_m) p(g_m, T_m | z^{(i)}_{1:t}). (2.2)$$

Notice that for each goal sequence $g_m$, we potentially have a different number of waypoints $g_{m_k}$.

The mgIGP density is Equation 2.1 with the mixture models substituted for $p(f^{(i)} | z^{(i)}_{1:t})$.

**Cooperative Planning and Inference** We introduce a sampling based inference algorithm for the mgIGP density. We interpret mgIGP as a navigation density, and derive action commands according to Section 2. We employ two different sampling steps to approximate the mgIGP density: a sample based approximation of the mixture process Equation 2.2, and a sample based approximation of the mgIGP posterior.
Sample based approximation of mixture models  Since Equation 2.2 is intractable, we employ a sample based approximation: \( p(\mathbf{g}_m, \bar{T}_m \mid z_{1:t}) \approx \sum_{k=1}^{N_p} w_k^{(i)} \delta \left[ \left( \mathbf{g}_m, \bar{T}_m \right)_k - \left( \mathbf{g}_m, \bar{T}_m \right) \right] \), where we utilize the empirically derived density \( \left( \mathbf{g}_m, \bar{T}_m \right)_k \sim p(\mathbf{g}_m, \bar{T}_m) \) and \( N_p \) samples. Substituting \( \sum_{k=1}^{N_p} w_k^{(i)} \delta \left[ \left( \mathbf{g}_m, \bar{T}_m \right)_k - \left( \mathbf{g}_m, \bar{T}_m \right) \right] \) into Equation 2.2, we generate

\[
p(f^{(i)} \mid z_{1:t}) \approx \sum_{k=1}^{N_p} w_k^{(i)} p(f^{(i)} \mid z_{1:t}, \mathbf{g}_k, \bar{T}_k).
\]  

(2.3)

The samples collapse the infinite sum of integrals to one finite sum. This illustrated in Figure 1(c).

In order to generate samples \( (\mathbf{g}_k, \bar{T}_k) \), we draw a sequence of waypoints \( \mathbf{g}_k \) and then the corresponding sequence of waypoint durations \( T_{k_0 \rightarrow k_t} \). To draw the waypoints, we sample \( \mathbf{g}_k \), uniformly from the set \( G \) goals. We then draw \( T_{k_0 \rightarrow k_t} \) according to a distribution with mean given by the average time to travel from the current point to \( \mathbf{g}_{k_t} \). Then, \( \mathbf{g}_{k_2} \) is drawn according to \( p(\mathbf{g}_{k_2} \mid \mathbf{g}_{k_1}) \), and \( T_{k_1 \rightarrow k_2} \) is consequently sampled. We continue until the sum of the duration waypoints reaches or exceeds \( T_{\text{max}} \), and then drop the most recently sampled goal.

Additionally, we evaluate the individual mixture component weights according to \( w_k^{(i)} = p(\mathbf{g}_m, \bar{T}_m) \bigg| (z_{1:t}) \propto p(z_{1:t} \mid (\mathbf{g}_m, \bar{T}_m)) \). That is, we evaluate the likelihood that \( z_{1:t} \) is true conditioned on the waypoint-duration pair \( (\mathbf{g}_m, \bar{T}_m) \). Specifically, \( p(z_{1:t} \mid (\mathbf{g}_m, \bar{T}_m)) \) is a GP conditioned on \( (\mathbf{g}_m, \bar{T}_m) \) and the first measurement \( z_1 \), and evaluated over \( z_{2:t} \).

Sample based approximation of mgIGP  We expand the mgIGP density to take goal and waypoint duration uncertainty into account by using Equation 2.3:

\[
p(f^{(R)} \mid f, z_{1:t}) = \frac{1}{Z} \psi(f) \prod_{i=1}^{N} p(f^{(i)} \mid z_{1:t}, \tilde{g}_i)
\]

\[
= \frac{1}{Z} \psi(f) \prod_{i=1}^{N} \left( \sum_{\mathbf{g}_m} \int_{\bar{T}_m} p(f^{(i)} \mid \mathbf{g}_m, \bar{T}_m \mid z_{1:t}) \right)
\]

\[
\approx \frac{1}{Z} \psi(f) \prod_{i=1}^{N} \left( \sum_{k=1}^{N_p} w_k^{(i)} p(f^{(i)} \mid z_{1:t}, \mathbf{g}_k, \bar{T}_k) \right).
\]

We wish to approximate \( p(f^{(R)} \mid f, z_{1:t}) \) using samples. To do this, we extend the method outlined in [5] by adding a step to account for the multiple GP components—that is, to draw a joint sample \( (f^{(R)}, \bar{T}) \) from the mgIGP density we first draw agent \( i \)'s mixture index from the discrete distribution \( \{w_1^{(i)}, w_2^{(i)}, \ldots, w_{N_p}^{(i)}\} \). Given the mixture index \( h \), we draw \( (f^{(i)})_t \sim p(f^{(i)} \mid z_{1:t}, \tilde{g}_h, \bar{T}_h) \). We iterate through all \( n_t + 1 \) agents (including the robot), and then arrive at the joint sample weight \( \eta_t = \psi((f^{(R)}, f)_t) \). With this collection of \( N_{\text{mgIGP}} \) weights, we arrive at the approximation

\[
p(f^{(R)} \mid f, z_{1:t}) \approx \sum_{l=1}^{N_{\text{mgIGP}}} \eta_l \delta[(f^{(R)}, f)_t - (f^{(R)}, f)].
\]

3 Experiments  In this Section, we study the navigation of a Pioneer 3-DX® differential drive mobile robot through dense crowds in a public cafeteria. The purpose of these experiments is to understand how cooperative navigation models affect robot safety and efficiency in human environments. To that end, we tested the following five navigation protocols: the mgIGP algorithm, the single goal IGP algorithm, a noncooperative planner (this algorithm predicts individual trajectories using the Gaussian process mixture models, but does not couple agent behavior), and a reactive planner, based on the
Dynamic Window approach of [18]. As an “upper bound” on navigation safety and efficiency, we benchmarked line of sight tele-operation.

Figure 1(a) provides an image of the actual robot workspace used in our experiments. Due to the available coverage of our pedestrian tracker, robot motions were limited to a 20m\(^2\) area between the buffet station, the pizza counter, and the soda fountain. Our pedestrian tracking system utilized three Point Grey Bumblebee2 stereo cameras mounted in an overhead configuration (Figure 1(b)) with overlapping workspace at a nominal height of 3.5m.

**Metrics** We discuss the human density metric. First, we have normalized to values between 0 and 1—thus, the highest density (1 person/m\(^2\)) is a shoulder to shoulder crowd. Further, patrons rarely stand still; this constant motion increases crowd complexity. Anecdotally, humans found crowd densities above 0.8 people/m\(^2\) to be extremely difficult to teleoperate through, and densities above 0.4 people/m\(^2\) challenging. We define safety as a binary variable: either the robot was able to navigate through the crowd without collision or it was not. Obviously, we could not allow the robot to collide with either walls or people, and so a protocol for the test operator was put in place: if the robot came within 1 meter of a human, and neither the robot nor the human was making progress towards avoiding collision, then the robot was “emergency stopped” (the velocity command was set to zero). Navigation efficiency is defined as the time elapsed from the start of the algorithm until arrival at the goal.

**Safety of Noncooperative Planner and mgIGP Planner** In Figure 2(a), we compare the safety of our state of the art noncooperative planner to the mgIGP planner. This data suggests the following: cooperative collision avoidance models can improve overall safety by up to a factor of \(0.63/0.19 \approx 3.31\). Additionally, the noncooperative planner is unsafe more than 50\% of the time at densities as low as 0.3 people/m\(^2\) and above. At densities of 0.55 people/m\(^2\) and above, it is unsafe more than 80\% of the time. In contrast, mgIGP is unsafe less than 30\% of the time for densities up to 0.65 people/m\(^2\); at densities near 0.8 people/m\(^2\), mgIGP is still safe more than 50\% of the time. The noncooperative planner is unsafe over 90\% of the time at this high density. Finally, the safety of both planners degrades reliably as crowd density increases (both planners cease to be safe above 0.8 people/m\(^2\)).

We present the following explanation: the noncooperative planner believes itself invisible, and so has trouble finding safe paths through the crowd, and thus tries to creep along the walls of the testing area. This resulted in many failed runs: the robot’s movement is not precise enough to avoid...
collisions when “wall hugging”. More generally, this is a manifestation of the freezing robot problem of [5]. In contrast, failures for mgIGP were rare because the robot was more likely to engage the crowd. By engaging the crowd, the robot elicited cooperation, which made navigation safer.

Safety of Noncooperative Planner and IGP planner In Figure 2(b), the noncooperative planner is compared to a “compromised” cooperative planner, IGP. The noncooperative planner retains the Gaussian process mixture model. Although the results are not as stark as in Section 3, IGP is still $0.63/0.28 \approx 2.25$ times as safe as the noncooperative planner. This result suggests that for navigation in dense crowds, modeling cooperation is more important than high fidelity individual trajectory predictive models.

Efficiency of mgIGP Planner, Reactive Planner, and Human Tele-Operation In figure 3(a), we present the efficiency for the reactive planner, the mgIGP planner, and human tele-operation. This figure demonstrates that, for most crowd densities, mgIGP was nearly as efficient as human tele-operation. We point out that (by definition) the human tele-operators never had to be emergency stopped. Whereas the efficiency of all the other planners (including human tele-operation) increased roughly linearly with crowd density, the reactive planner appears to grow super-linearly with crowd density. Additionally, no runs for the reactive planner were collected for densities above 0.55 people/m$^2$. This was a result of the following: when the reactive planner started at a density above 0.55 people/m$^2$, it moved extremely slowly. If the crowd density was too high, it stopped moving forward entirely. Essentially, the reactive algorithm was waiting until the density was low enough to ensure safety. By this time, however, the average crowd density over the duration of the run had dropped substantially from the maximum crowd density. Thus, the reactive algorithm was unable to make progress through a crowd with an average density above 0.55 people/m$^2$.

In Figure 3(b) we present the efficiency results for the mgIGP planner, the IGP planner, and human tele-operation. This figure provides insight into how efficiency is affected when the Gaussian process mixture model of independent trajectories is removed from the interactive formulation. Human tele-operation serves as an upper bound on efficiency.

Qualitative Studies A highly useful behavior of the robot was that it was always in motion. This was achieved safely by doing the following: if a collision was imminent, the forward velocity was set to zero. However, the rotational velocity was not set to zero. The navigation algorithm continued generating new plans (even though the forward velocity was held at zero until collision was not imminent), and each new plan potentially pointed the robot in a new direction. Indeed, the robot was searching for a way through a challenging crowd state (see movie snippet at http://resolver.caltech.edu/CaltechAUTHORS:20120911-130046401).
Sometimes, this resulted in quite humorous situations: at the beginning of one run, while the navigation algorithm was still starting up, a patron came up and began inspecting the robot. The robot, sensing an imminent collision, set its velocity to zero, and began searching for a clear path (i.e., rotating in place). The patron realized what was happening, and moved along with the robot, constantly staying in front of the robot’s forward velocity vector. This resulted in what we have since called the “robot dance” (see movie snippet at http://resolver.caltech.edu/CaltechAUTHORS:20120911-125945867).

This behavior can be quite useful in dense crowds. For instance, the reactive robot did not display this behavior—when a collision was imminent, it stopped completely. Unfortunately, a completely stopped robot is very hard for a human to understand. Is this robot turned off? Is this robot waiting for me? Meanwhile, the mgIGP robot displayed intentionality (see movie snippet at http://resolver.caltech.edu/CaltechAUTHORS:20120911-125828298). Animators call this behavior “readability”, and it can be employed to create a more human like intelligence (see [19]).

References