# Using modified incremental chart parsing to ascribe intentions to animated geometric figures

**David Pautler · Bryan L. Koenig · Boon-Kiat Quek ·
Andrew Ortony**

**Abstract** People spontaneously ascribe intentions on the basis of observed behavior, and research shows that they do this even with simple geometric figures moving in a plane. The latter fact suggests that 2-D animations isolate critical information—object movement—that people use to infer the possible intentions (if any) underlying observed behavior. This article describes an approach to using motion information to model the ascription of intentions to simple figures. Incremental chart parsing is a technique developed in natural-language processing that builds up an understanding as text comes in one word at a time. We modified this technique to develop a system that uses spatiotemporal constraints about simple figures and their observed movements in order to propose candidate intentions or non-agentive causes. Candidates are identified via partial parses using a library of rules, and confidence scores are assigned so that candidates can be ranked. As observations come in, the system revises its candidates and updates the confidence scores. We describe a pilot study demonstrating that people generally perceive a simple animation in a manner consistent with the model.

D. Pautler (✉) · B. L. Koenig · B.-K. Quek · A. Ortony
Computational Social Cognition, Agency for Science,
Technology and Research (A*STAR),
Institute of High Performance Computing,
1 Fusionopolis Way, #16-16,
Connexis 138632, Singapore
e-mail: pautlerd@ihpc.a-star.edu.sg

B. L. Koenig
Department of Psychology, National University of Singapore,
Block AS4, 9 Arts Link,
Singapore 117570

A. Ortony
Department of Psychology, Northwestern University,
Evanston, IL, USA

For social robots and intelligent systems to interact with humans in a believable and humanlike manner, they will have to be able to ascribe mental states (e.g., intentions, beliefs, and desires) to the people with whom they interact. Humans routinely ascribe mental states, even in infancy. For example, 3-month-olds attribute agency to self-propelled boxes (Luo, 2010), and 6-month-old infants can distinguish helpful versus hindering agents (Hamlin, Wynn, & Bloom, 2007). As adults, we ascribe mental states automatically, even in response to simple geometric objects moving in a 2-D plane (Scholl & Tremoulet, 2000). In a classic study (Heider & Simmel, 1944), people who observed two triangles and a circle moving against a white background (see Fig. 1) reported perceiving actions (e.g., chasing and hiding), intentions (e.g., to catch and to harm), and emotions (e.g., jealousy), and even sophisticated relationships (e.g., a love triangle). How do natural cognitive systems infer higher-order mental states from sparse, yet dynamic, spatially displayed information that is no more than moving objects in a 2-D plane? Can artificial systems be designed to make similar inferences?

## Animated 2-D objects: A good place to start

Given the complexity of the human social world, it might seem overly simplistic to draw inferences about intentions using only information from the movement of 2-D objects. Indeed, natural human social perception is based on information from a variety of sources: verbal and nonverbal
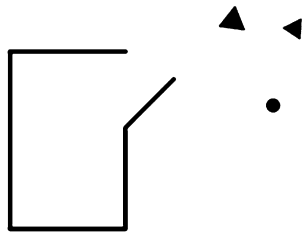
**Fig. 1** A frame from the Heider and Simmel (1944) animation. Available at http://anthropomorphism.org/img/Heider_Flash.swf

behavior (McNeill, 1992), background knowledge (Andersen & Klatzky, 1987), and biases internal to the observer (Maner et al., 2005; Waytz, Cacioppo, & Epley, 2010). Prior computational research has successfully categorized observed human actions by intention, but it has been somewhat restricted with respect to domain. For example, the robot *Nico* is capable of observing a group of people playing tag and identifying the chaser with near-human accuracy (Crick, Doniec, & Scassellati, 2007). But social inference goes well beyond identifying who is chasing whom, and the power of 2-D animations to evoke rich attributions, while also being perceptually simpler than other kinds of social information, makes 2-D animations well suited as a source of inspiration for a perceptually driven cognitive model of intention ascription. By constraining the input, 2-D animations simplify what is already a challenging computational task. Furthermore, the ease with which modern software can produce animations facilitates the generation of stimuli for use in experimental evaluation of models by comparing their inferences with those of human participants.

## Research objectives

Overall, our goal is to develop a computer system that could ascribe explanations similar to those provided by humans when they observe simple animations of moving shapes. We also want the system to be able to scale, beginning with a few types of explanations based on intentions and simple physical causes, and then expanding in terms of more intentions, other psychological states and traits, and potentially even other influences on observers, such as preinformation. This section describes the six key objectives that distinguish our approach in terms of increased cognitive plausibility and scalability.

The first objective is that the space of possible explanations should include not just *instrumental goals/intentions*, but also more social intentions, such as those found by Heider and Simmel (1944), in which *one agent tries to influence the thinking of another* (see also Abell, Happé, & Frith, 2000), as well as explanations based on *purely physical causes* (e.g., a preceding collision). Prior computational approaches to

generating explanations using observed movements have focused on ascribing physical causes only (Forbus, Usher, Lovett, Lockwood, & Wetzel, 2008; Siskind, 2003) or have ascribed intentions such as "chasing" or "playing tag" that involve more than one agent (and thus are social). These intentions typically do not suggest the more socially sophisticated ability in which one agent factors the thoughts of another into its plans (Barrett et al., 2005; Blythe, Todd, & Miller, 1999; Crick & Scassellati, 2008; Kerr & Cohen, 2010; Young, Igarashi, & Sharlin, 2008).

Second, just as people find that some animations can be explained in multiple ways unless and until decisive evidence emerges for or against candidate alternatives, a computer system should be able to generate and evaluate *multiple* candidates *all through the action*, not only at the end (because everyday experience rarely provides such convenient ending points). For example, one agent might flee from another only to turn and fight if cornered. The fleeing and fighting intentions are different, yet related. The prior computational approaches to ascribing intention mentioned above typically have used animations or movies suggestive of single intentions only, in which explanations are generated only after the animation ends. Our goal of being able to generate explanations at any time as events unfold during an animation is inspired by the psychological research of Newtson (1973) and Zacks and colleagues on event segmentation (i.e., how humans parse continuous activity into discrete events). Reynolds, Zacks, and Braver (2007) created a neural net that monitors movement properties (using hand-encodings of movies such as *The Red Balloon*) and detects *event boundaries*, or points at which one action ends and another begins. Although their approach is relevant to our objective of generating explanations as events unfold, it does not address our focal goal of generating *explanations*.

Third, *folk causal theories* rather than scientific theories should be used to model the knowledge drawn upon to generate explanations, because that is what we can only assume people use naively. Previous physical cause-ascribing research tended to use Newtonian laws to represent the explanations generated by everyday people, rather than folk concepts such as impetus (Kozhevnikov & Hegarty, 2001; McCloskey, 1983). Similarly, folk psychology should be used when possible instead of scientific psychological constructs for explanations of the behavior of agents.

Fourth, animations should have *rich environments*, similar to Heider and Simmel's (1944; e.g., obstacles should be present) to allow for richer, more social interpretations. Previous intention-ascribing research tended to use simple or empty environments (e.g., no obstacles; Blythe et al., 1999). Such research may overfit movement statistics on animations without obstacles, potentially resulting in miscategorization of intentions for animations *with* obstacles.

Fifth, initial versions of animations should be designed to tap just one or very few cues (we focus initially on movement-related cues) with the expectation that other kinds of cues (e.g., resemblance to real-world objects or creatures) can be added later. Ideally, *adding cue-based confidence functions or new types of intention should not require altering the representations of cues and categories already demonstrated to work*. Some previous strategies relied on optimized search procedures tailored to a set of intention categories (e.g., the relative speeds of two agents as a cue for a chasing intention; Blythe et al., 1999). Methodologically, this approach presents a problem whenever adding an intention category, because the set of criteria used to distinguish among the previous set of intentions may no longer be optimal for the new, larger set. Potentially, a new set of criteria would need to be collected for both new *and* old categories. That is, researchers would risk having to re-collect data whenever adding new categories.

Finally, the method of constructing and rating candidates should reflect *known psychological cues* that people use when interpreting similar animations (e.g., spatial context; Tremoulet & Feldman, 2006). Tenenbaum and colleagues have published several influential studies on computational models of intention ascription (Baker, Saxe, & Tenenbaum, 2009; Goodman, Baker, & Tenenbaum, 2009; Ullman et al., 2010). The core of their approach is modeling such ascriptions as Bayesian inferences on Markov decision processes (MDPs). For example, Baker et al. (2009) designed their MDP system to assume that observed movement is the result of an agent that is behaving *rationally.* Their system performed well given the constraints that the moving object can only be an agent with goals of going to different locations on a grid. We have not followed the MDP approach for two reasons. First, we believe that a utility-oriented interpretation would stretch the meaning of "utility" for most cues identified in the literature on the perception of agency. For example, how would utility explain the "sweet spot" for object speed relative to the background (Morewedge, Preston, & Wegner, 2007)? Or, how does utility explain cues internal to the observer, such as the reduction of attributed agency by people experiencing social isolation (Waytz et al., 2010)? Second, the assumption that the agent is rational implies that the system should perceive straight-line movement toward a potential goal as more agentic than any other movement, but Tremoulet and Feldman (2006) found just the opposite: Observers reported that movement paths toward a goal in which there was a change of direction in the path appeared more agentic than a straight movement path toward a goal.

In addition to our six objectives, we also want to compare our approach with Thibadeau's (1986) contributions. He used a hand-coded representation of the Heider and Simmel (1944) animation, together with a schema-based representation of intentions and relevant acts, to generate explanations. The system was designed to generate only one explanation candidate per event, although it could join sequential candidates into larger candidate narratives. Our approach expands on Thibadeau's in several ways: by adding "theory of mind" and physical cause explanation types, by extracting an input representation directly from an animation file, by generating and managing multiple candidate explanations, and by providing hooks for cue-based confidence-scoring functions to guide the ranking of candidates.

Our claim is that a parser-based abduction approach to simulating human attribution to animations, with hooks for adding functions that simulate the influence of cues on explanation confidence, is better suited to the objectives outlined above than prior work has been.

## Approach

In order to create a computational system that is able to "watch" an animation unfold and update its interpretations at the same time points as humans and in ways similar to theirs, we must answer several questions: On what kinds of ascriptions should one focus? How should the animation be encoded so that the system's view is roughly the same as that of human participants? (Basically, what is the form of a system's input?) How should the system track multiple objects across frames in order to determine their movement? How should the system generate ascriptions based on the movements of objects? How should the system connect its explanations in order to construct larger coherent narratives like people do? The following subsections describe how the design of our system, Wayang[1], addresses these questions.

## Targeted types of ascriptions

On what kinds of ascriptions should one focus? For example, participants in the Heider and Simmel (1944) experiments reported perceiving intentional actions, social roles, emotional states, personality traits, and even failed plans. Given all of these possible types of inferences, what should be the scope of a computational model's output? Rather than attempt to produce a model that could make all of the kinds of ascriptions made by participants in the Heider and Simmel experiment, our initial instantiation focuses primarily on inferring intentions. It also attributes inanimate physical causes (as alternatives to intentions). We

---

[1] "Wayang" is an Indonesian word for theater (literally, "shadow").

focused on intentions for three reasons. First, understanding people's intentions is helpful for predicting their future actions. Second, intention ascriptions are important in moral judgments (Hauser, 2006) and legal reasoning about past actions. Both of these reasons highlight the importance of intention in social interaction. Finally, the large body of research in artificial intelligence (AI) on plan recognition (i.e., intention recognition; see, e.g., Geib & Goldman, 2009) and on perception-as-abduction (e.g., Feldman, 2007; Shanahan, 2005) provides a fertile resource from which we can draw when formulating our system. The targeted set of explanation types is similar to the three-part distinction among "theory of mind," "goal-directed," and "random" ascriptions used in the psychological work of Happé, Frith, and colleagues (e.g., Abell et al., 2000), although we replace the "random" category with "physical causes."

To guide our initial selection of intentions, we created an animation that both highlights "goal-directed" intentions (inspired by animations developed by Happé, Frith, and colleagues; e.g., Abell et al., 2000; Castelli, Happé, Frith, & Frith, 2000) and resembles animations to which people might sometimes attribute physical causes rather than intentions (see, e.g., Wolff, 2007). We also developed a control animation so that we could demonstrate that participants responding to our intention animation weren't simply conforming to our expectations, but instead relied on cues in the animation itself. We used Adobe Flash CS4 Professional to develop the animations. The animation size was 500 x 400 pixels, and the circle diameters were 22 pixels. The animation ran at 24 frames per second. (See the Appendix for further animation design considerations.)

As illustrated in Fig. 2, the animation involves only one moving object, X. After a momentary pause, object X initially moves linearly up and to the right, such that V is behind X's direction of movement, while X's trajectory is toward Z and to the left of Y (see the top two input frames in Fig. 2). Wayang generates several explanations that are consistent with this initial movement:

> X intends to be farther from V.
> X intends to be closer to Z.
> X intends to be closer to Y.
> A physical force attracts X to (an immobile) Z.
> A physical force repels X from (an immobile) V.

Notice that whereas the first four explanations attribute agentic intentions, the last two attribute physical causes. Human observers, of course, might attribute causes in addition to those listed above.

The dotted outlines in the figure indicate predicted locations. In particular, dotted circles are predictions that a figure will remain stationary, and cone-shaped outlines are predictions that an object will move linearly or along a curve in a specific direction and within a distance.
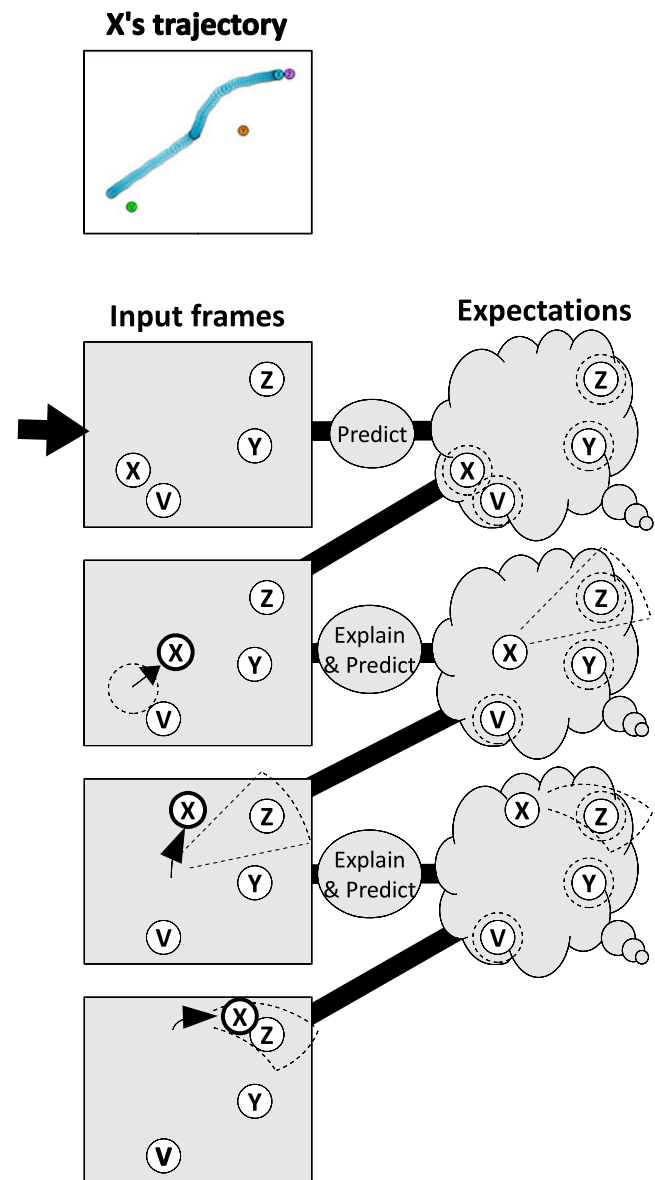


Fig. 2 Proposed alternating sequence of input frames and expectations about object locations. (Note: The actual animation has 164 frames.)

Predictions are a natural byproduct of Wayang's explanation-generating process, corresponding to parts of a partially matched rule that might match upcoming inputs (described below).

In the remainder of the animation, X comes to a momentary halt as it nears Y, then continues on a clockwise, circumventing trajectory around Y and toward Z, eventually contacting Z and staying there until the end of the animation (see the bottom two input frames in Fig. 2). During these events, the system adds to its set of explanations. In particular, we envisioned this sequence of events to result in a realization (in human observers and the system) that both movements could be explained simulta-

neously by assuming that X had two competing intentions: to be near Z and to avoid Y.

A pilot study confirmed that people generally perceive this animation as we predicted. Participants were unpaid volunteers recruited via email sent to colleagues and acquaintances naive to the research goals. They completed the study over the Internet. Thus, their displays may have varied the absolute size of the animations, but the relative sizes and speeds of the animation components were maintained. We had 38 volunteers watch an animation similar to the one described above (the "intention animation") and 35 volunteers watch an animation of the same length in which the objects had the same initial and final locations, but in which X moved in a straight line at a constant speed (the "control animation"). Three researchers coded participants' descriptions of what they thought "appeared to happen in the animation" (minimum Cohen's $\kappa = .58$). Spontaneous ascription to X of trying to get to Z occurred in most descriptions of the intention animation (26 of 38), but in only a few descriptions of the control animation (7 of 35), $\chi^2(1, N = 73) = 17.25, p < .001$. Ascription to X of trying to avoid Y occurred in many descriptions of the intention animation (15 of 38), but was absent in those of the control animation (0 of 35), $\chi^2(1, N = 73) = 17.39, p < .001$. Both animations can be viewed at http://csc.ihpc.a-star.edu.sg/archive/inferringIntent/BRM2011.htm.

In sum, the attributions we targeted for Wayang's output are intentions (or physical causes) that explain single movements, plus narratives that coherently explain multiple intentions and/or physical causes.

## Encoding of space and time for animations

How should the animation be encoded so that the system's view is roughly the same as that of human participants? Human visual perception is calibrated to the range of space–time in which humans live their daily lives. Some aspects of an animation, such as small loops or kinks in a trajectory, may be below human perceptual awareness but "noticeable enough" for a computational system, given a high-precision rendering. In such a case, the system's explanation of the animation may differ substantially from that of a human observer. Since we want to compare the output of our computational system with that of humans, we need to scale the encoding of the animations so that it is roughly comparable to that of human perception.

Regarding temporal encoding, one useful guideline comes from the study of "flicker fusion" in psychophysics, from which we know that humans perceive objects that "jump" short distances from one frame to the next as appearing to have continuous and unbroken movement

when frame rates are increased to about 12 frames per second (fps; Anderson & Anderson, 1993). Regarding spatial encoding, numerous features of the human perceptual system, including saccades and reduced resolution as one moves outward from the center of the fovea, make the standard computational approach to image encoding—a uniform coordinate grid for the entire scene—an imperfect fit for this application. Nevertheless, including such factors would greatly complicate the system, probably without improving it, so we have adopted a working assumption that the unit of spatial encoding should be 1 mm as seen from 50 cm away (i.e., 0.002 deg of arc).

Wayang currently processes only position information (i.e., frame-by-frame locations of otherwise unchanging, uniquely colored circles, from which the system can calculate movements). Wayang is not given advanced conceptual information—for example, that the shapes represent agents—and animations currently use only circles of a single, constant size in order to exclude orientation or other structural information. Once Wayang's rules are able to generate explanations solely from positional and movement cues, more scenarios will be added that will involve cues such as orientation, iconic resemblance to real-world objects, and so on.

## Tracking objects across frames

How should Wayang track multiple objects across frames in order to determine their movement? For example, if there are three identical objects in one frame and three identical objects in different positions in the next frame, which objects in the second frame correspond to those in the first? This is a well-known problem in computer vision, and we sidestep it by manually labeling all objects in our input frames. (In fact, a common technique for handling this problem in computer vision, "multiple hypothesis tracking" developed by Mann, Jepson, & El-Marghi, 2002, is similar to the chart-parsing algorithm we use for managing explanations across frames.)

## Generating explanations

How should Wayang generate explanations based on the movement of objects? Although we want eventually to accommodate top-down influences, for our first instantiation, clearly bottom-up information primarily drives this process (because all cues other than movement are absent). A hint at what intermediary representations people might generate bottom-up is provided by the participants in Heider and Simmel's (1944) experiment, some of whom described the action in purely geometric terms. One way to interpret these responses in the context of the majority of

responses is to view geometric description as an intermediate step between samples of object positions and ascriptions of causes—perhaps the minority who gave geometric descriptions simply did not go beyond the intermediate representation.

*Targeted features for the algorithm* A search of the AI literature for an algorithm that could generate multiple levels of description, bottom-up, as new inputs arrive, and could simultaneously allow for competing descriptions led us to text parsers, specifically *bottom-up incremental chart parsers that use a feature grammar*. Previous scholars have also seen similarities between intention ascription and parsing (e.g., Sidner, 1985). In essence, *chart parsers* apply dynamic programming to partial parse trees. That is, they store partial parse trees both by the spans of the word tokens that each partial parse tree covers and by the grammar rule of the highest level of each partial parse tree. Basically, such parsers store plausible, incomplete interpretations both by the observations underlying the interpretation and by the rules that they applied to the observations to produce the interpretations. Typically, text parsers apply their grammar rules in as many ways as possible to a complete list of word tokens in order to identify all conceivable interpretations. Chart parsing is relatively efficient when text ambiguities support multiple higher-level interpretations (i.e., parse trees). It stores and re-uses lower-level parse trees for relevant interpretations instead of having to regenerate them. This frugality is a key feature of chart parsing for our system, because it suggests more cognitive plausibility over other parsing techniques. Next we briefly justify our choices of chart-parsing techniques between top-down versus bottom-up, end-marker-driven versus incremental, and categorial versus feature grammars (see Gazdar & Mellish, 1989, for an overview of these distinctions.)

A *top-down* parser assumes that it will receive an entire clause, and only one clause, and tries to locate the parts of the clause among the input tokens. A *bottom-up* parser makes no analogous assumptions and must match input tokens directly to grammar rules. Bottom-up parsing is a close match to our targeted scenarios, in which no explanation categories are cued in advance and processing must rely on observed movements. The interface between input word tokens and grammar rules in text parsing is part-of-speech categories (POSs), such as nouns and verbs. In the domain of explaining observed movements, we propose that the corresponding interface between frame-by-frame object locations and explanations could be geometric descriptions of object trajectories. Unlike POSs in text parsing, which align one to one with input word tokens, the proposed trajectory categories accumulate two or more observed positions into segments of uniform acceleration

and direction. In Wayang, these categories currently include *stationary*, *linear*, and *curved* trajectories.

An *end-marker-driven* parser continuously collects input word tokens but waits to apply grammar rules until it encounters an end-marker such as a question mark. An *incremental* parser does not wait (Schwitter 2003). It applies grammar rules after receiving each input token. An end-marker-driven parser has more context at its disposal and can avoid generating spurious partial parses that an incremental parser might make. But convenient end-markers are generally absent in everyday action and in animations, so the system described here takes an incremental approach.

A *categorial grammar* uses only such atomic categories as nouns (N), verb phrases (VP), and clauses (S). A *feature grammar* is similar but allows for (1) labeling categories with attributes such as person and number and (2) constraining tree construction based on attribute values—for example, requiring equality between a subject noun and its verb in person and number (e.g., both must be first-person plural). Text parsing typically needs only one type of attribute constraint, namely, equality (e.g., the number attributes of the subject noun and the verb phrase must be equal). In contrast, movement parsing requires multiple types of constraints. For example, building a linear-trajectory description requires evaluating an observation based on a vector constraint: If it lies along the vector defined by prior observations, it is part of that linear trajectory; otherwise, it is part of a new trajectory. Higher levels of description require other specialized constraints. For example, "chasing" requires a constraint that the pursuer changes its direction of movement so that it might catch the pursued. Our knowledge representation has many types of constraints at different levels of description, making it resemble a feature grammar more than a categorial grammar.

*The main algorithm* Wayang uses the same parsing algorithm given in Gazdar and Mellish (1989, pp. 200–201) with some extensions. Our algorithm is also similar to that from Geib and Goldman's (2009) work on the PHATT system, which uses plan tree grammars for probabilistic plan recognition, although PHATT uses only plans of a fixed recursion depth. Wayang does not rely on a fixed recursion depth because it must be able to accumulate arbitrarily long sequences of observation into a coherent plan. Furthermore, because PHATT uses a Bayesian technique to compute *probability values* as confidence scores for its candidate explanations (so that they can be ranked relative to each other), it must wait for a complete set of candidate explanations before it can start computing scores (so their sum can be normed to 1.0). In contrast, Wayang computes scores heuristically, so that each of its scoring functions can compute its score as soon as all of its

inputs are available. The following pseudocode describes Wayang's main algorithm:

For each new *input token* (i.e., animation frame) do

1. Generate a predicate calculus description of the frame contents[2] that provides the time point at which the frame was observed, the dimensions and RGB color of the background, and for each object, an ID, position, shape/orientation/size description, and RGB color;

2. Filter out objects that would be too small, too similar in color to the background, etc., to be perceptible to a human.

3. Feed the token to the parser.

4. Parse the token using the knowledge base (which is analogous to a set of grammar rules) to generate partial parse trees (interpretations) with associated confidence values.

Wayang has two parts. The first is implemented in Java. It handles the first step above, using the JSwiff 8.0 third-party package for manipulating Flash SWF animation files. It then creates an instance of an ECLiPSe interpreter to perform the second, third, and fourth steps. ECLiPSe (available at eclipseclp.org) is a variant of Prolog that supports constraint logic programming. The following sections describe the knowledge base and then the parser.

*The knowledge base* Wayang's knowledge base uses rules to represent an observer's understanding of the cause–effect structure of the world. Wayang's knowledge base is expressed as grammar rules following this generic format:

Trigger [Trigger$_2$]

=causes=> Effect$_1$ Effect$_2$ ... Effect$_M$

: Contingency$_1$ Contingency$_2$ ... Contingency$_N$

This expression can be read "The listed triggers jointly cause the listed effects if all the listed contingencies are satisfied in the current situation." This format is very similar to that of feature grammar rules:

Category

=can be decomposed as=>

SubCat$_1$ SubCat$_2$ ... SubCat$_M$

: Constraint$_1$ Constraint$_2$ ... Constraint$_N$

For example, the following rule describes how a clause can be comprised of a noun phrase followed by a verb phrase, where the number and person attributes of the phrases agree:

clause(Index0, Index2)

=can be decomposed as=>

nounPhrase(Index0, Index1, NumberNP, PersonNP),

verbPhrase(Index1, Index2,NumberVP, PersonVP)

: NumberNP == NumberVP,

PersonNP == PersonVP.

Figure 3 illustrates how this grammar rule might be applied to a sequence of word tokens. The Index variables mark reference points before and after input tokens, like bookends. That is, the first incoming input token always sits between the indexes 0 and 1, the second between the

---

[2] For example, a frame observed at 41 elapsed milliseconds with a white background 167 mm × 122 mm, containing a blue circle centered at (87 mm, 52 mm) with diameter 13 mm, and a red triangle centered at (61 mm, 35 mm) with its longest inner projection 22 mm long and oriented at 45°, etc., would be rendered as:

frame([timestamp(41),

ground(167, 122, color(255, 255, 255)),

figure(1, position(87, 52), circle(13), color(0, 0, 255)),

figure(2, position(61, 35), triangle(22, 45,
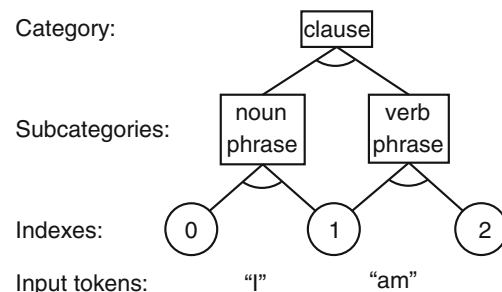
12, 298, 12, 332, 66), color(255, 0, 0))])



**Fig. 3** Example of a parse of the clause "I am" into its components: a noun phrase ("I") and a verb phrase ("am") that agree in number (singular) and person (first)

indexes 1 and 2, and so forth. The *span* of input tokens that a category label covers, such as the span between Index0 and Index2 of the clause, permits incremental chart parsers to index parse trees so that they can be included in more inclusive parse trees if more tokens arrive. Wayang's knowledge base represents spans using time points (integer values in milliseconds) rather than integer token indexes.

Rule R1 below is an example describing how a goal to be at a specific location can cause the goal-holding agent to follow a direct path to that location. Confidence here refers to Wayang's confidence in this inference, not the agent's psychological confidence. At a higher level (not represented here), the agent might want to go to that location because, for example, another agent is there.

```
R1: goal(  Agent,

              atPosition(Agent, (X2, Y2), Timepoint2),

              Timepoint1, Timepoint2,

              Confidence)

  =causes=>

    linearTrajectory(Agent, (X1, Y1), (X2, Y2),

                          (StartingSpeedX, StartingSpeedY),

                          (ConstantAccelX, ConstantAccelY),

                          Timepoint1, Timepoint2)

  : atPosition(Agent, (X1, Y1), Timepoint1),

    X1 != X2, Y1 != Y2,

    hasSpeed(Agent, (StartingSpeedX, StartingSpeedY), Timepoint1),

    hasMaxAbility(Agent, speed, MaxSpeed),

    MaxSpeed >= sqrt(StartingSpeedX^2 + StartingSpeedY^2),

    hasMaxAbility(Agent, acceleration, MaxAccel),

    MaxAccel >= sqrt(ConstantAccelX^2 + ConstantAccelY^2),

    ElapsedTime := Timepoint2 - Timepoint1,

    EndSpeedX := StartingSpeedX + ConstantAccelX*ElapsedTime,

    EndSpeedY := StartingSpeedY + ConstantAccelY*ElapsedTime,

    MaxSpeed >= sqrt(EndSpeedX^2 + EndSpeedY^2),

    X2 == X1 + StartingSpeedX*ElapsedTime + 0.5*ConstantAccelX*ElapsedTime^2,

    Y2 == Y1 + StartingSpeedY*ElapsedTime + 0.5*ConstantAccelY*ElapsedTime^2,

    ThisTrajectory :=

      linearTrajectory(Agent, (X1, Y1), (X2, Y2),

                          (StartingSpeedX, StartingSpeedY),

                          (ConstantAccelX, ConstantAccelY)

                          Timepoint1, Timepoint2),

    findAll(OtherActiveTrajectory,

              (OtherActiveTrajectory != ThisTrajectory,

              onInterceptCourse(ThisTrajectory, OtherActiveTrajectory)),

              []), %/* no figures on intercept course */

    Confidence := fn1(Agent,...,Timepoint2).
```

Rule R1 says, "If Agent has a goal to be at (X2, Y2) at future Timepoint2 and this goal persists between the current Timepoint1 and Timepoint2, if the contingencies are met, then the Agent will follow a linear trajectory (with constant acceleration) from its current position to the desired position, arriving at the desired time." The contingencies confirm that the Agent is not already at the desired position, that the Agent is capable of traveling fast enough to cover the targeted distance in the targeted time, and that the agent knows of nothing it might collide with on the way (assuming omniscience, in this case).

A rule with a physical cause for movement might be:

**R2:** imbuedWithImpetusVia(Attractor,

                        Attractee,

                        linearImpetus((X1, Y1), (XMagn, YMagn),

                        attraction, %/* vs repulsion */

                        Timepoint1, Timepoint2,

                        Confidence)

=causes=>

  linearTrajectory(Attractee, (X1, Y1), (X2, Y2),

                  (StartingSpeedX, StartingSpeedY),

                  (ConstantAccelX, ConstantAccelY),

                  Timepoint1, Timepoint2)

  : atPosition(Attractee, (X1, Y1), Timepoint1),

   atPosition(Attractor, (X3, Y3), Timepoint1),

   atPosition(Attractor, (X3, Y3), Timepoint2),

   X1 != X2, Y1 != Y2,

   X1 != X3, Y1 != Y3,

   collinearOrdered((X1, Y1), (X2, Y2), (X3, Y3)),

   ThisTrajectory :=

   linearTrajectory(Attractee, (X1, Y1), (X2, Y2),

                (StartingSpeedX, StartingSpeedY),

                (ConstantAccelX, ConstantAccelY),

                Timepoint1, Timepoint2),

  findAll(OtherActiveTrajectory,

      (OtherActiveTrajectory != ThisTrajectory,

      onInterceptCourse(ThisTrajectory, OtherActiveTrajectory)),

      []), %/* no figures on intercept course */

  XMagn := X2 - X1,

  YMagn := Y2 - Y1,

  Confidence := fn2(Attractor,…,Timepoint2).

This rule says, "if an object at some position (X1, Y1), the Attractee, is imbued with a linear impetus of magnitude (XMagn, YMagn) by an Attractor object at (X3, Y3) at Timepoint1 through Timepoint2, then at Timepoint2 the Attractee will have traced a linear trajectory ending at some intermediate point (X2, Y2), as long as there were no collisions along the way." A rule about impetus due to repulsion (e.g., between magnets) would be exactly the same, except that its collinear contingency would place the repulsor *behind* the repulsee: collinearOrdered((X3, Y3), (X1, Y1), (X2, Y2)). The concept of *impetus* is similar to that of force, but impetus is conceived as a property given to and held by an object, and it has different contingent effects than force does (e.g., the removal of an attractor does not cancel the impetus it may have imbued in another object).

Note that some constraints permit some flexibility by using configured margins of variance. For example, collinearOrdered computes a best-fit line among its point-coordinate arguments and computes the distance of each point argument from that line, which must be within the configured margin (currently set at 5 mm as a working value).

Unlike the effects or contingencies discussed so far, the final contingency of each rule computes a confidence score, which provides a reason to prefer one candidate over others. Confidence-computing contingencies always evaluate as true. They might depend on any values computed in the rule of which they are a part, so they are placed last. Unlike a Bayesian approach, the confidence functions used in these computations are unconstrained at design time and can be fitted to cue influences revealed by psychological experiments or by Bayesian-type considerations such as the base rate for the occurrence of the rule's triggers. The choice of which variables are relevant and should be passed in as parameters is made at rule implementation time.

The sample rules so far describe how *just one* uniform trajectory can be predicted or explained by a causal trigger. To accommodate arbitrarily long observation sequences under a single explanation (just as real-life plans and recipes comprise heterogeneous and/or recursive steps), some Wayang rules have goal states (and other unobservable states) as effects. For example, the following rule can be used with the preceding goal rule to explain that two aligned linear trajectories (e.g., an agent first accelerating to its stable speed and then continuing at that speed) could mean the agent wanted *all along* to

go to the final observed position:

**R3:** goal(Agent,

       atPosition(Agent, (X3, Y3), Timepoint3),

       Timepoint1, Timepoint3,

       Confidence3)

 =causes=>

  goal(Agent,

       atPosition(Agent, (X2, Y2), Timepoint2),

       Timepoint1, Timepoint2,

       Confidence1),

  goal(Agent,

       atPosition(Agent, (X3, Y3), Timepoint3),

       Timepoint2, Timepoint3,

       Confidence2)

 : atPosition(Agent, (X1, Y1), Timepoint1),

  atPosition(Agent, (X2, Y2), Timepoint2),

  X1 != X2, Y1 != Y2,

  collinearOrdered((X1, Y1), (X2, Y2), (X3, Y3)),

  Confidence := fn3(Confidence1,...,Timepoint3).

Note that (X3, Y3) can be the same as (X2,Y2), because an alternate way of satisfying R3 would be that the first goal corresponds to moving to reach a destination, and the second goal corresponds to stopping at the destination.

Finally, the Wayang rule format allows for an optional second trigger, Trigger$_2$. The prototypical case of a situation requiring two causal triggers is a curved trajectory, where an explanation in terms of linear forces would require one force to explain the "forward" component of movement, plus a second force to explain the "sideways" component of the same movement. Explanations involving goals instead of forces or impetus also sometimes need simultaneous causes: A bullied child might go to school while steering wide of a bully.

We shall return to these sample Wayang rules later to explain how they are used to generate candidate explanations for the initial frames of the animation in Fig. 2.

In addition to encoding Wayang's rules to make them useful for a parser (to generate abductions), we also deliberately encoded them to support potential use for generating predictions by simulating a chain of causes (via

deduction) or so that they could potentially be used by a planner. This helps avoid unintentionally tailoring the rules so that they would be applicable only for abduction, which runs the risk of overlooking important contingencies. For example, a rule meant only for abduction might neglect to include a contingency such as an agent's maximum speed (perhaps because speed is not salient in the examples used by the writer of the abductive rule to guide its formulation.) But if one adopts a discipline of always asking during rule implementation, "What might I be limited by, if I were to try enacting this goal or leveraging this physical cause?" one is more likely to avoid such oversights. Geib and Goldman (2009) adopted the same discipline for a similar reason.

*The parser algorithm* An observer can see effects, but must infer their causes. Similarly, the parser takes in observed effects and, using the knowledge base, infers (i.e., abduces) their causes. The algorithm is identical to that of Gazdar and Mellish (1989, pp. 200–201), except for these changes:

1. It does not require all tokens to be available at the outset.
2. It permits an optional second item on the left-hand side of rules.
3. It permits a rule to have a list of contingencies, all of which must be satisfied (or "delayed," if a contingency depends on a later, to-be-matched effect) for any matching attempt to succeed.
4. It allows confidence scores computed by rules to be propagated to other rules.
5. Because figures in a frame description might be ordered differently than they are in a relevant rule's conditions, and because multiple subsets of figures might match, the matcher tries different permutations *for any effect represented as a list* as needed.
6. There are cosmetic changes in the contents of output parse trees (referred to as Matched lists).

In rules that have multiple effects, there will be times when only some of the initial effects will have been matched to observations. The latter, unmatched effects represent predictions about upcoming inputs. Notice that in this case, in which some effects have not yet been matched, some contingencies may have unbound variables in their arguments. Ideally, such contingencies should be considered satisfied for the moment but should be re-evaluated if later effects are ever matched and thus provide bindings for all arguments. We were able to implement this ideal by using the constraint-logic programming language, ECLiPSe, mentioned above. It allows predicates to be declared "delayable" until a list of variables all have bindings. The delayable-predicates feature also allows us to implement arbitrarily complex contingencies as needed,

as described before. A contingency may be delayed up to the point where all its effects have candidate matches, at which point all variables have been bound, so all contingencies can be evaluated and a decision made as to whether all the effect matches succeed.

The following pseudocode describes Wayang's parser.

> Initialize the set of chart "edges" (i.e., partially and completely matched rules) to [ ] (i.e., empty list).
>
> Initialize the *CurrentSpanEnd* (i.e., the position after the most recent token, equivalent to a count of tokens seen so far) assertion to zero.
>
> For each new input token (i.e., frame(FrameItems)) do
>
> > Triggers := [FrameItems];
> >
> > Confidence := 1.0;
> >
> > CurrentSpanEnd1 := CurrentSpanEnd;
> >
> > NextSpanEnd := ++ CurrentSpanEnd;
> >
> > MatchedEffectsItems := [FrameItems];
> >
> > UnmatchedEffectsItems := [ ];
> >
> > **addEdge**(CurrentSpanEnd1, NextSpanEnd,
> >
> > > Triggers, Confidence,
> > >
> > > UnmatchedEffectsItems, MatchedEffectsItems).

> **addEdge**(SpanEnd0, SpanEnd1, %The definition of
>
> > addEdge  Triggers, Confidence,
> >
> > UnmatchedEffectsItems, MatchedEffectsItems) :-

If there is already a matching edge[3] (ignoring confidence scores) for the given arguments, then do nothing

else if UnmatchedEffectsItems is empty, then

> Add an edge using the given arguments;
>
> For each way of matching any Rule whose leftmost Effects item matches something in Triggers (*allowing for within-effect permutations*) do
>
> > Instantiate Rule using Triggers;
> >
> > Triggers2 := extractTriggers(Rule);
> >
> > Confidence2 := extractConf(Rule);
> >
> > Unmatched2 := extractUnmatched(Rule);
> >
> > addEdge(SpanEnd0, SpanEnd0, Triggers2,
> >
> > > Confidence2, Unmatched2, [])

---

[3] An "edge" means a stored 6-tuple of <SpanEnd0, SpanEnd1, Triggers, Confidence, UnmatchedEffectsItems, MatchedEffectsItems>.

For each Edge covering some earlier SpanEnd00 to SpanEnd0 whose leftmost unmatched Effects item matches something in Triggers (*allowing for within-effect permutations*) do

> Instantiate Edge using Triggers;
>
> Triggers3 := extractTriggers(Edge);
>
> Confidence3 := Confidence*extractConf(Edge);
>
> Unmatched3 := tailOf(extractUnmatched(Edge));
>
> Append extractTriggers(Edge)
>
> > and extractMatched(Edge) to MatchedEffectsItems,
> >
> > and assign to Matched3;
>
> addEdge(SpanEnd00, SpanEnd1, Triggers3,
>
> > Confidence3, Unmatched3, Matched3);

else

> Add an edge using the given arguments;
>
> For each Edge starting from SpanEnd1 (and reaching to some larger SpanEnd2) that has no unmatched Effects items but whose Triggers match the leftmost entry in UnmatchedEffectsItems (*allowing for within-effect permutations*) do
>
> > Instantiate Edge using Triggers;
> >
> > Confidence4 := Confidence; %//no change
> >
> > Unmatched4 := tailOf(UnmatchedEffectsItems);
> >
> > Matched4 := extractMatched(Edge);
> >
> > addEdge(SpanEnd0, SpanEnd2, Triggers,
> >
> > > Confidence4, Unmatched4, Matched4).

Our overall design goal for the system is that, after processing each input frame, the edge(s) with highest confidence score(s) be the same as the preferred explanation(s) that human observers, on average, would offer if the animation were stopped at that point and they were asked what they saw. In this way, the algorithm and knowledge base constitute a cognitive model of how explanations (specifically, those that invoke intentions or physical causes) are constructed and ranked as evidence unfolds.

As brief examples of how Wayang's rules would be used to generate explanations, consider the control and target "intention" animations used in our pilot. The next section provides a walk-through of the (simpler) control animation,

and the section following it provides a walk-through of the target animation.

*Sample walk-through: Control animation* The events in the control animation might be summarized as:
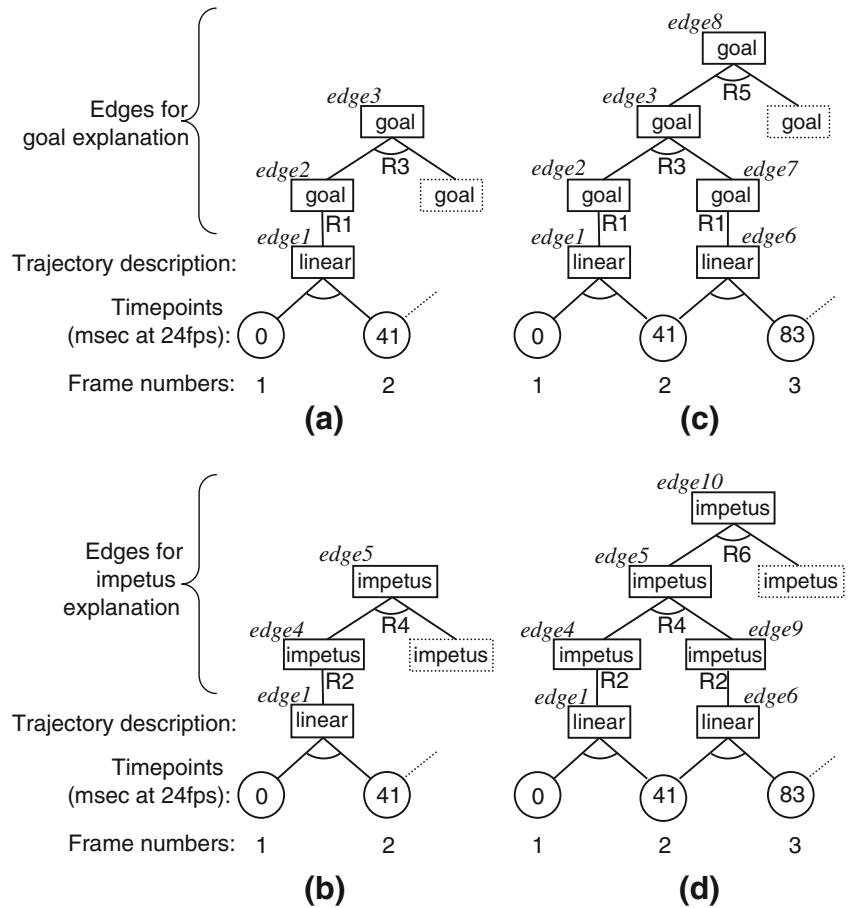
1. X and V are near the southwest corner, Y is near the center, and Z is near the northeast corner (Frame 1)
2. X moves northeast at constant speed (Frames 2–164)
3. X is in contact with Z (Frame 164)

Assume that there are grammar rules, not shown here, that compare adjacent animation frames and generate descriptions of stationary, linear, and curved trajectories. After the second frame, there would be a description of object X moving linearly up and to the right (as well as descriptions of all other objects remaining stationary). This observation of a linear trajectory matches the leftmost (and only) effect in both the goal-based and impetus-based rules above (i.e., R1 and R2, respectively). Bottom-up incremental parsers, such as Wayang's, take

an input token and search for grammar rules whose leftmost unmatched component (on the right-hand side of the rule) matches the input token. If the parser supports feature grammars, as Wayang's does, then after such a match, the parser tries to evaluate all the contingencies of the candidate rule. In this example, the contingencies of the goal-based rule are trivially satisfied by the properties of the trajectory itself (i.e., its starting position and time are different from its ending position and time), and by whether the speed of the observed movement is within the known abilities of the agent (perhaps using categorical knowledge of agents), and by the absence of any potentially colliding object. The contingencies of the impetus-based rule are also satisfied, but only because there is an object, Z, in a position that makes it a plausible attractor. So, after the second frame of the animation, there are two candidate explanations.

Actually, from the parser's point of view, the candidate explanations are each an edge added in different iterations of the processing loop, and after adding any edge the parser

**Fig. 4** Example of a parse of the first two (**a** and **b**) and three (**c** and **d**) frames from the control animation, showing edges for inferred explanations

tries to expand on it (within the same loop iteration) by calling `addEdge`. When `addEdge` is called on the edge created from the goal-based rule R1 (i.e., edge2 in Fig. 4a), it finds that the goal matches the leftmost effect of rule R3. The only contingency of R3 that can be evaluated at this point, because all of its variables can be bound, is the first one, which is trivially satisfied by matching against X's position in the first frame. All other contingencies must be delayed. Thus, the match succeeds, and a new edge (i.e., edge3 in Fig. 4a) is created showing that the first effect of R3 is satisfied (for the moment) and that the second effect, also a goal, is predicted.

Similar to R3 is a rule R4 (not shown) that says that two aligned linear trajectories, each explained by an impetus to move in the same direction with similar (but perhaps decaying) magnitudes, can be joined into a single larger span using the impetus as the common explanation. And, similar to the way that the parser expands on the R1-based edge by creating an edge based on a partially satisfied R3 (i.e., edge3 in Fig. 4a), the parser expands on its R2-based edge (i.e., edge4 in Fig. 4b) by creating an edge based on a partially satisfied R4 (i.e., edge5 in Fig. 4b). The not-yet-satisfied part of the R4-based edge represents a prediction that X will continue to move in a way that suggests it possesses a specific impetus.

When the third frame arrives, the same flow of inference using R1 and R2 repeats, resulting in two explanations, both spanning the time points represented by Frames 2 and 3. These explanations correspond to edge7 (Fig. 4c) and edge9 (Fig. 4d). These edges satisfy the unmatched second effects in edge3 and edge5, respectively. Fulfilling those edges leads to more calls to `addEdge`, which results in the partially satisfied edge8 (Fig. 4c) and edge10 (Fig. 4d).

Notice how the recursive rules R3 and R4 allow the system to accumulate arbitrarily long sequences of *consistent* observations into competing explanations. It is technically possible to achieve the same output using just the nonrecursive R1 and R2, but with recursive trajectory rules that generate a representation of a longer trajectory for each new observation. But using recursion at the level of goal and impetus concepts permits connecting *inconsistent* trajectories, such as an agent moving to a target and then remaining stationary there. Furthermore, it permits the confidence value associated with each goal or impetus explanation to be based at least in part on the confidence value of any goal or impetus explanation that fed into it. We believe that the confidence that people invest in their explanations at a late stage often depends on the confidence they adopted in earlier stages. Therefore, Wayang's rules are designed to be recursive at the level of explanatory concepts that can carry confidence values.

Wayang repeats the constructive steps described above for the first three frames of the control animation to as many following frames as it can, ultimately reaching Frame 164 (i.e., the last frame in the events listed above). In doing so, it builds one goal-based explanation and one impetus-based explanation that each cover that entire span.

How do the confidences of the two longest-spanning explanations so far compare? We are planning to do studies that will determine what events people identify in our animations, where the event boundaries are, what explanation(s) are given for each event (if any), and what the typical confidence is in each explanation. But in the meantime, we are relying on introspection and group consensus, which tells us that they are both highly likely—say, .8 for the goal-based explanation on a [0.0 … 1.0] real-valued scale of confidence, and .7 for the impetus-based one. One reason for these confidence levels to be similar is that the two candidate causes seem likely to occur frequently and at similar rates, at least in this simplified animated world (i.e., the base rates seem the same). In general, we imagine that confidence functions will tend to asymptote toward values higher than their initial value, assuming that no cues appear that would push the confidence higher or lower. In this case, a reasonable confidence function might start at .6 and asymptote toward .9 as the number of consistent frames approaches infinity. The reason the impetus-based explanation has lower confidence is that X does not move directly toward the center of Z. This variance is within the margin permitted by the `collinearOrdered` constraint, so the rule is applicable, but the confidence function nevertheless lowers the confidence due to the doubt such a cue induces.

Because X comes into contact with Z in Frame 164, rules R1 and R2 fail to activate, because they both have contingencies requiring no contact. Instead, a different set of rules (not shown) having contingencies that *require* contact become activated. One subset of these rules is impetus-based and explains a sequence of events in which the magnitude of the impetus is great enough that X bounces off Z (in ever smaller bounces as the magnitude decreases). Another subset explains a sequence of events in which the magnitude is small enough that X stops once it is in contact with Z. Depending on the magnitude abduced using rule R2 during Frames 2–164, Wayang will activate one subset of rules or the other, and the unmatched effects of the rules represent predictions of what would happen in later frames if the animation did not end at Frame 164.

*Sample walk-through: Target animation* Events in the target animation (see Fig. 2 again) might be summarized as:

1.  X and V are near the southwest corner, Y is near the center, and Z is near the northeast corner (Frames 1–24)
2.  X accelerates northeast with a very slight side-to-side motion (Frames 25–67)
3.  When nearing Y, X decelerates northeast with a very slight side-to-side motion (Frames 68–78)
4.  X is stationary while a moderate distance from Y (Frames 79–87)
5.  X follows a curved path north then northeast at constant speed (Frames 88–139)
6.  X is stationary and in contact with Z (Frames 140–164)

Notice that the initial placement of figures is the same in the two animations, that only X moves in both, and that the number of frames is the same.

To explain why X remains stationary in Frames 2–24 in this animation (and why V, Y, and Z remain stationary in both animations), rules similar to R1 and R2, but much simpler, can be used. For example, rule R5 below says that "If `Agent` has a goal to be at `(X1, Y1)` at future `Timepoint2`, and this goal persists between the current `Timepoint1` and `Timepoint2`, then if the contingencies are met, the `Agent` will remain stationary at least until the desired time." The contingencies confirm that the `Agent` is actually already at the desired position and that the agent knows of nothing that might collide with it during that time span (assuming omniscience, in this case).

```
R5: goal(Agent,

        atPosition(Agent, (X1, Y1), Timepoint2),

        Timepoint1, Timepoint2,

        Confidence)

 =causes=>

  stationaryTrajectory(Agent, (X1, Y1), Timepoint1, Timepoint2)

: atPosition(Agent, (X1, Y1), Timepoint1),

 ThisTrajectory :=

   stationaryTrajectory(Agent, (X1, Y1), Timepoint1, Timepoint2),

  findAll(OtherActiveTrajectory,

    (OtherActiveTrajectory != ThisTrajectory,

    onInterceptCourse(ThisTrajectory,

                        OtherActiveTrajectory)),

    []), %No other figures on intercept course

  Confidence := fn5(Agent,...,Timepoint2).
```

When used together with recursive goal-based rule R3, rule R5 can be used to explain arbitrarily long sequences of remaining stationary as fulfilling a goal to be in the agent's current position.

An impetus-based rule to explain remaining stationary would be similar to rule R2, but again, simpler. It could be used together with recursive impetus-based rule R4 to explain arbitrarily long sequences of remaining stationary as an object that is primarily under the influence of an inertia-like impetus to remain in place. The goal- and impetus-based explanations to remain stationary seem inherently equally likely, and there are no cues (yet) to suggest favoring one over the other, so the confidence functions of these rules would compute similar confidence values for the explanations covering each subspan.

Our walk-through of the target animation example now reaches a moment of decision, the change in X from remaining stationary to accelerating northeast, starting in Frame 25. Wayang has no rules for explaining a change from a period of remaining stationary to a period of moving *in terms of physical causes*, because it requires us to make an appreciable effort to deliberate and envision such explanations. If we did add such rules, they might require hypothesizing unseen actions, such as tilting the table and thus changing the angle of gravity (assuming that the action is imagined to take place on a tabletop) or that Z is an electromagnet that has just been switched on, and so forth, and all such rules would be given corresponding low initial confidence scores with slow-growing functions. Goal-based explanations for the change come to mind more easily, albeit with low initial confidence. Specifically, everyday agents often change their goals, and although such an explanation is more compelling if one has an idea of what motivated the goal change, it does not seem necessary to have a specific cause in mind. For example, in future work in which we allow figures to be more visually complex, including having eyes that indicate gaze direction, if the eyes point toward an object for the first time just before the agent moves toward that object, the specific cause might be taken to be that the agent was not previously aware that the object was in its position and wants to be near it.

Rule R6 below implements the concept of "goal change for no specific reason" outlined above. It provides no constraints on the kinds of goals that an agent might change to, and thus is not helpful for making predictions about such goals. Its role is to connect whatever goal-based explanation emerges from later evidence (if any) with the just-completed goal-based explanation. The confidence value of such a nonspecific goal-change explanation would

largely depend on the lower of the two confidence values of the explanations it connects.

**R6:** goalChanged(Agent,

goalState1, Timepoint2, goalState2,

Timepoint1, Timepoint3,

Confidence3)

=causes=>

goal(Agent, goalState1,

Timepoint1, Timepoint2,

Confidence1),

goal(Agent, goalState2,

Timepoint2, Timepoint3,

Confidence2)

: goalState1 != goalState2,

Confidence := fn6(Confidence1,...,Timepoint3).

We are still at Frame 25, but the task is now to explain all of the remaining frames. The frames up to 79 show a sequence of linear trajectories that alternately aim above and below Z, forming a gradual zigzag path. In the first portion of the path, X is accelerating, and in the latter portion, decelerating. As long as the angle points of the zigzag are within the margin of variance for collinearity, the initial portion of constant acceleration can be explained using rules R1 and R3, as can the latter portion of constant deceleration. Furthermore, the zigzag motion is suggestive of walking, and thus provides a cue that should boost the confidence level of this explanation in terms of an agent wanting to get to a location. Thus, Wayang would have higher confidence in this goal explanation than for a purely straight path of same length (such as appears in subsequences of the control animation). Finally, the acceleration portion and the deceleration portion can also be joined using rule R3, and since this change in acceleration also provides a cue for agency (see the "slow in and slow out" animation technique of Thomas & Johnston, 1995), it motivates an increase in confidence level over the confidence levels of its constituent explanations (i.e., the acceleration and deceleration portions).

Over the same sequence of frames, the parser also tries to apply the impetus-based rules. But the only times rule R2 is satisfied are for the linear trajectories that aim below Z where Y is a plausible attractor. There are no plausible attractors or repulsors, nor any colliding moving objects, that would plausibly explain the linear trajectories that aim above Z. Thus, there are unexplained gaps, and there is no recursive rule to bridge those gaps.

Starting in Frame 79, X stops and then remains still until Frame 87. In isolation, this sequence can be explained equally well using either the goal or moving-impetus concepts, just as Frames 1–24 were. But the impetus-based explanation cannot be connected to any similar explanation from earlier in the animation, while the goal-based explanation can be connected using rule R3 to infer that both the earlier accelerating-then-decelerating zigzag northeast and this period of remaining still are part of a goal to be at the current position.

Starting in Frame 88, we find a second instance of the relatively hard-to-explain case of an object starting to move after remaining stationary for a while. Furthermore, the object, X, moves along a kind of trajectory not seen before in this animation—a curved one. In some scenarios, a curved trajectory is explained using a single cause. For example, as part of their review of McCloskey's impetus studies, Kozhevnikov and Hegarty (2001) observed that

> many people also believe that an object constrained to move in a curved path acquires a curvilinear impetus that causes the object to follow a curved trajectory for some time after the constraints on its motion are removed. (p. 441)

In Wayang, such an explanation might be generated by a rule linking a *single* trigger, an impetus that imparts curved motion to the object possessing the impetus, to effects represented as curved trajectories. The contingencies of such a rule would require that the object be moving outside of any enclosure but that just previously it was travelling in a narrow enclosure whose curvature matches its current arc. Yet, in this case, there is no such enclosure, and instead rules suggesting *two simultaneous* triggers are available. As mentioned earlier, Wayang's rule format provides an optional second trigger, which was inspired by curved paths such as this one. For example, rule R7 below describes how two physical forces, oriented perpendicular to each other, can cause an object under the influence of both to follow a curved trajectory:

**R7:** exertForceUpon( Attractor,

attraction,

MovedObj,

Timepoint1, Timepoint3,

Confidence),

exertForceUpon(Repulsor,

repulsion,

MovedObj,

Timepoint1, Timepoint3,

Confidence)

=causes=>

curvedTrajectory( MovedObj,

secondToLastPosition(X2,Y2),

lastPosition(X3,Y3),

latestProjectedCircle(Xc,Yc,R),

(StartingSpeedX1, StartingSpeedY1),

(ConstantAccelX2, ConstantAccelY2),

Timepoint1, Timepoint3)

: atPosition(MovedObj, (X1, Y1), Timepoint1), %first of three points defining the curve

atPosition(Attractor, (X4, Y4), Timepoint1), %Attractor remains still

atPosition(Attractor, (X4, Y4), Timepoint2),

atPosition(Attractor, (X4, Y4), Timepoint3),

atPosition(Repulsor, (X5, Y5), Timepoint1), %Repulsor remains still

atPosition(Repulsor, (X5, Y5), Timepoint2),

atPosition(Repulsor, (X5, Y5), Timepoint3),

convexOrderedVerticesPolygon( %curve points toward Attractor, with Repulsor underneath

5, %/* 5 vertices; no points coincide */

[(X1,Y1),(X2,Y2),(X3,Y3),(X4,Y4),(X5,Y5)]),

```
ThisTrajectory :=

  curvedTrajectory( MovedObj,

                    secondToLastPosition(X2,Y2),

                    lastPosition(X3,Y3),

                    latestProjectedCircle(Xc,Yc,R),

                    (StartingSpeedX1, StartingSpeedY1),

                    (ConstantAccelX2, ConstantAccelY2),

                    Timepoint1, Timepoint3),

 findAll( OtherActiveTrajectory,

          (OtherActiveTrajectory != ThisTrajectory,

          onInterceptCourse(ThisTrajectory, OtherActiveTrajectory)),

          []), %No other figures on intercept course

euclideanDistance(X1,Y1,X4,Y4,Distance1_4),

euclideanDistance(X2,Y2,X4,Y4,Distance2_4),

euclideanDistance(X3,Y3,X4,Y4,Distance3_4),

euclideanDistance(X1,Y1,X5,Y5,Distance1_5),

euclideanDistance(X2,Y2,X5,Y5,Distance2_5),

euclideanDistance(X3,Y3,X5,Y5,Distance3_5),

Distance3_4 =< Distance2_4 =< Distance1_4,

Distance3_5 >= Distance2_5 >= Distance1_5,

Confidence := fn7(Attractor,...,Timepoint2).
```

The contingencies above require that there be one object positioned relative to the trajectory so that it could be an attractor, another object positioned so that it could be a repulsor, and that nothing is expected to collide with the path. Specifically, the convexOrderedVerticesPolygon contingency requires that the position of the potential attractor be "ahead" of the curved path and that the position of the potential repulsor be "under" the path. In the target animation, object Z has a position relative to X's curved trajectory that makes it a plausible attractor, and Y's position simultaneously makes it a plausible repulsor, so rule R7 can be used to explain the three frames starting at Frame 88: 88, 89, and 90. When a fourth frame arrives, R7 can again explain it and the two that preceded it: 89, 90, and 91. In this way, overlapping sequences of three frames are explained, and it would make sense to create a recursive rule (not

shown here) to collect such sub-sequences to cover entire coherent curves. For this animation, the rule could cover all frames of the curved path, Frames 89–139, under a single explanation that uses two simultaneous triggers.

Starting at Frame 140, X becomes stationary and remains that way through the end of the animation at Frame 164. This stationary trajectory can be explained using impetus-based rules in the manner already described for the stationary episode between Frames 1 and 24. Thus, over the entire animation, some of the trajectories can be explained using the impetus concept (or, similarly, by forces), yet others cannot be, because there are no objects that could serve as plausible attractors, repulsors, or colliding objects. Furthermore, there are no explanations that cover multiple trajectories in sequence; there are only piecemeal physical-cause explanations across the whole animation.

In contrast, goal-based explanations can cover the entire animation. As described so far, there is a goal-based explanation for the initial stationary period, and a second one for the zigzag movement to the northeast and its coming to rest. Connecting these two is a weaker goal-change explanation. And, for the curved section, an explanation that quickly comes to mind (at least to us) is that X wants to get to a position near Z while avoiding Y. Such a two-goal explanation is readily formulated in a two-trigger rule, R8 below, similar to R7 above.

**R8:** goal( Agent,

  atPosition(Agent, (X4, Y4), Timepoint3),

  Timepoint1, Timepoint3,

  Confidence),

  goal( Agent,

  avoid(Agent, ThreatObj, Timepoint3),

  Timepoint1, Timepoint3,

  Confidence)

=causes=>

  curvedTrajectory( Agent,

          secondToLastPosition(X2,Y2),

          lastPosition(X3,Y3),

          latestProjectedCircle(Xc,Yc,R),

          (StartingSpeedX1, StartingSpeedY1),

          (ConstantAccelX2, ConstantAccelY2),

          Timepoint1, Timepoint3)

  : atPosition(Agent, (X1, Y1), Timepoint1), %first of three points defining the curve

  atPosition(ThreatObj, (X5, Y5), Timepoint1), % ThreatObj remains still

  atPosition(ThreatObj, (X5, Y5), Timepoint2),

  atPosition(ThreatObj, (X5, Y5), Timepoint3),

  convexOrderedVerticesPolygon( %curve points toward (X4,Y4), with ThreatObj underneath

    5, %/* 5 vertices; no points coincide */

    [(X1,Y1),(X2,Y2),(X3,Y3),(X4,Y4),(X5,Y5)]),

```
ThisTrajectory :=
  curvedTrajectory( Agent,

                    secondToLastPosition(X2,Y2),

                    lastPosition(X3,Y3),

                    latestProjectedCircle(Xc,Yc,R),

                    (StartingSpeedX1, StartingSpeedY1),

                    (ConstantAccelX2, ConstantAccelY2),

                    Timepoint1, Timepoint3),

  findAll( OtherActiveTrajectory,

           (OtherActiveTrajectory != ThisTrajectory,

           onInterceptCourse(ThisTrajectory, OtherActiveTrajectory)),

           []), %No other figures on intercept course

  euclideanDistance(X1,Y1,X4,Y4,Distance1_4),

  euclideanDistance(X2,Y2,X4,Y4,Distance2_4),

  euclideanDistance(X3,Y3,X4,Y4,Distance3_4),

  euclideanDistance(X1,Y1,X5,Y5,Distance1_5),

  euclideanDistance(X2,Y2,X5,Y5,Distance2_5),

  euclideanDistance(X3,Y3,X5,Y5,Distance3_5),

  Distance3_4 =< Distance2_4 =< Distance1_4,

  Distance3_5 >= Distance2_5 >= Distance1_5,

  Confidence := fn8(Agent,...,Timepoint2).
```

The parser can apply rule R8 to successive, overlapping sequences of frames for as long as the movement follows a consistent curve at constant acceleration. And these mini-explanations can be collected into larger and larger spans by a recursive rule that is tailored to two-goal explanations (not shown). Starting at Frame 140, the curved movement ends, and X remains stationary until Frame 164, when the animation itself ends. We have already described how arbitrarily long stationary periods can be explained in terms of goals, and how goal-directed movement followed by goal-directed remaining still can be given an overall goal-based explanation that the agent wanted to be in the final position *all along*. For this curving-and-then-stopped portion to be connected to the preceding zigzagging-and-then-stopped portion, the best option discussed so far is a weak goal-change explanation. Explaining the entire animation in goal terms requires two such goal changes, because there are two times when remaining still is followed by movement: once when X's initial stillness is followed by the zigzag northeast, and a second when the stop after the zigzag is followed by the curved path. But, in retrospect, after watching the entire animation, one might infer that the first pause might be due to X not noticing at first that Z is present, or that Z is desirable, and the second pause might be due to X not noticing that Y lies on its path to Z until very near Y, or that Y is undesirable, and having to momentarily reassess options. These explanations relying on assumptions that an agent did not notice something right away can be formulated as specializations of the goal-change rule described earlier—they provide specific reasons for the initial goal to change.

It will be interesting to see in our planned event segmentation studies whether participants mark event boundaries at these pause points and whether they give explanations that strongly or weakly connect the events on either side. If participants do provide strong connecting explanations, it would motivate adding specializations of the goal-change rule as just described.

Two other explanations were listed earlier, "X intends to be farther from V" and "X intends to be closer to Y." To generate these, the system uses a goal-based rule about avoidance, not shown here, and rule R1, respectively. The avoidance rule's confidence function has a lower initial value than R1, because we believe people are biased toward approach explanations over avoidance ones, so "X intends to be farther from V" always has a lower confidence score than "X intends to be closer to Z." The interpretation "X intends to be closer to Y" fares as well as "X intends to be closer to Z" until the curved movement begins, at which point there is no matchable rule to carry this interpretation further.

## Composing more inclusive explanations

As we have just discussed, people sometimes connect actions and intentions into larger coherent narratives. How should the system connect its explanations in order to construct more overarching ones? In Wayang, the preferred solution is to use recursive grammar-like rules that accumulate mini-explanations that cover a few frames into explanations that cover arbitrarily long sequences. Some of the recursive rules are tailored to apparently consistent behavior, such as spans of remaining stationary or moving linearly or along a curve. Other recursive rules are tailored to join apparently inconsistent behavior, such as moving and then coming to a stop, into consistent patterns that a typical person might perceive. In Wayang, there are more goal-based rules for apparently inconsistent behavior than ones based on physical causes, because the physical forces modeled by the system either require contact with another object or exert uniform influence throughout the space (i.e., attraction and repulsion), and thus the rules must impose narrower constraints than goal-based ones do.

Despite these differences in Wayang's goal- and impetus-based rules, it may not be obvious that the two kinds of rules can make dramatically different predictions about a single object, yet they do. One reason this may not be obvious is that in the sample animations, only one object, X, moves. Imagine X and Z in similar starting positions, but in one new animation Z is an object that moves northwest and that attracts object X. In this case, X *mindlessly follows* Z and will always be "behind" it. Then imagine X is an agent interested in object Z. In this case, X might anticipate Z's heading and attempt to *head it off* to catch up with it. For many starting configurations of object placements, paths that emerge from "mindlessly following" versus "heading off" are easily distinguished, and "heading off" in particular provides a high-confidence cue that X is an agent. Finally, imagine an animation in which both X *and* Z are agents, and as before X is interested in Z, but in

this case Z wants to avoid X. As before, X might try to anticipate Z's direction and head it off, but Z will move to counter that, which X should notice, and now X must take Z's likely plans into account in order to catch up with it. This scenario is arguably the simplest scenario that suggests one observable agent is applying *theory of mind* to another observable agent, yet there are many different ways that X and Z might move in this case. Identifying a set of paired moves of X and Z that is representative of this variety, and designing a representation that captures their commonality as theory-of-mind, is a current knowledge-engineering challenge for us.

Formulating rules to support the Wayang approach is a difficult knowledge-engineering task. As the work reaches higher-level, more inclusive, narrative-like explanations in richer environments (such as animations of articulated figures), we hope to be able to leverage existing knowledge bases, including representations of actions in the parameterized action representation (Badler, Allbeck, Zhao, & Byun, 2002) and representations of action verbs as found in linguistic semantics (FrameNet Project, 2009; Goddard & Wierzbicka, 2009).

## Conclusion

The system we have described is currently under development, and inevitably refinements and adjustments will be made as we progress. But we have described how the design of Wayang meets the objectives listed earlier:

1. The approach handles goal- and physical-cause-based explanations equally well, and holds some promise that it will be expressive enough for theory-of-mind-based explanations as well.
2. Wayang's use of a bottom-up incremental parser allows it to generate and manage multiple alternate explanations as the action unfolds.
3. Wayang's use of the concept of impetus allows it to model the explanations of nonexperts in physics.
4. The use of multiple objects in our sample animations provides a rich environment, and thus more opportunity to evoke rich social explanations in our participants that we, in turn, can model.
5. Wayang's use of feature-grammar-like rules and (embedded) confidence functions that do not require optimizing or norming across competing explanations permits a way of doing knowledge engineering that does not require updating knowledge that has previously worked when adding new types of actions or explanations.
6. The concept of a confidence function has been designed to have no a priori interpretation (e.g., not

as utility), but instead merely to summarize the combined influences of psychological cues so that alternate explanations might be ranked.

Our immediate goals are as follows. We will continue formulating rules, and creating test animations to drive the rule refinement process.

The literature has suggested a number of factors that influence observers' perceptions of animacy and intentions (see Table 1). We are working to identify computable functions for each, as well as plausible ways of combining such computations when multiple cues are present. Note that some of the listed influences, such as temporary social isolation of the observer, will be relevant only if we implement simulated inputs for such observer states.

As mentioned earlier, we are planning to do studies that will determine what events people identify in our animations, where the event boundaries are, what explanation(s) are given for each event (if any), and what the typical confidence is in each explanation. The results will guide revisions to Wayang's rules and confidence functions. To study the interaction of multiple cues, the studies will use animations that have only single cues as well as animations with combinations of cues, to indicate the relative contribution of each cue.

Although Wayang generates explanations after each input, there is some evidence that people sometimes construct explanations only when their predictions fail

(Leake, 1995; Zacks & Swallow, 2007). We plan to do a deeper literature review on this question, and perhaps to alter Wayang accordingly.

During our modeling effort, we have had a working assumption that the value of an explanation's confidence score should depend solely on the positive evidence gathered in support of the explanation. There is no discounting due to negative evidence, nor due to stronger competing explanations. We plan a review of the psychological literature on inference making to determine whether or not our working assumption is supported.

## Appendix: Design considerations for animations

In order to reduce the types of cues our model had to detect and interpret, our animations included only four colored circles of the same size, and only one circle moved. This enabled the initial instantiation of the model to rely exclusively on translocation-type movement cues in relation to a background context. We included a letter on each circle to help participants refer to them. Thus, in contrast to animations that use multiple shapes, such as rectangles and triangles (e.g., Heider & Simmel, 1944; Martin & Weisberg, 2003; Pavlova, Guerreschi, Lutzenberger, & Krägeloh-Mann, 2010; R. T. Schultz et al., 2003; Wheatley, Milleville, & Martin, 2007), all shapes in our animations were circles of the same size but varied in color (J. Schultz, Friston, O'Doherty, Wolpert, & Frith, 2005; J. Schultz, Imamizu, Kawato, & Frith, 2004). Notably, we did not use shapes that indicate affordances, such as rectangles or lines that represent houses or barriers (e.g., Baker, Saxe, & Tenenbaum, 2009; Castelli et al., 2000; Heider & Simmel, 1944; Tavares, Lawrence, & Barnard, 2008), or background contexts that were cartoon characterizations of real-world objects (Wheatley et al., 2007). By using circles, which do not have a line of symmetry, we could use a simpler model that did not have to determine or track the direction the shapes were facing, a cue that people use to detect intentions (e.g., Blythe et al., 1999; Gao, Newman, & Scholl, 2009). We also tried to make the animation imply a top-down view (e.g., Heider & Simmel, 1944) to remove the issue of depth and thereby to simplify visual interpretation, in contrast to other animations that have suggested a side view (Gergely, Nádasdy, Csibra, & Bíró, 1995; Martin & Weisberg, 2003; Wheatley et al., 2007).

Within this simple circle world, we envisioned our focal shape, X, as an agent with two concurrent intentions: going to Circle Z while avoiding Circle Y. We targeted simulta-

**Table 1** Psychological cues and influences on the perception of agency and specific intentions

| Type of Influence | Sample Citation |
|---|---|
| Bottom-up (i.e., information in the stimulus) | |
| Motion cues: e.g., relative velocity | Blythe et al., 1999 |
| Orientation vs. direction of motion | Scholl & Tremoulet, 2000 |
| Speed relative to background | Morewedge et al., 2007 |
| Spatial context: e.g., obstacles and openings | Baker, Goodman, & Tenenbaum, 2008 |
| Animation techniques aimed at providing "an illusion of life" | Thomas & Johnston, 1995 |
| Top-down (i.e., schema-related preinformation) | |
| Preinformation about traits of present agents | Shor, 1957 |
| Preinformation about an agent's abilities, beliefs, and goals | Malle, 2006 |
| Prejudices for/against the agent's social group | Bodenhausen & Wyer, 1985 |
| Other | |
| Repeated exposure to an animation increases agentic explanation | Martin & Tversky, 2003 |
| Temporary social isolation reduces anthropomorphism | Waytz et al., 2010 |
| Social confidence reduces anthropomorphism | Waytz et al., 2010 |

neous, overlapping causes because they seem common in the real world and we wanted to avoid oversimplifying the problem (at least in the range of candidate explanations, although not in the richness of the stimulus, of course). We included multiple cues in our animation to suggest that X was an animate agent with these dual intentions. As the animation began, X was stationary for almost a second before starting to move. Based on the finding that a delay before linear movement is sufficient to elicit descriptions suggestive of animacy in some observers (Gelman, Durgin, & Kaufman, 1995), we thought that this would enhance the perception of animacy. Another animacy-enhancing cue in the animation was based on the finding that a moving object with an observable goal has been found to be perceived as more animate (Opfer, 2002). When Object X initially moved, it did so towards Z, and continued to do so even while moving in an arc around Y, suggesting that X's goal was to get to Z. We also found inspiration from two principles provided by animators of Walt Disney feature films (Thomas & Johnston, 1995). First, we applied the *easy-in, easy-out* principle, which suggests that self-propelled moving objects start moving slowly and need to accelerate to their top speed, and likewise that they slow down before stopping. In our intention animation, X sped up when starting from a stop and slowed down before stopping. Second, we applied the principle of *secondary action*, in which a main action is augmented by a secondary action that supports the primary action. In our animation, X wiggled back and forth as it moved, with the wiggle being a secondary action (such as rocking back and forth while walking) for the primary action of moving forward. Additionally, changes in movement direction and acceleration without any visible outside cause has been found to be related to perceptions of animacy (Tremoulet & Feldman, 2000, 2006) and specific intentions (Blythe et al., 1999).

# References

Abell, F., Happé, F., & Frith, U. (2000). Do triangles play tricks? Attribution of mental states to animated shapes in normal and abnormal development. *Cognitive Development, 15,* 1–16.

Andersen, S. M., & Klatzky, R. L. (1987). Traits and social stereotypes: Levels of categorization in person perception. *Journal of Personality and Social Psychology, 53,* 235–246.

Anderson, J., & Anderson, B. (1993). The myth of persistence of vision revisited. *Journal of Film and Video, 45,* 3–12.

Badler, N., Allbeck, J., Zhao, L., & Byun, M. (2002). Representing and parameterizing agent behaviors. In *Proceedings of Computer Animation 2002, IEEE Computer Society, Geneva, Switzerland* (pp. 133–143). New York: IEEE Press.

Baker, C. L., Goodman, N. D., & Tenenbaum, J. B. (2008). Theory-based social goal inference. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society* (pp. 1447–1452). Austin, TX: Cognitive Science Society.

Baker, C. L., Saxe, R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition, 113,* 329–349. doi:10.1016/j.cognition.2009.07.005

Barrett, H. C., Todd, P. M., Miller, G. M., & Blythe, P. W. (2005). Accurate judgments of intention from motion cues alone: A cross-cultural study. *Evolution and Human Behavior, 26,* 313–331.

Blythe, P. W., Todd, P. M., & Miller, G. F. (1999). How motion reveals intention: Categorizing social interactions. In G. Gigerenzer, P. M. Todd, & the ABC Research Group (Eds.), *Simple heuristics that make us smart* (pp. 257–286). New York: Oxford University Press.

Bodenhausen, G. V., & Wyer, R. S. (1985). Effects of stereotypes on decision making and information-processing strategies. *Journal of Personality and Social Psychology, 48,* 267–282.

Castelli, F., Happé, F., Frith, U., & Frith, C. (2000). Movement and mind: A functional imaging study of perception and interpretation of complex intentional movement patterns. *NeuroImage, 12,* 314–325.

Crick, C., Doniec, M., & Scassellati, B. (2007). Who is IT? Inferring role and intent from agent motion. In *Proceedings of the 6th IEEE International Conference on Development and Learning (ICDL 2007)* (pp. 134–139). Piscataway, NJ: IEEE Press.

Crick, C., & Scassellati, B. (2008). Inferring narrative and intention from playground games. In *Proceedings of the 7th IEEE International Conference on Development and Learning (ICDL 2008)* (pp. 13–18). Piscataway, NJ: IEEE Press.

Feldman, J. (2007). The formation of visual "objects" in the early computation of spatial relations. *Perception & Psychophysics, 69,* 816–827.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2008). CogSketch: Open-domain sketch understanding for cognitive science research and for education. In: *Proceedings of the Fifth Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Annecy, France.

Gao, T., Newman, G. E., & Scholl, B. J. (2009). The psychophysics of chasing: A case study in the perception of animacy. *Cognitive Psychology, 59,* 154–179.

Gazdar, G., & Mellish, C. S. (1989). *Natural language processing in Prolog: An introduction to computational linguistics.* Wokingham, England: Addison-Wesley.

Geib, C. W., & Goldman, R. P. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence, 173,* 1101–1132.

Gelman, R., Durgin, F., & Kaufman, L. (1995). Distinguishing between animates and inanimates: Not by motion alone. In D. Sperber, D. Premack, & A. J. Premack (Eds.), *Causal cognition: A multidisciplinary debate* (pp. 150–184). Oxford: Oxford University Press, Clarendon Press.

Gergely, G., Nádasdy, Z., Csibra, G., & Bíró, S. (1995). Taking the intentional stance at 12 months of age. *Cognition, 56,* 165–193.

Goddard, C., & Wierzbicka, A. (2009). Contrastive semantics of physical activity verbs: "Cutting" and "chopping" in English, Polish, and Japanese. *Language Sciences, 31,* 60–96. doi:10.1016/j.langsci.2007.10.002

Goodman, N., Baker, C., & Tenenbaum, J. (2009). Cause and intent: Social reasoning in causal learning. In N. Taatgen & H. van Rijn (Eds.), *Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society* (pp. 2759–2764). Austin, TX: Cognitive Science Society.

Hamlin, J. K., Wynn, K., & Bloom, P. (2007). Social evaluation by preverbal infants. *Nature, 450,* 557–559.

Hauser, M. D. (2006). *Moral minds: How nature designed our universal sense of right and wrong.* New York: HarperCollins.

Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology, 57,* 243–249.

Kerr, W., & Cohen, P. (2010). Recognizing behaviors and the internal state of the participants. In *IEEE 9th International Conference of Development and Learning (ICDL)* (pp. 33–38). Piscataway, NJ: IEEE Press.

Kozhevnikov, M., & Hegarty, M. (2001). Impetus beliefs as default heuristics: Dissociation between explicit and implicit knowledge about motion. *Psychonomic Bulletin & Review, 8,* 439–453.

Leake, D. B. (1995). Abduction, experience, and goals: A model of everyday abductive explanation. *Journal of Experimental and Theoretical Artificial Intelligence, 7,* 407–428.

Luo, Y. (2010). Three-month-old infants attribute goals to a non-human agent. *Developmental Science, 2,* 453–460.

Malle, B. (2006). *How the mind explains behavior: Folk explanations, meaning, and social interaction.* Cambridge, MA: MIT Press.

Maner, J. K., Kenrick, D. T., Becker, D. V., Robertson, T. E., Hofer, B., Neuberg, S. L., et al. (2005). Functional projection: How fundamental social motives can bias interpersonal perception. *Journal of Personality and Social Psychology, 88,* 63–78.

Mann, R., Jepson, A. D., & El-Marghi, M. (2002). Trajectory segmentation using dynamic programming. In *Proceedings of the 16th International Conference on Pattern Recognition, August 2002, Quebec City, Canada* (pp. 331–334). Piscataway, NJ: IEEE Press.

Martin, B. A., & Tversky, B. (2003). Segmenting ambiguous events. In R. Alterman & D. Kirsh (Eds.), *Proceeding of the 25th Annual Meeting of the Cognitive Science Society* (pp. 781–786). Mahwah, NJ: Erlbaum.

Martin, A., & Weisberg, J. (2003). Neural foundations for understanding social and mechanical concepts. *Cognitive Neuropsychology, 20,* 575–587.

McCloskey, M. (1983). Naïve theories of motion. In D. Gentner & A. Stevens (Eds.), *Mental models* (pp. 299–324). Hillsdale, NJ: Erlbaum.

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought.* Chicago: University of Chicago Press.

Morewedge, C. K., Preston, J., & Wegner, D. M. (2007). Timescale bias in the attribution of mind. *Journal of Personality and Social Psychology, 93,* 1–11.

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology, 28,* 28–38.

Opfer, J. E. (2002). Identifying living and sentient kinds from dynamic information: The case of goal-directed versus aimless autonomous movement in conceptual change. *Cognition, 86,* 97–122.

Pavlova, M., Guerreschi, M., Lutzenberger, W., & Krägeloh-Mann, I. (2010). Social interaction revealed by motion: Dynamics of neuromagnetic gamma activity. *Cerebral Cortex, 20,* 2361–2367.

Reynolds, J. R., Zacks, J. M., & Braver, T. S. (2007). A computational model of event segmentation from perceptual prediction. *Cognitive Science, 31,* 613–643.

Scholl, B. J., & Tremoulet, P. (2000). Perceptual causality and animacy. *Trends in Cognitive Sciences, 4,* 299–309.

Schultz, J., Friston, K. J., O'Doherty, J., Wolpert, D. M., & Frith, C. D. (2005). Activation in posterior superior temporal sulcus parallels parameter inducing the percept of animacy. *Neuron, 45,* 625–635.

Schultz, R. T., Grelotti, D. J., Klin, A., Kleinman, J., Van der Gaag, C., Marois, R., et al. (2003). The role of the fusiform face area in social cognition: Implications for the pathobiology of autism. *Philosophical Transactions of the Royal Society B, 358,* 415–427.

Schultz, J., Imamizu, H., Kawato, M., & Frith, C. D. (2004). Activation of the human superior temporal gyrus during observation of goal attribution by intentional objects. *Journal of Cognitive Neuroscience, 16,* 1695–1705.

Schwitter, R. (2003). Incremental chart parsing with predictive hints. In *Proceedings of the Australasian Language Technology Workshop* (pp. 1–8).

Shanahan, M. (2005). Perception as abduction: Turning sensor data into meaningful representation. *Cognitive Science, 29,* 103–134.

Shor, R. (1957). Effect of pre-information upon human characteristics attributed to animated geometric figures. *Journal of Abnormal and Social Psychology, 54,* 124–126.

Sidner, C. L. (1985). Plan parsing for intended response recognition in discourse. *Computational Intelligence, 1,* 1–10.

Siskind, J. M. (2003). Reconstructing force-dynamic models from video sequences. *Artificial Intelligence, 151,* 91–154.

Tavares, P., Lawrence, A. D., & Barnard, P. J. (2008). Paying attention to social meaning: An fMRI study. *Cerebral Cortex, 18,* 1876–1885.

The FrameNet Project. (2009). Retrieved August 25, 2009, from http://framenet.icsi.berkeley.edu/

Thibadeau, R. (1986). Artificial perception of actions. *Cognitive Science, 10,* 117–149.

Thomas, F., & Johnston, O. (1995). *The illusion of life: Disney animation.* New York: Hyperion.

Tremoulet, P. D., & Feldman, J. (2000). Perception of animacy from the motion of a single object. *Perception, 29,* 943–951. doi:10.1068/p3101

Tremoulet, P. D., & Feldman, J. (2006). The influence of spatial context and the role of intentionality in the interpretation of animacy from motion. *Perception & Psychophysics, 68,* 1047–1058.

Ullman, T. D., Baker, C. L., Macindoe, O., Evans, O., Goodman, N. D., & Tenenbaum, J. B. (2010). Help or hinder: Bayesian models of social goal inference. *Advances in Neural Information Processing Systems, 22,* 1874–1882.

Waytz, A., Cacioppo, J. T., & Epley, N. (2010). Who sees human? The stability and importance of individual differences in anthropomorphism. *Perspectives on Psychological Science, 5,* 219–232.

Wheatley, T., Milleville, S. C., & Martin, A. (2007). Understanding animate agents: Distinct roles for the social network and mirror system. *Psychological Science, 18,* 469–474.

Wolff, P. (2007). Representing causation. *Journal of Experimental Psychology. General, 136,* 82–111.

Young, J. E., Igarashi, T., & Sharlin, E. (2008). Puppet Master: Designing reactive character behavior by demonstration. In: M. Gross & D. James (Eds.), *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (pp. 183–191). European Association of Computer Graphics.

Zacks, J. M., & Swallow, K. (2007). Event segmentation. *Current Directions in Psychological Science, 16,* 80–84.