Grounding language in perception

Jeffrey Mark Siskind
University of Pennsylvania
Institute for Research in Cognitive Science
3401 Walnut Street, Room 407C
Philadelphia PA 19104
215/898–0367
internet: Qobi@CIS.UPenn.EDU

ABSTRACT

We describe an implemented computer program that recognizes the occurrence of simple spatial motion events in simulated video input. The program receives an animated line-drawing as input and produces as output a semantic representation of the events occurring in that movie. We suggest that the notions of support, contact, and attachment are crucial to specifying many simple spatial motion event types and present a logical notation for describing classes of events that incorporates such notions as primitives. We then suggest that the truth values of such primitives can be recovered from perceptual input by a process of counterfactual simulation, predicting the effect of hypothetical changes to the world on the immediate future. Finally, we suggest that such counterfactual simulation is performed using knowledge of naive physical constraints such as substantiality, continuity, gravity, and ground plane. We describe the algorithms that incorporate these ideas in the program and illustrate the operation of the program on sample input.

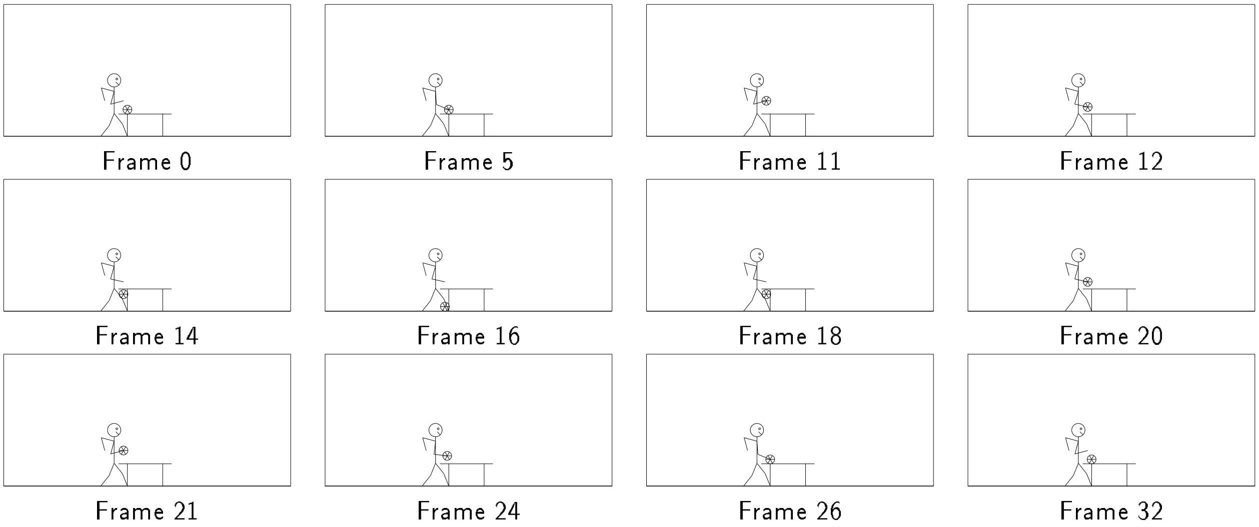
1 INTRODUCTION

People can describe what they see. Not only can they describe the objects that they see, they can also describe the events in which those objects participate. For example, when seeing a man throw a ball to a woman, a person might say *John threw the ball to Mary*. In this paper we present an implemented computer program called ABIGAIL that tries to mimic the human ability to describe visually observed events. In contrast to most prior work on visual recognition that attempts to segment a static image into distinct *objects* and classify those objects into distinct object types, this work instead focuses on segmenting a motion picture into distinct *events* and classifying those events into event types.

Our long-term goal is to apply the techniques described in this paper to actual video input. Since that is a monumental task, this paper describes a much more limited implementation. ABIGAIL watches a computer-generated stick-figure animation. Each frame in this animation is constructed out of figures, namely line segments and circles. ABIGAIL receives as input the positions, orientations, shapes, and sizes of these figures for each frame of the movie. From this input, ABIGAIL segments sequences of adjacent frames into distinct events, and classifies those events into event types such as dropping, throwing, picking up, and putting down. In segmenting and classifying events, ABIGAIL makes use of a library of event-type descriptions. These descriptions are analogous to the models used for model-based object recognition. We believe that the techniques described in this paper can be generalized to deal with real video.

The version of ABIGAIL described in this paper does not perform object segmentation or classification. Such information is provided as input to ABIGAIL. Siskind (1992) describes an automatic object segmentation technique that utilizes the same counterfactual simulation mechanisms that are discussed in this paper. No object models are needed to use that technique. We will not, however, discuss that object segmentation technique in this paper.

This paper advances three central claims. First, we claim that the notions of *support*, *contact*, and *attachment* are crucial to specifying the truth conditions for classifying a given event as an occurrence of a some event type as described by a simple spatial motion verb. For example, part of the standard meaning of the verb *throw* is the requirement that the object thrown be in unsupported motion after it leaves the hand of the thrower. Second, we claim that support relations between objects can be recovered by a process of *counterfactual simulation*, the ability to imagine the immediate future of the perceived world under the effect of forces such as gravity, and to project the effect of hypothetical changes to the world on the simulated outcome. For example, one can determine that an object is unsupported by predicting that it will fall immediately. Likewise, one can determine that an object A supports another object B if B is in fact supported but ceases to be supported when A is removed. We refer to this ability to perform counterfactual simulations as the *imagination capacity*. Finally, we claim that the human imagination capacity—if it exists—operates in a very different fashion from traditional kinematic simulators used to simulate the behavior of mechanisms under the effect of applied forces. Such simulators take physical accuracy to by primary—by performing numerical integration on Newton's Laws—and thus must take collision detection to be secondary. We propose a novel simulator that reverses these priorities. It is based instead on the naive physical notions of *substantiality*, *continuity*, *gravity*, and *ground plane*. The substantiality



Frame 0 Frame 5 Frame 11 Frame 12

Frame 14 Frame 16 Frame 18 Frame 20

Frame 21 Frame 24 Frame 26 Frame 32

Figure 1: Several key frames from a typical movie presented as input to Abigail.

constraint states that solid objects cannot pass through one another. The continuity constraint states that if an object first appears in one location—and later appears at a different location—it must have moved along a continuous path between those two locations. In other words, objects do not disappear and then reappear elsewhere later. The gravity constraint states that unsupported objects fall. Finally, the ground plane constraint states that the ground acts as universal support for all objects. We argue that a simulation strategy based on these naive physical notions is better suited to the event-recognition task.

The remainder of this paper is organized as follows. Section 2 describes the input to Abigail. Section 3 addresses the first claim, presenting an event logic for representing the truth conditions on the use of simple spatial motion verbs. Section 4 addresses the second claim, demonstrating the procedure used to recover support relations between objects. Section 5 addresses the third claim, discussing the simulation algorithms embodied in the imagination capacity. Section 6 gives an example of Abigail in operation. Section 7 discusses some related work. Finally, section 8 concludes with a discussion of the overall goals of this work.

2 THE INPUT TO ABIGAIL

Figure 1 shows several frames from a typical movie that Abigail can process. This movie depicts a man picking up a ball from the table, bouncing it on the floor, catching it, and placing it back on the table. We have a general facility for producing such movies from a *script*. Figure 2 illustrates the script used to produce the movie in figure 1. Abigail has no access to the script when processing the movie, for such access would be tantamount to providing the desired output as input. Abigail attempts to produce a description analogous to the script by tracking the changing position and orientation of the line segments and circles that constitute the frames of the movie, and comparing those changes to a library of event types.

The input to Abigail consists of a sequence of *frames*, each being a collection of *figures*. Figures have one of two *shapes*: line segment or circle. Each frame of the movie specifies the *position*, *orientation*, shape, and *size* of each figure. *Objects* such as table, balls, and people are aggregate entities constructed out of connected collections of figures.

Abigail imposes certain restrictions on the input movie. First, figures are always visible. They never appear or disappear. This implies that objects never enter or leave the field of view and are never occluded. Second, figures do not change shape or size during the course of the movie. Only their position and orientation can change. This implies that figures cannot bend, break into pieces, or fuse together. While figures as atomic entities cannot change shape or size, bend, break into pieces, or fuse together, objects—which are constructed out of collections of figures—can nonetheless undergo such changes in form.

The above constraints imply that it is possible to place the figures in adjacent frames into a one-to-one correspondence. For simplicity, Abigail is given this correspondence as input. It would not be difficult, however, to construct this

```
(define-movie a-ball-of-fun
    ((table (make-instance 'table :name 'table :x 16.0 :y 0.0))
     (ball (make-instance 'ball :name 'ball :x 14.0 :y 3.0))
     (john (make-instance 'man :name 'john :x 12.5 :y 0.0)))
 (pick-up (left-hand john) ball)
 (bounce john (x (left-hand john)) 0.5)
 (put-down (left-hand john)
           14.0
           (+ (y (p (top table))) (size (surface ball)))))
```

Figure 2: The script used to produce the movie depicted in figure 1.

correspondence given just the position and orientation of the figures in adjacent frames. One way of doing this would be to use a greedy algorithm to find the correspondence that minimized some cost function of the distances between paired figures. We refrain from implementing such automatic correspondence derivation as it is orthogonal to the main purpose of our work.

We also make some simplifying assumptions about the ability to perceive figures in certain situations. First, we assume that the orientation of line segments and circles can be perceived unambiguously, even though the orientation of a line segment is ambiguous between θ and $\theta + \pi$, and the orientation of a circle in indeterminate. Second, we assume that two collinear intersecting line segments can be perceived as distinct figures, even though they appear as one contiguous line segment. This means, for instance, that even though an elbow is straightened, the forearm and upper arm are perceived as distinct line segments. Finally, we assume that two concentric equiradial circles can be perceived as distinct figures, even though they appear to be a single circle due to the fact that they precisely overlap. These assumptions simplify the perceptual mechanisms to be described later.

The position and orientation of figures are directly perceivable quantities. Beyond these perceivable quantities, the perceptual mechanisms used by Abigail project a particular ontology onto the world in order to help recover information that is not directly perceivable. First, Abigail's ontology allows pairs of figures to be connected by *joints*. Such joints may be independently rigid or flexible along each of the three relative degrees of freedom between the two joined figures. We refer to such degrees of freedom as *joint parameters*. The rigidity of joint parameters may change over time. For example, observing someone bend their elbow requires implies that the elbow joint has a flexible rotation parameter. A later observation of that same arm supporting some grasped object requires the adoption of the belief that the elbow-joint rotation-parameter is now rigid in order to offer the necessary support to the grasped object. Similarly, the existence of joints may change over time. For example, the process of grasping an object when picking it up is modeled by the formation of a new joint between one's hand and the object. Likewise, the process of releasing an object when putting it down is modeled by the dissolution of that joint. The set of joints and their parameters collectively constitutes a *joint model*. Since joints are not directly perceivable, Abigail must construct and maintain a joint model that is consistent with the observed world.

Second, Abigail's ontology projects a third dimension onto the two-dimensional observed world. This is necessary since most movies depict events that would require objects to pass through one another if the world was two dimensional. For example, in figure 1 the ball might appear to pass through the table as it bounces. Humans are strongly biased against event interpretations that require one object to pass through another object. Such interpretations constitute violations of the substantiality constraint. A human observer would conjecture instead that the ball either passed in front of, or behind, the table during its bounce.

To model such phenomena, Abigail does not need a full third dimension—an impoverished one will do. Abigail's ontology assigns each figure to a *layer*. Layers are unordered. There is no notion of one layer being in front of, or behind, another. Furthermore, there is no notion of one layer being adjacent to another. Abigail's ontology allows only for the knowledge that two figures lie on the same, or on different, layers. Such knowledge is represented by *layer assertions* that specify that two figures are known to be on the same, or on different, layers. The collection of layer assertions constitutes a *layer model*. This layer model constitutes an equivalence relation, i.e. it is reflexive, symmetric, and transitive. Furthermore, it must be consistent. Two figures cannot simultaneously be on the same, and on different, layers.

Just as the joint model might need to change over time, the layer model too might need to change to remain consistent with the movie. For example, in figure 1 the ball must initially be on the same layer as the table top to account for the fact that the table top supports the ball, preventing it from falling. Later, as the ball bounces, it must be on a different layer than the table top to avoid a substantiality violation. Finally, as the ball comes to rest again on the table top at the end of the movie, they must again be on the same layer.

Joints and layer assertions are not directly perceivable quantities. Accordingly, ABIGAIL must construct and maintain joint and layer models that are consistent with the observed world. These models can change over time as the movie progresses. Siskind (1992) presents a mechanism whereby ABIGAIL can construct and update both the joint and layer models automatically, solely from the positions and orientations of the figures in each frame. Joint-model construction is currently not a robust process however. Thus, the version of ABIGAIL described in this paper incorporates only automatic layer-model construction. Accordingly, ABIGAIL is presented with an initial joint-model for the first frame as input, along with incremental changes to that model as the movie progresses.

3 REPRESENTATION OF EVENT TYPES

In order to recognize the occurrence of events in the world, we need some way of representing the truth conditions on occurrences of those events. Since we typically use verbs to name events, the truth conditions on event occurrence will constitute the definitions of the verbs used to name such events. We are not the first to attempt to construct formal verb definitions. Previous attempts include those of Miller (1972), Schank (1973), Jackendoff (1983, 1990), Borchardt (1984), and Pinker (1989). With the exception of Borchardt, these prior works did not attempt to ground the proposed verb definitions in perceptual input, i.e. they did not offer a procedure for determining whether some perceived event meets the truth conditions specified by some verb definition. In this section we present our *event logic*, a formal language for specifying the truth conditions on event types. We subsequently define several verbs using expressions in this event logic. We limit our consideration to simple spatial motion verbs in non-metaphoric uses, verbs like *throw*, *fall*, *drop*, *bounce*, *jump*, *put*, *pick up*, *carry*, *raise*, *slide*, *roll*, *step*, and *walk*. In subsequent sections we discuss how these definitions are grounded.

The task of formulating the necessary and sufficient truth conditions on the use of a verb to describe an event is immensely difficult. Many have argued that it is in-principle impossible to formulate definitions that clearly delineate occurrences from non-occurrences of an event. The problem arises in part because of the fuzzy nature of event classes. For any event type there will be events that are clear instances of that event type, those that clearly aren't instances, and those whose membership in that class of events is unclear. Philosophers often debate whether or not some instance really is in some class. We circumvent such epistemological issues of absolute truth and attempt only to construct a cognitive model of truth. More specifically, we require our definitions to label an event as an instance or non-instance of a given class precisely when humans would unequivocally make the same judgment. If humans were unsure as to whether a given event was an instance of some class we do not care whether or not our definition classifies the event as a member of the class, or whether it too indicates somehow that it is unsure.

Using the above criteria, one can assess the adequacy of a set of verb definitions only by experiment. Such experiments would rely crucially on the ability to compare human judgments with an impartial procedure for evaluating the truth values of those definitions. The adequacy of a given set of definitions can thus only be evaluated relative to the kind of perceptual input used to ground those definitions. Accordingly, we cannot claim that the definitions we give for verbs such as *pick up* constitute the truth conditions for all instances of pickings up that humans might perceive. Rather they only constitute the truth conditions for those instances of pickings up that can be formulated as animated line-drawings of the type that can be processed by ABIGAIL.

We have not yet performed the experiments necessary to test the adequacy of our definitions. Our definitions currently exhibit too many false positives and false negatives for such experiments to be meaningful. We do however, believe that our representations admit fewer false positives and negatives than those of our predecessors. We further believe that the goal of research in lexical semantics should be to strive for ever more robust definitions according to the aforementioned criteria.

We represent verb definitions as event-logic expressions. The truth value of an event-logic expression is relative to a particular interval of a particular movie. We write $e@(i, M)$ to denote the proposition that an event of the type described by the event description e occurred during interval i of the movie M. As the designated movie is usually invariant, we often write $e@i$ when the movie is clear from context. Note that we use the notation $e@i$ to specify that an event of type e started at the beginning of i and terminated at the end of i. The proposition $e@i$ would not be true, for instance, if e occurred during some subinterval or super-interval of i but not precisely during i. We will momentarily introduce a mechanism for specifying alternate commencement and termination requirements.

Our event logic has two components: a set of *perceptual primitives* that denote primitive event types, and a set of

Exists(x)	MovingPart(x)	$x = y$
Prominent(x)	Rotating(x)	Part(x, y)
Supported(x)	RotatingClockwise(x)	Disjoint(x, y)
Supports(x, y)	RotatingCounterClockwise(x)	
Contacts(x, y)	Translating(x)	
Attached(x, y)	TranslatingUp(x)	
At(x, y)	TranslatingDown(x)	
	TranslatingTowards(x, y)	
	TranslatingAwayFrom(x, y)	
	Flipping(x)	
	SlidingAgainst(x, y)	

Table 1: The event-logic perceptual primitives.

forms for combining event types into more complex aggregate event types. Table 1 lists the perceptual primitives that we currently use. We make no claim that these primitives are *sufficient* for defining all simple spatial motion verbs. We do however, believe that they are *necessary* for accurately specifying the truth conditions of the verbs we discuss in this paper, in the context of animated line-drawings.

The intuitive meaning of most of the perceptual primitives given in table 1 should be clear from their names. But such an intuitive interpretation does not suffice for defining their formal semantics. A significant limitation of the formal verb definitions given by Miller (1972), Schank (1973), Jackendoff (1983, 1990), Borchardt (1984), and Pinker (1989) is that they never precisely specify the semantics of the primitives they use to formulate those definitions. By grounding our primitives in the perceptual processes described in section 4 we give them a formal semantics, at least for the restricted domain of animated line-drawings. While perceptual grounding is not the only way one can precisely delineate the meaning of a calculus used to represent verb meaning, the desire to give a formal semantics for our event logic is a prime motivating force behind our attempt to provide such grounding. Lack of space limits our ability to present the definitions of all of our primitives in this paper. A future paper will contain such definitions.

The perceptual primitives from table 1 fall into three classes. The primitives in the rightmost column are time independent. Their truth values can be determined once for the whole movie. The truth values of those in the leftmost column can be determined from an individual frame in isolation. The truth values of such primitives are determined on a frame-by-frame basis. The truth values of the motion primitives in the central column cannot be determined from a single frame. The truth values of such primitives depend on an appropriate change between adjacent frames. By definition, when an requisite change happens between frame i and $i+1$ we say that the appropriate motion primitive is true both during frames i and $i+1$. Thus a motion primitive will be true during frame i if the requisite change happens either between frames $i-1$ and i, or between i and $i+1$. This introduces a slight anomaly that arises when an object moves between all pairs of adjacent frames in $i, \ldots, j$ and in $j+1, \ldots, k$ but is immobile for the single transition between frame j and $j+1$. In this situation, the motion primitive will be true for all frames i through j, filtering out the momentary immobility.

More complex event expressions can be composed out of simpler event expressions using the combining forms listed in table 2. The semantics of these combining forms is defined as follows. The proposition $(\neg e)@i$ is true if and only if $e@i$ is false. Note that $(\neg e)@i$ could be true even if e occurred during some subinterval or super-interval of i, just so long as no instance of e started precisely at the beginning of i and terminated precisely at the end of i. Similarly, $(e_1 \vee e_2)@i$ is true if and only if either $e_1@i$ is true or $e_2@i$ is true. Likewise, $(\forall x e)@i$ is true if and only if $e[o/x]@i$ is true for all objects o that have been seen so far. We use $e[o/x]$ to designate the expression derived by substituting o for all free occurrences of x in e. Similarly, $(\exists x e)@i$ is true if and only if $e[o/x]@i$ is true for some object o that has been seen so far. Note that $\forall$ and $\exists$ denote bounded quantification over only those objects (connected collections of figures) that have been observed.

The next three combining forms utilize a subscript R that ranges over subsets of the thirteen possible relations between two intervals as proposed by Allen (1983), namely $\{=,<,>,\mathsf{m},\mathsf{mi},\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}$. The proposition $(e_1 \wedge_R e_2)@i$ is true if and only if there exist two intervals j and k such that the relations $j\mathsf{s}i$ and $k\mathsf{f}i$ both hold, the propositions $e_1@j$ and $e_2@k$ are both true, and jrk for some $r \in R$. We abbreviate the special cases $e_1 \wedge_{\{=\}} e_2$ and $e_1 \wedge_{\{\mathsf{m}\}} e_2$ as $e_1 \wedge e_2$ and $e_1; e_2$ respectively. Thus $e_1 \wedge e_2$ describes an aggregate event where both e_1 and e_2 happen simultaneously, starting

$\neg e$	
$e_1 \vee e_2$	
$\forall x e$	
$\exists x e$	
$e_1 \wedge_R e_2$	Where $R \subseteq \{=,<,>,\mathsf{m},\mathsf{mi},\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}$
$\Diamond_R e$	Where $R \subseteq \{=,<,>,\mathsf{m},\mathsf{mi},\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}$
e^+	

Table 2: The event-logic combining forms.

and finishing at the same time, while $e_1;e_2$ describes an aggregate event of e_1 immediately followed by e_2. Similarly, $(\Diamond_R e)@i$ is true if and only if there exists an interval j such that $e@j$ and jri for some $r \in R$. The $\Diamond_R$ combining form can act as a tense operator. Expressions such as $\Diamond_{\{<\}}e$, $\Diamond_{\{>\}}e$, $\Diamond_{\{\mathsf{m}\}}e$, and $\Diamond_{\{\mathsf{mi}\}}e$ specify that e happened in the distant past, distant future, immediate past, or immediate future respectively. Finally $e^+@i$ is true if an only if there exists some set of intervals $\{i_1,\ldots,i_n\}$ such that $e@i_k$ for all $1 \leq k \leq n$ and $i_k\mathsf{m}i_{k+1}$ for all $1 \leq k < n$. The expression e^+ denotes contiguous repeated occurrences of e.

Table 3 illustrates some sample verb definitions formulated in our event logic. For example, the definition for *throw* states that x throws y if y is not a part of x and there is some z that is a part of x, typically x's hand, such that for some interval, z is attached to y, is touching y, and is moving with y, while in the immediate subsequent interval, z no longer is attached to y, no longer touches y, and y is in unsupported motion. Likewise, the definition for *fall* states that x falls if it is in unsupported downward motion. Similarly, the definition for *drop* states that x drops y if there is some z that is a part of x, typically x's hand, such that for some interval, z initially supports y by way of being attached to it, while in the immediate subsequent interval, z no longer supports, touches, or is attached to y, and y is unsupported.

Note that the notion of support plays a crucial role in these definitions. It is not sufficient for an object to be in motion after it leaves one's hand for one to have thrown that object. Rolling a ball does not constitute throwing. The object must be in unsupported motion after it leaves one's hand. Likewise, not all downward motion of objects constitutes falling. Airplanes (hopefully) are not falling when they descend to land. Objects must be in unsupported downward motion to be classified as falling.[1] Similarly, dropping too, requires that the dropped object be unsupported. One is not dropping a teacup when one places it gently in its saucer. Such a distinction between supported and unsupported motion plays a role in delineating the difference between the verbs *drop* and *put*. Siskind (1992) argues this case more extensively.

We do not claim that the definitions given in table 3 fully specify the necessary and sufficient truth conditions for the indicated event types. Nor do we claim that our current formulation of event-logic is sufficiently expressive to formulate such truth conditions. For instance, there is no way to specify the default expectation that an agent's hand is the part of the agent typically involved in throwing. Likewise, there is no model of prototypical classes, radial categories, or focus of attention.

When classifying events, prior researchers (Vendler 1967, Dowty 1979, Verkuyl 1989, Krifka 1992) have noted that some event types have the following two properties. First, if they are true during an interval i then they are also true during any subinterval of i. Second, if they are true during two intervals i and j such that $i\mathsf{m}j$ then they are also true during the encompassing interval that begins at the beginning of i and ends at the end of j. Events with these properties are termed *liquid*, following Shoham (1987). All of the event types denoted by perceptual primitives are liquid. Not all compound events are liquid however. Some compound events such as FALL, CARRY, RAISE, SLIDE, and ROLL are liquid, while others such as THROW, DROP, BOUNCE, JUMP, PUT, PICKUP, STEP, and WALK are not.

One aspect of our event logic deserves further mention. Note that an expression such as $(\neg e)@i$, does *not* mean that e *never* occurs during i. It means that no instance of e occurs beginning precisely at the beginning of i and ending precisely at the end of i. Thus an expression such as $\neg\textsc{Translating}(y)$ would be true of an interval if y was stationary for part of that interval but nonetheless moving for some other part. If one used $\neg\textsc{Translating}(y)$ in the definition of PUT, such a definition would admit events where the object continued to move for a while after it left the agents hand but before it eventually came to rest. To circumvent this problem we note that the proposition $(\Diamond_{\{=,\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}}e)@i$

[1] Appropriate cognitive modeling of the physical processes that constitute the support relationship for airplanes is beyond the scope of this paper.

$$\textsc{Throw}(x,y) \triangleq \neg\Diamond\textsc{Part}(y,x) \wedge \exists z \left(\begin{array}{c} \textsc{Part}(z,x)\wedge \\ \left[\left(\begin{array}{c}\textsc{Translating}(z)\wedge\\ \textsc{Contacts}(z,y)\wedge\\ \textsc{Attached}(z,y)\end{array}\right);\left(\begin{array}{c}\neg\Diamond\textsc{Contacts}(z,y)\wedge\\ \neg\Diamond\textsc{Attached}(z,y)\wedge\\ \neg\Diamond\textsc{Supported}(y)\end{array}\right)\right]\wedge\\ \textsc{Translating}(y)\end{array}\right)$$

$$\textsc{Fall}(x) \triangleq \neg\Diamond\textsc{Supported}(x) \wedge \textsc{TranslatingDown}(x)$$

$$\textsc{Drop}(x,y) \triangleq \exists z \left(\begin{array}{c} \textsc{Part}(z,x)\wedge \\ \left[\left(\begin{array}{c}\textsc{Contacts}(z,y)\wedge\\ \textsc{Attached}(z,y)\wedge\\ \textsc{Supports}(x,y)\wedge\\ \textsc{Supported}(y)\end{array}\right);\left(\begin{array}{c}\neg\Diamond\textsc{Contacts}(z,y)\wedge\\ \neg\Diamond\textsc{Attached}(z,y)\wedge\\ \neg\Diamond\textsc{Supports}(x,y)\wedge\\ \neg\Diamond\textsc{Supported}(y)\end{array}\right)\right]\end{array}\right)$$

$$\textsc{Bounce}(x) \triangleq \left(\begin{array}{c}\textsc{Translating}(x)\wedge\\ \exists y[\neg\Diamond\textsc{Contacts}(x,y);\textsc{Contacts}(x,y);\neg\Diamond\textsc{Contacts}(x,y)]\end{array}\right)$$

$$\textsc{Jump}(x) \triangleq \textsc{Supported}(x);\left(\begin{array}{c}\neg\Diamond\textsc{Supported}(x)\wedge\\ \textsc{TranslatingUp}(x)\end{array}\right)$$

$$\textsc{Put}(x,y) \triangleq \exists w \left(\begin{array}{c} \textsc{Part}(w,x)\wedge \\ \left[\left(\begin{array}{c}\textsc{Translating}(w)\wedge\\ \textsc{Contacts}(w,y)\wedge\\ \textsc{Attached}(w,y)\wedge\\ \textsc{Supports}(x,y)\wedge\\ \textsc{Translating}(y)\end{array}\right);\exists z\left(\begin{array}{c}\textsc{Disjoint}(z,w)\wedge\\ \neg\Diamond\textsc{Translating}(y)\wedge\\ \textsc{Supported}(y)\wedge\\ \textsc{Supports}(z,y)\end{array}\right)\right]\end{array}\right)$$

$$\textsc{PickUp}(x,y) \triangleq \exists w \left(\begin{array}{c} \textsc{Part}(w,x)\wedge \\ \left[\left(\exists z\left(\begin{array}{c}\textsc{Disjoint}(z,w)\wedge\\ \textsc{Supported}(y)\wedge\\ \textsc{Supports}(z,y)\wedge\\ \textsc{Contacts}(z,y)\end{array}\right)\right);\left(\begin{array}{c}\textsc{Translating}(w)\wedge\\ \textsc{Contacts}(w,y)\wedge\\ \textsc{Attached}(w,y)\wedge\\ \textsc{Supports}(x,y)\wedge\\ \textsc{Translating}(y)\end{array}\right)\right]\end{array}\right)$$

$$\textsc{Carry}(x,y) \triangleq \textsc{Translating}(x) \wedge \textsc{Translating}(y) \wedge \textsc{Supports}(x,y)$$

$$\textsc{Raise}(x,y) \triangleq \neg\Diamond\textsc{Part}(y,x) \wedge \textsc{Supports}(x,y) \wedge \textsc{TranslatingUp}(y)$$

$$\textsc{Slide}(x) \triangleq \exists y\textsc{SlidingAgainst}(x,y)$$

$$\textsc{Roll}(x) \triangleq \exists y\left(\begin{array}{c}\neg\Diamond\textsc{SlidingAgainst}(x,y)\wedge\\ [\textsc{RotatingClockwise}(x)\vee\textsc{RotatingCounterClockwise}(x)]\wedge\\ \textsc{Contacts}(x,y)\end{array}\right)$$

$$\textsc{Step}(x) \triangleq \exists y\left(\begin{array}{c}\textsc{Part}(y,x)\wedge\\ [\textsc{Contacts}(y,\mathbf{ground});\neg\Diamond\textsc{Contacts}(y,\mathbf{ground});\textsc{Contacts}(y,\mathbf{ground})]\end{array}\right)$$

$$\textsc{Walk}(x) \triangleq \left(\begin{array}{c}\textsc{Step}(x)^{+}\wedge\\ (\exists y[\textsc{Part}(y,x)\wedge\textsc{Contacts}(y,\mathbf{ground})])^{+}\wedge\\ \neg\Diamond\exists y[\textsc{Part}(y,x)\wedge\textsc{SlidingAgainst}(y,\mathbf{ground})]\end{array}\right)$$

Table 3: Some verb definitions formulated in our event logic.

can be used to express the statement that some part of some occurrence of e occurs sometime during some part of i, i.e. that an occurrence of e occurred during some subinterval or some super-interval of i, or during some other interval that overlaps with i. Similarly, the proposition $(\neg\Diamond_{\{=,\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}}e)@i$ can be used to express the statement that no part of e occurs during a part of i. We adopt the notation $\Diamond e$ as shorthand for $\Diamond_{\{=,\mathsf{o},\mathsf{oi},\mathsf{s},\mathsf{si},\mathsf{f},\mathsf{fi},\mathsf{d},\mathsf{di}\}}e$ allowing us to write $\neg\Diamond$Translating(y) to denote an interval during which y did not move at all.

4 RECOVERING SUPPORT RELATIONS

Abigail processes the movie a frame at a time. For each frame, Abigail performs the following operations in sequence. First, Abigail constructs a layer model consistent with the current frame.[2] For the first frame, the layer model must be computed from scratch. In subsequent frames, the layer model is updated incrementally from the layer model for the previous frame. Next, Abigail performs object segmentation, partitioning collections of figures in the frame into distinct objects. Then, Abigail computes the support relationships between the objects just segmented. Finally, Abigail determines the truth values of the perceptual primitives enumerated in table 1 for the current frame. Abigail tracks these changing truth values so that the truth values of the event types given in table 3 can be computed at the end of the movie.

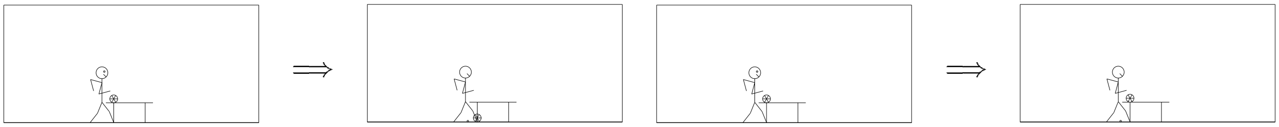
The layer model is computed by the following process. At all times Abigail maintains a layer-model L. This layer model is initially empty. Abigail uses the layer model from the previous frame to construct the layer model for the current frame. When processing each frame, Abigail first collects all pairs of figures that overlap and forms a set D of different-layer assertions between such figures. Such assertions are needed because the substantiality constraint would be violated if those figures were on the same layer. For example, when processing frame 0 of the movie shown in figure 1, Abigail would form a different-layer assertion between the man's left forearm and his torso. Abigail then collects all pairs of figures that touch but don't overlap as the set S of candidate same-layer assertions. For example, when processing frame 0, Abigail would form a candidate same-layer assertion between the surface of the ball and the table top. Note that $D \cup S$ may be inconsistent. For example, when processing frame 0, Abigail would form candidate same-layer assertions between the man's left forearm and upper arm, and between his left upper-arm and torso. Since same-layer assertions are transitive, these two candidates would imply that his left forearm was on the same layer as his torso, which is inconsistent with the different-layer assertion just formed. Also note that $D \cup S \cup L$ may be inconsistent—even if $D \cup S$ were consistent—since objects may change layers during the course of the movie. For example, in the movie shown in figure 1, while the surface of the ball must be on the same layer as the table top in frame 0, later on it must be on a different layer to avoid a substantiality violation as the ball falls. Thus, current different-layer assertions must take priority over both candidate, and previously-adopted, same-layer assertions. Furthermore, we wish to give previously-adopted same-layer assertions priority over newer candidate same-layer assertions. Let $C(D, S, L)$ denote a maximal consistent subset of $D \cup S \cup L$ where elements of D are given priority over elements of S, and elements of D and S are given priority over those elements of L that are not in either D or S. Abigail then uses the kinematic simulator described in section 5, with $C(D, S, L)$ as the layer model, to imagine the future and predict which figures fall and which do not. Abigail forms the set F of figures that did not fall and then selects a minimal subset S' of S such that no figure from F falls when imagining the future using $C(D, S', L)$ as the layer model instead of $C(D, S, L)$. Abigail then adopts $C(D, S', L)$ as the new layer model L for the current frame. In this way, Abigail decides that two figures are on the same layer only when necessary to generate a support relationship, and furthermore gives priority to previously-adopted same-layer assertions over newer candidate same-layer assertions when both generate the same support relationships.

Figure 3 gives an example of the layer-model construction performed when analyzing frame 0 of the movie shown in figure 1. Here, Abigail will adopt the same-layer assertion between the surface of the ball and the table top since the ball will fall without such an assertion but remains supported with that assertion.

The layer-model construction algorithm appears overly complicated. One may wonder why it is necessary to compute the set F and not simply compute a minimal subset S' of S such that no figure falls at all with $C(D, S', L)$ as the layer model. Figure 3 illustrates why this simpler algorithm will not work. In this example, since the eye is unsupported[3] no subset of S' will succeed in preventing all of the figures in the image from falling. Thus one can only find a minimal subset of S' that prevents those figures that could be supported from falling.

[2] Recall that in the implementation discussed here, the joint model is given as input to Abigail on a per-frame basis. In the implementation discussed in Siskind (1992), the joint model is computed from the image simultaneously with the layer model for each frame.

[3] The fact that there is no way to prevent the eye from falling during counterfactual simulation and also no way to have the eye be part of the man is a limitation of the world ontology currently incorporated into Abigail.



Frame 0 without same-layer assertion

Frame 0 with same-layer assertion

Figure 3: The use of counterfactual simulation during layer-model construction to determine that the surface of the ball is on the same layer as the table top in frame 0 of the movie depicted in figure 1.

Object segmentation is performed by a very simple procedure. Abigail finds the connected components in the graph whose vertices are figures and whose edges are joints that are rigid along all three joint parameters. Each such connected component is added to the list of objects. Objects are never removed from this list, even though in subsequent frames they may no longer be connected components as just defined. Nonetheless, objects may cease to exist and later come back into existence. An object *exists* if it is connected. For an object to exist, it need not be a connected component. It could be a part of a new object comprising a larger connected component. Also, for an object to exist, the joint parameters need not be rigid. It can consist of flexibly connected parts. The stronger connected-component and rigid-joint criteria are used only for determining the initial existence of newly detected objects and not for tracking their continued existence.

The perceptual primitive Exists(x) can be used to formulate definitions of verbs that depend on changes in the state of an object's existence. For example, *break* might be modeled as a change from existence to nonexistence. Likewise, *make* might be modeled as a change from nonexistence to existence. Similarly, *fix* might be modeled as a change from existence to nonexistence and then back again to existence.

Once model construction and object segmentation have been completed, the support relationships between objects can be computed. Such support relationships are also computed using counterfactual simulation. An object is supported in the current frame if it does not fall when imagining the immediate future of that frame. Similarly, an object A supports another object B in the current frame if B is supported but ceases to be supported when imagining the immediate future of an image that is identical to the one in the current frame except that the figures comprising A have been removed. A single run of the simulator can be used to determine the truth value of the Supported(x) primitive for all of the objects x in the movie. Those objects that fall during this single simulation are unsupported while those that remain stable are supported. Computing the truth value of the Supports(x, y) primitive for all pairs of objects x and y can be done with n calls to the simulator where n is the number of objects in the image. One calls the simulator once for each object x, removing that object from the image, and determining which other objects y fall or remain supported.

Figure 4 illustrates the use of counterfactual simulation to recover support relations when processing the movie shown in figure 1. The ball is supported in both frames 0 and 11 because it does not fall. The ball is unsupported in frame 14 since it does fall. The table supports the ball in frame 0 since the ball falls when the table is removed. Similarly the man supports the ball in frame 11 since the ball falls when the man is removed.

5 THE IMAGINATION CAPACITY

Much of our event-recognition procedure relies on counterfactual simulation, the ability to predict the immediate future subject to hypothetical changes to the world. This imagination capacity is used both for layer-model construction and to recover support relationships. Nominally, it would appear that such simulation could be performed by kinematic simulators typically used by mechanical engineers, since in Abigail's world ontology, objects correspond essentially to mechanisms and figures correspond to the links comprising such mechanisms. Figure 5 illustrates a problem with such a view. Conventional kinematic simulators use numerical integration to solve the differential equations obtained by applying Newton's Laws to the mechanism being simulated. Numerical integration requires the use of a nonzero step-size. Collision detection is performed after each step to determine whether or not to allow the state change entailed by that step. If this step size is large, there is the possibility that one object will pass totally through another object in the transition from one step to the next allowing a substantiality violation to go undetected. There is no way in principle to use a numerical integration procedure as a simulator that soundly avoids substantiality violations. If the step size is reduced as a practical attempt to mitigate this unsoundness, the numerical integration process becomes much slower. The simulation time becomes dependent on the distance that objects must move until they come to rest.

People appear to use a different process when projecting the future. First, from early infancy humans are very strongly

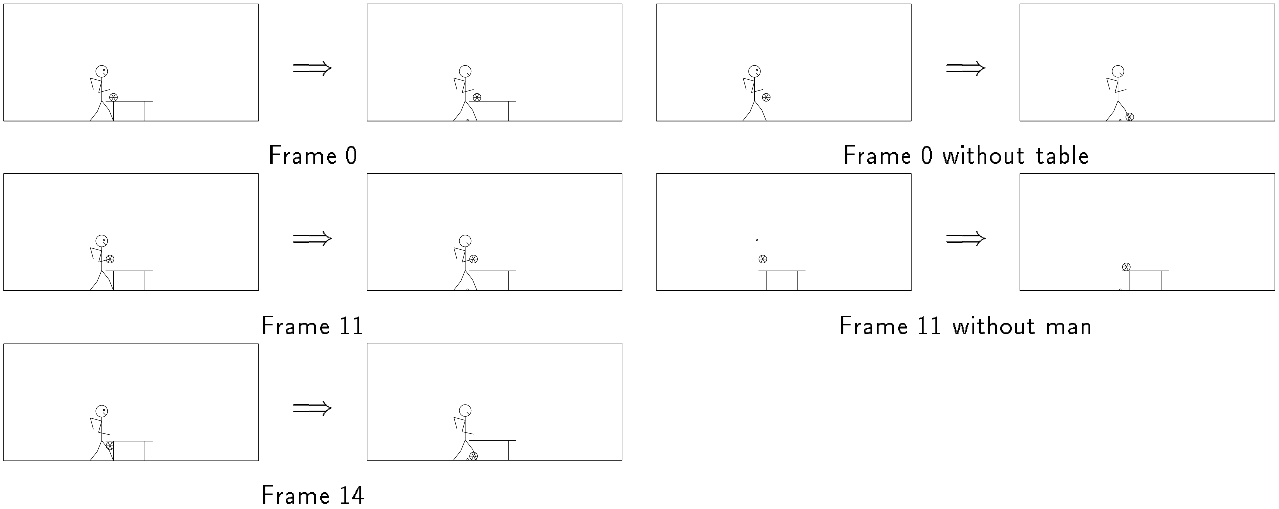


Figure 4: The use of counterfactual simulation during the recovery of support relations while processing the movie in figure 1. Abigail determines both that the ball is supported in frames 0 and 11 but unsupported in frame 14, and that the table supports the ball in frame 0 while the man supports the ball in frame 11.

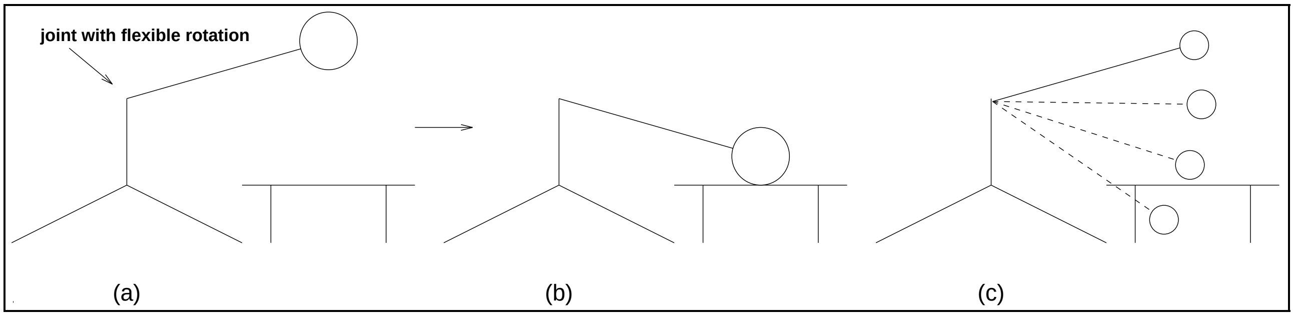


Figure 5: The simulator incorporated into Abigail's imagination capacity can predict in a single step that the joint will pivot exactly the amount needed until the ball lands on the table. Classical kinematic simulators based on numerical integration repeatedly vary the joint angle by a small step size until the ball collides with the table. If the step size is too small the simulation is slow. If the step size is too large the collision might not be detected, resulting in a simulation that violates the substantiality and continuity constraints. Abigail never produces such an anomalous prediction.

biased against visualizing substantiality violations (cf. Spelke 1988, though see Leslie 1988 for some exceptions). Second, from early infancy humans also are very strongly biased against imagining objects disappearing and later reappearing elsewhere. People appear to enforce a continuity constraint on object motion that is incompatible with the quantized motion of numerical simulators. Finally, people appear to be able to quickly predict that the joint in figure 5 will pivot precisely the amount needed to bring the ball in contact with the table, no more and no less. Such a capacity appears incompatible with numerical simulation.

The simulator used by Abigail is therefore based on very different principles. It directly incorporates the notions of substantiality and continuity, in addition to gravity—the fact that unsupported objects fall—and ground plane—the fact that the ground can support all objects. The simulator operates by examining all rigid collections of figures and considering all motions that such collections can exhibit as a result of gravity. Qualitatively, objects can exhibit four kinds of motion.

1. They can fall straight downward.
2. They can slide down an inclined surface.
3. They can fall over, pivoting about a contact point between a corner and a surface.
4. A part of an object can move along the degree of freedom of a flexible joint.

Abigail considers each such case separately and can calculate how much objects can move under these cases until they collide with other objects. Such calculation is performed analytically. The simulator consists simply of a loop that chooses an object to move—along with a type of motion—and moves that object the analytically defined amount in a single step.

Abigail's simulator suffers from numerous limitations. It cannot accurately predict the time course of events since it does not model velocity, momentum, or kinetic energy. Thus it can incorrectly predict the outcome of a situation exhibiting simultaneous motion of interacting objects. Furthermore, the analytical calculations can only be done for objects moving along linear or circular paths. Thus this technique is not suitable for simulating mechanisms with closed loop kinematic chains, for such mechanisms exhibit more complex motion.

Humans however, appear to exhibit the same limitations as Abigail. Furthermore, despite the limitations of Abigail's simulator, it is nonetheless capable of sufficient accuracy for its intended uses: constructing layer models and determining support relationships. A kinematic simulator based on the notions of substantiality, continuity, gravity, and ground plane is well suited to this task since these naive physical notions are intimately intertwined with the notions of support, contact, and attachment. Furthermore, the latter notions form the basis of event perception as discussed in section 3. We conjecture that the human imagination capacity is designed the way it is precisely because an imagination capacity based on naive physics is better matched to the task of event perception than one based on Newtonian physics. While the primary objective of our work is the engineering goal of building machines capable of visually recognizing events, such speculation on the design of the human perceptual and cognitive apparatus is an interesting and entertaining diversion.

6 AN EXAMPLE

Figure 6 shows the output generated by Abigail when processing the movie from figure 1. Each line indicates a detected event from the event definitions given in table 3. Notation such as `[BALL]` denotes collections of figures that constitute objects that partake in the detected events. Since Abigail does not perform object classification, these names are given as input. They are used solely to make the output more readable and are not accessed in any other way during the event perception process. The notation $[i:j,k:l]e$ specifies that an occurrence of the event type e was detected for all intervals starting in frames between i and j and ending in frames between k and l. We call such a concise representation of numerous intervals a *spanning interval*. The abbreviations $[i:j,k]e$, $[i,k:l]e$, $[i:k]e$, $[i,k]e$, and $[i]e$ are shorthand for $[i:j,k:k]e$, $[i:i,k:l]e$, $[i:k,i:k]e$, $[i:i,k:k]e$, and $[i:i,i:i]e$ respectively. The algorithm used by Abigail for evaluating the truth value of event-logic expressions computes directly with spanning intervals. This affords a substantial savings in both the space and time needed for event recognition. This algorithm will be discussed in a future paper.

Note that Abigail successfully recognizes occurrences of the putting down, bouncing, dropping, falling, and throwing events, and places them at the correct points in time. The event recognition process, however, suffers from a number of false positives and negatives. Abigail fails to recognize the picking up event. This is because the definition of PickUp consists primarily of an expression of the form $e_1;e_2$, and e_1 is true during the interval ending in frame 5 while e_2 is true during the interval beginning in frame 7, but these intervals do not precisely meet. There are also numerous spurious

```
[20:21](RAISE [(LEFT-FOREARM JOHN)] [BALL])
[7:11](RAISE [(LEFT-FOREARM JOHN)] [BALL])
[27:32](RAISE [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[20:21](RAISE [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[6:11](RAISE [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[20:21](RAISE [(LEFT-UPPER-ARM JOHN)] [BALL])
[6:11](RAISE [(LEFT-UPPER-ARM JOHN)] [BALL])
[27:32](RAISE [JOHN-part 5] [(LEFT-FOREARM JOHN)])
[20:21](RAISE [JOHN-part 5] [(LEFT-FOREARM JOHN)])
[6:11](RAISE [JOHN-part 5] [(LEFT-FOREARM JOHN)])
[28:32](RAISE [JOHN-part 5] [(LEFT-UPPER-ARM JOHN)])
[25:26](RAISE [JOHN-part 5] [(LEFT-UPPER-ARM JOHN)])
[7:12](RAISE [JOHN-part 5] [(LEFT-UPPER-ARM JOHN)])
[4:5](RAISE [JOHN-part 5] [(LEFT-UPPER-ARM JOHN)])
[20:21](RAISE [JOHN-part 5] [BALL])
[6:11](RAISE [JOHN-part 5] [BALL])
[20:21](RAISE [JOHN-part 3] [BALL])
[7:11](RAISE [JOHN-part 3] [BALL])
[20:26](CARRY [(LEFT-FOREARM JOHN)] [BALL])
[7:12](CARRY [(LEFT-FOREARM JOHN)] [BALL])
[27:32](CARRY [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[20:26](CARRY [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[6:12](CARRY [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[1:5](CARRY [(LEFT-UPPER-ARM JOHN)] [(LEFT-FOREARM JOHN)])
[20:26](CARRY [(LEFT-UPPER-ARM JOHN)] [BALL])
[6:12](CARRY [(LEFT-UPPER-ARM JOHN)] [BALL])
[26](CARRY [JOHN-part 3] [BALL])
[20:21](CARRY [JOHN-part 3] [BALL])
[12](CARRY [JOHN-part 3] [BALL])
[7](CARRY [JOHN-part 3] [BALL])
[20:26,27](PUT [JOHN-part 3] [BALL])
[16,17:19](JUMP [BALL])
[6:15,17:26](BOUNCE [BALL])
[7:12,13:15](DROP [JOHN-part 3] [BALL])
[13:15](FALL [BALL])
[6:12,13:15](THROW [JOHN-part 3] [BALL])
```

Figure 6: The events recognized by ABIGAIL when processing the movie from figure 1.

recognitions of raising and carrying events, as well as a spurious jumping event. This is because the current definitions of *raise*, *carry*, and *jump* are too loose and admit many false positives. Much work remains to be done to more accurately characterize the necessary and sufficient truth conditions on the use of simple spatial motion verbs. We believe however, that the methodology presented in this paper offers an appropriate framework for continuing that work.

7 RELATED WORK

The work described in this paper sits in the context of much prior work. Miller (1972), Schank (1973), Jackendoff (1983, 1990), and Pinker (1989) all present alternate representations for simple spatial motion verbs. None of this prior work perceptually grounds the presented representations. Thibadeau (1986) describes a system that processes the movie created by Heider and Simmel (1944) and determines when events occur. The Heider and Simmel movie depicts two-dimensional geometric objects moving in a plane. When viewing that movie, most people project an elaborate story onto the motion of abstract objects. Thibadeau's system does not classify event types. It just produces a single binary function over time delineating when an 'event' is said to have occurred. Badler (1975), Adler (1977), Tsotsos (1977), Tsuji et al. (1977), Tsotsos and Mylopoulos (1979), Tsuji et al. (1979), Okada (1979), Abe et al. (1981), and Borchardt (1984) all describe various implemented and unimplemented strategies for grounding the non-metaphoric meanings simple spatial motion verbs in animated line-drawings though only Borchardt's system utilizes changing support, contact, and attachment relations as a central part of the definition of event types. Borchardt's system receives support and contact information as input, in contrast to our system, which calculates such information using counterfactual simulation. Herskovits (1986) describes an unimplemented theory of the truth conditions underlying English spatial prepositions. Regier (1992) describes an implemented system that can learn the truth conditions on the use of spatial terms in a variety of languages in a language-independent fashion. Funt (1980) describes a counterfactual simulator for determining support relationship between objects in a static image. His system operates on a concentric retinotopic bitmap representation of the image rather than on line drawings. Cremer (1889) and Kramer (1990a, 1990b) describe more conventional kinematic simulators that take physical accuracy to be primary and collision detection to be secondary. Freyd et al. (1988) presents experimental evidence that humans subconsciously imagine things falling when their source of support is removed.

8 CONCLUSION

We have described a computer program that recognizes the occurrence of simple spatial motion events in animated line-drawings. We presented an event logic for describing classes of event types and used that logic to define a number of simple spatial motion verbs. We argued that the truth conditions of such verbs depends crucially on the notions of support, contact, and attachment. We showed how the truth values of such primitives can be recovered from perceptual input by a process of counterfactual simulation, predicting the effect of hypothetical changes to the world on the immediate future. Finally, we argued that such counterfactual simulation should be based on naive physical constraints such as substantiality, continuity, gravity, and ground plane, and not on Newtonian physics.

The main goal of this work is to develop a sound methodology for formalizing the meanings of verbs and studying the relationship between verb meaning and perception. This paper however, offers only an initial attempt at developing such a methodology. We do not claim that the verb definitions we give in table 3 are correct in their current form. They clearly suffer from numerous deficiencies. We also do not claim that the perceptual primitives given in table 1 or the combining forms given in table 2 are adequate for constructing the ultimate necessary and sufficient truth conditions for all verbs, let alone the ones discussed in this paper. We nonetheless do claim that the methodology that we employ in this paper—of precisely defining the semantics of the verb meaning representation language via perceptual grounding and experimentally verifying that the representation language combined with the definitions phrased in that language accurately collectively reflect the truth conditions on verb use—is the only methodology that will yield quality lexical semantic representations. Future work will attempt to remove the deficiencies of the current system within this methodological framework.

9 ACKNOWLEDGMENTS

This research was supported in part by an AT&T Bell Laboratories Ph.D. scholarship to the author, by a Presidential Young Investigator Award to Professor Robert C. Berwick under National Science Foundation Grant DCR–85552543, by a grant from the Siemens Corporation, and by the Kapor Family Foundation. This research was also supported in part by ARO grant DAAL 03–89–C–0031, by DARPA grant N00014–90–J–1863, by NSF grant IRI 90–16592 and by Ben Franklin grant 91S.3078C–1. Part of this research was performed while the author was visiting Xerox PARC as a research intern and as a consultant. I wish to thank Stuart Russell for pointing me to Shoham (1987), and Philip Resnik and Michael White for reading and commenting on earlier drafts of this paper. Any remaining errors are, of course, my own.

10 REFERENCES

1. Norihiro Abe, Itsuya Soga, and Saburo Tsuji. A plot understanding system on reference to both image and language. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 77–84, 1981.

2. Mark R. Adler. Computer interpretation of peanuts cartoons. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, page 608, August 1977.

3. James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery*, 26(11):832–843, November 1983.

4. Norman I. Badler. Temporal scene analysis: Conceptual descriptions of object movements. Technical Report 80, University of Toronto Department of Computer Science, February 1975.

5. Gary Borchardt. A computer model for the representation and identification of physical events. Technical Report T-142, Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, May 1984.

6. James F. Cremer. *An Architecture for General Purpose Physical System Simulation—Integrating Geometry, Dynamics, and Control.* PhD thesis, Cornell University, April 1989. Available as TR 89–987.

7. David R. Dowty. *Word Meaning and Montague Grammar.* D. Reidel Publishing Company, 1979.

8. Jennifer J. Freyd, Teresa M. Pantzer, and Jeannette L. Cheng. Representing statics as forces in equilibrium. *Journal of Experimental Psychology, General*, 117(4):395–407, December 1988.

9. Brian V. Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13:201–230, 1980.

10. Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *Journal of Psychology*, 57:243–259, 1944.

11. Annette Herskovits. *Language and Spatial Cognition: An interdisciplinary study of the prepositions in English.* Cambridge University Press, 1986.

12. Ray Jackendoff. *Semantics and Cognition.* M. I. T. Press, Cambridge, MA, 1983.

13. Ray Jackendoff. *Semantic Structures.* M. I. T. Press, Cambridge, MA, 1990.

14. Glenn Andrew Kramer. Geometric reasoning in the kinematic analysis of mechanisms. Technical Report TR–91–02, Schlumberger Laboratory for Computer Science, October 1990.

15. Glenn Andrew Kramer. Solving geometric constraint systems. In *Proceedings of the Eighth National Conference on Artifical Intelligence*, pages 708–714. Morgan Kaufmann Publishers, Inc., July 1990.

16. Manfred Krifka. Thematic relations as links between nominal reference and temporal constitution. In Ivan A. Sag and Anna Szabolcsi, editors, *Lexical Matters*. CSLI, 1992.

17. Alan M. Leslie. The necessity of illusion: Perception and thought in infancy. In L. Weiskrantz, editor, *Thought without Language*, chapter 8, pages 185–210. Clarendon Press, 1988.

18. George A. Miller. English verbs of motion: A case study in semantics and lexical memory. In Arthur W. Melton and Edwin Martin, editors, *Coding Processes in Human Memory*, chapter 14, pages 335–372. V. H. Winston and Sons, Inc., Washington, DC, 1972.

19. Naoyuki Okada. SUPP: Understanding moving picture patterns based on linguistic knowledge. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 690–692, 1979.

20. Steven Pinker. *Learnability and Cognition*. M. I. T. Press, Cambridge, MA, 1989.

21. Terrance Philip Regier. *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization*. PhD thesis, University of California at Berkeley, 1992.

22. Roger C. Schank. The fourteen primitive actions and their inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory, March 1973.

23. Yoav Shoham. Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence*, 33(1):89–104, September 1987.

24. Jeffrey Mark Siskind. *Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, January 1992.

25. Elizabeth S. Spelke. The origins of physical knowledge. In L. Weiskrantz, editor, *Thought without Language*, chapter 7, pages 168–184. Clarendon Press, 1988.

26. Robert Thibadeau. Artificial perception of actions. *Cognitive Science*, 10(2):117–149, 1986.

27. John K. Tsotsos. Some notes on motion understanding. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, page 611, August 1977.

28. John K. Tsotsos and John Mylopoulos. ALVEN: A study on motion understanding by computer. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 890–892, 1979.

29. Saburo Tsuji, Akira Morizono, and Shinichi Kuroda. Understanding a simple cartoon film by a computer vision system. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 609–610, August 1977.

30. Saburo Tsuji, Michiharu Osada, and Masahiko Yachida. Three dimensional movement analysis of dynamic line images. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 896–901, 1979.

31. Zeno Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.

32. H. J. Verkuyl. Aspectual classes and aspectual composition. *Linguistics and Philosophy*, 12(1), 1989.