Computational Perception of Scene Dynamics

Richard Mann, Allan Jepson*, and Jeffrey Mark Siskind**

Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4 CANADA

Abstract. Understanding observations of interacting objects requires one to reason about qualitative scene dynamics. For example, on observing a hand lifting a can, we may infer that an 'active' hand is applying an upwards force (by grasping) to lift a 'passive' can. We present an implemented computational theory that derives such dynamic descriptions directly from camera input. Our approach is based on an analysis of the Newtonian mechanics of a simplified scene model. Interpretations are expressed in terms of assertions about the kinematic and dynamic properties of the scene. The feasibility of interpretations can be determined relative to Newtonian mechanics by a reduction to linear programming. Finally, to select plausible interpretations, multiple feasible solutions are compared using a preference hierarchy. We provide computational examples to demonstrate that our model is sufficiently rich to describe a wide variety of image sequences.

1 Introduction

Understanding observations of image sequences requires one to reason about qualitative scene dynamics. As an example of the type of problem we are considering, refer to the image sequence in the top row of Figure 1, where a hand is reaching for, grasping, and then lifting a coke can off of a table. Given this sequence, we would like to be able to infer that an 'active' hand (and arm) is applying an upward force (by grasping) on a 'passive' coke can to raise the can off of the table. In order to perform such reasoning, we require a representation of the basic force generation and force transfer relationships of the various objects in the scene. In this work we present an implemented computational system that derives symbolic force-dynamic descriptions directly from camera input.

The use of domain knowledge by a vision system has been studied extensively for both static and motion domains. Many prior systems have attempted to extract event or conceptual descriptions from image sequences based on spatio-temporal features of the input [1, 23, 18, 4, 14]. A number of other systems have attempted to represent structure in static and dynamic scenes using qualitative physical models or rule-based systems [6, 8, 13, 20, 22, 5]. In contrast to both

* Also at Canadian Institute for Advanced Research.

** Current address: Department of Electrical Engineering, Technion, Haifa 32000, ISRAEL

of these approaches, our system uses an explicit physically-based representation based on Newtonian physics.

A number of other systems have used physically-based representations. In particular, Ikeuchi and Suehiro [10] and Siskind [21] propose represention of events based on changing kinematic relations in time-varying scenes. Also, closer to our approach, Blum *et. al.* [3] propose a representation of forces in static scenes. Our system extends these approaches to consider both kinematic and dynamic properties in time-varying scenes containing rigid objects.

Fig. 1. The example sequences: coke, cars, arch, and tip. The frame numbers are given below each image.

2 Ontology

In this section we describe the form of the system's representation for its domain. This representation must be suitable for specifying the geometry of the scene interpretation and the type of forces that can be generated on the various objects. Moreover, in order to avoid unphysical interpretations, there must be a notion of consistency for particular scene models. We describe the representation of the geometry, the types of forces, and the notion of consistency in the next sections.

2.1 Kinematic Model

The basic primitive for an object part is a rigid two-dimensional convex polygon. A single *object* is a rigid union of convex polygons.

To represent the spatial relationship between objects in the scene we use a *layered* scene model. In our layered model there is no depth ordering. Instead, we represent only whether two objects are in the same layer, in adjacent layers, or in layers separated in depth. Objects can contact either within the same layer or between adjacent layers. The first type of contact, called *abutting contact*, occurs when two objects in the same layer contact at a point or at an edge along their boundary. The second type of contact, called *overlapping contact*, occurs when two objects in adjacent depth layers contact over part of their surfaces and the region of overlap has non-zero area.

In order for a given assignment of contacts to be *admissible* two types of constraints must be satisfied. First, each pair of objects considered to be contacting must actually intersect (but possibly just on their boundary). Second, in the case of abutment, the contact is admissible only if the relative motion between the two objects is tangential to the contacting region (i.e. objects can slide along their contact region, but cannot penetrate or separate). Together these constraints provide a weak kinematic model involving only pairwise constraints between objects.

2.2 Dynamic Model

In order to check the consistency of an interpretation, we need to represent dynamic information about each object. This involves specifying the motion of each object along with its mass, center of mass, and moment of inertia. In our system the 2D velocities, angular velocities, and accelerations of the objects are all provided by the image observations. An object's total mass is taken to be a positive, but otherwise unknown, parameter. We take each object's center of mass to be at the object's geometric center. For the case of two-dimensional motion considered in this paper the inertial tensor I is a scalar. In order to reflect the uncertainty of the actual mass distribution, we allow a range for I. An upper bound for I is provided by considering an extreme case where all of the mass is placed at the furthest point from the center. A lower bound is provided by considering an alternate case where all of the mass is distributed uniformly inside a disk inscribed in the object.

An object is subject to gravitational and inertial forces, and to forces and torques resulting from contact with other objects. The dynamics of the object under these forces is obtained from the physics-based model described in §3.

Finally, particular objects may be designated as *ground*. We typically use this for the table top. Forces need not be balanced for objects designated as ground.

It is convenient to define a *configuration* to be the set of scene properties that are necessarily present, given the image data and any restrictions inherent in the ontology. For example, in the current system, the positions, velocities, and accelerations of the objects are provided by the image observations, and the positions of the centers of mass are fixed, by our ontology, to be at the object centroids.

2.3 Assertions

In order to supply the information missing from a configuration, we consider *assertions* taken from a limited set of possibilities. These assertions correspond to our hypothesis about the various contact relations and optional types of force generation and force transfer relationships between objects.

Currently, our implementation uses the following *kinematic assertions* which describe the contact relationships between objects:

- Contact(o_1, o_2, c) — objects o_1 and o_2 contact in the scene with the region of contact c;
- Attach(o_1, o_2, p) — objects o_1 and o_2 are attached at some set p of points in the contact region.

The intuitive meaning is that attachment points are functionally equivalent to rivets, fastening the objects together. Attached objects can be pulled, pushed, and sheared without coming apart while, without the attachment, the contacting objects may separate or slide on each other depending on the applied forces and on the coefficient of friction.

In addition we consider the following *dynamic assertions* which determine the types of optional forces which might be generated:

- BodyMotor(o) — object o has a 'body motor' that can generate an arbitrary force and torque on itself;
- LinearMotor(o_1, o_2, c) — a linear motor exists between the abutting objects o_1 and o_2. This motor can generate an arbitrary tangential shear force across the motor region c. This region must be contained within the contact region between the objects;
- AngularMotor(o_1, o_2, p) — an angular motor exists at a single point p that can generate an arbitrary torque about that point. The point p must be within the contact region between the objects.

The intuitive meaning of a BodyMotor is that the the object can generate an arbitrary force and torque on itself, as if it had several thrusters. LinearMotors are used to generate a shear force across an abutment (providing an abstraction for the tread on a bulldozer). AngularMotors are used to generate torques at joints.

We apply the following admissibility constraints to sets of assertions. First the contact conditions described in §2.1 must be satisfied for each assertion of contact. Second, linear motors are admissible only at point-to-edge and edge-to-edge abutments but not at point-to-point abutments or overlapping contacts. Finally, angular motors are admissible only at a single point within the contact region between two objects and the objects must be attached at this point.

We define an interpretation $i = (C,\ A)$ to consist of the configuration C, as dictated by the image data, along with a complete set of assertions A. (A set of assertions is complete when every admissible assertion has been specified as being true or false.) In the next section we will show how to test the feasibility of various interpretations.

3 Feasible Interpretations

Given an interpretation $i = (C,\ A)$ we can use a theory of dynamics to determine if the interpretation has a feasible force balance. In particular, we show how the test for consistency within the physical theory can be expressed as a set of algebraic constraints that, when provided with an admissible interpretation, can be tested with linear programming. This test is valid for both two and three dimensional scene models.

For rigid bodies under continuous motion, the dynamics are described by the Newton-Euler equations of motion [9] which relate the total applied force and torque to the observed accelerations of the objects. Given a scene with convex polygonal object parts, we can represent the forces between contacting parts by a set of forces acting on the vertices of the convex hull of their contact region [7, 2]. Under this simplification, the equations of motion for each object can be written as a set of equality constraints which relate the forces and torques at each contact point to the object masses and accelerations.

The transfer of forces between contacting objects depends on whether the objects are in resting contact, sliding contact, or are attached. Attached objects have no constraints on their contact forces. However, contacts which are not asserted to be Attached are restricted to have a positive component of normal force. In addition, contact points that are not part of a LinearMotor have tangential forces according to the *Coulombic* model of friction. In particular, the magnitude of the tangential force is bounded by some multiple of the magnitude of the normal force. Both sliding and resting friction are modeled.

An interpretation is dynamically feasible if these motion equations can be satisfied subject to the contact conditions and the bounds on the mass and inertia described in §2.2. Since we can approximate these constraints by a set of linear equations and inequalities, dynamic feasibility can be tested using linear programming (see [17] for details).

4 Preferences

Given a fairly rich ontology, it is common for there to be multiple feasible interpretations for a given scene configuration. For example, for the lifting phase of the coke sequence in Figure 1 there are five feasible interpretations, as shown in Figure 2. Indeed, for any scene configuration there is always at least one trivial interpretation in which every object has a body motor, and thus multiple interpretations can be expected.

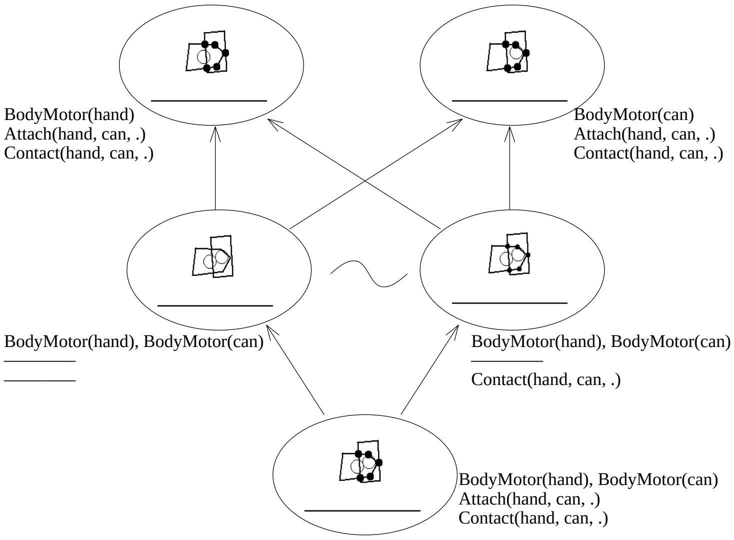

Fig. 2. The preference ordering for the five feasible interpretations of the lifting phase in the coke sequence (Frame 63). A large open circle at the object center denotes a BodyMotor. The small black disks denote contact points while the larger disks denote attachment points. A textual form of the assertions appears adjacent to each interpretation. The three levels of priority are represented by each line of text. The absence of an assertion denotes its negation.

Rather than searching for all interpretations, we seek interpretations that require, in some specified sense, the *weakest* properties of the various objects. We use model preference relations, as discussed by Richards, Jepson, and Feldman [19], to express a suitable ordering on the various interpretations. The basic idea is to compare pairs of interpretations using a prioritised set of elementary preference relations.

Our current ontology includes the following elementary preferences for the *absence* of any motor:

– $P_{bodymotor}(o) : \neg\textsc{BodyMotor}(o) \succ \textsc{BodyMotor}(o)$;

- $P_{linearmotor}(c) : \neg\textsc{LinearMotor}(o_1, o_2, c) \succ \textsc{LinearMotor}(o_1, o_2, c)$;
- $P_{angularmotor}(c) : \neg\textsc{AngularMotor}(o_1, o_2, p) \succ \textsc{AngularMotor}(o_1, o_2, p)$.

Here $\neg$ denotes the negation of the predicate that follows. These elementary preference relations all encode the specification that it is preferable not to resort to the use of a motor, all else being equal. The absence of a motor is considered to be a weaker assumption about an object's properties. These elementary preference relations appear at the highest priority.

At the next level of priority we have

- $P_{attach}(o_1, o_2, p) : \neg\textsc{Attach}(o_1, o_2, p) \succ \textsc{Attach}(o_1, o_2, p)$,

so the absence of an attachment assertion is also preferred. Finally, at the lowest level of priority, we have the indifference relation

- $P_{contact}(o_1, o_2, c) : \neg\textsc{Contact}(o_1, o_2, c) \sim \textsc{Contact}(o_1, o_2, c)$,

so the system is indifferent to the presence or absence of contact, all else being equal.

All of the above preferences, except for the indifference to contact, have the form of a preference for the negation of an assertion over the assertion itself. It is convenient to use the absence of an assertion to denote its negation. When the elementary preferences can be written in this simple form, the induced preference relation on interpretations is given by prioritised subset ordering on the sets of assertions made in the various feasible interpretations. As illustrated in Figure 2, we can determine the preference order for any two interpretations by first comparing the assertions made at the highest priority. If the highest priority assertions in one interpretation are a subset of the highest priority assertions in a second interpretation, the first interpretation is preferred. Otherwise, if the two sets of assertions at this priority are not ordered by the subset relation, that is neither set contains the other, then the two interpretations are considered to be unordered. Finally, in the case that the assertions at the highest priority are the same in both interpretations, then we check the assertions at the next lower priority, and so on. This approach, based upon prioritised ordering of elementary preference relations, is similar to prioritised circumscription [15].

To find maximally-preferred models, we search the space of possible interpretations. We perform a breadth-first search, starting with the empty set of assertions, incrementally adding new assertions to this set. Each branch of the search terminates upon finding a minimal set of assertions required for feasible force balancing. Note that because we are indifferent to contacts, we explore every set of admissible contact assertions at each stage of the search. While in theory this search could require the testing of every possible interpretation, in practice it often examines only a fraction of the possible interpretations since the search terminates upon finding minimal models.

Moreover, when the assertions are stratified by a set of priorities we can achieve significant computational savings by performing the search over each priority level separately. For example, under our preference ordering, we can search for minimal sets of motors using only interpretations that contain all admissible attachments. It is critical to note that this algorithm is only correct because of the special structure of the assertions and the domain. The critical property is that if there is a feasible interpretation $i = (C, A)$, and if A' is the set obtained by adding all of the admissible attachments to A, then the interpretation $i = (C, A')$ is also feasible. This property justifies the algorithm above where we set all of the lower priority assertions to the most permissive settings during each stage of the minimization. In general we refer to this property as *monotonicity* [16].

5 Examples

We have applied our system to several image sequences taken from a desktop environment (see Figure 1). The sequences were taken from a video camera attached to a SunVideo imaging system. MPEG image sequences were acquired at a rate of thirty frames per second and a resolution of 320×240 pixels. The 24-bit colour image sequences were converted to 8-bit grey-scale images used by the tracker.

As described in §2.1, we model the scene as a set of two-dimensional convex polygons. To obtain estimates for the object motions we use a view-based tracking algorithm similar to the optical flow and stereo disparity algorithms described in [12, 11]. The input to the tracker consists of the image sequence, a set of object template images (including a polygonal outline for each object), and an estimate for the object positions in the first frame of the sequence. In addition, we provide an estimate for the position of the table top which is designated as a *ground* object in our ontology. The tracking algorithm then estimates the position and orientation of these initial templates throughout the image sequence by successively matching the templates to each frame. The position of the object polygons is obtained by mapping the original outlines according to these estimated positions. Finally, the velocity and acceleration of the polygons are obtained using a robust interpolation algorithm on these position results.

In the current system we consider interpretations for each frame in isolation. Given estimates for the shapes and motions of the objects in each frame, we determine possible contact relations assuming a layered model as described in §2.1. For each possible contact set[1] we determine the admissible attachment and motor assertions described in §2.3. Finally, a breadth-first search is performed to find the preferred interpretations for each frame.

Figure 3 shows the preferred interpretations found for selected frames from each sequence. (Note that the selected frames do not necessarily match those shown in Figure 1.) For each sequence we show frames ordered from left to

[1] In the current system we consider only a single maximal contact set in which every admissible contact is added to the assertion set. Since there are no depth constraints in our layered model, this single contact hypothesis will not disallow any of the remaining assertions.

right.[2] While the preferred interpretations are often unique, at times there are multiple interpretations, particularly when objects interact. We highlight frames with multiple preferred interpretations by grey shading.

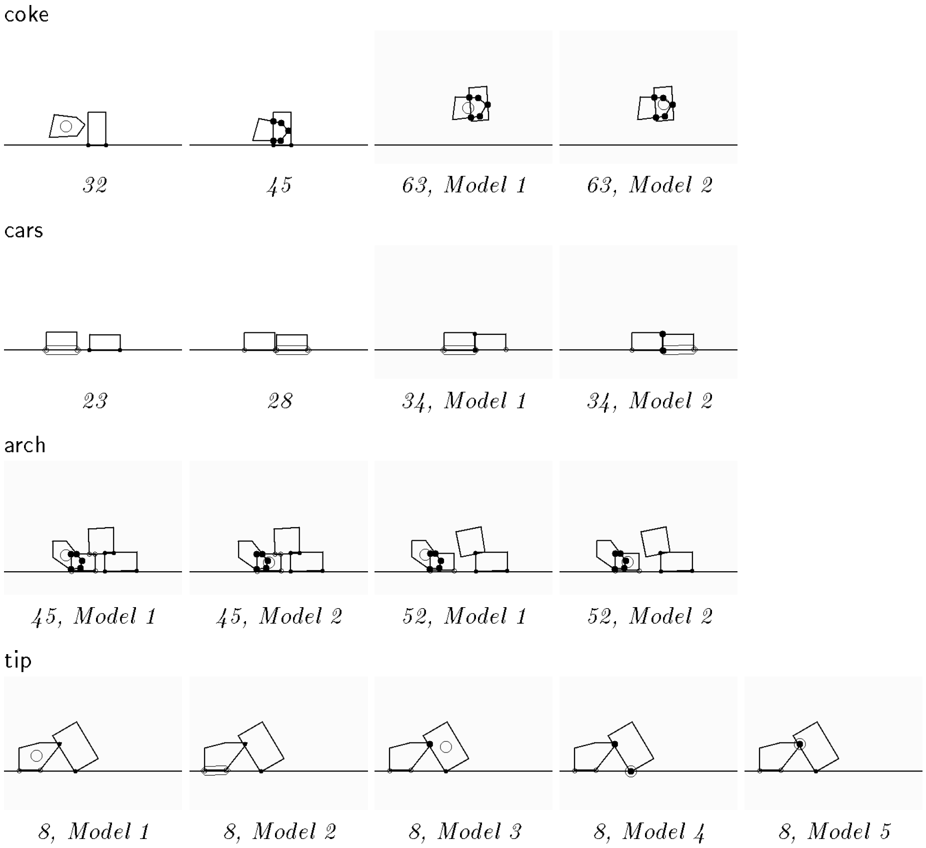

Fig. 3. Some preferred models for: coke, cars, arch, and tip. Frames with a non-unique maximally-preferred interpretation are shown with a grey background. We use the following symbols. For contacts, small disks denote contact points at rest while small circles denote sliding contacts. Larger disks denote attachment points. Motors are denoted by either a circle at the center of the object (BodyMotor), a circle around a contact point (AngularMotor), or by a closed curve around a contact region (LinearMotor).

[2] For the cars and arch sequences there is an ambiguity in the type of motors used, with body motors being interchangeable with linear motors (except on the hand). For clarity we show only linear motors in the cars sequence, and only body motors in the arch sequence.

Our machine interpretations are surprisingly intuitive. For example, the difference between models 1 and 2 in frame 63 of the coke sequence can be interpreted as the hand 'lifting' the can versus the can 'lifting' the hand. Similarly, the difference between models 1 and 2 in frame 34 of the cars sequence can be interpreted as the rear car 'pushing' the front car versus the front car 'pulling' the rear car. (Note that the system correctly hypothesises an attachment between the front and rear cars in the 'pulling' interpretation, but does not do so in the 'pushing' interpretation.) The third row of Figure 3 shows the interpretations for the arch sequence in which a hand removes the left block from an arch causing the top block to tip over. The system correctly infers that the top block is *supported* in frame 45, and *tipping* in frame 52, but is not able to determine whether the hand is 'pulling' the left block or whether the left block is 'carrying' the hand. Finally, the last row of Figure 3 shows the results for the tip sequence where a hand raises a box onto its corner and allows it to tip over. There are five interpretations corresponding to various assertions of an active hand, active box, and various types of linear and angular motors.

While encouraging, our current implementation exhibits a number of anomalies. These anomalies generally fall into three classes. The first problem is that because we consider single frames in isolation, in many cases the system cannot find unique interpretations. In particular, since the system does not have any prior information about the objects in the scene, it cannot rule out interpretations such as an active coke can lifting the passive hand in the coke sequence or an active block pulling a passive hand in the tip sequence. In addition, because of our preference for minimal sets of assertions, certain degenerate interpretations may occur. An example of this is shown in frame 45 of the coke sequence, where the hand is interpreted as a passive object (which is attached to the coke can). Since the system does not have any prior information about object properties and since it considers single frames in isolation, all of these interpretations are reasonable.

A second problem concerns the detection of collisions and changing contact relations between objects. In particular, when objects collide, the estimates for relative velocity and acceleration at their contact points may differ, resulting in the contact relation being deemed inadmissible. An example of this is shown in frame 28 of the cars sequence where the contact between the colliding cars is missed. Note that the acceleration of the cars should be equal (since they remain in contact after the collision), but the interpolator has smoothed over this discontinuity and given unreliable estimates of the acceleration.

Finally, a third problem occurs because we do not use a complete kinematic model, as mentioned in §2.1. An example of this problem is shown in the tip sequence in Figure 3. While all of the interpretations have a feasible force balance, the last three are not consistent with rigid-body motion since it is not *kinematically* feasible for the hand to be both attached to the box and in sliding contact with the table. Since our system considers only pairwise constraints between contacting objects, it does not check for global kinematic consistency. Further tests could be implemented to rule out these interpretations.

6 Conclusion

We have presented an implemented computational theory that can derive force-dynamic representations directly from camera input. Our system embodies a rich ontology that includes both kinematic and dynamic properties of the observed objects. In addition, the system provides a representation of uncertainty along with a theory of preferences between multiple interpretations.

While encouraging, this work could be extended in several ways. First, in order to work in a general environment, 3D representations are required. While our current system is able to represent 3D scenes provided it has suitable input, further work will be required to determine what type of 3D representation is suitable and how accurate the shape and motion information will have to be. Second, in order to deal with collisions and changing contact relations, a theory of impulses (transfer of momentum) will be required. Third, as indicated by the tip example, a more complete kinematic model is needed. Finally, in order to represent the structure of time-varying scenes, we require a representation of object properties and a method to integrate such information over multiple frames. We believe our current system provides the building blocks for such a representation, but additional work will be required to show how our ontology can be built into a more complex system.

Acknowledgments

The authors would like to thank Whitman Richards, Michael Black and Chakra Chennubhotla for helpful comments on this work.

References

1. Norman I. Badler. Temporal scene analysis: Conceptual descriptions of object movements. Technical Report 80, University of Toronto Department of Computer Science, February 1975.
2. David Baraff. Interactive simulation of solid rigid bodies. *IEEE Computer Graphics and Applications*, 15(3):63–75, May 1995.
3. M. Blum, A. K. Griffith, and B. Neumann. A stability test for configurations of blocks. A. I. Memo 188, M. I. T. Artificial Intelligence Laboratory, February 1970.
4. Gary C. Borchardt. Event calculus. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 524–527, Los Angeles, CA, August 1985.
5. Matthew Brand, Lawrence Birnbaum, and Paul Cooper. Sensible scenes: Visual understanding of complex scenes through causal analysis. In *Proceedings of the Eleventh National Conference on Artifical Intelligence*, pages 588–593, July 1993.
6. Scott Elliott Fahlman. A planning system for robot construction tasks. *Artificial Intelligence*, 5(1):1–49, 1974.
7. Roy Featherstone. *Robot Dynamics Algorithms.* Kluwer, Boston, 1987.
8. Brian V. Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13(3):201–230, May 1980.
9. Herbert Goldstein. *Classical Mechanics.* Addison-Wesley, second edition, 1980.

10. Katsushi Ikeuchi and T. Suehiro. Towards an assembly plan from observation, part i: Task recognition with polyhedral objects. *IEEE Transactions on Robotics and Automation*, 10(3):368–385, 1994.
11. Michael Jenkin and Allan D. Jepson. Detecting floor anomalies. In *Proceedings of the British Machine Vision Conference*, pages 731–740, York, UK, 1994.
12. Allan D. Jepson and Michael J. Black. Mixture models for optical flow. In *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition*, pages 760–761, 1993.
13. Leo Joskowicz and Elisha P. Sacks. Computational kinematics. *Artificial Intelligence*, 51(1–3):381–416, October 1991.
14. Yasuo Kuniyoshi and Hirochika Inoue. Qualitative recognition of ongoing human action sequences. In *IJCAI93*, pages 1600–1609, August 1993.
15. Vladimir Lifschitz. Computing circumscription. In *IJCAI85*, pages 121–127, 1985.
16. Richard Mann. PhD thesis, Department of Computer Science, University of Toronto. To appear.
17. Richard Mann, Allan Jepson, and Jeffrey Mark Siskind. The compuational perception of scene dynamics. 1996. Submitted.
18. Bernd Neumann and Hans-Joachim Novak. Event models for recognition and natural language description of events in real-world image sequences. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 724–726, August 1983.
19. Whitman Richards, Allan D. Jepson, and Jacob Feldman. Priors, preferences and categorical percepts. In Whitman Richards and David Knill, editors, *Perception as Bayesian Inference*. Cambridge University Press. To appear.
20. Jeffrey Mark Siskind. *Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, January 1992.
21. Jeffrey Mark Siskind. Axiomatic support for event perception. In Paul McKevitt, editor, *Proceedings of the AAAI-94 Workshop on the Integration of Natural Language and Vision Processing*, pages 153–160, Seattle, WA, August 1994.
22. Jeffrey Mark Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371–391, 1995.
23. John K. Tsotsos, John Mylopoulos, H. Dominic Covvey, and Steven W. Zucker. A framework for visual motion understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):563–573, November 1980.

This article was processed using the LaTeX macro package with ECCV'96 style