Visual Event Perception

Jeffrey Mark Siskind*
NEC Research Institute, Inc.
4 Independence Way
Princeton NJ 08540–6620 USA
609/951–2705
fax: 609/951–2483
Qobi@research.nj.nec.com
http://www.neci.nj.nec.com/homepages/qobi

March 15, 1999

Abstract

This paper presents a novel framework for training models to recognise simple spatial-motion events, such as those described by the verbs *pick up*, *put down*, *push*, *pull*, *drop*, *tip*, and *tap* and classifying novel observations into previously trained classes. Simple colour- and motion-based segmentation and tracking techniques are used to produce a time series of feature vectors constructed from the 2D object positions, orientations, shapes, and sizes. Hidden Markov models are trained on this time series data and used to classify novel occurrences into previously trained classes. The particular choice of features used allows the system to construct meaningful semantic representations of the event classes that it has learned.

KEYWORDS: Event classification, Motion analysis, Segmentation, Tracking, Learning, Hidden Markov models, Lexical semantics

1 Introduction

People can describe what they see. If I were to pick up a block and ask you what you saw, you could say *Jeff picked up the block*. In doing so, you describe both *objects*, like people and blocks, and *events*, like pickings up. Most recognition research in machine vision has focussed on recognising objects. In contrast, the system described in this paper is part of my on-going work on recognising events.

This paper describes an implemented system called HOWARD. HOWARD takes short (3–4 second) video clips as input and labels them with one of the seven event types *pick up*, *put down*, *push*, *pull*, *drop*, *tip*, and *tap*. HOWARD can also learn to recognise new events from a small number of examples of that event. More importantly, it can describe the meanings of the event types that it learns in pseudo-English. For example, it learns and knows that *pick up* involves a sequence

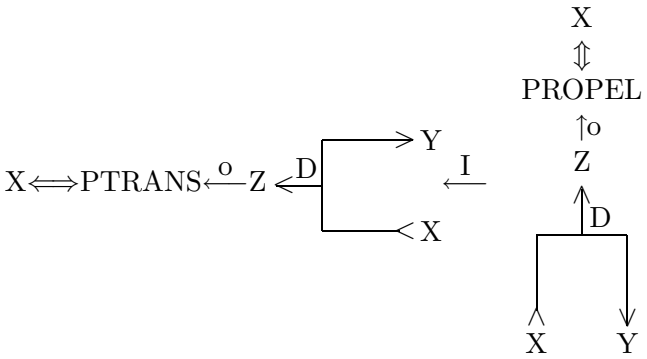
*This research was supported, in part, by a Samuel and Miriam Wein Academic Lectureship and by the Natural Sciences and Engineering Research Council of Canada. Part of this research was performed while the author was at the University of Toronto Department of Computer Science, the Technion Department of Electrical Engineering, and the University of Vermont Department of Electrical Engineering and Computer Science. Part of this work was done jointly with Quaid Morris and was reported in Siskind and Morris (1996).

of two actions: the hand moving downward towards the object being picked up, while the object being picked up is at rest, followed by the hand moving with the object upward away from the original location of the object. Examples of such learning and classification are presented later in this paper.

An important goal of my on-going work in event recognition is for systems to construct meaningful semantic representations of their knowledge. Hence the emphasis on producing pseudo-English descriptions of internal representations, not just on raw classification accuracy. Some may ask whether this work is more properly viewed as machine-vision research or as natural-language research. Or, for that matter, as knowledge representation, machine learning, or linguistics. I feel that such pigeon-holing is irrelevant. Rather, I believe that the link between language on one hand, and perception and action on the other hand, is the cornerstone of higher cognition. Understanding and modelling how we talk and reason about what we see and do is the key for us to understand our own minds and ultimately create artificial ones.

This work focuses on describing events. For the most part, we describe events using verbs. Most languages contains, perhaps, a few thousand verbs. Many of these describe mental states, like beliefs, goals, desires, feelings, and perceptions, or social interactions, like promises and betrayals. Modelling the meanings of such verbs in a fashion that is grounded in perception and action is clearly well beyond the scope of this work. This work limits itself to simple spatial-motion verbs: people physically interacting with objects in their environment. Often, however, people metaphorically extend the meanings of simple spatial-motion verbs to describe mental states and social interactions. We 'pick up' dates, 'put down' colleagues, 'push' people to do things, 'pull' the wool over our eyes, 'drop' the ball, 'tip' people off, and 'tap' people on the shoulder. Lakoff and Johnson (1980) have argued that, in fact, all human understanding is based on metaphorical extension of how we perceive our own bodies and their interaction with the physical world. Thus, while this work only attempts to describe simple spatial-motion verbs in non-metaphoric uses, the hope is that this will ultimately serve as the foundation for the larger scope of metaphoric reasoning.

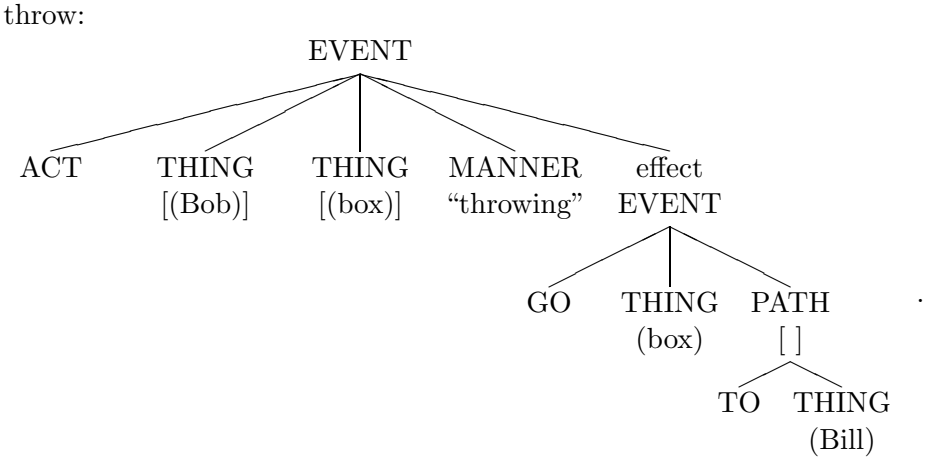
There is a long tradition of linguists describing the meanings of simple spatial-motion verbs. (See Siskind 1992, 1995 for a discussion of this.) Miller (1972) strings together primitives like **to apply force**, **by hand**, **to cause**, **to begin**, **to travel**, and **through air** to describe the meanings of verbs like *throw*. Schank (1973) uses a similar collection of primitives with a more graphical notation to describe the meaning of *throw*.



Jackendoff (1983) also uses a similar collection of primitives, but with a notation reminiscent of mathematical logic. He describes the meaning of *Beth threw the ball out the window* as

$$\text{CAUSE}(\textbf{Beth}, \text{GO}(\textbf{ball}, \text{OUT}(\textbf{window}))).$$

Pinker (1989) uses the same collection of primitives as Jackendoff, but with a tree-like notation that is more familiar to linguists. He describes the meaning of *Bob threw the box to Bill* as



This paper follows this tradition, extending it by attempting to ground many of the above primitives in visual input.

Linguists have observed that each event class has a set of characteristic participants. And linguists have given names to typical participants. For example, the AGENT is the participant doing the action. The PATIENT is the participant being acted upon. The SOURCE is the initial location of the PATIENT before the event takes place. And the GOAL is the final location of the PATIENT as a result of the event. Linguists have further observed an important characteristic of human language. To a large extent, the human conceptualization of the world treats events orthogonally from the objects that fill the roles of those events. Thus, a pick-up event is still a pick-up event regardless of whether one picks up a block, a ball, a pen, or a penny. And a block is still a block regardless of whether one picks it up, puts it down, pushes it, or pulls it. By and large, with the exception of incorporated instruments like *to hammer*, we don't have different nouns to denote picking up a block versus pushing a block and we don't have different verbs to denote picking up a block versus picking up a ball. This is important from the perspective of visual event perception. It means that one can recognise events without recognising objects. Thus, this work does not attempt to perform any object recognition. It does an extremely coarse segmentation of the input movies into participant objects and tracks their changing position, orientation, shape, and size over time, again at a very coarse level. It appears that such coarse segmentation and tracking is sufficient for the purpose of event recognition.

HOWARD represents each event type as a sequence of sub-events. Each sub-event prescribes the motion profile of the participant objects during that portion of the event. This motion profile is characterised using a collection of features that describe the relative and absolute motions of the participants. For example, HOWARD constructs and uses representations like the following for verbs like *pick up* and *drop*:

Pick Up	part-1	AGENT moves towards PATIENT
		PATIENT at rest above SOURCE
	part-2	AGENT and PATIENT move away from SOURCE
		SOURCE remains at rest
Drop	part-1	AGENT is close to PATIENT
	part-2	PATIENT moves downward away from AGENT

Note that the set of features used by HOWARD was chosen to be analogous to the semantic primitives used by Miller, Schank, Jackendoff, and Pinker.

Representing event types as sequences of sub-events with different motion profiles lends itself to a natural formulation using hidden Markov models. Each event class will correspond to a different model and each sub-event will correspond to a state within that model. HOWARD uses models that have continuous output-probability distributions. Segmentation and tracking procedures are used to produce a time series of object positions, orientations, shapes, and sizes from the input movies. From this, a time series of larger feature vectors is computed. The feature vector contains relative and absolute object positions and motions. The output-probability distributions of the hidden Markov models attempt to fit this feature-vector data. The feature vector currently contains two kinds of features: ones, like angular data, with finite range and ones, like velocity magnitude, with infinite range. The former are modelled with parametric Von-Mises distributions while the latter are modelled with Gaussian distributions. Because the feature vector contains a mixture of features with different distribution classes, and there is no analogue of a covariant multivariate Gaussian distribution for the Von-Mises distribution, the features are taken to be independent and the collective probability of the feature vector is taken to be the product of the probabilities of the individual features.

HOWARD currently learns event models in a supervised fashion, i.e. with labelled training data. A collection of movies, all of the same event type, is filmed and one hidden Markov model is fit to that data. Training is done with a variant of the Baum-Welch reestimation procedure (Baum, Petrie, Soules, & Weiss, 1970). Currently, two parameters to the training procedure must be provided manually: the number of states for the event class and the number of participant objects. All other parameters are learned automatically. Furthermore, since all of the events that are currently modelled have non-repetitive sequences of sub-events, the transition matrices of the hidden Markov models are constrained to be upper triangular. Once models are trained, new event observations are classified using the Viterbi procedure (Viterbi, 1967).

Figure 1 illustrates the structure of the HOWARD implementation. Input movies are processed by segmentation and tracking procedures to produce a time series of object positions, orientations, shapes, and sizes. The output of segmentation and tracking is fed to the training procedure to produce event models. After training, new movies are processed by the segmentation and tracking procedures and fed, along with the event models, to the classification procedure. The particular choice of segmentation and tracking procedures is largely irrelevant. Any procedure that provides this information is suitable, since this phase operates independently of subsequent processing. Over time, numerous different segmentation and tracking procedures have been integrated into HOWARD. This paper presents just one particular pair of procedures that is robust but special purpose. This particular pair of procedures is presented here because it was the one used to conduct the training and classification experiments described later in this paper.

2 Segmentation

Segmentation constitutes the first stage of processing during event recognition. The input to the segmentation procedure consists of a sequence of frames, each frame being a colour image. The output of the segmentation procedure is a list of approximate 2D positions, orientations, shapes, and sizes of the participant objects for each frame of the movie. Currently, the position, orientation, shape, and size information is represented as a set of ellipses, one ellipse centered on each object in each frame. Each ellipse is specified by five parameters: the x and y coordinates of its centre, the angle of orientation of its major axis, its area, and its eccentricity. Figure 2 shows several key

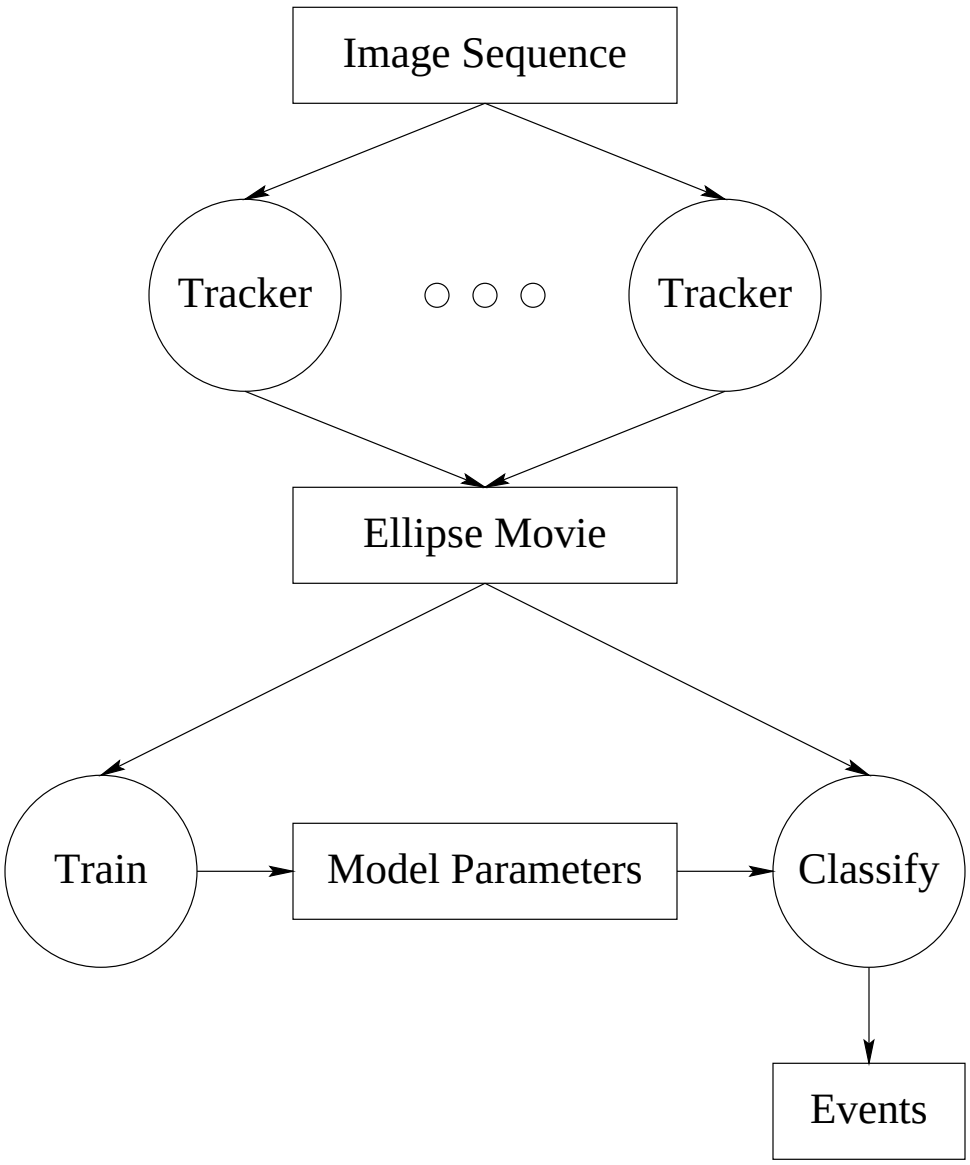


Figure 1: The event-recognition architecture used by HOWARD.

frames for each of seven movies. Figure 3 shows the output of the segmentation procedure for each of the corresponding frames in figure 2.

A very simple special-purpose approach to segmentation is used. There are two separate segmentation channels, one based on colour and one based on motion. These channels are largely independent until their output is combined. Both colour and motion are used because each alone is insufficient to segment all of the participants all of the time. In the movies used to train and test the event-recognition procedure, the hand does not have a particular fixed colour and thus must be segmented using motion. Likewise, in many of the events, the block moves only for a portion of the event, i.e. only after it is picked up or before it is put down, and thus must be segmented using colour.

The colour channel labels each pixel in each frame of each movie as either 'red,' 'green,' 'blue,' 'yellow,' or 'other.' Red, green, blue, and yellow pixels are defined as those having RGB values within specified non-overlapping rectangular regions of RGB space. Since this process is noisy, giving false negatives and false positives, a region grower is used to filter the output of the colour matcher. This region grower is applied separately for each of the four colour channels. The region grower forms a graph with pixels of a given colour as vertices and with edges between every pair of pixels that are less than a specified distance apart. The region grower finds connected components in this graph. Connected components with fewer than a specified number of vertices are discarded as noise, while the large connected components are taken as objects. An ellipse is fit to each object by computing the mean and the covariance matrix of the 2D coordinates of the pixels for that object. The mean is taken as the centre of the ellipse and the eigenvectors of the covariance matrix are taken as the axes of the ellipse.

The motion channel operates in a similar fashion to the colour channel. The Euclidean distance in RGB space between corresponding pixels in adjacent frames is thresholded to label each pixel in each frame as 'moving' or 'stationary.' Since this process is also noisy, the same region grower is applied to cluster the moving pixels into connected components, again discarding components with fewer than a specified number of pixels. And again, an ellipse is fit to each remaining connected component.

Two slight complications arise with this simple process. First, a given pixel might be labelled as both coloured and moving. This would lead to two closely aligned ellipses being generated for a single object. To solve this problem, coloured moving pixels are counted only as coloured pixels. Second, during some events, like *pick up* and *put down*, the hand stops moving for a brief period, just as it grasps or releases the object, and the motion channel fails to detect the hand during that period. To solve this problem, the motion channel detects when the number of moving pixels in a frame falls below a specified threshold and copies the moving objects from the previous frame. In essence, this applies the intuitive notion that an object that stops moving remains in the same place as where it was last seen. I call this process *conjuring*.

This colour- and motion-based segmentation procedure is extremely simple yet special purpose. It was selected over more sophisticated and general-purpose alternatives (e.g. Weiss & Adelson, 1995, Black & Jepson, 1996, Cox, Rao, & Zhong, 1996, and Shi & Malik, 1997 for several reasons. First, it is robust. The experiments reported later in this paper required segmenting and tracking 210 movies comprising $25,200$ frames with no manual intervention. Of these, only 9 movies needed to be discarded due to poor segmentation. I know of no current general-purpose segmentation procedure that can process that much data with that level of reliability without manual intervention. Second, it is fast. A three-second movie comprising 90 frames can be processed in about a minute of CPU time on a standard personal computer. Fast processing is crucial to tuning the event training and classification procedures by repeated experimental trials. Finally, segmen-

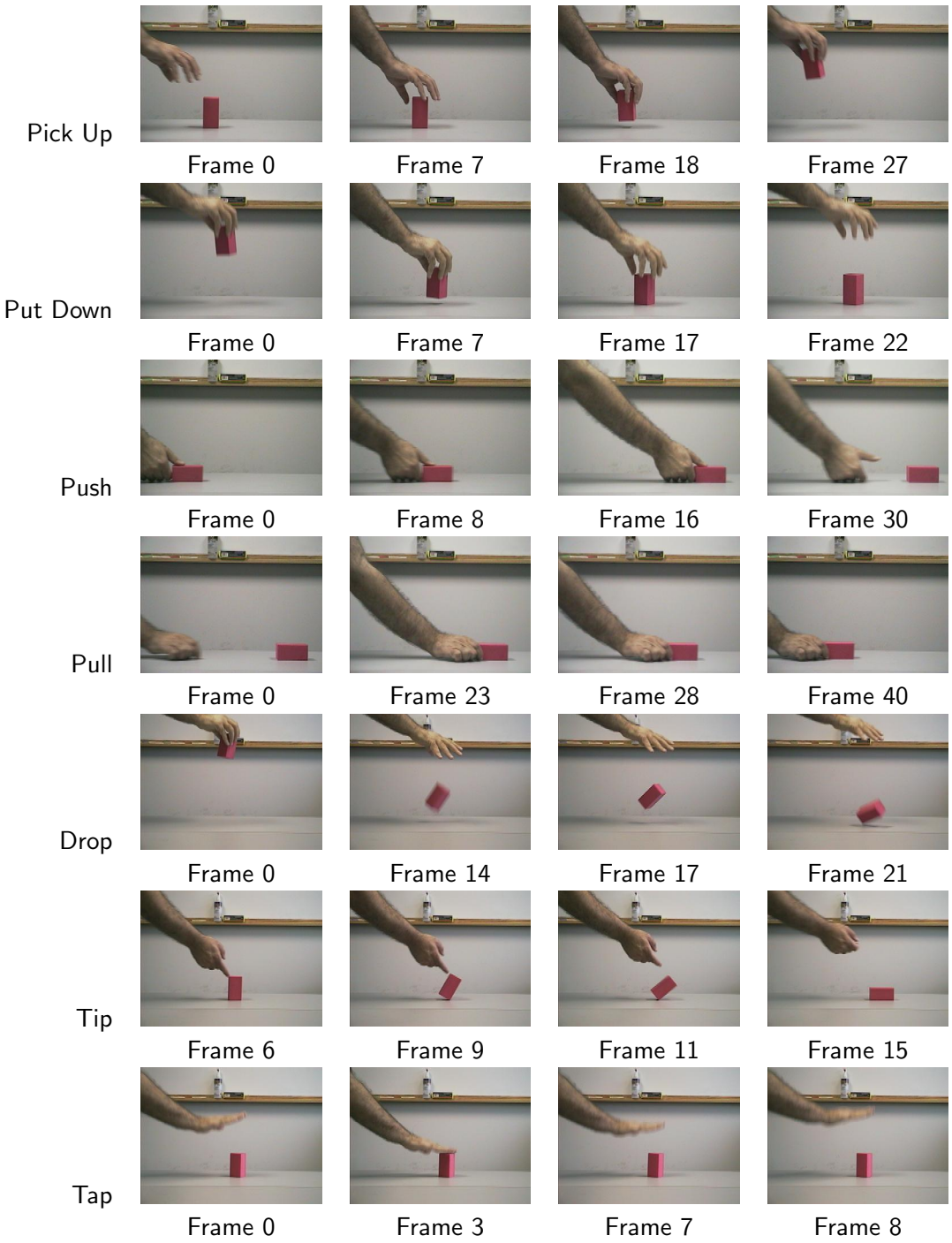


Figure 2: Sample frames from several movies depicting seven different event types.

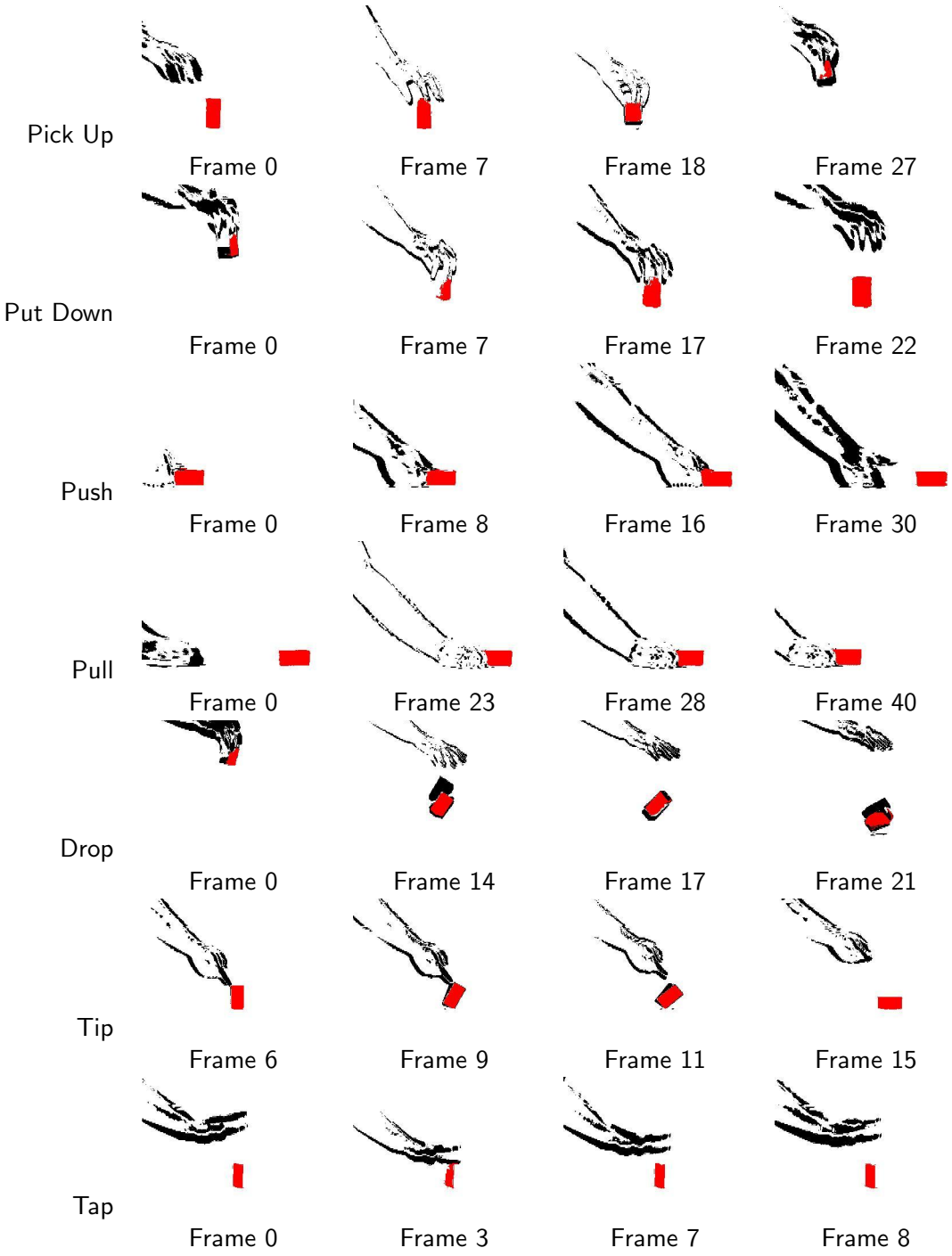


Figure 3: The output of the segmentation procedure on the frames from the movies in figure 2.

tation is a rich research area in its own right. I chose to focus my efforts on event recognition and spend as little effort as possible on segmentation. Since my work is largely orthogonal to segmentation, the event-recognition techniques described here could easily be coupled with alternate approaches to segmentation.

3 Tracking

The procedure described above just performs segmentation. It does not track objects. It simply associates a set of ellipses with each frame of each movie. It does not produce a correspondence between the ellipses of consecutive frames. In fact, there is even no guarantee that it will produce the same number of ellipses from frame to frame. The number of ellipses detected can vary both for legitimate reasons, like occlusions and entering or leaving the field of view, and for illegitimate ones, like noise, that result in missing or spurious ellipses. Thus, the output of the segmentation procedure is post-processed to compute a correspondence between the ellipses in adjacent frames of each movie. This correspondence is called the *internal correspondence*, in contrast to a later correspondence called external, because it relates ellipses within a single movie. A by-product of computing this internal correspondence is the elimination of spurious ellipses and interpolation of missing ellipses.

A simple method is used to compute the internal correspondence. Two ellipses e_1 and e_2 from adjacent frames are associated if e_1 is the closest ellipse to e_2, among all of the ellipses in its frame, e_2 is the closest ellipse to e_1, among all of the ellipses in its frame, and the distance between e_1 and e_2 is less than some specified threshold. A weighted, five-dimensional Euclidean distance metric is used for this purpose. This association criterion induces a partial one-to-one correspondence between ellipses in adjacent frames and collects ellipses into chains that span the frames of the movie. Some chains will be short and span only a few frames. Others will be long and span most, or all, of the movie. Long chains usually correspond to participant objects. Short chains arise either because of spurious ellipses or from long chains that are broken into smaller pieces because of missing ellipses. An attempt is made to connect together such shorter broken chains. Whenever the endpoints of two chains differ in both space and time by at most some specified limit, the chains are connected together by interpolating the missing ellipses of the gap region or splicing out the extra ellipses of the overlap region. Similarly, chains that come close to, but do not reach, the beginning and end of the movie are extended by the same interpolation process. As a result of such chain merging and extension, chains that span the entire movie usually correspond to legitimate tracked objects, while short chains are discarded as corresponding to noisy output of the segmenter. Figure 4 shows the output of the internal-correspondence procedure on the segmentations shown in figure 3.

4 The Feature Vector

The output of the internal-correspondence procedure consists of a set of sequences of ellipse parameters for each movie. In essence, this is a sequence of $5k$ parameters per frame, where k is the number of ellipse chains. These $5k$ parameters are used to compute a much larger feature vector that is more relevant for differentiating the different event classes. This feature vector contains both relative and absolute features, i.e. features that are computed for each ellipse as well as features that are computed for every pair of ellipses. The following features are computed:

- Absolute Features

    1. the magnitude of the velocity vector of the centre of each ellipse

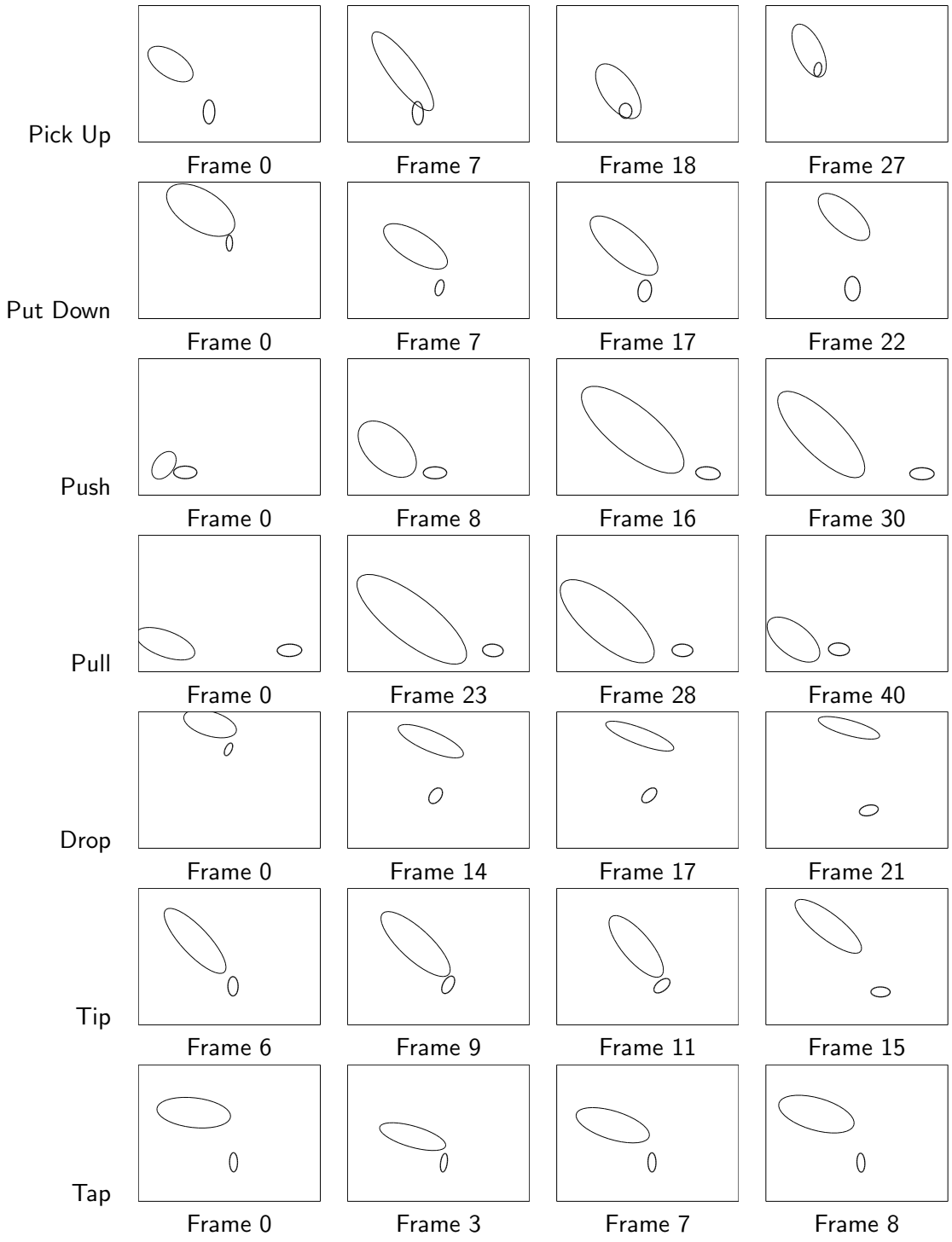


Figure 4: The output of the internal-correspondence procedure on the segmentation data in figure 3.

2. the orientation of the velocity vector of the centre of each ellipse
3. the angular velocity of each ellipse
4. the first derivative of the area of each ellipse
5. the first derivative of the eccentricity of each ellipse
6. the derivatives of each of the above five features

- Relative Features

1. the distance between the centres of every pair of ellipses
2. the orientation of the vector between the centres of every pair of ellipses
3. the difference between the orientations of the major axes of every pair of ellipses
4. for every pair of ellipses, the difference between the orientation of the major axis of the first ellipse and the orientation of a vector from the centre of that ellipse to the centre of the second ellipse
5. the derivatives of each of the above four features

This vector contains $5k^2 + 5k$ entries when there are k ellipse chains.

These features were chosen because they represent the kind of information that is latent in the semantic primitives used by Miller (1972), Schank (1973), Jackendoff (1983), Pinker (1989), and others to represent the meanings of simple spatial-motion verbs. For example, the magnitude and orientation of the velocity of an ellipse tells whether an object is moving, whether it is moving quickly or slowly, and whether it is moving upward, downward, leftward, or rightward. The angular velocity tells whether an object is rotating clockwise or counterclockwise in the image plane. The first derivative of the area tells whether an object is moving towards or away from the observer. The first derivative of the eccentricity tells whether an object is rotating outside of the image plane. The distance between the ellipse centres tells whether objects are close together or far apart. The derivative of this distance tells whether one object is moving towards or away from another object. The orientation of the vector between ellipse centres tells whether one ellipse is above, below, to the left of, or to the right of another ellipse. And the relative orientation of two ellipses tells whether one object is facing towards or away from another object. Thus, these features constitute quantitative analogues of the same qualitative semantic primitives that are typically used to represent verb meanings.

5 External Correspondence

As discussed earlier, each event class has a number of roles that are filled with objects that participate in the event. This gives rise to an additional correspondence during training and classification: how does one assign observed objects to the different roles? Linguistically, thematic roles have a certain generic character as indicated by their traditional name. For example, the AGENT denotes the doer of the action while the PATIENT denotes the participant that is acted upon. Determining, from visual input, which participant is the doer or which is acted upon is a difficult problem and beyond the scope of this work. (Though see Mann, Jepson, & Siskind, 1997 and Mann & Jepson, 1998 for some approaches to this problem.) So the work described here solves a much more limited problem. It simply determines whether two participants are playing the same role in a given event type without determining what that role is. Thus, it computes a correspondence between ellipse chains in different movies of the same event type, hence the name *external correspondence*.

The external-correspondence problem arises both during training and classification. During classification, the problem is mapping the observed objects to the roles of a trained event model. During training, the problem is finding a correspondence between the observed objects across the different training movies for a given event model. Both cases are complicated by the fact that there may be additional observed objects in some or all of the movies that are irrelevant to the event in question. So the external-correspondence problem actually requires selecting a subset of observed objects in addition to matching that subset to the roles of the event.

Solving the external-correspondence problem optimally during training is computationally intractable. Finding the correspondence that allows the best-fit model, given n training movies, each with l observed objects, for an event with k roles requires selecting the best match from $[l!/(l-k)!]^{n-1}$ different possible correspondences. Thus, the current system takes a heuristic greedy approach. It selects the first two training movies, examines all $l!/(l-k)!$ possible correspondences between the objects in those two movies, applies the Baum-Welch training procedure to each correspondence, and selects the one that gives the best likelihood. It then fixes this correspondence between the first two movies and examines all $l!/(l-k)!$ possible correspondences between these first two movies and the third movie. And so on for all of the movies. This results in examining only $(n-1)[l!/(l-k)!]$ different correspondences. Since, in practice, both k and l are fixed and small, this yields a training procedure that is linear in the number of training movies instead of exponential. Furthermore, in practice, this technique very often yields the correct correspondence when compared with ground truth.

The current system enhances the external correspondence yielded by the above approach with an iterative refinement technique. After computing the initial external correspondence, it cycles through each of the n training movies and reconsiders all $l!/(l-k)!$ possible correspondences between that training movie and the remaining training movies, while keeping the correspondence among the remaining training movies fixed. It repeats such cycles until no further increase in the likelihood of the whole training set is obtained. Each such pass considers only $n[l!/(l-k)!]$ correspondences so is also linear in the number of training movies. In practice, only a small number of passes is required until convergence. Furthermore, in practice, this technique, when combined with the previous technique, almost always yields the correct correspondence when compared with ground truth.

Solving the external-correspondence problem during classification is much easier. Since only one movie is involved, all $l!/(l-k)!$ possible correspondences can be considered. Furthermore, since the Viterbi classification procedure is much less costly than the Baum-Welch training procedure, applying the Viterbi procedure to each possible correspondence and selecting the best match is not overly costly.

6 Experimental Results

To test the approach to event classification discussed in this paper, I selected 7 different event types: *pick up*, *put down*, *push*, *pull*, *drop*, *tip*, and *tap*. I filmed 3 different subjects performing each of the 7 event types under each of 2 conditions: from the left side and from the right side. Each subject–event-type–condition triple was filmed 5 times for a total of 210 movies.

The movies were filmed with a Canon VC-C3 camera connected to a non-PPB Matrox Meteor/RGB frame grabber. The movies were filmed at 320×240 resolution, at 24 bits per pixel colour and 30 frames per second, and compressed off-line to disk in JPEG using the `libjpeg` software. Each movie was 4 seconds (120 frames) long, for a total of $25,200$ images.

Each of the 210 movies was processed by the segmentation and tracking procedures described

earlier. Of the 210 movies, 9 were manually discarded due to poor tracking. Of the remaining 201 movies, 140 were randomly selected as training movies and the remaining 61 were left as unseen test movies. The training set consisted of 10 movies for each event type and condition. Fourteen 2-state hidden Markov models were trained, one for each event type and condition. Thus, there were two hidden Markov models per event type, one for each condition. For this experiment, the external correspondence was specified manually.

The classification procedure was applied to the training set, this time computing external correspondence automatically. 135 out of the 140 training movies (96%) were correctly classified. The classification procedure was also applied to the unseen test set, again computing external correspondence automatically. 47 out of the 61 test movies (77%) were correctly classified. Note that this is significantly better than chance, which would yield 14% correct classification.

In addition to the above experiments in controlled surroundings, the system has been demoed to live audiences on three occasions outside of my office, two of them off site. And the system has been demoed live in my office to visitors on numerous occasions. Informal assessment of the recognition rate across these demos is around 75%. Figure 5 shows a screen dump of the program after one such successful recognition.

The training procedure has also been demoed live, both off site and in my office. In this demo, three instances of a new event type are filmed and a new hidden Markov model is fit to these movies, computing the external correspondence automatically. Additionally, the hidden Markov model that is produced by the training procedure is described qualitatively, in pseudo-English, by quantising the means and variances of the output probabilities for the states of the model and associating different ranges of values with different keywords. The following keywords are used to describe the models:

- Absolute Features
  1. the magnitude of the velocity vector of the centre of each ellipse
     - `is moving`
  2. the orientation of the velocity vector of the centre of each ellipse
     - `leftward`, `rightward`, `upward`, `downward`
  3. the angular velocity of each ellipse
     - `is rotating`
  4. the first derivative of the area of each ellipse
     - `is growing`, `is shrinking`
  5. the first derivative of the eccentricity of each ellipse
     - `is becoming more flat`, `is becoming more round`
  6. the derivatives of each of the above five features
     - `is accelerating`
- General Modifiers
  - `not`, `not known`, `slowly`, `quickly`, `clockwise`, `counterclockwise`
- Relative Features
  1. the distance between the centres of every pair of ellipses

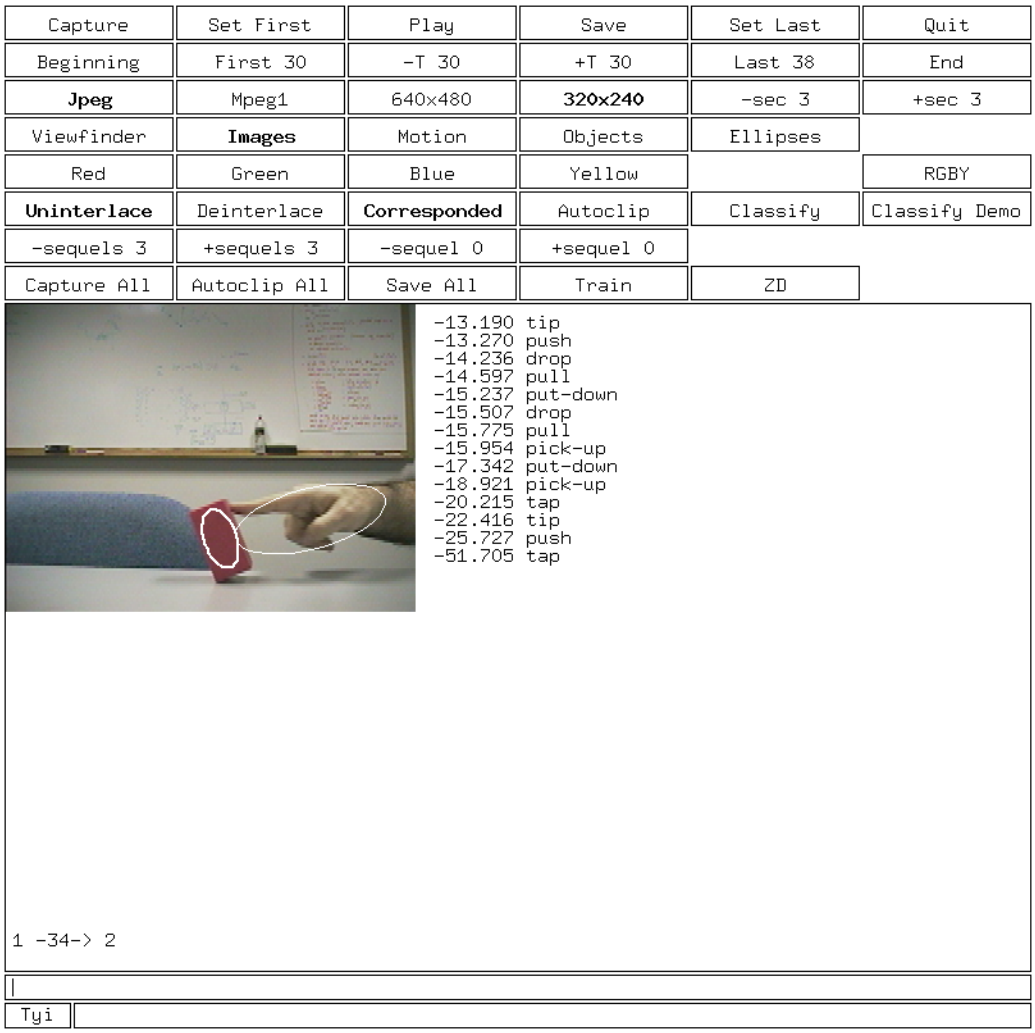


Figure 5: A screen dump of the classification program after it has successfully recognised a *tip* event.

   - `is far from`, `is close to`

2. the orientation of the vector between the centres of every pair of ellipses
   - `above`, `below`, `to the left of`, `to the right of`
3. the difference between the orientations of the major axes of every pair of ellipses
4. for every pair of ellipses, the difference between the orientation of the major axis of the first ellipse and the orientation of a vector from the centre of that ellipse to the centre of the second ellipse
   - `facing towards`, `facing away from`
5. the derivatives of each of the above four features
   - `is moving towards`, `is moving away from`
   - `is moving around`

Figure 6 shows a screen dump of the program after it has been trained on three instances of a *pick up* event. Note that it has produced a reasonable explanation of what constitutes a *pick up* event.

7 Conclusion

Visual event recognition is a growing area of research interest. And there have been a number of recent attempts to use hidden Markov models to classify visual events. Among them, Yamoto, Ohya, and Ishii (1992) describes a system that classifies tennis strokes using discrete-output hidden Markov models. They separate the agent from the background using background subtraction, subsample and vector quantise each image in the image sequence, and train and classify the resulting codebook-index sequence. Starner (1995) describes an ASL recognition system that uses a colour blob tracker to place ellipses around the hands of a subject signing while wearing coloured dish gloves behind a diffuse screen and uses hidden Markov models to recognise ASL given the ellipse-parameter sequences. Bobick and Ivanov (1998) also use hidden Markov models to classify human body motion.

While there are numerous minor differences between these systems and the one described in this paper, both in intention and in methods, there is one overriding fundamental difference. Unlike the related work, the system described in this paper attempts to build meaningful semantic representations of the event types that it has learned. We know from linguistics that most events have multiple participants. Thus, it appears necessary to track multiple objects to recognise events. And we know that most simple spatial-motion events involve changing spatial relations between those objects. Thus, it appears necessary for the feature vector to contain relative features and their temporal derivatives. Furthermore, the recurring collection of qualitative semantic primitives used by many researchers to describe the semantics of simple spatial-motion verbs has motivated the particular choice of quantitative features that I have chosen to incorporate in the feature vector used in the system described in this paper. The ability of the system to produce reasonably coherent pseudo-English descriptions of the events that it has learned, as shown in figure 6, gives evidence that this approach has promise.

The interface between language and vision is the cornerstone of higher cognition. Fundamentally, humans use language for three generic purposes: to describe what we see, to describe what to do, and to convince others to hold our beliefs. The ability to describe what we see allows a community to share the collective eyes of its individuals. The ability to describe what to do allows a community to share the collective limbs of its individuals. The ability to convince others

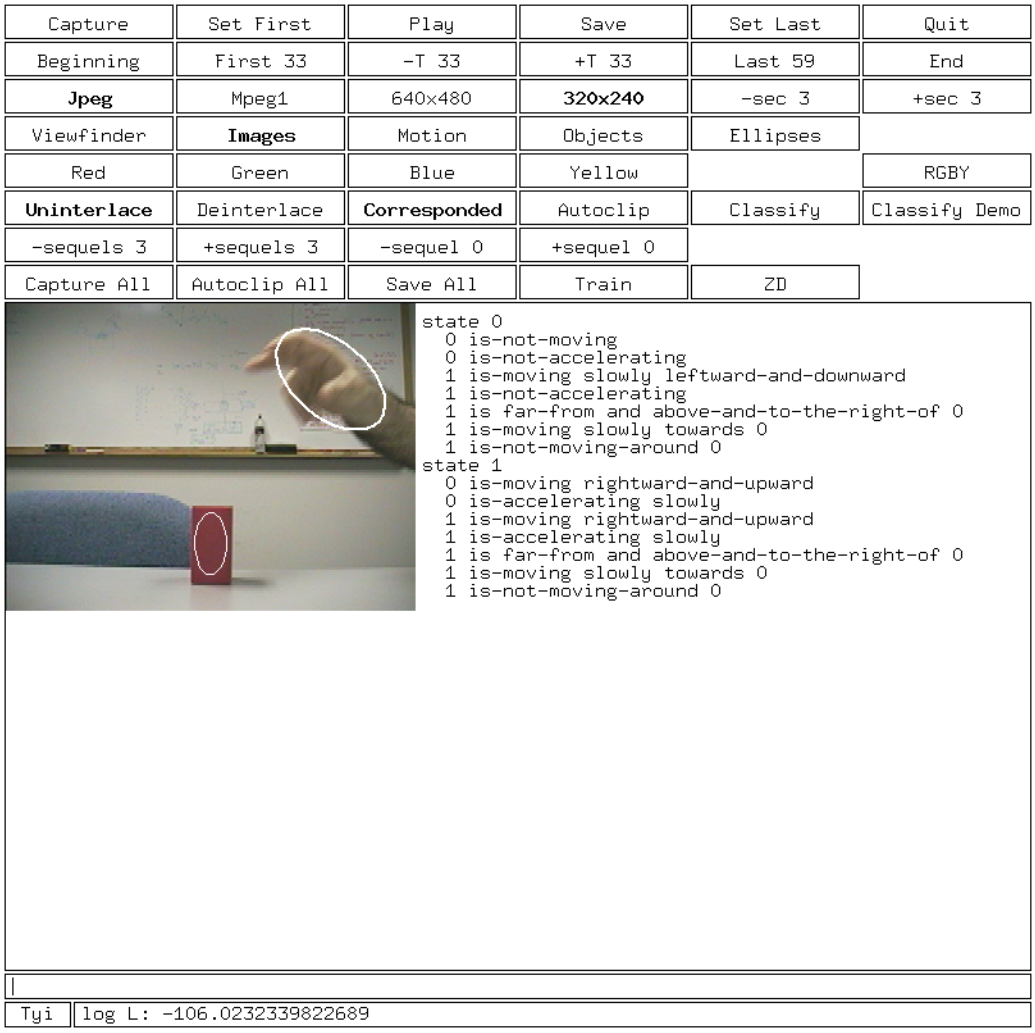


Figure 6: A screen dump of the training program after it has been trained on three instances of a *pick up* event.

to hold our beliefs allows a community to share the collective minds of its individuals. The first two intimately tie language to perception and motor skills. Only the latter involves disembodied language. It is reasonable to assume that language evolved primarily to serve the first two goals, with the latter being a later, parasitic development. It is more important for immediate survival to be able to say *Watch out for the lion!* to a hearer who cannot see it, or to say *Lift the rock off my leg!* when one is caught and incapable of doing it oneself, than to prove a theorem, book an airline flight, or search the Web for citations or pretty pictures. Yet, almost all of artificial intelligence research for the past forty years has focussed on such disembodied knowledge and use of language.

I believe that to understand intelligence, and ultimately build artificially intelligent systems, we will need to build meaningful semantic representations. And those representations will need to be grounded in the physical world around us. Via perception and manipulation. The research reported in this paper is one modest attempt in that direction.

References

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, *41*(1), 164–171.

Black, M. J., & Jepson, A. D. (1996). EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. In *Proceedings of the Fourth European Conference on Computer Vision*, pp. 329–342, Cambridge, UK.

Bobick, A. F., & Ivanov, Y. A. (1998). Action Recognition Using Probabilistic Parsing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 196–202, Santa Barbara, CA.

Cox, I. J., Rao, S. B., & Zhong, Y. (1996). Ratio Regions: A Technique for Image Segmentation. In *Proceedings of the International Conference on Pattern Recognition*, pp. 557–564, Santa Barbara, CA.

Jackendoff, R. (1983). *Semantics and Cognition*. Cambridge, MA: The MIT Press.

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. The University of Chicago Press.

Mann, R., & Jepson, A. D. (1998). Toward the Computational Perception of Action. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 794–799, Santa Barbara, CA.

Mann, R., Jepson, A. D., & Siskind, J. M. (1997). The Computational Perception of Scene Dynamics. *Computer Vision and Image Understanding*, *65*(2).

Miller, G. A. (1972). English Verbs of Motion: A Case Study in Semantics and Lexical Memory. In A. W. Melton & E. Martin (Eds.), *Coding Processes in Human Memory*, chap. 14, pp. 335–372. Washington, DC: V. H. Winston and Sons, Inc.

Pinker, S. (1989). *Learnability and Cognition*. Cambridge, MA: The MIT Press.

Schank, R. C. (1973). The Fourteen Primitive Actions and Their Inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory.

Shi, J., & Malik, J. (1997). Motion Segmentation and Tracking Using Normalized Cuts. Technical report CSD-97-962, University of California, Berkeley.

Siskind, J. M. (1992). *Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Siskind, J. M. (1995). Grounding Language in Perception. *Artificial Intelligence Review*, *8*, 371–391.

Siskind, J. M., & Morris, Q. (1996). A Maximum-Likelihood Approach to Visual Event Classification. In *Proceedings of the Fourth European Conference on Computer Vision*, pp. 347–360, Cambridge, UK: Springer-Verlag.

Starner, T. E. (1995). Visual Recognition of American Sign Language Using Hidden Markov Models. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymtotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, *13*, 260–267.

Weiss, Y., & Adelson, E. H. (1995). Perceptually organized EM: A framework for motion segmentation that combines information about form and motion. Tech. rep. 315, MIT Media Lab Vismod.

Yamoto, J., Ohya, J., & Ishii, K. (1992). Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model. In *Proceedings of the 1992 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–385. IEEE Press.