Fall 2013

Reasoning Across Language and Vision in Machines and Humans

Andrei Barbu
Purdue University

Follow this and additional works at: http://docs.lib.purdue.edu/open_access_dissertations

Part of the Computer Sciences Commons, Neuroscience and Neurobiology Commons, and the Robotics Commons

Recommended Citation

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
This is to certify that the thesis/dissertation prepared

By  Andrei Barbu

Entitled  Reasoning Across Language and Vision in Machines and Humans

For the degree of  Doctor of Philosophy

Is approved by the final examining committee:

Jeffrey M. Siskind  
Chair

Robert L. Givan

Thomas M. Talavage

Anthony G. Cohn

To the best of my knowledge and as understood by the student in the Research Integrity and Copyright Disclaimer (Graduate School Form 20), this thesis/dissertation adheres to the provisions of Purdue University’s “Policy on Integrity in Research” and the use of copyrighted material.

Approved by Major Professor(s): Jeffrey Mark Siskind

Approved by: Venkataramanan (Ragu) Balakrishnan  11/18/2013
Head of the Graduate Program  Date
REASONING ACROSS LANGUAGE AND VISION

IN MACHINES AND HUMANS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Andrei Barbu

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

December 2013

Purdue University

West Lafayette, Indiana
ACKNOWLEDGMENTS


This work was supported, in part, by NSF grant CCF-0438806, by the Naval Research Laboratory under Contract Number N00173-10-1-G023, by the Army Research Laboratory accomplished under Cooperative Agreement Number W911NF-10-2-0060, and by computational resources provided by Information Technology at Purdue through its Rosen Center for Advanced Computing. Any views, opinions, findings, conclusions, or recommendations contained or expressed in this document or material are those of the author(s) and do not necessarily reflect or represent the views or official policies, either expressed or implied, of NSF, the Naval Research Laboratory, the Office of Naval Research, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xii</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 GAME LEARNING</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Our Custom Robot</td>
<td>6</td>
</tr>
<tr>
<td>2.2 The Space of Games Considered</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Game-State Reconstruction</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Rule-Independent Piece Manipulation</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Learning Game Rules by ILP</td>
<td>15</td>
</tr>
<tr>
<td>2.6 Results</td>
<td>17</td>
</tr>
<tr>
<td>2.7 Comparison With Related Work</td>
<td>18</td>
</tr>
<tr>
<td>2.7.1 Game-Rule Learning</td>
<td>19</td>
</tr>
<tr>
<td>2.7.2 Game-State Reconstruction from Visual Input</td>
<td>21</td>
</tr>
<tr>
<td>2.7.3 Robotic Manipulation of Board-Game Hardware</td>
<td>22</td>
</tr>
<tr>
<td>2.8 Conclusion</td>
<td>23</td>
</tr>
<tr>
<td>3 VIDEO IN SENTENCES OUT</td>
<td>26</td>
</tr>
<tr>
<td>3.1 The mind’s eye corpus</td>
<td>29</td>
</tr>
<tr>
<td>3.2 Overall system architecture</td>
<td>31</td>
</tr>
<tr>
<td>3.2.1 Object detection and tracking</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2 Body-posture codebook</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 Event classification</td>
<td>37</td>
</tr>
<tr>
<td>3.2.4 Generating sentences</td>
<td>42</td>
</tr>
<tr>
<td>3.3 Experimental results</td>
<td>47</td>
</tr>
</tbody>
</table>
3.4 Conclusion ............................................. 48
4 SIMULTANEOUS OBJECT DETECTION, TRACKING, AND EVENT RECOGNITION ................................. 50
  4.1 Detection-Based Tracking ................................ 53
  4.2 Evaluation of Detection-Based Tracking ...................... 58
  4.3 Combining Object Detection and Tracking ....................... 60
  4.4 Combining Tracking and Event Recognition ...................... 63
  4.5 Combining Object Detection, Tracking, and Event Recognition ... 66
  4.6 Demonstration ............................................. 67
  4.7 Related Work ............................................. 70
  4.8 Conclusions and Future Work ................................ 72
5 SAYING WHAT YOU’RE LOOKING FOR: LINGUISTICS MEETS VIDEO SEARCH ........................................ 75
  5.1 Tracking ...................................................... 78
  5.2 Word recognition ............................................ 80
  5.3 Sentence tracker .............................................. 82
  5.4 Retrieval ...................................................... 90
  5.5 Results ....................................................... 93
    5.5.1 The new corpus ........................................ 93
    5.5.2 Ten westerns ........................................... 94
    5.5.3 Comparison ............................................. 99
  5.6 Discussion ..................................................... 100
  5.7 Conclusion .................................................... 106
6 THE COMPOSITIONAL NATURE OF VERB AND ARGUMENT REPRESENTATIONS IN THE HUMAN BRAIN ............................ 108
  6.1 Compositionality ............................................. 110
  6.2 Approach ..................................................... 112
    6.2.1 fMRI procedures ...................................... 112
    6.2.2 fMRI processing ....................................... 113
<table>
<thead>
<tr>
<th>Chapter/Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Experiment 1: Verb Representation</td>
<td>114</td>
</tr>
<tr>
<td>6.4</td>
<td>Experiment 2: Argument Representation</td>
<td>119</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusion</td>
<td>123</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>LIST OF REFERENCES</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>VITA</td>
<td>140</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.1</td>
<td>The vocabulary used to generate sentential descriptions of video.</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>Trained models for object classes and their mappings to (a) nouns, (b) restrictive adjectives, and (c) size adjectives.</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Sentential templates for the action classes indicated in bold.</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>(a) The number of videos in common, (b) the mean overlap, and (c) the standard deviation in overlap between each pair of annotation sources.</td>
<td>59</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Learning physically-instantiated game play through visual observation. Two robotic agents, the <strong>protagonist</strong> and <strong>antagonist</strong>, play a board game like Chess. A third robotic agent, the <strong>wannabe</strong>, does not know the rules of the game but must infer the rules by visually observing the play of the <strong>protagonist</strong> and <strong>antagonist</strong>. The <strong>wannabe</strong> must then use these rules for further physically-instantiated play.</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>One of our three novel custom robots designed specifically to support play of physically-instantiated board games. Note the 5 DOF arm with two independently-controllable fingers mounted on the upper level of a two-level housing that serves as the game-play surface. Also note the binocular pan-tilt cameras mounted on a pendulum arm that can pivot 180° around the game-play surface by being mounted on the lower level of the housing.</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>As demonstrated in the video on our website, our system is able to play a physically-instantiated board game given rule specifications, such as these for Tic Tac Toe, in a subset of English.</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Background knowledge encoded in Prolog. Progol-specific settings and mode, type, and pruning declarations have been omitted.</td>
<td>24</td>
</tr>
<tr>
<td>2.5</td>
<td>Rules for two of the six games discussed in Section 2.2 learned automatically from visual observation of autonomous physically-instantiated game play.</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>The overall architecture of our system for producing sentential descriptions of video.</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Sample clusters from our body-posture codebook.</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>ROC curves for each of the 48 action classes for Experiment I omitting all discrete and body-posture-related features.</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>ROC curves for each of the 48 action classes for Experiment II omitting only the discrete features.</td>
<td>40</td>
</tr>
<tr>
<td>3.5</td>
<td>ROC curves for each of the 48 action classes for Experiment III omitting only the continuous body-posture-related.</td>
<td>41</td>
</tr>
</tbody>
</table>
3.6 (a) Endogenous and (b) exogenous prepositional-phrase adjuncts to describe subject motion direction. (c) Prepositional phrases incorporated into subject noun phrases describing viewer-relative 2D spatial relations between the subject X and the reference object Y.

3.7 Key frames from four videos in our test set along with the sentence generated for the most-likely action class.

4.1 The lattice constructed by the Viterbi algorithm performing detection-based tracking.

4.2 The operation of a detection-based tracker. Three frames from the same video are shown, one in each column. The rows indicate successive information computed by the tracker for each frame. (a) Output of the detection sources, biased to yield false positives. (b) The top-scoring output of the detection source. Note that the top-scoring detection does not track the person or the object as the video progresses. (c) Augmenting the output of the detection sources with forward-projected detections. (d) The optimal tracks selected by the Viterbi algorithm track both the person and the object.

4.3 Improved performance of simultaneous object detection and tracking. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector using models for people, motorcycles, and balls. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous object detection and tracking, as described in Section 4.3.

4.4 Improved performance of simultaneous tracking and event recognition. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous tracking and event recognition, as described in Section 4.4.

4.5 Improved performance of simultaneous object detection, tracking, and event recognition. A single frame from each of four different videos is shown. Each column depicts the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous object-detection, tracking, and event recognition, as described in Section 4.5.
Figure

5.1 Predicates which accept detections, denoted by $a$ and $b$, formulated around 9 parameters. These predicates are used for the second and third experiment, Sections 5.5.2 and 5.5.3. The function project projects a detection forward one frame using optical flow. The functions flow-orientation and flow-magnitude compute the angle and magnitude of the average optical-flow vector inside a detection. The function $a_{cx}$ accesses the $x$ coordinate of the center of a detection. The function $a_{width}$ computes the width of a detection. The functions $\cup$ and $\cap$ compute the area of the union and intersection of two detections respectively. The function $|\cdot|^{\circ}$ computes angular separation. Words are formed as regular expressions over these predicates.

5.2 Regular expressions which encode the meanings of each of the 15 words or lexicalized phrases in the lexicon used for the second and third experiment, Sections 5.5.2 and 5.5.3. These are composed from the predicates shown in Figure 5.1. We use an extended regular-expression syntax where an exponent of $\{t,\}^t$ allows a predicate to hold for $t$ or more frames.

5.3 Tracker lattices are used to track each participant. Word lattices constructed from word FSMs for each word in the sentence recognize collections of tracks for participants that exhibit the semantics of that word as encoded in the FSM. We take the cross product of multiple tracker and word lattices to simultaneously track participants and recognize words. This ensures that the resulting tracks are described by the desired sentence.

5.4 Different sentential queries lead to different cross products. The sentence is parsed and the role of each participant, show in red, is determined. A single tracker lattice is constructed for each participant. Words and lexicalized phrases, shown in blue, have associated word lattices which encode their semantics. The arrows between words and participants represent the track-to-role mappings, $\theta$, required to link the tracker and word lattices in a way that faithfully encodes the sentential semantics. Some words, like determiners, shown in grey, have no semantics beyond determining the parse tree and track-to-role mapping. The dashed lines indicate that the argument order is essential for words which have more than one role. In other words, predicates like ride and away from are not symmetric. Detection sources are shown in black, in this case two object detectors. The tracker associated with each participant has access to all detection sources, hence the bipartite clique between the trackers and the detection sources.
5.5 The grammar for sentential queries used in Section 5.5. Items in black are shared between all experiments. Items in red are exclusive to the first experiment, Section 5.5.1. Items in blue are exclusive to the second and third experiments, Sections 5.5.2 and 5.5.3.

89

5.6 The 21 sentential queries used in Section 5.5.1. Differences in corresponding minimal pairs are highlighted in red and green.

95

5.7 Sentential-query-based video search: returning the best-scoring video, in a corpus of 94 videos, for a given sentence.

96

5.8 The template used to generate the 141 query sentences where X, Y, and Z are either person or horse. The template generates 204 sentences out of which 63 are removed because they involve people riding people and horses riding people or other horses for which no true positives exist in our video corpus.

98

5.9 A comparison between our approach and a baseline system constructed out of state-of-the-art components on the top 10 hits returned for various sentential queries.

101

5.10 (a) Average precision of the top 10 hits for the 141 query sentences as a function of the threshold on the sentence-tracker score. Without a threshold, (b) the number of sentences with at most the given number of hits and (c) the number of sentences with at least the given number of correct hits.

102

5.11 Frames from the top 6 hits for two sentential queries. True positives are shown in green and false positives in red. In both cases, half are true positives. The fact that the results are different shows that our method encodes the meaning of the entire sentence along with which object fills which role in that sentence.

103

6.1 Key frames from sample stimuli for each of the six verbs in Experiment 1.

116

6.2 Results for Experiment 1. (a) Per-subject classification accuracy on 1-out-of-6 verb classes averaged across class and fold. Horizontal line indicates chance performance, 16.66%. (b) Corresponding confusion matrix averaged across subject and fold is mostly diagonal, with the highest numbers of errors being made distinguishing hold and carry, two ambiguous stimuli.

117

6.3 (top) Searchlight analysis for Experiment 1 indicating the classification accuracy of different brain regions on the anatomical scans from subject 1, averaged across stimulus, class, and run. (bottom) A similar analysis using a $w(i)^2$ metric.

118
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Key frames from sample stimuli in Experiment 2.</td>
<td>124</td>
</tr>
<tr>
<td>6.5</td>
<td>Per-subject mean classification accuracies averaged across fold for Experiment 2. Note that all six analyses perform above chance.</td>
<td>125</td>
</tr>
<tr>
<td>6.6</td>
<td>Per-subject classification accuracies for Experiment 2 showing the means and variances of performance across the different folds for each class. The horizontal line indicates chance performance.</td>
<td>126</td>
</tr>
<tr>
<td>6.7</td>
<td>Confusion matrices for Experiment 2, averaged across subject and fold. Note that they are mostly diagonal.</td>
<td>127</td>
</tr>
<tr>
<td>6.8</td>
<td>Searchlight analysis for Experiment 2 indicating the classification accuracy of different brain regions on the anatomical scans from subject 1 averaged across stimulus, class, and run.</td>
<td>127</td>
</tr>
</tbody>
</table>
ABSTRACT

Barbu, Andrei. Ph.D., Purdue University, December 2013. Reasoning Across Language and Vision in Machines and Humans. Major Professor: Jeffrey M. Siskind.

Humans not only outperform AI and computer-vision systems, but use an unknown computational mechanism to perform tasks for which no suitable approaches exist. I present work investigating both novel tasks and how humans approach them in the context of computer vision and linguistics. I demonstrate a system which, like children, acquires high-level linguistic knowledge about the world. Robots learn to play physically-instantiated board games and use that knowledge to engage in physical play. To further integrate language and vision I develop an approach which produces rich sentential descriptions of events depicted in videos. I then show how to simultaneously detect and track objects, recognize events, and produce sentences. This tighter integration of language and vision enables a novel task: sentential video retrieval. A video corpus can be searched for clips which depict a target sentence rather than just a collection of individual query words. This work assumes a compositional representation of events, composing sentence models from word models. Perhaps the reason why humans perform tasks such as the above with ease is because of a tight integration of language and vision exploiting the compositionality inherent in both modalities. I present work indicating that this may be the case. Humans are shown videos while fMRI data is acquired and sentences which describe those videos are recovered compositionally.
1. INTRODUCTION

This thesis is a collection of five papers on the topic of integrating language and vision. These papers are tied together by a common thread: reasoning across multiple modalities can improve the performance of AI systems and allow them to perform novel tasks for which no existing approaches exist. What is meant by reasoning across modalities here is that high-level cognition and low-level perception should influence one another and one should reason jointly about both. One reason for investigating this subject is that evidence exists that humans reason across modalities. Clearly we are able to observe the world and describe it in language. The converse is also true: we can use knowledge about the world in order to organize and understand what we see. For example, one might recognize an otherwise unrecognizable object by putting it into context.

Reasoning across modalities may be more fundamental than the above might suggest. It doesn’t simply increase performance, but also enables performing novel tasks. Humans perform tasks which seem to crucially rely on tight integration between modalities. For example, deciding if a sentence describes a video requires that the meaning of the sentence be grounded in vision. Other tasks, such as manipulating a complex structure, a task which kids perform with ease when playing with assembly toys such as Legos, require far more integration between modalities. Manipulating a Lego structure requires reasoning about physics, the affordances of the parts, vision to recognize structures, language to describe them, etc.

Chapter 2 shows a domain where high-level knowledge can be acquired from low-level perceptual input. In some ways this is a fundamental prerequisite, high-level knowledge that is used to reason jointly across language and vision must be learnable from low-level perception. Two robots are provided with the natural-language rules of a board game. They use these rules to play a physical board game while a third
robot watches. This third robot does not know the rules of the game, but learns the rules of the game by watching a small number of games, typically two to ten. It then uses its learned rules to engage in further physical play.

Reasoning across modalities is facilitated when a single representation can encode inherently-unreliable information about the world and high level, possibly symbolic, information available to higher-level cognition. Chapters 3 through 5 discuss a domain, event recognition, where we demonstrate such tight integration between vision and language. A single cost function simultaneously detects objects, tracks them, and recognizes complex events described by a natural-language sentence. Chapter 3 presents a pipeline approach to describing videos with sentences: object detection is followed by tracking the output of which is used to recognize events and ultimately to generate sentences. Chapter 4 integrates three steps of this pipeline (object detection, tracking, and event recognition) into a single cost function. This allows higher-level knowledge, such as the event being recognized, to affect low-level perception, the object detector, and mid-level perception, the tracker. High-level knowledge can overrule poor decisions made by low-level detectors. Chapter 5 extends this approach to recognizing an event described by a sentence rather than a single word. This integration enables a novel task: sentential video retrieval. We construct a video search engine which accepts sentential queries and finds clips which depict those queries in a corpus of videos. We demonstrate this approach on a corpus of 10 Hollywood movies.

The representations employed in the work described above are compositional. Models for previously-unseen sentences are composed from individual word models using a grammar. Compositionality seems to exist in both vision and language and may be a general principle applicable to many different settings. As such, it may be reflected in the representations employed by the human brain. Using neuroimagining, fMRI, we show that this may be the case. Subjects are shown a video corpus constructed compositionally where each video depicts one of four human actors carrying out one of three actions with one of three objects performed on either side of the field of view. Scans are acquired while subjects watch video clips and the activities in
those clips are decoded from their brain scans. Subjects have no other task than to watch the videos and think about the event they are observing. A four-word sentence which describes the video stimulus is recovered compositionally. Each word in the sentence is recovered by a separate classifier for that part of speech. This indicates that the representations used in the brain may be linguistic in nature. Moreover, the performance of joint classifiers for multiple words, such as verbs and objects, is similar to that of individual word classifiers. This indicates that the representations employed may also be compositional.

Chapter 2 was derived from [1] and contains no modifications. It is joint work with N. Siddharth and J. M. Siskind. Chapter 3 was derived from [2] and contains no modifications. It is joint work with A. Bridge, Z. Burchill, D. Coroian and S. J. Dickinson, S. Fidler, A. Michaux, S. Mussman and N. Siddharth, D. Salvi, L. Schmidt, J. Shangguan and J. M. Siskind, J. W. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Chapter 4 was derived from [3] and contains no modifications. It is joint work with N. Siddharth, A. Michaux, and J. M. Siskind. Chapter 5 was derived from [4] and is expanded to include additional details and experiments. It is joint work with N. Siddharth and J. M. Siskind. Chapter 6 was derived from [5] and is expanded to include additional experiments. It is joint work with N. Siddharth, C. Xiong, J. J. Corso, C. D. Fellbaum, C. Hanson, S. J. Hason, S. Hélie, E. Malaia, B. A. Pearlmutter, J. M. Siskind, T. M. Talavage, and R. B. Wilbur.
2. GAME LEARNING

Children learn to play games by watching others play. While both formal board games, like Chess, Checkers, and Backgammon, and less formal play like Hopscotch, Tag, and Dodgeball all have well defined rules that children ultimately come to know, they are rarely told those rules explicitly. Knowledge of how to play many classic board games is largely passed down culturally, with children never reading, and often even explicitly ignoring, the formally-specified rules (e.g. Monopoly®).

We are engaged in a long-term research effort to emulate on robots this ability to learn to play games by observing others play. The work presented here is part of a larger effort to ground learning, reasoning, and language in visual perception and motor control. Physical instantiation is crucial to our effort of situating learning, visual perception, and manipulation in the real world. We want physical robots to play a physical game where knowledge of game play allows their vision systems to determine game progress and motor systems to effect game progress. We also want a physical learner to visually observe that play to learn the game rules and ultimately be able to use the learned rules to support physical game play.

Our long-term vision for this overall task is depicted in Figure 2.1. In this task, two robotic agents, the protagonist and antagonist, play a board game like Chess. A third robotic agent, the wannabe, does not know the rules of the game but must infer the rules by visually observing the play of the protagonist and antagonist. The wannabe must then use these rules for further physically-instantiated play. In the long term, we wish to be able to do this for a wide variety of off-the-shelf game hardware for a wide variety of common physically-instantiated board games. Our

objective is to learn to play legally, not necessarily well. Expert computer game play is one of the most extensively studied and successful sub-disciplines of AI. Our goal is orthogonal to that enterprise.

We have constructed a novel custom robot to support this enterprise, and have used this robot to successfully learn six different physically-instantiated games. While
one long-term goal is to learn a wide variety of common board games, like Chess, Checkers, Backgammon, and Go, with differing physical game hardware, the work presented in this paper is limited to games which share the same game hardware. And while another long-term goal is to use three separate robotic agents to play the roles of protagonist, antagonist, and wannabe, the work presented here uses a single robot to play all three roles. We do however, use a unique capability of our novel custom robot to simulate play by multiple distinct agents by robotically moving the camera to image the game play from different viewpoints.

2.1 Our Custom Robot

We have designed a custom robot and built three copies thereof, one of which is shown in Figure 2.2. While much of our robot is constructed with off-the-shelf parts, many crucial parts were custom designed, milled, or repurposed to meet the particular needs of the game-playing task. The two most-novel parts are the overall housing and camera-mount assembly. The overall housing consists of a two-level wood platform, where the upper level constitutes the game-play surface and the lower level serves as the mounting point for the camera assembly. A 5 DOF arm with two independently-controllable fingers, is bolted to the upper level. The size of the overall housing and the arm link lengths were designed to support game play with off-the-shelf game hardware.

Our robot hand contains a number of sensors to support fine motor control for manipulating game pieces: a palm-mounted camera, an ultrasonic range sensor, a laser pointer, and tactile force sensors on each fingertip. The camera assembly consists of a pair of pan-tilt USB webcams mounted on a 1 DOF pendulum arm which is in turn mounted on a servo base bolted to the lower level. We found Logitech QuickCam Orbit cameras well-suited to our task, as we were able to strip them down to a lightweight assembly containing the camera, pan-tilt motors, and electronics, allowing them to be mounted on the pendulum arm. This allows them to pivot, under computer control,
180° around the center of the game-play surface and affords a binocular view of the whole game board through the entire pivot range. We have conducted experiments both where the pendulum head mount is fixed throughout game play, imaging the game-play surface from a single viewpoint shared by the protagonist, antagonist, and wannabe and also where we pivot the camera under computer control to have the three agents view the game from different viewpoints.

2.2 The Space of Games Considered

For reasons to be discussed momentarily, our games share common physical game hardware consisting of an off-the-shelf Tic Tac Toe set (see Figure 2.2). This particular game hardware simplifies the necessary robotic manipulation in several ways. First, the fact that the board positions are depressions makes piece placement somewhat self correcting. Second, the piece size is well matched to our manipulator. The game hardware also simplifies the process of finding board positions while reconstructing symbolic game states from visual input. In addition to the board, we have caches for storing off-board game pieces that are not in play. Since the off-the-shelf Tic Tac Toe set did not include such, we constructed our own that contain self-correcting circular depressions of the same size as the board.

Our long-term objective is to be able to learn any typical board game such as Chess, Checkers, Backgammon, Go, Stratego, etc. Such a wide variety of games would require more-general perceptual and motor abilities than we have implemented. We wish to leverage our implemented perceptual and motor abilities as much as possible yet verify the generality of our overall approach by evaluating its ability to learn a variety of games. Thus we have chosen a collection of six simple games that can all be played with the same physical game hardware, robotic hardware, and perceptual and motor software. Two, Tic Tac Toe and Hexapawn\textsuperscript{1} [6], are

\textsuperscript{1}On a 3 × 3 board, three white pieces start on one edge and three black pieces start on the opposite edge. Pieces move and capture like Chess pawns without en passant or initial two-square moves. Players win by queening and lose when unable to move.
Fig. 2.2. One of our three novel custom robots designed specifically to support play of physically-instantiated board games. Note the 5 DOF arm with two independently-controllable fingers mounted on the upper level of a two-level housing that serves as the game-play surface. Also note the binocular pan-tilt cameras mounted on a pendulum arm that can pivot 180° around the game-play surface by being mounted on the lower level of the housing.
commonly-known games, while the remaining four are minor variants of Hexapawn.
We summarize these variants below:

**Variant A** Non-capturing moves are forward along the diagonal instead of straight ahead.

**Variant B** Variant A augmented with backward diagonal non-capturing moves.

**Variant C** Hexapawn allowing backward vertical non-capturing moves.

**Variant D** Variant C augmented with sideways non-capturing moves.

Our system contains a generic game-playing engine that is parameterized by a game specification containing an initial board configuration, legal-move generator, and outcome predicate. It accepts game specifications coded in either Scheme, Prolog (see Figure 2.5), or a subset of English (see Figure 2.3). Specifications in any of these three forms can be used to drive physical game play where the robot (Figure 2.2) alternates play between X and O and determines the game state by visual observation of the physical game between each move, optionally moving the pendulum arm to determine the game state from different viewpoints for the different players.

We provide hand-coded game specifications, in any of the forms, to the protagonist and antagonist to generate the training set for the wannabe, which does not have access to these hand-coded specifications. The learning component, to be described in Section 2.5, can learn game specifications in Prolog for all of the above six games from visual observation of physical game play which can then be used to drive new physical game play. While our game-playing engine also supports optimal play via minimax game-tree search with alpha-beta pruning, optimal play does not facilitate learning of games rules as many situations necessary to learn the complete rule set do not arise, particularly for learning the outcome predicate in deterministic games. Thus for learning we employ random legal play.
Every cache square for every player in the initial state has some piece of that player.

A player moves by moving some piece of that player from some cache square for that player to some empty board square.

A player wins when every square in some row has some piece of that player.
A player wins when every square in some column has some piece of that player.
A player wins when every square in some diagonal has some piece of that player.

A player draws when no player wins and that player has no move.

Fig. 2.3. As demonstrated in the video on our website, our system is able to play a physically-instantiated board game given rule specifications, such as these for Tic Tac Toe, in a subset of English.
2.3 Game-State Reconstruction

Physically-instantiated game play requires recovery of the game state from visual information, which nominally is a two-stage process. First, one must determine the world state, i.e. the shapes and positions of various pieces and board regions. Then, one must map this world-state information into game states. The former process is nominally a game-independent general-vision task of scene reconstruction and may incorporate camera calibration, segmentation, object recognition, metric reconstruction, and pose estimation. The latter process requires game-specific knowledge to determine which features of the reconstructed scene are relevant to the particular game being played. For example, in some games, like Chess and Checkers, pieces are placed between edges, while in others, like Go, pieces are placed on edges. In the longer term, we intend to learn such game-specific knowledge of how world states map to game states along with the initial board configuration, legal-move generator, and outcome predicate. In this paper, we restrict consideration to games played with the particular game hardware discussed in Section 2.2 where all games use the same hard-coded mapping from world state to game state. Thus we conflate the two-stage process into a direct mapping from images to game states.

We employ two different game-state reconstruction methods. Our older method calibrates the reconstruction process prior to game play by determining the image regions that correspond to board positions using the ellipse detector (cvFitEllipse) in OPENCV [7] constrained to find a $3 \times 3$ grid. The content (empty, X, or O) of each elliptical board position during game play is determined by detecting Os as smaller ellipses and Xs as the absence of any ellipse. We make this process highly robust by sampling the ellipse-detector output at multiple thresholds and voting on the outcome. This method, however, is suitable only for a stationary head position because the ellipse detector is highly sensitive to the small changes in camera viewpoint that result when moving the head camera to a different position and back again.
Our newer game-state reconstruction method works even when the head camera moves to view the game-play surface from the perspective of different agents. Prior to game play, we calibrate the reconstruction process independently for each viewpoint by finding the image regions that correspond to board positions by dilating and thresholding the raw color image, finding connected components in the resulting binary image, and filtering the results under critical-point detection constrained to find a $3 \times 3$ grid. Critical-point detection is an iterative process on the convex hull of the centroids of the connected components, finding and removing the bottom-most three-element row of connected components from the collection of connected components in each of three iterations and recomputing the convex hull. We also calibrate color templates for the possible position states (empty, X, and O), from each viewpoint, prior to game play, by computing mean and covariance measures of HSV representations of sampled pixels from board positions with known states and classify unknown position states during game play using Mahalanobis distance.

We improve the robustness of both methods by enhancing the contrast of the circular depressions on the game board with white inserts to highlight the edges. With this, the older game-state reconstruction process is sufficiently robust that it made only two errors in the approximately 2000 reconstructions performed during the 62 games played autonomously for the experiments reported in this paper. As both occurred during the cleanup at the end of the training sequence, they did not impact the correctness of learning or subsequent physical play with the learned rules. The newer game-state reconstruction process is also sufficiently robust to complete an entire training regimen for Tic Tac Toe while varying the camera viewpoint for each agent (approximately 200 reconstructions in all) without any errors and supports learning a sound and complete game-rule specification.

While the cognitive portion of game play relies solely on the game state represented in the $3 \times 3$ grid of board positions, the robotic portion of game play requires knowing the positions of off-board pieces in the caches. We currently do not determine those positions from visual input and rely on properties of the particular robotic
manipulation strategy discussed in Section 2.4 that allow inferring the positions of off-board pieces from the observed game state.

2.4 Rule-Independent Piece Manipulation

Physically-instantiated game play also requires robotic ability to effect the desired changes in the physical game state. This also nominally can be divided into a two-stage process. First, one must determine the necessary changes in the world state that correspond to the desired changes in the game state, i.e. a legal move. Then, one must effect that change to the world state. The latter process is again nominally a game-independent robotic manipulation task which may incorporate forward and inverse kinematics, grasp planning, and path planning. The former process, however, requires game-specific knowledge, essentially the inverse of the game-specific knowledge needed for game-state reconstruction from visual input. Like before, we restrict consideration to games played with the particular game hardware discussed in Section 2.2 but do not conflate this into a single-stage process.

Our legal-move generator is formulated as a mapping from old game states to new game states. We formulate a generic method, particular to our class of games played on a $3 \times 3$ grid but applicable to any game in that class, that finds a minimal number of pick up and put down operations to effect the target change in the physical game state. Such operations may move pieces between two board positions, or between the caches and the board. In the case of the latter, we assume that game rules do not constrain the choice of cache location for any particular legal move and treat each cache as a last-in-first-out stack, one on one side for Xs and on the other for Os. This stack behavior is what allows indirect inference of the positions of off-board pieces from the observed game state without direct visual observation. We also have a generic “clean up” capability that can return all pieces to a state that corresponds to an arbitrary (but learned) initial board configuration. This allows completely autonomous robotic
play of a sequence of games to provide training data for the learner and evaluate autonomous play with the learned rules.

The above constitutes the first stage of the two-stage process, namely mapping from target game-state changes to world-state changes. Again, we employ two different methods for the second stage, namely affecting the desired change to the world state by picking up and putting down pieces. The older method uses an open-loop dead-reckoning process. We hard-code the 3D world coordinates of the board and cache positions for our robots and employ inverse kinematics to determine a sequence of joint-angle configurations to effect a desired *pick up* or *put down* operation, parameterized by a specific board or cache position. The nature of board-game play allows straightforward planning of a collision-free path by approaching board and cache positions from above. Restriction of the game hardware to a particular piece set means that we can hard-code the grasp planning for each piece type. This is implemented by providing the *pick up* or *put down* operations with piece type as an additional parameter, derived visually. Our newer method improves upon the older method by automatically determining the 3D world coordinates of the board positions given a 3D model of the board together with visually-determined board pose. Such board-pose determination requires camera calibration, which is done automatically. Our newer method also automatically determines the parameters of the inverse kinematics via training on a fiducial. Finally, it augments the open-loop dead-reckoning process, now used only for coarse motor control, with a closed-loop visual-servoing process employing the palm camera and tactile sensors to implement the fine motor control for grasping the playing pieces.

We improve the robustness of both of the above methods by using visual feedback to confirm the success of a desired *pick up* or *put down* operation and compensating upon failure. Our combined vision and robotic-manipulation systems are sufficiently robust that the approximately 2000 *pick up* and *put down* operations during autonomous play of the 62 games for the experiments reported in this paper required fewer than 20 human interventions to correct robotic errors.
2.5 Learning Game Rules by ILP

We employ inductive logic programming (ILP) with PROGOL [8] to learn perspicuous Prolog specifications for the game rules from autonomous physically-instantiated game play. We currently do this on a single robot that plays all three roles of protagonist, antagonist, and wannabe, taking care not to allow information flow that could not happen if this were done on three distinct robots. The protagonist and antagonist autonomously play a sequence of random but legal games, at least six for Tic Tac Toe and at least ten for Hexapawn and its variants. As discussed earlier, we do not train on optimal play because this does not provide sufficient information to infer the game rules. Furthermore, games ending in a draw do not contribute to learning the outcome predicate and hence are ignored. Thus when randomly-generated training games end in a draw, additional games are played until a requisite number of non-draw games have been collected.

Each game starts with the robot autonomously setting up the physical game hardware to correspond to the initial board configuration, given an image of the current world state, which for any game in the training set but the first, contains pieces remaining on the board from the previous game. The protagonist and antagonist then alternate play by taking an image to determine the current game state (which is not stored from the previous turn), selecting a random desired next state from the legal-move generator, planning a sequence of pick up and put down operations to effect that move, and executing those operations on the robot. Independent from this, the wannabe takes an image between each turn to determine the sequence of game states corresponding to a game and is given the outcome of the game at each turn, which may indicate that the game is not yet over. We conduct such autonomous game play both with a stationary head camera as well as a moving head camera to image the game from different viewpoints for each agent. The only communication between the three agents is the labeling of the outcome at each turn as well as the
turn-taking coordination that informs each agent of the transition time between game states.

Upon completion of the autonomous training play, the wannabe formulates the training set as input to PROGOL in the following form. A game state is formulated as a $3 \times 3$ matrix. Each game is formulated as a fact initial_board($G,P$), followed by alternating facts legal_move($P,G_1,G_2$) and outcome($P,G,O$) for each turn, where $G$ denotes a game-state matrix, $P$ denotes a player, either X or O, and $O$ denotes a win by either X or O. Note that we provide an outcome training fact for all moves, negated for moves that do not end in a win, which provides negative evidence for learning the outcome predicate. We then use PROGOL to learn definitions for initial_board/2, legal_move/3, and outcome/3. To facilitate learning, we augment the training data with the background knowledge in Figure 2.4. This background knowledge is the same for all six games discussed in Section 2.2. Much of it encodes general elementary physical and mathematical properties known by almost all children, such as arithmetic (inc/2, dec/2, row_to_int/2, and col_to_int/2), the concept of a line (linear_test/7 and linear_obj/3), and the frame axiom (replace/4, frame/4, and frame_obj/6). Some (player/1, opponent/2, piece/1, empty/1, owns/2, win_outcome/1, owns_outcome/2, owns_piece/2, row/1, col/1, board/1, ref/3, at/4, and at/5) encode knowledge about the mapping between world state and game state which we currently do not learn but anticipate learning in the future. The remainder (forward/3 and sideways/2) encode combinations of that mapping with general spatial-relation knowledge known by almost all children. To drastically reduce the learning time, we use a non-generative version of at/4 (with cuts) while learning the legal-move generator.

During training, we first learn both the initial board configuration and the legal-move generator with the background knowledge from Figure 2.4. Then, to learn the outcome predicate, we augment the training set and the background knowledge with the learned legal-move generator, along with two predicates, has_move/2 and has_no_move/2, that access that learned legal-move generator. Furthermore, due to
significant overlap in coverage of the clauses generated by Progol for the outcome predicate, we replace Progol’s internal redundancy algorithm with one that searches for a minimal subset of the candidate clauses that covers the training set. Finally, to drastically reduce the learning time, we replace the definition of the frame axiom with one that is vacuously true while training the outcome predicate.

2.6 Results

Our system can learn the rules of all six games discussed in Section 2.2 from autonomous physically-instantiated play by robotic agents using the methods discussed. The learned game rules for Tic Tac Toe and Hexapawn are given in Figure 2.5. From simulated non-robotic play, we have determined that six training examples are almost always sufficient in practice to correctly determine the game rules for Tic Tac Toe (ten for Hexapawn and its variants). Due to the fact that we train on random legal play, it is possible but unlikely that the training set can be pathological and contain a skewed mix of the possible game situations which can lead to incorrect generalization. Furthermore, random play can make it unlikely to observe the specific events that are necessary to learn certain aspects of the rules, such as the requirement in Hexapawn that players lose when unable to move. We have determined, from simulated non-robotic play, that losing in this fashion occurs with high probability among a sample of ten games but not six.

We have set up a website\(^2\) that contains videos that demonstrate the autonomous play of the training set for each of the six games as well as subsequent autonomous play using each of the six sets of learned game rules. These video sequences were gathered with a fixed head-camera position and our older game-state-reconstruction and robotic-manipulation methods, where the protagonist and antagonist play the training set with rules specified in Scheme and then the protagonist and wannabe play with the learned Prolog rules (Figure 2.5). The website also con-

\(^2\)http://www.ece.purdue.edu/~qobi/icra2010
tains a video gathered with head-camera position varying for each of the agents and our newer game-state-reconstruction and robotic-manipulation methods, where the protagonist and antagonist play the training set with rules specified in a subset of English (Figure 2.3) and then the protagonist and wannabe play with the learned PROLOG rules (Figure 2.5). This website also contains the full source code for our system as well as engineering drawings for the design of our custom robot and cache hardware allowing others to replicate and build upon our work.

2.7 Comparison With Related Work

On the surface, it might appear that our work resembles that of the general game playing community in that we both share the goal of 'learning to play' games [9,10]. Deeper inspection, however, reveals that the apparent similarity is misleading. Our work aims to learn to play legally; their work aims to learn to play well. In more-technical terms, we take a sequence of exemplar game-play instances as input and produce a game-rule description (in the form of an initial board configuration, legal-move generator, and outcome predicate), as output. They take such a game-rule description as input and produce a strategy as output. That strategy might take the form of a static evaluator or heuristic function. As such, we are addressing complementary problems. An interesting opportunity for future work would be to cascade the two into a unified system that learns to play both legally and well from exemplar game-play instances. Indeed, such an endeavor is facilitated by the fact that the general-game-playing community has adopted a standard game description language (GDL) [11] for inputting the game rules to their learning systems. Fortuitously, GDL is virtually identical to the PROLOG game-rule specifications output by our system.

Our work involves three components: game-rule learning, game-state reconstruction from visual input, and robotic manipulation of board-game hardware. We know of no other work that integrates all three of these. However, there has been some work that addresses each of these components individually.
2.7.1 Game-Rule Learning

We have found surprisingly little prior work on learning game rules from example game play. Michalski and Negri [12] employed ILP to learn a static evaluator for Chess endgames. However, this constitutes learning to play (only part of the game) well, not legally. Levinson [13] formulated the task of learning a legal-move generator for Tic Tac Toe and Hexapawn as reinforcement learning, but it unfortunately does not usually converge to the correct result. Furthermore, the legal-move generator is not represented in a perspicuous human-readable format. In the above work, the game-play examples are provided symbolically, and not derived from visual input. As far as learning game rules from visual examples of game play, our efforts are most similar to those of Needham et al. [14–16] and Antanas et al. [17]. The former line of work processes bird’s-eye view video of several simple games (Paper Scissors Stone played with cards and three variants of Snap, two played with cards and one played with dice) to produce a categorical symbolic representation of the exemplar game-state sequences and then uses ILP to learn the rules of these games. Note that the size of the game state in this work is small, \((3 \times 3 \times 2)^2 = 324\) for the card-based games and \(7^2 = 49\) for dice Snap. In contrast, the size of the game state for our games is significantly larger, \(3^9\). Moreover, the rules of our games are significantly more complex than theirs. Dice Snap is specified with Prolog rules with at most four variables and three goals. The Prolog rules for the card-game specifications require at most seven variables and three goals. Moreover, in both of these systems, the rules contain numerous ground terms produced by the perceptual system (e.g. rollboth, rollone, pickuplowest, tex0, tex1, tex2, pos0, pos1, pos2), indicating tabular encoding of the rules with little conceptual generalization. In contrast, as can be seen in Figure 2.5, the rules of our games have as many as fifteen variables and thirteen goals, and contain no ground terms other than the x, o, none, and player.x that appear in the initial board configuration, indicating significantly greater conceptual generalization in the legal-move generator and outcome predicate. They use
essentially no background knowledge that is comparable to the background notions in our work: arithmetic, the concept of a line, spatial relations, and the frame axiom.

The above line of work, like ours, first processes the perceptual input to yield categorical symbolic descriptions that serve as the input to a purely symbolic learning process. Like ours, this pipeline is brittle in the face of errors in categorical perception. They report accuracy levels in categorical perception between 83% and 100% for dice Snap but not those for their card games. Moreover, they report that errors in categorical perception lead to errors in game-rule learning: *We are learning from small amounts of data which is locally sparse thus a classification error may make up a significant amount of the data used to form generalisation. This may seem very fragile.*

Of the nine experiments reported (three runs for each of the three games), only one experiment yielded a sound and complete game-rule specification. In contrast, our game-state-reconstruction front end is sufficiently robust to support learning sound and complete game-rule specifications for all six of our games. Moreover, discussing the above line of work [18] Muggleton states *a key challenge will be to push the system to learn more difficult things.* “It would be interesting to see if this approach will scale up to more complex games such as noughts-and-crosses [...]” he adds. Our work does just that.

Antanas et al. [17] report a method for learning a subset of Uno where the categorical output of game-state reconstruction is replaced with soft evidence that is then processed by probabilistic ILP. However, they expressly disregard the softness of that evidence in the experiments that they report. Moreover, they report only approximately 95% accuracy on noise-free training data and less than 90% accuracy on noisy training data. And even when trained on noise-free data, their method produces a probabilistic game-rule specification that places 9% of the probability mass on incorrect game rules. Nonetheless, we consider this a promising approach to support robust learning of game rules despite noisy game-state reconstruction.
2.7.2 Game-State Reconstruction from Visual Input

Some prior work processes game-board images from a bird’s-eye view while other work, like ours, processes game-board images from the perspective of a typical human player. Shiba and Mori [19] present a method for recovering the perimeter of a Go board from a single human-view image but do not attempt to recover piece positions or game state. Ren et al. [20] present a method for recovering the perimeter of a game board from a bird’s-eye-view image sequence and then recover piece positions relative to this perimeter. Their method applies to an arbitrary rectangular board but they do not report using piece position to reconstruct game state and do not report accuracy. Hirsimäki [21] presents a method for recovering Go-board states from human-view images. However, they only report testing their method on six images, out of which it only succeeds in recovering the correct game state on four. Kang et al. [22] present a method for recovering a sequence of Go-game moves from bird’s-eye-view image sequences. However, they do not report any quantitative evaluation of game-state reconstruction accuracy. Torre et al. [23] present a method for recovering a sequence of Checker-game moves from human-view image sequences of games played on a game board with a known geometric model. However, they report neither qualitative nor quantitative assessment of game-state reconstruction accuracy. Scher et al. [24] present a method for recovering a sequence of Go-game moves from human-view image sequences. A novel aspect of this work is that they improve upon the purely vision-based game-state reconstruction by incorporating the rules of Go as constraints. Without such constraint, their reconstruction accuracy is extremely poor: a total of 650 piece classification errors out of 265 moves. Even adding the game-rule constraints only reduces the number of piece classification errors on this data set to 25. Note that these numbers constitute errors in recognizing the presence or absence of a black or white stone at individual positions, not the aggregate whole-game-state recognition accuracy, which would be significantly lower. Seewald [25] presents a method for recovering game states from human-view images
of Go boards with a reported whole-game-state recognition accuracy of 72.7%. In contrast, we recover game moves from human-view image sequences with only two erroneously reconstructed game states out of approximately 2000. We are unaware of any prior work that achieves anywhere near this level of accuracy. Such performance is crucial for accurately learning game rules. Moreover, we cannot avail ourselves of the technique employed by Scher et al. as the wannabe does not yet know the game rules.

2.7.3 Robotic Manipulation of Board-Game Hardware

The robotic ability to manipulate board-game pieces by dead reckoning is straightforward and commonplace, and there have been attempts to integrate such ability with automated board-game play based on visual perception of game states and piece positions [26–28]. However, we are unaware of any prior work that reports such a combined sensorimotor game-play system that achieves the level of robustness needed for fully-automated play of a training set that is sufficiently large to support game-rule learning. Indeed, despite the fact that they use a bird’s-eye-view camera together with a vacuum grip to pick up Trax tiles, alleviating the need for fine motor control, Bailey et al. [27] state At present, the robot is a little clumsy, frequently placing tiles slightly over the top of other tiles.

Furthermore, an important aspect of our work is the way our robot was designed specifically to support game-rule learning. While our robot uses a common 5 DOF arm, this arm is mounted in a novel environment. We know of no prior robot that incorporates our two-level housing design and its associated pendulum head mount. Moreover, we know of no prior attempt to model multi-agent robotic board-game-play systems by automatically changing the camera view by such robotic means to model each agent. Our combined vision and robotic system is sufficiently robust not only to support game-state reconstruction and game-piece manipulation from varying
camera views, it can do so even with the imaging variation due to inherent inaccuracy in repeated attempts to move the head camera to the same position.

2.8 Conclusion

We have presented the first integration of vision, robotics, and game-rule learning in the realm of board-game play. Our work is also a significant advance over prior independent work in each of these areas. The richness of board-game play allows for an open-ended research program investigating perceptual and motor grounding of natural language, reasoning, and learning yet its circumscribed nature allows for robust incremental progress.
inc(X,Y):- Y is X+1.
dec(X,Y):- Y is X-1.
player(player_x).
player(player_o).
opponent(player_x,player_o).
opponent(player_o,player_x).
piece(x).
piece(o).
piece(none).
empty(none).
owns(player_x,x).
owns(player_o,o).
owns_outcome(player_x,x_wins).
owns_outcome(player_o,o_wins).
owns_piece(x_wins,x).
owns_piece(o_wins,o).
row(r0).
row(r1).
row(r2).
col(c0).
col(c1).
col(c2).
frame(Board).
frame_obj(Board1,Board2).
forward(player_x,R1,R2).
sideways(C1,C2).
linear_test(X1,Y1,X2,Y2,X3,Y3,S).
linear(R1,C1,R2,C2,R3,C3):=
row_to_int(R0,0).
row_to_int(R1,1).
row_to_int(R2,2).
col_to_int(C0,0).
col_to_int(C1,1).
col_to_int(C2,2).
col_to_int(C3,3).
col_to_int(C4,4).
col_to_int(C5,5).
row_to_int(R1,R11).
row_to_int(R2,R12).
row_to_int(R3,R13).
replace(COL1,ignore,C1,L1).
replace(COL2,ignore,C2,L2).
replace(COL3,ignore,C3,L3).
replace(COL4,ignore,C4,L4).
replace(COL5,ignore,C5,L5).
replace(COL6,ignore,C6,L6).

Fig. 2.4. Background knowledge encoded in Prolog. Progol-specific settings and mode, type, and pruning declarations have been omitted.
Tic Tac Toe

\[
\text{initial_board}([[\text{none,none,none}],
[\text{none,none,none}],
[\text{none,none,none}]],
\text{player}_x).
\]

\[
\text{legal_move}(A,B,C) :- \text{owns}(A,D),
\text{row}(E),
\text{col}(F),
\text{at}(E,F,B,\text{none},G),
\text{at}(E,F,C,D,H),
\]

\[
\text{outcome}(A,B,C) :- \text{owns_piece}(C,D),
\text{at}(E,F,B,D),
\text{at}(G,H,B,D),
\text{at}(I,J,B,D),
\text{linear}(E,F,G,H,I,J).
\]

Hexapawn

\[
\text{initial_board}([[\text{x,x,x}],
[\text{none,none,none}],
[\text{o,o,o}]],
\text{player}_x).
\]

\[
\text{legal_move}(A,B,C) :- \text{row}(D),
\text{col}(E),
\text{owns}(A,F),
\text{empty}(G),
\text{forward}(A,H,D),
\text{at}(H,E,B,F,I),
\text{at}(H,E,G,J),
\text{frame_obj}(I,K,J,L,B,C).
\]

\[
\text{outcome}(A,B,C) :- \text{opponent}(A,D),
\text{has_no_move}(A,B),
\text{owns_outcome}(D,C).
\]

Fig. 2.5. Rules for two of the six games discussed in Section 2.2 learned automatically from visual observation of autonomous physically-instantiated game play.
3. VIDEO IN SENTENCES OUT

We present a system that produces sentential descriptions of short video clips. These sentences describe *who* did *what* to *whom*, and *where* and *how* they did it. This system not only describes the observed action as a verb, it also describes the participant objects as noun phrases, properties of those objects as adjectival modifiers in those noun phrases, the spatial relations between those participants as prepositional phrases, and characteristics of the event as prepositional-phrase adjuncts and adverbial modifiers. It incorporates a vocabulary of 118 words: 1 coordination, 48 verbs, 24 nouns, 20 adjectives, 8 prepositions, 4 lexical prepositional phrases, 4 determiners, 3 particles, 3 pronouns, 2 adverbs, and 1 auxiliary, as illustrated in Table 3.1.

Production of sentential descriptions requires recognizing the primary action being performed, because such actions are rendered as verbs and verbs serve as the central scaffolding for sentences. However, event recognition alone is insufficient to generate the remaining sentential components. One must recognize object classes in order to render nouns. But even object recognition alone is insufficient to generate meaningful sentences. One must determine the roles that such objects play in the event. The agent, i.e. the doer of the action, is typically rendered as the sentential subject while the patient, i.e. the affected object, is typically rendered as the direct object. Detected objects that do not play a role in the observed event, no matter how prominent, should not be incorporated into the description. This means that one cannot use common approaches to event recognition, such as spatiotemporal bags of words [29–31], spatiotemporal volumes [32–34], and tracked feature points [35–37] that do not determine the class of participant objects and the roles that they play. Even combining such approaches with an object detector would likely detect objects that don’t participate in the event and wouldn’t be able to determine the roles that any detected objects play.
**coordination**: and

**verbs**: approached, arrived, attached, bounced, buried, carried, caught, chased, closed, collided, digging, dropped, entered, exchanged, exited, fell, fled, flew, followed, gave, got, had, handed, hauled, held, hit, jumped, kicked, left, lifted, moved, opened, passed, picked, pushed, put, raised, ran, received, replaced, snatched, stopped, threw, took, touched, turned, walked, went

**nouns**: bag, ball, bench, bicycle, box, cage, car, cart, chair, dog, door, ladder, left, mailbox, microwave, motorcycle, object, person, right, skateboard, SUV, table, tripod, truck

**adjectives**: big, black, blue, cardboard, crouched, green, narrow, other, pink, prone, red, short, small, tall, teak, toy, upright, white, wide, yellow

**prepositions**: above, because, below, from, of, over, to, with

**lexical PPs**: downward, leftward, rightward, upward

**determiners**: an, some, that, the

**particles**: away, down, up

**pronouns**: itself, something, themselves

**adverbs**: quickly, slowly

**auxiliary**: was

| Table 3.1 | The vocabulary used to generate sentential descriptions of video. |
Producing elaborate sentential descriptions requires more than just event recognition and object detection. Generating a noun phrase with an embedded prepositional phrase, such as *the person to the left of the bicycle*, requires determining spatial relations between detected objects, as well as knowing which of the two detected objects plays a role in the overall event and which serves just to aid generation of a referring expression to help identify the event participant. Generating a noun phrase with adjectival modifiers, such as *the red ball*, not only requires determining the properties, such as color, shape, and size, of the observed objects, but also requires determining whether such descriptions are necessary to help disambiguate the referent of a noun phrase. It would be awkward to generate a noun phrase such as *the big tall wide red toy cardboard trash can* when *the trash can* would suffice. Moreover, one must track the participants to determine the speed and direction of their motion to generate adverbs such as *slowly* and prepositional phrases such as *leftward*. Further, one must track the identity of multiple instances of the same object class to appropriately generate the distinction between *Some person hit some other person* and *The person hit themselves*.

A common assumption in Linguistics [38, 39] is that verbs typically characterize the interaction between event participants in terms of the gross changing motion of these participants. Object class and image characteristics of the participants are believed to be largely irrelevant to determining the appropriate verb label for an action class. Participants simply fill roles in the spatiotemporal structure of the action class described by a verb. For example, an event where one participant (the agent) *picks up* another participant (the patient) consists of a sequence of two sub-events, where during the first sub-event the agent moves towards the patient while the patient is at rest and during the second sub-event the agent moves together with the patient away from the original location of the patient. While determining whether the agent is a *person* or a *cat*, and whether the patient is a *ball* or a *cup*, is necessary to generate the noun phrases incorporated into the sentential description, such information is largely irrelevant to determining the verb describing the action. Similarly, while determining
the shapes, sizes, colors, textures, etc. of the participants is necessary to generate adjectival modifiers, such information is also largely irrelevant to determining the verb. Common approaches to event recognition, such as spatiotemporal bags of words, spatiotemporal volumes, and tracked feature points, often achieve high accuracy because of correlation with image or video properties exhibited by a particular corpus. These are often artefactual, not defining properties of the verb meaning (e.g. recognizing diving by correlation with blue since it ‘happens in a pool’ [35, p. 2002] or confusing basketball and volleyball ‘because most of the time the [...] sports use very similar courts’ [40, p. 506]).

3.1 The mind’s eye corpus

Many existing video corpora used to evaluate event recognition are ill-suited for evaluating sentential descriptions. For example, the Weizmann dataset [32] and the KTH dataset [36] depict events with a single human participant, not ones where people interact with other people or objects. For these datasets, the sentential descriptions would contain no information other than the verb, e.g. The person jumped. Moreover, such datasets, as well as the Sports Actions dataset [34] and the YouTube dataset [35], often make action-class distinctions that are irrelevant to the choice of verb, e.g. wave1 vs. wave2, jump vs. pjump, Golf-Swing-Back vs. Golf-Swing-Front vs. Golf-Swing-Side, Kicking-Front vs. Kicking-Side, Swing-Bench vs. Swing-SideAngle, and golf_swing vs. tennis_swing vs. swing Other datasets, such as the Ballet dataset [37] and the UCF50 dataset [35], depict larger-scale activities that bear activity-class names that are not well suited to sentential description, e.g. Basketball, Billiards, BreastStroke, CleanAndJerk, HorseRace, HulaHoop, MilitaryParade, TaiChi, and YoYo.

The year-one (Y1) corpus produced by DARPA for the Mind’s Eye program, however, was specifically designed to evaluate sentential description. This corpus contains two parts: the development corpus, C-D1, which we use solely for training,
and the evaluation corpus, C-E1, which we use solely for testing. Each of the above is further divided into four sections to support the four task goals of the Mind’s Eye program, namely recognition, description, gap filling, and anomaly detection. In this paper, we use only the recognition and description portions and apply our entire sentential-description pipeline to the combination of these portions. While portions of C-E1 overlap with C-D1, in this paper we train our methods solely on C-D1 and test our methods solely on the portion of C-E1 that does not overlap with C-D1.

Moreover, a portion of the corpus was synthetically generated by a variety of means: computer graphics driven by motion capture, pasting foregrounds extracted from green screening onto different backgrounds, and intensity variation introduced by postprocessing. In this paper, we exclude all such synthetic video from our test corpus. Our training set contains 3480 videos and our test set 749 videos. These videos are provided at 720p@30fps and range from 42 to 1727 frames in length, with an average of 435 frames.

The videos nominally depict 48 distinct verbs as listed in Table 3.1. However, the mapping from videos to verbs is not one-to-one. Due to polysemy, a verb may describe more than one action class, e.g. leaving an object on the table vs. leaving the scene. Due to synonymy, an action class may be described by more than one verb, e.g. lift vs. raise. An event described by one verb may contain a component action described by a different verb, e.g. picking up an object vs. touching an object. Many of the events are described by the combination of a verb with other constituents, e.g. have a conversation vs. have a heart attack. And many of the videos depict metaphoric extensions of verbs, e.g. take a puff on a cigarette. Because the mapping from videos to verbs is subjective, the corpus comes labeled with DARPA-collected human judgments in the form of a single present/absent label associated with each video paired with each of the 48 verbs, gathered using Amazon Mechanical Turk. We use these labels for both training and testing as described later.
3.2 Overall system architecture

The overall architecture of our system is depicted in Figure 3.1. We first apply detectors [41, 42] for each object class on each frame of each video. These detectors are biased to yield many false positives but few false negatives. The Kanade-Lucas-Tomasi (KLT) [43,44] feature tracker is then used to project each detection five frames forward to augment the set of detections and further compensate for false negatives in the raw detector output. A dynamic-programming algorithm [45] is then used to select an optimal set of detections that is temporally coherent with optical flow, yielding a set of object tracks for each video. These tracks are then smoothed and used to compute a time-series of feature vectors for each video to describe the relative and absolute motion of event participants. The person detections are then clustered based on part displacements to derive a coarse measure of human body posture in the form of a body-posture codebook. The codebook indices of person detections are then added to the feature vector. Hidden Markov Models (HMMs) are then employed as time-series classifiers to yield verb labels for each video [37, 46–49], together with the object tracks of the participants in the action described by that verb along with the roles they play. These tracks are then processed to produce nouns from object classes, adjectives from object properties, prepositional phrases from spatial relations, and adverbs and prepositional-phrase adjuncts from track properties. Together with the verbs, these are then woven into grammatical sentences. We describe each of the components of this system in detail below: the object detector and tracker in Section 3.2.1, the body-posture clustering and codebook in Section 3.2.2, the event classifier in Section 3.2.3, and the sentential-description component in Section 3.2.4.

3.2.1 Object detection and tracking

We employ detection-based tracking as described in Section 2 of a parallel submission (id: 568). In detection-based tracking an object detector is applied to each frame of a video to yield a set of candidate detections which are composed into tracks
Fig. 3.1. The overall architecture of our system for producing sentential descriptions of video.
by selecting a single candidate detection from each frame that maximizes temporal coherency of the track. Felzenszwalb et al. detectors are used for this purpose. Detection-based tracking requires biasing the detector to have high recall at the expense of low precision to allow the tracker to select boxes to yield a temporally coherent track. This is done by depressing the acceptance thresholds. To prevent massive over-generation of false positives, which would severely impact run time, we limit the number of detections produced per-frame to 12.

Two practical issues arise when depressing acceptance thresholds. First, it is necessary to reduce the degree of non-maximal suppression incorporated in the Felzenszwalb et al. detectors. Second, with the star detector [42], one can simply decrease the single trained acceptance threshold to yield more detections with no increase in computational complexity. However, we prefer to use the star cascade detector [41] as it is far faster. With the star cascade detector, though, one must also decrease the trained root- and part-filter thresholds to get more detections. Doing so, however, defeats the computational advantage of the cascade and significantly increases detection time. We thus train a model for the star detector using the standard procedure on human-annotated training data, sample the top detections produced by this model with a decreased acceptance threshold, and train a model for the star cascade detector on these samples. This yields a model that is almost as fast as one trained by the star cascade detector on the original training samples but with the desired bias in acceptance threshold.

The Y1 corpus contains approximately 70 different object classes that play a role in the depicted events. Many of these, however, cannot be reliably detected with the Felzenszwalb et al. detectors that we use. We trained models for 25 object classes that can be reliably detected, as listed in Table 3.2. These object classes account for over 90% of the event participants. Person models were trained with approximately 2000 human-annotated positive samples from C-D1 while nonperson models were trained with approximately 1000 such samples. For each positive training sample, two negative training samples were randomly generated from the same frame
Table 3.2

Trained models for object classes and their mappings to (a) nouns, (b) restrictive adjectives, and (c) size adjectives.

<table>
<thead>
<tr>
<th>(a)</th>
<th>bag→bag</th>
<th>bench→bench</th>
<th>bicycle→bicycle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>big-ball→ball</td>
<td>cage→cage</td>
<td>cardboard-box→box</td>
</tr>
<tr>
<td></td>
<td>cart→cart</td>
<td>car→car</td>
<td>chair→chair</td>
</tr>
<tr>
<td></td>
<td>dog→dog</td>
<td>door→door</td>
<td>ladder→ladder</td>
</tr>
<tr>
<td></td>
<td>mailbox→mailbox</td>
<td>microwave→microwave</td>
<td>motorcycle→motorcycle</td>
</tr>
<tr>
<td></td>
<td>person-crouch→person</td>
<td>person-down→person</td>
<td>person→person</td>
</tr>
<tr>
<td></td>
<td>skateboard→skateboard</td>
<td>small-ball→ball</td>
<td>suv→SUV</td>
</tr>
<tr>
<td></td>
<td>table→table</td>
<td>toy-truck→truck</td>
<td>tripod→tripod</td>
</tr>
<tr>
<td></td>
<td>truck→truck</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b)</th>
<th>cardboard-box→cardboard</th>
<th>person→upright</th>
<th>person-crouch→crouched</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>person-down→prone</td>
<td>toy-truck→toy</td>
<td></td>
</tr>
</tbody>
</table>

| (c) | big-ball→big | small-ball→small |

constrained to not overlap substantially with the positive samples. We trained three distinct person models to account for body-posture variation and pool these when constructing person tracks. The detection scores were normalized for such pooled detections by a per-model offset computed as follows: A (50 bin) histogram was computed of the scores of the top detection in each frame of a video. The offset is then taken to be the minimum of the value that maximizes the between-class variance [50] when bipartitioning this histogram and the trained acceptance threshold offset by a fixed, but small, amount (0.4).

We employed detection-based tracking for all 25 object models on all 749 videos in our test set. To prune the large number of tracks thus produced, we discard all tracks corresponding to certain object models on a per-video basis: those that exhibit high detection-score variance over the frames in that video as well as those whose detection-score distributions are neither unimodal nor bimodal. The parameters governing such
pruning were determined solely on the training set. The tracks that remain after this pruning still account for over 90% of the event participants.

### 3.2.2 Body-posture codebook

We recognize events using a combination of the motion of the event participants and the changing body posture of the human participants. Body-posture information is derived using the part structure produced as a by-product of the Felzenszwalb et al. detectors. While such information is far noisier and less accurate than fitting precise articulated models [51–55] and appears unintelligible to the human eye, as shown in Section 3.2.3, it suffices to improve event-recognition accuracy. Such information can be extracted from a large unannotated corpus far more robustly than possible with precise articulated models.

Body-posture information is derived from part structure in two ways. First, we compute a vector of part displacements, each displacement as a vector from the detection center to the part center, normalizing these vectors to unit detection-box area. The time-series of feature vectors is augmented to includes these part displacements and a finite-difference approximation of their temporal derivatives as continuous features for person detections. Second, we vector-quantize the part-displacement vector and include the codebook index as a discrete feature for person detections. Such pose features are included in the time-series on a per-frame basis. The codebook is trained by running each pose-specific person detector on the positive human-annotated samples used to train that detector and extract the resulting part-displacement vectors. We then pool the part-displacement vectors from the three pose-specific person models and employ hierarchical $k$-means clustering using Euclidean distance to derive a codebook of 49 clusters. Figure 3.2 shows sample clusters from our codebook. Codebook indices are derived using Euclidean distance from the means of these clusters.
Fig. 3.2. Sample clusters from our body-posture codebook.
3.2.3 Event classification

Our tracker produces one or more tracks per object class for each video. We convert such tracks into a time-series of feature vectors. For each video, one track is taken to designate the agent and another track (if present) is taken to designate the patient. During training, we manually specify the track-to-role mapping. During testing, we automatically determine the track-to-role mapping by examining all possible such mappings and selecting the one with the highest likelihood [46].

The feature vector encodes both the motion of the event participants and the changing body posture of the human participants. For each event participant in isolation we incorporate the following single-track features:

1. $x$ and $y$ coordinates of the detection-box center
2. detection-box aspect ratio and its temporal derivative
3. magnitude and direction of the velocity of the detection-box center
4. magnitude and direction of the acceleration of the detection-box center
5. normalized part displacements and their temporal derivatives
6. object class (the object detector yielding the detection)
7. root-filter index
8. body-posture codebook index

The last three features are discrete; the remainder are continuous. For each pair of event participants we incorporate the following track-pair features:

1. distance between the agent and patient detection-box centers and its temporal derivative
2. orientation of the vector from agent detection-box center to patient detection-box center

Our HMMs assume independent output distributions for each feature. Discrete features are modeled with discrete output distributions. Continuous features denoting linear quantities are modeled with univariate Gaussian output distributions, while those denoting angular quantities are modeled with von Mises output distributions.
For each of the 48 action classes, we train two HMMs on two different sets of time-series of feature vectors, one containing only single-track features for a single participant and the other containing single-track features for two participants along with the track-pair features. A training set of between 16 and 200 videos was selected manually from C-D1 for each of these 96 HMMs as positive examples depicting each of the 48 action classes. A given video could potentially be included in the training sets for both the one-track and two-track HMMs for the same action class and even for HMMs for different action classes, if the video was deemed to depict both action classes.

During testing, we generate present/absent judgments for each video in the test set paired with each of the 48 action classes. We do this by thresholding the likelihoods produced by the HMMs. By varying these thresholds, we can produce an ROC curve for each action class, comparing the resulting machine-generated present/absent judgments with the Amazon Mechanical Turk judgments. When doing so, we test videos for which our tracker produces two or more tracks against only the two-track HMMs while we test ones for which our tracker produces a single track against only the one-track HMMs.

We performed three experiments, training 96 different 200-state HMMs for each. Experiment I omitted all discrete features and all body-posture related features. Experiment II omitted only the discrete features. Experiment III omitted only the continuous body-posture related features. ROC curves for each experiment are shown in Figure 3.3, Figure 3.4 and Figure 3.5. Note that the incorporation of body-posture information, either in the form of continuous normalized part displacements or discrete codebook indices, improves event-recognition accuracy, despite the fact that the part displacements produced by the Felzenszwalb et al. detectors are noisy and appear unintelligible to the human eye.
Fig. 3.3. ROC curves for each of the 48 action classes for Experiment I omitting all discrete and body-posture-related features.
Fig. 3.4. ROC curves for each of the 48 action classes for Experiment II omitting only the discrete features.
Fig. 3.5. ROC curves for each of the 48 action classes for Experiment III omitting only the continuous body-posture-related.
3.2.4 Generating sentences

We produce a sentence from a detected action class together with the associated tracks using the templates from Table 3.3. In these templates, words in *italics* denote fixed strings, words in *bold* indicate the action class, X and Y denote subject and object noun phrases, and the categories Adv, PP\_endo, and PP\_exo denote adverbs and prepositional-phrase adjuncts to describe the subject motion. The processes for generating these noun phrases, adverbs, and prepositional-phrase adjuncts are described below. One-track HMMs take that track to be the agent and thus the subject. For two-track HMMs we choose the mapping from tracks to roles that yields the higher likelihood and take the agent track to be the subject and the patient track to be the object except when the action class is either approached or fled, the agent is (mostly) stationary, and the patient moves more than the agent.

Brackets in the templates denote optional entities. Optional entities containing Y are generated only for two-track HMMs. The criteria for generating optional adverbs and prepositional phrases are described below. The optional entity for received is generated when there is a patient track whose category is mailbox, person, person-crouch, or person-down.

We use adverbs to describe the velocity of the subject. For some verbs, a velocity adverb would be awkward:

\[ X \text{ slowly} \] had Y \hspace{2cm} *X had slow Y

Furthermore, stylistic considerations dictate the syntactic position of an optional adverb:

X jumped *slowly* over Y \hspace{2cm} X *slowly* jumped over Y
X *slowly* approached Y \hspace{2cm} *X approached slowly* Y

?X *slowly* fell \hspace{2cm} X fell *slowly*

The verb-phrase templates thus indicate whether an adverb is allowed, and if so whether it occurs, preferentially, preverbally or postverbally. Adverbs are chosen subject to three thresholds \( v^1_{\text{action class}}, v^2_{\text{action class}}, \) and \( v^3_{\text{action class}} \) determined empirically on a per-action-class basis: We select those frames from the subject track where
X [Adv] approached Y [PP_{exo}]
X [Adv] attached an object to Y
X buried Y
X caught Y [PP_{exo}]
X closed Y
X dropped Y
X [Adv] exchanged an object with Y
X fell [Adv] [because of Y] [PP_{endo}]
X flew [Adv] [PP_{endo}]
X gave an object to Y
X had Y
X [Adv] hauled Y [PP_{endo}]
X hit [something with] Y
X [Adv] kicked Y [PP_{endo}]
X [Adv] lifted Y
X opened Y
X picked Y up
X put Y down
X ran [Adv] [to Y] [PP_{endo}]
X [Adv] replaced Y
X [Adv] stopped [Y]
X [Adv] took an object from Y
X turned [PP_{endo}]
X was digging [with Y]

X arrived [Adv] [PP_{exo}]
X bounced [Adv] [PP_{endo}]
X [Adv] carried Y [PP_{endo}]
X [Adv] chased Y [PP_{endo}]
X [Adv] collided with Y [PP_{exo}]
X [Adv] entered Y [PP_{endo}]
X [Adv] exited Y [PP_{endo}]
X fled [Adv] [from Y] [PP_{endo}]
X [Adv] followed Y [PP_{endo}]
X got an object from Y
X handed Y an object
X held Y
X jumped [Adv] [over Y] [PP_{endo}]
X left [Adv] [PP_{endo}]
X [Adv] moved Y [PP_{endo}]
X [Adv] passed Y [PP_{exo}]
X [Adv] pushed Y [PP_{endo}]
X raised Y
X received [an object from] Y
X [Adv] snatched an object from Y
X [Adv] threw Y [PP_{endo}]
X touched Y
X walked [Adv] [to Y] [PP_{endo}]
X went [Adv] away [PP_{endo}]

Table 3.3
Sentential templates for the action classes indicated in bold.
the magnitude of the velocity of the box-detection center is above \(v_1^{\text{action class}}\). An optional adverb is generated by comparing the magnitude of the average velocity \(v\) of the subject track box-detection centers in these frames to the per-action-class thresholds:

\[
\begin{align*}
\text{quickly} & \quad v > v_2^{\text{action class}} \\
\text{slowly} & \quad v_1^{\text{action class}} \leq v \leq v_3^{\text{action class}}
\end{align*}
\]

We use prepositional-phrase adjuncts to describe the motion direction of the subject. Again, for some verbs, such adjuncts would be awkward:

\[
\begin{align*}
& \ast X \text{ had } Y \text{ leftward} \quad \ast X \text{ had } Y \text{ from the left} \\
& \text{Moreover, for some verbs it is natural to describe the motion direction endogenously, from the perspective of the subject, while for others it is more natural to describe the motion direction exogenously, from the perspective of the viewer:}
\end{align*}
\]

\[
\begin{align*}
X & \text{ fell leftward} \quad \text{X fell from the left} \\
X & \text{ chased } Y \text{ leftward} \quad \ast X \text{ chased } Y \text{ from the left} \\
\ast X & \text{ arrived leftward} \quad \text{X arrived from the left}
\end{align*}
\]

The verb-phrase templates thus indicate whether an adjunct is allowed, and if so whether it is preferentially endogenous or exogenous. The choice of adjunct is determined from the orientation of \(v\), as computed above and depicted in Figure 3.6(a,b). We omit the adjunct when \(v < v_1^{\text{action class}}\).

We generate noun phrases X and Y to refer to event participants according to the following grammar:

\[
\begin{align*}
\text{NP} & \rightarrow \text{themselves} \mid \text{itself} \mid \text{something} \mid \text{D A}^* \text{ N [PP]} \\
\text{D} & \rightarrow \text{the} \mid \text{that} \mid \text{some}
\end{align*}
\]

When instantiating a sentential template that has a required object noun-phrase Y for a one-track HMM, we generate a pronoun. A pronoun is also generated when the action class is entered or exited and the patient class is not car, door, suv, or truck. The anaphor themselves is generated if the action class is attached or raised, the anaphor itself if the action class is moved, and something otherwise.
Fig. 3.6. (a) Endogenous and (b) exogenous prepositional-phrase adjuncts to describe subject motion direction. (c) Prepositional phrases incorporated into subject noun phrases describing viewer-relative 2D spatial relations between the subject X and the reference object Y.
As described below, we generate an optional prepositional phrase for the subject noun phrase to describe the spatial relation between the subject and the object. We choose the determiner to handle coreference, generating the when a noun phrase unambiguously refers to the agent or the patient due to the combination of head noun and any adjectives,

\textit{The person jumped over the ball.}

\textit{The red ball collided with the blue ball.}

that for an object noun phrase that corefers to a track referred to in a prepositional phrase for the subject,

\textit{The person to the right of the car approached that car.}

\textit{Some person to the right of some other person approached that other person.}

and some otherwise:

\textit{Some car approached some other car.}

We generate the head noun of a noun phrase from the object class using the mapping in Table 3.2(a). Four different kinds of adjectives are generated: color, shape, size, and restrictive modifiers. An optional color adjective is generated based on the average HSV values in the eroded detection boxes for a track: \textit{black} when \( V \leq 0.2 \), \textit{white} when \( V \geq 0.8 \), one of \textit{red}, \textit{blue}, \textit{green}, \textit{yellow}, \textit{teal}, or \textit{pink} based on \( H \), when \( S \geq 0.7 \). An optional size adjective is generated in two ways, one from the object class using the mapping in Table 3.2(c), the other based on per-object-class image statistics. For each object class, a mean object size \( \bar{a}_{\text{object class}} \) is determined by averaging the detected-box areas over all tracks for that object class in the training set used to train HMMs. An optional size adjective for a track is generated by comparing the average detected-box area \( a \) for that track to \( \bar{a}_{\text{object class}} \):

\begin{align*}
\text{big} & \quad a \geq \beta_{\text{object class}} \bar{a}_{\text{object class}} \\
\text{small} & \quad a \leq \alpha_{\text{object class}} \bar{a}_{\text{object class}}
\end{align*}

The per-object-class cutoff ratios \( \alpha_{\text{object class}} \) and \( \beta_{\text{object class}} \) are computed to equally tripartition the distribution of per-object-class mean object sizes on the training set.
Optional shape adjectives are generated in a similar fashion. Per-object-class mean aspect ratios $\bar{r}_{\text{object class}}$ are determined in addition to the per-object-class mean object sizes $\bar{a}_{\text{object class}}$. Optional shape adjectives for a track are generated by comparing the average detected-box aspect ratio $r$ and area $a$ for that track to these means:

- **tall**: $r \leq 0.7\bar{r}_{\text{object class}} \land a \geq \beta_{\text{object class}}\bar{a}_{\text{object class}}$
- **short**: $r \geq 1.3\bar{r}_{\text{object class}} \land a \leq \alpha_{\text{object class}}\bar{a}_{\text{object class}}$
- **narrow**: $r \leq 0.7\bar{r}_{\text{object class}} \land a \leq \alpha_{\text{object class}}\bar{a}_{\text{object class}}$
- **wide**: $r \geq 1.3\bar{r}_{\text{object class}} \land a \geq \beta_{\text{object class}}\bar{a}_{\text{object class}}$

To avoid generating shape and size adjectives for unstable tracks, they are only generated when the detection-score variance and the detected aspect-ratio variance for the track are below specified thresholds. Optional restrictive modifiers are generated from the object class using the mapping in Table 3.2(b). Person-pose adjectives are generated from aggregate body-posture information for the track: object class, normalized part displacements, and body-posture codebook indices. We generate all applicable adjectives except for color and person pose. Following the Gricean Maxim of Quantity [56], we only generate color and person-pose adjectives if needed to prevent coreference of nonhuman event participants. Finally, we generate an initial adjective *other*, as needed to prevent coreference. Generating *other* does not allow generation of the determiner *the* in place of *that* or *some*. We order any adjectives generated so that *other* comes first, followed by size, shape, color, and restrictive modifiers, in that order.

For two-track HMMs where neither participant moves, a prepositional phrase is generated for subject noun phrases to describe the static 2D spatial relation between the subject X and the reference object Y from the perspective of the viewer, as shown in Figure 3.6(c).

### 3.3 Experimental results

We used the HMMs generated for Experiment III to compute likelihoods for each video in our test set paired with each of the 48 action classes. For each video, we
generated sentences corresponding to the three most-likely action classes. Figure 3.7 shows key frames from four videos in our test set along with the sentence generated for the most-likely action class. Human judges rated each video-sentence pair to assess whether the sentence was true of the video and whether it described a salient event depicted in that video. 26.7% (601/2247) of the video-sentence pairs were deemed to be true and 7.9% (178/2247) of the video-sentence pairs were deemed to be salient. When restricting consideration to only the sentence corresponding to the single most-likely action class for each video, 25.5% (191/749) of the video-sentence pairs were deemed to be true and 8.4% (63/749) of the video-sentence pairs were deemed to be salient. Finally, for 49.4% (370/749) of the videos at least one of the three generated sentences was deemed true and for 18.4% (138/749) of the videos at least one of the three generated sentences was deemed salient.

3.4 Conclusion

Integration of language and vision [60–64] and recognition of action in video [32–37, 46–49] have been of considerable interest for a long time. Yet we are unaware of any prior work that generates as rich sentential video descriptions as we describe here. Producing such rich descriptions requires determining event participants, the mapping of such participants to roles in the event, and their motion and properties. This is incompatible with common approaches to event recognition, such as spatiotemporal bags of words, spatiotemporal volumes, and tracked feature points that cannot determine such information. The approach presented here recovers the information needed to generate rich sentential descriptions by using detection-based tracking and a body-posture codebook. We demonstrated the efficacy of this approach on a corpus 749 videos.
The upright person to the right of the motorcycle went away leftward.

The person walked slowly to something rightward.

The narrow person snatched an object from something.

The upright person hit the big ball.

Fig. 3.7. Key frames from four videos in our test set along with the sentence generated for the most-likely action class.
4. SIMULTANEOUS OBJECT DETECTION, TRACKING, AND EVENT RECOGNITION

People recognize events in videos using the motion, changing pose, and mutual interaction of the objects that participate in those events. They are able to detect the event participants, track them over time, and recognize the event. Many approaches exist for performing each of these three tasks in isolation. Humans perform these tasks simultaneously; knowing that you are looking for a particular event makes it far likelier that you will detect the event participants as well as the detect that event. This allows humans to detect and track objects and recognize events that have very little supporting evidence. For example, in a video of a person using a screwdriver, the screwdriver might be only a few pixels across and always partially occluded; its identity is established by the context in which it is used despite the dearth of visual information.

We present a cognitive system which, like humans, performs object recognition, tracking, and event recognition simultaneously. We demonstrate that, as expected, such a system is able to out-perform its components when used in isolation. We introduce novel computational techniques that allow such a system to be efficient, with linear asymptotic complexity. We present a framework that can be extended to include other kinds of low-level features, such as the output of an object-segmentation system, as well as other kinds of high-level features, such as an entire natural-language understanding component.

The ultimate goal of cognitive-systems research is to build an artificial human, an agent that performs high-level inference from low-level perceptual input. Recognizing and reasoning about actions performed on objects as observed in video input requires detecting those objects, determining their type, and tracking their position over time. Most research in cognitive systems defers solution to the prob-
lem of object detection and tracking to the computer-vision community [65,66], assuming that community can, or someday will, deliver a module that can reliably and categorically detect and track objects, yielding symbolic representations like \( \text{ON(ball,ground)} \) and \( \text{HOLD(HAND(person),ball)} \). The cognitive-systems community assumes such representations are available as input to subsequent processing, for example, inferring \( \text{PICKUP(person, ball)} \) from a transition from \( \text{ON(ball,ground)} \) to \( \text{HOLD(HAND(person),ball)} \).

However, the computer-vision community has struggled long and hard, and been mostly unsuccessful in extracting reliable categorical information from images and video. It is now widely believed in the computer-vision community that it is unrealistic to expect to be able to produce such robust symbolic representations. The best we can hope for, at least at present, is noisy, vague, metric information like scored bounding boxes around objects, replete with false positives and negatives. The dilemma and challenge this poses for the cognitive-systems community is how to perform high-level inference from such information.

However, it also opens new possibilities: high-level inference can inform and assist low-level perception. In this paper, we show one concrete example of how this can work. Object detectors are unreliable; simply stringing together the top-ranked detection in each frame yields an incoherent track from which it is not possible to reliably detect events. If instead, we let the object detector produce multiple scored detections in each frame, and form a track by selecting detections across frames that optimizes a combination of low-level features, like detection scores, mid-level features, like the temporal coherence of a track, and high-level features, like the fact that a track depicts a known event, we can produce much better tracks that support much better event recognition. The essential characteristic of this method is that it lets the mid- and high-level information to override the noisy and unreliable low-level information. While we do this in this paper for only one small problem of tracking in the context of event recognition, we believe that the general approach of eschewing the assumption of the availability of robust categorical symbolic output of perceptual
processing in a purely bottom-up fashion and instead using high-level top-down information to constrain and assist low-level perception in its own non-symbolic terms is crucial to achieving the ultimate goal of cognitive systems.

Many common approaches to event recognition [2, 37, 46, 47] classify events based on their motion profile. This requires detecting and tracking the event participants. Adaptive approaches to tracking [67], such as Kalman filtering [68], suffer from three difficulties that impact their utility for event recognition. First, they must be initialized. One cannot initialize on the basis of motion since many event participants move only for a portion of the event, and sometimes not at all. Second, they exhibit drift and often must be periodically reinitialized to compensate. Third, they have difficulty tracking small, deformable, or partially-occluded objects as well as ones whose appearance changes dramatically. This is particularly of concern since many events, such as picking things up, involve humans interacting with objects that are sufficiently small for humans to grasp and where such interaction causes appearance change by out-of-plane rotation, occlusion, or deformation.

Detection-based tracking is an alternative approach that attempts to address these issues. In detection-based tracking, an object detector is applied to each frame of a video to yield a set of candidate detections which are composed into tracks by selecting a single candidate detection from each frame that maximizes temporal coherency of the track. However, current object detectors are far from perfect. On the PASCAL VOC Challenge, they typically achieve average-precision scores of 40% to 50% [69]. Directly applying such detectors on a per-frame basis would be ill-suited to event recognition. Since the failure modes include both false positives and false negatives, interpolation does not suffice to address this shortcoming. A better approach is to combine object detection and tracking with a single objective function that maximizes temporal coherency to allow object detection to inform the tracker and vice versa.

We can carry this approach even further and integrate event recognition with both object detection and tracking. One way to do this is to incorporate coherence with a target event model into the temporal-coherency measure. For example, a top-down
expectation of observing a *pick up* event can bias the object detector and tracker to search for event participants that exhibit the particular joint motion profile of that event: an object in close proximity to the agent, the object starting out at rest while the agent approaches the object, then the agent touching the object, followed by the object moving with the agent. Such information can also flow bidirectionally. Mutual detection of a *baseball bat* and a *hitting* event can be easier than detecting each in isolation or having a fixed direction of information flow.

The common internal structure and algorithmic organization, described in Section 4.1, of current object detectors [41], detection-based trackers [70], and HMM-based approaches to event recognition [46, 47, 71] facilitates a general approach to integrating these three components. We demonstrate an approach to integrating object detection and tracking (in Section 4.3), an approach to integrating tracking and event recognition (in Section 4.4), an approach to integrating object detection, tracking, and event recognition (in Section 4.5), and show how it improves each of the these three components in isolation (in Section 4.6). We demonstrate the effectiveness of this approach by qualitatively assessing its ability to track objects. Correct object tracks are crucial; without object tracks we have no hope of recognizing events. We show a number of examples where each component in isolation cannot correctly track the objects while combinations of the components can do so. Further, although prior detection-based trackers exhibit quadratic complexity, we show how such integration can be fast, with linear asymptotic complexity.

### 4.1 Detection-Based Tracking

The methods described in Sections 4.3, 4.4, and 4.5 extend a popular dynamic-programming approach to detection-based tracking. We review that approach here to set forth the concepts, terminology, and notation that will be needed to describe the extensions.
Detection-based tracking is a general framework where an object detector is applied to each frame of a video to yield a set of candidate detections which are composed into tracks by selecting a single candidate detection from each frame that maximizes temporal coherency of the track. This general framework can be instantiated with answers to the following questions:

1. What is the representation of a detection?
2. What is the detection source?
3. What is the measure of temporal coherency?
4. What is the procedure for finding the track with maximal temporal coherency?

We answer questions 1 and 2 by taking a detection to be a scored axis-aligned rectangle (box), such as produced by the Felzenszwalb et al. [41] object detector, though our approach is compatible with any method for producing scored axis-aligned rectangular detections. Let \( j \) be the index of a detection and \( b^j_t \) be a particular detection in frame \( t \) with score \( f(b^j_t) \). Let \( T \) denote the number of frames. A sequence of detections \( j = \langle j_1, \ldots, j_T \rangle \), one in each frame, denotes a track consisting of detections \( b^j_t \) in each frame \( t \). For example, in Figure 4.1, we might have the track \( j = \langle 1, \ldots, 1 \rangle \) that consists of detections \( \langle b^1_1, \ldots, b^T_T \rangle \). If the output of our detector is sorted by score, this track represents choosing the top-scoring detection in each frame.

We answer question 3 by formulating temporal coherency of a track \( j = \langle j_1, \ldots, j_T \rangle \) as

\[
\max_{j_1, \ldots, j_T} \left( \sum_{t=1}^{T} f(b^j_t) + \sum_{t=2}^{T} g(b^{j-1}_{t-1}, b^j_t) \right),
\]

(4.1)

where \( g \) scores the local temporal coherency between detections in adjacent frames. We define \( g \) to be the negative Euclidean distance between the center of \( b^j_t \) and the center of \( b^{j-1}_{t-1} \) projected forward one frame, though, as described below, our approach is compatible with a variety of functions discussed by Felzenszwalb et al. [72]. The forward projection internal to \( g \) can be computed in a variety of ways including by using optical flow and the Kanade-Lucas-Tomasi [44] feature tracker.
We answer question 4 by observing that Equation (4.1) can be optimized in polynomial time using dynamic-programming with the Viterbi [45] algorithm. Equation (4.2) maximizes over a combinatorial set of tracks whose size is exponential in the video length $T$. It does so in polynomial time by incrementally growing tracks forward from the beginning of the video towards the end of the video. It keeps the best track that ends in detection $j$ in frame $t$ in the memoization variable $\delta^t_j$ and inductively computes all of the $\delta$ values in frame $t + 1$ from those in frame $t$. This factors the optimization over an exponential set into a polynomial-time process:

\[
\begin{align*}
\text{for } j = 1 & \text{ to } J_t & \delta^1_j := f(b^1_j) \\
\text{for } t = 2 & \text{ to } T & \text{do} \left\{ \text{for } j = 1 \text{ to } J_t \text{ do} \delta^t_j := f(b^t_j) + \max_{j' = 1}^{J_{t-1}} (g(b^{t-1}_{j'}, b^t_j) + \delta^{t-1}_{j'}) \right\} \tag{4.2}
\end{align*}
\]

where $J_t$ is the number of detections in frame $t$. This leads to a lattice as shown in Figure 4.1.

Detection-based trackers exhibit less drift than adaptive approaches to tracking due to fixed target models. They also tend to perform better than simply picking the best detection in each frame [73]. The reason is that one can allow the detection
source to produce multiple candidates and use the combination of the detection score $f$ and the adjacent-frame temporal-coherency score $g$ to select the track. The essential attribute of detection-based tracking is that $g$ can overpower $f$ to assemble a more coherent track out of weaker detections. The nonlocal nature of Equation (4.1) can allow more-reliable tracking with less-reliable detection sources.

A crucial practical issue arises: How many candidate detections should be produced in each frame? Producing too few may risk failing to produce the desired detection that is necessary to yield a coherent track. In the limit, it is impossible to construct any track if even a single frame lacks any detections. The current state-of-the-art in object detection is unable to simultaneously achieve high precision and recall and thus it is necessary to explore the trade-off between the two [69]. A detection-based tracker can bias the detection source to yield higher recall at the expense of lower precision and rely on temporal coherency to compensate for the resulting lower precision. This can be done in at least three ways. First, one can depress the detection-source acceptance thresholds. One way this can be done with the Felzenszwalb et al. [41] detector is to lower the trained model thresholds. Second, one can pool the detections output by multiple detection sources with complementary failure modes. One way this can be done is by training multiple models for people in different poses. Third, one can use adaptive-tracking methods to project detections forward to augment the raw detector output, by including these as candidate detections in subsequent frames, and compensate for detection failure. This can be done in a variety of ways including by using optical flow and KLT. The essence of our paper is a more principled collection of approaches for compensating for low recall in the object detector.

A practical issue arises when pooling the detections output by multiple detection sources. It is necessary to normalize the detection scores for such pooled detections by a per-model offset. One can derive an offset by computing a histogram of scores of the top detection in each frame of a video and taking the offset to be the minimum of the value that maximizes the between-class variance when thresholding this bimodal histogram and the trained acceptance threshold offset by a small but fixed amount.
Fig. 4.2. The operation of a detection-based tracker. Three frames from the same video are shown, one in each column. The rows indicate successive information computed by the tracker for each frame. (a) Output of the detection sources, biased to yield false positives. (b) The top-scoring output of the detection source. Note that the top-scoring detection does not track the person or the object as the video progresses. (c) Augmenting the output of the detection sources with forward-projected detections. (d) The optimal tracks selected by the Viterbi algorithm track both the person and the object.

The operation of a detection-based tracker is illustrated in Figure 4.2. This example demonstrates several things of note. First, reliable tracks are produced despite an unreliable detection source. Second, the optimal track contains detections with suboptimal score. Row (b) demonstrates that selecting the top-scoring detection does not yield a temporally-coherent track. Third, forward-projection of detections from the second to third column in row (c) compensates for the lack of raw detections for the person in the third column of row (a).
Detection-based tracking runs in time $O(TJ^2)$ on videos of length $T$ with $J$ detections per frame. In practice, the run time is dominated by the detection process and the dynamic-programming step. Limiting $J$ to a small number speeds up the tracker considerably while minimally impacting track quality.

A further optimization is possible. The Felzenszwalb et al. [41] detector processes the input image to extract edge information before applying a filter to detect objects. It extracts histogram-of-oriented-gradient (HOG) features which represent the distribution of the edge orientations in patches of the image. In order to detect objects of different scales, this edge information is computed for different scalings of the input image. This forms a HOG pyramid, where the top is computed by down-scaling the image and the bottom is computed by up-scaling the image. Since this same HOG pyramid is used when detecting any object class, we further improve the speed of our method by factoring out the computation of the HOG pyramid and reusing it when running multiple object classes.

4.2 Evaluation of Detection-Based Tracking

We evaluated detection-base tracking using the year-one (Y1) corpus produced by DARPA for the Mind’s Eye program. These videos are provided at 720p@30fps and range from 42 to 1727 frames in length, with an average of 438.84 frames, and depict people interacting with a variety of objects to enact common English verbs.

Four Mind’s Eye teams (University at Buffalo, SBU; Stanford Research Institute, SRI; University of California Berkeley, UCB; University of Southern California, USC) independently produced human-annotated tracks for different portions of Y1. We used these sources of human-annotated tracks to evaluate the performance of detection-based tracking by computing human-human intercoder agreement between all pairs of the four sources of human-annotated tracks and human-machine intercoder agreement between a detection-based tracker and all four of these sources. Since each
Table 4.1
(a) The number of videos in common, (b) the mean overlap, and (c) the standard deviation in overlap between each pair of annotation sources.

<table>
<thead>
<tr>
<th>N</th>
<th>SBU</th>
<th>SRI</th>
<th>UCB</th>
<th>USC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>8</td>
<td>20</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SRI</td>
<td>8</td>
<td>1201</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>UCB</td>
<td>20</td>
<td>1201</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>USC</td>
<td>8</td>
<td>95</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>us</td>
<td>48</td>
<td>1254</td>
<td>1829</td>
<td>360</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>µ</th>
<th>SBU</th>
<th>SRI</th>
<th>UCB</th>
<th>USC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>0.76</td>
<td>0.68</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>SRI</td>
<td>0.76</td>
<td>0.55</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>UCB</td>
<td>0.68</td>
<td>0.55</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>USC</td>
<td>0.59</td>
<td>0.59</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>us</td>
<td>0.54</td>
<td>0.40</td>
<td>0.35</td>
<td>0.43</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>σ</th>
<th>SBU</th>
<th>SRI</th>
<th>UCB</th>
<th>USC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>0.06</td>
<td>0.14</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>SRI</td>
<td>0.06</td>
<td>0.27</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>UCB</td>
<td>0.14</td>
<td>0.27</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>USC</td>
<td>0.10</td>
<td>0.16</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>us</td>
<td>0.26</td>
<td>0.24</td>
<td>0.23</td>
<td>0.20</td>
</tr>
</tbody>
</table>

(c)
team annotated different portions of Y1, each such intercoder-agreement measure was computed only over the $N$ videos shared by each pair, as reported in Table 4.1(a).

The overall mean and standard deviation measures, reported in Table 4.1(b, c), indicate that the mean human-human overlap is only marginally greater than the mean human-machine overlap by about one standard deviation. This suggests that improvement in tracker performance is unlikely to lead to significant improvement in event-recognition performance and any subsequent processing that depends on event recognition such as generating sentences [2].

4.3 Combining Object Detection and Tracking

While detection-based tracking is resilient to low precision, it requires perfect recall; it cannot generate a track through a frame that has no detections and it cannot generate a track through a portion of the field of view which has no detections, regardless of how good the temporal coherence of the resulting track would be. This brittleness means that any detection source employed will have to significantly over-generate detections to achieve near-perfect recall. This has a downside; although the Viterbi algorithm has linear complexity in the number of frames, it is quadratic in the number of detections per frame. This drastically limits the number of detections that can reasonably be processed, leading to the necessity of tuning the thresholds on the detection sources. We have developed a novel mechanism to eliminate the need for a threshold and track every possible detection, at every position and scale in the image, in time linear in the number of detections and frames. At the same time, our approach eliminates the need for forward projection since every detection is already present. Our approach involves simultaneously performing object detection and tracking, optimizing the joint object-detection and temporal-coherency score.

Our general approach is to compute the distance between pairs of detection pyramids for adjacent frames, rather than using $g$ to compute the distance between pairs of individual detections. These pyramids represent the set of all possible detections
at all locations and scales in the associated frame. Employing a distance transform makes this process linear in the number of location and scale positions in the pyramid. Many detectors, such as that of Felzenszwalb et al., use such a scale-space representation of frames to represent detections internally even though they might not output such. Our approach requires instrumenting such a detector to provide access to this internal representation.

At a high-level, the Felzenszwalb et al. detector learns a forest of HOG [74] filters for each object class. Detection proceeds by applying each HOG filter at every position in an image pyramid followed by computing the optimal displacements at every position in that image pyramid, thereby creating a new pyramid, the detection pyramid. Finally, the detector searches the detection pyramid for high-scoring detections and extracts those above a threshold.

The detector employs a dynamic-programming algorithm to efficiently compute the optimal part displacements for the entire image pyramid. This algorithm is very similar to the Viterbi algorithm. It is made tractable by the use of a generalized distance transform [72] that allows it to scale linearly with the number of image-pyramid positions. Given a set $G$ of points (which in our case denotes an image pyramid), a distance metric $d$ between pairs of points $p$ and $q$, and an arbitrary function $\phi : G \rightarrow \mathbb{R}$, the generalized distance transform $D_\phi(q)$ computes

$$D_\phi(q) = \min_{p \in G} (d(p, q) + \phi(q))$$

in linear time for certain distance metrics including squared Euclidean distance.

Instead of extracting and tracking just the thresholded detections, one can directly track all detections in the entire pyramid simultaneously by defining a distance measure between detection pyramids for adjacent frames and performing the Viterbi tracking algorithm on these pyramids instead of sets of detections in each frame. To allow comparison between detections at different scales in the detection pyramid, we convert the detection pyramid to a rectangular prism by scaling the coordinates of the detections at scale $s$ by $\pi(s)$, chosen to map the detection coordinates back to the
coordinate system of the input frame. We define the distance between two detections, \( b \) and \( b' \), in two detection pyramids as a scaled squared Euclidean distance,

\[
d(b_{sys}, b'_{x'y's'}) = (\pi(s)x - \pi(s')x')^2 + (\pi(s)y - \pi(s')y')^2 + \alpha(s - s')^2,
\]

(4.3)

where \( x \) and \( y \) denote the original image coordinates of a detection center at scale \( s \). Nominally, detections are boxes. Two such can be compared using a four-dimensional distance metric. However, with a detection pyramid, the aspect ratio of detections is fixed, reducing this to a three-dimensional distance metric. The coefficient \( \alpha \) in the distance metric weights a difference in detection area differently than detection position.

The above amounts to replacing detections \( b_j^t \) with \( b^t_{sys} \), lattice values \( \delta_j^t \) with \( \delta^t_{sys} \), and Equation (4.2) with:

\[
\text{for } x = 1 \text{ to } X \text{ do } \{ \text{for } y = 1 \text{ to } Y \text{ do } \{ \text{for } s = 1 \text{ to } S \text{ do } \delta^t_{sys} := f(b^t_{sys}) \} \} \text{ (4.4)}
\]

\[
\text{for } t = 2 \text{ to } T \text{ do for } x = 1 \text{ to } X \text{ do for } y = 1 \text{ to } Y \text{ do for } s = 1 \text{ to } S \text{ do } \delta^t_{sys} := f(b^t_{sys}) + \max_{x', y', s'} (g(b^t_{sys} - b^t_{sys}), b^t_{sys} + \delta^t_{sys})
\]

(4.4)

where \( X \) and \( Y \) denote the image size and \( S \) denotes the maximal scale.

The above formulation allows us to employ the generalized distance transform as an analog to \( g \) in Equation (4.1), although it restricts consideration of \( g \) to be squared Euclidean distance rather than Euclidean distance. We avail ourselves of the fact that the generalized distance transform operates independently on each of the three dimensions \( x \), \( y \), and \( s \) in order to incorporate \( \alpha \) into Equation (4.3). While linear-time use of the distance transform restricts the form of \( g \), it places no restrictions on the form of \( f \).

One way to view the above is that the vector of \( \delta_j^t \) for all \( 1 \leq j \leq J_t \) from Equation (4.2) is being represented as a pyramid and the loop

\[
\text{for } j = 1 \text{ to } J_t \text{ do } \delta_j^t := f(b_j^t) + \max_{j' = 1}^{J_t} (g(b_{j'}^{t-1}, b_j^t) + \delta_{j'}^{t-1})
\]

(4.5)
is being performed as a linear-time construction of a generalized distance transform rather than a quadratic-time nested pair of loops. Another way to view the above is that it generalizes the notion of a detection pyramid from representing per-frame detections $b_{xys}$ at three-dimensional pyramid positions $\langle x, y, s \rangle$ to representing per-video detections $b'_{xys}$ at four-dimensional pyramid positions $\langle x, y, s, t \rangle$ and finding a sequence of per-video detections for $1 \leq t \leq T$ that optimizes a variant of Equation (4.1): 

$$
\max_{x_1, \ldots, x_T, y_1, \ldots, y_T, s_1, \ldots, s_T} \left( \sum_{t=1}^{T} f(b_{xtys}^t) + \sum_{t=2}^{T} g(b_{xtys}^{t-1}, b_{xtys}^t) \right)
$$

(4.6)

This combination of the detector and the tracker is performing simultaneous detection and tracking by integrating information between these two processes. Previously, the tracker was affected by the detector but the detector was unaffected by the tracker: potential low-scoring but temporally-coherent detections would not even be generated by the detector despite the fact that they would yield good tracks. The detector no longer chooses which detections to produce but instead scores all detections at every position and scale. Thus the tracker is able to choose among any possible detection. Such tight integration of higher- and lower-level information will be revisited when integrating event models into this framework.

### 4.4 Combining Tracking and Event Recognition

It is popular to use hidden Markov models (HMMs) to perform event recognition [2, 37, 46, 47]. When doing so, the log likelihood of a video conditioned on an event model is

$$
\log \sum_{k_1, \ldots, k_T} \exp \left( \sum_{t=1}^{T} h(k_t, b_{jt}^t) + \sum_{t=2}^{T} a(k_{t-1}, k_t) \right),
$$

where $k_t$ denotes the state of the HMM for frame $t$, $h(k, b)$ denotes the log probability of generating a detection $b$ conditioned on being in state $k$, $a(k', k)$ denotes the log probability of transitioning from state $k'$ to $k$, and $j_t^*$ denotes the index of the detection produced by the tracker in frame $t$. This log likelihood can be computed with
the forward algorithm [71], which is analogous to the Viterbi algorithm. Maximum likelihood, the standard approach to using HMMs for classification, selects the event model that maximizes the likelihood of an observed event. However, one can instead select the model with the maximum a posteriori (log) probability,

$$\max_{k_1, \ldots, k_T} \left( \sum_{t=1}^{T} h(k_t, b^*_t) + \sum_{t=2}^{T} a(k_{t-1}, k_t) \right), \quad (4.7)$$

which can be computed with the Viterbi algorithm. The advantage of doing so is that one can combine the Viterbi algorithm used for detection-based tracking with the Viterbi algorithm used for event recognition.

In particular, we can combine Equation (4.1) with Equation (4.7) to yield a unified cost function

$$\max_{j_1, \ldots, j_T, k_1, \ldots, k_T} \left( \sum_{t=1}^{T} f(b^*_t) + \sum_{t=2}^{T} g(b^*_{t-1}, b_t^*) + \sum_{t=1}^{T} h(k_t, b_t^*) + \sum_{t=2}^{T} a(k_{t-1}, k_t) \right) \quad (4.8)$$

that computes the joint MAP of the best possible track and the best possible state sequence by replacing \( j_t^* \) with \( j_t \) inside nested quantification. This too can be computed with the Viterbi algorithm by taking the lattice values \( \delta^t_{jk} \) to be indexed by the detection index \( j \) and the state \( k \), forming the cross product of the tracker lattice nodes and the event lattice nodes:

\[
\begin{align*}
\text{for } j = 1 \text{ to } J_1 & \text{ do } \{ \text{for } k = 1 \text{ to } K \text{ do } \delta^1_{jk} := f(b^*_j) + h(k, b^*_j) \} \\
\text{for } t = 2 \text{ to } T & \\
\text{do for } j = 1 \text{ to } J_t & \\
\text{do for } k = 1 \text{ to } K & \\
\text{do } \delta^t_{jk} := f(b^*_j) + h(k, b^*_j) + \max_{j'} \max_{k'} \left( g(b^*_{j'-1}, b^*_j) + a(k', k) + \delta^{t-1}_{j'k'} \right) \\
\end{align*}
\]

(4.9)

This finds the optimal path through a graph where the nodes at every frame represent the cross product of the detections and the HMM states.

Doing so performs simultaneous tracking and event recognition. The event recognizer described earlier was affected by the tracker but the tracker was unaffected by the event recognizer: potential low-scoring tracks would not even be generated
by the tracker despite the fact that they would yield a high MAP estimate for some event class. The tracker no longer chooses which tracks to produce but instead scores all tracks. Thus, the event recognizer can choose among all possible tracks. This amounts to a different kind of temporal-coherency measure that is tuned to specific events. Such a measure might otherwise be difficult to achieve without top-down information from the event recognizer. For example, applying this method to a video of a running person, along with an event model for running, will be more likely to compose a track out of person detections that has high velocity and low change in direction.

Processing each frame \( t \) with the algorithm in Equation (4.9) is quadratic in \( J_t K \). This can be problematic since \( J_t K \) can be large. As before, we can make this linear in \( J_t \) using a generalized distance transform. One can make this linear in \( K \) for suitable state-transition functions \( a \) [75].

Two practical issues arise when applying this method. First, one can factor Equation (4.10) as Equation (4.11):

\[
J_t - 1 \max_{j' = 1}^{J_t - 1} \max_{k' = 1}^K \left( g(b_{j'}^{t-1}, b_j^t) + a(k', k) + \delta_{t,j'k'}^{t-1} \right) \tag{4.10}
\]

\[
J_t - 1 \max_{j' = 1}^{J_t - 1} \left( g(b_{j'}^{t-1}, b_j^t) + \max_{k' = 1}^K \left( a(k', k) + \delta_{t,j'k'}^{t-1} \right) \right) \tag{4.11}
\]

This is important because the computation of \( g(b_{j'}^{t-1}, b_j^t) \) might be expensive, as it involves a projection of \( b_{j'}^{t-1} \) forward one frame (e.g. using optical flow or KLT). Second, when applying this method to multiple event models, the same factorization can be extended to cache the computation of \( g(b_{j'}^{t-1}, b_j^t) \) across different event models as this term does not depend on the event model.

Note that the algorithm in Equation (4.9) does not technically recognize events. Rather, it assumes that the event class is known. That is our central claim: top-down knowledge of the event class being observed can help the low-level perceptual component in the task of producing tracks that are needed to recognize the event. But there is no chicken-and-egg problem; even if one did not know what event was being observed, one can simply run multiple simultaneous instances of Algorithm 4.9,
one for each event class, and select the highest-scoring result. Doing such will perform simultaneous tracking and event recognition.

4.5 Combining Object Detection, Tracking, and Event Recognition

One can combine the methods of Sections 4.3 and 4.4 to optimize a cost function,

$$\max_{x_1,\ldots,x_T,\ y_1,\ldots,y_T,\ s_1,\ldots,s_T,\ k_1,\ldots,k_T} \left( \sum_{t=1}^{T} f(b^t_{x_t y_t s_t}) + h(k_t, b^t_{x_t y_t s_t}) + \sum_{t=2}^{T} g(b^{t-1}_{x_{t-1} y_{t-1} s_{t-1}}, b^t_{x_t y_t s_t}) + a(k_{t-1}, k_t) \right),$$

(4.12)

that combines Equation (4.6) with Equation (4.8) by forming a large Viterbi lattice with values $\delta_{t, k, y, s, k}^t$.

One practical issue arises when applying the above method. In Equation (4.12), $h$ is a function of $b^t_{x_t y_t s_t}$, the detection in the current frame. This allows the HMM event model to depend on static object characteristics such as position, shape, and pose. However, many approaches to event recognition using HMMs use temporal derivatives of such characteristics to provide object velocity and acceleration information [46,47]. This means that $h$ must also be a function of $b^{t-1}_{x_{t-1} y_{t-1} s_{t-1}}$, the detection in the previous frame. In addition, $h$ must also be incorporated into the generalized distance transform in order to efficiently compute the optimal track as described in Section 4.3. This restricts our choice of $h$, the features we compute, to those with known generalized distance transforms like the squared Euclidean distance and the $L_1$ norm.

This combined formulation performs simultaneous object detection, tracking, and event recognition, integrating information across all three tasks. Without such information integration, the object detector is unaffected by the tracker, which is in turn unaffected by the event model. With such integration, the event model can influence the tracker and both of these can influence the object detector.

This is important because current object detectors cannot reliably detect small, deformable, or partially-occluded objects. Moreover, current trackers also fail to track
such objects. Information from the event model can focus the object detector and tracker on those particular objects that participate in a specified event. An event model for recognizing an agent picking an object up can bias the object detector and tracker to search for an object that exhibits a particular profile of motion relative to the agent, namely where the object is in close proximity to the agent, the object starts out being at rest while the agent approaches the object, then the agent touches the object, followed by the object moving with the agent.

A traditional view of the relationship between object detection and event recognition suggests that one recognizes a hammering event, in part, because one detects a hammer. Our unified approach inverts the traditional view, suggesting that one can detect a hammer, in part, by recognizing a hammering event. Furthermore, a strength of our approach is that such relationships are not encoded explicitly, do not have to be annotated in the training data for the event models, and are learned automatically as part of learning the parameters of the different event models. This is to say that the relationship between a person and the objects they manipulate can be learned from the co-occurrence of tracks in the training data, rather than from manually annotated symbolic relationships.

4.6 Demonstration

We demonstrate the effectiveness of the approach presented in this paper by qualitatively assessing its ability to track objects. Figure 4.3 demonstrates improved performance of simultaneous object detection and tracking (c), as computed by the methods in Section 4.3, over object detection (a) and tracking (b) in isolation, as computed by the methods in Section 4.1. This happens for different reasons: motion blur, even for large objects, can lead to poor detection results and hence poor tracks, small objects are difficult to detect and track, and integration can improve detection and tracking of deformable objects, such as a person transitioning from an upright pose to sitting down.
Fig. 4.3. Improved performance of simultaneous object detection and tracking. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector using models for people, motorcycles, and balls. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous object detection and tracking, as described in Section 4.3.

Fig. 4.4. Improved performance of simultaneous tracking and event recognition. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous tracking and event recognition, as described in Section 4.4.
Fig. 4.5. Improved performance of simultaneous object detection, tracking, and event recognition. A single frame from each of four different videos is shown. Each column depicts the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 4.1. (c) Tracks produced by simultaneous object-detection, tracking, and event recognition, as described in Section 4.5.
Figure 4.4 demonstrates improved performance of simultaneous tracking and event recognition (c), as computed by the methods in Section 4.4, over tracking (b) in isolation, as computed by the methods in Section 4.1. These results were obtained with object and event models that were trained independently. Object models were trained on isolated frames using the standard Felzenszwalb et al. training software, while the event models were trained using tracks produced by the detection-based-tracking method in Section 4.1 and human-labeled event occurrences. Articulated appearance change and motion blur make it difficult to track the person running with detection-based tracking alone. Imposing the prior of detecting running biases the tracker to find the desired track.

Figure 4.5 demonstrates improved performance of simultaneous object detection, tracking, and event recognition (c), as computed by the methods in Section 4.5, over object detection (a) and tracking (b) in isolation, as computed by the methods in Section 4.1. As before, these results were obtained with object and event models that were trained independently.

4.7 Related Work

Detection-based tracking using dynamic programming has a long history [70, 76], as do motion-profile-based approaches to event recognition using HMMs [37, 46, 47]. Moreover, there have been attempts to integrate object detection and tracking [77, 78], tracking and event recognition [79]; and object detection and event recognition [80–82]. However, we are unaware of prior work that integrates all three and does so in a fashion that efficiently finds a global optimum to a simple unified cost function.

We have demonstrated a general framework for simultaneous object detection, tracking, and event recognition. Many object detectors can naturally be transformed into trackers by introducing time into their cost functions, thus tracking every possible detection in each frame. Furthermore, the distance transform can be used to reduce the complexity of doing so from quadratic to linear. The common internal
structure and algorithmic organization of object detection, detection-based tracking, and event recognition further allows an HMM-based approach to event recognition to be incorporated into the general dynamic-programming approach. This facilitates multidirectional information flow where not only can object detection influence tracking and, in turn, event recognition, event recognition can influence tracking and, in turn object detection.

One may be tempted to ask whether the methods of this paper are overkill and unnecessary. Perhaps one can use purely symbolic categorical methods to track objects and recognize events. However, as pointed out at the beginning of this chapter, robust production of symbolic representations from images and video is beyond the current state of the art in computer vision and is likely to remain so for a very long time, if not forever. While there has been some attempt to build purely symbolic systems for tracking objects and recognizing events, such systems tend to be highly tuned to particular environments and scenarios rather than automatically learning the event models such as is possible with our approach as it models events as HMMs [83]. Moreover, they are also limited to processing a small number of largely accurate hypotheses meaning that they cannot employ any current object-detection methods as developed in the computer-vision community for these generate huge numbers of inaccurately scored and ranked hypotheses, replete with false positives and negatives [65].

Authors of these systems highlight some of their deficiencies. Their behavior is difficult to understand and reason about because they are composed of multiple complex interacting modules. Information flow between high-level processing and low-level perception must be explicitly coded, usually in the form of a separate error-reasoning module. Not only are such error-reasoning modules difficult to construct, they themselves are difficult to understand and reason about, which exacerbates the difficulty of predicting the behavior of the entire system. All of these problems are, in part, a consequence of the fact that such systems do not optimize an explicit cost function which by its nature integrates high-level and low-level information in a transparent fashion and instead rely on a complex ad hoc architecture.
4.8 Conclusions and Future Work

The approach presented in this paper integrates top-down and bottom-up information in order to improve the quality of tracks and the reliability of event recognition, but it does have a number of limitations and failure modes. The object detectors employed are unreliable. They may completely fail to detect an object, i.e. there may be false negatives. In addition, the scores produced by the detectors are unreliable assessments of the presence or absence of an object in the field of view, i.e. there may be false positives which may manifest themselves as higher-ranked misdetections that steer the tracker away from lower-ranked correct detections. The methods described in this paper are unable to compensate for the former but attempt to compensate for the latter. However, they are not always successful when objects enter or leave the field of view. When the desired object is not in the field of view, the tracker may incorrectly track a background object, making it difficult to switch to tracking the correct object as it enters the scene due to the temporal-coherency score. One way to address this might be to use a sliding temporal window over the video. Another alternative might be to change the coherence measure to be weaker when the detections are weaker. This approach, like many other trackers, also suffers from interchanging tracks between objects that are near each other. For example, when two people pass each other and overlap in the image, the track may switch from one person to the other. This problem is somewhat ameliorated by the use of an event model to guide tracking but is not entirely eliminated. One way to address this may be to split longer tracks at track-intersection points and stitch them back together using an appearance model.

The approach presented here can combine object recognition, tracking, and event recognition but event recognition only models a verb. This opens the door for incorporating an entire natural-language-understanding component in place of the event-recognition component as a richer source of top-down constraint over the tracker, simultaneously tracking multiple objects whose collective motion is consistent with
a rich sentential description, rather than tracking a single object whose motion is consistent with just a verb. To do this, we assume that one can represent meanings of individual words as some form of temporally-changing constraint over the relative positions and motions of one or more objects. One can then use traditional symbolic natural-language-understanding techniques to parse a multi-word sentence and use the resulting parse tree to guide a process of compositional semantics that combines the meanings of individual words into an overall constraint over the collective set of objects that fill roles in the event described by the sentence. The dynamic-programming methods from Section 4.1 can be extended to fill these roles with tracks and construct said tracks incrementally in polynomial time in a fashion that jointly optimizes detection score, temporal coherence, and sentential score instead of event-recognition score. When making the leap from verbs to sentences, the semantic representations may require richer features than what is possible to compute efficiently with the parabolic-envelope generalized distance transform that we currently employ [72]. Other kinds of generalized distance transforms, such as those based on the Legendre transform [84], may address this problem.

A system built around sentences rather than verbs could, given a grammar, search the space of sentences and generate an appropriate sentence for a video. Given a complex video with multiple simultaneous events, it could find one particular event that matches a sentential query. It could search a long video or video database to find a particular video clip that depicts a complex sentential query. Furthermore, it may be possible to co-train object and event models by combining Baum-Welch [85, 86] with the training procedure for the object models [41]. Ultimately, one can imagine learning the meanings of individual words—nouns that correspond to object detectors, verbs that correspond to event recognizers, adjectives that correspond to meta-level property modifiers of object detectors, prepositions that correspond to spatial-relation detectors, and adverbs that correspond to meta-level property modifiers of event recognizers—from video annotated with whole sentences. Doing such would constitute
a model of how children learn the meanings of words in their native language from their combined perceptual and linguistic environments.

Other related tasks such as speech recognition can also be incorporated into the framework presented here. Most current speech recognizers, like the approach taken here to event recognition, also employ HMMs. Such speech recognizers create a lattice of hidden states at each frame and employ the same dynamic-programming algorithm [45] to recover the optimal hidden state sequence. In the same way that we take the cross product between the hidden states of the tracker and that of the event-recognition component, we can extend this cross product to include the hidden states of the speech recognizer. The remainder of the algorithm would remain largely unchanged. This would allow a system to resolve ambiguity in a spoken word or phrase which refers to a video while simultaneously integrating information from speech, the object detector, the tracker, and the event recognizer.

The framework we have presented bridges three separate research areas in order to fashion a cognitive system that brings to bear the human ability to integrate information across multiple sources. Like humans, the system can be biased, or primed, to detect one particular event, or set of events. Like humans, even when faced with little evidence, or as is the case for object detection, very poor detectors, this approach is still able to detect objects and recognize events. This approach is quite general; it provides a framework for inference and reasoning across modalities all the way from low-level vision to high-level cognition.
5. SAYING WHAT YOU’RE LOOKING FOR:
LINGUISTICS MEETS VIDEO SEARCH

Video search engines lag behind text search engines in their wide use and performance. This is in part because the most attractive interface for finding videos remains a natural-language query in the form of a sentence but determining if a sentence describes a video remains a difficult task. This task is difficult for a number of different reasons: unreliable object detectors which are required to determine if nouns occur, unreliable event recognizers which are required to determine if verbs occur, the need to recognize other parts of speech such as adverbs or adjectives, and the need for a representation of the semantics of a sentence which can faithfully encode the desired natural-language query. We propose an approach which simultaneously addresses all of the above problems. Approaches to date generally attempt to independently address the various aspects that make this task difficult. For example, they attempt to separately find videos that depict nouns and videos that depict verbs and essentially take the intersection of these two sets of videos. This general approach of solving these problems piecemeal cannot represent crucial distinctions between otherwise similar input queries. For example, if you search for *The person rode the horse* and for *The horse rode the person*, existing systems would give the same result for both queries as they each contain the same words, but clearly the desired output for these two queries is very different. We develop a holistic approach which both combines tracking and word recognition to address the problems of unreliable object detectors and trackers and at the same time uses compositional semantics to construct the meaning of a sentence from the meaning of its words in order to make crucial but otherwise subtle distinctions between otherwise similar sentences. Given a grammar and an input sentence, we parse that sentence and, for each video clip in a corpus, we si-
multaneously track all objects that the sentence refers to and enforce the constraint that all tracks must be described by the target sentence using an approach called the *sentence tracker*. Each video is scored by the quality of its tracks, which are guaranteed by construction to depict our target sentence, and the final score correlates with our confidence that the resulting tracks correspond to real objects in the video. We produce a score for every video-sentence pair and return multiple video hits ordered by their scores.

In a recent survey of video retrieval, Hu et al. [87] note that work on semantic video search focuses on detecting nouns and verbs, as well as using language to search already-existing video annotation. The state of the art in image retrieval is similar [88]. The approach presented here, by design, would fare poorly on still images as it uses the fact that the input is a video in order to mutually inform and constrain object detection, tracking, and event recognition. Unlike earlier approaches, the work presented here requires no pre-existing annotations aside from a tiny training corpus.

Retrieving clips or frames in which a query object occurs has been addressed both using query-by-example and object detection. Sivic and Zisserman [89] present a statistical local-feature approach to query-by-example. A bounding box is placed around a target object, and frames in which that object occurs are retrieved. Unlike the work presented here, this search is not performed using an object detector, but instead relies on detecting regions with similar statistical features. Moreover, it does not exploit the fact that the input is a video, and instead treats each frame of the video independently. Yu et al. [90] detect and track a single object, a soccer ball, and recognize actions being performed on that object during a soccer match. They extract gross motion features by examining the position and velocity of the object in order to recognize events and support a small number of domain-specific actions limited to that specific single object. Anjulan and Canagarajah [91] track stable image patches to extract object tracks over the duration of a video and group similar tracks into object classes. Without employing an object detector, these methods cannot search a collection of videos for a particular object class but instead must search by example.
Byrne et al. [92] employ statistical local features, such as Gabor features, to perform object detection. These do not perform as well as more recent object detectors on standard benchmarks such as PASCAL VOC. Sadeghi and Farhadi [93] recognize objects, in images, in the context of their spatial relations, using an object detector. They train an object detector not just for an object class, but for a combination of multiple interacting objects. This allows them to detect more complex scenarios, such as a person riding a horse, by building targeted object detectors. Moreover, knowledge of the target scenario improves the performance of the object detector. Similarly, in our work, knowledge about the query improves the performance of each of the individual detectors for each of the words in the query. But their approach differs fundamentally from the one presented here because it is not compositional in nature. In order to detect *The person rode the horse*, one must train on examples of exactly that entire sentence, whereas in the work presented here, independent detectors for *person*, *horse*, and *ride* combine together to encode the semantics of the sentence and to perform retrieval of a sentence that may never have occurred in the training set.

Prior work on verb detection does not integrate with work on object detection. Chang et al. [96] find one of four different highlights in basketball games using hidden Markov models and the expected structure of a basketball game. They do not detect objects but instead classify entire presegmented clips, are restricted to a small number of domain-specific actions, and support only single-word queries. Event recognition is a popular subarea of computer vision but has remained limited to single-word queries [97–101]. We will avail ourselves of such work later [102] to show that the work presented here both allows for richer queries and improves on the performance of earlier approaches.

Prior work on more complex queries involving both nouns and verbs essentially encodes the meaning of a sentence as a conjunction of words, largely discarding the compositional semantics of the sentence reflected by sentence structure. Christel et al. [103], Worring et al. [104], and Snoek et al. [105] present various combinations of
text search, verb retrieval, and noun retrieval, and essentially allow for finding videos which are at the intersection of multiple search mechanisms. Aytar et al. [106] rely on annotating a video corpus with sentences that describe each video in that corpus. They employ text-based search methods which given a query, a conjunction of words, attempt to find videos of similar concepts as defined by the combination of an ontology and statistical features of the videos. Their model for a sentence is a conjunction of words where higher-scoring videos more faithfully depict each individual word but the relationship between words is lost. None of these methods attempt to faithfully encode the semantics of a sentence and none of them can encode the distinction between *The person hit the ball* and *The ball hit the person*.

In what follows, we describe a system, which unlike previous approaches, allows for a natural-language query of video corpora which have no human-provided annotation. Given a sentence and a video corpus, we retrieve a ranked list of videos which are described by that sentence. We show a method for constructing a lexicon with a small number of parameters, which are reused among multiple words, making training those parameters easy and ensuring the system need not be shown positive examples of every word in the lexicon. We present a method for combining models for individual words into a model for an entire sentence and for recognizing that sentence while simultaneously tracking objects in order to score a video-sentence pair. To demonstrate this approach, we run 141 natural-language queries on a corpus of 10 full-length Hollywood movies using a grammar which includes nouns, verbs, adjectives, adverbs, spatial-relation prepositions, and motion prepositions. This is the first approach which can search for complex queries which include multiple phrases, such as prepositional phrases, and modifiers, such as adverbs.

5.1 Tracking

We begin by describing the operation of a detection-based tracker on top of which the sentence tracker will be constructed. To search for videos which depict a sentence,
we must first track objects that participate in the event described by that sentence. Tracks consist of a single detection per frame per object. To recover these tracks, we employ detection-based tracking. An object detector is run on every frame of a video, producing a set of axis-aligned rectangles along with scores which correspond to the strength of each detection. We employ the Felzenszwalb et al. [41,42] object detector, specifically the variant developed by Song et al. [107]. There are two reasons why we need a tracker and cannot just take the top-scoring detection in every frame. First, there may be multiple instances of the same object in the field of view. Second, object detectors are extremely unreliable. Even on standard benchmarks, such as the PASCAL Visual Object Classes (VOC) Challenge, even the best detectors for the easiest-to-detect object classes achieve average-precision scores of 40% to 50% [69]. We overcome both of these problems by integrating the intra-frame information available from the object detector with inter-frame information computed from optical flow.

We expect that the motion of correct tracks agrees with the motion of the objects in the video which we can compute separately and independently of any detections using optical flow. We call this quantity the motion coherence of a track. In other words, given a detection corresponding to an object in the video, we compute the average optical flow inside that detection, forward-project the detection along that vector, and expect to find a strong detection in the next frame at that location. We formalize this intuition into an algorithm which finds an optimal track given a set of detections in each frame. For each frame $t$ in a video of length $T$, each detection $j$ has an associated axis-aligned rectangle $b^t_j$ and score $f(b^t_j)$ and each pair of detections in adjacent frames has an associated motion coherence score $g(b_{j-1}^{t-1}, b^t_j)$. We formulate the score of a track $j = \langle j^1, \ldots, j^T \rangle$ as

$$\max_{j^1, \ldots, j^T} \sum_{t=1}^{T} f(b^t_{j^t}) + \sum_{t=2}^{T} g(b_{j-1}^{t-1}, b^t_j) \quad (5.1)$$

where we take $g$, the motion coherence, to be a nonincreasing function of the squared Euclidean distance between the center of $b_{j-1}^{t-1}$ and the center of $b^t_j$ projected one
frame forward. While the number of possible tracks is exponential in the number of frames in the video, Eq. 5.1 can be maximized in time linear in the number of frames and quadratic in the number of detections per frame using dynamic programming, the Viterbi [45] algorithm.

The development of this tracker follows that of Barbu et al. [3] which presents additional details of such a tracker, including an extension which allows generating multiple tracks per object class using non-maxima suppression. That earlier tracker used the raw detection scores from the Felzenszwalb et al. [41, 42] object detector. These scores are difficult to interpret because the mean and variance of scores varies by object class making it difficult to decide whether a detection is strong. To get around this problem, we pass all detections through a sigmoid \( \frac{1}{1+\exp(-b(t-a))} \) whose center, \( a \), is the model threshold and whose scaling factor \( b \), is 2. This normalizes the score to the range \([0, 1]\) and makes scores more comparable across models. In addition, the motion coherence score is also passed through a similar sigmoid, with center 50 and scale \(-1/11\).

### 5.2 Word recognition

Given tracks, we want to decide if a word describes one or more of those tracks. This is a generalization of event recognition, generalizing the notion of an event from verbs to other parts of speech. To recognize if a word describes a collection of tracks, we extract features from those tracks and use those features to formulate the semantics of words. Word semantics are formulated in terms of finite state machines (FSMs) which accept one or more tracks. Figure 5.2 provides an overview of the FSMs used in Sections 5.5.2 and 5.5.3, rendered as regular expressions. This approach is a limiting case of that taken by Barbu et al. [108] which used hidden Markov models (HMMs) to encode the semantics of verbs. In essence, our FSMs are unnormalized HMMs with binary transition matrices and binary output distributions. This allows the same recognition mechanism as that used by Barbu et al. [108] to be employed here.
We construct word meanings in two levels. First, we construct 18 predicates, shown in Figure 5.1, which accept one or more detections. We then construct word meanings for our lexicon of 15 words, shown in Figure 5.2, as regular expressions which accept tracks and are composed out of these predicates. This two-level construction allows sharing low-level features and parameters across words. All words share the same predicates which are encoded relative to 9 parameters: \textbf{far}, \textbf{close}, \textbf{stationary}, \textbf{Δclosing}, \textbf{Δangle}, \textbf{Δpp}, \textbf{Δquickly}, \textbf{Δslowly}, and \textbf{overlap}. These parameters are learned from a tiny number of positive and negative examples that cover only a fraction of the words in the lexicon. To make predicates independent of the video resolution, detections are first rescaled relative to a standard resolution of 1280 × 720, otherwise parameters such as \textbf{far} would need to vary with resolution.

Given a regular expression for a word, we can construct a nondeterministic FSM, with one accepting state, whose allowable transitions are encoded by a binary transition matrix $h$, giving score zero to allowed transitions and $-\infty$ to disallowed transitions, and whose states accept detections which agree with the predicate $a$, again with the same score of zero or $-\infty$. With this FSM, we can recognize if a word describes a track $\langle \hat{j}_1, \ldots, \hat{j}_T \rangle$, by finding

$$\max_{k^1, \ldots, k^T} \sum_{t=1}^{T} h(k^t, b^t_{jt}) + \sum_{t=2}^{T} a(k^{t-1}, k^t)$$

(5.2)

where $k^1$ through $k^{T-1}$ range over the set of states of the FSM and $k^T$ is the singleton set containing the accepting state. If this word describes the track, the score yielded by Eq. 5.2 will be zero. If it does not, the score will be $-\infty$. The above formulation can be generalized to multiple tracks and is the same as that used by Barbu et al. [3]. We find accepting paths through the lattice of states again using dynamic programming, the Viterbi algorithm. Note that this method can be applied to encode not just the meaning of verbs but also of other parts of speech. For example, the meaning of a static concept, such as a preposition like \textit{left-of} that encodes a temporally invariant spatial relation, can be encoded as a single-state FSM whose output predicate encodes that relation. The meaning of a dynamic concept, such as a preposition like \textit{towards}
that encodes temporally variant motion, can be encoded in a multi-state FSM much like a verb. It is well known in linguistics that the correspondence between semantic classes and parts of speech is flexible. For example, some verbs, like hold, encode static concepts, while some nouns, like wedding, encode dynamic concepts. Employing a uniform but powerful representation to encode the meaning of all parts of speech supports this linguistic generality and further allows a single but powerful mechanism to build up the semantics of sentences from the semantics of words. This same general mechanism admits some resiliency to noisy input by allowing one to construct FSMs with ‘garbage’ states that accept noisy segments. We avail ourselves of this capacity by incorporating true$^+$ into many of the word FSMs in Figure 5.2.

5.3 Sentence tracker

Our ultimate goal is to search for videos described by a natural-language query in the form of a sentence. The framework developed so far falls short of supporting this goal in two ways. First, as we attempt to recognize multiple words that constrain a single track, it becomes unlikely that the tracker will happen to produce an optimal track which satisfies all the desired predicates. For example, when searching for a person that is both running and doing so leftward, the chance that there may be a single noisy frame that fails to satisfy either the running predicate or the leftward predicate is greater than for a single-word query. Second, a sentence is not a conjunction of words, even though a word is represented here as a conjunction of features, so a new mechanism is required to faithfully encode the compositional semantics of a sentence as reflected in its structure. Intuitively, we must encode the mutual dependence in the sentence The tall person rode the horse so that the person is tall, not the horse, and the person is riding the horse, not vice versa.

We address the first point by biasing the tracker to produce tracks which agree with the predicates that are being enforced. This may result in the tracker producing tracks which have to consist of lower-scoring detections, which decreases the probability that
Fig. 5.1. Predicates which accept detections, denoted by \( a \) and \( b \), formulated around 9 parameters. These predicates are used for the second and third experiment, Sections 5.5.2 and 5.5.3. The function project projects a detection forward one frame using optical flow. The functions flow-orientation and flow-magnitude compute the angle and magnitude of the average optical-flow vector inside a detection. The function \( a_{cx} \) accesses the \( x \) coordinate of the center of a detection. The function \( a_{width} \) computes the width of a detection. The functions \( \cup \) and \( \cap \) compute the area of the union and intersection of two detections respectively. The function \( |\cdot|^\circ \) computes angular separation. Words are formed as regular expressions over these predicates.
Fig. 5.2. Regular expressions which encode the meanings of each of the 15 words or lexicalized phrases in the lexicon used for the second and third experiment, Sections 5.5.2 and 5.5.3. These are composed from the predicates shown in Figure 5.1. We use an extended regular-expression syntax where an exponent of \{t,\} allows a predicate to hold for t or more frames.
these tracks correspond to real objects in the video. This is not a concern as we will present the users with results ranked by their tracker score. In essence, we pay a penalty for forcing a track to agree with the enforced predicates and the ultimate rank order is influenced by this penalty. The computational mechanism that enables this exists by virtue of the fact that our tracker and word recognizer have the same internal representation and algorithm, namely, each finds optimal paths through a lattice of scored detections, \( f(b^t_{jt}) \), for the tracker, or states scored by their output predicate, \( h(k^t, b^t_{jt}) \), for the word recognizer, and each weights the links in that lattice by a score, the motion coherence, \( g(b^{t-1}_{jt-1}, b^t_{jt}) \), for the tracker, and state-transition score, \( a(k^{t-1}, k^t) \), for the word recognizer. We simultaneously find the track \( j^1, \ldots, j^T \) and state sequence \( k^1, \ldots, k^T \) that optimizes a joint objective function

\[
\max_{j^1, \ldots, j^T} \max_{k^1, \ldots, k^T} \left( \sum_{t=1}^{T} f(b^t_{jt}) + \sum_{t=2}^{T} g(b^{t-1}_{jt-1}, b^t_{jt}) + \sum_{t=1}^{T} h(k^t, b^t_{jt}) + \sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \quad (5.3)
\]

which ensures that, unless the state sequence for the word FSM leads to an accepting state, the resulting aggregate score will be \(-\infty\). This constrains the track to depict the word and finds the highest-scoring one that does so. Intuitively, we have two lattices, a tracker lattice and a word-recognizer lattice, and we find the optimal path, again with the Viterbi algorithm, through the cross-product of these two lattices. This cross-product lattice construction is shown in Figure 5.3.

The above handles only a single word, but given a sentential query we want to encode its semantics in terms of multiple words and multiple trackers. We parse an input sentence with a grammar, shown in Figure 5.5, and extract the number of participants and the track-to-role mapping. Each sentence that describes an event has a number of roles that must be filled with entities that serve as participants in that event. For example, in the sentence \textit{The person rode the horse quickly away from the other horse}, there are three participants, one person and two horses, and each of the three participants plays a different role in the sentence, \textit{agent} (the entity performing the action, in this case the person), \textit{patient} (the entity affected by the action, in this case the first horse), and \textit{goal} (the destination of the action, in this case the
second horse). Each word in this sentence refers to a subset of these three different participants, as shown in Figure 5.4, and words that refer to multiple participants, such as *ride*, must be assigned participants in the correct argument order to ensure that we encode *The person rode the horse* rather than *The horse rode the person*. We use a custom natural-language parser which takes as input a grammar, along with the arity and thematic roles of each word, and computes a track-to-role mapping: which participants fill which roles in which words. We employ the same mechanism as described above for simultaneous word recognition and tracking, except that we instantiate one tracker for each participant and one word recognizer for each word.

The thematic roles, $\theta^w_n$, map the $n$th role in a word $w$ to a tracker. Figure 5.4 displays an overview of this mapping for a sample sentence. Trackers are shown in red, word recognizers are shown in blue, and the track-to-role mapping is shown using the arrows. Given a sentential query that has $W$ words, $L$ participants, and track-to-role mapping $\theta^w_n$, we find a collection of tracks $\langle j^1_1, \ldots, j^T_1 \rangle, \ldots, \langle j^1_L, \ldots, j^T_L \rangle$, one for each participant, and accepting state sequences $\langle k^1_1, \ldots, k^T_1 \rangle, \ldots, \langle k^1_W, \ldots, k^T_W \rangle$, one for each word, that optimizes a joint objective function

$$\max_{j^1_1, \ldots, j^T_L} \max_{k^1_1, \ldots, k^T_W} \left( \sum_{l=1}^{L} \sum_{t=1}^{T} f(b^t_{j^t_l}) + \sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}_l}, b^t_{j^t_l}) + \sum_{w=1}^{W} \sum_{t=1}^{T} h_w(k^t_w, b^t_{j^t_{\theta^w_n}}, b^{t-1}_{j^{t-1}_{\theta^w_n}}) + \sum_{t=2}^{T} a_w(k^{t-1}_w, k^t_w) \right)$$  \hspace{1cm} (5.4)$$

where $a_w$ and $h_w$ are the transition matrices and predicates for word $w$, $b^t_{j^t_l}$ is a detection in the $t$th frame of the $l$th track, and $b^t_{j^t_{\theta^w_n}}$ connects a participant that fills the $n$th role in word $w$ with the detections of its tracker. Since the aggregate score will be $-\infty$ if even a single word-recognizer score would be $-\infty$, this equation constrains the subcollection of tracks that play roles in each of the words in the sentence to satisfy the semantic conditions for that word, collectively constraining the entire collection of tracks for all of the participants to satisfy the semantic conditions for the entire sentence. Further, it finds that collection of tracks with maximal tracker score sum.
Fig. 5.3. Tracker lattices are used to track each participant. Word lattices constructed from word FSMs for each word in the sentence recognize collections of tracks for participants that exhibit the semantics of that word as encoded in the FSM. We take the cross product of multiple tracker and word lattices to simultaneously track participants and recognize words. This ensures that the resulting tracks are described by the desired sentence.

In essence, for each word, we take the cross product of its word lattice with all of the tracker lattices that fill roles in that word, collectively taking a single large cross product of all word and tracker lattices in a way that agrees with the track-to-role mapping, and find the optimal path through the resulting lattice. This allows us to employ the same computational mechanism, the Viterbi algorithm, to find this optimal node sequence. The resulting tracks will satisfy the semantics of the input sentence, even if this incurs a penalty by having to choose lower-scoring detections.
The tall person rode the horse quickly leftward away from the other horse.

Fig. 5.4. Different sentential queries lead to different cross products. The sentence is parsed and the role of each participant, show in red, is determined. A single tracker lattice is constructed for each participant. Words and lexicalized phrases, shown in blue, have associated word lattices which encode their semantics. The arrows between words and participants represent the track-to-role mappings, $\theta$, required to link the tracker and word lattices in a way that faithfully encodes the sentential semantics. Some words, like determiners, shown in grey, have no semantics beyond determining the parse tree and track-to-role mapping. The dashed lines indicate that the argument order is essential for words which have more than one role. In other words, predicates like ride and away from are not symmetric. Detection sources are shown in black, in this case two object detectors. The tracker associated with each participant has access to all detection sources, hence the bipartite clique between the trackers and the detection sources.
S → NP VP
NP → D [A] N [PP]
D → an | the
A → blue | red
N → person | horse | backpack | trash can | chair | object
PP → P NP
P → to the left of | to the right of
VP → V NP [Adv] [PP_M]
V → approached | lead | carried | picked up | put down | rode
Adv → quickly | slowly
PP_M → P_M NP | from the left | from the right
P_M → towards | away from

Fig. 5.5. The grammar for sentential queries used in Section 5.5. Items in black are shared between all experiments. Items in red are exclusive to the first experiment, Section 5.5.1. Items in blue are exclusive to the second and third experiments, Sections 5.5.2 and 5.5.3.
5.4 Retrieval

We employ the mechanisms developed above to perform video retrieval given a sentential query. Given a corpus of videos, we retrieve short clips which depict a full sentence from these longer videos. To do so, we use the fact that the sentence tracker developed above scores a video-sentence pair. The sentence-tracker score sums the scores of the participant trackers and the scores of the word recognizers. As explained in the previous section, the word-recognizer score, and thus the sum of all such, is either 0 or $-\infty$. This means that the aggregate sentence-tracker score will be $-\infty$ if no tracks can be found which depict the query sentence. Otherwise, it will simply be the tracker-score sum. This score indicates our confidence in how well a video depicts a query sentence, the better the tracker score the more confident we can be that the tracks correspond to real objects in the video. The fact that those tracks are produced at all ensures that they depict the query sentence. We take this correlation between score and whether a video depicts a sentence to perform video retrieval. Given a corpus of clips, we run the sentence tracker with the query sentence on each clip. Clips are then ranked by their sentence-tracker score.

The above approach retrieves short clips from a corpus of such. Our ultimate goal, however, is to take, as input, videos of arbitrary length and find short clips which depict the query sentence from these longer videos. The sentence tracker is able to find a single instance of an event in a long video because, as shown in Figure 5.2, word meanings have garbage states of unbounded length prepended and appended to them. But this would produce a single detected event for each long video instead of potentially many short clips for each input video. To produce multiple clips, we split all input videos into short, several second long, clips and produce a corpus of clips on which we perform video retrieval. The exact clip length is unimportant as long as the query sentences can be fully depicted in the clip length because, as noted above, the sentence tracker will find shorter events in a longer clip. This also motivates the use of fixed-length clips as all words in our chosen lexicon depict short events.
One downside of this is the inability to detect events that straddle clip boundaries. To address this problem, we segment input videos into short but overlapping clips, ensuring that each clip boundary is contained within another clip.

Given the corpus of clips to be searched, the other piece of information required is the query sentence. The sentence is first parsed according to the grammar shown in Figure 5.5. The grammar presented is context-free and the sentence is parsed using a standard recursive-descent parser. Note that the grammar presented here is infinitely recursive. Noun phrases optionally contain prepositional phrases which contain other noun phrases. For one example one might say: *The person to the left of the horse to the right of the person to the left of the horse*. The words shown in Figure 5.2 require arguments and each of these arguments has one of five thematic roles: agent, patient, referent, goal, and source. The parse tree, together with the role information, are used to determine the number of participants and which participants fill which roles in the event described by the sentence. This provides the track-to-role mapping, \( \theta \), in Eq. 5.4.

An alternate method for producing this mapping would be to employ a more general natural-language parser such as the Stanford Parser [109]. Given an input sentence such as *The person rode the horse toward the horse*, the Stanford Parser produces the following dependencies:

\[
\begin{align*}
det(person-2, The-1) \\
nsubj(rode-3, person-2) \\
root(ROOT-0, rode-3) \\
det(horse-5, the-4) \\
dobj(rode-3, horse-5) \\
det(horse-8, the-7) \\
prep_toward(rode-3, horse-8)
\end{align*}
\]

which can also be used to construct the requisite track-to-role mapping. The output above correctly identifies three participants, *person-2, horse-5*, and *horse-8*. Note how the transitive verb *rode-3* distinguishes between its two arguments, identifying *person-
as its subject and horse-5 as its direct object. Using a general natural-language parser would allow a retrieval system to handle a much larger space of sentences and alleviate the need to specify the grammar and track-to-role mapping mechanism for each word. This approach would still require specification of the semantics of each word. We construct the exposition and experiments around the first approach, with a custom grammar and parser, to render the algorithm and its requirements more transparent.

The above procedure for searching a corpus of clips can be sped up significantly when searching the same corpus with multiple sentential queries. First, the object detections required for the sentence tracker are independent of the query sentence. In other words, the object detector portion of the lattice, namely the score, position, and optical flow for each detection, are unaffected by the query sentence even though the tracks produced are affected by it. This can be seen in Eq. 5.4 where neither \( f \) (the detection score), \( g \) (the motion coherence), nor either of their arguments depend on \( k \) (the lexical entry of a word), or \( \theta \) (the track to role mapping). This allows us to preprocess the video corpus and compute object detections and optical-flow estimates which can be reused with different sentential queries. This constitutes the majority of the runtime of the algorithm; object detection and optical-flow estimation are an order of magnitude slower than parsing and sentence-tracker inference.

The first speedup addressed how to decrease the computation for each clip in the corpus. The second addresses the fact that the resulting retrieval algorithm still requires inspecting every clip in the corpus to determine if it depicts the query sentence. We ameliorate this problem by first noting that the lexicon and grammar presented in Figs. 5.2 and 5.5 have no negation. This means that in order for a video to depict a sentence it must also depict any fragment of that sentence. By sentence fragment, we mean any subsequence of a word string that can be generated by any terminal or nonterminal in the grammar. For example, the sentence The person approached the horse quickly has sentence fragments person, horse, approached, approached the horse, quickly, and approached the horse quickly. Any video depicting
this entire sentence must also depict these fragments. Were our grammar to have
negation, this would not be true; a video depicting the sentence *The person did not
approach the horse* would not depict the fragment *approach the horse*. This leads to
an efficient algorithm for reusing earlier queries to speed up novel queries. Intuitively,
if you’ve already determined that nothing approaches a horse in a clip, nothing will
approach a horse quickly in that clip. In other words, one can parse the query sentence
and look through all previous queries, potentially queries of sentence fragments, to
see which queries form subtrees of the current query. All clips which have score $-\infty$
for these shorter queries can be eliminated from consideration when searching for
the longer query. This enables scaling to much larger video corpora by immediately
eliminating videos which cannot depict the query sentence.

5.5 Results

We present three experiments which test video retrieval using sentential queries.
The first employs a corpus of clips collected specifically for this task which facilitates
more complex queries with a larger number of nouns and participants while being
designed to stress the system by employing a multitude of nearly-identical query
sentences. The second and third employ a corpus of 10 full-length Hollywood movies
showing the ability of this approach to handle videos found in the wild and not filmed
specifically for this task.

5.5.1 The new$^3$ corpus

We first evaluate this approach on a corpus (called new$^3$) of 94 short clips shot
outdoors from a stationary camera. These clips show between one and two people
performing actions with one or two objects, selected from a collection of three objects,
all present in the field of view in every clip. The language fragment supported by
this corpus includes five nouns: one for each of the objects, one for people, and
a generic noun *object*. The later experiments on Hollywood movies only support
two nouns: *person* and *horse*, due to the fact that object detector performance is
much lower on this more challenging set of videos. Frames from sample videos in new\textsuperscript{3} are shown in Figure 5.7. To search this corpus using sentential queries we first formulate the semantics of a small fragment of English consisting of 17 lexical items (5 nouns, 2 adjectives, 4 verbs, 2 adverbs, 2 spatial-relation prepositions, and 2 motion prepositions). The grammar and lexicon for this fragment of English are presented in Figure 5.5 and consist of the black and red portions of the figure. The semantics of the words are formed similarly to Figs. 5.1 and 5.2. We form 21 query sentences, shown in Figure 5.6, including in these all possible minimal pairs which can be constructed using the given grammar while ignoring the infinite recursion in the noun phrase. By a minimal pair we mean two sentences which differ only in one part of speech. For example the sentences

\begin{quote}
The red object approached the chair.

The blue object approached the chair.
\end{quote}

form a minimal pair which differs only in the adjective in the subject noun phrase. This ensures a more difficult test where each part of speech in each query must be correctly interpreted.

We evaluate this approach by running each sentential query against the video corpus. Chance performance, the probability that a sentence is depicted by a randomly selected video, is 13.12\%. Given a sentential query, the top-scoring video for that sentence depicts that sentence 85.71\% of the time. This shows that we are able to successfully retrieve clips given sentential queries even when many queries form minimal pairs which are difficult to distinguish and even when we restrict ourselves to only the top hit.

5.5.2 Ten westerns

We further demonstrate the utility of this approach on a more challenging corpus composed of 10 Hollywood westerns: Black Beauty (Warner Brothers, 1994), The Black Stallion (MGM, 1979), Blazing Saddles (Warner Brothers, 1974), Easy Rider
The backpack approached the trash can.
The chair approached the trash can.
The red object approached the chair.
The blue object approached the chair.
The person to the left of the trash can put down an object.
The person to the right of the trash can put down an object.
The person put down the trash can.
The person put down the backpack.
The person carried the red object.
The person carried the blue object.
The person picked up an object to the left of the trash can.
The person picked up an object to the right of the trash can.
The person picked up an object.
The person put down an object.
The person picked up an object quickly.
The person picked up an object slowly.
The person carried an object towards the trash can.
The person carried an object away from the trash can.
The backpack approached the chair.
The red object approached the trash can.
The person put down the chair.

Fig. 5.6. The 21 sentential queries used in Section 5.5.1. Differences in corresponding minimal pairs are highlighted in red and green.
The person carried an object away from the trash can.

The person picked up an object to the left of the trash can.

Fig. 5.7. Sentential-query-based video search: returning the best-scoring video, in a corpus of 94 videos, for a given sentence.
(Columbia Pictures, 1969), The Good the Bad and the Ugly (Columbia Pictures, 1966), Hidalgo (Touchstone Pictures, 2004), National Velvet (MGM, 1944), Once Upon a Time in Mexico (Columbia Pictures, 2003), Seabiscuit (Universal Pictures, 2003), and Unforgiven (Warner Brothers, 1992). In total, this video corpus has 1187 minutes of video, roughly 20 hours. We temporally downsampled all videos to 6 frames per second but kept their original spatial resolutions which varied from $336 \times 256$ pixels to $1280 \times 544$ pixels with a mean resolution of $659.2 \times 332.8$ pixels. We split these videos into 37187 clips, each clip being 18 frames (3 seconds) long, while overlapping the previous clip by 6 frames. This overlap ensures that actions that might otherwise occur on clip boundaries will also occur as part of a clip. While there is prior work on shot segmentation [110] we did not employ it for two reasons. First, it complicates the system and provides an avenue for additional failure modes. Second, the approach taken here is able to find an event inside a longer video with multiple events. The only reason why we split the videos into clips is to return multiple hits.

We adopt the grammar from Figure 5.5, specifically the black and blue portions. This grammar allows for sentences that describe people interacting with horses, hence our choice of genre for the video corpus, namely westerns. A requirement for determining whether a video depicts a sentence, and the degree to which it depicts that sentence, is to detect the objects that might fill roles in that sentence. Previous work has shown that people and horses are among the easiest-to-detect objects, although the performance of object detectors, even for these classes, remains extremely low. To ensure that we did not test on the training data, we employed previously-trained object models that have not been trained on these videos but have instead been trained on the PASCAL VOC Challenge [69]. We use models trained by the UoCTTI_LSVM-MDPM team (the authors of Felzenszwalb et al. [41,42]) for the 2009 Challenge. On the 2009 Challenge, the person model achieves an AP score of 41.5% and the horse model achieves an AP score of 38.0%. We note that the improvement in AP scores for these object classes in subsequent years of the Challenge has been minor. We also require settings for the 9 parameters, shown in Figure 5.1, which are required to
X {approached Y {quickly, slowly} {from the left, from the right},
lead, rode} Y {quickly, slowly}
{leftward, rightward, towards, away from} Z

Fig. 5.8. The template used to generate the 141 query sentences where X,
Y, and Z are either person or horse. The template generates 204 sentences
out of which 63 are removed because they involve people riding people and
horses riding people or other horses for which no true positives exist in
our video corpus.

produce the predicates which encode the semantics of the words in this grammar. We
trained all 9 parameters simultaneously on only 3 positive examples and 3 negative
examples. Note that these training examples cover only a subset of the words in the
grammar but are sufficient to define the semantics of all words because this word sub-
set touches upon all the underlying parameters. Training proceeded by exhaustively
searching a small uniform grid, with between 3 and 10 steps per dimension, of all nine
parameter settings to find a combination which best classified all 6 training samples
which were then removed from the test set. Yu and Siskind et al. [111] present an
alternate strategy for training the parameters of a lexicon of words given a video
corpus.

We generated 204 sentences that conform to the grammar in Figure 5.5 from the
template shown in Figure 5.8. We eliminated the 63 queries that involve people riding
people and horses riding people or other horses, as our video corpus has no positive
examples for these sentences. This leaves us with 141 queries which conform to our
grammar. For each sentence, we scored every clip paired with that sentence and
return the top 10 best-scoring clips for that sentence. Each of these top 10 clips was
annotated by a human judge with a binary decision: is this sentence true of this clip?
In Figure 5.10(a), we show the average precision of the system over all 141 queries
on the top 10 hits for each query as a function of a threshold on the scores. As
the threshold nears zero, the system may return fewer than 10 results per sentence.
because it eliminates query results which are unlikely to be true positives. As the threshold tends to $-\infty$, the average precision across all top 10 clips for all sentences is 22.9%, and at its peak, the average precision is 72.4%. In Figure 5.10(b), we show the number of results returned per sentence, eliminating those results which have a score of $-\infty$ since that means that no tracks could be found which agree with the semantics of the sentence, On average, there are 7.96 hits per sentence, with standard deviation 3.61, and with only 14 sentences having no hits. In Figure 5.10(c), we show the number of correct hits per sentence. On average, there are 1.83 correct hits per sentence, with standard deviation 2.26, and with 80 sentences having at least one true positive.

We highlight the usefulness of this approach in Figure 5.11 where we show the top 6 hits for two similar queries: *The person approached the horse* and *The horse approached the person*.\(^1\) Hits are presented in order of score, with the highest scoring hit at the top and scores decreasing as one moves down. Note how the results for the two sentences are very different from each other and each sentence has 3 true positives and 3 false positives. With existing systems, both queries would provide the same hits as they treat sentences as conjunctions of words.

### 5.5.3 Comparison

We compare our results against a baseline method that employs the same approach that is used in state-of-the-art video-search systems. We do not compare against any particular existing system because no current system employs state-of-the-art object or event detectors and thus any such system would be severely handicapped in its inability to reliably detect people, horses, and the particular events we search for. Our baseline operates as follows. We first apply an object detector to each frame of every clip to detect people and horses. For comparison purposes, we employ the same

---

\(^1\) A video search engine that supports all 10 full-length Hollywood movies and all 141 sentential queries discussed in the text is available at [http://0xab.com/research/video-events/westerns.html](http://0xab.com/research/video-events/westerns.html).
object detector and pretrained models as used for the experiments in Section 5.5.2, including passing the raw detector score through the same sigmoid. We rank the clips by the average score of the top detection in each frame. If the query sentence contains only the word *person*, we rank only by the person detections. If the query sentence contains only the word *horse*, we rank only by the horse detections. If the query sentence contains both the words *person* and *horse*, we rank by the average of the top person and top horse detection in each frame. We then apply a binary event detector to eliminate clips from the ranking that do not depict the event specified by the entire query sentence. For this purpose, we employ a state-of-the-art event detector, namely that of Kuehne et al. [102]. We train that detector on six samples of each entire query sentence and remove those samples from the test set. We then report the top 10 ranked clips that satisfy the event detector and compare those clips against the top 10 clips produced by our method.

We compared our system against this baseline on three different sentential queries: *The person rode the horse*, *The person lead the horse*, and *The person approached the horse*. The results are summarized in Figure 5.9. Note that our approach yields significantly higher precision on each of the queries as well as higher overall average precision. Further note that this baseline system was trained on a total of 18 training samples: six samples for each of three query sentences. In contrast, our method was trained on a total six training samples. Not only was our method trained on one third as many training samples, our method can support all 141 distinct queries with its training set, while the baseline only supports three queries with its training set.

5.6 Discussion

As discussed at the beginning of this chapter, previous work falls into two categories: search by example and attribute-based approaches. In the former, a sample image or video is provided and similar images or videos are retrieved. Conventional event-recognition systems are of this type. They train models on collections of query
<table>
<thead>
<tr>
<th>Query</th>
<th>our TP</th>
<th>baseline TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>The person rode the horse.</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>The person lead the horse.</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>The person approached the horse.</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 5.9. A comparison between our approach and a baseline system constructed out of state-of-the-art components on the top 10 hits returned for various sentential queries.
Fig. 5.10. (a) Average precision of the top 10 hits for the 141 query sentences as a function of the threshold on the sentence-tracker score. Without a threshold, (b) the number of sentences with at most the given number of hits and (c) the number of sentences with at least the given number of correct hits.
Fig. 5.11. Frames from the top 6 hits for two sentential queries. True positives are shown in green and false positives in red. In both cases, half are true positives\textsuperscript{1}. The fact that the results are different shows that our method encodes the meaning of the entire sentence along with which object fills which role in that sentence.
clips and find the target clips which best match the trained model. In the limit, such systems find the target clips most-similar to a single query clip. Attribute-based approaches are usually applied to images, not videos. Such approaches, given a sentence or sentence fragment, extract the words from that sentence and use independent word models to score each image or video clip [112, 113]. Some variants of these approaches, such as that of Siddiquie et al. [114], learn correlations between multiple features and include feature detectors which are not present in the input query. Some systems present various combinations of the approaches described above such as those of Christel et al. [103], Worring et al. [104], and Snoek et al. [105].

None of the approaches described above link features in a way that is informed by the structure of the sentence, hence they are unable to support sentential queries. What we mean by this is they cannot show the key difference that we underscore in this work, the ability to encode the semantics of a query sentence with enough fidelity to differentiate between *The person rode the horse* and *The horse rode the person*. The baseline system we compare against in Section 5.5.3 was specifically designed to model the predominant current methodology, updated to use state-of-the-art object and event recognizers. Specifically, it modeled queries as bags of words with no reflection of argument structure.

In the experiments in Section 5.5.2, we report only true positives and the associated precision, not true negatives nor the associated recall. The reason is simple: reporting true negatives would require annotating the entire corpus of 37187 clips with truth values for all 141 queries, a monumental and tedious task. We only annotate the top ten hits for each of the 141 queries as to their truth value, allowing us only to report true positives. That raises a potential question: what is the chance that we may have missed potential hits for our queries. We note that movies have very different properties from surveillance video and standard action-recognition corpora. Most time is spent showing people engaged in dialog rather than performing actions. Thus we contend that the false negative rate is very low. Moreover, we contend that chance performance on this retrieval task is also very low. This is further supported
by the extreme low performance of the baseline from Section 5.5.3. Thus we contend that the underlying retrieval task is difficult and the performance of our method as described in Section 5.5.2 is good. Moreover, we have annotated negatives for the smaller experiment in Section 5.5.1 which allowed us to compute chance performance and demonstrate that our method far exceeds such.

In the future, one can imagine scaling our approach along a variety of axes: larger and more varied video corpora, a larger lexicon of nouns, verbs, adjectives, adverbs, and prepositions, and a more complex query grammar. Let us consider the advances needed to achieve such scaling.

Scaling the size of the video corpus is easy. For a fixed-size query language, processing time and space is linear in the corpus size. Further, such processing is trivially parallelizable and, as discussed in Section 5.4, many components of the process, such as object detection, can be precomputed and cached in a query-independent fashion. Moreover, as discussed in Section 5.4, results of earlier queries can be cached and used to speed up processing of later queries, potentially leading to reduction of the search complexity below linear time.

Scaling up to support a larger lexicon of nouns, largely depends on the state-of-the-art in object detection. While current methods appear to work well only for small numbers of object classes, recent work by Dean et al. [115] has shown that object detection may scale to far larger collections of objects. Since our method simply requires scored detections, it can avail itself of any potential future advances in object detection, including combining the results of multiple detection methods, potentially even for the same object class as part of the same object track.

Scaling up to support a larger lexicon of verbs also appears possible. Our approach performs event recognition on time series of feature vectors extracted from object tracks. This general approach has already been demonstrated to scale to 48 distinct event classes [108]. However, this can only be used for verbs and other parts of speech whose meanings are reflected in motion profile: the changing relative and absolute positions, velocities, and accelerations of the event participants. Scaling beyond this,
to encode the meanings of words like *sit*, *pour*, *build*, or *break*, or semantic distinctions like the difference between *abandon* and *leave* or between *follow* and *chase*, would require modeling facets of human perception and cognition beyond motion profile, such as body posture [116], functionality, intention, and physical processes.

Scaling sentence length and complexity requires lattices of greater width. The dynamic-programming algorithm which performs inference on the sentence-tracker lattice takes time quadratic in the width of the cross-product lattice. Unfortunately the width of this cross-product lattice increases exponentially in the number of participants and the query-sentence length. While this approach will not scale indefinitely, it is able to process our current queries with three participants and 6-14 words with an acceptable runtime, a few dozen seconds per clip. Scaling further will require either a faster dynamic-programming algorithm or inexact inference. Barbu et al. [3] present an algorithm which employs Felzenszwalb and Huttenlocher’s [72] generalized distance transform to perform inference in linear time in the lattice width, as opposed to quadratic time, for a one-word sentence tracker. Such an approach can be generalized to an entire sentence tracker but carries the added weight of restricting the form of the features that are extracted from tracks when formulating the per-state predicates in the event model. At present, the constant-factor overhead of this approach outweighs the reduced asymptotic complexity, but this may change with increased query-sentence complexity. Alternatively one might perform inexact inference using beam search to eliminate low-scoring lattice regions. Inexact inference might also employ sampling methods such as MCMC.

5.7 Conclusion

We have developed an approach to video search which takes as input a video corpus and a sentential query. It generates a list of results ranked by how well they depict the query sentence. This approach provides two novel video-search capabilities. First, it can encode the semantics of sentences compositionally, allowing it to
express subtle distinctions such as the difference between *The person rode the horse* and *The horse rode the person*. Such encoding allows it to find depictions of novel sentences which have never been seen before. Second, it extends video search past nouns and verbs allowing sentences which can encode modifiers such as adverbs and entire prepositional phrases. Unlike other approaches which allow for textual queries of images or videos, we do not require any prior video annotation. The entire lexicon shares a small number of parameters and, unlike previous work, this approach does not need to be trained on every word or even every related word. We have evaluated this approach on a large video corpus of 10 Hollywood movies, comprising roughly 20 hours of video, by running 141 sentential queries.
6. THE COMPOSITIONAL NATURE OF VERB AND ARGUMENT REPRESENTATIONS IN THE HUMAN BRAIN

The compositional nature of thought is taken for granted by many in the cognitive-science and artificial-intelligence communities. For example, in computer vision, representations for nouns, such as those used for object detection, are independent of representations for verbs, such as those used for event recognition. Humans need not employ compositional representations; indeed, many argue that such representations may be doomed to failure in AI systems [117]. This is because concepts like verb or even object are human constructs; there is debate as to how they arise from percepts [118]. Recent advances in brain-imaging techniques enable exploration of the compositional nature of thought. To that end, subjects underwent functional magnetic resonance imaging (fMRI) during which they were exposed to stimuli which evoke complex brain activity which was decoded, piece by piece. The video stimuli depicted events described by entire sentences composed of a verb, an object, an actor and a location or direction of motion. By decoding complex brain activity into its constituent parts, we show evidence for the neural basis of the compositionality of verb and argument representations.

Recent work on decoding brain activity corresponding to nouns has recovered object identity from nouns presented as image and orthographic stimuli. Hanson and Halchenko [119] perform classification on still images of two object classes: faces and houses, and achieve an accuracy above 93% on a one-out-of-two classification task. Connolly et al. [120] perform classification on still images of objects, two instances of each of three classes: bugs, birds, and primates, and achieve an accuracy between 60% and 98% on a one-out-of-two within-class classification task and an accuracy between 90% and 98% on a one-out-of-three between-class classification task. Just
et al. [121] perform classification on orthographically presented nouns, 5 exemplars from each of 12 classes, achieving a mean rank accuracy of 72.4% on a one-out-of-60 classification task, both within and between subjects. Pereira et al. [122] incorporate semantic priors and achieve a mean accuracy of 13.2% on a one-out-of-12 classification task and 1.94% on a one-out-of-60 classification task when attempting to recover the object being observed. Miyawaki et al. [123] recover the position of an object in the field of view by recovering low resolution images from the visual cortex. Object classification from video stimuli has not been previously demonstrated.

Recent work on decoding brain activity corresponding to verbs has primarily been concerned with identifying active brain regions. Kable and Chatterjee [124] present the brain regions which attempt to distinguish between the different agents of actions and between the different kinds of actions they perform. Kemmerer et al. [125] analyze the regions of interest (ROI) of brain activity associated with orthographic presentation of twenty different verbs in each of five different verb classes. Kemmerer and Castillo [126] analyze the brain activity associated with verbs in terms of the motor components of event structure and attempt to localize the ROIs of such motor components. While prior work analyzes regions which are activated when subjects are presented verbs as stimuli, we recover the content of the resulting brain activity by classifying the verb from brain scans.

Recent work demonstrates the ability to decode the actor of an event using personality traits. Hassabis et al. [127] demonstrate the ability to recover the identity of an imagined actor from that actor’s personality. Subjects are informed of the two distinguishing binary personality traits of four actors. During fMRI, they are presented sentences orthographically which describe an actor performing an action. The subjects are asked to imagine this scenario with this actor and to rate whether the actions of the actor accurately reflect the personality of that actor. The resulting brain activation corresponding to these two binary personality traits is used to recover the identity of the actor. No prior work has recovered the identity of an actor without
relying on that actor’s personality. In the work presented here, the personality of the actor has no bearing on the actions being performed.

In this paper, two new experiments are presented. In Experiment 1, subjects are shown videos and asked to think of verbs that characterize those videos. Their brains are imaged via fMRI and measured neural activation is decoded to recover the verb that the subjects are thinking about. Decoding is done by means of a support vector machine (SVM) trained on brain scans of those same verbs. We know of no other work that decodes brain activity corresponding to verbs. We show early evidence that the regions identified by this decoding process are not intimately tied to a particular subject via an additional analysis that trains on one subject and tests on another. In Experiment 2, subjects are shown videos and asked to think of complex sentences composed of multiple components that characterize those videos. We show a novel ability to decode brain activity corresponding to multiple objects: the identity of an actor and the identity of an object. We decode the identity of an actor without relying on the personality traits of that actor. We know of no other work which recovers an entire sentence composed of multiple constituents. We find evidence that suggests underlying neural representations of mental states are independent and compose into sentences largely without modifying one another.

6.1 Compositionality

We discuss a particular kind of compositionality as it applies to sentence structure: objects fill argument positions in predicates that combine to form the meaning of a sentence. Bemis and Pylkkänen [128] reviews work which attempts to show this kind of compositionality using a task called complement coercion. Subjects in this task are presented with sentences whose meaning is richer than their syntax. For example, the sentence *The boy finished the pizza* is understood as meaning that the pizza was eaten, even though the verb *eat* does not appear anywhere in the sentence [129]. The presence of *pizza*, belonging to the category *food*, coerces the interpretation of *finish*.
as *finish eating*. By contrast, *He finished the newspaper* induces the interpretation *finish reading*. Because the syntactic complexity in this prior experiment was held constant, the assumption is that coercion is a purely semantic meaning-adding function application, with little consequence for the syntax. The participants completed this task, and brain activity was measured using magnetoencephalography (MEG). The results show activity related to coercion in the anterior midline field. This result suggests an initial localization for at least some function application, but it is difficult to use MEG to distinguish whether this activity is read from the ventromedial prefrontal cortex or the anterior cingulate cortex. Earlier work on the representation of objects and actions in the brain also indicates that these representations may be independent.

**Representing objects in the brain** Objects are static entities that can be represented by a (mostly) static neural representation. For example, the 3D representation of a soda can will look the same in many different contexts, and the appearance of the soda can is not unfolding in time. It is generally believed that the lexicon of object concepts is represented in the medial temporal lobe while different areas of the temporal lobe may be combinatoric in constructing object types [130] although there may be modal areas associated with different representational functions. For example, lesion data suggests that the temporal pole is associated with naming people, the inferior temporal cortex is associated with naming animals, and the anterior lateral occipital regions are associated with naming tools. In addition, some regions involved in object representation are modality specific. For example, spoken-word processing involves the superior temporal lobe (part of the auditory associative cortex [131]) while reading words representing objects activates occipito-temporal regions because of the visual processing [132]. Specifically, auditory word processing involves a stream of information starting in Heschl’s gyri that is transferred to the superior temporal gyrus. Once the superior temporal gyrus has been reached, the modality of stimulus presentation is no longer relevant. In contrast, the initial processing for written words
starts in the occipital lobe (V1 and V2), and moves on to occipito-temporal regions specialized in identifying orthographic units. The information then moves rostrally to the temporal lobe proper, where modality of presentation is no longer relevant [131].

**Representing actions in the brain**  Unlike objects, verbs are dynamic entities that unfold in time. For instance, observing someone pick up a ball takes time as the person’s movement unfolds. Evidence reviewed in Coello and Bidet-Ildel [133] suggests that action verbs activate both semantic units in the temporal cortex and a motor network. The motor network includes the premotor areas (including the supplementary motor area), the primary motor cortex, and the posterior parietal cortex. Some researchers went as far as suggesting that the well-known ventral/dorsal distinction in the visual pathways corresponds to a semantic (ventral) and action (dorsal) distinction. Representation of action may involve ‘mirror neurons’ that have been shown in macaque to respond jointly in perception/action tasks, where the similarity of the self action is to the perceived action of an observed individual.

### 6.2 Approach

All experiments reported follow the same procedure and are analyzed using the same methods and classifiers. Videos are shown to subjects who are asked to think about some aspect(s) of the video while whole-brain fMRI scans are acquired every two seconds. Because fMRI acquisition times are slow, roughly equal to the length of the video stimuli, a single brain volume that corresponds to the brain activation induced by that video stimulus is classified to recover the features that the subjects were asked to think about. Multiple runs separated by several minutes of rest, where no data is acquired, are performed per subject.

#### 6.2.1 fMRI procedures

Imaging performed at Purdue University used a 3T GE Signa HDx scanner (Waukesha, Wisconsin) with a Nova Medical (Wilmington, Massachusetts) 16 channel brain
array to collect whole-brain volumes via a gradient-echo EPI sequence with 2000ms TR, 22ms TE, 200mm×200mm FOV, and 77° flip angle. We acquired 35 axial slices with a 3.000mm slice thickness using a 64×64 acquisition matrix resulting in 3.125mm×3.125mm×3.000mm voxels.

Imaging performed at St. James Hospital in Dublin, Ireland, used a 3T Phillips Achieva scanner (Best, The Netherlands) using a gradient-echo EPI sequence with 2000ms TR and 240mm×240mm FOV. We acquired 37 axial slices with a 3.550mm slice thickness using an 80×80 acquisition matrix resulting in 3.000mm×3.000mm×3.550mm voxels.

6.2.2 fMRI processing

Data was acquired in runs, with between three and eight runs per subject per experiment, and each axis of variation of each experiment was counterbalanced within each run. fMRI scans were processed using AFNI [134] to skull-strip each volume, motion correct and detrend each run, and align each subject’s runs to each other. Voxels within a run were z-scored, subtracting the mean value of that voxel for the run and dividing by its variance. Because each brain volume has very high dimension, between 143,360 and 236,800 voxels, we eliminate voxels by computing a per-voxel Fisher score on our training set and keeping the 4,000 highest-scoring voxels. The Fisher score of a voxel v for a classification task with C classes where each class c has \(n_c\) examples is computed as

\[
\sum_{c=1}^{C} \frac{n_c(\mu_{c,v} - \mu)^2}{\sum_{c=1}^{C} n_c \sigma_{c,v}^2} \tag{6.1}
\]

where \(\mu_{c,v}\) and \(\sigma_{c,v}\) are the per-class per-voxel means and variances and \(\mu\) is the mean for the entire brain volume. The resulting voxels are then analyzed with Linear Discriminant Dimensionality Reduction [135] which selects a smaller number of potentially-relevant voxels. A linear SVM classifies the selected voxels.
One run was taken as the test set and the remaining runs were taken as the training set. The third brain volume after the onset of each stimulus was taken along with the class of the stimulus to train an SVM. This lag of three brain volumes is required because fMRI does not measure neural activation but instead measures the flow of oxygenated blood, the blood-oxygen-level-dependent (BOLD) signal, which correlates with increased neural activation. It takes roughly five to six seconds for this signal to peak which puts the peak in the third volume after the stimulus presentation. Cross validation was performed by choosing each of the different runs as the test set.

To understand our results and to demonstrate that they are not classifying noise or irrelevant features, we perform an analysis to understand the brain regions that are relevant to each experiment. We determine these regions by two methods. First we employ a spatial searchlight [136] which slides a small sphere across the entire brain volume and repeats the above analysis keeping only the voxels inside that sphere. We use a sphere of radius three voxels, densely place its center at every voxel, and do not perform any dimensionality reduction on the remaining voxels. We then perform an eight-fold cross validation as described above for each position of the sphere. For Experiment 1 we also back-project the SVM coefficients onto the anatomical scans—the higher the absolute value of the coefficient the more that voxel contributes to the classification performance of the SVM—and use a classifier with a different metric, $w(i)^2$, as described by Hanson and Halchenko [119].

6.3 Experiment 1: Verb Representation

We conducted an experiment to evaluate the ability to identify brain activity corresponding to verbs denoting actions. Subjects are shown video clips of humans interacting with objects and are told to think of the verb being enacted, but otherwise have no task. The subjects were shown clips depicting each of these verbs prior to the experiment and were instructed about the intended meaning of each verb. One difficulty with such an experiment is that there is disagreement between human
subjects as to whether a verb occurred in a video or not. To overcome this difficulty, we asked five humans to annotate the DARPA Mind’s Eye year 2 video corpus with the extent of every verb. From this corpus, we chose video clips where at least two out of the five annotators agreed on the depiction. We selected between twenty seven and thirty 2.5s video clips depicting each of six different verbs (carry, dig, hold, pick up, put down, and walk). Key frames from one clip for each of the six verbs are shown in Figure 6.1. Despite multiple annotators agreeing on whether a video depicts a verb, the task of classifying each clip remains very difficult for human subjects as it is easy to confuse similar verbs such as carry and hold. We address this problem by presenting, in rapid succession, pairs of video clips which depict the same verb and asking the subjects to think about the verb that would best describe both videos.

We employed a rapid event-related design similar to that of Just et al. [121]. We presented pairs of 2.5s video clips at 12fps, depicting the same verb, separated by 0.5s blanking and followed by an average of 4.5s (minimum 2.5s) fixation. While the video clips within each pair depicted the same verb, the clips across pairs within a run depicted different verbs, randomly counterbalanced. Each run comprised 48 stimulus presentations spanning 254 captured brain volumes and ended with 24s of fixation. Eight runs for each of subjects 1 through 3 and subjects 6 through 10 were collected at Purdue University. Three runs for subject 4 and four runs for subject 5 were collected at St. James Hospital.

We performed an eight-fold cross validation (fewer for subjects 4 and 5) for a six-way classification task, where runs constituted folds. The results are presented in Figure 6.2. The per-subject accuracies, averaged across class and fold, were: 79.42%, 88.80%, 75.00%, 28.12%, 35.41%, 38.28%, 90.62%, 83.85%, 51.04%, and 50.78% (chance 16.66%). Note that the last two were trained on fewer runs than the first three. This demonstrates the ability to recover the verb that the subjects were thinking about. The robustness of this result is enhanced by the fact that it was replicated on two different fMRI scanners at different locations run by different experimenters.
Fig. 6.1. Key frames from sample stimuli for each of the six verbs in Experiment 1.
Fig. 6.2. Results for Experiment 1. (a) Per-subject classification accuracy on 1-out-of-6 verb classes averaged across class and fold. Horizontal line indicates chance performance, 16.66%. (b) Corresponding confusion matrix averaged across subject and fold is mostly diagonal, with the highest numbers of errors being made distinguishing *hold* and *carry*, two ambiguous stimuli.
To evaluate whether the brain regions used for classification generalize across subjects, we performed an additional analysis on the data for subjects 1 and 2. One run out of the eight was selected as the test set and the data for one of the two subjects was classified. The training set consisted of all seven other runs for the subject whose data does not appear in the test set. The test was performed on the run omitted from the training set, even though it was gathered from a different subject, to preclude the possibility that the same stimulus sequence appeared in both the training and test sets. We performed cross validation by varying which subject contributes the test data and which subject contributes the training data, and within each of these folds we varied which of the eight runs is the test set. These two cross validations yielded accuracies of 33.59% (subject 1→subject 2) and 41.41% (subject 2→subject 1), averaged across class and fold, where chance again is 16.66%.
To locate regions of the brain used in the previous analysis, we used a spatial-searchlight linear-SVM method on subject 1. We use the accuracy to determine the sensitivity of each voxel and threshold upward to less than 5% of the cross-validation measures. These measures are overlaid and (2-stage) registered to MNI152 2mm anatomicals shown in Figure 6.3(top). Notable are visual-pathway areas (lateral occipital-LO, lingual gyrus-LG, and fusiform gyrus) as well as prefrontal areas (inferior frontal gyrus, middle frontal gyrus, and cingulate) and areas consistent with the ‘mirror system’ [137] and the so-called ‘theory of mind’ (pre-central gyrus, angular gyrus-AG, and superior parietal lobule-SPL) areas [138,139]. Figure 6.3(bottom) shows the decoded ROIs from a similar SVM classifier with a different metric, \( w(i)^2 \) Hanson and Halchenko [119], showing similar brain areas but, due to higher sensitivity, also indicates sub-cortical regions (hippocampal) associated with encoding processes not seen with the cross-validation accuracy metric. As argued in Section 6.1, lateral-occipital areas are involved in visual processing specifically related to language, and the fusiform gyrus is a hetero-modal area that could hold abstract representations of the elements contained in the videos (e.g. semantics). This data brings initial support for the hypothesis that concepts have both modality-specific and abstract representations. Hence, the elements used by the SVM to classify the videos are also neuroscientifically meaningful.

6.4 Experiment 2: Argument Representation

We conducted a further experiment to evaluate the ability to recover compositional semantics for entire sentences. Subjects were shown videos that depict sentences of the form: the actor verb the object direction/location. They were asked to think about the sentence depicted in each video and otherwise had no task. Videos depicting three verbs (carry, fold, and leave), each performed with three objects (chair, shirt, and tortilla), each performed by four human actors, and each performed on either side of
the field of view were filmed for this task. The verbs were chosen to be discriminable based on features described by Kemmerer et al. [125]:

\[
\begin{align*}
\text{leave} & : -\text{state-change} -\text{contact} \\
\text{fold} & : +\text{state-change} +\text{contact} \\
\text{carry} & : -\text{state-change} +\text{contact}
\end{align*}
\]

Nouns were chosen based on categories found to be easily discriminable by Just et al. [121]: chair (furniture), shirt (clothing), and tortilla (food) and also selected to allow each verb to be performed with each noun. Because these stimuli are not as ambiguous as the ones from Experiment 1, they were not shown in pairs. All stimuli enactments were filmed against the same nonvarying background, which contained no other objects except for a table (Fig 6.4).

This experiment, like Experiment 1, also used a rapid event-related design. We collected multiple videos, between 4 and 7, for each cross product of the verb, object and human actor. Variation along the side of field of view and direction of motion was accomplished by mirroring the videos about the vertical axis. Such mirroring induces variation in direction of motion (leftward vs. rightward) for the verbs carry and leave and induces variation in the location in the field of view where the verb fold occurs (left half vs. right half). We presented 2s video clips at 10fps followed by an average of 4s (minimum 2s) fixation. Each run comprised 72 stimulus presentations spanning 244 captured brain volumes, with eight runs per subject, and ended with 24s of fixation. Each run was individually counterbalanced for each of the four conditions (verb, object, actor, and mirroring). We collected data for three subjects at Purdue University but discarded the data for one of the three due to subject motion. One subject did eight runs without exiting the scanner. One subject exited the scanner between runs six and seven, which required cross-session registration. All subjects were aware of the experiment design, were informed of the intended depiction of each stimulus prior to the scan, and were instructed to think of the intended depiction after each presentation.

This experimental design supports the following classification analyses:
event one-out-of-9 verb&noun (carry, fold, and leave, each performed on chair, shirt, and tortilla)

verb one-out-of-3 verb (carry, fold, and leave)

object one-out-of-3 noun (chair, shirt, and tortilla)

actor one-out-of-4 actor identity

direction one-out-of-2 motion direction for carry and leave (leftward vs. rightward)

location one-out-of-2 location in the field of view for fold (right vs. left)

The analysis performed was exactly the same as that for Experiment 1, including eight-fold cross validation for each of our analyses, where runs constituted folds. Figures 6.5 through 6.7 present an overview of the results along with per-subject classification accuracies and aggregate confusion matrices for the each of the above analyses. Note that we achieve significantly above-chance performance on all six analyses with only a single fold for a single subject across all six analyses performing below chance.

Verb performance is well above chance (78.91%, chance 11.11%). This replicates Experiment 1 with different videos and a new verb and adds to the evidence that brain activity corresponding to verbs can reliably be decoded from fMRI scans. Object performance was significant as well (59.79%, chance 33.33%). Given neural activation, we can decode which object the subjects are thinking about. We know of no other work that decodes brain activity corresponding to objects from videos. The fact that the verb and object can be decoded independently already provides evidence of argument compositionality. Were the neural representations not compositional at this level, decoding would not be possible. For example, if the representation of carry was neurally encoded as a combination of walk and a particular object, verb performance would not exceed chance, because our experiment is counterbalanced with respect to the object with which the action is being performed. While this indicates that the representations for verbs and objects are independent of each other to some degree, we also seek to quantify the level of independence. If the representation of carry is somewhat different depending on which object is being carried, we expect that
performance would increase when we jointly classify the object and the verb. This seems to not be the case. The accuracy of **event** is almost identical to the joint independent accuracy of **verb** and **object**: 49.58 ≈ 47.19 = 78.92 × 59.80, indicating that the representation of these verbs is independent of the objects that the verbs are being performed with. This is also confirmed by the confusion matrix for **event** in Figure 6.7 which remains diagonal.

To decode complex brain activity corresponding to an entire sentence, we can combine **actor**, **verb**, **object**, and **direction** or **location**. We perform significantly above chance on this one-out-of-72 (4 × 3 × 3 × 2) classification for each of the subjects:

\[
\begin{align*}
0.3038 \times 0.7760 \times 0.5538 \times (\frac{2}{3} \times 0.7986 + \frac{1}{3} \times 0.6979) &= 0.0999 \gg 0.0139 = \frac{1}{72} \\
0.3142 \times 0.6736 \times 0.5417 \times (\frac{2}{3} \times 0.6771 + \frac{1}{3} \times 0.6146) &= 0.0752 \gg 0.0139 = \frac{1}{72} \\
0.3559 \times 0.8299 \times 0.6233 \times (\frac{2}{3} \times 0.8507 + \frac{1}{3} \times 0.6771) &= 0.1459 \gg 0.0139 = \frac{1}{72} \\
0.3524 \times 0.8160 \times 0.6597 \times (\frac{2}{3} \times 0.7760 + \frac{1}{3} \times 0.7344) &= 0.1446 \gg 0.0139 = \frac{1}{72} \\
0.3316 \times 0.8698 \times 0.6632 \times (\frac{2}{3} \times 0.8524 + \frac{1}{3} \times 0.7552) &= 0.1569 \gg 0.0139 = \frac{1}{72} \\
0.3229 \times 0.7760 \times 0.5747 \times (\frac{2}{3} \times 0.7917 + \frac{1}{3} \times 0.7969) &= 0.1143 \gg 0.0139 = \frac{1}{72} \\
0.3524 \times 0.7830 \times 0.5694 \times (\frac{2}{3} \times 0.8090 + \frac{1}{3} \times 0.7135) &= 0.1221 \gg 0.0139 = \frac{1}{72}
\end{align*}
\]

(Since **direction** applied to **carry** and **leave** while **location** disjointly applied to **fold**, this yields a binary classification task across all verbs.) Thus we are able to classify **entire sentences** compositionally from their individual words.

To locate regions of the brain used in the previous analyses, we applied the same searchlight linear-SVM method that was performed in Experiment 1 to subject 1’s data from this experiment and identified similar areas in visual-pathway, parietal, and prefrontal areas. The resulting ROIs, shown in Figure 6.8, are overlaid and color coded according to the specific visual feature being decoded. In general, it is clear that the decoding is sensitive to action/category information and various visual object-and-motion features. Many of the same regions active for **verb** in Experiment 1 also show activity in this experiment. **Direction** and **location** activity is present in the visual cortex with significant **location** activity occurring in the early visual cortex.
Object activity is present in the temporal cortex, and agrees with previous work on object-category encoding [140].

6.5 Conclusion

We have demonstrated that it is possible to read a subject’s brain activity and decode a complex action tableau corresponding to a sentence from its constituents. To do so, we showed novel work which decodes brain activity associated with verbs and simultaneously recovers lexical aspects of different parts of speech. Our results indicate that the neural representations for verbs and objects compose together to form the meaning of a sentence apparently without modifying one another. These results indicate that representations which attempt to decompose meaning into constituents may have a neural basis.
Fig. 6.4. Key frames from sample stimuli in Experiment 2.
<table>
<thead>
<tr>
<th>Subject</th>
<th>event</th>
<th>verb</th>
<th>object</th>
<th>actor</th>
<th>direction</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52.25%</td>
<td>77.60%</td>
<td>55.38%</td>
<td>30.38%</td>
<td>79.86%</td>
<td>69.79%</td>
</tr>
<tr>
<td>2</td>
<td>38.88%</td>
<td>67.36%</td>
<td>54.16%</td>
<td>31.42%</td>
<td>67.70%</td>
<td>61.45%</td>
</tr>
<tr>
<td>3</td>
<td>53.99%</td>
<td>82.98%</td>
<td>62.32%</td>
<td>35.59%</td>
<td>85.06%</td>
<td>67.70%</td>
</tr>
<tr>
<td>4</td>
<td>54.51%</td>
<td>81.59%</td>
<td>65.97%</td>
<td>35.24%</td>
<td>77.60%</td>
<td>73.43%</td>
</tr>
<tr>
<td>5</td>
<td>59.54%</td>
<td>86.97%</td>
<td>66.31%</td>
<td>33.15%</td>
<td>85.24%</td>
<td>75.52%</td>
</tr>
<tr>
<td>6</td>
<td>46.18%</td>
<td>77.60%</td>
<td>57.46%</td>
<td>32.29%</td>
<td>79.16%</td>
<td>79.68%</td>
</tr>
<tr>
<td>7</td>
<td>41.66%</td>
<td>78.29%</td>
<td>56.94%</td>
<td>35.24%</td>
<td>80.90%</td>
<td>71.35%</td>
</tr>
<tr>
<td>chance</td>
<td>11.11%</td>
<td>33.33%</td>
<td>33.33%</td>
<td>25.00%</td>
<td>50.00%</td>
<td>50.00%</td>
</tr>
</tbody>
</table>

Fig. 6.5. Per-subject mean classification accuracies averaged across fold for Experiment 2. Note that all six analyses perform above chance.
Fig. 6.6. Per-subject classification accuracies for Experiment 2 showing the means and variances of performance across the different folds for each class. The horizontal line indicates chance performance.
Fig. 6.7. Confusion matrices for Experiment 2, averaged across subject and fold. Note that they are mostly diagonal.

Fig. 6.8. Searchlight analysis for Experiment 2 indicating the classification accuracy of different brain regions on the anatomical scans from subject 1 averaged across stimulus, class, and run.
7. CONCLUSION

This thesis suggests that reasoning across multiple modalities can improve the performance of AI systems and allow them to perform novel tasks. Knowledge about the world can facilitate such reasoning. We presented a domain where we acquired such knowledge: the rules of social interaction in the context of learning to play board games. Humans solve a similar problem; children do not receive written instructions on how to interact with each other or how to play games. And similarly to the approach presented here humans learn from a small number of examples. This task is a first step to modeling a teacher-student interaction.

Humans reason across multiple modalities with ease and this ability seems to underlie many of the tasks that humans perform. We presented a domain, event recognition, along with an algorithm that simultaneously performs object detection, object tracking, and recognizes complex events described by a sentence. A single cost function that integrates both higher-level cognition and low-level perception allows the two to influence each other. This single cost function enabled a new task, sentential retrieval of videos.

The representations used the in the above task are compositional in nature: the semantics of sentences are constructed from the semantics of individual words. We presented work that indicates that the human brain may also employ compositional representations. To do so, we compositionally recovered sentences describing visual stimuli being observed by subjects. Further understanding of the human brain may reveal other principles underlying multiple modalities which may lead to more robust AI algorithms.
LIST OF REFERENCES
LIST OF REFERENCES


VITA
VITA

Andrei Barbu received the BCS degree from the University of Waterloo in 2008. He is currently a Ph.D. candidate in the ECE department at Purdue University. His research interests lie at the intersection of computer vision, natural language, and robotics. He is particularly interested in how both machines and humans can use language to transfer knowledge between multiple modalities and reason across both language and vision simultaneously.