Saying What You're Looking For: Linguistics Meets Video Search

Daniel Paul Barrett, *Student Member, IEEE*, Andrei Barbu, *Student Member, IEEE*, N. Siddharth, *Member, IEEE*, and Jeffrey Mark Siskind, *Senior Member, IEEE*

Abstract—We present an approach to searching large video corpora for clips which depict a natural-language query in the form of a sentence. Compositional semantics is used to encode subtle meaning differences lost in other approaches, such as the difference between two sentences which have identical words but entirely different meaning: *The person rode the horse* versus *The horse rode the person*. Given a sentential query and a natural-language parser, we produce a score indicating how well a video clip depicts that sentence for each clip in a corpus and return a ranked list of clips. Two fundamental problems are addressed simultaneously: detecting and tracking objects, and recognizing whether those tracks depict the query. Because both tracking and object detection are unreliable, our approach uses the sentential query to focus the tracker on the relevant participants and ensures that the resulting tracks are described by the sentential query. While most earlier work was limited to single-word queries which correspond to either verbs or nouns, we search for complex queries which contain multiple phrases, such as prepositional phrases, and modifiers, such as adverbs. We demonstrate this approach by searching for 2,627 naturally elicited sentential queries in 10 Hollywood movies.

Index Terms—Retrieval, video, language, tracking, object detection, event recognition, sentential video retrieval

1 Introduction

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. Most approaches to date attempt to independently address the various aspects that make this task difficult. For example, they usually 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 simultaneously 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 what follows, we describe a system which, unlike most 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. 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 2,627 natural-language queries on a corpus of 10 full-length Hollywood movies using a grammar which includes nouns, verbs, adverbs, and prepositions. This is one of the first approaches which can search for complex queries which include multiple phrases, such as prepositional phrases, and modifiers, such as adverbs.

2 Related Work

In a recent survey of video retrieval, Hu et al. [1] note that work on semantic video search focuses on detecting nouns

• *The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035. E-mail: {dpbarret, qobi}@purdue.edu, andrei@0xab.com, siddharth@iffsid.com.*

Manuscript received 25 Nov. 2013; revised 18 Sept. 2015; accepted 9 Nov. 2015. Date of publication 2 Dec. 2015; date of current version 12 Sept. 2016. Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2505297

and verbs, as well as using language to search already-existing video annotation. The state of the art in image retrieval is similar [2], [3], [4]. Note that 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 most earlier approaches, the work presented here requires no pre-existing annotations.

Retrieving clips or frames in which a query object occurs has been addressed both using query-by-example and object detection. Sivic and Zisserman [5] 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. [6] 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 [7] 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. [8] 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 the PASCAL Visual Object Classes (VOC) Challenge [9]. Sadeghi and Farhadi [10] 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 *rode* combine together to encode the semantics of the sentence and to perform retrieval of a sentence without any particular training for that sentence.

Prior work on verb detection does not integrate with work on object detection. Chang et al. [15] find one of four different highlights in basketball games using hidden Markov models (HMMs) 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 [16], [17], [18], [19], [20]. We will avail ourselves of such work later [21] to show that the work presented here both allows for richer queries and improves on the performance of earlier approaches.

Most 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. [22], Worring et al. [23], Snoek et al. [24], and Tapaswi et al. [25] 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. [26] 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*.

İkizler and Forsyth [27], [28] present early work on using finite-state models as event recognizers and use such to perform video retrieval. However, unlike the work presented here, they only model verbs and not other parts of speech such as nouns, adverbs, and prepositions, and particularly do not model how such word meanings combine to form sentence meanings. Lin et al. [29] present an approach to video retrieval with multi-word sentential queries. Their work differs from ours, *inter alia*, in that their semantic representation only supports unary predicates while ours supports predicates of any arity. Moreover, they adopt what they call the *no coreference* constraint so that each object is assigned to at most one predicate; we have no such restriction. The no coreference constraint precludes handling the example shown in Fig. 4 because *person* and *horse* must be arguments of the verb *rode* and at least one of them must also be an argument of the spatial relation *leftward*, the adverb *quickly*, and the preposition *towards*. Kiros et al. [30] present an approach for producing text descriptions of still images and retrieving still images from a dataset that match multi-word text queries. One major difference from the work presented here is that Kiros et al. [30] represent the meanings of words by an association with visual features and not their truth conditions. Moreover, unlike the work presented here, they do not formulate a precise mechanism by which the truth conditions for the words combine to yield truth conditions for phrases and sentences. Thus their system can often produce text descriptions that are not true of the input images.

3 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. [31], [32] object detector, specifically the variant developed by Sadeghi and Forsyth [33]. 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 not totally reliable. 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 $g(b^{t-1}_{j^{t-1}}, b^t_{j^t})$. We formulate the score of a track $\mathbf{j} = \langle j^1, \ldots, j^T \rangle$ as

$$\max_{j^1,\ldots,j^T} \sum_{t=1}^{T} f(b^t_{j^t}) + \sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}), \tag{1}$$

where we take g, the motion coherence, to be a nonincreasing function of the squared Euclidean distance between the center of $b^{t-1}_{j^{t-1}}$ and the center of $b^t_{j^t}$ projected one frame forward. While the number of possible tracks is exponential in the number of frames in the video, Eq. (1) can be maximized in time linear in the number of frames and quadratic in the number of detections per frame using dynamic programming [34], the Viterbi [35], [36] algorithm.

The development of this tracker follows that of Barbu et al. [37] which presents additional details of such a tracker, including an extension which allows generating multiple tracks per object class using non-maximal suppression. That earlier tracker used the raw detection scores from the Felzenszwalb et al. [31], [32] 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 $-1/11$.

4 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. Fig. 2 provides an overview of the FSMs used in Section 7, rendered as regular expressions. This approach is a limiting case of that taken by Barbu et al. [38] which used hidden Markov models 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. [38] to be employed here.

We construct word meanings in two levels. First, we construct 17 predicates, shown in Fig. 1, which accept one or more detections. We then construct word meanings for our lexicon of 15 words, shown in Fig. 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 nine parameters: **far**, **close**, **stationary**, **Δclosing**, **Δangle**, **Δpp**, **Δquickly**, **Δslowly**, and **overlap**. To make predicates independent of the video resolution, the scale-dependent parameters **far**, **close**, **stationary**, **Δclosing**, **Δpp**, **Δquickly**, and **Δslowly** are scaled relative to a nominal horizontal resolution of 1,280 pixels.

Given a regular expression for a word, we can construct a nondeterministic FSM whose allowable transitions are encoded by a binary transition matrix a, giving score zero to allowed transitions and $-\infty$ to disallowed transitions, and whose states accept detections which agree with the predicate h, 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

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

where k^1 through k^T range over the set of states of the FSM, constraining k^1 to be an allowed initial state and k^T to be an allowed final state. If this word describes the track, the score yielded by Eq. (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. [37]. 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 meanings of verbs but also of other parts of speech. For example, the meaning of a static concept, such as a preposition like *to the 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 *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

$$
\begin{aligned}
&\text{FAR}(a,b) && \triangleq\ |a_{cx}-b_{cx}|-\tfrac{a_{width}}{2}-\tfrac{b_{width}}{2}>\textbf{far}\\
&\text{REALLY-CLOSE}(a,b) && \triangleq\ |a_{cx}-b_{cx}|-\tfrac{a_{width}}{2}-\tfrac{b_{width}}{2}<\tfrac{\textbf{close}}{2}\\
&\text{CLOSE}(a,b) && \triangleq\ |a_{cx}-b_{cx}|-\tfrac{a_{width}}{2}-\tfrac{b_{width}}{2}<\textbf{close}\\
&\text{STATIONARY}(b) && \triangleq\ \text{flow-magnitude}(b)\leq\textbf{stationary}\\
&\text{CLOSING}(a,b) && \triangleq\ |a_{cx}-b_{cx}|>|\text{project}(a)_{cx}-b_{cx}|+\Delta\textbf{closing}\wedge\text{STATIONARY}(b)\\
&\text{DEPARTING}(a,b) && \triangleq\ |a_{cx}-b_{cx}|<|\text{project}(a)_{cx}-b_{cx}|+\Delta\textbf{closing}\wedge\text{STATIONARY}(b)\\
&\text{MOVING-DIRECTION}(a,\alpha) && \triangleq\ |\text{flow-orientation}(a)-\alpha|^{\circ}<\Delta\textbf{angle}\wedge\neg\text{STATIONARY}(a)\\
&\text{LEFT-OF}(a,b) && \triangleq\ a_{cx}<b_{cx}-\Delta\textbf{pp}\\
&\text{RIGHT-OF}(a,b) && \triangleq\ a_{cx}>b_{cx}+\Delta\textbf{pp}\\
&\text{LEFTWARD}(a) && \triangleq\ \text{MOVING-DIRECTION}(a,0)\\
&\text{RIGHTWARD}(a) && \triangleq\ \text{MOVING-DIRECTION}(a,\pi)\\
&\text{STATIONARY-BUT-FAR}(a,b) && \triangleq\ \text{FAR}(a,b)\wedge\text{STATIONARY}(a)\wedge\text{STATIONARY}(b)\\
&\text{STATIONARY-BUT-CLOSE}(a,b) && \triangleq\ \text{CLOSE}(a,b)\wedge\text{STATIONARY}(a)\wedge\text{STATIONARY}(b)\\
&\text{MOVING-TOGETHER}(a,b) && \triangleq\ |\text{flow-orientation}(a)-\text{flow-orientation}(b)|^{\circ}<\Delta\textbf{angle}\wedge\\
&&&\quad\neg\text{STATIONARY}(a)\wedge\neg\text{STATIONARY}(b)\\
&\text{QUICKLY}(a) && \triangleq\ \text{flow-magnitude}(a)>\Delta\textbf{quickly}\\
&\text{SLOWLY}(a) && \triangleq\ \neg\text{STATIONARY}(a)\wedge\text{flow-magnitude}(a)<\Delta\textbf{slowly}\\
&\text{OVERLAPPING}(a,b) && \triangleq\ \tfrac{a\cap b}{a\cup b}\geq\textbf{overlap}
\end{aligned}
$$

Fig. 1. Predicates which accept detections, denoted by a and b, formulated around nine parameters. These predicates are used for the experiments in Section 7. 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.

concepts. Employing a uniform but powerful representation to encode the meanings 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 $\boldsymbol{t}true^{+}$ into many of the word FSMs in Fig. 2.

5 Sentence Tracker

Our ultimate goal is to search for videos described by a natural-language query in the form of a sentence. The

$$
\begin{aligned}
&\texttt{person}(a) && \triangleq\ (a_{\textit{object-class}}=\text{“person”})^{+}\\
&\texttt{horse}(a) && \triangleq\ (a_{\textit{object-class}}=\text{“horse”})^{+}\\
&\texttt{to the left of}(a,b) && \triangleq\ \textbf{true}^{+}\ \text{LEFT-OF}(a,b)^{\{3,\}}\ \textbf{true}^{+}\\
&\texttt{to the right of}(a,b) && \triangleq\ \textbf{true}^{+}\ \text{RIGHT-OF}(a,b)^{\{3,\}}\ \textbf{true}^{+}\\
&\texttt{approached}(a,b) && \triangleq\ \textbf{true}^{+}\ \text{CLOSING}(a,b)^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{led}(a,b) && \triangleq\ \textbf{true}^{+}\ (\neg\text{REALLY-CLOSE}(a,b)\wedge\text{MOVING-TOGETHER}(a,b)\wedge\text{CLOSE}(a,b))^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{rode}(a,b) && \triangleq\ \textbf{true}^{+}\ (\text{MOVING-TOGETHER}(a,b)\wedge\text{OVERLAPPING}(a,b))^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{quickly}(a) && \triangleq\ \textbf{true}^{+}\ \text{QUICKLY}(a)^{\{3,\}}\ \textbf{true}^{+}\\
&\texttt{slowly}(a) && \triangleq\ \textbf{true}^{+}\ \text{SLOWLY}(a)^{\{3,\}}\ \textbf{true}^{+}\\
&\texttt{leftward}(a) && \triangleq\ \textbf{true}^{+}\ \text{LEFTWARD}(a)^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{rightward}(a) && \triangleq\ \textbf{true}^{+}\ \text{RIGHTWARD}(a)^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{towards}(a,b) && \triangleq\ \text{STATIONARY-BUT-FAR}(a,b)^{+}\ \text{CLOSING}(a,b)^{\{3,\}}\ \text{STATIONARY-BUT-CLOSE}(a,b)^{+}\\
&\texttt{away from}(a,b) && \triangleq\ \text{STATIONARY-BUT-CLOSE}(a,b)^{+}\ \text{DEPARTING}(a,b)^{\{3,\}}\ \text{STATIONARY-BUT-FAR}(a,b)^{+}\\
&\texttt{from the left}(a,b) && \triangleq\ \textbf{true}^{+}\ \text{LEFT-OF}(a,b)^{\{5,\}}\ \textbf{true}^{+}\\
&\texttt{from the right}(a,b) && \triangleq\ \textbf{true}^{+}\ \text{RIGHT-OF}(a,b)^{\{5,\}}\ \textbf{true}^{+}
\end{aligned}
$$

Fig. 2. Regular expressions which encode the meanings of each of the 15 words or lexicalized phrases in the lexicon used for the experiments in Section 7. These are composed from the predicates shown in Fig. 1. We use an extended regular-expression syntax where an exponent of $\{t,\}$ allows a predicate to hold for t or more frames.

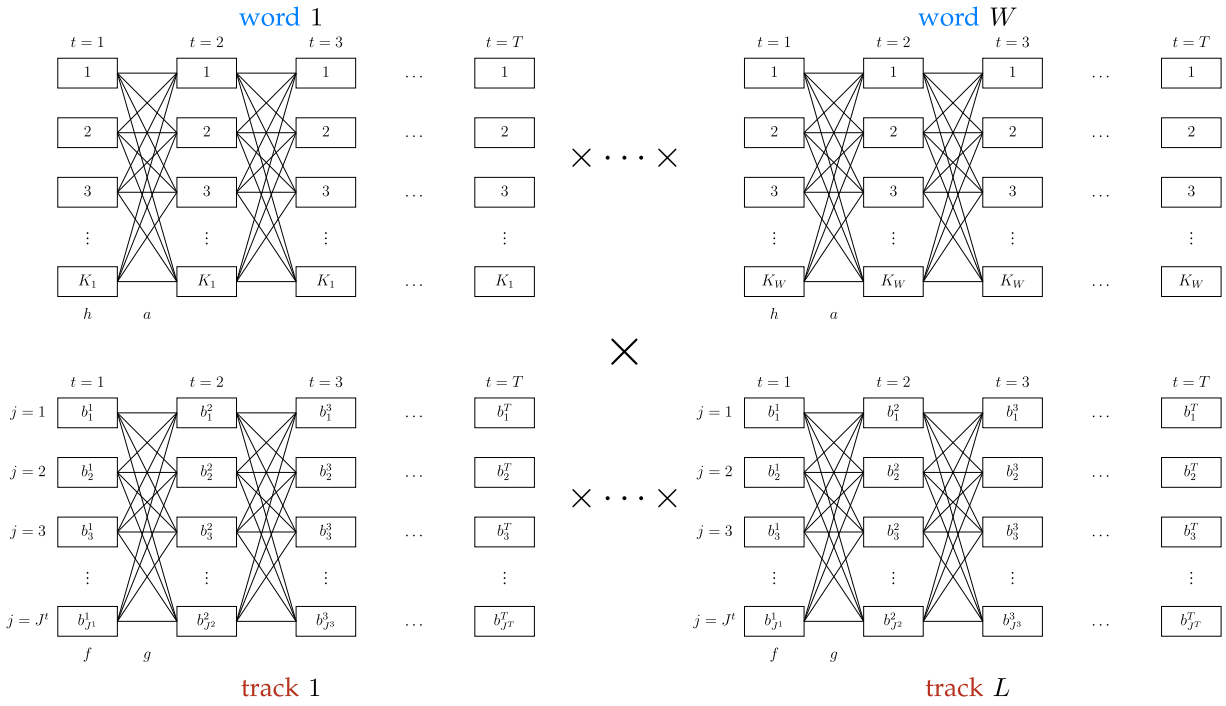

Fig. 3. Tracker lattices are used to track each participant. Word lattices constructed from word FSMs for each word in the sentence recognize collections of tracks for participants that exhibit the semantics of that word as encoded in the FSM. We take the cross product of multiple tracker and word lattices to simultaneously track participants and recognize words. This ensures that the resulting tracks are described by the desired sentence.

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 enforced. This may result in the tracker producing tracks which have to consist of lower-scoring detections, which decreases the probability that 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, $f(b^t_{j^t})$, for the tracker, or states scored by their output predicate, $h(k^t, b^t_{j^t})$, for the word recognizer, and each weights the links in that lattice by a score, the motion coherence, $g(b^{t-1}_{j^{t-1}}, b^t_{j^t})$, 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

$$\max_{j^1,\ldots,j^T} \max_{k^1,\ldots,k^T} \left(\sum_{t=1}^{T} f(b^t_{j^t}) + \sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) + \sum_{t=1}^{T} h(k^t, b^t_{j^t}) + \sum_{t=2}^{T} a(k^{t-1}, k^t) \right), \tag{3}$$

which ensures that, unless the state sequence for the word FSM starts at an allowed initial state and 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 lattice construction is shown in Fig. 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 Fig. 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 *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, *agent* (the entity performing the action, in this case the person), *patient* (the entity affected by the action, in this case the first horse), and *goal* (the destination of the action, in this case the second horse). Each word in this sentence refers to a subset of these

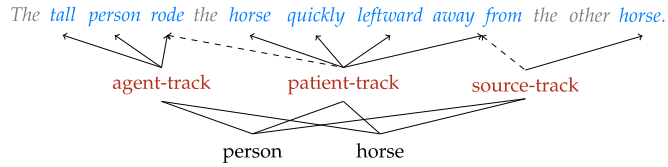

Fig. 4. Different sentential queries lead to different cross products. The sentence is parsed and the role of each participant, show in red, is determined. A single tracker lattice is constructed for each participant. Words and lexicalized phrases, shown in blue, have associated word lattices which encode their semantics. The arrows between words and participants represent the track-to-role mappings, θ, 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 *rode* 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.

three different participants, as shown in Fig. 4, and words that refer to multiple participants, such as *rode*, 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, θ^i_w, map the ith role in a word w to a tracker. Fig. 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 θ^i_w, 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

$$\max_{\substack{j^1_1,\ldots,j^T_1 \\ \vdots \\ j^1_L,\ldots,j^T_L}} \max_{\substack{k^1_1,\ldots,k^T_1 \\ \vdots \\ k^1_W,\ldots,k^T_W}} \left(\sum_{l=1}^{L} \sum_{t=1}^{T} f\left(b^t_{j^t_l}\right) + \sum_{t=2}^{T} g\left(b^{t-1}_{j^{t-1}_l}, b^t_{j^t_l}\right) + \sum_{w=1}^{W} \sum_{t=1}^{T} h_w\left(k^t_w, b^t_{j^t_{\theta^1_w}}, \ldots, b^t_{j^t_{\theta^I_w}}\right) + \sum_{t=2}^{T} a_w\left(k^{t-1}_w, k^t_w\right) \right), \tag{4}$$

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 tth frame of the lth track, and $b^t_{j^t_{\theta^i_w}}$ connects a participant that fills the ith 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. 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.

6 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 use 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 Fig. 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

$$
\begin{array}{lcl}
\mathrm{S} & \rightarrow & \mathrm{NP\ VP} \\
\mathrm{NP} & \rightarrow & \mathrm{D\ N\ [PP]} \\
\mathrm{PP} & \rightarrow & \mathrm{P\ NP} \\
\mathrm{VP} & \rightarrow & \mathrm{V\ NP\ [Adv]\ [PP_M]} \\
\mathrm{PP_M} & \rightarrow & \mathrm{P_M\ NP} \mid \textit{leftward} \mid \textit{rightward} \mid \textit{from the left} \mid \textit{from the right}
\end{array}
\qquad
\begin{array}{lcl}
\mathrm{D} & \rightarrow & \textit{the} \\
\mathrm{N} & \rightarrow & \textit{person} \mid \textit{horse} \\
\mathrm{P} & \rightarrow & \textit{to the left of} \mid \textit{to the right of} \\
\mathrm{V} & \rightarrow & \textit{approached} \mid \textit{led} \mid \textit{rode} \\
\mathrm{Adv} & \rightarrow & \textit{quickly} \mid \textit{slowly} \\
\mathrm{P_M} & \rightarrow & \textit{towards} \mid \textit{away from}
\end{array}
$$

Fig. 5. The grammar for sentential queries used for the experiments in Section 7.

is first parsed according to the grammar shown in Fig. 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 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 Fig. 2 require arguments and each of these arguments has one of five thematic roles: *agent*, *patient*, *referent*, *source*, and *goal*. 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, θ, in Eq. (4).

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 even though the tracks produced are affected by it. This can be seen in Eq. (4) where neither f (the detection score), g (the motion coherence), nor either of their arguments depend on the (words in the) sentence. This allows us to preprocess the video corpus and compute object detections and optical-flow estimates which can be reused with different 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. 2 and 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*, *The person approached the horse*, and *The person 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 *The person approached 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.

7 Experiments

We present three experiments which test video retrieval using sentential queries. All three use the same video corpus but use different query corpora.

7.1 The *10 Westerns* Video Corpus

Our video corpus consists of 10 full-length Hollywood movies, nominally of the genre westerns. This corpus is very challenging and demonstrates the ability of our approach to handle videos found in the wild and not filmed specifically for this task: Black Beauty (Warner Brothers, 1994), The Black Stallion (MGM, 1979), Blazing Saddles (Warner Brothers, 1974), Easy Rider (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 1,187 minutes of video, roughly 20 hours. The appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2505297, specifies the duration and spatial and temporal resolution of each movie.

We temporally downsampled all videos to 6 fps but kept their original spatial resolutions, splitting them into 37,186 clips, each clip nominally being 18 frames (3 seconds) long, while overlapping the previous clip by six 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 [39] 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.

7.2 Query Corpora

We adopt the grammar from Fig. 5. This grammar allows for queries that describe people interacting with horses, hence our choice of genre for the video corpus, namely

westerns. We generated two of the three query corpora from this grammar. The first consisted of nine SVO queries generated by the grammar. We omitted the three SVO queries that involve people riding people, horses riding people, and horses riding horses. We refer to this collection of nine queries as the SVO queries. The second consisted of the 204 queries generated by the template included in the appendix, available in the online supplemental material. This consisted of all queries generated by the grammar in Fig. 5 except those that involve a PP in an NP and further restricting the lexical $\mathrm{PP_M}$ to be appropriate for the verb. For this query corpus we included all queries, including those that involve people riding people, horses riding people, and horses riding horses. We refer to this collection of 204 queries as the synthetic queries.

The third collection of queries was elicited from 300 distinct, disinterested, independent, and anonymous humans via Amazon Mechanical Turk through a mock up of our system, as described in the appendix. We obtained 3,000 unrestricted queries, completely unconstrained as to grammar and lexicon. We discarded 22 blank queries and 351 that violated the instructions given to workers, as described in the appendix. We processed all remaining 2,627 queries by mapping them to synthetic queries using a spelling and grammar correction process based on Levenshtein distance, as described in the appendix. We refer to this collection of 2,627 queries as the human queries. The mock up did not expose this spelling and grammar correction process to the workers, who simply entered queries, which were recorded, and obtained search results. We evaluated the truth of the retrieved results relative to the original human queries, not their mapping to the synthetic queries.

7.3 Models

A requirement for determining whether a video depicts a query, and the degree to which it depicts that query, is to detect the objects that might fill roles in that query. 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 PASCAL VOC. We use models provided with the software release associated with Sadeghi and Forsyth [33][1] which were trained by the UoCTTI_LSVM-MDPM team (the authors of Felzenszwalb et al. [31], [32]) for the 2009 Challenge. On the 2009 Challenge, the *person* model achieves an AP score of 41.5 percent and the *horse* model achieves an AP score of 38.0 percent. When running the object detectors, we set the non-maximal-suppression parameter to 0.7 and use at most the top 4 detections returned for each class in each frame.

We also require settings for the nine parameters, shown in Fig. 1, which are required to produce the predicates which encode the semantics of the words in this grammar. For this purpose, we judiciously selected values for these parameters that are consistent with their intent: $\mathbf{far} = 180$, $\mathbf{close} = 120$, $\mathbf{stationary} = 2$, $\mathbf{\Delta closing} = 3$, $\mathbf{\Delta angle} = 45°$, $\mathbf{\Delta pp} = 50$, $\mathbf{\Delta quickly} = 30$, $\mathbf{\Delta slowly} = 30$, and $\mathbf{overlap} = 0.1$. Yu and Siskind [40] present a strategy for training the parameters of a lexicon of words given a video corpus.

1. http://vision.cs.uiuc.edu/ftvq/

7.4 Baseline

We compared the performance of the sentence tracker against a baseline on the SVO queries. We compare against a baseline only for the SVO queries and not the synthetic and human queries because we know of no other system that can support the more complex syntax and ontology in these query corpora. This baseline employs the same approach that is used in state-of-the-art video-search systems in that it searches independently for the subject and object of an SVO query using object detection and the verb of an SVO query using event detection. We did not compare against any particular existing system because, at the time of submission, there was no system for which end-to-end code was publicly available.

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 object detector and pretrained models as used for the experiments with the sentence tracker, 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 verb. For this purpose, we employed one of the highest performing event detectors for which code was available at the time of submission, namely that of Kuehne et al. [21]. We train that detector on 70 positive and 70 negative samples of each verb and remove those samples from the test set. We then report the top 1, 3, 5, and 10 ranked clips that satisfy the event detector and compare those clips against the top 1, 3, 5, and 10 clips produced by our method.

7.5 Evaluation Procedure

For each query, we scored every clip paired with that query and return the top 1, 3, 5, and 10 best-scoring clips for that query. Each of these top 10 clips was annotated by a collection of nominally five distinct, disinterested, independent, and anonymous humans via Amazon Mechanical Turk. Each judge was presented with a query and associated hit and asked: is this query true of this clip? The precise details of how such assessment was performed are described in the appendix.

7.6 Results

Our results are summarized in Fig. 6. The left column summarizes the experiments with the SVO queries. Our approach yields significantly higher precision than the baseline on the SVO queries. Precision of the sentence tracker on the SVO queries varies as a function of recall as controlled by the threshold on the sentence-tracker score. Note that it is not possible to achieve high recall with our method, because we employ hard FSMs to model sentential semantics which cannot be overcome by any threshold on sentence-tracker score because such is $-\infty$ when the FSM is

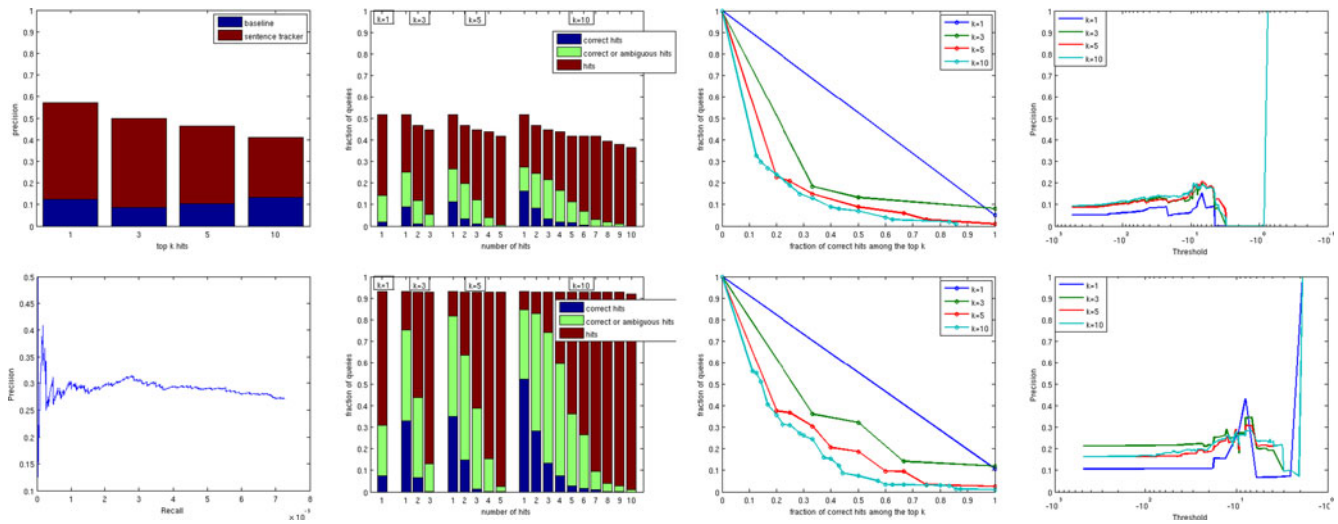

Fig. 6. (Top left) Comparison of average precision in the top 1, 3, 5, and 10 hits, over the SVO queries for both the baseline and the sentence tracker. (bottom left) Precision/recall curve over the SVO queries for the sentence tracker. Results for synthetic (top row) and human (bottom row) queries in the top 1, 3, 5, and 10 hits (right three columns). (second column) Fraction of queries with at least the the indicated number of hits, correct or ambiguous hits, and correct hits. (third column) Fraction of queries that have at least the indicated fraction of correct hits. (fourth column) Precision of returned hits as a function of threshold.

violated. Recall is thus limited to about 10^{-2}. Precision is around 0.3 for most of the attainable recall range. It reaches a peak of 0.5 when recall is about 2×10^{-5}. Its lowest value is 0.125 with a similar recall.

The right three columns summarize the experiments with the synthetic and human queries in the top and bottom rows respectively. For the second and third columns, no threshold on sentence-tracker score was employed; we evaluated the top 1, 3, 5, and 10 hits returned. Because of the stringent FSM model, the sentence tracker can return fewer than the requisite number of hits, even without a threshold. Thus the red bars in the second column depict the fraction of the queries for which at least the indicated number of hits were returned. Because the human judges were sometimes divided as to whether the queries were true of the hits, we classified these hits as correct, ambiguous, or incorrect, as described in the appendix. The green bars depict the fraction of the queries for which the indicated number of correct or ambiguous hits were returned, while the blue bars depict the fraction of the queries for which the indicated number of correct hits were returned.

The third column depicts the fraction of the queries that yield at least the indicated fraction of correct hits. For example, with the synthetic queries, slightly more than 30 percent of the queries yield 10 percent or more correct hits in the top 10. As a point of comparison, with the human queries, slightly more than 55 percent of the queries yield 10 percent or more correct hits in the top 10. Note that for much of the range, the precision in the top hits requested for human queries exceeds that of the synthetic queries.

The fourth column depicts the variation in average precision as a function of a threshold on the sentence-tracker score. As the threshold nears zero, the sentence tracker becomes very precise. As the threshold tends to $-\infty$, the average precision asymptotes. Again note that overall precision for the human queries is significantly higher than that of the synthetic queries over almost all of the range of thresholds.

We highlight the usefulness of this approach in Fig. 7 where we show one of the top few hits for a variety of different synthetic and human queries. Note that two pairs of similar queries, both *The person approached the horse* and *The horse approached the person* as well as *The person approached the horse slowly from the left* and *The horse approached the person slowly from the left*, yield different but appropriate results. With existing systems, both queries in each pair would provide the same hits as they treat sentences as conjunctions of words.

8 Discussion

As discussed in Section 1, most 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 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 [41], [42]. Some variants of these approaches, such as that of Siddiquie et al. [43], 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. [22], Worring et al. [23], and Snoek et al. [24].

None of the above approaches link features in a way that is informed by sentence structure, 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 query semantics with enough fidelity to differentiate between *The person rode the horse* and *The horse rode the person*. The baseline we compare against in

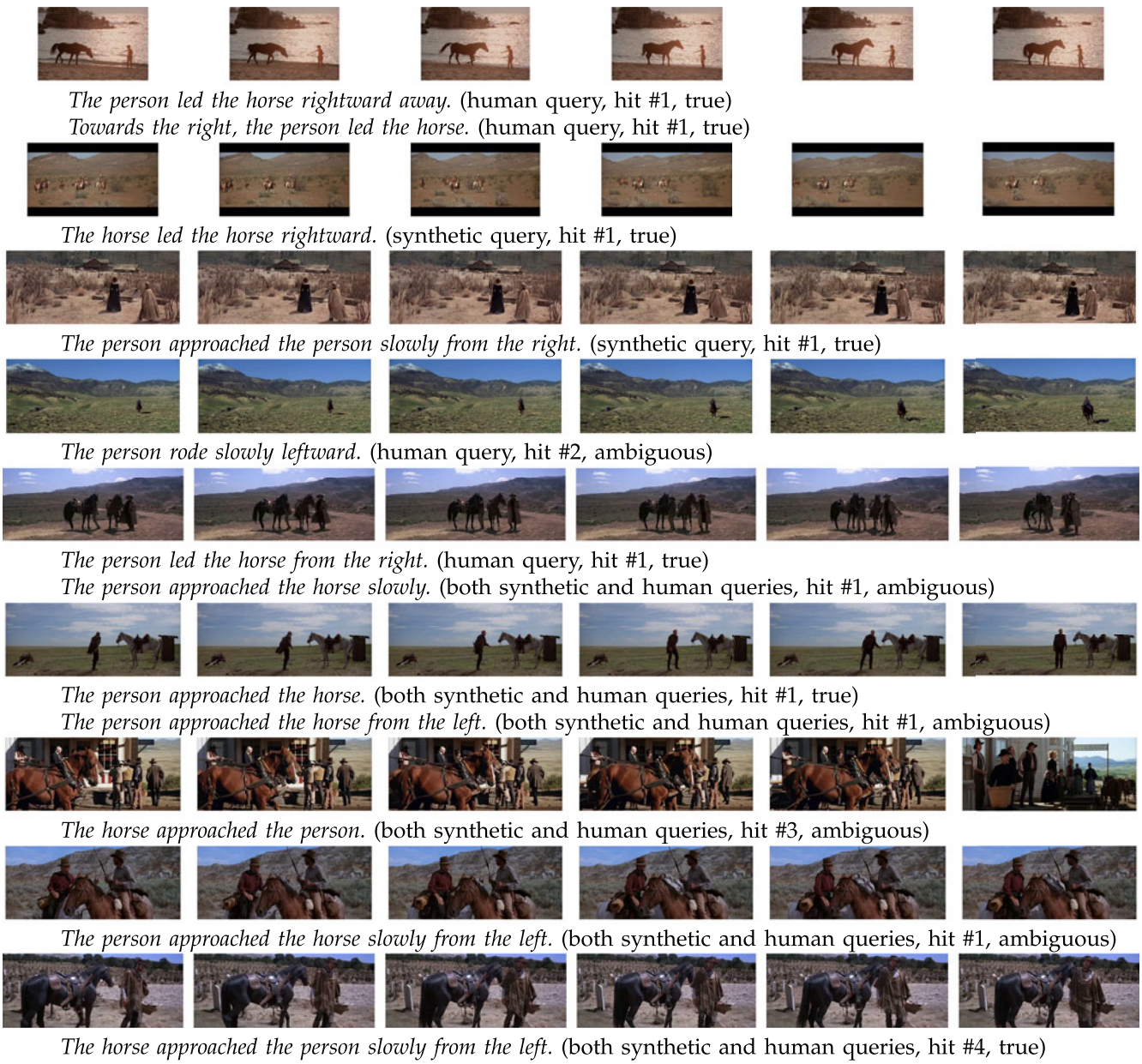

Fig. 7. Frames from hits returned for several synthetic and human queries. Some clips are returned for multiple queries. As indicated above, these hits were judged as correct or ambiguous for the associated query by human judges.

Section 7.4 was designed to model the predominant current methodology of modeling queries with no reflection of argument structure.

In the experiments in Section 7, we report recall only for the SVO queries. The reason is simple: computing recall requires determining true negatives which would require annotating the entire corpus of 37,186 clips with truth values for all 204 synthetic queries and 2,627 human queries, a monumental and tedious task. We only annotate the top ten hits for each of the synthetic and human 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 synthetic and human 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 target clips that satisfy complex queries are likely to be few and far between. 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 7.4. Thus we contend that the underlying retrieval task is difficult and the performance of our method, as described in Section 7, is good.

Our work is closest to that of Lin et al. [29]. They are the only work that we are familiar with that composes the meaning of complex textual video queries out of meanings of constituent words and phrases. Our work is similar to Lin et al. [29] in that neither learn the grammar, the lexicon, the method of mapping parse trees to semantic representations, or the concept vocabulary encoded in the ontology of predicates. It is instructive to examine the similarities and differences more closely. Like our work, they formulate the meaning of a query as a collection of cost functions applied to object tracks. Each

cost function corresponds to some word or phrase in the query. Each object track corresponds to some noun or noun phrase in the query. In our notation, they too apply cost functions h_w to object tracks b_j as $h_w(b_j)$ and seek to maximize the collective costs. But the similarity ends there. First, they employ solely unary cost functions that apply to a single track. We support cost functions of arbitrary arity, in particular binary functions. This allows us to support transitive verbs, like *The person rode the horse*, which would be represented as RODE(**person**, **horse**), and prepositional phrases attached to either nouns or verbs, like *the person to the left of the horse* and *The person rode towards the horse*, which would be represented as LEFT-OF(**person**, **horse**) and TOWARDS(**person**, **horse**). Handling such with only unary functions would require lexicalizing one of the arguments with the verb or preposition, as in *rode horse*, *left of the horse*, or *towards the horse*, which would be represented as RODE-HORSE(**person**), LEFT-OF-HORSE(**person**), or TOWARDS-HORSE(**person**). Such would not generalize because the associate predicates are specialized to specific incorporated arguments.

Second, they perform object tracking as a preprocessing step, having each object tracker produce a collection of candidate tracks which are then evaluated by semantic processing. Their semantic processing is disjoint from and subsequent to object tracking. Their trackers overgenerate and their semantic processing selects among such overgenerated tracks. Our approach performs object detection as a preprocessing step, having each object detector produce a collection of detections in each frame which are evaluated by our joint tracking and semantic processing step. Our object detectors overgenerate and our joint tracking and semantic process selects among such detections and assembles them into tracks. The difference lies in the fact that an exponential number of possible tracks can be constructed from a given number of detections. Our method considers all such and finds the global optimum in polynomial time. Their method considers only the tracks produced by preprocessing.

Third, they allow a cost function to be assigned to a dummy track "no-obj," effectively removing the cost function from the semantic representation of the text query. They do this to support resilience to tracker failures. However, this allows their system to return hits that match a subset of the cost functions in the query, potentially returning hits that do not depict the query and for which the query is semantically false. We avoid the necessity to do so by performing tracking jointly with semantic processing. When we fail to find a track that satisfies the semantic processing we refrain from returning the clip as a hit.

Fourth, they allow a given track to be associated with at most one cost function. They refer to such as "no coreference." This means that they would represent *The person rode the horse* as RODE(**person**, **horse**) where **person** and **horse** are not cost functions but rather filters on potential tracks to consider as arguments to the cost function RODE. We instead represent this as the aggregation of cost functions PERSON(x), HORSE(y), and RODE(x, y). The no-coreference constraint would preclude such. This further precludes sentences like *The person rode the horse towards the person* which would require an aggregation of PERSON(x), HORSE(y), PERSON(z), RODE(x, y), and either TOWARDS(x, z) or TOWARDS(y, z). We see no way to represent such without coreference.

Fifth, their cost functions apply to feature vectors extracted from entire object tracks. This requires that any temporal alignment must be performed inside the feature-extraction process since the temporal nature of the track, i.e., the sequence of detections, is not exposed to the semantic cost functions. Our cost functions take the form of FSMs where the output predicates apply to detections in individual frames, not entire tracks. This allows the FSM to perform temporal alignment within the semantic processing, not in the feature-extraction process.

In the future, one can imagine scaling our approach along a variety of axes: larger, 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 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 6, many components of the process, such as object detection, can be precomputed and cached in a query-independent fashion. Moreover, as discussed in Section 6, 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. [44] 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 [38]. However, this has only been 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 perception and cognition beyond motion profile, such as body posture [45], functionality, intention, and physical processes.

Our current implementation processes all ten full-length Hollywood movies with all 204 synthetic queries and all 2,627 human queries with no precomputed information in about a day on ten workstations. With precomputed object detections and optical flow, it processes all nine SVO queries against all ten full-length Hollywood movies in about a half an hour on ten workstations. One can easily imagine processing every Hollywood movie ever made on the computing infrastructure available to an organization like Google in a few seconds.

Scaling query 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 lattice increases exponentially in the number of participants and the query length. However, as our corpus of human queries shows, people rarely enter long queries with a large number of participants. Scaling beyond our current capacity will require either a faster dynamic-programming algorithm or inexact inference. Barbu et al. [37] present an algorithm which employs Felzenszwalb and Huttenlocher's [46] 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 used 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 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. Lazy Viterbi [47] offers another alternative which maintains the optimality of the algorithm but only visits nodes in the lattice as needed.

9 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 largely 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 most other approaches which allow for textual queries of images or videos, we do not require any prior video annotation. We have evaluated this approach on a large video corpus of 10 full-length Hollywood movies, comprising roughly 20 hours of video, by running 2,627 naturally elicited queries.

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0060. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory or the US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

[1] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern.*, vol. 41, no. 6, pp. 797–819, Nov. 2011.

[2] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby talk: Understanding and generating simple image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1601–1608.

[3] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *arXiv:1412.2306*, 2014.

[4] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? Text-to-image coreference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3558–3565.

[5] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1470–1477.

[6] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video," in *Proc. ACM Int. Conf. Multimedia*, 2003, pp. 11–20.

[7] A. Anjulan and N. Canagarajah, "A unified framework for object retrieval and mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 63–76, Jan. 2009.

[8] D. Byrne, A. R. Doherty, C. G. M. Snoek, G. J. F. Jones, and A. F. Smeaton, "Everyday concept detection in visual lifelogs: Validation, relationships, and trends," *Multimedia Tools Appl.*, vol. 49, no. 1, pp. 119–144, 2010.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[10] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1745–1752.

[11] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III, "Midge: Generating image descriptions from computer vision detections," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2012, pp. 747–756.

[12] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retrieval*, vol. 2, no. 4, pp. 215–322, 2008.

[13] N. Siddharth, A. Barbu, and J. M. Siskind, "Seeing what you're told: Sentence-guided activity recognition in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 732–739.

[14] A. Barbu, N. Siddharth, and J. M. Siskind, "Language-driven video retrieval," in *Proc. CVPR Workshop Vis. Meets Cognition*, 2014, http://engineering.purdue.edu/~qobi.

[15] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 1, pp. 609–612.

[16] S. Sadanand and J. J. Corso, "Action Bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1234–1241.

[17] C.-Y. Chen and K. Grauman, "Watching unlabeled videos helps learn new human actions from very few labeled snapshots," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 572–579.

[18] Y. Song, L.-P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3562–3569.

[19] I. Everts, J. C. van Gemert, and T. Gevers, "Evaluation of color STIPs for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2850–2857.

[20] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2642–2649.

[21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.

[22] M. G. Christel, C. Huang, N. Moraveji, and N. Papernick, "Exploiting multiple modalities for interactive video retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, vol. 3, pp. 1032–1035.

[23] M. Worring, C. G. M. Snoek, O. De Rooij, G. P. Nguyen, and A. W. M. Smeulders, "The MediaMill semantic video search engine," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, vol. 4, pp. 1213–1216.

[24] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders, "A learned lexicon-driven paradigm for interactive video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 280–292, Feb. 2007.

[25] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Story-based video retrieval in TV series using plot synopses," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2014, pp. 137–144.

[26] Y. Aytar, M. Shah, and J. Luo, "Utilizing semantic word similarity measures for video retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[27] N. İkizler and D. A. Forsyth, "Searching video for complex activites with finite state models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[28] N. İkizler and D. A. Forsyth, "Searching for complex human activities with no visual examples," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 337–357, 2008.
[29] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual semantic search: Retrieving videos via complex textual queries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2657–2664.
[30] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 595–603.
[31] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 2241–2248.
[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IIEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[33] M. A. Sadeghi and D. Forsyth, "Fast template evaluation with vector quantization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2949–2957.
[34] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.
[35] A. J. Viterbi, "Error bounds for convolutional codes and an asymtotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–267, Apr. 1967.
[36] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun.*, vol. C-19, no. 5, pp. 751–772, Oct. 1971.
[37] A. Barbu, N. Siddharth, A. Michaux, and J. M. Siskind, "Simultaneous object detection, tracking, and event recognition," *Adv. Cognitive Syst.*, vol. 2, pp. 203–220, 2012.
[38] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. J. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. W. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang, "Video in sentences out," in *Proc. Conf. Uncertainty Artif. Intell.*, 2012, pp. 102–112.
[39] M. Cooper, T. Liu, and E. Rieffel, "Video segmentation via temporal pattern classification," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 610–618, Apr. 2007.
[40] H. Yu and J. M. Siskind, "Grounded language learning from video described with sentences," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2013, pp. 53–56.
[41] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk, "Attribute-based people search in surveillance environments," in *Proc. Workshop Appl. Comput. Vis.*, 2009, pp. 1–8.
[42] N. Kumar, P. Belhumeur, and S. Nayar, "FaceTracer: A search engine for large collections of images with faces," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 340–353.
[43] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 801–808.
[44] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1814–1821.
[45] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1365–1372.
[46] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory Comput.*, vol. 8, no. 1, pp. 415–428, 2012.
[47] J. Feldman, I. Abou-Faycal, and M. Frigo, "A fast maximum-likelihood decoder for convolutional codes," in *Proc. IEEE Veh. Technol. Conf.*, 2002, vol. 1, pp. 371–375.

Daniel Paul Barrett received the BSCmpE degree from Purdue University in 2011. He is currently working toward the PhD degree in the School of Electrical and Computer Engineering, Purdue University. His research interests include computer vision, robotics, and artificial intelligence, particularly their intersection, where a robot perceives and learns about the world through noisy real-world camera input. He is a student member of the IEEE.

Andrei Barbu received the BCS degree from the University of Waterloo in 2008 and the PhD degree in electrical and computer engineering from Purdue University in 2013. He is currently a postdoctoral fellow at the MIT Computer Science and Artificial Intelligence Laboratory. 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. He is a student member of the IEEE.

N. Siddharth received the BE degree in electronics and communication engineering from Anna University, Chennai, India, in 2008 and the PhD degree in electrical and computer engineering from Purdue University in 2014. He is currently a postdoctoral fellow at the Department of Psychology, Stanford University. His research interests include artificial intelligence, computer vision, computational linguistics, machine learning, robotics, cognitive science, and cognitive neuroscience. He is a member of the IEEE.

Jeffrey Mark Siskind received the BA degree in computer science from the Technion, Israel Institute of Technology in 1979, the SM degree in computer science from MIT in 1989, and the PhD degree in computer science from MIT in 1992. He did a postdoctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science from 1992 to 1993. He was an assistant professor at the Department of Computer Science, University of Toronto, from 1993 to 1995, a senior lecturer at the Technion Department of Electrical Engineering in 1996, a visiting assistant professor at the Department of Computer Science and Electrical Engineering, University of Vermont from 1996 to 1997, and a research scientist at NEC Research Institute, Inc., from 1997 to 2001. He joined the School of Electrical and Computer Engineering, Purdue University, in 2002, where he is currently an associate professor. He is a senior member of the IEEE.