Advances in Cognitive Systems 2 (2012) 203-220 Submitted 9/2012; published 12/2012

Simultaneous Object Detection, Tracking, and Event Recognition

Andrei Barbu ANDREI@0XAB.COM
Siddharth Narayanaswamy SIDDHARTH@IFFSID.COM
Aaron Michaux AMICHAUX@PURDUE.EDU
Jeffrey Mark Siskind QOBI@PURDUE.EDU
Purdue University, School of Electrical and Computer Engineering, 465 Northwestern Avenue, West Lafayette, IN 47907-2035 USA

Abstract

Integrating information across modalities is a long-standing challenge for cognitive systems. The common internal structure and algorithmic organization of object detection, detection-based tracking, and event recognition facilitates a general approach to integrating these three components. This supports multidirectional information flow between these components allowing object detection to influence tracking and event recognition; and event recognition to influence tracking and object detection. The performance of the combination can exceed the performance of the components in isolation when inspecting the quality of the object tracks produced. We demonstrate this qualitatively on a number of videos which show how failures in each of the components are resolved when they are integrated together. This can be done with linear asymptotic complexity.

1. Introduction

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 outperform 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 problem of object detection and tracking to the computer-vision community (Auer et al., 2005; Cohn et al., 2006), assuming that community can, or someday will, deliver a module that can reliably and categorically detect and track objects, yielding symbolic representations like `ON(ball,ground)` and `HOLD(HAND(person),ball)`. The cognitive-systems community assumes such representations are available as input to subsequent processing, for example, inferring `PICKUP(person, ball)` from a transition from `ON(ball,ground)` to `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 (Siskind & Morris, 1996; Starner, Weaver, & Pentland, 1998; Wang et al., 2009; Barbu et al., 2012) classify events based on their motion profile. This requires detecting and tracking the event participants. Adaptive approaches to tracking (Yilmaz, Javed, & Shah, 2006), such as Kalman filtering (Comaniciu, Ramesh, & Meer, 2003), 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% (Everingham et al., 2010). 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 2, of current object detectors (Felzenszwalb, Girshick, & McAllester, 2010), detection-based trackers (Wolf, Viterbi, & Dixon, 1989), and HMM-based approaches to event recognition (Baum & Petrie, 1966; Siskind & Morris, 1996; Starner, Weaver, & Pentland, 1998) facilitates a general approach to integrating these three components. We demonstrate an approach to integrating object detection and tracking (in Section 4), an approach to integrating tracking and event recognition (in Section 5), an approach to integrating object detection, tracking, and event recognition (in Section 6), and show how it improves each of the these three components in isolation (in Section 7). 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.

2. Detection-Based Tracking

The methods described in Sections 4, 5, and 6 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. (2010) 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 $\mathbf{j} = \langle j_1, \ldots, j_T \rangle$, one in each frame, denotes a track consisting of detections $b_{j_t}^t$ in each frame t. For example, in Figure 1, we might have the track $\mathbf{j} = \langle 1, \ldots, 1 \rangle$ that consists of detections $\langle b_1^1, \ldots, b_1^T \rangle$. 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 $\mathbf{j} = \langle j_1, \ldots, j_T \rangle$ as

$$\max_{j_1,\ldots,j_T} \left(\sum_{t=1}^{T} f(b_{j_t}^t) + \sum_{t=2}^{T} g(b_{j_{t-1}}^{t-1}, b_{j_t}^t) \right), \tag{1}$$

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 $b_{j_t}^t$ and the center of $b_{j_{t-1}}^{t-1}$ projected forward one frame, though, as described below, our approach is compatible with a variety of functions discussed by Felzenszwalb and Huttenlocher (2004). The forward projection internal to g can be computed in a variety of ways including by using optical flow and the Kanade-Lucas-Tomasi (KLT; Tomasi & Kanade, 1991) feature tracker.

We answer question 4 by observing that Equation (1) can be optimized in polynomial time using dynamic-programming with the Viterbi (1971) algorithm. Equation (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 δ_j^t and inductively computes all of the δ values in frame $t+1$ from those in frame t. This factors the optimization over an exponential set into a polynomial-time process:

$$\begin{aligned} &\textbf{for } j = 1 \textbf{ to } J_1 \textbf{ do } \delta_j^1 := f(b_j^1) \\ &\textbf{for } t = 2 \textbf{ to } T \textbf{ do } \left\{ \textbf{for } j = 1 \textbf{ to } J_t \textbf{ do } \delta_j^t := f(b_j^t) + \max_{j'=1}^{J_{t-1}} \left(g(b_{j'}^{t-1}, b_j^t) + \delta_{j'}^{t-1} \right) \right\} \end{aligned} \tag{2}$$

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

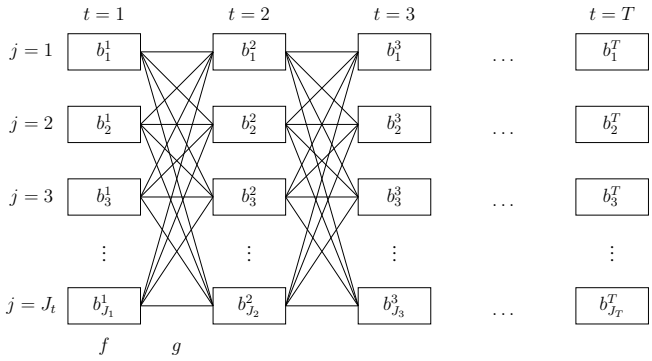

Figure 1. The tracking lattice constructed by the Viterbi algorithm performing detection-based tracking.

(Wu et al., 2008). 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 (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 (Everingham et al., 2010). 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. (2010) 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. The operation of a detection-based tracker is illustrated in Figure 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)

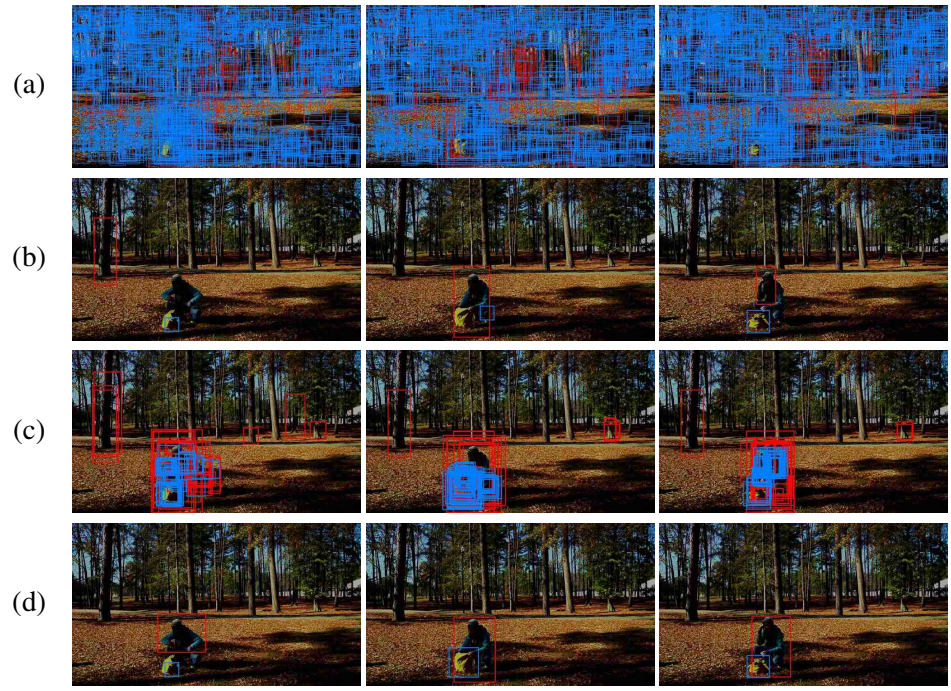

Figure 2. The operation of a detection-based tracker. Three frames from the same video are shown, one in each column. The rows indicate successive information computed by the tracker for each frame. (a) Output of the detection sources, biased to yield false positives. (b) The top-scoring output of the 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.

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. (2010) 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.

Table 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.

N	SBU	SRI	UCB	USC
SBU		8	20	8
SRI	8		1201	95
UCB	20	1201		204
USC	8	95	204	
us	48	1254	1829	360

(a)

μ	SBU	SRI	UCB	USC
SBU		0.76	0.68	0.59
SRI	0.76		0.55	0.59
UCB	0.68	0.55		0.48
USC	0.59	0.59	0.48	
us	0.54	0.40	0.35	0.43

(b)

σ	SBU	SRI	UCB	USC
SBU		0.06	0.14	0.10
SRI	0.06		0.27	0.16
UCB	0.14	0.27		0.23
USC	0.10	0.16	0.23	
us	0.26	0.24	0.23	0.20

(c)

3. 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 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 1(a).

The overall mean and standard deviation measures, reported in Table 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 (Barbu et al., 2012).

4. 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 (Freeman & Roth, 1995) 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 (Felzenszwalb & Huttenlocher, 2004) that allows it to scale linearly with the number of image-pyramid positions. Given a set $\mathcal{G}$ 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 $\phi : \mathcal{G} \to \mathbb{R}$, the generalized distance transform $D_\phi(q)$ computes

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

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,

$$d(b_{xys}, b'_{x'y's'}) = (\pi(s)x - \pi(s')x')^2 + (\pi(s)y - \pi(s')y')^2 + \alpha(s - s')^2, \tag{3}$$

where x and y denote the original image coordinates of a detection center at scale s. Nominally, detections are boxes. Comparing two such boxes involves 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 α 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_{xys}^t, lattice values δ_j^t with δ_{xys}^t, and Equation (2) with:

$$\begin{array}{l}\textbf{for } x = 1 \textbf{ to } X \textbf{ do } \left\{\textbf{for } y = 1 \textbf{ to } Y \textbf{ do } \left\{\textbf{for } s = 1 \textbf{ to } S \textbf{ do } \delta^1_{xys} := f(b^1_{xys})\right\}\right\} \\ \textbf{for } t = 2 \textbf{ to } T \\ \quad \textbf{do for } x = 1 \textbf{ to } X \\ \quad\quad \textbf{do for } y = 1 \textbf{ to } Y \\ \quad\quad\quad \textbf{do for } s = 1 \textbf{ to } S \textbf{ do } \delta^t_{xys} := f(b^t_{xys}) + \max\limits_{x',y',s'} \left(g(b^{t-1}_{x'y's'}, b^t_{xys}) + \delta^{t-1}_{x'y's'}\right)\end{array} \tag{4}$$

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 (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 α into Equation (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 δ^t_j for all $1 \leq j \leq J_t$ from Equation (2) is being represented as a pyramid and the loop

$$\textbf{for } j = 1 \textbf{ to } J_t \textbf{ do } \delta^t_j := f(b^t_j) + \max_{j'=1}^{J_{t-1}} \left(g(b^{t-1}_{j'}, b^t_j) + \delta^{t-1}_{j'}\right) \tag{5}$$

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^t_{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 (1):

$$\max_{\substack{x_1,\ldots,x_T \\ y_1,\ldots,y_T \\ s_1,\ldots,s_T}} \left(\sum_{t=1}^{T} f(b^t_{x_t y_t s_t}) + \sum_{t=2}^{T} g(b^{t-1}_{x_{t-1} y_{t-1} s_{t-1}}, b^t_{x_t y_t s_t})\right) \tag{6}$$

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.

5. Combining Tracking and Event Recognition

It is popular to use hidden Markov models (HMMs) to perform event recognition (Siskind & Morris, 1996; Starner, Weaver, & Pentland, 1998; Wang et al., 2009; Barbu et al., 2012). When doing so, the log likelihood of a video conditioned on an event model is

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

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 (Baum & Petrie, 1966), 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,

$$\max_{k_1,\ldots,k_T} \left(\sum_{t=1}^{T} h(k_t, b_{j_t^*}^t) + \sum_{t=2}^{T} a(k_{t-1}, k_t)\right), \tag{7}$$

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 (1) with Equation (7) to yield a unified cost function

$$\max_{j_1,\ldots,j_T} \max_{k_1,\ldots,k_T} \left(\sum_{t=1}^{T} f(b_{j_t}^t) + \sum_{t=2}^{T} g(b_{j_{t-1}}^{t-1}, b_{j_t}^t) + \sum_{t=1}^{T} h(k_t, b_{j_t}^t) + \sum_{t=2}^{T} a(k_{t-1}, k_t)\right) \tag{8}$$

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 δ_{jk}^t 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:

$$\begin{aligned}
&\textbf{for } j = 1 \textbf{ to } J_1 \textbf{ do } \left\{\textbf{for } k = 1 \textbf{ to } K \textbf{ do } \delta_{jk}^1 := f(b_j^1) + h(k, b_j^1)\right\} \\
&\textbf{for } t = 2 \textbf{ to } T \\
&\quad \textbf{do for } j = 1 \textbf{ to } J_t \\
&\qquad \textbf{do for } k = 1 \textbf{ to } K \\
&\qquad\quad \textbf{do } \delta_{jk}^t := f(b_j^t) + h(k, b_j^t) + \max_{j'=1}^{J_{t-1}} \max_{k'=1}^{K} \left(g(b_{j'}^{t-1}, b_j^t) + a(k', k) + \delta_{j'k'}^{t-1}\right)
\end{aligned} \tag{9}$$

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 (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 (Felzenszwalb, Huttenlocher, & Kleinberg, 2003).

Two practical issues arise when applying this method. First, one can factor Equation (10) as Equation (11):

$$\max_{j'=1}^{J_{t-1}} \max_{k'=1}^{K} \left(g(b_{j'}^{t-1}, b_j^t) + a(k', k) + \delta_{j'k'}^{t-1} \right) \tag{10}$$

$$\max_{j'=1}^{J_{t-1}} \left(g(b_{j'}^{t-1}, b_j^t) + \max_{k'=1}^{K} \left(a(k', k) + \delta_{j'k'}^{t-1} \right) \right) \tag{11}$$

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 (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 9, one for each event class, and select the highest-scoring result. Doing such *will* perform simultaneous tracking and event recognition.

6. Combining Object Detection, Tracking, and Event Recognition

One can combine the methods of Sections 4 and 5 to optimize a cost function,

$$\max_{\substack{x_1,\ldots,x_T \\ y_1,\ldots,y_T \\ s_1,\ldots,s_T \\ k_1,\ldots,k_T}} \left(\sum_{t=1}^{T} f(b_{x_t y_t s_t}^t) + h(k_t, b_{x_t y_t s_t}^t) + \sum_{t=2}^{T} g(b_{x_{t-1} y_{t-1} s_{t-1}}^{t-1}, b_{x_t y_t s_t}^t) + a(k_{t-1}, k_t) \right), \tag{12}$$

that combines Equation (6) with Equation (8) by forming a large Viterbi lattice with values δ_{xysk}^t.

One practical issue arises when applying the above method. In Equation (12), h is a function of $b_{x_t y_t s_t}^t$, 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 (Siskind & Morris, 1996; Starner, Weaver, & Pentland, 1998). This means that h must also be a function of $b_{x_{t-1} y_{t-1} s_{t-1}}^{t-1}$, 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. 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.

7. Demonstration

We demonstrate the effectiveness of the approach presented in this paper by qualitatively assessing its ability to track objects. Figure 3 demonstrates improved performance of simultaneous object detection and tracking (c), as computed by the methods in Section 4, over object detection (a) and tracking (b) in isolation, as computed by the methods in Section 2. 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.

Figure 4 demonstrates improved performance of simultaneous tracking and event recognition (c), as computed by the methods in Section 5, over tracking (b) in isolation, as computed by the methods in Section 2. 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 2 and human-labeled event occurrences. Articulated appearance

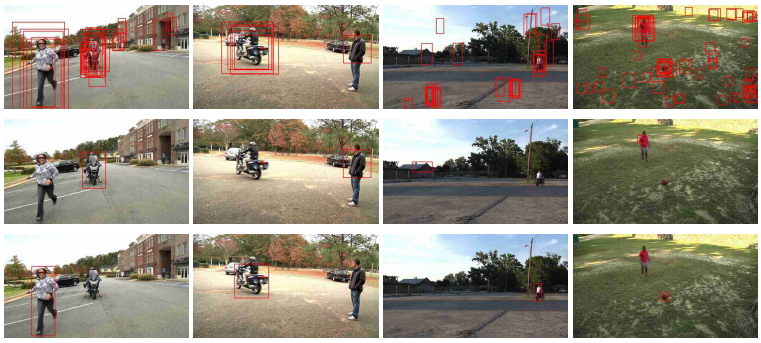

Figure 3. Improved performance of simultaneous object detection and tracking. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector using models for people, motorcycles, and balls. (b) Tracks produced by detection-based tracking, as described in Section 2. (c) Tracks produced by simultaneous object detection and tracking, as described in Section 4.

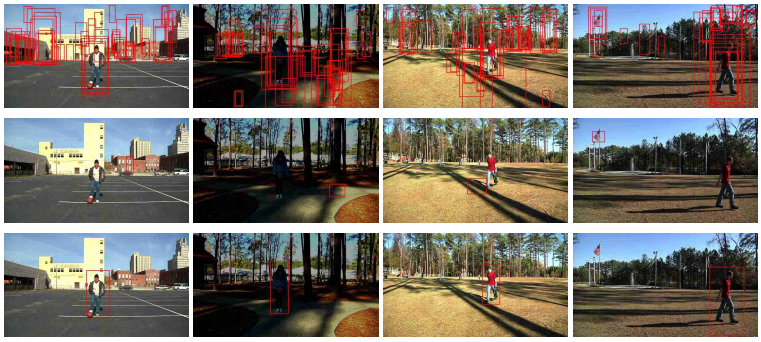

Figure 4. Improved performance of simultaneous tracking and event recognition. A single frame from each of four different videos is shown. Rows depict the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 2. (c) Tracks produced by simultaneous tracking and event recognition, as described in Section 5.

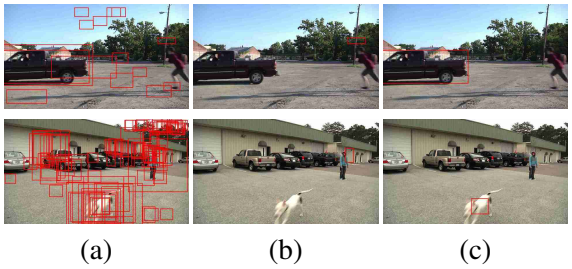

Figure 5. Improved performance of simultaneous object detection, tracking, and event recognition. A single frame from each of four different videos is shown. Each column depicts the output of a different method when processing that frame. (a) Output of the Felzenszwalb et al. detector. (b) Tracks produced by detection-based tracking, as described in Section 2. (c) Tracks produced by simultaneous object-detection, tracking, and event recognition, as described in Section 6.

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 5 demonstrates improved performance of simultaneous object detection, tracking, and event recognition (c), as computed by the methods in Section 6, over object detection (a) and tracking (b) in isolation, as computed by the methods in Section 2. As before, these results were obtained with object and event models that were trained independently.

8. Related Work

Detection-based tracking using dynamic programming has a long history (Wolf, Viterbi, & Dixon, 1989; Castanon, 1990), as do motion-profile-based approaches to event recognition using HMMs (Siskind & Morris, 1996; Starner, Weaver, & Pentland, 1998; Wang et al., 2009). Moreover, there have been attempts to integrate object detection and tracking (Li & Nevatia, 2008; Pirsiavash, Ramanan, & Fowlkes, 2011), tracking and event recognition (Li & Chellappa, 2002); and object detection and event recognition (Moore, Essa, & Heyes, 1999; Peursum, West, & Venkatesh, 2005; Gupta & Davis, 2007). 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 in Section 1, 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 (Qureshi, Terzopoulos, & Jasiobedzki, 2004). 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 (Auer et al., 2005).

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.

9. 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 2 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 (Felzenszwalb & Huttenlocher, 2004). Other kinds of generalized distance transforms, such as those based on the Legendre transform (Lucet, 2009), 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 (Baum et al., 1970; Baum, 1972) with the training procedure for the object models (Felzenszwalb, Girshick, & McAllester, 2010). 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 (Viterbi, 1971) 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.

Acknowledgements

This work was supported, in part, by NSF Grant No. CCF-0438806, NRL Contract No. N00173-10-1-G023, ARL Cooperative Agreement No. W911NF-10-2-0060, and the Rosen Center for Advanced Computing. Any views or conclusions expressed in this document are those of the authors and do not necessarily reflect or represent the views or official policies, expressed or implied, of NSF, NRL, ONR, ARL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

Auer, P., et al. (2005). *A research roadmap of cognitive vision* (Technical Report 5). European Research Network for Cognitive Computer Vision Systems.

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S. J., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., Schmidt, L., Shangguan, J., Siskind, J. M., Waggoner, J. W., Wang, S., Wei, J., Yin, Y., & Zhang, Z. (2012). Video in sentences out. *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence* (pp. 102–112). Catalina Island, CA: AUAI Press.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, *3*, 1–8.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, *37*, 1554–1563.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, *41*, 164–171.

Castanon, D. (1990). Efficient algorithms for finding the K best paths through a trellis. *IEEE Transactions on Aerospace and Electronic Systems*, *26*, 405–410.

Cohn, A. G., Hogg, D. C., Bennett, B., Devin, V. E., Galata, A., Magee, D. R., Needham, C. J., & Santos, P. E. (2006). Cognitive vision: Integrating symbolic qualitative representations with computer vision. In H. I. Christensen & H.-H. Nagel (Eds.), *Cognitive vision systems*, Vol. 3948 of *Lecture Notes in Computer Science*, 221–246. Dagstuhl, Germany: Springer-Verlag.

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 564–575.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010). Cascade object detection with deformable part models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2241–2248). San Francisco, CA: IEEE Press.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). *Distance transforms of sampled functions* (Technical Report TR2004-1963). Cornell Computing and Information Science.

Felzenszwalb, P. F., Huttenlocher, D. P., & Kleinberg, J. M. (2003). Fast algorithms for large-state-space HMMs with applications to web usage analysis. *Neural Information Processing Systems Conference*. Vancouver, Canada: MIT Press.

Freeman, W. T., & Roth, M. (1995). Orientation histograms for hand gesture recognition. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition* (pp. 296–301). Zurich, Switzerland: IEEE Press.

Gupta, A., & Davis, L. S. (2007). Objects in action: An approach for combining action understanding and object perception. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 1–8). Minneapolis, MN: IEEE Press.

Li, B., & Chellappa, R. (2002). A generic approach to simultaneous tracking and verification in video. *IEEE Transactions on Image Processing*, *11*, 530–544.

Li, Y., & Nevatia, R. (2008). Key object driven multi-category object recognition, localization, and tracking using spatio-temporal context. *Proceedings of the Tenth European Conference on Computer Vision* (pp. 409–22). Marseille, France: Springer-Verlag.

Lucet, Y. (2009). New sequential exact Euclidean distance transform algorithms based on convex analysis. *Image Vision Computation*, *27*, 37–44.

Moore, D. J., Essa, I. A., & Heyes, M. H. (1999). Exploiting human actions and object context for recognition tasks. *Proceedings of the Seventh International Conference on Computer Vision* (pp. 80–86). Corfu, Greece: IEEE Press.

Peursum, P., West, G., & Venkatesh, S. (2005). Combining image regions and human activity for indirect object recognition in indoor wide-angle views. *Proceedings of the Tenth International Conference on Computer Vision* (pp. 82–89). Beijing, China: IEEE Press.

Pirsiavash, H., Ramanan, D., & Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 1201–1208). Colorado Springs, CO: IEEE Press.

Qureshi, F., Terzopoulos, D., & Jasiobedzki, P. (2004). A cognitive vision system for space robotics. *Proceedings of the Workshop on Applications of Computer Vision* (pp. 120–128). Prague, Czech Republic: IEEE Press.

Siskind, J. M., & Morris, Q. (1996). A maximum-likelihood approach to visual event classification. *Proceedings of the Fourth European Conference on Computer Vision* (pp. 347–360). Cambridge, UK.

Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 1371–1375.

Tomasi, C., & Kanade, T. (1991). *Detection and tracking of point features* (Technical Report CMU-CS-91-132). Carnegie Mellon University, Pittsburgh, PA.

Viterbi, A. J. (1971). Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication*, *19*, 751–772.

Wang, Z., Kuruoglu, E. E., Yang, X., Xu, Y., & Yu, S. (2009). Event recognition with time varying hidden Markov model. *Proceedings of the 34th International Conference on Accoustics, Speech, and Signal Processing* (pp. 1761–1764). Taipei, Taiwan: IEEE Press.

Wolf, J., Viterbi, A., & Dixon, G. (1989). Finding the best set of K paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, *25*, 287–296.

Wu, B., Zhang, L., Singh, V. K., & Nevatia, R. (2008). Robust object tracking based on detection with soft decision. *Proceedings of the 2008 IEEE Workshop on Motion and Video Computing* (pp. 1–8). Washington, DC: IEEE Press.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computational Surveys*, *38*.