

# Actor-Action Semantic Segmentation with Grouping Process Models

Chenliang Xu and Jason J. Corso  
Electrical Engineering and Computer Science  
University of Michigan, Ann Arbor  
Fcl i angxu, j j corso@umich. edu

## Abstract

*Actor-action semantic segmentation made an important step toward advanced video understanding: what action is happening; who is performing the action; and where is the action happening in space-time. Current methods based on layered CRFs for this problem are local and unable to capture the long-ranging interactions of video parts. We propose a new model that combines the labeling CRF with a supervoxel hierarchy, where supervoxels at various scales provide cues for possible groupings of nodes in the CRF to encourage adaptive and long-ranging interactions. The new model defines a dynamic and continuous process of information exchange: the CRF influences what supervoxels in the hierarchy are active, and these active supervoxels, in turn, affect the connectivities in the CRF; we hence call it a grouping process model. By further incorporating the video-level recognition, the proposed method achieves a large margin of 60% relative improvement over the state of the art on the recent A2D large-scale video labeling dataset, which demonstrates the effectiveness of our modeling.*

## 1. Introduction

Advances in modern high-level computer vision have helped usher in a new era of capable, perceptive physical platforms, such as automated vehicles. As the performance of these systems improves, the expectations of their capabilities and tasks will also increase, commensurately, with platforms moving from the highways into our homes, for example. The need for these platforms to understand not only **what** action is happening, but also **who** is doing the action and **where** is the action happening in space-time, will be increasingly critical to extracting semantics from videos and, ultimately, to interacting with humans in our complex world. For example, a home kitchen robot must distinguish and locate *adult-eating*, *dog-eating* and *baby-crying* in order to decide how to prepare and when to serve food.

Despite the recent successes in many aspects of this problem, such as action recognition [8, 15, 19, 30, 33, 37,

38], action segmentation [11, 22] and video object segmentation [7, 20, 21, 26, 28, 46], the collective problem had not been codified until [40], which posed a new actor-action semantic segmentation task on a large-scale YouTube video dataset called A2D. This dataset contains seven classes of actors including both articulated (e.g. *baby*, *cat* and *dog*) and rigid (e.g. *car* and *ball*) ones, and eight classes of actions (e.g. *flying*, *walking* and *running*). The task is to label each pixel in a video with a pair of actor and action labels or a null actor/action; one third of the A2D videos contain multiple actors and actions.

This task is challenging—the benchmarked leading method, the *trilayer* model, only achieves a 26.46% per-class accuracy for the joint actor-action video labeling [40]. The method builds a large three-layer CRF on video supervoxels, where random variables are defined for sets of actor, actor-action, and action labels, respectively. It connects layers with potential functions that capture conditional probabilities (e.g. conditional distribution of actions given a specific actor class). Although the model accounts for the interplay of actors and actions, the interactions are restricted to the local CRF neighborhoods, which, based on the low absolute performance, is insufficient to solve this unique actor-action problem for three reasons.

First, we believe the pixel-level model must be married to a secondary process that captures instance-level or video-level global information in order to properly model the actors performing actions. Lessons learned from images strongly supports this argument—the performance of semantic image segmentation on the MSRC dataset seems to hit a plateau [31] until information from secondary processes, such as context [16, 25], object detectors [17] and a holistic scene model [43], are added. However, to the best of our knowledge, there is no method in video semantic segmentation that directly leverages the recent successes in action recognition.

Second, the two sets of labels, actors and actions, exist at different granularities. For example, we want to label *adult-clapping* in a video. The actor, *adult*, can probably be recognized by looking only at the lower human body, e.g.

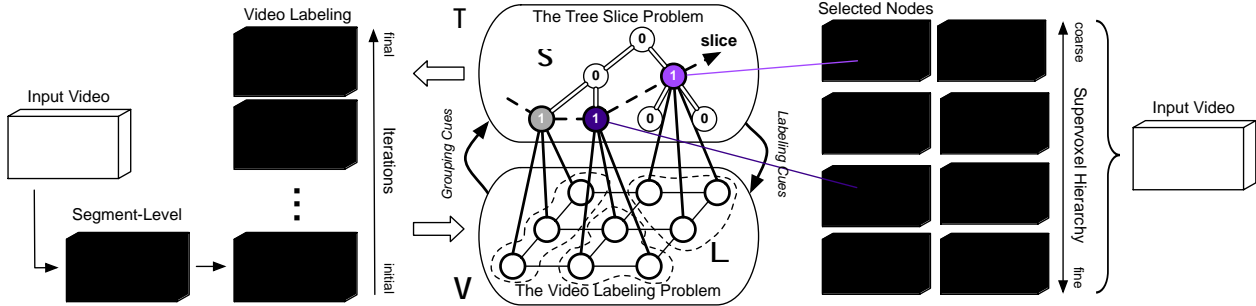


Figure 1. An overview of the grouping process model. The left side shows an input video and its segment-level segmentation. The right side shows the same video being segmented into a supervoxel hierarchy. During inference, the CRF defined on the segment-level starts with a coarse video labeling. It influences what supervoxels are active in the hierarchy. The active supervoxels, in turn, affect the connectivities in the CRF. This process is dynamic and continuous, where the video labeling is being iteratively refined.

legs. However, in order to recognize the *clapping* action, we have to either locate the acting parts of the human body or simply look at the whole actor body.

Third, actors and actions have different emphases on space and time in a video. Actors are more space-oriented—they can be fairly well labeled using only still images, as in semantic image segmentation [32, 43], whereas actions are space- and time-oriented. Although one can possibly identify actions by still images alone [42], there are strong distinctions between actions in time. For example *running* is faster and thus may result more repeated motion patterns than *walking* for a common time duration; and *walking* performed by a *baby* is very different compared to an *adult*, despite the two actor classes may easily confuse a spatially trained detector.

Our method overcomes the above limitations in two ways: (1) we propose a novel grouping process model (GPM) that adaptively adds long-ranging interactions to the labeling CRF; and (2) we incorporate the video-level recognition into segment-level labeling by the means of global labeling cost and the GPM. The GPM models a dynamic and continuous process of information exchange of a labeling CRF and a supervoxel hierarchy. The supervoxel hierarchy provides a rich multi-scale decomposition of video content, where object parts, identities, deformations and actions are retained in space-time supervoxels across various levels in the hierarchy [11, 27, 41]. Rather than using object and action proposals as separate processes, we directly locate the actor and action groupings in the supervoxel hierarchy by the labeling CRF. During inference, the labeling CRF influences what supervoxels in a hierarchy are *active*, and these active supervoxels, in turn, influence the connectivities in the CRF, thus refining the labeling.

Directly solving the joint energy function of GPM is hard. However, it can be efficiently solved by decomposing it into two subproblems, a video labeling problem and a tree slice problem [41], where the former one can be solved by graph cuts and the latter one can be rewritten into a bi-

nary linear program. Therefore, the inference of GPM is dynamic and iterative as shown in Fig. 1. Throughout the entire process, information is being exchanged at various levels in the supervoxel hierarchy, thus the multi-scale space-time representation is explicitly explored in our model.

We conduct thorough experiments on the large-scale actor-action video dataset (A2D) [40]. We compare the proposed method to the previous benchmarked leading method, the trilinear model, as well as two leading semantic segmentation methods [14, 16] that we have extended to the actor-action problem. The experimental results show that our proposed method outperforms the second best method by a large margin of 17% per-class accuracy (60% relative improvement) and over 10% global pixel accuracy, which demonstrates the effectiveness of our modeling.

## 2. Related Work

The actor-action semantic segmentation problem is first proposed in [40], where the paper demonstrates that inference jointly over actors and actions outperforms inference independently over them. The best performance in [40] is due to the trilinear model; although it does consider the interplay of actor and action variables, it only models interactions in local CRF pairwise neighborhoods. In contrary, the method in this paper considers the interplays at various granularities in space and time introduced by a supervoxel hierarchy.

Supervoxels are shown to capture object boundaries and follow object motions [39], and have the ability to locate objects and actions [11, 27]. They have been used as higher-order potentials for human action segmentation [22] and video object segmentation [12]. Here, we use supervoxel hierarchies for video labeling of actors and actions. We use the tree slice constraint to select supervoxels in a hierarchy as in [41], but the difference is that the tree slices here are drawn in an iterative fashion, where each time the slice also modifies the underlying labeling graph.

Our work also differs from the emerging works in action localization, action detection, and video object segmentation for two reasons. First, our segmentation contains clear semantic meanings of actors and actions, whereas most existing works in action localization and detection do not [11, 18, 24, 36]. Second, we consider multiple actors performing actions in a video and explicitly model the types of actors, whereas existing works assume one human actor [10, 23, 34, 44, 45] or do not model the types of actors at all [7, 20, 46, 47]. Although there have been some works on action detection [34], this remains an open challenge.

We relate our work to AHRF [16] and FCRF [14] in Section 4 after presenting the new model.

### 3. Grouping Process Model

In this section, we give the general form of GPM, and Fig. 1 shows an overview. We define the detailed potentials adapted to the actor-action problem in Sec. 5.

**Segment-Level.** Without loss of generality, we define  $V = \{q_1, q_2, \dots, q_N\}$  as a video with  $N$  voxels or a video segmentation with  $N$  segments. A graph  $G = (V, E)$  is defined over the entire video, where the neighborhood structure  $E(\cdot)$  is induced by the connectivities in the voxel lattice<sup>3</sup> or the segmentation graph over space-time in a video. We define a set of random variables  $L = \{l_1, l_2, \dots, l_N\}$  where the subscript corresponds to a certain node in  $V$  and each  $l_i$  takes some label from a label set  $\mathcal{L}$ . The GPM is inherently a labeling CRF, but it leverages a supervoxel hierarchy to dynamically adjust its non-local grouping structure.

**Supervoxel Hierarchy.** Given a supervoxel hierarchy generated by a hierarchical video segmentation method, such as GBH [9], we extract a supervoxel tree<sup>1</sup>, denoted as  $T = \{T_1, T_2, \dots, T_S\}$  with  $S$  total supervoxels in the tree, by ensuring that each supervoxel at a finer level segmentation has one and only one parent at its coarser level (Sec. 6 details the tree extraction process in the general case). We define a set of random variables  $s = \{s_1, s_2, \dots, s_S\}$  on the tree supervoxels, where  $s_t \in \{0, 1\}$  takes a binary label to indicate whether the  $t$ th supervoxel is active or not. Each supervoxel in the hierarchy connects to a set of nodes in the segment-level according to their overlap in voxel lattice<sup>3</sup>. Thus we have  $s_t$ , which is connected to a set of random variables at the segment-level CRF, denoted as  $L_t \subseteq \mathcal{L}$ . Intuitively, when  $s_t$  is active, the fully-connected clique containing all nodes in  $L_t$  is considered in the labeling CRF; otherwise, when  $s_t$  is inactive, that fully-connected clique is not evaluated.

Supervoxel hierarchies, such as [5, 9], are built by iteratively recomputing and merging finer supervoxels into coarser ones based on appearance and motion features,

where the body parts of an actor and its local motion are contained at the finer levels and the identity of the actor and its long-ranging action are contained at the coarser levels. However, choosing an arbitrary level in a hierarchy can be risky—going too coarse will cause overmerging and going too fine will lose the meaningful actions. It is challenging to locate the supervoxels in a hierarchy that best describe the actor and its action. Here, the GPM uses the evidence directly from the segment-level CRF to locate supervoxels across various scales that are best supported by the labeling  $\mathcal{L}$ . Once the supervoxels  $s$  are selected, they provide strong labeling cues to the segment-level CRF—the CRF nodes connected to the same supervoxel are encouraged to have the same label.

The objective of GPM is to find the best labeling  $L$  and the best selection  $s$  that minimize the following energy:

$$(L, s) = \arg \min_{L, s} E(L, s|V, T)$$

$$E(L, s|V, T) = E^v(L|V) + E^h(s|T) + \sum_{t \in T} (E^h(L_t|s_t) + E^h(s_t|L_t)), \quad (1)$$

where  $E^v(L|V)$  and  $E^h(s|T)$  encode the energies at the segment-level and in the supervoxel hierarchy, respectively;  $E^h(L_t|s_t)$  and  $E^h(s_t|L_t)$  are conditional energy functions defined as directional edges in Fig. 1. To keep the discussion general, we do not define the specific form of  $E^v(L|V)$  here—it can be any labeling CRF, such as [14, 16, 31]. We define the other terms next.

#### 3.1. Labeling Cues from Supervoxel Hierarchy

Given an active node  $s_t$  in the supervoxel hierarchy, we use it as a cue to refine the segment-level labeling  $L_t$  and we define the energy of this process as:

$$E^h(L_t|s_t) = \begin{cases} \prod_{i \in L_t} \prod_{j \in L_t} h_{ij}(l_i, l_j) & \text{if } s_t = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Here,  $h_{ij}(\cdot)$  has the form:

$$h_{ij}(l_i, l_j) = \begin{cases} \tau & \text{if } l_i = l_j \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where  $\tau$  is a parameter to be tuned.  $h_{ij}(l_i, l_j)$  penalizes any two nodes in the field  $L_t$  that contain different labels. Eq. 2 changes the graph structure in  $L_t$  by fully connecting the nodes inside, and has clear semantic meaning—this set of nodes in  $L_t$  at the segment-level are linked to the same supervoxel node  $s_t$  and hence expected to be from the same object, taking evidences from the appearance and motion features used in a typical supervoxel segmentation method.

<sup>1</sup>We add one virtual node as root to make it a tree if the segmentation at the coarsest level contains more than one supervoxel.

### 3.2. Grouping Cues from Segment Labeling

If the selected supervoxels are too fine, they are subject to losing object identity and long-ranging actions; if they are too coarse, they are subject to overmerging with the background. Therefore, we set the selected supervoxels to best reflect the segment-level labeling while also respecting a selection prior. Given a video labeling  $L$  at the segment-level, we select the nodes in the supervoxel hierarchy that best correspond to the current labeling:

$$E^h(s_t|L_t) = (H(L_t)|L_t| + \eta)s_t, \quad (4)$$

where  $|\cdot|$  denotes the number of video voxels and  $\eta$  is a parameter to be tuned that encodes a prior of the node selection in the hierarchy.  $H(\cdot)$  is defined as the entropy of the labeling field connected to  $s_t$ :  $H(L_t) = -\sum_{l \in L_t} P(l; L_t) \log P(l; L_t)$ , where  $P(l; L_t) = \frac{\sum_{i \in L_t} \mathbb{1}(l_i = l)}{|L_t|}$  and  $\mathbb{1}(\cdot)$  is an indicator function. Intuitively, the first term in Eq. 4 pushes down the selection of nodes in the hierarchy such that they only include the labeling field that has the most consistent labels, and the second term pulls up the node selection, giving penalties for going down the hierarchy.

### 3.3. Tree Slice Constraint

The active nodes in  $s$  define what groups of segments the GPM will enforce during labeling; hence the name grouping process model. However, not all instances of  $s$  are permissible: since we seek a single labeling over the video, we enforce that each segment in  $V$  is associated with one and only one active node in  $s$ . This notion was introduced in [41] by a way of *tree slice*: for every root-to-leaf path in  $T$ , there is one and only one node being active.

We follow [41] to define a matrix  $P$  that encodes all root-to-leaf paths in  $T$ .  $P_p$  is one row in  $P$ , and it encodes the path from the root to pth leaf with 1s for nodes on the path and 0s otherwise. We define the energy to regulate  $s$  as:

$$E^h(s|T) = \sum_{p=1}^P (P_p^T s = 1) \quad (5)$$

where  $P$  is the total number of leaves (also the number of such root-to-leaf paths) and  $\lambda$  is a large constant to penalize an invalid tree slice. The tree slice selects supervoxel nodes to form a new video representation that has a one-to-one mapping to the 3D video lattice <sup>3</sup>.

## 4. Iterative Inference for GPM

Directly solving the objective function defined in Eq. 1 is hard. Here, we show that we can use an iterative inference schema to efficiently solve it—given the segment-level labeling, we find the best supervoxels in the hierarchy; and

given the selected supervoxels in the hierarchy, we refine the segment-level labeling.

**The Video Labeling Problem.** Given a tree slice  $s$ , we would like to find the best  $L$  such that:

$$\begin{aligned} L &= \arg \min_L E(L|s, V, T) \\ &= \arg \min_L E^v(L|V) + \sum_{t \in T} E^h(L_t|s_t). \end{aligned} \quad (6)$$

The above can have a standard CRF form depending on how  $E^v(L|V)$  is defined. The second energy term  $E^h(L_t|s_t)$  can be decomposed to a locally fully connected CRF, and its range is constrained by  $s_t$  such that the inference is feasible even without Gaussian kernels [14].

**The Tree Slice Problem.** Given the current labeling  $L$ , we would like to find the best  $s$  such that:

$$\begin{aligned} s &= \arg \min_s E(s|L, V, T) \\ &= \arg \min_s E^h(s|T) + \sum_{t \in T} E^h(s_t|L_t). \end{aligned} \quad (7)$$

The above equation can be rewritten as a binary linear program of the following form:

$$\min_{t \in T} \sum_t s_t \quad \text{s.t. } P s = \mathbf{1}_P \quad \text{and } s \in \{0, 1\}^S, \quad (8)$$

where  $\lambda = H(L_t)|L_t| + \eta$ . Note that this optimization is different than that proposed by the original tree slice paper [41], which incorporated quadratic terms in a binary quadratic program. We use a standard solver (IBM CPLEX) to solve the binary linear programming problem.

**Iterative Inference.** The inference of the above two sub-problems is iteratively carried out, as depicted in Fig. 1. To be specific, we initialize a coarse labeling  $L$  by solving Eq. 6 without the second term, then we solve Eq. 8 and 6 in an iterative fashion. Each round of the tree slice problem enacts an updated set of grouped segments, which are then encouraged to be assigned the same label during the subsequent labeling process. Although we do not include a proof of convergence in this paper, we notice that the solution converges after a few rounds.

**Relation to AHRF.** The associative hierarchical random field (AHRF) [16] performs inference exhaustively from finer levels to coarser levels in the segmentation tree  $T$ , whereas the GPM explicitly models the best set of active supervoxels by the means of a tree slice. AHRF defines a full multi-label random field on the hierarchy; our model leverages the hierarchy to adaptively modify the labeling field. Our model is hence more scalable to videos. Furthermore, the GPM assumes that the best representations of the video content exist in a tree slice rather than enforcing the agreement across different levels as in AHRF. For example,



a video of *long jumping* often contains *running* in the beginning. The running action exists and has a strong classifier signal at a fine-level in a supervoxel hierarchy, but it quickly diminishes when one goes to higher levels in the hierarchy where supervoxels capture longer range in the video and would then favor the *jumping* action.

**Relation to FCRF.** The fully-connected CRF (FCRF) in [14] imposes Gaussian mixture kernels to regularize the pairwise interactions. Although our model fully connects the nodes in each  $L_t$  for a given iteration of inference, we explicitly take the evidence from the supervoxel groupings. Equation 4 restricts the selected supervoxels to avoid over-merging. Although a more complex process, in practice, our inference is efficient (see Sec. 6 for running time).

## 5. The Actor-Action Problem Modeling

Following semantic segmentation systems [35, 40], we train segment-level classifiers to capture the local appearance and motion of the actors' body parts. They have some ability to localize the actor-action, but the predictions are noisy; they use no context, for example. In contrast, video-level recognition, as a secondary process, captures the global information of actors performing actions and have good prediction performance at the video-level. However, it is not able to tell where the action is happening. These two streams of information are captured at the segment-level and at the video-level, and hence are complementary to each other. In this section, we fuse them together in a single model, leveraging the grouping process model as a means of marrying the two.

Let us first define notation, extending that from Sec. 3 where possible. We use  $X$  to denote the set of actor labels (e.g. *adult*, *baby* and *dog*) and  $Y$  to denote the set of action labels (e.g. *eating*, *walking* and *running*). The segment-level random field  $L$  now takes two sets of labels—for the  $i$ th segment,  $l_i^X \in X$  takes a label from the actors and  $l_i^Y \in Y$  from the actions. We denote  $Z = X \times Y$  as the joint product space of the actor-action labels. We define a set of binary random variables  $v = \{v_1, v_2, \dots, v_{|Z|}\}$  on the video-level, where  $v_z = 1$  denotes the  $z$ th actor-action label is active at the video-level. They represent the video-level multi-label recognition problem. Again, we have the set of binary random variables  $s$  defined on the supervoxel hierarchy as in Sec. 3.

Therefore, we have the total energy function of the actor-action semantic segmentation defined as:

$$\begin{aligned} (L, s, v) &= \arg \min_{L, s, v} E(L, s, v | V, T) \\ E(L, s, v | V, T) &= E^V(L | V) + \sum_{z \in Z} E^V(v_z | V) + E^V(L, v) \\ &+ E^h(s | T) + \sum_{t \in T} (E^h(L_t, v | s_t) + E^h(s_t | L_t)) , \quad (9) \end{aligned}$$

where the term  $E^h(L_t, v | s_t)$  now models the joint potentials of the segment-level labeling field  $L_t$  and the video-level label  $v$ , which is slightly different from its form in Eq. 2. We have two new terms,  $E^V(v_z | V)$  and  $E^V(L, v)$ , from the video-level, where  $v_z$  is the  $z$ th coordinate in  $v$ . We explain these new terms next.

### 5.1. Segment-Level CRF $E^V$

At the segment-level, we use the same bilayer actor-action CRF model from [40] to capture the local pairwise interactions of the two sets of labels:

$$\begin{aligned} E^V(L | V) &= \sum_{i \in V} \psi_i(l_i^X) + \sum_{i \in V} \sum_{j \in E(i)} \psi_{ij}(l_i^X, l_j^X) \quad (10) \\ &+ \sum_{i \in V} \psi_i(l_i^Y) + \sum_{i \in V} \sum_{j \in E(i)} \psi_{ij}(l_i^Y, l_j^Y) + \sum_{i \in V} \psi_i(l_i^X, l_i^Y) , \end{aligned}$$

where  $\psi_i$  and  $\psi_{ij}$  encode separate potentials for random variables  $l_i^X$  and  $l_i^Y$  to take the actor and action labels, respectively.  $\psi_i$  is a potential to measure the compatibility of the actor-action tuples on segment  $i$ , and  $\psi_{ij}$  and  $\psi_{ij}$  capture the pairwise interactions between segments, which have the form of a contrast sensitive Potts model [3, 31]. We use the code from [40] to capture the local pairwise interactions of the two sets of labels.

### 5.2. Video-Level Potentials $E^V$

Rather than a uniform penalty over all labels [6], we use the video-level recognition signals as global multi-label labeling costs to impact the segment-level labeling. We define the unary energy at the video-level as:

$$E^V(v_z | V) = -(\psi^V(z) - \tau) \beta v_z , \quad (11)$$

where  $\psi^V(\cdot)$  is the video-level classification response for a particular actor-action label, and Sec. 6 describes its training process. Here,  $\tau$  is a parameter to control response threshold, and  $\beta$  is a large constant parameter. In other words, to minimize Eq. 11, the label  $v_z = 1$  only when the classifier response  $\psi^V(z) > \tau$ .

We define the interactions between the video-level and the segment-level:

$$E^V(L, v) = \sum_{x \in X} x(L) h_x(v) + \sum_{y \in Y} y(L) h_y(v) , \quad (12)$$

where  $x(\cdot)$  is an indicator function to determine whether the current labeling  $L$  at the segment-level contains a particular label  $x \in X$  or not:

$$x(L) = \begin{cases} 1 & \text{if } \exists i \in V : l_i^X = x \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

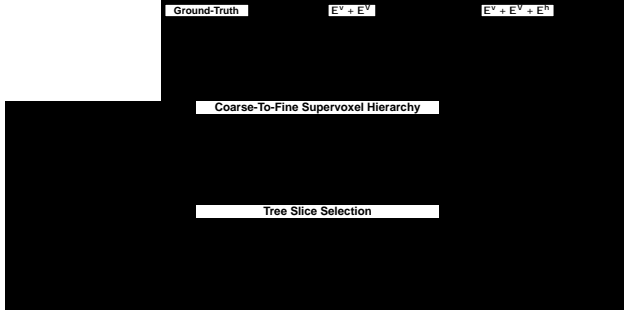


Figure 2. The video labeling of actor-action is refined by GPM. First row shows a test video *car-jumping* with its labelings. The second row shows a supervoxel hierarchy and the third row shows the active nodes in the hierarchy with their dominant labels.

Similarly,  $h_x(\cdot)$  is another indicator function to determine whether a particular label  $x$  is supported at the video-level or not:

$$h_x(v) = \begin{cases} 0 & \text{if } z \in Z : v_z = 1 \quad g(z) = x \\ 1 & \text{otherwise,} \end{cases} \quad (14)$$

where  $g(\cdot)$  maps a label in the joint actor-action space to the actor space.  $v$  is a constant cost for any label that exists in  $L$  but not supported at the video-level. We define  $h_y(\cdot)$  and  $h_z(\cdot)$  similarly. To make the cost meaningful, we set  $v_B > 2v$ . In practice, we observe that these labeling costs from video-level recognition help the segment-level labeling to achieve a more parsimonious-in-labels result that enforces more global information than using local segments alone (see results in Table 1).

### 5.3. The GPM Potentials $E^h$

The energy terms  $E^h(s_t|T)$  and  $E^h(s_t|L_t)$  involved in the tree slice problem are defined the same as in Sec. 3. Now, we define the new labeling term:

$$E^h(L_t, v|s_t) = \begin{cases} \sum_{i \in L_t} \sum_{j \in L_t} h_{ij}(l_i^X, l_j^X, v) + \sum_{i \in L_t} \sum_{j \in L_t} h_{ij}(l_i^Y, l_j^Y, v) & \text{if } s_t = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

Here,  $h_{ij}(\cdot)$  has the form:

$$h_{ij}(l_i^X, l_j^X, v) = \begin{cases} v & \text{if } l_i^X = l_j^X, \quad z \in Z : v_z = 1 \quad g(z) = f(s_t) \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where  $f(\cdot)$  denotes the dominant actor label in the segment-level labeling field  $L_t$  that connected to  $s_t$ , and we define  $h_{ij}(l_i^Y, l_j^Y, v)$  similarly. This new term selectively refines the segmentation where the majority of the segment-level labelings agree with the video-level multi-label labeling.

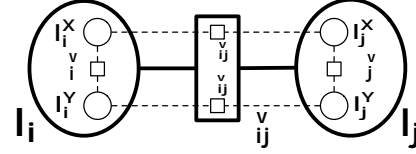


Figure 3. Visualization of two nodes of the bilayer model in our efficient inference.

We show in Fig. 2 how this GPM process helps to refine the actor's shape (the car) in the labeling process. The initial labelings from  $E^V + E^V$  propose a rough region of interest, but they do not capture the accurate boundaries or shape. After two iterations of inference, the tree slice selects the best set of supervoxels in the GBH hierarchy that represents the actor (the car), and they regroup the segment-level labelings such that the labelings can better capture the actor shape. Notice that the car body in the third column merges with the background, but our full model (fourth column) overcomes the limitation by selecting different parts from the hierarchy to yield the final labeling.

### 5.4. Inference

The inference of the actor-action problem defined in Eq. 9 follows the iterative inference described in Sec. 4. The tree slice problem is efficiently solved by binary linear programming. Although we could solve the video labeling problem with loopy belief propagation, it would be expensive due to the two sets of labels over which the CRF is defined. Here, we derive a way to solve it efficiently using graph cuts inference with label costs [1, 2, 6]. We show this conceptually in Fig. 3 and rewrite Eq. 10 as:

$$E^V(L|V) = \sum_{i \in V} v_i(l_i) + \sum_{i \in V} \sum_{j \in E(i)} v_{ij}(l_i, l_j), \quad (17)$$

where we define the new unary as:

$$v_i(l_i) = v_i(l_i^X) + v_i(l_i^Y) + v_i(l_i^X, l_i^Y), \quad (18)$$

and the pairwise interactions as:

$$v_{ij}(l_i, l_j) = \begin{cases} v_{ij}(l_i^X, l_j^X) & \text{if } l_i^X = l_j^X \quad l_i^Y = l_j^Y \\ v_{ij}(l_i^Y, l_j^Y) & \text{if } l_i^X = l_j^X \quad l_i^Y = l_j^Y \\ v_{ij}(l_i^X, l_j^X) + v_{ij}(l_i^Y, l_j^Y) & \text{if } l_i^X = l_j^X \quad l_i^Y = l_j^Y \\ 0 & \text{if } l_i^X = l_j^X \quad l_i^Y = l_j^Y. \end{cases} \quad (19)$$

We can rewrite Eq. 15 in a similar way, and they satisfy the submodular property according to the triangle inequality [13]. The label costs can be solved as in [6].

**Parameters.** We manually explore the parameter space based on the pixel-level accuracy in a heuristic fashion. We first tune the parameters involved in the video-level recognition, then those involved in the segment-level labeling, and finally, those involved in GPM by running the iterative inference as in Sec. 4.

## 6. Experiments

We evaluate our method on the recently released A2D dataset [40] and use the same benchmark to evaluate the performance; this is the only dataset we are aware of that incorporates actors and actions together. We compare with the top-performing trilayer model, and two strong semantic image segmentation methods, AHRF [16] and FCRF [14]. For AHRF, we use the publicly available code from [16] as it contains a complete pipeline from training classifiers to learning and inference. For FCRF, we extend it to use the same features as our method.

**Data Processing.** We experiment with two distinct supervoxel trees: one is extracted from the hierarchical supervoxel segmentations generated by GBH [9], where supervoxels across multiple levels natively form a tree structure hierarchy, and the other one is extracted from multiple runs of a generic non-hierarchical supervoxel segmentation by TSP [4]. To extract a tree structure from the non-hierarchical video segmentations, we first sort the segmentations by the number of supervoxels they contain. Then we enforce the supervoxels in the finer level segmentation to have one and only one parent supervoxel in the coarser level segmentation, such that the two supervoxels have the maximal overlap of the video pixels. We use four levels from a GBH hierarchy, where the number of supervoxels varies from a few hundred to less than one hundred. We also use four different runs of TSP to construct another segmentation tree where the final number of nodes contained in the tree varies from 500 to 1500 at the fine level, and from 50 to 150 at the coarse level.

We also use TSP to generate the segments for the base labeling CRF. We extract the same set of appearance and motion features as in [40] and train one-versus-all linear SVM classifiers on the segments for three sets of labels: actor, action, and actor-action pair, separately. At the video-level, we extract improved dense trajectories [38], and use Fisher vectors [29] to train linear SVM classifiers at the video-level for the actor-action pair. We use the inference schema described in Sec. 4 and Sec. 5.4, and follow the train/test splits used in [40]. The output of our system is a full video pixel labeling. We evaluate the performance on sampled frames where the ground-truth is labeled.

**Results and Comparisons.** We follow the benchmark evaluation in [40] and evaluate performance for joint actor-action and separate individual tasks. Table 1 shows the overall results of all methods in three different calculations: when all test videos are used; when only videos containing single-label actor-action are used; and when only videos containing multiple actor-action labels are used. Roughly one-third of the videos in the A2D dataset have multiple actor-action labels. Overall, we observe that our methods (both GPM-TSP and GPM-GBH) outperform the next best one, the trilayer method, by a large margin of 17% average

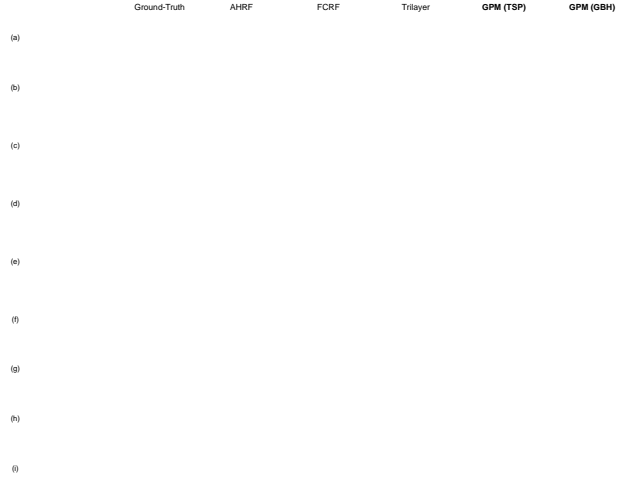


Figure 4. Visual example of the actor-action video labelings for all methods. (a) - (c) are videos where most methods get correct labelings; (d) - (g) are videos where only GPM models get the correct labelings; (h) - (l) are difficult videos in the dataset where the GPM models get partially correct labelings. Colors used are from the A2D benchmark.

per-class accuracy and more than 10% global pixel accuracy over all test videos. The improvement of global pixel accuracy is consistent over the two sub-divisions of test videos, and the improvement of average per-class accuracy is larger on videos that only contain single-label actor-action. We suspect that videos containing multiple-label actor-action are more likely to confuse the video-level classifiers.

We also observe that the added grouping process in GPM-TSP and GPM-GBH consistently improves the average per-class accuracy over the intermediate result ( $E^V + E^V$ ) on both single-label and multiple-label actor-action videos. There is a slight decrease on the global pixel accuracy. We suspect the decrease mainly comes from the background class, which contributes a large portion of the total pixels in evaluation. To verify that, we also show the individual actor-action class performance in Tab. 2 when all test videos are used. We observe that GPM-GBH has the best performance on majority classes and improves  $E^V + E^V$  on all classes except *dog-crawling*, which further shows the effectiveness of the grouping process. The performance of our method using the GBH hierarchy is slightly better than our method using the TSP hierarchy. We suspect that this is due to the GBH method's greedy merging process that complements the Gaussian process in TSP, such that the resulting segmentation complements the segment-level TSP segmentation we used.

Figure 4 shows the visual comparison of video labelings for all methods, where (a)-(c) show cases where methods output correct labels and (d)-(g) show cases where our proposed method outperforms other methods. We also show

Model	All Test Videos						Single Actor-Action Videos						Multiple Actor-Action Videos					
	Actor		Action		<A, A>		Actor		Action		<A, A>		Actor		Action		<A, A>	
	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.	Ave.	Glo.
E <sup>v</sup>	45.9	76.9	47.2	76.8	24.8	75.0	46.7	76.5	50.0	76.9	31.5	74.8	41.7	77.7	42.1	76.5	18.2	75.4
E <sup>v</sup> + E <sup>v</sup>	57.3	85.7	59.4	85.9	42.4	84.8	60.4	86.0	67.0	86.5	55.4	85.4	50.6	85.1	55.1	84.4	33.3	83.6
AHRF	38.0	64.9	29.0	63.9	13.9	63.0	38.1	66.6	29.7	65.8	16.6	64.8	37.0	60.6	28.3	59.3	11.3	58.5
FCRF	44.8	77.9	45.5	77.6	25.4	76.2	45.9	77.6	47.4	77.7	32.1	76.1	40.2	78.8	42.2	77.5	19.4	76.5
Trilayer	45.7	74.6	47.0	74.6	26.5	72.9	47.0	74.1	50.3	74.6	33.9	72.7	41.0	75.6	42.3	74.5	20.4	73.4
GPM (TSP)	58.3	85.2	60.5	85.3	43.3	84.2	61.5	85.4	68.2	86.0	56.5	84.8	51.7	84.5	56.2	83.8	33.9	83.0
GPM (GBH)	61.2	84.9	59.4	84.8	43.9	83.8	63.1	85.1	69.3	85.7	57.6	84.5	51.7	84.1	56.3	83.3	33.9	82.5

Table 1. The overall performance on the A2D dataset. The top two rows are intermediate results of the full model. The middle three rows are comparison methods. The bottom two rows are our full models with different supervoxel hierarchies for the grouping process.

Model	BK	adult								cat								dog							
		climb	crawl	eat	jump	roll	run	walk	none	climb	eat	jump	roll	run	walk	none		crawl	eat	jump	roll	run	walk	none	
E <sup>v</sup>	81.0	22.1	60.4	45.2	20.0	18.9	32.3	26.8	31.5	25.3	29.8	4.4	29.5	45.2	6.5	0.0		17.0	26.6	1.1	38.1	29.8	38.7	0.0	
E <sup>v</sup> + E <sup>v</sup>	<b>89.9</b>	73.3	77.6	68.0	47.1	49.4	49.8	39.8	0.0	41.9	48.0	31.0	69.8	48.0	18.7	0.0		<b>45.8</b>	58.9	30.7	61.4	25.1	72.4	0.0	
AHRF	69.2	0.0	56.0	6.1	1.1	0.0	0.0	15.3	10.9	18.3	38.8	0.0	8.8	0.0	9.3	0.0		13.2	16.4	0.0	0.0	0.0	0.0	0.0	
FCRF	82.2	21.6	64.5	46.3	25.3	12.0	<b>50.9</b>	26.9	<b>33.8</b>	25.3	33.6	2.5	33.9	<b>48.9</b>	<b>21.5</b>	<b>0.8</b>		11.7	35.7	2.2	31.9	25.2	40.2	0.0	
Trilayer	78.5	33.1	59.8	49.8	19.9	27.6	40.2	31.7	24.6	33.1	27.2	6.1	49.8	48.5	6.6	0.0		9.9	31.0	2.0	27.6	23.6	39.4	0.0	
GPM (TSP)	89.1	74.6	79.8	70.7	<b>49.3</b>	51.5	50.6	40.4	0.0	42.5	49.3	31.9	71.1	46.4	18.8	0.0		45.3	60.2	31.3	62.5	25.8	74.0	0.0	
GPM (GBH)	88.4	<b>74.8</b>	<b>81.0</b>	<b>76.4</b>	<b>49.3</b>	<b>52.4</b>	50.4	<b>41.0</b>	0.0	<b>42.8</b>	<b>52.3</b>	<b>33.7</b>	<b>71.7</b>	48.0	19.1	0.0		44.1	<b>61.5</b>	<b>31.4</b>	<b>62.6</b>	25.7	<b>74.2</b>	0.0	

Model	baby				ball				bird				car				Ave.	Glo.
	climb	crawl	roll	walk	none	fly	jump	roll	none	climb	eat	fly	jump	roll	walk	none		
E <sup>v</sup>	13.8	32.8	38.3	20.0	0.0	3.8	10.4	4.5	0.0	28.1	14.1	51.6	18.2	33.1	7.2	0.0	25.7	78.0
E <sup>v</sup> + E <sup>v</sup>	63.6	64.0	55.4	60.6	0.0	<b>11.3</b>	26.7	20.5	0.0	58.7	35.4	65.8	17.2	44.2	41.1	0.0	40.8	83.4
AHRF	21.3	5.5	39.8	13.5	0.0	3.2	2.3	13.6	<b>1.5</b>	14.6	11.4	19.9	5.0	29.6	7.5	0.0	18.1	68.0
FCRF	3.4	23.4	41.0	17.8	0.0	3.7	0.3	1.0	0.0	25.9	16.1	57.3	17.1	35.0	7.4	0.0	13.7	78.4
Trilayer	20.4	21.7	39.3	25.3	0.0	1.0	11.9	6.1	0.0	28.1	18.2	55.3	<b>20.3</b>	42.5	9.0	0.0	24.4	75.9
GPM (TSP)	65.3	64.7	57.2	60.5	0.0	<b>11.3</b>	27.0	20.8	0.0	<b>62.2</b>	37.1	<b>66.6</b>	17.4	45.4	42.2	0.0	<b>42.9</b>	84.5
GPM (GBH)	<b>65.4</b>	<b>65.0</b>	<b>58.4</b>	<b>61.5</b>	0.0	<b>11.3</b>	<b>28.3</b>	<b>21.1</b>	0.0	60.6	<b>38.8</b>	66.5	17.5	<b>45.9</b>	<b>47.9</b>	0.0	41.2	<b>86.3</b>

Table 2. The performance on individual actor-action labels using all test videos. The leading scores for each label are in bold font.

failure cases in (h) and (i) where videos contain complex actors and actions. For example, our method correctly labels the *ball-rolling* but confuses the label *adult-running* as *adult-walking* in (h); we correctly label *adult-crawling* but miss the label *adult-none* in (i).

**Inference Speed.** We empirically set the stopping criteria by observing a balance between the performance gain and the running time. We set two iterations for all experiments. For all the test videos, GPM-GBH has an average inference speed of 8.6 seconds-per-video (spv) faster than 26.7 spv of GPM-TSP. Both of them are faster than 142 spv of the trilayer model in [40]. The experiments are conducted with a Linux server with AMD Opteron 6380 2.5GHz CPU.

## 7. Conclusion

Our thorough experiments on the A2D dataset show that when the segment-level labeling is combined with secondary processes, such as our grouping process models and video-level recognition signals, the semantic segmentation performance increases dramatically. For example, GPM-GBH improves almost every class of actor-action labels compared to the intermediate result without the supervoxel hierarchy, i.e., without the dynamic grouping of CRF labeling variables. This finding strongly supports our motivating argument that the two sets of labels, actors and actions, are best modeled at different levels of granularities and that they have different emphases on space and time in a video.

In summary, our paper makes the following contributions to the actor-action semantic segmentation problem:

1. A novel model that dynamically combines segment-level labeling with a hierarchical grouping process that influences connectivities of the labeling variables.
2. An efficient inference method that iteratively solves the two conditional tasks by graph cuts for labeling and binary linear programming for grouping allowing for continuous exchange of information.
3. A new framework that uses video-level recognition signals as cues for segment-level labeling thru global labeling costs and the grouping process model.
4. Our proposed method significantly improves performance (60% relative improvement over the next best method) on the recently released large-scale actor-action semantic video dataset [40].

Our implementations as well as the extended baselines are available on authors' website.

**Future Work.** We set two directions for our future work. First, although our model is able to improve the labeling performance dramatically, the opportunity of this joint modeling to improve video-level recognition is yet to be explored. Second, our grouping process does not incorporate semantics in the supervoxel hierarchy; we believe this would further improve results.

**Acknowledgments** This work has been supported in part by Google, Samsung, DARPA W32P4Q-15-C-0070 and ARO W911NF-15-1-0354.



## References

- [1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. **6**
- [2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. **6**
- [3] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in nd images. In *IEEE International Conference on Computer Vision*, 2001. **5**
- [4] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. **7**
- [5] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille. Efficient multilevel brain tumor segmentation with integrated bayesian model classification. *Medical Imaging, IEEE Transactions on*, 27(5):629–640, 2008. **3**
- [6] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27, 2012. **5, 6**
- [7] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. **1, 3**
- [8] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007. **1**
- [9] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. **3, 7**
- [10] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009. **3**
- [11] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. **1, 2, 3**
- [12] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*, 2014. **2**
- [13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. **6**
- [14] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011. **2, 3, 4, 5, 7**
- [15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision*, 2011. **1**
- [16] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1056–1077, 2014. **1, 2, 3, 4, 7**
- [17] L. Ladický, P. Sturges, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *European Conference on Computer Vision*, 2010. **1**
- [18] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *IEEE International Conference on Computer Vision*, 2011. **3**
- [19] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005. **1**
- [20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *IEEE International Conference on Computer Vision*, 2011. **1, 3**
- [21] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *IEEE International Conference on Computer Vision*, 2013. **1**
- [22] J. Lu, R. Xu, and J. J. Corso. Human action segmentation with hierarchical supervoxel consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. **1, 2**
- [23] S. Ma, L. Sigal, and S. Sclaroff. Space-time tree ensemble for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. **3**
- [24] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff. Action recognition and localization by hierarchical space-time segments. In *IEEE International Conference on Computer Vision*, 2013. **3**
- [25] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. **1**
- [26] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014. **1**
- [27] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *European Conference on Computer Vision*, 2014. **2**
- [28] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *IEEE International Conference on Computer Vision*, 2013. **1**
- [29] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *European Conference on Computer Vision*, 2014. **7**
- [30] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *IEEE International Conference on Pattern Recognition*, 2004. **1**
- [31] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006. **1, 3, 5**
- [32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009. **2**
- [33] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CRCV-*

TR-12-01, 2012. 1

- [34] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 3
- [35] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, 2012. 5
- [36] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *Advances in Neural Information Processing Systems*, 2012. 3
- [37] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 2013. 1
- [38] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, 2013. 1, 7
- [39] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2
- [40] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? action understanding with multiple classes of actors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2, 5, 7, 8
- [41] C. Xu, S. Whitt, and J. J. Corso. Flattening supervoxel hierarchies by the uniform entropy slice. In *IEEE International Conference on Computer Vision*, 2013. 2, 4
- [42] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *IEEE International Conference on Computer Vision*, 2011. 2
- [43] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2
- [44] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3
- [45] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3
- [46] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *European Conference on Computer Vision*, 2014. 1, 3
- [47] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3