Multiple Granularity Analysis for Fine-grained Action Detection

Bingbing Ni
Advanced Digital Sciences Center
Singapore 138632
bingbing.ni@adsc.com.sg

Vignesh R. Paramathayalan
Advanced Digital Sciences Center
Singapore 138632
vignesh.r@adsc.com.sg

Pierre Moulin
UIUC
IL 61820-5711 USA
moulin@ifp.uiuc.edu

Abstract

We propose to decompose the fine-grained human activity analysis problem into two sequential tasks with increasing granularity. Firstly, we infer the coarse interaction status, i.e., which object is being manipulated and where it is. Knowing that the major challenge is frequent mutual occlusions during manipulation, we propose an "interaction tracking" framework in which hand/object position and interaction status are jointly tracked by explicitly modeling the contextual information between mutual occlusion and interaction status. Secondly, the inferred hand/object position and interaction status are utilized to provide 1) more compact feature pooling by effectively pruning large number of motion features from irrelevant spatio-temporal positions and 2) discriminative action detection by a granularity fusion strategy. Comprehensive experiments on two challenging fine-grained activity datasets (i.e., cooking action) show that the proposed framework achieves high accuracy/robustness in tracking multiple mutually occluded hands/objects during manipulation as well as significant performance improvement on fine-grained action detection over state-of-the-art methods.

1. Introduction

Understanding human activities in fine-grained detail has attracted increasing research interest during recent years [22, 15, 21]. Solution to this is of particular interest to computer assisted daily living application. The key to detect fine-grained actions, especially those with rich human object interactions, is to answer two sequential questions with increasing granularity: 1) which object is currently being manipulated (regarded as *interaction status*) and 2) which type of interaction is performed (*i.e.*, cutting a fruit or peeling a fruit). The advantage of this two-step coarse-to-fine visual understanding pipeline is that the output of the first step can significantly benefit the second step. On the one hand, as the spatio-temporal locations of candidate interaction actions are identified in the first step, one can easily search and prune the large spatio-temporal video volume and quickly identify a small set of sub volumes that contain the target action. On the other hand, there exists strong correlation between the action type and the type of object-in-use. For example, it is most likely to perform *cutting* with a knife in hand. Therefore, knowing what object is currently being operated gives us very important information on what action is being performed.

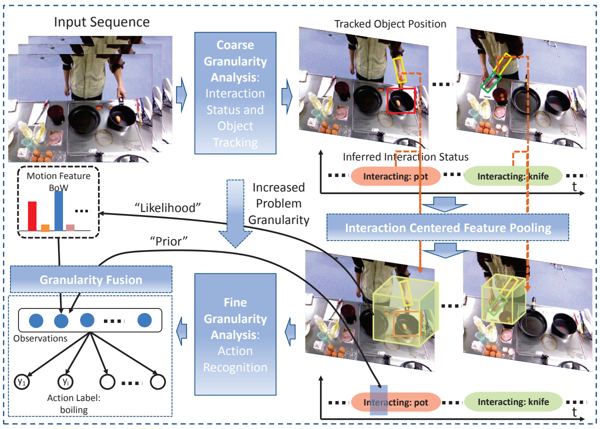
Input Sequence
Coarse Granularity Analysis: Interaction Status and Object Tracking
Tracked Object Position
Inferred Interaction Status
Interacting: pot
Interacting: knife
t
Motion Feature BoW
"Likelihood"
"Prior"
Increased Problem Granularity
Interaction Centered Feature Pooling
Granularity Fusion
Observations
y1
yi
Action Label: boiling
Fine Granularity Analysis: Action Recognition
Interacting: pot
Interacting: knife
t

Figure 1. Motivation of the proposed work. Fine-grained action detection is decomposed into two sequential tasks with increasing granularity: 1) interaction tracking; and 2) action detection. The tracked interaction status and object position (coarse granularity) can significantly aid action detection (fine granularity).

There exist comprehensive literature in human activity recognition, and the most promising methods are based on locally extracted spatio-temporal features [14, 26]. These methods can be divided into two major groups: 1) global methods compute histogram representation on densely extracted local features for action classification [14, 20, 26] and 2) local methods search the maximally confident video sub-volume for action detection based on sparsely extracted local features [24, 31]. These methods, however, cannot cope with the challenging fine-grained action detection. On the one hand, informative local motion features only exist at the time of interaction, therefore globally pooling local features results in a noisy histogram representation with a large portion of irrelevant features. On the other hand, sparsely located features convey insufficient information for representing an action, therefore without any prior information on the possible spatio-temporal sub volumes that contain the target action, it is infeasible to exhaustively search the whole video volume and to precisely detect the action subvolume. To reliably detect sparse local features in a realistic fine-grained action video with large content variations is therefore extremely difficult.

Motivated by the above observations, we propose to decompose the difficult fine-grained action detection problem into two sequential sub-tasks with increasing granularity, which are simpler and more tractable than directly performing action detection. The first sub-task is to recognize the coarse status of interaction (i.e., which object is currently being manipulated, and therefore the temporal duration of the interaction) and where is the occurrence of interaction (i.e., spatial locations of the objects and hands involved in the interaction). This sub-task requires jointly tracking multiple interacting objects (parts) and hands, which is very challenging. Previous methods [17, 5, 9, 32, 3, 29, 4, 2] on multiple object tracking cannot be simply applied since during object manipulation, very frequent mutual occlusions exist. Our key observation is that there exists rich contextual information between the interaction status and the occurrence of mutual occlusion. Namely, if we know the status of interaction (whether a certain object is being manipulated at the moment), we can predict the occurrence of mutual occlusion and take this information into account during tracking. For example, if we know the person is holding a knife, we can confidently predict that the knife handle is most likely to be occluded and the hand is close to the knife blade. Therefore, even if we cannot directly detect or track the handle of the knife, we can still know where it is since the geometric relationship between the hand and the blade implicitly encodes the position of the handle. To this end, we propose a probabilistic graphical model that utilizes the contextual information between interaction status and mutual occlusion to jointly track multiple interacting object parts/hands under frequent mutual occlusions. Our tracking framework is called **interaction tracking**.

Output of the first sub-task significantly benefits the second sub-task, which is to effectively represent fine-grained action and detect it. On the one hand, knowing the positions of the hands and objects-in-use as well as the interaction status (i.e., the start and the end time of the interaction) guides us when and where to extract informative local motion features and to effectively prune large number of irrelevant and noisy ones. We therefore propose an **interaction centered motion feature pooling** scheme, which represents action more compactly and discriminatively. On the other hand, we note that strong correlation exists between which kind of action is being performed (i.e., action label) and which object is being manipulated (i.e., interaction status). We therefore propose a **granularity fusion** approach which combines prior information given by the tracked interaction status and the pooled interaction centered motion feature bag-of-words representations into a spatial-temporal action graph. This graph encodes both the object-action co-occurrence probability and the action likelihood indicated by local motion features. Action detection is then performed by efficient inference on the constructed spatio-temporal action graph.

2. Related Work

Object-in-use contextual information has been commonly used for recognizing actions which involve human and object interaction [18, 28, 30, 12, 27]. However, these methods often represent object-in-use information in a global and coarse way, e.g., co-occurrence, which is ineffective for representing fine-grained action. Packer et al. [21] presented a system that is able to recognize complex, fine-grained human actions involving the manipulation of objects in cooking action sequences. Koppula et al. [13] proposed a framework that jointly detects human activities and object affordances. These works heavily rely on 3D skeleton tracker (i.e., Kinect); however, in real-world interactions, some body parts are often occluded and the 3D skeleton tracking easily fails. There exist many works on hand detection and tracking [7, 25, 16]. However, in this work, we are not interested in tracking hand alone. Instead, we focus on tracking the interaction between hand and object (part). To our best knowledge, the idea of explicitly modeling the contextual information between the interaction status and mutual occlusion for jointly tracking hand and object interaction has never been explored.

3. Coarse Granularity: Interaction Tracking

We first introduce the notations. Assume we are given a video sequence with T frames consisting of human and object interactions (*i.e.*, manipulations by hand such as baking an egg, cutting an apple, mixing using chopstick etc.), the task of tracking interaction is to jointly estimate at each frame: 1) which objects are being manipulated; and 2) where are the objects and hands involved in the interaction.

To this end, we assume that we need to track the positions of H objects of interest. We note that when operating an object, part of the object is hold by hand thus it will be always occluded, *e.g.*, handle of the pan. To explicitly utilize this prior knowledge, we partition an object into multiple parts, *e.g.*, a knife can be divided into two parts which are blade and handle, and a fry pan can also be divided into a handle and a main body. This decomposition also facilitates detection, since detecting a part is easier than detecting the whole object. We assume M object parts, i.e., $H < M$. For the ease of presenting our method, we use two sets of indices, namely object index and part index. Namely, an object is indexed by $h \in \{1, \cdots, H\}$ and an object part is indexed by $m \in \{1, \cdots, M\}$. There is a one-to-one mapping from part index to object index, and according to its parent object index, the M part indices can be divided into H groups. We denoted by $\pi(m) = h$ that part m belongs to object h, e.g., the pan handle is part of the fry pan. Note that some objects have only one part, e.g., bowl, hand, etc. In the meantime, we also divide M parts into two groups, with one group including the parts which will be occluded during hand interaction, denoted by $\mathcal{I}$, e.g., the pan handle. We index hand (part) as $m = 1$.

Our method falls in the *tracking by detection* category. In each frame, we apply individual object part detectors and generate multiple candidate detections by thresholding the detection confidence scores. For object part m at frame t, its candidate detections are indexed as $\{1, \cdots, d^t_m, \cdots, N^t_m\}$. Accordingly, we denote by $\mathbf{x}(d^t_m)$ the image coordinate of the detection d^t_m. We denote by $\boldsymbol{\phi}(d^t_m)$ the visual feature vector extracted from the detection d^t_m. For each frame t, we define a set of variables $\mathbf{p}^t = \{p^t_1, \cdots, p^t_M\}$, where each $p^t_m \in \{1, \cdots, N^t_m\}$ indicates which candidate detection is selected. We denote $\mathbf{P} = \{\mathbf{p}^1, \cdots, \mathbf{p}^T\}$. In the meantime, for each frame t, we introduce an interaction status variable v_t, where $v_t = h, h \in \{1, 2, \cdots, H\}$ means that object h is currently being manipulated. $v_t = 0$ means no object is being interacted with, i.e., hand idle. We denote $\mathbf{v} = \{v_1, \cdots, v_T\}$. Note that the formulation developed in the rest of the paper applies to single hand interaction case; for two hands case, we run our tracking framework twice.

The objective function for "tracking interaction" is formulated as

$$\mathcal{Q}(\mathbf{P}, \mathbf{v}) = \mathcal{Q}_D(\mathbf{P}, \mathbf{v}) + \mathcal{Q}_I(\mathbf{P}, \mathbf{v}) + \mathcal{Q}_M(\mathbf{P}, \mathbf{v}), \quad (1)$$

where $\mathcal{Q}_D$ denotes the detection cost which measures how the selected candidate detections match object models; $\mathcal{Q}_I$ models the interactions between different object parts/hands; and $\mathcal{Q}_M$ enforces the (motion) dynamic model. The graphical representation of our tracking framework is illustrated in Figure 2.

3.1. Hand and Object Part Detection Cost

The hand and object part detection cost can be expanded as

$$\mathcal{Q}_D(\mathbf{P}, \mathbf{v}) = \sum_{t=1}^{T} \sum_{m=1}^{M} E_D(p^t_m, v_t). \quad (2)$$

For object part m at time stamp t, the matching cost $E_D(p^t_m = d^t_m, v_t)$ measures the loss of selecting a candidate detection d^t_m for p^t_m given the interaction status for the current frame v_t, which is further defined as

$$E_D(p^t_m = d^t_m, v_t) = w_D(m, v_t) \times s(p^t_m, d^t_m), \quad (3)$$

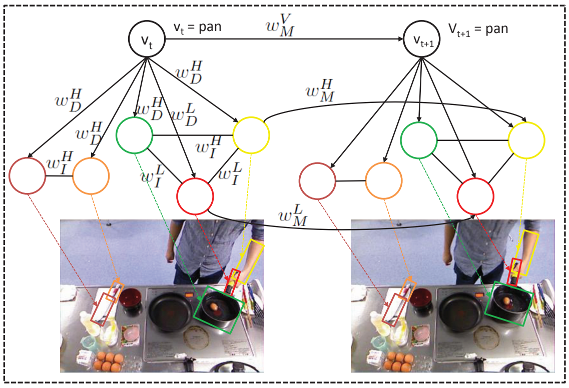

Figure 2. Graphical model of the proposed tracking framework. Note that both fry pan and knife have two parts (i.e., handle + main body). Different weighting parameters are illustrated according to the interaction status $v_t = v_{t+1} = pan$.

where $s(p^t_m, d^t_m)$ measures the dissimilarity between candidate detection d^t_m and the object part m's appearance model as

$$s(p^t_m, d^t_m) = -\mathrm{lnp}(\boldsymbol{\phi}(\mathrm{d}^\mathrm{t}_\mathrm{m})|\boldsymbol{\theta}(\mathrm{m})), \quad (4)$$

Here $\boldsymbol{\theta}(m)$ denotes the classification model trained for object part m. Using the extracted visual feature vectors from positive and negative object samples (patches) from the training data, we train a kernel SVM model $f(\mathbf{x}|\boldsymbol{\theta}(m))$ for each object class (using RBF kernel). We take the sigmoid function of the SVM output score $z = f(\mathbf{x} = \boldsymbol{\phi}(d^t_m)|\boldsymbol{\theta}(m))$ to represent the detection confidence as

$$p(\boldsymbol{\phi}(d^t_m)|\boldsymbol{\theta}(m)) = \frac{1}{1 + \exp(-z)}. \quad (5)$$

The visual features for each candidate image patch, i.e., $\boldsymbol{\phi}(d^t_m)$, are the concatenated feature vector consisting of histogram of oriented gradients (HOG) [6] and HSV color histogram. To cope with object deformation, multiple aspect ratios are modeled for objects.

The weighting parameter $w_D(m, v_t)$ is defined as

$$w_D(m, v_t) = \begin{cases} w_D^L, & \mathrm{if}\ \pi(\mathrm{m}) = \mathrm{v_t} \wedge \mathrm{m} \in \mathcal{I} \\ w_D^H, & \mathrm{else.} \end{cases} \quad (6)$$

The weighting coefficients w_D^H, w_D^L indicate how important it is to find a good match for object part m. The interaction status v_t plays the role as a switch variable which adjusts the weighting coefficient according to the interaction status. Namely, when object $\pi(m)$ is being interacted, it is most likely that some part m of it is occluded by hand (formally, $\pi(m) = v_t \wedge m \in \mathcal{I}$). In this case, finding a good match for this object part is less important, *i.e.*, smaller value of the weighting factor w_D^L. In other case, it is less possible that the object part m is occluded, and detecting it is important, i.e.,$w_D^H > w_D^L$. We denote $\mathbf{w}_D = (w_D^H, w_D^L)^T$ and we impose $w_D^H \geq w_D^L$.

3.2. Interacting Hand and Object Part Cost

The cost induced by the interaction between hand and object is defined as

$$\mathcal{Q}_I(\mathbf{P},\mathbf{v}) = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{n=m+1}^{M} E_I(p_m^t, p_n^t, v_t). \tag{7}$$

At time stamp t, the interaction cost induced by object part n and m is defined as

$$E_I(p_m^t, p_n^t, v_t) = w_I(m, n, v_t) \times s(p_m^t, p_n^t). \tag{8}$$

The compatibility function $s(p_m^t, p_n^t)$ (the smaller, the more compatible) for object part m and n is defined as

$$\begin{aligned} s(p_m^t, p_n^t) &= \frac{1}{\widehat{\sigma_{mn}^x}} \|\Delta\mathbf{x}_{mn}^t - \widehat{\delta_{mn}}\|_2 \\ &+ \frac{1}{\widehat{\sigma_v}} \|\dot{\mathbf{x}}_m^t - \dot{\mathbf{x}}_n^t\|_2. \end{aligned} \tag{9}$$

Here $\Delta\mathbf{x}_{mn}^t$ is the measured distance between object part m and n. $\widehat{\delta_{mn}}$ is the empirical mean distance between the interacting object part m and n estimated from the annotated training data. $\widehat{\sigma_{mn}^x}$ is the corresponding estimated variance for $\widehat{\delta_{mn}}$. We denote by $\dot{\mathbf{x}}_m^t$ the velocity of object part m at frame t, i.e., $\dot{\mathbf{x}}_m^t = \mathbf{x}_m^{t+1} - \mathbf{x}_m^t$. $\widehat{\sigma_v}$ is the empirical variance of object velocity estimated from the annotated training data. Namely, the compatibility between two object parts m and n consists of two measurements: 1) the difference between their current and empirical mean relative distance; and 2) their current relative speed. The definition for the compatibility measure is motivated by the observations that 1) if two object parts are interacting, the displacement between them follows some prior distribution (e.g., the distance between two parts of the same object is fixed) and 2) the relative movement between them should be small (e.g., when the hand is holding an object, they are moving with the same velocity).

The weighting parameter $w_I(m, n, v_t)$ is defined as

$$w_I(m, n, v_t) = \begin{cases} w_I^L, & \text{if} \quad \mathrm{C1(m,n,v_t)} = 1 \\ w_I^H, & \text{elseif } \mathrm{C2(m,n,v_t)} = 1 \\ 0, & \text{else,} \end{cases} \tag{10}$$

where the corresponding indicator functions C1(.) and C2(.) are defined as:

$$\begin{aligned} \mathrm{C1(m,n,vt)} &= I\{\pi(m) = \pi(n) \wedge \pi(n) = v_t \wedge n \in \mathcal{I}\} \\ &\vee \; I\{\pi(m) = \pi(n) \wedge \pi(m) = v_t \wedge m \in \mathcal{I}\} \\ &\vee \; I\{m = 1 \wedge \pi(n) = v_t \wedge n \in \mathcal{I}\}. \end{aligned} \tag{11}$$

$$\begin{aligned} \mathrm{C2(m,n,vt)} &= I\{\pi(m) = \pi(n) \wedge \pi(n) \neq v_t\} \\ &\vee \; I\{\pi(m) = \pi(n) \wedge \pi(n) = v_t \wedge (n,m) \notin \mathcal{I}\} \\ &\vee \; I\{m = 1 \wedge \pi(n) = v_t \wedge n \notin \mathcal{I}\}. \end{aligned} \tag{12}$$

Three cases are considered for the interaction between two object parts (or hand and object part). The interaction status variable v_t again serves as a switch variable which adjusts the importance weighting of the geometrical relation between two parts depending on different situations.

1. Case I: There are two sub cases. In the first sub case, two object parts belong to the same object and are not being manipulated by hand. In the second sub case, two parts belong to the hand and the visible part of the object which is being manipulated by hand. In both sub cases, both parts are most probably visible and their fixed geometric relation should be enforced, *i.e.*, the weighting coefficient w_I^H should be large.

2. Case II: There are two sub cases. In the first sub case, two parts correspond to the same object which is currently being manipulated, with one part hold by hand. In the second sub case, two parts belong to the hand and the possibly occluded part of the object which is currently being manipulated by hand. In both sub cases, the part which is hold by hand is most probably occluded, therefore we apply a small value coefficient w_I^L to *softly* enforce the geometric relationship between these two parts.

3. Case III: Those parts which are not belonging to the same object or the interacting hand-object pair are considered to be irrelevant parts, and no geometric relationship should be imposed on them.

We denote $\mathbf{w}_I = \{w_I^H, w_I^L\}$ and we impose $w_I^H \geq w_I^L$.

3.3. Hand and Object Tracking Cost

The hand and object tracking cost includes two parts. The first part considers the object movement and the second part considers the transition property between two interaction status (i.e., how probable is the interaction status changed from object a to b). The tracking cost is defined as

$$\mathcal{Q}_M(\mathbf{P},\mathbf{v}) = \sum_{t=1}^{T-1}\sum_{m=1}^{M} E_M^1(p_m^t, p_m^{t+1}) + \sum_{t=1}^{T-1} E_M^2(v^t, v^{t+1}). \tag{13}$$

The motion tracking cost for object part m from frame t to $t+1$ is defined as

$$E_M^1(p_m^t, p_m^{t+1}) = w_M(m, v_t) \times s(p_m^t, p_m^{t+1}). \tag{14}$$

We assume a constant velocity motion model and therefore the motion compatibility function $s(p_m^t, p_m^{t+1})$ is defined as

$$s(p_m^t, p_m^{t+1}) = \frac{1}{\widehat{\sigma_v}} ||\mathbf{x}_m^{t+1} - \mathbf{x}_m^t - \widehat{\mathbf{v}_m}||_2, \tag{15}$$

where $\widehat{\mathbf{v}_m}$ is empirical mean speed of object part m estimated from the annotated training data.

The weighting parameter $w_M(m, v_t)$ is defined as

$$w_M(m, v_t) = \begin{cases} w_M^L, & \text{if } \mathrm{m} \in \mathcal{I} \wedge \mathrm{v_t} = \pi(\mathrm{m}) \\ w_M^H, & \text{else.} \end{cases} \tag{16}$$

Again, the interaction variable v_t adjusts the importance of tracking individual object part under different situations. When the object part is being interacted with, it is mostly likely we cannot reliably track its movement due to occlusion, and therefore we should apply a median value weighting coefficient w_M^L. When the object is not being interacted with, tracking is easier and we therefore apply a large value weighting coefficient w_M^H.

The second part of the cost involves interaction status transition, *i.e.*, from interacting with one object to another and the cost is defined as

$$E_M^2(v^t = h, v^{t+1} = l) = w_M^V \times (-ln(\widehat{\pi_{hl}})) \tag{17}$$

where $\{\widehat{\pi_{hl}}\}, h, l \in \{0, \cdots, H\}$ are the transition probabilities. Each transition probability $\widehat{\pi_{hl}}$ is estimated from the annotated training data. The parameter set is denoted by $\mathbf{w}_M = \{w_M^H, w_M^L, w_M^V\}$ and we impose $w_M^H \geq w_M^L$.

3.4. Model Learning and Inference

The learning task is to estimate the optimal values for the parameter set $\mathbf{w} = \{\mathbf{w}_D^T, \mathbf{w}_I^T, \mathbf{w}_M^T\}$, given N training videos ($j = 1, \cdots, N$) with the corresponding annotations. We also denote $\mathbf{w}^H = (w_D^H, w_I^H, w_M^H)^T$ and $\mathbf{w}^L = (w_D^L, w_I^L, w_M^L)^T$. Full annotation (i.e., bounding box for each object and the interaction status for each frame) for the whole training sequence is very time consuming. Instead, we use *sparsely* annotated data, i.e., object bounding boxes and interaction status labels are only given for some discontinued frames and the majority of video frames are unlabeled. For training video j, we denote by $(\mathbf{P}^{O,j}, \mathbf{v}^{O,j})$ the labeled data (with annotated value $(\widetilde{\mathbf{P}^{O,j}}, \widetilde{\mathbf{v}^{O,j}})$) and $(\mathbf{P}^{H,j}, \mathbf{v}^{H,j})$ the unlabeled data, respectively. We have $\mathbf{P}^j = \mathbf{P}^{O,j} \cup \mathbf{P}^{H,j}$ and $\mathbf{v}^j = \mathbf{v}^{O,j} \cup \mathbf{v}^{H,j}$, respectively. Formally, the objective of learning is casted as

$$\begin{aligned} & \min \sum_{j=1}^{N} \mathcal{Q}^j(\widetilde{\mathbf{P}^{O,j}}, \mathbf{P}^{H,j}, \widetilde{\mathbf{v}^{O,j}}, \mathbf{v}^{H,j} | \mathbf{w}) + \lambda \sum_{j=1}^{N} \xi_j \quad, \\ & w.r.t. \quad \mathbf{w}, \{\mathbf{P}^{H,j}\}, \{\mathbf{v}^{H,j}\}, \\ & s.t. \quad\; \mathbf{1}^T \mathbf{w} = 1, \quad \mathbf{w} \succeq 0, \\ & \qquad\quad \mathbf{w}^H - \mathbf{w}^L \succeq 0, \\ & \min_{\mathbf{P}^j, \mathbf{v}^j} Q^j(\mathbf{P}^j, \mathbf{v}^j) - \min_{\mathbf{P}^{H,j}, \mathbf{v}^{H,j}} Q^j(\mathbf{P}^{H,j}, \mathbf{v}^{H,j}, \widetilde{\mathbf{P}^{O,j}}, \widetilde{\mathbf{v}^{O,j}}) \\ & \geq \delta(\mathbf{P}^{O,j}, \mathbf{v}^{O,j} | \widetilde{\mathbf{P}^{O,j}}, \widetilde{\mathbf{v}^{O,j}}) - \xi_j,\ \xi_j \geq 0,\ \forall j. \end{aligned} \tag{18}$$

Here $\delta(\mathbf{x}|\mathbf{x}') = 0$ if $\mathbf{x} = \mathbf{x}'$, otherwise $\delta(\mathbf{x}|\mathbf{x}') = 1$. To simplify notation, note that $\mathcal{Q}(\mathbf{P}, \mathbf{v}|\mathbf{w})$ can be equivalently written as $\mathbf{w}^T \mathbf{\Psi}(\mathbf{P}, \mathbf{v})$ by simple rearrangements of the variables. Therefore the optimization problem is a constrained linear program. While the number of constraints is exponential in the number of options for the configurations $\{\mathbf{P}^j, \mathbf{v}^j\}$, we solve it efficiently using the cutting-plane algorithm [10]. The key step in optimization is to efficiently compute the minimal values for $Q^j(\mathbf{P}^{H,j}, \mathbf{v}^{H,j})$ and $Q^j(\mathbf{P}^j, \mathbf{v}^j)$ Since the graphical model is not a tree structure (it has cycles), there exists no efficient method to compute the exact solution. We therefore solve the problem approximately using loopy belief propagation method [19]. Given the learned $\mathbf{w}$, for a new video sample with detected object and hand candidates, we jointly infer $\mathbf{P}$ and $\mathbf{v}$ by loopy belief propagation [19].

4. Fine Granularity: Activity Detection

In this section, we will introduce a novel activity detection framework, which takes great advantage of the inferred interaction status and object/hand tracking results, to perform 1) efficient feature pooling and 2) accurate action detection by integrating interaction status information (prior) and motion features (likelihood) into a graphical model. The goal of activity detection is to label each video segment with appropriate action labels, e.g., cutting, seasoning, etc. For detecting activities, we first temporally segment the video sequence into overlapping small clips. We adopt over-segmentation so that we end up with more segments and avoid merging two activities into one segment. Multiple temporal segmentation windows with sizes of 30, 60, 90 frames and 10, 20, 30 overlapped frames are utilized. Note that interaction status change boundaries are preserved.

Interaction Centered Feature Pooling: Given a spatio-temporal video volume, the common way to compute a feature representation is to compute the histogram (bag-of-words) of all local motion features extracted inside the volume, known as global pooling. However, this scheme is incapable of removing noisy and redundant background motion features, resulting in noisy video level representation.

Based on the inferred interaction status and the positions of the interacting hands/objects, we propose an **Interaction Centered Feature Pooling** scheme to perform feature extraction and representation (pooling) efficiently and compactly. Specifically, within each video segment local feature extraction and pooling is **ONLY** performed within a sub volume centered on the position where interaction occurs (with the size of the object-in-use), i.e., the spatio-temporal sub volume of the objects-in-use and hands obtained from the proposed interaction tracking module (as illustrated in Figure 1). Large number of irrelevant background and human body motion features that could harm the action representation are thus removed. The local features we extract are dense motion trajectories [26]. For each trajectory, we extract histogram of oriented gradient (HoG), motion boundary histogram (MBH), histogram of optical flow

(HoF) and trajectory shape (TS) as in [26]. These features are encoded using a dictionary pre-trained on the training data using K-means algorithm ($K = 2000$). Each video segment i is then represented by a bag-of-words vector $\mathbf{x}_i$ (we abuse the notation here).

Granularity Fusion for Action Inference: Due to the strong correlation between action label and the type of object-in-use, the inferred interaction status serves as very important prior information on what action is being performed. We therefore develop a CRF-graph based method to integrate this prior information with local motion feature histogram pooled from the positions of object-of-interest for more robust action detection. We assume a video is pre-segmented into L segments. Each video segment $i = 1, 2, \cdots, L$ is represented by a node in the graph. We denote by $\mathbf{x}_i$ the histogram representation of local motion features for the video segment i, and by y_i its corresponding action label. The interaction status label for segment i is denoted by v_i. We denote $\mathbf{v} = \{v_1, \cdots, v_L\}$, $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_L\}$ and $\mathbf{y} = \{y_1, \cdots, y_L\}$. The energy function for the conditional model is defined as

$$E(\mathbf{y}|\mathbf{v}, \mathbf{X}) = \sum_{i=1}^{L} \varphi_i(\mathbf{x}_i, y_i|v_i) + \sum_{i \neq j} \varphi_{ij}(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j). \tag{19}$$

The unary potential is given by

$$\varphi_i(y_i = c, \mathbf{x}_i|v_i = s) = d(\mathbf{x}_i, c) \times e(c, s), \quad \forall c, s. \tag{20}$$

The detection score (i.e., likelihood) given motion feature $\mathbf{x}_i$ is defined as (we assume C action categories)

$$d(\mathbf{x}_i, c) = \frac{\exp(f(c|\mathbf{x}_i))}{\sum_{c'=1}^{C} \exp(f(c'|\mathbf{x}_i))}, \quad \forall c, \tag{21}$$

where $f(c|\mathbf{x}_i))$ denotes the detection (SVM classifier output) score for action label c, given motion feature representation $\mathbf{x}_i$. The empirical compatibility score between the action type c and the object-of-interest type s (i.e., prior) is estimated from the annotated training data as

$$e(c, s) = \frac{\#\{y(\mathbf{x}_i) = c, v_i = s\}}{\#\{v_i = s\}}, \forall c,\ s. \tag{22}$$

Edges link spatio-temporal nearby nodes i and j, i.e., $i \in \mathcal{N}(j)$. The corresponding pair-wise potential is defined as

$$\varphi_{ij}(y_i = c_i, y_j = c_j, \mathbf{x}_i, \mathbf{x}_j) = \left\{ \begin{array}{cc} a(c_i, c_j), & i \in \mathcal{N}(j) \\ 0, & \text{else.} \end{array} \right. \tag{23}$$

The neighboring node compatibility score $a(c_i, c_j)$ can be empirically estimated from the training data as

$$a(c_i, c_j) = \frac{\#\{y_i = c_i, y_j = c_j\}}{\#\{y_i = c_i\} + \#\{y_j = c_j\}}, \forall c_i,\ c_j,\ i \in \mathcal{N}(j). \tag{24}$$

The optimal action labels $\mathbf{y}$ is obtained by maximizing the energy function $E(\mathbf{y}|\mathbf{v}, X)$ using loopy belief propagation [19].

5. Experiments

Our evaluations are two-fold. First, we show the advantage of the proposed joint interaction status and multiple objects' tracking framework over various state-of-the-art trackers that do not employ interactional contextual information. Second, we show how the inferred interaction status (coarse granularity) facilitates fine-grained action recognition. Experiments are performed on two challenging fine-grained action benchmarks with complicated human and object interactions. 1) **ICPR 2012 Kitchen Scene Context based Gesture Recognition dataset (KSCGR) [1]**. There are five candidate cooking menus cooked by five different actors. Each of the videos are from 5 to 10 minutes long containing $9,000$ to $18,000$ frames. The task is to recognize eight types of cooking motions such as *baking*, *boiling*, *breaking*, etc. The objects of interest (which we track) are *fry pan*, *oil bottle*, *salt bottle*, *bowl*, *knife*, *spoon*, *chopstick*, *spatula*, *chopping board*, *egg* and *ham.* 2) **MPII Fine-grained Kitchen Activity Dataset (MPII) [23]**. It contains 65 different cooking activities, such as *cut slices*, *pour spice*, etc., recorded from 12 participants. In total there are 44 videos with a total length of more than 8 hours or $881,755$ frames. The dataset contains a total of $5,609$ annotations of 65 activity categories. It has high variations because participants are just asked to prepare one to six of a total of 14 dishes without any guide on how to perform individual steps. For this dataset, the objects that we detect and track are *bottle*, *bowl*, *bread*, *charger*, *electric range*, *cup*, *cupboard*, *chopping board*, *dough*, *drawer*, *egg*, *lid*, *food wrapper*, *knife*, *pan*, *slicer*, *plate*, *pot*, *blender*, *seasoning bottle*, *bottle rack*, *juicers*, *tin*, *tin opener* and *towel.* For both datasets, training and testing set are pre-partitioned. For each object part, we annotate 1000 positive samples and 10000 randomly cropped negative samples from the training data to train the detector for our method. These annotated samples are also used to train the initial tracker for other tracking methods being compared.

5.1. Interaction Tracking Results

We select eight manipulation sequences with average frame number about 2000 from the testing set of KSCGR to evaluate the interaction tracking performance. These sequences are manually annotated with object positions and interaction status labels. It is intractable to provide annotations for all testing sequences for both datasets. Nevertheless, the manipulation sequences we select are representative sequences which contain all kinds of human object interactions with frequent occlusions and we believe they are sufficient for qualitatively evaluating the tracking perfor-

Sequence	OAB		TLD		Ours	
	Err.	Prec.	Err.	Prec.	Err.	Prec.
baking (3786)	42.9	0.27	36.2	0.34	**28.9**	**0.56**
boiling (3320)	40.7	0.30	35.2	0.38	**25.5**	**0.59**
breaking (299)	36.7	0.32	34.5	0.35	**20.4**	**0.64**
cutting (1373)	38.9	0.31	40.5	0.29	**24.8**	**0.66**
mixing (705)	36.6	0.40	32.8	0.52	**17.9**	**0.68**
peeling (3241)	45.3	0.21	40.7	0.24	**30.1**	**0.62**
seasoning (303)	37.8	0.35	34.2	0.33	**12.3**	**0.69**
turning (3402)	39.1	0.29	33.8	0.37	**15.4**	**0.71**

Table 1. Comparisons of tracking performances of various methods. Numbers of frames are indicated in brackets.

mance. We compare our interaction tracking method with two state-of-the-art trackers including the OAB tracker [8] and the TLD tracker [11]. These two trackers are applied to track each object and hand separately. The measuring metrics we use are: 1) **Average Distance Error (Err.)**: the average distance between the center of the identified bounding box and the center of the ground-truth bounding box; and 2) **Precision (Prec.)**: the average percentage of frames for which the overlap between the identified bounding box and the ground-truth bounding box is at least 50 percent. In Table 1, measurements are averaged over all target objects and over all frames in the video sequence. Figure 3 visualizes the inferred interaction status with the corresponding ground truth annotations for two manipulation video sequences. We also show several example frames of the tracking results given by both 1) TLD tracker (dashed line rectangle) and 2) our tracker (solid line rectangle).

From Table 1 we observe that our proposed interaction tracking method outperforms TLD and OAB trackers significantly, which demonstrates that modeling the contextual information between interaction status and mutual occlusion for joint hand and object tracking leads to substantial performance improvements over the methods that ignore this important cue. From Figure 3 we note 1) the inferred interaction status is quite precise; and 2) for TLD tracker, the tracked target positions drift when occlusion occurs during manipulation. For example, in frame 3374 and 3565 of *seasoning*, the interacting hand occludes the spoon, therefore the TLD tracker fails. In frame 3530 and 3565, the TLD tracker recognizes the oil bottle as salt bottle due to similar appearance. In contrast, our method which explicitly models the contextual information between tracking status and mutual occlusion alleviates these issues.

5.2. Fine-grained Action Detection Results

The inferred interaction status is then utilized to detect fine-grained actions. To demonstrate the effectiveness of integrating interaction status (prior) and local motion features (likelihood), we perform action detection using: 1) the proposed granularity fusion based action detection framework in Section 4; 2) the same action detection framework but by utilizing interaction status information only (i.e., setting $\varphi_i(y_i = c, \mathbf{x}_i | v_i = s) = e(c, s)$ in Eqn. (20)), denoted as **Interaction Status**; 3) the same action detection framework but by utilizing local motion feature information only (i.e., setting $\varphi_i(y_i = c, \mathbf{x}_i | v_i = s) = d(\mathbf{x}_i, c)$ in Eqn. (20)), denoted as **Motion Features**. For KSCGR dataset, the evaluation metric is the mean recognition F-score over all action categories. We also compare our method to the best reported result in the contest by Doman and Kuai [1], denoted as **Best@KSCGR**. The comparison results are summarized in Table 2. For MPII dataset, we follow experimental configuration and evaluation metric defined by the dataset developer [23]. In brief, leave-one-person-out cross validation is used. We also compare our method to the best reported result in [23], denoted as **Best@MPII**. Multi-class precision (Pr) and recall (Rc), and the mean value of single class average precision (AP) are reported in Table 3.

Method	Best@KSCGR	Interaction Status	Motion Features	Ours
Mean F-score	0.74	0.56	0.69	**0.79**

Table 2. Detection performance (mean F-score for all classes) comparisons for KSCGR dataset.

Approach	Prec.	Recall	AP
Best@MPII	19.8	40.2	45.0
Interaction Status	15.8	30.3	38.6
Motion Features	22.3	44.7	49.6
Ours	**28.6**	**48.2**	**54.3**

Table 3. Detection performance comparisons for MPII dataset.

We note 1) using coarse interaction status information only (i.e., which object is being manipulated) already yields good detection performances. This shows that interaction status conveys important prior information on the type of action being performed; 2) combining interaction status with motion features within the proposed action detection framework significantly boosts the detection performances, compared with using local motion features alone; 3) for MPII dataset, the method Best@MPII uses global motion feature pooling for action representation. Our interaction centered pooling method (Motion Features) outperforms the global pooling method. This demonstrates that interaction centered pooling significantly attenuates the background noisy motion features and therefore yields better action representation; and 4) our multiple modality analysis framework significantly outperforms the state-of-the-art detection performances for both benchmarks.

6. Conclusions

We utilized interactional context information for tracking multiple interacting objects and hands under mutual occlusions. Based on this tracking framework, we further proposed a multiple granularity analysis framework for fine-grained action detection, which outperforms the state-of-the-art on two challenging fine-grained action benchmarks.

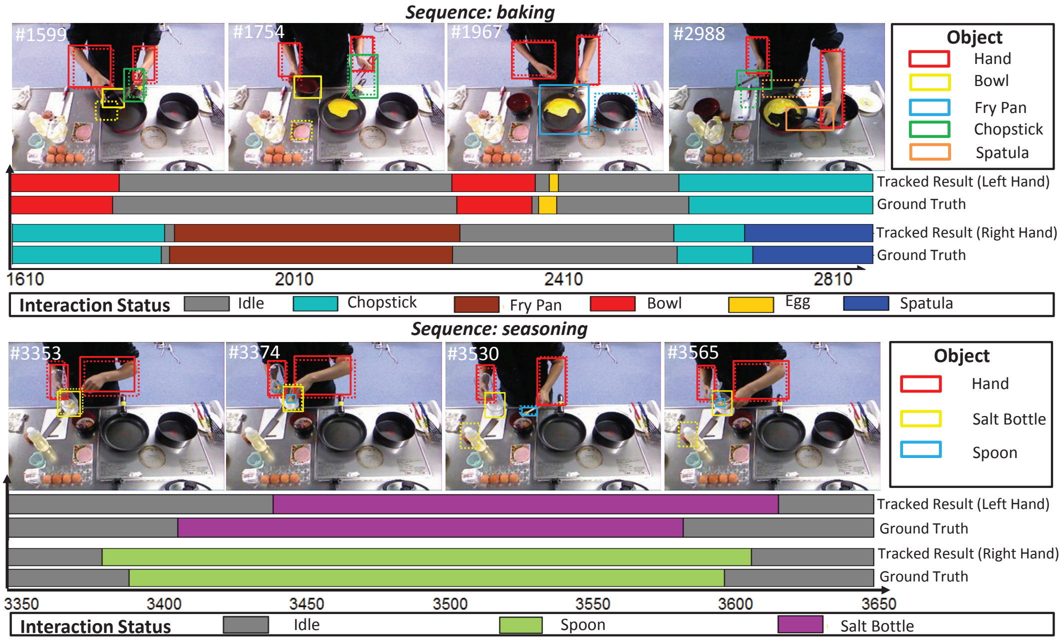

Figure 3. Examples of the tracked interaction status together with sample frames with object tracking results. Top: action sequence *baking*. Bottom: action sequence *seasoning*. The objects tracked by our method are annotated with solid line rectangle and those tracked by TLD tracker are with dashed line rectangle. The annotation rectangles are only shown for objects under interaction.

Acknowledgment

This study is supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

References

[1] http://www.murase.m.is.nagoya-u.ac.jp/kscgr/index.html.

[2] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *T-PAMI*, 33(9):1806–1819, 2011.

[3] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[4] C. Chang, R. Ansari, and A. Khokhar. Multiple object tracking with kernel particle filter. In *CVPR*, volume 1, pages 566–573, 2005.

[5] H.-T. Chen, H.-H. Lin, and T.-L. Liu. Multi-object tracking using dynamical graph matching. In *CVPR*, volume 2, pages 210–217, 2001.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] M. de La Gorce, N. Paragios, and D. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, 2008.

[8] H. Grabner, M. Grabner, and H. Bischof. Quicktime reference guide, quicktime cd. In *BMVC*, pages 47–56, 2006.

[9] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. In *CVPR*, volume 1, pages 864–871, 2004.

[10] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56, 2010.

[12] H. Kjellström, J. Romero, D. Martínez, and D. Kragić. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, pages 336–349, 2008.

[13] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *CoRR*, 2012.

[14] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[15] J. Lei, X. Ren, and D. Fox. Fine-grained kitchen activity recognition using rgb-d. In *ACM Conference on Ubiquitous Computing*, pages 208–211, 2012.

[16] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *CVPR*, 2013.

[17] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *ICCV*, volume 1, pages 572–578, 1999.

[18] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV*, Corfu, Greece, 1999.

[19] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999.

[20] J. C. Niebles, H. Wang, and L. Fei-fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[21] B. Packer and D. Koller. A combined pose, object, and feature model for action understanding. In *CVPR*, 2012.

[22] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[23] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, pages 1194–1201, 2012.

[24] M. S. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, pages 1593–1600, 2009.

[25] H. Trinh, Q. Fan, P. Gabbur, and S. Pankanti. Hand tracking by binary quadratic programming and its application to retail activity recognition. In *CVPR*, pages 1902–1909, 2012.

[26] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[27] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *T-PAMI*, 33(7):1310–1323, 2011.

[28] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, 2007.

[29] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *ICCV*, volume 1, pages 212–219, 2005.

[30] B. Yao, A. Khosla, and L. Fei-fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In *ICML*, 2011.

[31] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *T-PAMI*, 33(9):1728–1743, 2011.

[32] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *CVPR*, volume 2, pages 406–413, 2004.