STARE: Spatio-Temporal Attention Relocation for Multiple Structured Activities Detection

Kyuhwa Lee, Dimitri Ognibene, Hyung Jin Chang, Tae-Kyun Kim, and Yiannis Demiris, *Senior Member, IEEE*

***Abstract*— We present a spatio-temporal attention relocation (STARE) method, an information-theoretic approach for efficient detection of simultaneously occurring structured activities. Given multiple human activities in a scene, our method dynamically focuses on the currently most informative activity. Each activity can be detected without complete observation, as the structure of sequential actions plays an important role on making the system robust to unattended observations. For such systems, the ability to decide where and when to focus is crucial to achieving high detection performances under resource bounded condition. Our main contributions can be summarized as follows: 1) information-theoretic dynamic attention relocation framework that allows the detection of multiple activities efficiently by exploiting the activity structure information and 2) a new high-resolution data set of temporally-structured concurrent activities. Our experiments on applications show that the STARE method performs efficiently while maintaining a reasonable level of accuracy.**

***Index Terms*— Activity detection, visual attention, resource allocation, stochastic context-free grammars.**

I. Introduction

ACTIVE vision [1] and visual attention [2] systems dynamically select parts of a visual input for efficient processing, which have high importance when the amount of visual input produces an excessive computational load. Common techniques used are applying a region of interest (ROI), camera view selection, and changing the pan, tilt, zoom (PTZ) camera parameters. The active vision systems have been gaining more interest in vision community as their performances are comparable or even surpass conventional passive vision systems [2]–[5]. These systems have been applied in a wide range of areas such as surveillance [6], [7], object detection and tracking [8]–[10], object recognition [4], [5], [11], camera view selection [6] and action recognition [12]–[15].

One of the main purposes of these methods is to discard in *a priori* a part of the information using one or more attention policies. This additional layer of complexity can be, however, non-trivial to deal with if a system has to process complex dynamic scenarios. Thus, active vision systems have been mostly applied on the recognition of isolated and temporally unstructured actions [14]. We are instead interested in scenarios where several independent, long structured activities can occur in parallel. To deal with these scenarios, we develop a dynamic attention relocation method which makes use of the information acquired during the activity detection process, aiming to detect such activities with less computational resources while maintaining comparable detection performances.

State-of-the-art complex structured activity analysis techniques [16] often utilize syntactic approaches [17], [18]. We employ stochastic context-free grammars (SCFG) [19], [20] to both represent and detect human activities. SCFG-based methods have been primarily used as activity *recognizer* instead of *generator* (e.g. [19], [21]–[24]), which is not sufficient for our purpose. Hence, we augment the conventional parser by exploiting the structural information encoded in the SCFG to predict the next successive actions while recognizing, which will provide crucial information to the attention relocation process.

In this paper, we present a spatio-temporal attention relocation (STARE) method for efficient activity detection using the knowledge about long-term structured activities. Although our system does not necessarily aim to work in real time, we aim to detect multiple activities efficiently with reasonable accuracy while not committing a full observation. This is done by exploiting the structural constraints of actions. An overview of our system is illustrated in Figure 1.

Our approach is implemented in three layers: The low layer extracts visual features from the attended area, while the middle layer recognizes elementary human actions. The high layer predicts the next successive actions given the current and previous action observations. These predicted actions play a vital role for computing the following information-theoretic attention relocation policies: *a) Allocate resources on an area*

Manuscript received October 26, 2014; revised May 31, 2015; accepted September 14, 2015. Date of publication October 7, 2015; date of current version November 3, 2015. This work was supported in part by the European Commission Seventh Framework Programme through the WYSIWYD Project under Grant FP7612139 and in part by the DARWIN Project under Grant FP7270138. The dataset collection was supported by the Engineering and Physical Sciences Research Council Network on Vision and Language under Grant Scheme Pump-priming Vision & Language Research 2013-1. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Janusz Konrad. *(Corresponding authors: Kyuhwa Lee and Hyung Jin Chang.)*

K. Lee is with the Chair in Brain-Machine Interface Laboratory, Center for Neuroprosthetics, School of Engineering, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: kyu.lee@epfl.ch).

D. Ognibene is with the Defense Technical Information Center, Universitat Pompeu Fabra, Barcelona 08018, Spain (e-mail: dimitri.ognibene@upf.edu).

H. J. Chang and Y. Demiris are with the Personal Robotics Laboratory, Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: hj.chang@imperial.ac.uk; y.demiris@imperial.ac.uk).

T.-K. Kim is with the Computer Vision and Learning Laboratory, Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: tk.kim@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2487837

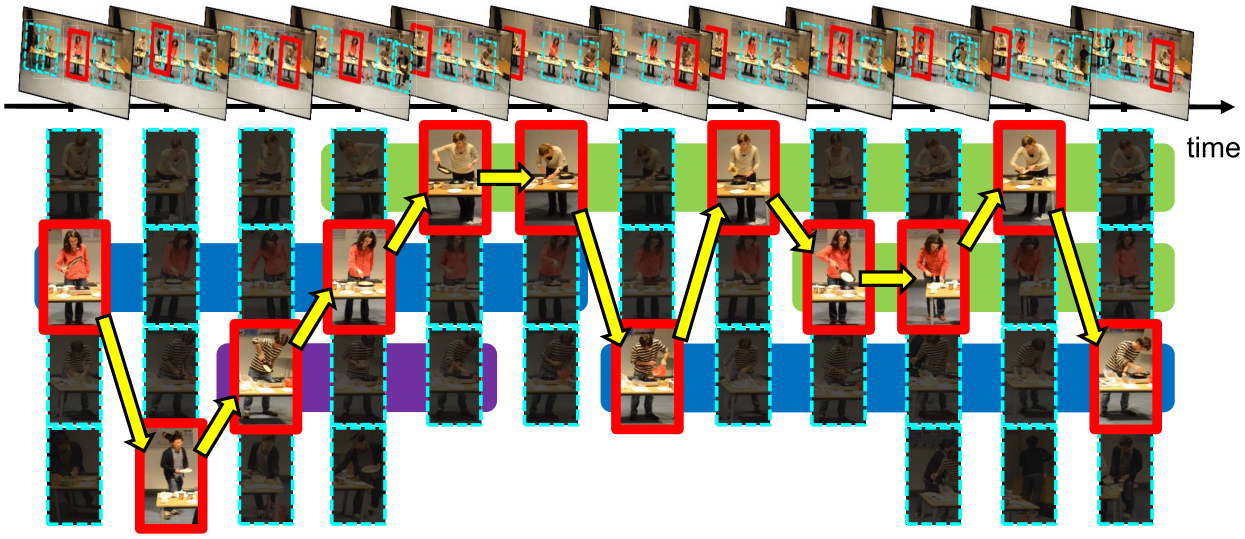


Fig. 1. Illustrative explanation of attention relocation by STARE method. The red boxes indicate where the system made an observation at each sampling time to dynamically focus on the action that is expected to be the most informative. The color bars on the background denote different activity classes. (Best shown in color).

with the highest activity detection confidence. b) Allocate more resources on an area that is most likely to contain an undetected activity. c) The combination of these two policies for balanced approach. The main contributions of our work can be summarized as follows:

1) Information-Theoretic Active Attention Relocation: We present an attention relocation system suitable for detecting concurrent activities under bounded computational resources by exploiting activity structures. This is done through an SCFG-based action prediction, which can predict the successive actions given the previous observation actions by exploiting temporally structured activity information. The predicted future actions provide crucial information to our attention relocation system. This method differs from the traditional use of SCFG parsing, e.g. [19], [21], [24]–[27], where the parser is used as a recognizer (classifiers). We use the parser as a recognizer as well as generator to predict the next possible actions and determine where to allocate resources in the next time step.

2) New dataset of Temporally Structured Concurrent Activities: We present a new activity dataset containing 6 different structured cooking activities with distractors where multiple activities may occur concurrently. To the best of our knowledge, this is the first long-term structured human activity dataset with multiple classes in full-HD resolution (1080p).

Last but not least, it is also worth mentioning that the principles of our model are inspired from the neuroscience field, such as the hierarchical representation of actions [25], [28]–[30], the internal simulation of future actions [28] and attention control [31]–[36].

II. Related Works

Attention relocation (control) systems have been widely used to reduce the computational cost of visual processing by focusing on more informative parts. As explained in several recent reviews [2], [37], the current studies focus mainly on approaches which are based on spatially and temporally localized visual signals, which are not sufficient to model the information requirements of complex tasks in a dynamic context, such as activity detection. As admitted in [2], such approaches put more emphasis on predicting human fixations using local and unstructured temporal information. To elaborate the differences, we list some representative works in three major categories related to our work.

1) Bottom-Up Attention Relocation Systems: Bottom-up attention controllers, both bio-inspired approach [38] and information-theoretic approach [39], make decisions based on low-level visual features. Jiang *et al.* [8] recently presented a visual-cue-based attentional region detection in a static image, where [40] combines an outdoor scene classification system based on 'gist', computed from multi-scale set of early-visual features, with a saliency based attention mechanism. Denzler *et al.* [41] studied dynamic attention selection for object tracking using the uncertainty information acquired from Kalman filter. Chang *et al.* [42] presented a dynamic bottom-up attention control scheme for speeding up the background subtraction process. They generated a dynamic attention probability map by considering specifically designed attention properties of the application. Since these bottom-up systems do not consider high-level structures, the information they select does not necessarily contribute to improve high-level tasks such as activity detection and recognition.

2) Top-Down Attention Relocation Systems: Navalpakkam and Itti [43] augment the bio-inspired model [38] by adding a top-down component of visual attention, which computes the gist of a scene to acquire the prior distribution of a given object and select task-related features to optimize static object detection. Vijayanarasimhan and Kapoor [9] tackle an object detection problem which uses an approximated value of information to prioritize more informative features among the

TABLE I
COMPARISON WITH PREVIOUS DATASETS

Dataset	Resolution	Example Categories (not exhaustive)	Comments
BEHAVE	640x480	Walk together, meet, chase	Multiple people but no structured activity.
UCR Videoweb	640x480	Talk on phone, push button, wave hand, walk backwards	Multiple long-sequence videos and multiple people, but no structured activities.
CMU Kitchen	1024x768	Cooking: brownie, egg, pizza, salad, sandwich	single person per sample.
CMU Mocap	352x240	1 person: martial arts, basketball, acrobatics 2 people: quarrel, pull-resist, conversation	Up to 2 people for unstructured short-term activities, and 1 person with actions only (no structured activities).
MuHAVi	720x576	Crawl on knees, climb a ladder, pick up and throw an object	Multiple views but single person per sample. Short-term activities only.
ViSOR	800x600	Drinking, leaving an object, sitting on a chair, running	multiple people but no structured activity.
Weizmann	180x144	Jumping jack, gallop sideways, swing a bag, two-hand wave	No structured activities, low resolution, single person.
CAVIAR	384x288	Set1: Walking (straight, return, B-line), browsing a shop Set2: People entering and exiting a store	Low resolution, short-term activities only. Not enough samples due to a large number of classes. (less than 10 samples per class)
CANDELA	352x288	Pick up object, sit and leave bag, sit and handover a bag, three people gather and depart, park car and walk away	Low resolution, no concurrent activities.
UT-Tower	360x240	Digging, carrying, jumping, Walking	Low resolution, short-term activities only.
UT-Interaction	720x480	Shake-hands, point, hug, push, kick, Punch	No structured activities, single person.
KTH	160x120	Walking, jogging, boxing, hand clapping	No structured activities, low resolution, single person.
TRECVID	720x576	Running, meeting, embracing, pointing, opposing flow	No structured activities.
VIRAT	1920x1080	Person loading/unloading an object to vehicle, person opening/closing a trunk, person getting into/out of a vehicle.	2 types of long-term structured activities (delivery/take away) exist in less than 20 video samples. Only few scenarios have concurrent activities.

pool of features such as local descriptors, textures and color histograms. The above mentioned methods depend on static features, which is not optimal in dynamic scenarios.

For dynamic top-down attention relocation, Sommerlade and Reid [6] use mutual information maximization technique to detect and track multiple targets based on their motion, whereas our method is based on type of the action being performed. Although [12] represents human actions using layered hidden Markov models, they model only unstructured actions such as phone conversation or face-to-face conversation, whereas our approach considers the detection of concurrently occurring temporally structured activities. Ognibene and Demiris [14] model a set of actions as a mixture of Kalman Filters and compute the maximum information gain to select the view of the camera between the human's hand and the expected positions of action target candidates. However, this method differs from our approach since they rely only on the object position and considers only a single, unstructured action at time.

Finally, although there have been also efforts [44], [45] to detect actions as early as possible without watching in full, their models are more suited for detecting actions rather than activities and cannot be used to predict future actions which are crucial for our purpose.

3) Comparison of Our Dataset With Previous Datasets: Existing benchmark datasets are not suitable for our evaluation. Our requirements are: (1) multiple human objects in a scene with independent activities occurring in parallel; (2) multiple samples per activity class; (3) temporally structured and long activities (at least 60 seconds); (4) high resolution. We compare our new dataset with the previous datasets using the above criteria in Table I. A detailed description about the new dataset will be given in Section IV-C.

III. SPATIO-TEMPORAL ATTENTION RELOCATION (STARE) FOR EFFICIENT ACTIVITY DETECTION

Our goal is to detect ongoing activities in scenarios where several different activities may evolve simultaneously over time with different speeds while attending (watching) only one of the candidate areas at a time. The base idea of our approach is that past observed actions and the temporal structure of activities are valuable sources of information not only for detecting ongoing activities but also for deciding where to attend among candidate areas in the next time step.

We adopt a probabilistic generative model, Stochastic Context-Free Grammars (SCFG), to represent the temporal structure of activities and encode the observed actions history. By efficiently interleaving the action (observations) parsing and action prediction, in parallel for each activity we can predict the next expected action observation distribution using the activity structure and past observations. Using this expected action observation distributions we estimate the potential improvement of the activity recognition achievable corresponding to the different candidate areas to attend.

We implement and compare three information-theoretic attention policies, each of which having distinctive properties suitable for different situations. Intuitively, they are: 1) always prefer to watch an area with the highest activity detection confidence, 2) watch less on highly predictable areas and prefer to focus more on the areas which are likely to provide a higher amount of new information, and 3) the combination of these two policies for balanced approach. The attention policy formulations can be found in Section III-C. Figure 2 shows an overview of the proposed STARE system.

A. Visual Feature Extraction and Action Detection

The lower layer of our system computes visual features only for the selected window regions of an input video using dense trajectory features [46], motion boundary histogram [47], and histogram of optical flow and oriented gradients [48]. A supervised visual codebook (check also [49]) is learned from these visual features using an extremely clustered random forest [50]. The middle layer of the system computes a histogram of codewords obtained from sliding windows over time and classify actions with random forest [51]. For other approaches applicable at this stage see [49], [52]. At this step, we obtain a sequence of actions with their likelihoods.

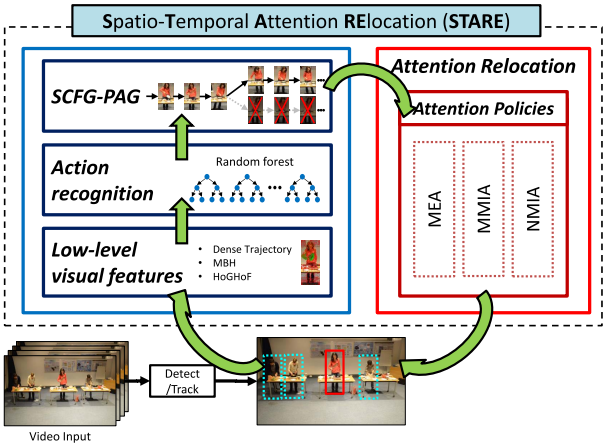


Fig. 2. Overview of the proposed STARE system.

B. Probabilistic Parsing of Activities

The higher layer of the system performs activity detection and predictive action generation by making use of the likelihoods computed by the action classifiers in the middle layer. The input is the distribution of action likelihoods, which is passed to the SCFG parser in a similar fashion as in [19]. We compute the next predicted action likelihoods after parsing each input symbol. This generated action prediction information is used in Section III-C. It is worth noting that we use a robust version of SCFG presented presented in [19] which takes into account the likelihoods of each input symbol, i.e. action likelihoods, in addition to the standard rule probabilities.

In this paper, we use the same notations of the SCFG framework used in [19] for consistency. An activity is represented as an SCFG composed of 4 conventional components, i.e. $G = \{R, T, N, S\}$: Production rules (R), Terminal symbols (T), Non-terminal symbols (N) and the Start symbol (S), a special non-terminal symbol. The production rules R are similar to the ones used in standard context-free grammars except that every rule is specified with a rule probability. Similarly, terminal symbols T are specified with probability values corresponding to action symbols in our case. The non-terminals N correspond to the abstraction of symbols.

A production rule states how a non-terminal symbol X should generate a set of symbols, which takes the form $X \rightarrow \lambda$, where $\lambda \in (N \cup T)^*$ and $P(X \rightarrow \lambda)$ states the probability of X generating λ. The order of the symbols in the right-hand side of the production rule enforces the actions to be occurred in the same order. The input is a time series of n-dimensional vectors where the n-th element of a vector contains the likelihood of the n-th action class produced by the corresponding action detector (sec. III-A).

A *state* of the parser is expressed as:

$$i : X \rightarrow \lambda . Y \mu \tag{1}$$

where $X \rightarrow \lambda Y \mu$ is a production rule defined in the grammar. '.' marker is the parser's current reading position in the rule and i is the position in the input stream, i.e. i-th observation. X and Y denote non-terminal symbols while λ denotes the parsed symbols in the context of X and μ denotes the expected terminal symbols as the next input.

For each input, the parser iteratively executes the three steps *scanning step*, *completion step* and *prediction step* to build the parse tree. 1) *Scanning step* reads a symbol from the input stream and matches it with the initial set of rules, generating a new set of states by increasing i index in the state definition (Eq. 1). 2) *completion step* updates the marker positions in all pending derivations. 3) *prediction step* hypothesizes the possible continuation of the input symbols. Please refer to [19] and [53] for more detailed explanation of each step as it is outside of the scope of this paper. Here, we explain the *prediction step* since it provides the information we use to predictively control attention. In the *prediction step*, the parser computes the next expected input symbols to the system:

$$\begin{cases} i : X_k \rightarrow \lambda . Y \mu [\alpha, \gamma] \\ Y \rightarrow \mu \end{cases} \Rightarrow \quad i : Y_i \rightarrow . \nu [\alpha', \gamma'] \tag{2}$$

$$\alpha' = \sum_{\forall \lambda, \mu} \alpha (i : X_k \rightarrow \lambda . Y \mu) P(Y \rightarrow \nu),$$

$$\gamma' = P(Y \rightarrow \nu)$$

where $\Rightarrow$ denotes a transition between parser states when the grammar rule $Y \rightarrow \mu$ is applied. α is a forward probability that represents the probability of the parsed terminal symbols until the i-th index of the input stream, whereas γ is the inner probability, which represents the probability of a substring that starts at input index k and ends at i. ν denotes the possible continuation of input symbols at the current parsing step, i.e. the expected observation in the next time step.

Let j be a state of the parser, ν_0^j be the first possible continuation symbol of ν^j in Eq. 2 and $\alpha_i{}'(j)$ a forward probability of the hypothesis at the i-th index. Let o_t be an observation at time t. We can compute the most probable likelihood vector o_{t+1} given the past observations $o_{1,\ldots,t}$ and the grammar structure. For each symbol $s \in T$ the corresponding likelihood $o_{t+1}(s)$ can be computed by searching for the state j with the highest forward probability in $\mathcal{J}$ among those where s corresponds to ν_0^j:

$$o_{t+1}(s) = P(s|o_{1,\ldots,t}) = \frac{1}{\eta} \arg\max_{j \in \mathcal{J}} (\alpha_t'(j))|_{\nu_0^j = s}. \tag{3}$$

Here, η is the normalization factor so that $\sum_{s \in T} P(s|o_{1,\ldots,t}) = 1$. This is an approximation since the parse trees with very low likelihood are pruned instead of being expanded exhaustively to increase the parsing speed, as also done in [19], [54], and [55]. o_{t+1} will be used to characterize the expected observation distribution for a given grammar.

C. Information-Theoretic Attention Relocation

Computation of Information Scores: Let $\mathcal{A}^w$ be our stochastic variable that describes the activity taking place in a candidate area (window) w. $\mathcal{A}^w$ can be one of L activities, $A_1, A_2, \ldots, A_L$. Each kind of activity $A_l \in L$ corresponds to a different grammar. An observation o_t^w is the likelihood distribution of actions after making an observation, provided

by the middle layer (Sec. III-A), at time t. For simplicity, w will be omitted when there is no ambiguity and we will denote observations up to time t, $o_{1,\ldots,t}$, as simply $\mathbf{o}_t$. Let $\hat{o}_{t+1}$ be a random variable that denotes the predicted future observation in the next time step and $H(\mathcal{A}|\mathbf{o}_t)$ the *entropy* of current activity given past observations, *i.e.*

$$H(\mathcal{A}|\mathbf{o}_t) = -\sum_{A\in\mathcal{A}} P(A|\mathbf{o}_t) log\, P(A|\mathbf{o}_t) \tag{4}$$

where $P(A|\mathbf{o}_t)$ term can be obtained from the SCFG parser of grammar A after finishing the *completion step* by choosing the maximum forward probability computed so far. Then the *mutual information* between the current activity and the expected observation at time $t+1$ given the past observations is:

$$I(\mathcal{A}; \hat{o}_{t+1}|\mathbf{o}_t) = H(\mathcal{A}|\mathbf{o}_t) - H(\mathcal{A}|\hat{o}_{t+1}, \mathbf{o}_t). \tag{5}$$

which tells us how much the uncertainty of activities will change if we make an observation in the next time step.

In equation 3, we assumed we have only a single grammar. Since we have multiple grammars, one parser running for each grammar, we re-write the equation 3 as:

$$\hat{o}^A_{t+1}(s) = P(s|\mathbf{o}_t, A) = \frac{1}{\eta} \arg\max_{j\in\mathcal{J}} (\alpha'_t(j))|^{\mathcal{A}=A}_{v^j_0=s}. \tag{6}$$

The computation of $H(\mathcal{A}|\hat{o}_{t+1}, \mathbf{o}_t)$ requires $P(\mathcal{A}|\hat{o}_{t+1}, \mathbf{o}_t)$, which can be obtained by exploiting the internal parser states according to Eq. 6. For each activity grammar A, we obtain different predicted observations $\hat{o}^A_{t+1}$, i.e. a vector of symbol likelihoods. For each of these predicted observations, we advance each parser corresponding to a different grammar by feeding back $\hat{o}^A_{t+1}$ to obtain $P(\mathcal{A} = A|\hat{o}^A_{t+1}, \mathbf{o}_t)$. After computing $P(\mathcal{A} = A|\hat{o}^A_{t+1}, \mathbf{o}_t)$, we roll back the parser's state to the previous state before feeding $\hat{o}^A_{t+1}$ to get ready for the next input.

At time t an observation is made only on the attended object w_t. $\mathcal{A}^w$ will be updated by advancing the parser with the observation received at every time step. Let's denote an attended object at time t and t+1 as $\tilde{w}_t$ and $\tilde{w}_{t+1}$ respectively. For all other objects that are not going to be attended ($w \neq \tilde{w}_{t+1}$), observations are given as a uniform distribution over the features since there is no new information. This "dummy" observation can be understood as a "missing" data from the parser's point of view. Let $W_N = \{w_1, w_2 \ldots\}$ be a set of objects that were not attended and W_A be a set of attended objects ($W = W_A \bigcup W_N$). An object w that was not observed maintain the same activity distribution of the previous time step:

$$H(\mathcal{A}^w|\hat{o}^w_{t+1}, \mathbf{o}^w_t) = H(\mathcal{A}^w|\mathbf{o}^w_t) \quad \forall w \in W_N. \tag{7}$$

We now discuss three policies for selecting which object w to attend in the next time step.

1) Minimum Entropy Attention (MEA) Policy: A straightforward object selection policy could be to always select the object with the minimum expected entropy at time $t+1$:

$$\tilde{w}^{MEA}_{t+1} = \arg\min_{w\in W} H(\mathcal{A}^w|\hat{o}^w_{t+1}, \mathbf{o}^w_t, \tilde{w}_{t+1} = w). \tag{8}$$

This approach drives the system to always relocate attention by following an object that is most likely to have a known activity. Once the current object activity turns out to be reliable, the system will keep focusing on the current object. However, in scenarios where there are more than one object performing an activity, the system will fail to detect multiple activities.

2) Maximum Mutual Information Attention (MMIA) Policy: Instead of following an object with the minimum entropy, we try to minimize the overall expected entropy across all existing object in the scene. Thus, for each candidate attended object $w \in W$ at time t, we define a attention score $\mathcal{S}^w$ as a form of summation of entropies:

$$\begin{aligned}\mathcal{S}^w &= \sum_{v\in W} H(\mathcal{A}^v|\hat{o}^v_{t+1}, \mathbf{o}^v_t, \tilde{w}_t = w) \\ &= H(\mathcal{A}^w|\hat{o}^w_{t+1}, \mathbf{o}^w_t, \tilde{w}_{t+1} = w) + \sum_{v\in W_N} H(\mathcal{A}^v|\mathbf{o}^v_t).\end{aligned}$$

The optimal attention relocation policy after making an observation at time t is thus:

$$\tilde{w}^{MMIA}_{t+1} = \arg\min_{w\in W} \mathcal{S}^w.$$

Consider now two different object selection cases for the next time step. One is selecting w_j and the other is selecting w_k, with $j \neq k$. Using Eq. 7 and Eq. 5, the attention score difference between the two selections can be derived as follows:

$$\begin{aligned}&\mathcal{S}^{w_j} - \mathcal{S}^{w_k} \\ &= \Big(H(\mathcal{A}^{w_j}|\hat{o}^{w_j}_{t+1}, \mathbf{o}_t^{w_j}, \tilde{w}_{t+1} = w_j) + H(\mathcal{A}^{w_k}|\mathbf{o}_t^{w_k})\Big) \\ &\quad - \Big(H(\mathcal{A}^{w_j}|\mathbf{o}_t^{w_j}) + H(\mathcal{A}^{w_k}|\hat{o}^{w_k}_{t+1}, \mathbf{o}_t^{w_k}, \tilde{w}_{t+1} = w_k)\Big) \\ &= I(\mathcal{A}^{w_k}; \hat{o}^{w_k}_{t+1}|\mathbf{o}_t^{w_k}) - I(\mathcal{A}^{w_j}; \hat{o}^{w_j}_{t+1}|\mathbf{o}_t^{w_j}).\end{aligned}$$

Thus, selecting an object w which makes the attention score minimum is equivalent to selecting w of the maximum mutual information between $\mathcal{A}^w$ and $\hat{o}^w_{t+1}$:

$$\tilde{w}^{MMIA}_{t+1} = \arg\min_{w\in W} \mathcal{S}^w = \arg\max_w I(\mathcal{A}^w; \hat{o}^w_{t+1}|\mathbf{o}^w_t).$$

3) Normalized Mutual Information Attention (NMIA) Policy: The previous two policies have distinctive contrasting behaviors. The MEA policy has a strong tendency towards an object with the lowest uncertainty, which can lead to higher accuracy but poor at detecting multiple activities occurring in parallel. The MMIA policy prefers objects that are less predictable. This is good for multiple activity detection while minimizing the accuracy loss. However, it may prefer an object that does not perform any activity, i.e. distractors, compared to other objects. Hence, we propose the third policy which aims to balance between the previous two. We adopt the normalized mutual information suggested in [56] and [57], which has the effect of reducing the tendency of getting stuck in the lowest uncertainty window as well as the effect of the distractors.

$$\tilde{w}^{NMIA}_{t+1} = \arg\max_w \left(\frac{I(\mathcal{A}^w; \hat{o}^w_{t+1}|\mathbf{o}^w_t)}{max(H(\mathcal{A}^w|\mathbf{o}^w_t), H(\hat{o}^w_{t+1}|\mathbf{o}^w_t))}\right).$$

IV. Evaluation and Analysis

In our evaluation, we focus on how the proposed STARE system performs on multiple activity tracks. The goal of the system is to detect as many activities as possible in various conditions. We assume that we use electronic Pan-Tilt-Zoom (ePTZ) cameras, where human object detection is already performed, as we are interested in the selection of human object using our activity-level attention system. For comparison, we conduct our experiments using classical scheduling policies such as random and round-robin, a bottom-up saliency model proposed by Itti *et al.* [38], and active attentional framework proposed by Chang *et al.* [42]. For [38], we obtain a saliency map using their method and add up the saliency values within each human bounding box, and select the human object with the maximum value. For [42], we apply the action likelihood value to be used as the detection mask score in their method. We adapt their sampling policy only, without their low-level and mid-level processing steps, to compare the performance of different attention policies.

A. Implementation Details

1) Visual Codebook: We use the low-level visual feature extraction methods reported in [46] to compute the dense trajectories of points over a fixed temporal length (15 frames) and the motion boundary histograms [47] and histograms of optical flow and oriented gradients [48] around the tracked points (32 neighborhood pixels, 2 spatial cells, 3 temporal cells). From the extracted descriptors in the training set, we compute a supervised 5000-dimensional visual codebook using an extremely clustered random forest [50] (5 trees, 1000 leaf nodes per tree) from 5000 samples per class.

2) Action Recognition: The histograms are computed using the codebook over a fixed temporal window size (60 frames) for training classifiers. We train a random forest classifier implemented in [51] (50 trees, 50 maximum depth) over the training histograms. The output of the classifier is a likelihood distribution of actions, which is fed into our high-level activity detector.

3) SCFG Parsing and Action Prediction: We use the SCFG parsing and action prediction method explained in Section III-B to compute the expected change in action and activity likelihoods. To speed up the SCFG parsing process, we prune parse trees having very low likelihoods. The structure of the activities are given in grammars which directly reflects the defined activity specifications, and the rule parameters for each grammar are computed from the relative occurrences of action symbol in each activity from the training set.

Probabilistic grammars have often been effectively used by manually defining simple grammars using prior task knowledge without excessive parameter tuning [19], [21]. We followed a similar approach in this work and defined the grammars in a straightforward manner to measure the robustness of the system with noisy input. Note, however, that activity grammars can be also learned from data [23], [58]–[60] when enough data are available. We have previously shown that SCFG can be constructed from data with limited hand labeling [20], [26].

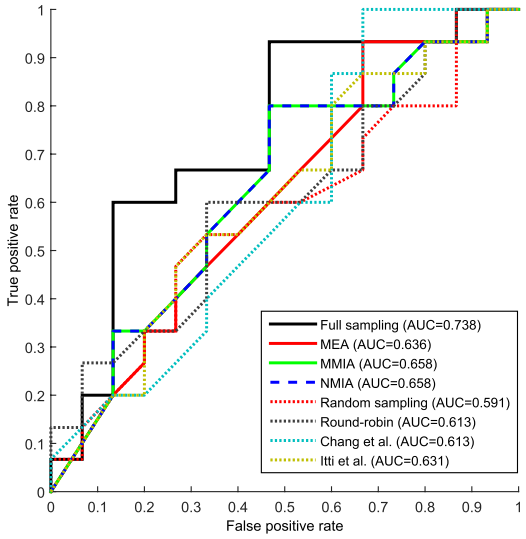


Fig. 3. VIRAT dataset. ROC curves obtained under different attention policies and respective area values. Policies MMIA and NMIA exhibit the same performance in this graph. Best viewed in color.

B. VIRAT Dataset

We test the STARE framework using a high-resolution VIRAT dataset [61]. We use the videos from "VIRAT_S_000001.mp4" to "VIRAT_S_000102.mp4" since they contain long-term temporally structured activities of two interesting scenarios: *Collection* and *Delivery*. There are 16 activities, 76 actions (excluding wandering), and 152 human objects in total. It comes with annotation which includes: *Load/Unload* an object (2 actions), *Open/Close* a car trunk (2 actions), *Get in/out* of a car (2 actions). In addition to these actions, we denote all standing/wandering movements between any two actions as "wander" action. Since the annotation is provided only at the action level, we define the temporal range activities that contain the sequence of these actions. At the end of every activity, Viterbi parsing is performed to compute the activity likelihoods, normalized by the number of observations. We show ROC curves in Fig. 3.

The *Delivery* activity is defined as: *Get out* of a car, *Open* a trunk, *Unload* an object, *Close* the trunk and *Get in* to the car, whereas *Collection* activity is the same as *Delivery* except it has *load* instead of *unload*. The temporal lengths of these activities allow enough time for our attention system to show effect on our visual system over long time period. It is important to note that in this dataset, an activity may not be carried out by a single person, e.g. one person unloading while another closing the trunk, with abundance of occlusions, which makes the dataset quite challenging.

C. Crêpe Dataset

The set-up we choose in this new dataset[1] is similar to that of a restaurant kitchen scenario, where several chefs and waiters/waitresses are preparing dishes for customers. The chefs are target objects and waiters/waitresses

[1]The Crêpe dataset can be obtained on request by contacting the first author.

TABLE II
TEMPORALLY STRUCTURED COOKING ACTION DESCRIPTIONS OF THE PROPOSED CRÊPE DATASET. EVEN THOUGH MANY ACTIONS ARE SHARED, THERE ARE CHARACTERISTIC ACTIONS AND THEIR ORDERS THAT CAN DISTINGUISH EACH RECIPE FROM OTHERS

Activity Class	Action-based Description
Lemon sugar	**Stir/Pour/Spread** mixture - **Flip - Pour** lemon juice - **Sprinkle** sugar - **Fold**
Banana chocolate	**Stir/Pour/Spread** mixture - **Cut** banana - **Flip - Transfer** banana - **Grate** chocolate - **Fold**
Cheese ham	**Stir/Pour/Spread** mixture - **Cut** ham - **Grate** cheese into bowl - **Flip - Transfer** cheese & ham - **Fold**
Cheese ham parsley	**Stir/Pour/Spread** mixture - **Cut** ham - **Grate** cheese into bowl - **Cut** parsley - **Flip - Transfer** cheese&ham - **Sprinkle** parsley - **Fold**
Goat cheese spinach	**Stir/Pour/Spread** mixture - **Cut** goat cheese - **Flip - Transfer** cheese and spinach - **Fold**
Goat cheese spinach nutmeg	**Stir/Pour/Spread** mixture - **Cut** goat cheese - **Flip - Transfer** cheese&spinach - **Sprinkle** nutmeg - **Fold**

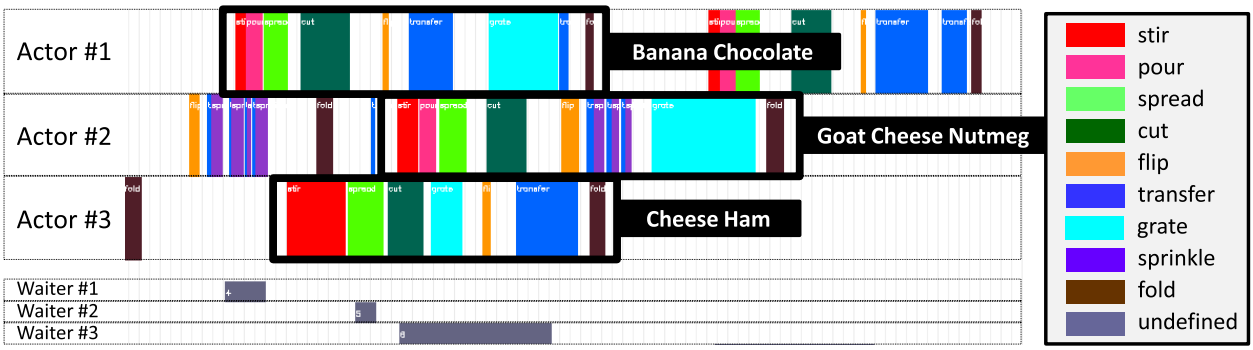


Fig. 4. An example scenario of the Crêpe dataset, where three chefs cook banana chocolate crêpe, goat cheese nutmeg crêpe and cheese ham crêpe, respectively. Three waiters are also in the background, who clean the table and bring plates to the chefs. The x-axis represents time.

are distractors. The chefs only cook, while waiters and waitresses enter and exit the scene at any time while taking away food, cleaning or resting. The dataset presents several structured activities that are composed of different short-term actions, which are shared among different activities.

The Table II shows the 6 different cooking recipes (activities) used in this dataset. 9 action classes are shown in bold font. In addition to training from these action classes, we train "undefined" action class using histogram features from the frames out of the ground-truth action boundaries. The dataset has 53 different long-term activities in total, where the first 28 activities are used for training. Each activity typically lasts between 3000-6000 frames and some activities may temporally overlap with each other. We define a "scene" as a various combination of these activities. We control the types of scenarios by selectively adding and removing the human objects. This allows us to have 93 different scenes for testing. Please refer to Fig. 4 for an example scenario of the dataset.

Figure 6 shows an example grammar we used for the activity Lemon sugar. (Refer to Table II for a full list of activity classes). The non-terminal symbols are defined in the same manner except for the noise symbol, which is added for robustness as similarly defined in [19] and [54]. The noise symbol can expand to any terminal symbol, corresponding to the uniform probability on all symbols. It allows a parser to accept any symbol that are inconsistent with the defined activity structure. For other non-terminals except S and N, the first row represents the probability of self-recursion. The second row is the symbol observation probability and the third row represents noise probability.

We test with two different scenarios in this experiment. The first scenario contains 50 different scenes where at least two distractors and any number of chefs are included. The second scenario contains 34 scenes where at least two chefs and any number of distractors are included.

Figure 5 shows the action classification accuracies, implemented using the method described in Section IV-A. Figure 5(a) shows the case where an undefined action, trained from randomly selected samples that do not belong to any of the defined classes, is included. We include this undefined class throughout our experiment. For reference purposes, we also include the result obtained using only the actions defined in Table II.

The Figure 7a shows the scenario containing at least two chefs with any number of distractors. This is the case where multiple known activities need to be detected. The NMIA policy shows better performance than full sampling due to noise in action classification. On the other hand, the Figure 7b shows a scenario containing at least two distractors with any number of chefs. This scenario contains actions that are not relevant to any specific activity, where attending at the correct information source (chefs) is critical.

It is important to note that in all of our policies except "Full sampling", we only attend one window at a time, which greatly reduces the amount of time required for low- and mid-level processes. As a trade off, it is expected that any system utilizing an attention policy will have a lower detection performance if there are many chefs (real activities). However, Figure 7a shows that this is not always the case as the NMIA system performs better than Full attention system based on Area Under Curve (AUC) scores (Full AUC=0.577 versus NMIA AUC=0.697).

It has been previously shown that a selective sampling does not necessarily deteriorate recognition/detection performance, e.g. [13], [42]. Sometimes it even improve the performance by

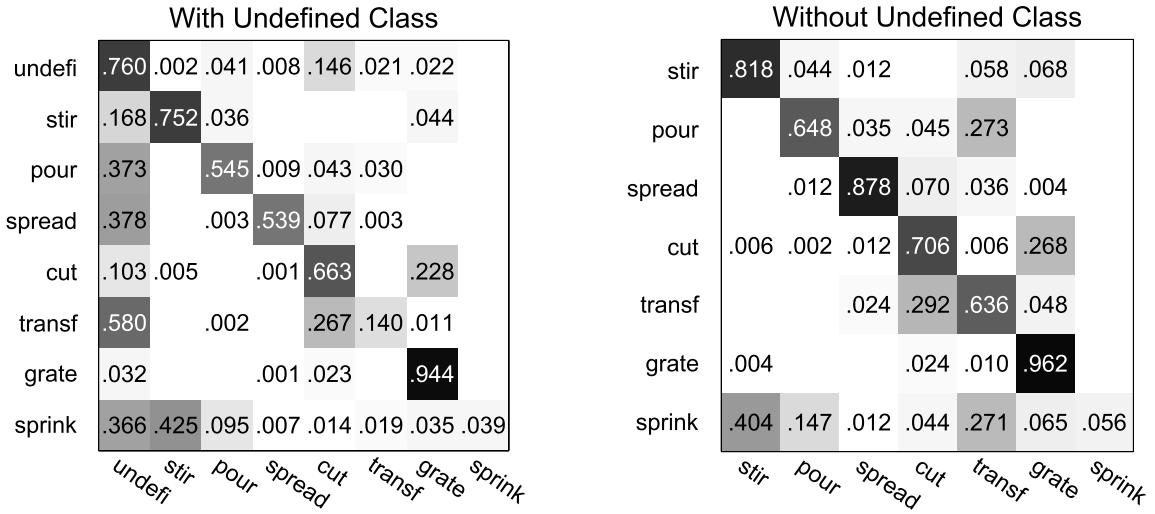


Fig. 5. Confusion matrices of recognized actions using the classification method described in Section IV-A. (a) Including an undefined action, which is trained from randomly selected samples that do not belong to any of the defined classes. In our experiment, we include this undefined class. (b) The result with the actions defined in Table II only, for reference purpose.

```
S  →  N I P D N F P R N   [1.0]
N  →  c   [ 0.125 ]
   |  p   [ 0.125 ]
   |  i   [ 0.125 ]
   |  d   [ 0.125 ]
   |  t   [ 0.125 ]
   |  r   [ 0.125 ]
   |  f   [ 0.125 ]
   |  g   [ 0.125 ]
I  →  I I   [ 0.75 ]
   |  i     [ 0.15 ]
   |  N     [ 0.10 ]
P  →  P P   [ 0.75 ]
   |  p     [ 0.15 ]
   |  N     [ 0.10 ]
(Other action non-terminals D, F, ... are defined similarly.)
```

Fig. 6. An example grammar for the activity *Lemon sugar* of Crêpe Dataset, which corresponds to the activity description defined in Table II.

TABLE III
AUC SCORES WITH ADDITIONAL NOISE CONDITIONS FOR A SCENARIO HAVING AT LEAST TWO CHEFS

Policy \ Noise	0%	5%	10%	15%	20%
Full	0.577	0.577	0.558	0.534	0.511
MEA	0.598	0.598	0.589	0.577	0.516
MMIA	0.589	0.589	0.589	0.554	0.516
NMIA	0.697	0.697	0.682	0.612	0.525

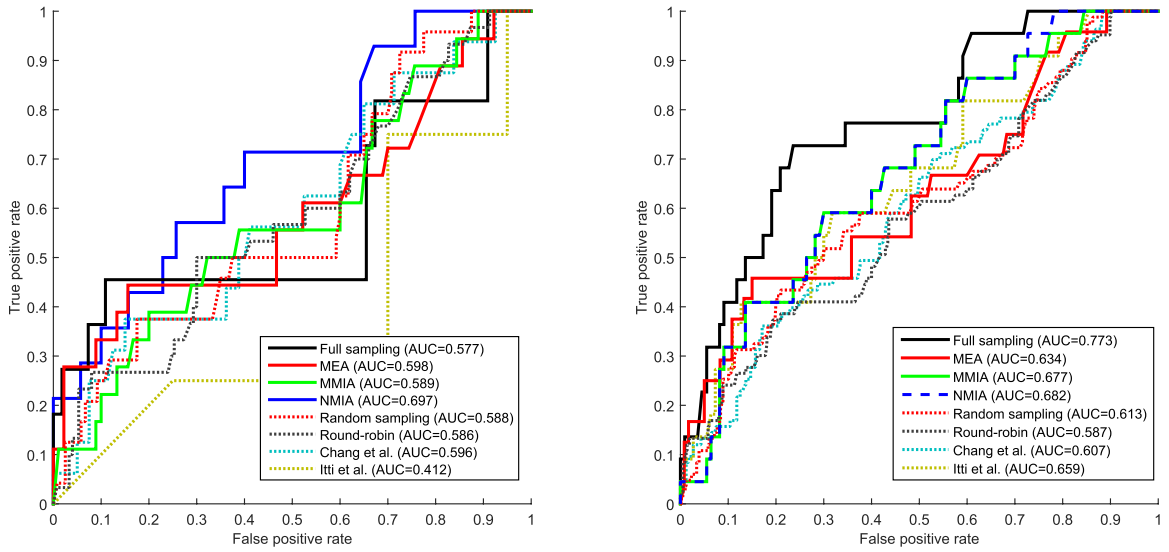
TABLE IV
AUC SCORES WITH ADDITIONAL NOISE CONDITIONS FOR A SCENARIO HAVING AT LEAST TWO DISTRACTORS

Policy \ Noise	0%	5%	10%	15%	20%
Full	0.773	0.764	0.764	0.751	0.697
MEA	0.634	0.625	0.625	0.610	0.577
MMIA	0.677	0.672	0.672	0.663	0.623
NMIA	0.682	0.667	0.667	0.658	0.623

filtering out noise [13], especially if the sampling strategy is well designed enough not to miss the valuable information of the problem. In [42], Figure 7, the selective sampling does not miss critical points, resulting in a lower error rate compared to the uniform sampling. Furthermore, in [13], Tables 1 and 2, the authors show that the recognition performance of selective sampling can be even higher than full sampling while maintaining shorter computational time especially when the action sequence is long and noisy.

A sample scenario is shown in Figure 8. The blue and red boxes show the currently attended window based on MMIA and MEA policies, respectively, among candidate windows. We show only MEA and MMIA policies to clearly demonstrate the comparison between these two contrasting policies. As explained in Section III-C, the MEA policy prefers a window with the lowest estimated score where MMIA policy prefers a window with the highest estimated score. In frame 5088, the chef in the middle (ID #0) is finishing a previous task by folding the crêpe thus having a higher MEA and lower MMIA scores, while the chef on the left (ID #1) is initiating a new task by pouring the mix, resulting in the opposite scores. In subsequent frames, ID #7 and ID #8 are waiter and waitress, respectively, who simply help chef by cleaning the table and bringing out the dishes. While MEA policy keeps focusing on ID #1 from frame 6614, it can be observed that the MMIA policy actively jumps among 3 windows.

Finally, we test the robustness of our proposed attention policies by adding a noise perturbation to the symbol observation likelihoods. We run the experiments again for each noise amount condition from 5% to 20% with a step size of 5% to track the changes of AUC score. As can be seen in Tables III and IV, the AUC scores do not change much until 10% noise perturbation and starts exhibiting some change at 15% mark. The cause of the small amount of changes would be due to the structural constraints defined in the grammars.



(a) (b)

Fig. 7. (a) A scenario containing at least two chefs, which is the case where multiple known activities need to be detected. The attention systems generally show better performances than the Full sampling system. (b) A scenario containing at least two distractors, where the distractors perform actions that are irrelevant to cooking. The detection performance is better in overall as there is usually a clear difference in activity likelihoods of distractors than that of chefs. Best viewed in color.

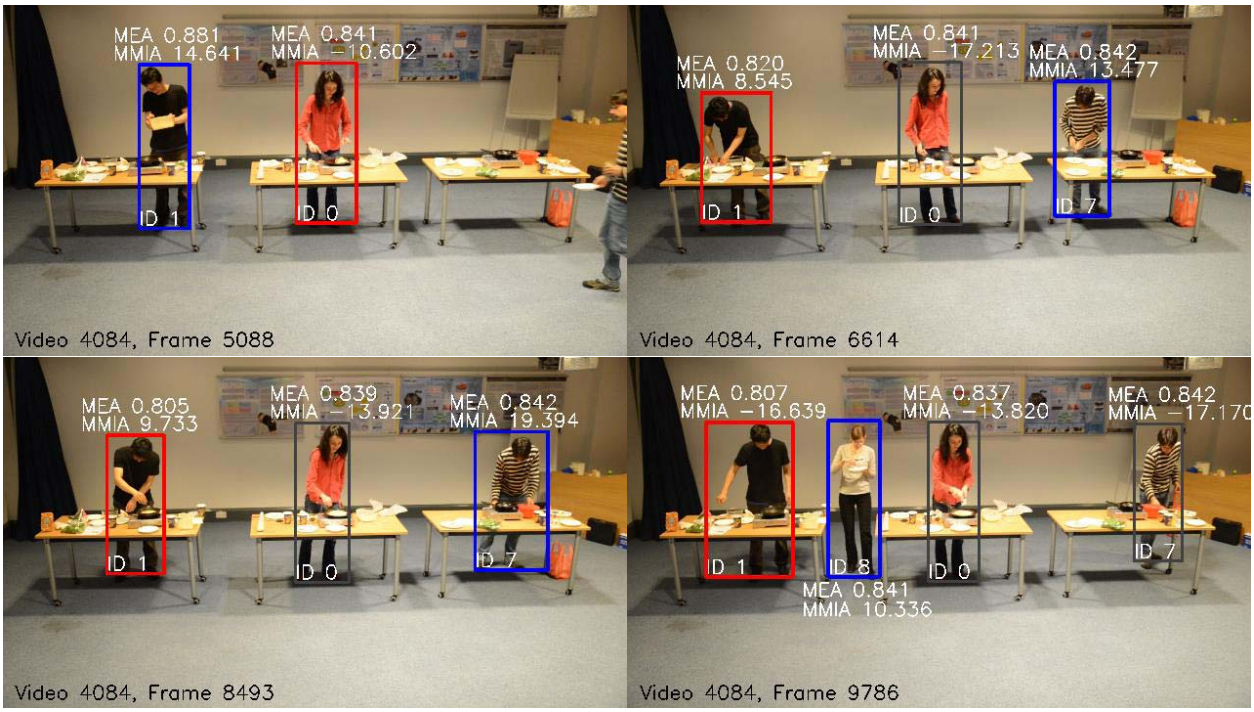


Fig. 8. A sample scenario. The blue and red box respectively shows the currently attended window based on MMIA and MEA policies among candidate windows. MEA policy chooses a window with the lowest score (lowest uncertainty) whereas MMIA policy chooses a window with tee highest score (highest information gain). In frame 5088, the chef in the middle (ID #0) is finishing a previous task by folding the crêpe thus having a higher MEA and lower MMIA scores, while the chef on the left (ID #1) is initiating a new task by pouring the mix, resulting in the opposite scores. In subsequent frames, ID #7 and ID #8 are distractors who simply helps chef by cleaning the table and bring out the dish. While MEA policy keeps focusing on #1 from frame 6614, MMIA policy actively jumps among 3 windows. Best viewed in color.

However, it can be observed that there is a drastic change from the 20% mark.

D. Computational Cost

We used an intel i7-2GHz, 12GB RAM, Ubuntu 12.10 laptop. The main code which implements various attention policies was programmed in Python, while the SCFG parser was programmed in C++ compiled with GNU C++. To interface between the main code and the parser, we implemented a Python wrapper to directly access all the parser functionalities. The low-level feature extraction library [46] and the extremely clustered random forest [50] for generating codebook were

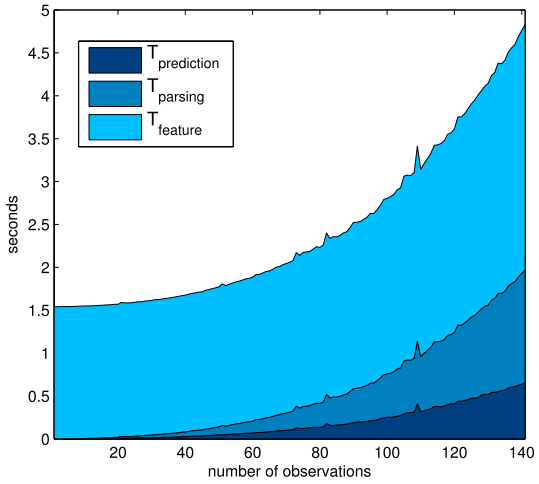


Fig. 9. Computational time overhead analysis. The different areas show the amount of time spent for each process layer. $T_{prediction}$ is the overhead caused by using our predictive parsing method, $T_{parsing}$ is the parsing time required to compute the likelihood of an activity and $T_{feature}$ is the average time required to extract spatio-temporal features per frame. The sampling interval between observations is 60.

both implemented in C++. The random forest [51] we used for detecting actions were implemented in C++ with Python bindings.

The majority of the time consumed while processing videos was in the low-level visual feature extraction part. The average processing time taken to compute a descriptor of a visual feature point was 20ms. On average, 75 descriptors were computed from each boundary window of a person per frame, resulting in 1.54 seconds on average per frame. The histogram computation and action classification using random forest was negligible.

A computational time overhead is shown in Figure 9. The different areas show the time spent for each process layer. $T_{prediction}$ is the overhead caused by using our predictive parsing method, $T_{parsing}$ is the parsing time required to compute the likelihood of an activity and $T_{feature}$ is the average time required to extract visual features per frame. Although the complexity of the SCFG parsing also increases over time due to the increased number of parsing hypotheses, we found that the amount of time required for parsing is reasonable because the parsing is performed every 2 seconds (60 frames) and the overall parsing and action prediction time was less than 1 second even after passing the 6000th frame on a consumer-grade computer.

It is worth mentioning that except for the Full attention method, all other attention methods attend only a single window at a time, resulting in nearly identical processing time of $T_{prediction}$, $T_{parsing}$ and $T_{feature}$ regardless of which sampling strategy used. If we assume that there are N candidate windows on average in a given scene and the Full attention policy requires $T_{feature}$ for feature computation, then all other attention methods have the feature computation time of only $T_{feature}/N$.

V. Conclusions & Future Work

We presented the Spatio-Temporal Attention Relocation (STARE) system, which is capable to dynamically allocate an attention to efficiently detect activities under resource-bounded condition. To realize our purpose, we presented an activity detection method based on SCFG capable of generating predictions of possible actions while recognizing input streams by considering the structural information of activities. We also proposed three attention policies calculated from the amount of information contained in an action. Each of the policies showed a characteristic performance for different scene situations. For evaluation, we presented a new structured activity dataset of concurrent multiple human objects of high resolution video. Through the whole evaluation the STARE achieved the comparable activity detection performance while consuming a relatively much lower computational load.

As shown in Section IV-D, the additional computational complexity required for computing an attention policy is reasonable. Although the system has been only tested in offline due to the high complexity in low-level processing, it would not be impossible to run in near-real time by adopting parallel processing paradigm in both visual feature extraction and SCFG parsing. As another extension, we plan to study a unified framework of the three attentional policies which can dynamically change the best policy depending on the scenario progression. Also, instead of attending only a single window at a time, the number of windows the system can attend at a time can be changed depending on the amount of resources available. Furthermore, a bottom-up attention approach can be combined together to improve the detection performance by incorporating low-level saliency features and spatial predictions.

Acknowledgment

The authors acknowledge Miguel Sarabia del Castillo for improving the efficiency of the original SCFG parser used in [19].

References

[1] A. Andreopoulos and J. K. Tsotsos, "A theory of active object localization," in *Proc. IEEE ICCV*, Sep./Oct. 2009, pp. 903–910.
[2] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.
[3] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, Jul. 2002.
[4] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order boltzmann machine," in *Proc. NIPS*, 2010, pp. 1243–1251.
[5] J. Denzler and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 145–157, Feb. 2002.
[6] E. Sommerlade and I. Reid, "Probabilistic surveillance with multiple active cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 440–445.
[7] C. Micheloni, B. Rinner, and G. L. Foresti, "Video analysis in pan-tilt-zoom camera networks," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 78–90, Sep. 2010.
[8] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Proc. IEEE ICCV*, Dec. 2013, pp. 1976–1983.
[9] S. Vijayanarasimhan and A. Kapoor, "Visual recognition and detection under bounded computational resources," in *Proc. IEEE CVPR*, Jun. 2010, pp. 1006–1013.
[10] D. Ognibene and G. Baldassarre, "Ecological active vision: Four bioinspired principles to integrate bottom–up and adaptive top–down attention tested with a simple camera-arm robot," *IEEE Trans. Auton. Mental Develop.*, vol. 7, no. 1, pp. 3–25, Mar. 2015.

[11] L. Paletta, G. Fritz, and C. Seifert, "Cascaded sequential attention for object recognition with informative local descriptors and q-learning of grouping strategies," in *Proc. IEEE CVPR Workshops*. Los Alamitos, CA, USA, 2005, p. 94.

[12] N. Oliver and E. Horvitz, "Selective perception policies for guiding sensing and computation in multimodal systems: A comparative analysis," *Comput. Vis. Image Understand.*, vol. 100, nos. 1–2, pp. 198–224, 2005.

[13] H. J. Chang, J. Kim, J. Cho, S. Oh, K. M. Yi, and J. Y. Choi, "Action chart: A representation for efficient recognition of complex activity," in *Proc. BMVC*, 2013, pp. 81.1–81.12.

[14] D. Ognibene and Y. Demiris, "Towards active event recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2495–2501.

[15] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal localization and categorization of human actions in unsegmented image sequences," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1126–1140, Apr. 2011.

[16] X. Chang, W.-S. Zheng, and J. Zhang, "Learning person–person interaction in collective activity recognition," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1905–1918, Jun. 2015.

[17] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, 2011, Art. ID 16.

[18] M. S. Ryoo and J. K. Aggarwal, "Stochastic representation and recognition of high-level group activities," *Int. J. Comput. Vis.*, vol. 93, no. 2, pp. 183–200, 2010.

[19] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 852–872, Aug. 2000.

[20] K. Lee, Y. Su, T.-K. Kim, and Y. Demiris, "A syntactic approach to robot imitation learning using probabilistic activity grammars," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1323–1334, 2013.

[21] D. Moore and I. Essa, "Recognizing multitasked activities from video using stochastic context-free grammar," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Cambridge, MA, USA, 2002, pp. 770–776.

[22] D. Minnen, I. Essa, and T. Starner, "Expectation grammars: Leveraging high-level expectations for activity recognition," in *Proc. IEEE CVPR*, Jun. 2003, pp. II-626–II-632.

[23] K. M. Kitani, Y. Sato, and A. Sugimoto, "Recovering the basic structure of human activities from noisy video-based symbol strings," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 22, no. 8, pp. 1621–1646, 2008.

[24] K. Lee and Y. Demiris, "Towards incremental learning of task-dependent action sequences using probabilistic parsing," in *Proc. IEEE Int. Conf. Develop. Learn.*, vol. 2. Frankfurt, Germany, Aug. 2011, pp. 1–6.

[25] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1346–1353.

[26] K. Lee, T.-K. Kim, and Y. Demiris, "Learning action symbols for hierarchical grammar induction," in *Proc. ICPR*, Tsukuba, Japan, 2012, pp. 3778–3782.

[27] K. Lee, T.-K. Kim, and Y. Demiris, "Learning reusable task components using hierarchical activity grammars with uncertainties," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 1994–1999.

[28] K. Friston and S. Kiebel, "Predictive coding under the free-energy principle," *Philos. Trans. Roy. Soc. B, Biol. Sci.*, vol. 364, no. 1521, pp. 1211–1221, 2009.

[29] D. Ognibene, Y. Wu, K. Lee, and Y. Demiris, "Hierarchies for embodied action perception," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*, G. Baldassarre and M. Mirolli, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 81–98.

[30] L. Zhang, Z. Zeng, and Q. Ji, "Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2401–2413, Sep. 2011.

[31] T. S. Lee and S. X. Yu, "An information-theoretic framework for understanding saccadic eye movements," in *Proc. NIPS*, 1999, pp. 834–840.

[32] G. Rotman, N. F. Troje, R. S. Johansson, and J. R. Flanagan, "Eye movements when observing predictable and unpredictable actions," *J. Neurophysiol.*, vol. 96, no. 3, pp. 1358–1369, 2006.

[33] D. Ognibene, E. Chinellato, M. Sarabia, and Y. Demiris, "Contextual action recognition and target localization with an active allocation of attention on a humanoid robot," *Bioinspiration Biomimetics*, vol. 8, no. 3, p. 035002, 2013.

[34] K. Friston, R. A. Adams, L. Perrinet, and M. Breakspear, "Perceptions as hypotheses: Saccades as experiments," *Frontiers Psychol.*, vol. 3, no. 151, pp. 151–170, 2012.

[35] K. Friston, F. Rigoli, D. Ognibene, C. Mathys, T. Fitzgerald, and G. Pezzulo, "Active inference and epistemic value," *Cognit. Neurosci.*, vol. 6, no. 4, pp. 187–214, 2015.

[36] B. W. Tatler, M. M. Hayhoe, M. F. Land, and D. H. Ballard, "Eye guidance in natural vision: Reinterpreting salience," *J. Vis.*, vol. 11, no. 5, pp. 1–23, 2011.

[37] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes, "Information-seeking, curiosity, and attention: Computational and neural mechanisms," *Trends Cognit. Sci.*, vol. 17, no. 11, pp. 585–593, 2013.

[38] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[39] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," in *Proc. IEEE CVPR*, vol. 1. Jun. 2005, pp. 631–637.

[40] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," *Proc. SPIE*, vol. 5200, pp. 64–78, Aug. 2003.

[41] J. Denzler, M. Zobel, and H. Niemann, "Information theoretic focal length selection for real-time active 3D object tracking," in *Proc. IEEE ICCV*, Oct. 2003, pp. 400–407.

[42] H. J. Chang, H. Jeong, and J. Y. Choi, "Active attentional sampling for speed-up of background subtraction," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2088–2095.

[43] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vis. Res.*, vol. 45, no. 2, pp. 205–231, 2005.

[44] M. S. Ryoo and W. Yu, "One video is sufficient? Human activity recognition using active video composition," in *Proc. IEEE WMVC*, Jan. 2011, pp. 634–641.

[45] M. Hoai and F. De la Torre, "Max-margin early event detectors," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2863–2870.

[46] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE CVPR*, Jun. 2011, pp. 3169–3176.

[47] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. ECCV*, 2006, pp. 428–441.

[48] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.

[49] G. Yu, J. Yuan, and Z. Liu, "Action search by example using randomized visual vocabularies," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 377–390, Jan. 2013.

[50] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Proc. NIPS*, 2007, pp. 985–992.

[51] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.

[52] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, and C. Sun, "Action recognition using nonnegative action component representation and sparse basis selection," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 570–581, Feb. 2014.

[53] K. Lee, "A syntactic approach to robot learning of human tasks from demonstrations," Ph.D. dissertation, Dept. Elect. Electron. Eng., Imperial College London, London, U.K., 2013.

[54] A. F. Bobick and Y. A. Ivanov, "Action recognition using probabilistic parsing," in *Proc. IEEE CVPR*, Jun. 1998, pp. 196–202.

[55] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," *Comput. Linguistics*, vol. 21, no. 2, pp. 165–201, 1995.

[56] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, "Hierarchical clustering using mutual information," *EPL (Europhys. Lett.)*, vol. 70, no. 2, p. 278, 2005.

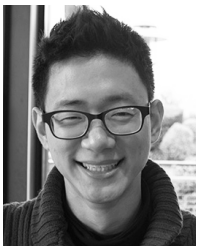
[57] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

[58] A. S. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," in *Dynamical Vision*. New York, NY, USA: Springer-Verlag, 2007, pp. 115–126.

[59] P. Langley and S. Stromsten, "Learning context-free grammars with a simplicity bias," in *Proc. Eur. Conf. Mach. Learn.*, vol. 1810. 2000, pp. 220–228.

[60] A. Stolcke and S. Omohundro, "Inducing probabilistic grammars by Bayesian model merging," in *Grammatical Inference and Applications*, vol. 862. New York, NY, USA: Springer-Verlag, 1994, pp. 106–118.

[61] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. IEEE CVPR*, Jun. 2011, pp. 3153–3160.

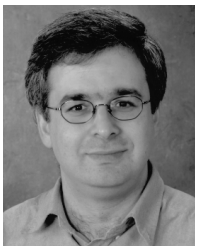


Kyuhwa Lee received the Ph.D. degree from the Electrical Engineering Department, Imperial College London. He is currently a Post-Doctoral Scientist with Campus Biotech, École Polytechnique Fédérale de Lausanne. His current research interests focus on brain–machine interfaces, vision-based shared control, biosignal analysis, pattern recognition, and temporal structure learning.

Dimitri Ognibene is a Marie Skłodowska-Curie COFUND Fellow at DTIC UPF, Spain, and is Visiting Researcher at the Centre for Robotics Research in King's College London. He obtained is Ph.D. in Robotics from the University of Genoa. He was previously Research Associate in Imperial College London and at Institute of Cognitive Science and Technologies of the Italian Research Council. Dr. Ognibene presented his work on active perception in social and dynamic environments and adaptation of attention in several international conferences on artificial intelligence and cognitive science and published on international peer-reviewed journals. Dr. Ognibene was invited to speak several symposiums as well as in prestigious neuroscience, robotics and machine-learning institutes. Dr. Ognibene is Associate Editor of *Paladyn*, *Journal of Behavioral Robotics*, and has been part of the Program Committee of several conferences and symposiums.

Hyung Jin Chang received the B.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. He is currently a Post-Doctoral Researcher with the Department of Electrical and Electronic Engineering, Imperial College London. His current research interests include articulated structure learning, human–robot interaction, object tracking, human action understanding, and user modeling.

Tae-Kyun Kim received the Ph.D. degree from the University of Cambridge, in 2007. He was a Research Fellow with Sidney Sussex College, Cambridge, from 2007 to 2010. He has been a Lecturer in the computer vision and learning with Imperial College London, U.K., since 2010. He has co-authored over 40 journal and conference papers, six MPEG7 standard documents, and 17 international patents. His co-authored algorithm is an international standard of MPEG-7 ISO/IEC for face image retrieval. His research interests span various topics, including object recognition, tracking, face recognition and surveillance, action/gesture recognition, and semantic image segmentation and reconstruction.



Yiannis Demiris is a Reader of Imperial College London. His research interests include assistive robotics, multirobot systems, robot–human interaction, and learning by demonstration. His research is funded by the U.K. Engineering and Physical Sciences Research Council, the Royal Society, BAE Systems, and the EU FP7 program through projects ALIZ-E and EFAA, both addressing novel machine learning approaches to human–robot interaction. He has guest edited special issues of the IEEE IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B specifically on Learning by Observation, Demonstration, and Imitation, and the *Adaptive Behavior Journal on Developmental Robotics*. He has organized six international workshops on Robot Learning, Bioinspired Machine Learning, Epigenetic Robotics, and Imitation in Animals and Artifacts, and was the Chair of the IEEE International Conference on Development and Learning in 2007 and the Program Chair of the ACM/IEEE International Conference on Human–Robot Interaction in 2008. He received the Rector's Award for Teaching Excellence, and the Faculty of Engineering Award for Excellence in Engineering Education in 2012. He is a member of the Institute of Engineering and Technology of Britain.