Surveillance Video Mining

Prithwijit Guha
Dept. of Electrical Engg.
IIT Kanpur, India
Email: pguha@iitk.ac.in

A. Biswas, A. Mukerjee, P. Sateesh
Dept. of Computer Sc. & Engg.
IIT Kanpur, India
Email: {arindam,psateesh,amit}@cse.iitk.ac.in

K.S. Venkatesh
Dept. of Electrical Engg.
IIT Kanpur, India
Email: venkats@iitk.ac.in

Abstract

We propose occlusion primitives to define a set of time-varying predicates on trackers for heterogeneous objects moving in unknown environments. Input pixel information is processed at inter-dependent levels to generate abstract categories for agents and actions. The scene background is learned online at the lowest layer, using feedback from the tracking level to robustly identify multiple agents. Agent shape and color features, status history and trajectories are clustered to discover categories for agents as well as their actions. Unlike existing surveillance systems, the proposed approach does not assume any prior model and aims at learning the scene/agent/event models from the acquired visual information. Results are demonstrated for traffic videos involving humans, vehicles and animals.

I. Introduction

Systems that attempt to learn conceptual categories from image schema can be constituted into two classes. First, *Vision-only systems* that use visual priors to construct models for object classes (e.g. [1], [2]) and second, *Multi-modal systems* that combine audio/text and video streams to match image categories with commentries/annotations [3], [4]. Vision-only systems construct models that require supervised input in the form of object priors. In the second class, co-occurrent language streams are associated with image features, often mediated by attentive focus [3]. However, there is considerable cognitive evidence for pre-linguistic conceptualization [5], [6], where perceptual inputs are categorized based on objects and object movements, and categories such as animals and vehicles are learned by five to six months of age ([7], p. 602).

In this work, we investigate this approach and attempt to learn agent and activity universals with neither supervisory visual input nor the parallel linguistic tokenization. The system is presumed to have the capacity for learning a background, updating objects that start moving or come to a stop against this background, tracking image blobs as they move across the scene, and taking note of merging and de-merging behavior with respect to other objects and scene fragments. The categorization takes place in a multi-layered learning system, with continuous sensory inputs mediating with higher level characterization in both forward and feedback modes. Thus, the raw image sequences (acquired with a static camera) are processed at inter-dependent layers where the information feedback and feed forward between different levels play a vital role in enhancing the adaptive learning efficiency of scene, agent and activity models. The temporal pixel intensity histogram and inter-frame motion information is computed at the lowest layer for simultaneous learning of pixel-wise background model and foreground blob extraction. The agents are characterized by their color distribution, occupied pixel set and trajectories and are initialized with the features computed from the foreground blobs at their very first instant of detection. These are tracked further in the image sequences with a motion based prediction initialized mean-shift tracker [8] in the next higher layer. The agent position information is then fed back to the lower layer for selective adaptation of the background model.

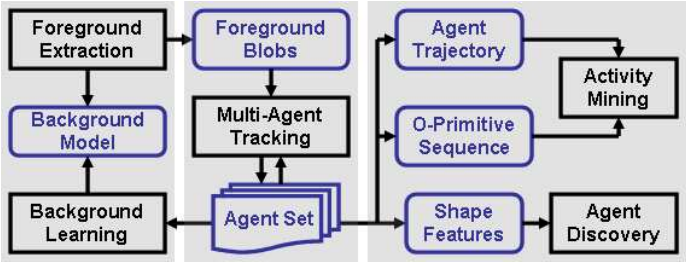


Fig. 1. The functional block diagram of the proposed system. Background modeling is performed in the lowest layer (left column) of processing with feedbacks from multi-agent tracking at the mid level. The agent discovery and activity mining are performed in the higher layer (right column) by processing agent features obtained in the mid layer.

At the next layer, we derive a set of occlusion primitives (O-primitives, henceforth) from the association measures of the agent pixels and the detected foreground blobs to identify them in one of the states of *isolation* (not connected to any other agents), *partial occlusion* (by background objects), *crowding* (with other agents resulting to partial or full occlusions), *disappearance* (due to track loss or complete occlusion by background objects), *entry* (the initial emergence of the agent) and *exit* (disappearing near the borders). Additionally, we identify the detection of *new agent(s) in the neighborhood* that aids in discovering certain classes of (homo)heterogeneous multi-agent interactions. These occlusion primitives are fed back to the tracking layer for selective updates of the agent features (e.g. color distribution and pixel set are updated only in the state of isolation). The agents are categorized in an unsupervised manner by learning mixture of Gaussians over the space of shape features. Results are demonstrated in successful discovery of man, tempo (a short distance public transport found in some Indian cities), cars, heavy vehicles (bus, truck, tractors), man on bike/cycle, rickshaw and animal (cow). The agent activities (actions and interactions with other agents) are generated by mining the occlusion primitive transition sequences. We demonstrate the results of successful discovery of people embarking/disembarking vehicles, agents crossing/overtaking each other from raw traffic surveillance videos.

This paper presents our work in the following manner. Section II explains the lower level processing for foreground blob extraction and simultaneous adaptive background learning with inter-frame motion and higher layer agent position information. Multi-agent tracking with O-primitive identification and selective feature updates are discussed in Section **??**. Section IV describes the process of unsupervised agent categorization by using incrementally learned mixture of Gaussians. The process of O-primitive transition sequence mining for action/interaction discovery is detailed in Section V. The results of multi-agent tracking, agent and activity discovery are presented in VI. Finally, we conclude in Section VII and outline the future extensions to the present work.

II. Background Modeling

Agents are identified as foreground regions based on one of two kinds of evidence: first, as regions of change with respect to a learned background model; and second, as regions exhibiting motion. Learning the background model in presence of agents is a challenging problem in itself. Several approaches have been proposed to incrementally learn the background scene model in the presence of agents. The most commonly adopted algorithms include the computation of median [1] or fitting (temporally evolving) Gaussian mixture models [9], [10] on the temporal pixel color histogram of the image sequence. These approaches continuously learn the multimodal mixture models with the assumption that the moving objects appear at a certain pixel only temporarily and the *true* background remains accessible to the system more frequently leading to higher weight of the corresponding mode. However, such an approach is prone to transient errors persisting over a number of frames (depending on the learning rate), resulting in two types of errors. First, if agents learned as part of the background suddenly start moving, ghosts and holes appear in the foreground segmentation. Second, when a moving agent comes to stasis, it is eventually learned as a part of the background, which may not be desirable in itself, and also in the transition period, objects interacting with it would not be identified. Both these problems are averted in the present approach by combining background-model and motion evidence, and updating based on tracking / previous motion-history feedback.

Generally, the background model $\mathcal{B}_t$ at the t^{th} instant is selectively updated based on the classification results of the t^{th} frame Ω_t. Classification based on $\mathcal{B}_{t-1}$ first results in a set of foreground pixels $\mathbf{F}_b(t) \subset \Omega_t$. Next, an inter-frame motion estimation [11] is performed between Ω_t and Ω_{t+1} to delineate the set of moving foreground pixels $\mathbf{F}_m(t) \subset \Omega_t$. This results in single-frame latency that helps us in identifying the regions that suddenly start moving or come to a stop.

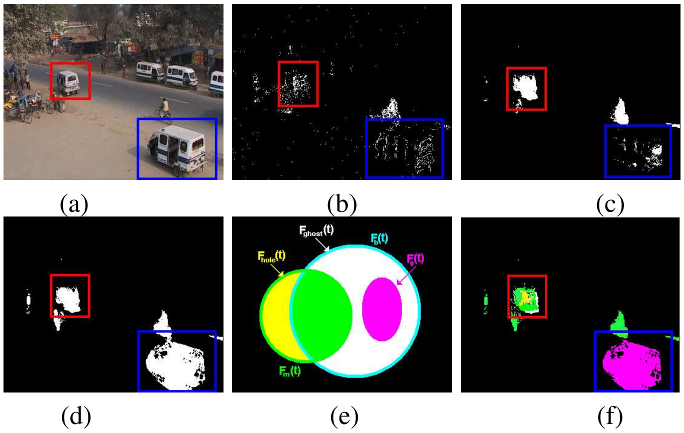


Fig. 2. Results of foreground detection. (a) The *tempo* in the red bounding box starts moving (Frame 1628), and the *tempo* highlighted by blue has already come to rest; Foreground extraction results (b) using only per pixel Gaussian mixture model with traditional exponential forgetting; (c) with motion detection only ($\mathbf{F}_m$) and (d) combining motion evidence with tracking feedback ($\mathbf{F}_t$); (e) Venn diagram indicating sets $\mathbf{F}_b(t)$, $\mathbf{F}_m(t)$, $\mathbf{F}_{hole}(t)$, $\mathbf{F}_{ghost}(t)$ and $\mathbf{F}_0(t)$; (f) Final pixel classification: moving regions (yellow and green), holes (yellow), ghosts (white) and agents at rest (pink).

Pixels identified as both foreground and moving are clearly identified as agent pixels. Among the mismatched pixels, moving pixels not identified as foreground, are denoted as $\mathbf{F}_{hole}(t) = \mathbf{F}_m(t) - \mathbf{F}_b(t)$. On the other hand, the set of non-moving pixels in $\mathbf{F}_b(t)$, is given by $\mathbf{F}_{ghost}(t) = \mathbf{F}_b(t) - \mathbf{F}_m$ and is identified as a possible background candidate. However, these non-moving ghost pixels may contain actual agent regions which have not shown up in the optical flow, or where an agent has actually come to a stasis. Using information from the motion history and tracking (discussed in Section III) we delineate the set of agents pixels that are known to have come to rest, $\mathbf{F}_0(t) \subset \mathbf{F}_{ghost}(t)$. The set of agent pixels that emerge from this analysis is defined as $\mathbf{F}(t) = (\mathbf{F}_b(t) - \mathbf{F}_{ghost}(t)) \cup \mathbf{F}_{hole}(t) \cup \mathbf{F}_0(t)$. Now, the complement of $\mathbf{F}(t)$ is used to update the background model to $\mathcal{B}_t$. The set of detected foreground pixels $\mathbf{F}(t)$ is further subjected to shadow removal (based on the criteria of equality among sub-unity intensity modulations in the 3 color chanels), neighbourhood voting, followed by connected component analysis to obtain the set of disjoint foreground blobs $\mathcal{F}(t) = \{F_i(t)\}_{i=1}^{n_t}$. These blobs constitute the basic units (putative agents) that are tracked over the entire sequence, and it is their participation in occlusion that results in O-primtive identification, and eventually in the activity discovery.

III. Multi-Agent Tracking

Here we adopt the multi-agent tracking algorithm proposed in [12], which works by associating the foreground blobs at time t, $F_i(t)$ with the predicted agent regions. The same foreground (agent) pixel being claimed by more than one agent (foreground blob) is one of the primary indicators of occlusion. We define several elementary occlusion behaviors according to the **Persistence Hypothesis**: Objects continue to exist even when hidden from view. The agent-blob association is performed over an *active* set $\mathcal{S}_A(t)$ containing agents tracked till the t^{th} instant and also a set $\mathcal{S}_{lost}(t-1)$ of agents which have disappeared within the viewing window. The system initializes itself with empty sets and the agents are (removed) added as they (dis)appear in the field of view. The proposed approach is a two stage process. Initially, the agents in $\mathcal{S}_A(t-1)$ are localized in the current frame Ω_t. This is followed by the identification of O-primitives by the process of agent-blob association with selective updates of agent features.

The j^{th} agent $\mathcal{A}_j(t)$ is characterized by its supporting region $a_j(t)$ (the set of pixels it occupies), the color distribution $h_j(t)$ (weighted by the Epanechnikov kernel [8] supported over the minimum bounding ellipse of $a_j(t)$) and the finite length position history of the centers $\{c_j(t-t')\}_{t'=0}^{\tau-1}$ of the minimum bounding rectangle of $a_j(t)$. The agent features are initially learned from the foreground blob extracted at its very first appearance and are updated throughout the sequence whenever it is in isolation. An estimate $c_j^{(0)}(t)$ is obtained by extrapolating from the trajectory $\{c_j(t-t')\}_{t'=1}^{\tau}$. The mean-shift iterations [8], initialized at an elliptic region centered at $c_j^{(0)}(t)$ further localize the agent region at $a_j(t) \in \Omega_t$.

The extent of association between a predicted agent region $a_j(t)$ for an agent in $\mathcal{S}_A(t-1) = \{\mathcal{A}_j(t-1)\}_{j=1}^{m_t-1}$ and the foreground blob $F_i(t) \in \mathcal{F}(t)$ is estimated by constructing a thresholded *localization confidence matrix* $\Theta_{AF}(t)$ and the *attribution confidence matrix* $\Psi_{FA}(t)$. These confidences are computed by a fractional overlap measure $\gamma(\omega_1, \omega_2) = \frac{|\omega_1 \cap \omega_2|}{|\omega_1|}$ signifying the fraction of the region ω_1 overlapped with ω_2.

$$\Theta_{AF}[j,i](t) = \begin{cases} 1; & \gamma(a_j(t), F_i(t)) \geq \eta_A \\ 0; & \text{Otherwise} \end{cases} \tag{1}$$

$$\Psi_{FA}[i,j](t) = \begin{cases} 1; & \gamma(F_i(t), a_j(t)) \geq \eta_F \\ 0; & \text{Otherwise} \end{cases} \tag{2}$$

Where the thresholds η_A and η_F signify the extent of allowable localization and attribution confidences. The number of foreground regions attributed to the j^{th} agent ($\Theta_A[j](t) = \sum_{i=1}^{n_t} \Theta_{AF}[j,i](t)$ and $\Psi_A[j](t) = \sum_{i=1}^{n_t} \Psi_{FA}[i,j](t)$) and agents localized in $F_i(t)$ ($\Theta_F[i](t) = \sum_{j=1}^{m_t-1} \Theta_{AF}[j,i](t)$ and $\Psi_F[i](t) = \sum_{j=1}^{m_t-1} \Psi_{FA}[i,j](t)$) are further computed from these matrices to identify the occlusion primitives.

The j^{th} agent in $\mathcal{S}_A(t-1)$ is **isolated** or unoccluded ($\mathcal{O}(I)[j,t]$), if the localization confidence is significantly high and the associated foreground blob is not overlapped with other agents. However, when the agent **disappears** ($\mathcal{O}(D)[j,t]$) both localization and attribution confidences fall below η_A and η_F signifying very poor or no association of the agent to any foreground blob. In case of **partial occlusions** ($\mathcal{O}(P)[j,t]$), the attribution confidence of one or more foreground blobs to the j^{th} agent remains high, although the localization confidence falls significantly. On the other hand, while in a **crowd** ($\mathcal{O}(C)[j,t]$), the localization confidence of the j^{th} agent in the crowded blob (overlapped with more than one agent) remains high although the attribution confidence of that blob to the agent remains low. Thus the four Boolean predicates for these occlusion primitives can be constructed as follows.

$$\mathcal{O}(I)[j,t] = \exists i[\Theta_{AF}[j,i](t) = 1] \wedge [\Theta_F[i](t) = 1] \tag{3}$$

$$\mathcal{O}(D)[j,t] = [\Theta_A[j](t) = 0] \wedge [\Psi_A[j](t) = 0] \tag{4}$$

$$\begin{aligned} \mathcal{O}(P)[j,t] = {} & \forall i[\Psi_{FA}[i,j](t) = 1] \\ & \wedge [\Theta_F[i](t) = 1] \wedge [\Psi_A[j](t) \geq 1] \end{aligned} \tag{5}$$

$$\mathcal{O}(C)[j,t] = \exists i[\Theta_{AF}[j,i] = 1] \wedge [\Theta_F[i](t) > 1] \tag{6}$$

To obtain the current active set $\mathcal{S}_A(t)$, updates are applied to all of color, shape and trajectory of individual agents under $\mathcal{O}(I)$, but only to the trajectory of agents under $\mathcal{O}(P)$ and $\mathcal{O}(C)$. Agents under $\mathcal{O}(D)$ are moved from the active set to the putative set. This enables the system to remain updated with agent features while keeping track of them.

The **entry/reappearance** of an agent is attributed to the existence of a foreground blob $F_i(t)$ in the scene having no association with any agent from $\mathcal{S}_A(t-1)$ and is thus detected as $\mathcal{O}(N)_i(t) = [\Theta_F[i](t) = 0] \wedge [\Psi_F[i](t) = 0]$. The features of the new blob $F_i(t)$ are matched against those in $\mathcal{S}_{lost}(t-1)$ to search for the reappearance of agents. If a match is found, the agent is moved from $\mathcal{S}_{lost}(t-1)$ to $\mathcal{S}_A(t)$ and a *reappearance* ($\mathcal{O}(R)[j,t]$) is noted. Otherwise, a new agent is added to $\mathcal{S}_A(t)$ and the system detects an *entrance* ($\mathcal{O}(E)[j,t]$). Similarly, an agent is declared to **exit** the scene ($\mathcal{O}(X)[j,t]$), if its motion predicted region lies outside the image region and is thus removed from the active set.

IV. Unsupervised Agent Categorization

We characterize an agent by its weighted color distribution, occupied pixel set and trajectory. However, the instances of the same category can have significantly different color and motion features, thereby leaving the shape as a more reliable descriptor. The considerable change in shape of agents due to either local deformations or perspective distortions demand robust shape descriptors. Shape features used in visual computing fall in to three basic categories, viz. contour, region and skeleton based descriptors [13]. The choice of the proper shape feature is always a compromise between classification power and computational complexity. The VSAM system [14] has shown significant agent categorization performance in a supervised learning framework by using only three simple descriptors, viz. area, shape dispersion $\left(\frac{perimeter^2}{area}\right)$ and the apparent aspect ratio or the height to width ratio of the minimum bounding box of the agent.

This work on the other hand, approaches the problem in an unsupervised learning framework while employing the same shape features (as in VSAM system) and incrementally learns a Gaussian mixture model for each agent category. The shape features are extracted for the first agent logged in the active set in the multi-agent tracking stage, which initialize a mixture model for a particular category. The system keeps track of this agent through out its presence in the viewing domain and the shape features are computed from the different appearances of the agent in states of isolation to update the mixture model of its category. The shape features are computed for every agent in their first instantiation and are compared against the existing mixture models of different categories. The new agent is declared to be of a certain class, if a match is found. Otherwise, the mixture model of the agent is declared a new category, and the new category is initialized from the agent's shape features. We adopt the learning algorithm proposed by Zivcovic [**?**] to construct the Gaussian mixture model. The results of unsupervised agent categorization are further illustrated in Section VI.

V. Activity Discovery

Activities can be broadly classified into two different categories. First, *Single agent actions*, that are characterized by the trajectories in agent feature-time space (e.g. a car's trajectory in a traffic scenario or the pose sequence exhibited by a dancer). Second, *Agent-object interactions* involving two or more participants. These activities may involve actual contact (e.g. (dis)embarking a vehicle) or may involve interactions at a distance (e.g. following/chasing). In terms of image space, actual contacts are necessarily reflected in O-primitive structures, but non-contact situations do not necessarily characterized by non-overlap. More so, the agents participating in agent-object interactions may be either homogeneous (e.g. "car1 overtaking car2") or heterogeneous (e.g. "man entering the tempo").

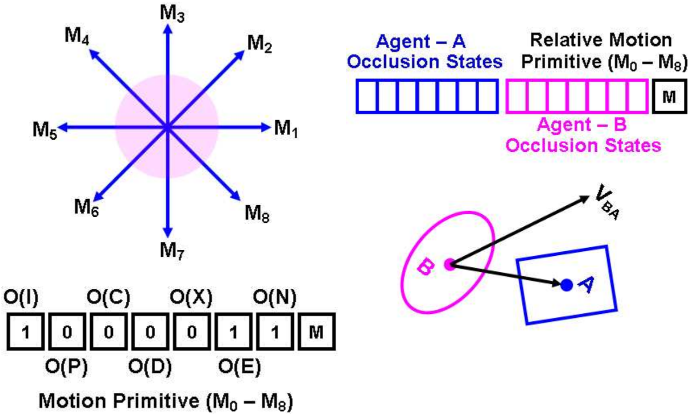
Both single-agent actions and agent-object interactions can be expressed as temporal sequences of agent states (actions) or co-occurrent states of interacting agents. Thus, the domain of activity analysis demands efficient statistical sequence modeling techniques for recognizing significant temporal patterns from the time-series data of action/interaction features. A number of methodologies employing hidden Markov models, time-delay neural networks, recurrent networks etc. have been proposed for modeling and recognition of action/interaction sequences in a supervised learning framework. On the other hand, unsupervised learning of activity patterns have also been proposed by trajectory clustering [15] or variable length Markov model learning [16]. A good overview of such techniques can be found in [17].

Supervised activity modeling techniques are mostly task oriented and hence fail to capture the corpus of events from the time-series data provided to the system. Unsupervised data mining algorithms, on the other hand, discover the modes of spatio-temporal patterns thereby leading to the identification of a larger class of events. The use of VLMMs in the domain of activity analysis was introduced for automatic modeling of the actions in exercise sequences [18] and interactions like handshaking [16] or overtaking of vehicles [19] in a traffic scenario. These approaches propose to perform a vector quantization over the agent feature and trajectory space to generate temporally indexed agent-state sequences from video data. These sequences are parsed further to learn VLMMs leading to the discovery of behavioral models of varying temporal durations.

Motion (pose) primitives derived from agent (state) trajectories are a necessary set of activity descriptors but are not sufficient as they lack the power to describe the interactions involving agent-region contacts in the image space. We, thus augment the activity feature space with the set of occlusion primitives which form a more fundamental notion of interaction signatures. More so, we identify that the occlusion state transition sequence form a more significant interaction description than the occlusion state sequences themselves. In this work, we aim to discover the interactions arising out of agents moving in complex environments undergoing both static and dynamic occlusions with background objects and other agents respectively. In the following subsections we discuss the methodologies adopted for sequence modeling and event primitive representations for interaction modeling.

A. Unsupervised Interaction Learning

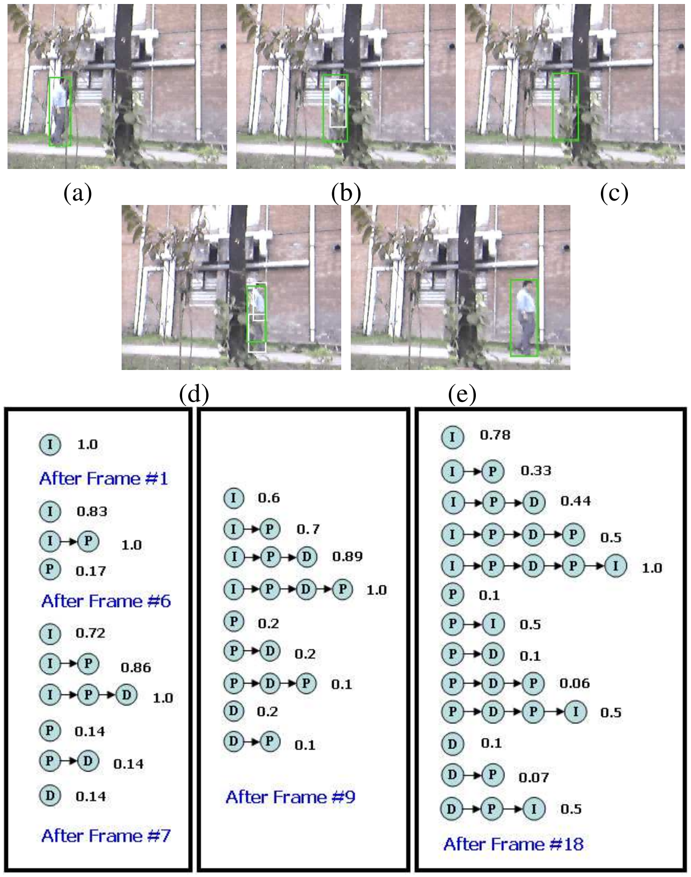
We construct event primitives for agents by combining their occlusion states and motion primitives. The occlusion states of *isolation* ($\mathcal{O}(I)$), *partial occlusion* ($\mathcal{O}(P)$), *crowded* ($\mathcal{O}(C)$), *disappeared* ($\mathcal{O}(D)$), *exit* ($\mathcal{O}(X)$), *entry* ($\mathcal{O}(E)$) and *entrance of new agent in neighborhood* ($\mathcal{O}(N)$) to form a 7-bit occlusion driven interaction descriptor. The direction of (relative) motion of the agent is quantized to assign one of the eight motion primitives $\mathcal{M}_1$ to $\mathcal{M}_8$ signifying the directions of *East*, *North-East*, *toSouth-East* (going anti-clockwise) respectively. Besides, a motion primitive $\mathcal{M}_0$ is used to signify the state of stasis of the agent. The final event descriptor for a single agent is formed by augmenting the occlusion and motion primitives as shown in figure 3(a).


M3
M4
M2
M5
M1
M6
M8
M7
O(I)
O(C)
O(X)
O(N)
1 0 0 0 0 1 1 M
O(P)
O(D)
O(E)
Motion Primitive (M_0 – M_8)
Agent – A Occlusion States
Relative Motion Primitive (M_0 – M_8)
M
Agent – B Occlusion States
V_{BA}
B
A

(a) (b)

Fig. 3. Event primitive descriptors. (a) Monadic Action Model: The seven occlusion primitives and a unary motion primitive from $\{\mathcal{M}_i\}_{i=0}^{8}$ are combined to obtain the single-agent atomic event descriptor. (b) Dyadic Action model: Attentive focus (query agent) is A. Agent B, if within the attentional window of A, is characterized by its motion relative to that of A. The two 7-vector occlusion primitives for A and B, together with this relative motion primitive, constitute the atomic interaction descriptor. Temporally ordered sequence of these descriptors are parsed to discover meaningful activity.

Consider a short video sequence where a person walks across a tree from left to right in the image space from which we sample 18 frames to illustrate the process of agent-background object interaction discovery. Key frames from this sequence are shown in figure 4(a)-(e). Incremental transition sequence learning is performed with a maximum depth of $L = 10$ and a learning rate η inversely proportional to the frame number. The growth of the activity tree is shown in figure 4(f).



(a) (b) (c)

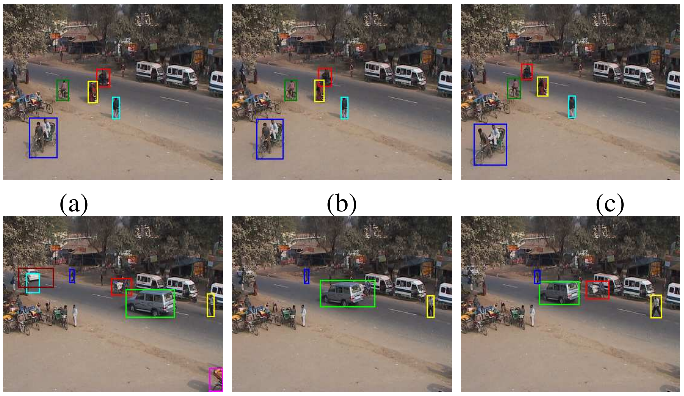
(d) (e)

(f)

Fig. 4. Example video sequence: Man walks left to right behind a tree. Frames and Agent states: (a) 1-5: *isolated*; (b) 6: *partially occluded*, (c) 7-8: *disappeared*, (d) 9: *partially occluded* (e) 10-18: *isolated*. (f) Learning **Activity Tree**. The left-most nodes are just below the root of the growing tree. Results of incremental transition sequence learning are shown after frames 1, 6, 7, 9 and 18. Branches encode different variable length event sequences along with relative frequencies. Thus, in column 2 (after Frame 9), the sequence $\{(I \rightarrow P \rightarrow D), 0.89\}$ corresponds to the event primitive sequence $\{(\mathcal{O}(I), \mathcal{M}_1) \rightarrow (\mathcal{O}(P), \mathcal{M}_1) \rightarrow (\mathcal{O}(D), \mathcal{M}_0)\}$; i.e. the event sequence "coming from the left and getting hidden" occurs with relative frequency 89% among observed 3-length sequences.

Semantic labels can be assigned to the sequences in the occlusion-primitive space to denote different activities, and subsequences may constitute sub-activities. For example, consider the longest path $\{(\mathcal{O}(I), \mathcal{M}_1) \rightarrow (\mathcal{O}(P), \mathcal{M}_1) \rightarrow (\mathcal{O}(D), \mathcal{M}_0) \rightarrow (\mathcal{O}(P), \mathcal{M}_1) \rightarrow (\mathcal{O}(I), \mathcal{M}_1)\}$ learned in the activity tree from the aforementioned video that correspond to the activity of **walking across a tree from left to right**. Subsequences of this path viz. $\{(\mathcal{O}(I), \mathcal{M}_1) \rightarrow (\mathcal{O}(P), \mathcal{M}_1) \rightarrow (\mathcal{O}(D), \mathcal{M}_0)\}$ and $\{(\mathcal{O}(D), \mathcal{M}_0) \rightarrow (\mathcal{O}(P), \mathcal{M}_1) \rightarrow (\mathcal{O}(I), \mathcal{M}_1)\}$ also correspond to the visually significant events of **going to hide from left to right** and **reappearing and moving to the right**.

We consider the agent B to be interacting with A, if the center of the minimum bounding box of the former lies within an attentional window of the later [19]. The interaction primitives are formed by combining the co-occurrent occlusion states of the interacting agents (taken two at a time) along with the motion primitive obtained from the relative velocity between the agents (figure 3(b)). The relative motion primitive is computed by quantizing the angle measured from the vector $\overline{BA}$ to the relative velocity (of B with respect to A) vector $\vec{V}_{BA}$ in an anti-clockwise direction. Figure 5 shows the results of discovering the interaction sequences of **overtaking** and **crossing** from a traffic video.



(a) (b) (c)

(d) (e) (f)

Fig. 5. **Overtaking sequence** (frame 853 − 868). (a)-(c) A *man on bike* (Agent B, marked by red bounding box) overtaking another *man on bike* (Agent A, marked by yellow bounding box) generating a sequence $\{(\mathcal{O}_A(I), \mathcal{O}_B(I), \mathcal{M}(4)) \rightarrow (\mathcal{O}_A(C), \mathcal{O}_B(C), \mathcal{M}(3)) \rightarrow (\mathcal{O}_A(I), \mathcal{O}_B(I), \mathcal{M}(2))\}$. **Crossing sequence** (frame 124 − 138). (d)-(f) A *car* (Agent A, marked by green bounding box) crossing a *rickshaw* (Agent B, marked by red bounding box) generating a sequence $\{(\mathcal{O}_A(I), \mathcal{O}_B(I), \mathcal{M}(1)) \rightarrow (\mathcal{O}_A(C), \mathcal{O}_B(C), \mathcal{M}(1)) \rightarrow (\mathcal{O}_A(I), \mathcal{O}_B(I), \mathcal{M}(1))\}$.

VI. Results

Experiments are performed on a traffic surveillance video of 30 minutes duration consisting of a wide variety of vehicles like bikes, rickshaw, cars, heavy vehicles etc. along with men and animals. The background modeling is performed by learning pixel-wise mixture of Gaussians over the RGB color space with a learning rate of $\alpha = 0.01$ and a diagonal covariance matrix $\Sigma_{init} = \{4.0\}$. The foreground extraction is performed with inter-frame motion information and selective model update with higher layer agent position feedback. Comparative results of foreground extraction are shown in figure 2, Section II.

Multiple agents in the traffic video are tracked with O-primitive identification. The tracking performance of the j^{th} agent at the t^{th} instant is evaluated by the fraction of the ground-truth region of the same $(G_j(t))$ overlapped with the region $a_j(t)$, localized by the proposed algorithm and is thus given by the quantity $\gamma(G_j(t), a_j(t))$. Hence, if there are $m_g(t)$ number of agents present in the ground-truth marked images at the t^{th}, instant, then the overall performance $\mathcal{P}$ for a video of T frames is given by,

$$\mathcal{P} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{m_g(t)}\sum_{j=1}^{m_g(t)}\gamma(G_j(t), a_j(t)) \qquad (7)$$

The above measure of overall performance $\mathcal{P}$ signifies the average fraction of the actual agent regions (or ground-truth regions) localized by the tracking algorithm in a certain video sequence. The overall performance varies, as the thresholds η_A and η_F are changed. It is evident from equations 1 and 2 that, as the thresholds η_A and η_F are increased, the detection rates of correspondences between predicted agent regions and foreground blobs reduce and thus the rate of track loss increases. On the other hand, too low values of these thresholds would increase the number of false detections of the O-primitives. Thus, to achieve optimal performances, we have chosen $\eta_A = \eta_F = 0.6$ and an overall tracking performance of approximately 68% was observed.

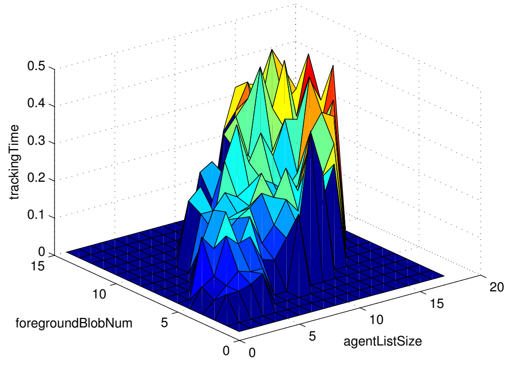


Fig. 6. Surface plot of tracking time (in seconds) with respect to number of active agents and foreground blobs

The tracking time largely depends on the number of agents in the active and putative set along with the number of foreground blobs. The variation of the tracking time with respect to these two factors is shown in figure 6. It is worth noting, that an estimate of the algorithm execution time with respect to crowding can also be obtained from this graph. The results of tracking in the traffic surveillance video are shown in figure 7.

The results of multi-agent tracking are logged into a database, where each agent is stored with its various appearances (learned only when isolated), image space trajectory and occlusion state sequence for its scene presence in the surveillance video. These constitute the surveillance logs from which the agent information can be retrieved with simple SQL queries. We assume the availability of object recognition modules that can categorize the agents based on their appearance features. A few samples from the surveillance logs (as seen in the HTML front-end) are shown in figure 8.

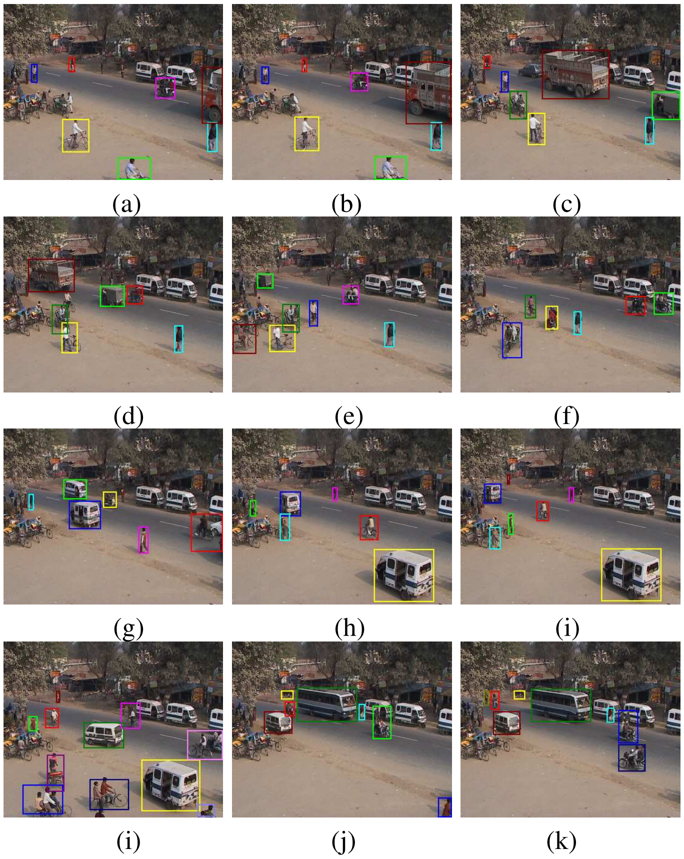


Fig. 7. (a)-(k) Results of tracking in the traffic surveillance video

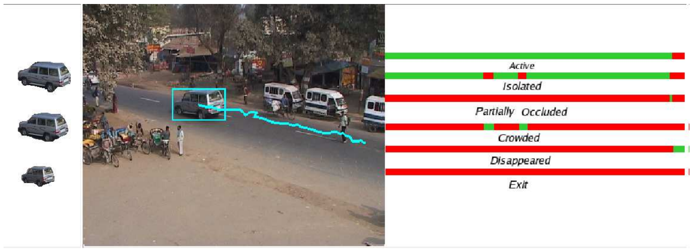
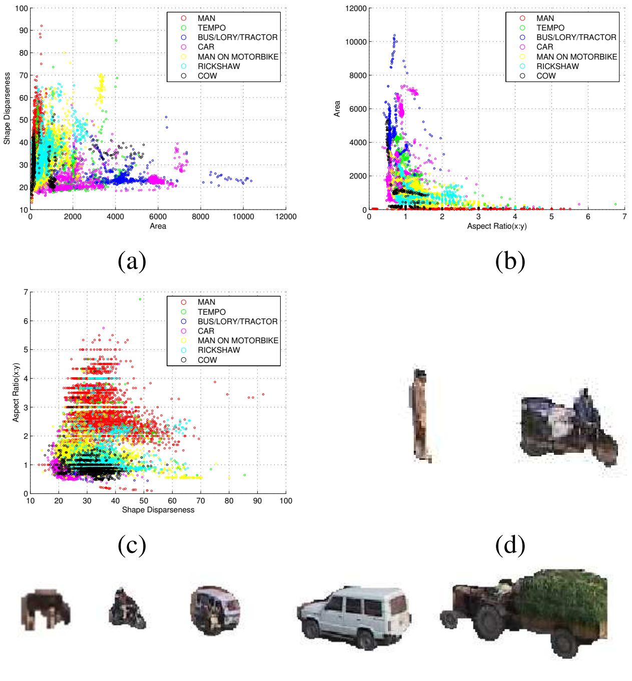


Fig. 8. Sample surveillance log for a car: agent appearances (left), trajectories (middle) and occlusion primitive timelines (right column). Occlusion primitives are shown (green) on each timeline.

We have adopted the GMM learning algorithm proposed by Zivcovic [10] for the purpose of unsupervised agent categorization. The mixture models were learned with a learning rate of $\alpha = 0.001$ and the initial co-variance matrix is assumed to be a diagonal matrix whose elements are computed from the data vector with a coefficient of variance of 0.05. The agent categorization is performed over the aforementioned traffic video, out of which we have discovered a total of 375 agents, categorized into 19 different classes, out of which only 7 categories corresponded to real world objects of significance. These are *man* (91 appearances), *man on motorbike* (73 appearances), *cars* (16 appearances), *heavy vehicles* (buses, trucks and tractors - 2 appearances detected properly), *tempo* (9 appearances), *rickshaw* (15 appearances) and *cow* (17 appearances). The other categories mostly consisted of outliers arising due to track losses and mostly the shapes learned from crowds. The projections of the scatter plot of the shape features of these significant agents in the *Area-Dispersion*, *Area-Aspect Ratio* and *Dispersion-Aspect Ratio* plane are shown in figure 9(a)-(c). The different appearances of the discovered agents (of significance) are shown in figure 9(d)-(e).



(a) (b)

(c) (d)

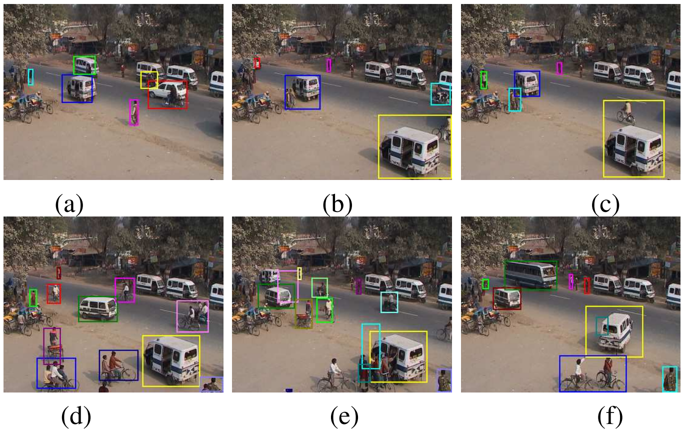
(e)

Fig. 9. Projection of shape feature scatter plot in (a) shape dispersion - area, (b) aspect ratio - area and (c) aspect ratio - shape dispersion planes. The appearances of discovered (d) *man* and *rickshaw* (e) *cow* , *man on bike* , *tempo* , *cars* and *heavy vehicle - tractor* (ordered left to right)

The activities are learned with a maximum depth of $L = 10$ and a learning rate of $\eta_t = max\left(\frac{1}{t}, 0.01\right)$ at the t^{th} instant. Activities are discovered for a particular query agent by mining its monadic and dyadic occlusion and motion primitive sequences. We have empirically chosen an attentional window of size 1.5 times of the minimum bounding box of the agent for all our experiments. In addition to overtaking and crossing, we have discovered the activities of **(dis)embarking** vehicles in the traffic video. The results of these interactions are shown in figure 10.

VII. Conclusion

This paper demonstrates the extent to which unsupervised activity discovery is possible merely by constructing sequences of occlusion events along with the image plane motion. Temporal sequences of O-primitives are posited as a powerful tool for identifying multi-agent interactions. The computation of occlusion is made possible by robust foreground extraction (even in the presence of gross occlusion), that enable us to track an agent across lengthy image sequences, the occlusion patterns during which are a surprisingly rich indicator of the activity involved.



(a) (b) (c)

(d) (e) (f)

Fig. 10. Results of Activity Discovery. (a-c) **Disembarking from Vehicle** (blue bounding box). (a) *Tempo* comes to stop (frame 1439); (b) Fragmentation due to people disembarking (frame 1594); (c) New Agents (people) formed in neighborhood of Tempo (frame 1624). (d-f) **Embarking on vehicle** (yellow bounding box). (d) *People* approaching *Tempo*, entering its attentional horizon (frame 1923); (e) people crowded with tempo (frame 2027); (f) people disappear, tempo still tracked (frame 2319)

The generality of occlusion as a phenomena that pervades all types of agent interactions clearly makes it an important area of study. To our knowledge, this is the first work to focus on this domain. Perhaps owing to the same reason, the child learner also quickly becomes sensitive to the presence of objects that are occluded from sight, and occlusion is perhaps the key perceptual indicator for fundamental spatial notions such as containment and contact. In addition, image-plane motions are indicative of other perceptually salient features such as path, source and goal, etc.

In future work, we plan to explore other low-level tools available for activity recognition. With qualitative information on camera calibration one can add detailed spatial characterizations for the motions - *translate left/right/towards/away*, *rotate*, *speed-up*, *halt*, etc. which can by themselves be informative for many actions.

Event predicates are characterized by the type-of-activity (modeled as a fine-grained image schema), the ordered set of agents participating in it, as well as optional characteristics such as time, place, manner etc. These arguments also emerge from the work, as the dimensions in the feature space where the events are discovered.

In this work, we have classified agents only by their shape and motion characteristics, but possibly a more important characterization is in terms of actions that an agent participates in (e.g. what objects participate in embark/disembark events?). This leads to a chicken-and-egg problem - one needs agents to recognize events, and actions to characterize agents. This will remain an important area for agent discovery for many years to come.

Based on these low-level categories, one can build up to higher level constructs based on several sources of additional information:

- *Multimodal Learning*: Given cotemporaneous linguistic descriptions, and given the event and agent characterization already at hand, it would be simple enough matter to build grounded models of the head verb and its noun subcategories.

- *Camera Calibration / Ground-Plane Assumption*: By using camera calibration data and making ground-plane assumptions for the agents in a given domain, considerable detail can be added to the event characterizations.
- *Shape and Scene priors*: While supervised agent and event characterization may be extremely useful, we would like to avoid this for some time since it limits the scalability of the approach.

In addition to these aspects, it would be important to extend the work to more general situations, e.g. cameras that can move (initially with pan-tilt motions), and for dynamic backgrounds (trees, fountains).

References

[1] I. Haritaoglu, D. Harwood, and L. Davis, "W4 : Real time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809–830, August 2000.

[2] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environments," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, July 2004, pp. 406–413.

[3] C. Yu and D. H. Ballard, "Learning to recognize human action sequences," *Proceedings IEEE International Conference on Development and Learning, ICDL02*, pp. 12–15, June 2002.

[4] D. K. Roy, "Learning visually grounded words and syntax for a scene description task," *Computer Speech Language*, vol. 16, no. 3-4, pp. 353–385, July-October 2002.

[5] J. Mandler, "How to build a baby: Ii. conceptual primitives," *Psychological Review*, vol. 99, no. 4, pp. 587–604, 1992.

[6] A. Karmiloff-Smith, *Beyond Modularity: A Developmental Perspective on Cognitive Science*. Cambridge, MA: MIT Press, 1992.

[7] P. representation and language, "Mandler, j.m." in *Language and Space*, P. e. a. Bloom, Ed. MIT Press, 1996, pp. 365–384.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transaction on Pattern Analysis Machine Intelligence*, vol. 25, no. 5, pp. 564–575, 2003.

[9] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, June 1999, pp. 246–252.

[10] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, 2004, pp. 28–31.

[11] M. Proesmans, L. V. Gool, E. Pauwels, and A. Osterlinck, "Determination of optical flow and its discontinuities using non-linear diffusion," in *The 3rd Eurpoean Conference on Computer Vision*, vol. 2, 1994, pp. 295–304.

[12] P. Guha, A. Mukerjee, and K. Venkatesh, "Efficient occlusion handling for multiple agent tracking with surveillance event primitives," in *The Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2005.

[13] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Computer Society, 2000, pp. 424–429.

[14] Collins, Lipton, Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto, and Hasegawa, "A system for video surveillance and monitoring: Vsam final report," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-00-12, May 2000.

[15] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proceedings of the 6th British conference on Machine vision (Vol. 2)*. BMVA Press, 1995, pp. 583–592.

[16] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length markov models of behavior," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 398–413, 2001.

[17] H. Buxton, "Learning and understanding dynamic scene activity: a review," *Image and Vision Computing*, vol. 21, no. 1, pp. 125–136, 2003.

[18] N. Johnson, A. Galata, and D. Hogg, "The acquisition and use of interaction behavior models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1998, pp. 866–871.

[19] A. Galata, A. G. Cohn, D. Magee, and D. Hogg, "Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models," in *Proceedings of European Conference on Artificial Intelligence*, F. van Harmelen, Ed., 2002, pp. 741–745.