Image and Vision Computing 31 (2013) 603–615

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Multi-agent event recognition by preservation of spatiotemporal relationships between probabilistic models ☆

S. Khokhar *, I. Saleemi, M. Shah

University of Central Florida, Orlando, FL 32816, USA

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 22 May 2013
Accepted 4 June 2013
Available online 11 June 2013

Keywords:
Football play recognition
Multi-agent activity modeling and recognition
Graph matching
Lie algebra

ABSTRACT

We present a new method for multi-agent activity analysis and recognition that uses low level motion features and exploits the inherent structure and recurrence of motion present in multi-agent activity scenarios. Our representation is inspired by the need to circumvent the difficult problem of tracking in multi-agent scenarios and the observation that for many visual multi-agent recognition tasks, the spatiotemporal description of events irrespective of agent identity is sufficient for activity classification.

We begin by learning generative models describing motion induced by individual actors or groups, which are considered to be agents. These models are Gaussian mixture distributions learned by linking clusters of optical flow to obtain contiguous regions of locally coherent motion. These possibly overlapping regions or segments, known as motion patterns are then used to analyze a scene by estimating their spatial and temporal relationships. The geometric transformations between two patterns are obtained by iteratively warping one pattern onto another, whereas the temporal relationships are obtained from their relative times of occurrence within videos. These motion segments and their spatio-temporal relationships are represented as a graph, where the nodes are the statistical distributions, and the edges have geometric transformations between motion patterns transformed to Lie space, as their attributes. Two activity instances are then compared by estimating the cost of attributed inexact graph matching. We demonstrate the application of our framework in the analysis of American football plays, a typical multi-agent activity. The performance analysis of our method shows that it is feasible and easily generalizable.

Published by Elsevier B.V.

1. Introduction

Recognition and analysis of multi-agent activity has been an important area of research in artificial intelligence [32] as well as computer vision [7]. A significant amount of effort in both these areas has attempted to leverage a symbolic representation of atomic behaviors [4], and first-order predicate calculus [1] as tools for analysis and understanding of complex activities. Although such principled approaches are desirable in general, they do not explicitly account for the difficulties and uncertainties in obtaining symbolic representations by visual analysis. Moreover, in practical scenarios, it is a prohibitively cumbersome task to manually encode semantically meaningful symbols, rules and productions etc., that would account for all possible permutations in large state spaces. We observe that visual recognition of multi-agent activities can be an unsupervised learning process where the goal is to estimate an appropriate measure of similarity between videos while taking into account the uncertainty in low level representation.

We propose a graph theoretic framework which encodes not only the statistical representation of low level, agent-specific actions or behaviors, but also comprehensive, continuous spatial and temporal relationships between such behaviors, as opposed to discrete ones like Allen algebra [1]. In terms of low level behaviors inference, current multi-agent activity analysis methods in computer vision rely on models based on tracking of agents [7,19], or body parts thereof [25], detections without tracking [28,2], or short high confidence tracklets [29,30], etc. In practical scenarios however, tracking is unreliable due to occlusion and unpredictable motion of actors, which is a significant drawback in many of the methods that employ tracking. These methods also do not explicitly model the inherent spatiotemporal structure present in multi-agent activities. Other methods can be found in a recent survey [24].

To recognize multi-agent activities, event-based methods are often used. An activity is assumed to be composed of a set of events and is characterized by the relationship of these events. Events were detected based on the interactions of agents in [11,31] and based on individual actions in [10,8,4]. Ivanov and Bobick [11] used probabilistic detectors to propose event candidates. The event set was analyzed

☆ This paper has been recommended for acceptance by Rama Chellappa.

* Corresponding author. Tel.: +1 407 962 5895.
E-mail addresses: skhokhar@eecs.ucf.edu (S. Khokhar), saleemi@eecs.ucf.edu (I. Saleemi), shah@eecs.ucf.edu (M. Shah).

0262-8856/$ – see front matter. Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.imavis.2013.06.004

with a context-free stochastic parser to recognize interactions between persons and vehicles. Hongeng et al. [8] and Hakeem and Shah [4] detected sub-events performed by individuals, and represented the logical and temporal dependencies between sub-events using graphic models. This kind of methods requires information about all possible events. It becomes quite difficult for a large number of agents and complex interactions between them.

One of the specific examples of multi-agent activities is field sports, e.g., football and soccer, analysis of which has been an active research topic [16,9]. These efforts attempted to detect or recognize dynamics and behaviors using camera motion, color, low-level motion, field markers, lines and texture etc. Intille and Bobick in [10] first detected individual goals. These goals and temporal relations between players served as children for a higher level in a Bayesian tree. Li et al. [20] proposed a discriminative temporal interaction manifold based framework for the same problem. Manually annotated player trajectories were used as low level features in both [10,20], while [19] obtained them using a multi-target tracker.

Swears and Hoogs [29,30] have proposed to use short, high confidence tracklets in a Non-Stationary Kernel Hidden Markov Model (NSKHMM) for football play recognition, in an attempt to overcome problems associated with tracking of agents throughout activity videos. Recognition of American football plays is a complex problem and much attention has been paid to it, not only because it is an important problem in itself but more so because it is an ideal example for multi-agent activity and is therefore a popular choice for demonstration of recognition algorithms. We observe that the methods most relevant to our approach are by Lin et al. [21,22], Li and Chellappa [18] and Swears and Hoogs [29,30]. We discuss these in more detail in subsequent sections.

1.1. Graphical representation of global motion and activity comparison

Individual events are spatio-temporally localized within a multi-agent activity. For example in a traffic scenario, a semantic scene description of an intersection may read as follows: motion from north west to south is followed by north bound motion from south east. Each of these traffic flows can be considered to be one event. For a computer vision algorithm this description would translate to the knowledge of the temporal and spatial beginning and end of each event and other features of these events such as their density and variance of direction and speed etc. Similarly we propose to use salient motion information in a sports scenario to understand sporting activity without requiring tracks for individual players. For example a football play is defined by the motion of certain players which is planned before the play is executed. The motion of the defensive players follows that of the offensive players in most cases. We therefore do not need to keep track of both sets of players, but only the significant activity within the play. Fig. 1 shows an example for a clip of a football play along with the expected and automatically generated sub-events occurring within the clip. In order to get a reasonable understanding of the global activity we propose to explicitly compute and leverage the spatial and temporal relationships between motion patterns within the video of an activity. As opposed to symbolic or quantified representations of individual behaviors, in this work we attempt to represent the entire activity as a *complete* graph, where a vertex represents an agent's behavior, and the edges between two vertices depict the spatiotemporal relationships between them. It is worth mentioning here that, first, due to the probabilistic nature of agent behavior representations, we do not assume the presence or absence of a predefined atomic action, rather any arbitrary action is possible with a certain probability, which in turn is used in the matching and recognition of activity instances. Secondly, the relationships between agent behaviors need not be quantized in space (e.g., above, below or adjacent etc.), or in time (e.g., before, after, during etc.). Once we obtain a graphical representation of an activity, we may compare two activities using inexact graph matching.

2. Single agent behavior: Motion representation

The purpose of our motion representation is to accurately capture the spatio-temporal location of motion in an event in addition to motion features. To this end we propose to use the motion patterns framework [27] described in detail in the following sub-section.

2.1. Agent behavior discovery and representation

The motion patterns framework was originally proposed for event modeling in static camera scenarios with persistent motion. The entire framework can be broken down into these steps: Optical flow computation [23], clustering of flow and a hierarchical linking of flow clusters based on spatial and temporal proximity. Each connected group of flow clusters is one motion pattern. A feature vector (an optical flow point) given as $\mathbf{x} = (x,y,u,v)$, belonging to the motion pattern of the ith agent, can then be written as:

$$\mathbf{x} \sim \sum_{k=1}^{N_i} \omega_{i,k} \mathcal{N}\left(\cdot \middle| \mu_{i,k}, \Sigma_{i,k}\right), \tag{1}$$

where $\mu_{i,k}$, $\Sigma_{i,k}$, and $\omega_{i,k}$ are the parameters of the kth component of the ith motion pattern, and there are a total of N_i, 4d Gaussian components in the mixture.

In the American football domain, a play is defined by the spatio-temporal relationships between subevents. These events may be defined as an observed motion in a certain direction with an associated time. Note that we do not distinguish between offense and defense players but instead make use of the fact that the motion of defense players largely depends on the motion of the offense players and their chosen strategy. Therefore motion of offense and defense players who are partaking in the same subevent is jointly modeled. We demonstrate that the motion pattern representation is ideally suited to modeling events in non-persistent settings such as those in sports where related methods fail.

We begin by performing some pre-processing on the videos. This involves ego-motion compensation so that the observed motion is strictly related to the motion of actors rather than the camera. The video of an activity instance, f, is divided into Z_f video clips, each of which is z frames long. We then compute optical flow for the videos which results in a feature vector (x, y, u, v, t) for each pixel in a frame, where t is quantized into clips. In order to keep the method simple, and avoid costly optimization algorithms for Gaussian mixture learning, we adapt a method similar to the one proposed in [27], which performs a hierarchical clustering of optical flow to simultaneously segment motion in space and time, as well as learning of the parameters of the Gaussian mixture representing each motion pattern. We therefore obtain a set of Gaussian mixtures, $V = \{v_i\}$, $1 \le i \le Q_f$, so that the activity f is represented by Q_f agents or motion patterns, and $v_i = \{(\mu_{i,k},\Sigma_{i,k},\omega_{i,k},\tau_{i,k})\}$, $1 \le k \le N_i$, where $\tau_{i,k}$ is the time at which the kth component of v_i is observed. Each motion pattern v_i therefore, comprises of a set of N_i quadruplets representing a 4d Gaussian component's mean, covariance matrix, weight, and the time of occurrence respectively. The obtained motion patterns are then warped using manually computed homographies from the stabilized clip to a field model. Fig. 2 shows the outputs from some of these pre-processing steps and a brief illustration of the subsequent graph construction whose details will be provided in the following sections of the paper. For visualization, we illustrate

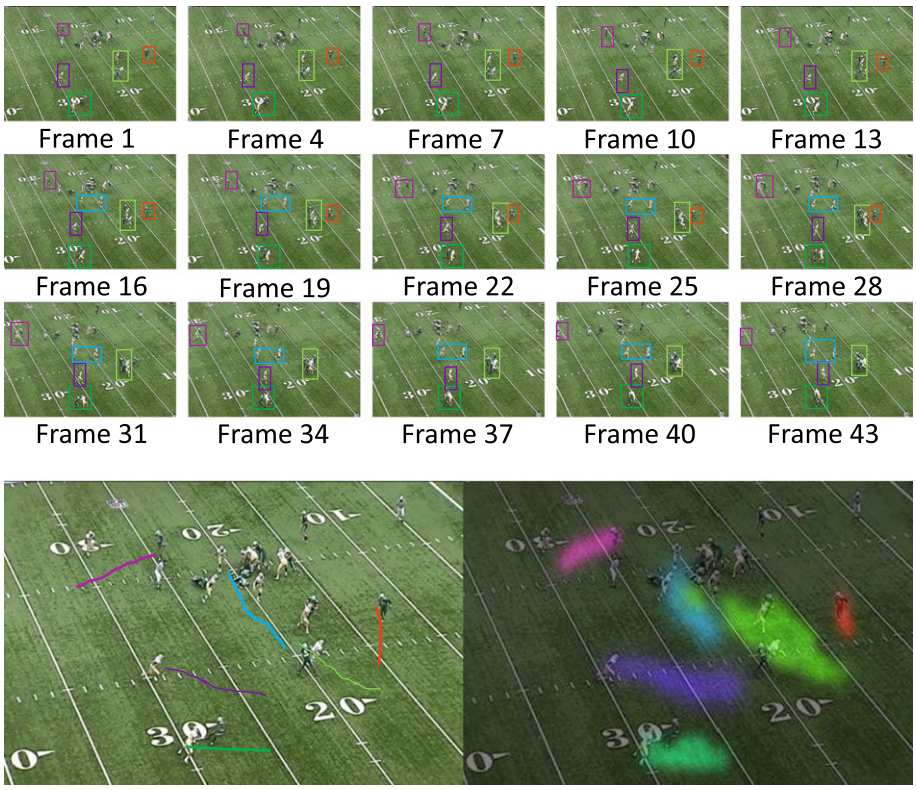

Fig. 1. Some frames from a football play from our dataset are shown. Each group of one or more players causing a sub-event within the play is tracked manually in the subsequent frames shown row-wise. Notice that we are considering groups of players that move together as a single entity, and a different color is used for each group. The image on the bottom left shows all tracks overlaid on one frame using the same color as for the boxes. The image on the bottom right shows our automatically obtained motion representation (which is explained in Section 2) which closely follows the manually tracked groups of players. The motion detected for each group of players is shown in the same color as used for the corresponding manual track for that group of players in the image in the bottom left of the figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a motion pattern as the conditional expectation of optical flow given pixel locations.

The framework for estimating motion patterns is complimented by a three phase filtering step to remove noisy optical flow. Firstly, weak optical flow is thresholded, and the clusters obtained from flow are filtered using their covariances. A final filtering step is performed on the obtained patterns in which any patterns with a very small number of constituent clusters are removed. This provides

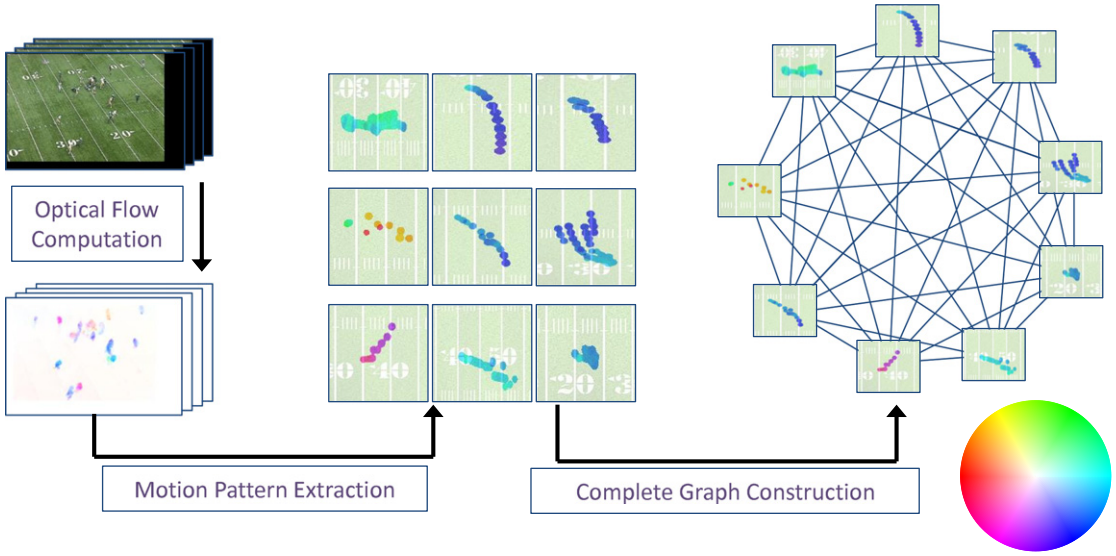

Fig. 2. An illustration of the processing steps involved in obtaining motion patterns is shown. We obtain optical flow from a clip of a football play. Patterns of motion within the play are extracted from the flow using a hierarchical clustering framework. We describe the entire activity using a dense graph constructed using all observed motion patterns as vertices of the graph.

good robustness to clutter in the scene, for problems such as camera shake, imperfect video registration, and in the specific case of football, any motion in the crowd.

We now perform an in depth evaluation of the merits of our framework with the related methods. The most recent work in this direction is by [18]. The authors use a driving force model to characterize a group of agents that share a goal. However in that work, the number of driving forces has to be manually specified. Since different plays may have varying levels of complexity, the actual number of factors that influence motion may be different between classes. Their motion model is defined by a single affine matrix, which is counter-intuitive in a sports scenario even in small regions. The representation does not appear to be discriminatory between play classes and the results in the paper only demonstrate segmentation of sub-activities within a single play.

Similar work on motion modeling has been using HDP/LDA models [14]. Those frameworks require repeated flow to construct a model, therefore they do not apply to sports scenarios. However, our motion pattern representation may be considered to be a simplified version of the HDP framework that is applicable to the current setting. Similarly work done by Li et al. [21,22] also requires persistent flow for modeling. In addition, HDP/LDA approaches generate actor representations that are not very easy to manipulate, e.g., transform spatially, which is required for our graph matching method.

3. Graph theoretic framework

The multi-agent activity instance f, is represented by a planar, directed, complete graph, $G_f = (V_f, E_f)$, where V, the set of vertices, is a collection of the parameters of Gaussian mixtures described earlier, and $|V| = Q_f$. Each element of the $Q_f \times Q_f$ matrix, E of edges contains a vector representative of the optimal spatial transformation between Gaussian mixtures of two vertices connected by that edge. Fig. 3 provides an illustration of one node pair and the edge between them in a graph.

Using this global representation of an activity in time and space using our proposed event modeling we can compare two activities. We do this by attributed inexact graph matching. For comparison between two activities we require a comparison of the location and flow of individual events, their relative times of occurrence, the pairwise spatial relationships between these events and the relative saliency of an event within the entire activity. We now describe how we encode this information into our framework.

3.1. Agent behavior similarity

We can compare the vertices of two graphs which probabilistically represent an agent's behavior by simply computing the KL divergence between the Gaussian mixtures. There are however a few problems with this approach. First, KL divergence is not a distance metric, and is not symmetric. It has a high dynamic range for dissimilar distributions, especially in different directions. Second, KL divergence between Gaussian mixture distributions does not have a closed form, and Monte Carlo point sampling is often used to estimate it, which is a computationally expensive operation. We therefore define a distance measure comparing two Gaussian mixtures which takes into account their location and shape in (x,y), and their motion in (u,v). Specifically, given graphs, G_f and G_g for activity instances f and g respectively, we compute a vertex–vertex similarity matrix $\mathcal{D}^{\alpha}$ of size $Q_f \times Q_g$, which is symmetric and positive. Given two agent behaviors, v_i and v_j from graphs, G_f and G_g, the elements of matrix, $\mathcal{D}^{\alpha}_{f,g}$ are defined as:

$$d^{\alpha}_{i,j} = \exp\left(-\frac{\Delta\theta^2_{i,j}}{2\sigma^2_{\theta}} - \frac{\xi^2_{i,j}}{2\sigma^2_{\xi}} - \frac{\left(1-\delta_{i,j}\right)^2}{2\sigma^2_{\delta}}\right), \tag{2}$$

where,

$$\Delta\theta_{i,j} = \tan^{-1}\left(\frac{\sum_{k=1}^{N_i}\omega_{i,k}\mu^{(4)}_{i,k}}{\sum_{k=1}^{N_i}\omega_{i,k}\mu^{(3)}_{i,k}}\right) - \tan^{-1}\left(\frac{\sum_{k=1}^{N_j}\omega_{j,k}\mu^{(4)}_{j,k}}{\sum_{k=1}^{N_j}\omega_{j,k}\mu^{(3)}_{j,k}}\right), \tag{3}$$

is the difference of mean optical flow orientations of the Gaussian mixtures (adjusted for phase change). The variable ξ computes the minimum Euclidean distance between any two components in each mixture,

$$\xi_{i,j} = \min_{1\le m\le N_i, 1\le n\le N_j}\left\|\mu^{(1,2)}_{i,m} - \mu^{(1,2)}_{j,n}\right\|_2, \tag{4}$$

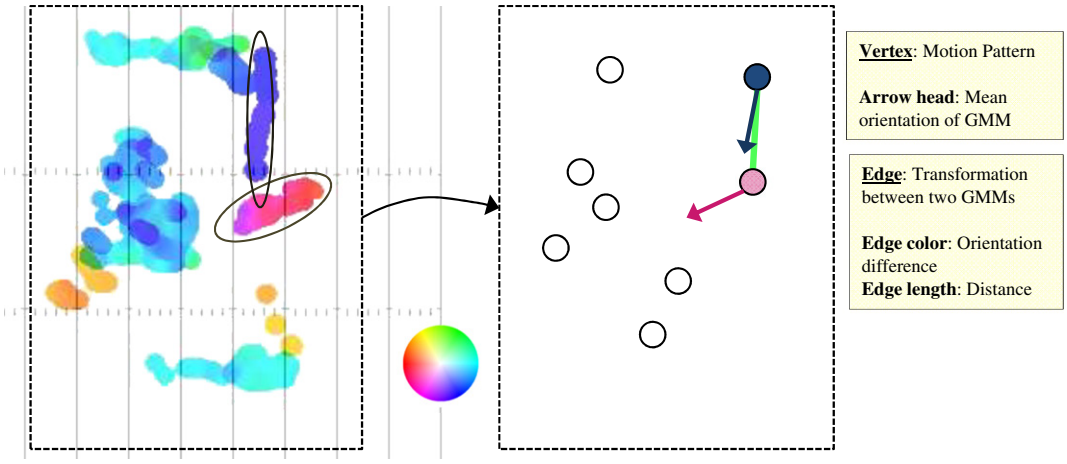

Fig. 3. An illustration of the proposed multi-agent activity graph, where nodes are Gaussian mixture distributions, and edges have Lie space representations of geometric transformations that align the Gaussian mixtures. The figure shows one node pair and the edge between them. The mean orientation of a motion pattern is depicted by the arrow head and its color as per the color wheel. Edge color corresponds to the difference in mean orientation of the patterns that it connects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and the shape similarity between two motion patterns is measured by the amount of spatial overlap between Gaussian mixtures,

$$\delta_{i,j} = \frac{R_i \cap R_j}{R_i \cup R_j}, \tag{5}$$

where the regions are sets of pixels with high probability of belonging to the motion pattern,

$$R_i = \left\{ \mathbf{x}^{(1,2)} \quad s.t. \quad \int\int p_i(\mathbf{x}|v_i) du dv > \kappa \right\}. \tag{6}$$

The parameter κ was fixed for all our experiments. We now explain how we leveraged the temporal information in the agent behaviors for improved graph matching.

3.2. Temporal similarity of occurrence

Another useful cue towards robust matching of two activity instances is the relative temporal position of each agent behavior within the activity. For example, two instances of football plays should be more likely to match if a specific type of pass is executed after approximately the same relative duration into the play. We observe that by nature of the process of motion patterns estimation, the quantized time (video clip) at which each component of the mixture was observed is known, i.e., $\tau_{i,k}$. We can then estimate the approximate time of occurrence of an agent behavior as the mean time of all components. The *relative* time of occurrence for the ith motion pattern is then written as,

$$\Gamma_i = \frac{1}{N_i \cdot Z_f} \sum_{k=1}^{N_i} \tau_{i,k}, \tag{7}$$

so that $0 < \Gamma_i \leq 1$, since $\tau_{i,k} \in \{1, \ldots, Z_f\}$, and represents an additional property of the node v_i, which will be helpful in more accurate graph matching, vertex correspondences, and therefore estimation of similarity between activity instances. The approximate relative temporal location of agent behaviors within the activity video will be compared with that of behaviors (or nodes) in other activity graphs.

Using the relative time of occurrence, we can determine whether two events in two different plays are similar with respect to their temporal positions within the respective plays. We define their temporal similarity of occurrence of two motion patterns v_i and v_j in distinct graphs G_f and G_g as, $d^{\lambda}_{i,j} = 1 - |\Gamma_i - \Gamma_j|$, where $d^{\lambda} \in [0,1]$. Therefore two agent behaviors occurring at the beginning and at the end of their respective activities will have the minimum temporal similarity, and vice versa. Computing the temporal similarity of occurrence for all pairs of activities between two plays gives us a $Q_f \times Q_g$ matrix, $\mathcal{D}^{\lambda}_{f,g}$.

We now describe our method for estimating edge to edge similarities between the graphs, G_f and G_g.

3.3. Spatial layout similarity

One of the main reasons multi-agent activities are modeled as collections of atomic, agent specific behaviors, is that subtle differences in spatiotemporal relationships between these behaviors can represent distinct high level activities. Conversely, two instances of the same activity are likely to have non-rigid transformations between agent behaviors across instances.

In order to estimate the spatial relationship between two patterns, we employ an iterative warping procedure which attempts to align multivariate Gaussian mixtures such as the ones representing agent behaviors. If a large set of 4d points are sampled from each Gaussian mixture, this problem is analogous to registration of point clouds. A recent method ideally suited to this problem is proposed in [12]. We leverage a similar method proposed in [13]. The estimated relationship consists of the transformation parameters that would optimally warp one pattern onto the second pattern so that their KL divergence is minimized after warping. A 3×3 similarity matrix depicting this transformation is then written as:

$$\mathbf{T} = \begin{bmatrix} s\mathbf{R} & \mathbf{t}^{\top} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{8}$$

where $\mathbf{R}$ is a 2×2 rotation matrix, s is the scale, and $\mathbf{t} = \{t_x, t_y\}$, is the translation vector. Since G is complete, a transformation $\mathbf{T}_{ij}$ is computed between all pairs of vertices i and j. It can be noticed however that we only need to compute either the upper or lower triangle of matrix E, since $\mathbf{e}_{ij} = \mathbf{e}_{ji}^{-1}$, where $\mathbf{e}_{ij} = [s\mathbf{R}_{(1,1)}, s\mathbf{R}_{(2,1)}, \ldots, t_x, t_y]$. The 6-long vector $\mathbf{e}$ therefore represents how two Gaussian mixtures are translated, rotated, and scaled with respect to each other, and the edge attributes matrix E captures the geometric relationships between all pairs of mixtures.

In order to obtain a measure of similarity between the relationships of vertices within two distinct graphs, we need to compare the attributes of the edges connecting the vertices. We therefore create a $Q_f^2 \times Q_g^2$ matrix Φ, where elements of the matrix will be the similarity values between all possible pairs of edges in the two graphs. The problem however is that the edge attribute $\mathbf{e}$ computed previously (Eq. (8)) is not a vector space, and therefore not closed under vector addition or scalar multiplication. Indeed, a simple Euclidean distance between $\mathbf{e}_{ij}$ and $\mathbf{e}_{mm}$ makes little sense without careful but aggressive scaling of the individual elements of the 2d transform, which include translation in pixels, trigonometric functions of rotation angle, and a scaling parameter. It is therefore desirable to map the multiplicative structure of Similarity transforms to a vector space such that the intrinsic geometric structure of the transformation is preserved. To this end, we propose to leverage Lie algebra which can be used to find exactly such a mapping. Several recent works in the literature have used this approach to allow analysis on Affine and Projective groups [21,22]. A more detailed treatment of Lie algebra and Lie groups can be seen in [17,5]. Using the Lie algebraic approach, we therefore define the edge to edge similarity as, $\phi_{ij,mn} = \|\mathbf{X}_{ij} - \mathbf{X}_{mn}\|_2$, where $\mathbf{X}$ is the Lie algebraic representation of the Similarity transformation $\mathbf{T}$ computed as,

$$\mathbf{X} = \log(\mathbf{T}) = \sum_{a=1}^{\infty} \frac{(-1)^{a+1}}{a} (\mathbf{T} - \mathbf{I})^a. \tag{9}$$

It has been shown in [3] that $\mathbf{X}$ can be represented as a linear combination of basis vectors called 'generators' of the Lie group, so that the coefficients of the combination serve as a representation of the original transformation in Lie space. Moreover [3], shows that for transformations near identity, the higher order terms of Eq. (9) can be ignored, thus making the mapping tractable. Given an edge to edge similarity matrix, Φ, we seek to convert it into a $Q_f \times Q_g$, vertex to vertex similarity matrix, $\mathcal{D}^{\beta}$,

$$\mathcal{D}^{\beta}_{v,v'} = Pr\left(m(v) = v' \middle| G_f, G_g\right), \tag{10}$$

where the 'm' function denotes a mapping between the nodes of the two graphs G_f and G_g. For this purpose, we employ the method of [33], which performs a probabilistic soft hyper-graph matching. An optimization is defined as the following minimization,

$$\min_X dist\left(\Phi, \otimes^2 \mathcal{D}^{\beta}\right). \tag{11}$$

The symbol '$\otimes$' here represents the Kronecker delta product. Given two matrices $A_{p \times q}$ and $B_{r \times s}$ the Kronecker delta product is the resulting $pr \times qs$ block matrix,

$$A \otimes B = \begin{bmatrix} a_{11}B & \ldots & a_{1q}B \\ \vdots & \ddots & \vdots \\ a_{p1}B & \ldots & a_{pq}B \end{bmatrix}. \tag{12}$$

This optimization allows us to estimate an edge based similarity of vertices. Fig. 4 illustrates how we may use both edges and vertices to compute vertex–vertex similarities between two graphs. Further details are provided in the following sub-sections.

3.4. Spatial saliency of agent behaviors

One final cue that we estimate for each agent behavior is the spatial saliency. We notice that in videos of football plays, most plays will essentially begin with very similar behaviors, but as the play progresses they become more distinct and discriminative for different classes of plays, which is a rather obvious and intuitive observation. Since the end of the play is signaled by a tackle or touchdown, it would be useful if the location of the corresponding behavior was known. To this end, we leverage a very simple approach, whereby we transform the location of the center of the last frame to the mosaic generated by ego-motion compensation. The assumption implicit in this step is that most of the broadcast videos are captured from dynamic cameras which focus on the point of action within the play, which almost always is the point of tackle or touchdown towards the end of the play. Given the tackle or touchdown location for play f in global coordinates as a 2d vector, $\mathbf{p}_f$, we compute the spatial saliency of a motion pattern, v_i as,

$$L_i = \exp\left(-\frac{1}{2\sigma_L^2}\left\|\sum_{k=1}^{N_i}\omega_{i,k}\mu_{i,k}^{\{1,2\}} - \mathbf{p}_f\right\|_2^2\right), \tag{13}$$

where σ_L is a constant value fixed at 30 pixels. Notice that although the spatial saliency, L_i, is a property of the node v_i, it cannot be *compared* with the saliency of another node. It can however, be combined to indicate the joint saliency of a particular pair of nodes, each from a distinct graph, as explained in detail in Section 3.5. For the entire activity video, we create a Q_f long vector, $\mathbf{L}_f = \{L_i\}$, where $i \in \{1, \ldots, Q_f\}$. It should be mentioned that although the proposed approach is formally evaluated for the application of football plays analysis, in general, it is applicable to different kinds of multi-agent activity, and the saliency mentioned above is the only application specific step in the process. It is conceivable however, that sub-events in other activity domains can also benefit from different measures of saliency. For example, in a traffic intersection scenario, proximity of an event to the center of the intersection may indicate its importance.

We argue that the proposed graph based representation of a multi-agent activity video captures its dynamics and structure in an efficient and comprehensive manner. Information about number of agents, their temporal occurrence, spatial occurrence and inter-agent spatio-temporal relationships is inferred in a completely unsupervised fashion, without the need for tracking individuals, or in the case of football plays, distinction between opposing teams, specific player or ball detection. In addition agent behaviors have a rich probabilistic representation that captures their density, magnitude and direction of per-pixel motion, and spatio-temporal localization. Given such an efficient representation an activity model can be compared to another for retrieval and recognition as we describe in the following section.

3.5. Graph matching by multi-cue fusion

Finally, given vertex to vertex similarities using multiple criteria, we compute a weighted average to obtain a single matrix that is used to perform graph matching and estimate the cost of the match, which serves as the final similarity metric between videos containing multi-agent activities. Specifically, we write,

$$\mathcal{D}_{f,g} = \mathbf{L}_f^{\top}\mathbf{L}_g \cdot \mathcal{D}^{\lambda}\left(c_\alpha \mathcal{D}^{\alpha} + c_\beta \mathcal{D}^{\beta}\right), \tag{14}$$

where the matrices $\mathbf{L}_f^{\top}\mathbf{L}_g$ and $\mathcal{D}^{\lambda}$ serve as element-wise weights for correspondence, based on the joint spatial saliency and temporal similarity of pairs of agent behaviors, while scalars c act as corresponding weights for each cue.

Using the vertex to vertex matrix $\mathcal{D}_{f,g}$, we now assume the nodes of each graph, V_f and V_g to be independent sets of a complete bipartite graph, and attempt to find a set of correspondences between them so that the probability of the global assignment is maximized. We employ the well known Hungarian [15] (aka Munkres [26]) algorithm for this purpose, and obtain a binary $Q_f \times Q_g$ matrix, $\mathcal{M}$ of correspondences where an element is 1 if the relevant agent behaviors in each activity video match. Final similarity between the two videos of multi-agent activities is then given as,

$$\mathcal{S}_{f,g} = \sum_{\text{all rows}} \sum_{\text{all columns}} \mathcal{D}_{f,g} \cdot \mathcal{M}_{f,g}. \tag{15}$$

4. Experiments and results

We applied the proposed activity recognition framework to the specific application of recognizing offensive strategies in American football plays. This is an extremely challenging problem within multi-agent activity analysis due to severe and persistent person to person occlusion, while tracking in general is currently not a viable approach. Many of the methods proposed in the literature including the state of the art results for this problem for similar datasets, use manual annotations for not only player tracking but also player role identification. To the best of our knowledge ours is the first method that employs only low level optical flow to learn the representation of a football play. We show that our method is a practical solution which performs well in a challenging real world scenario with imperfect input data.

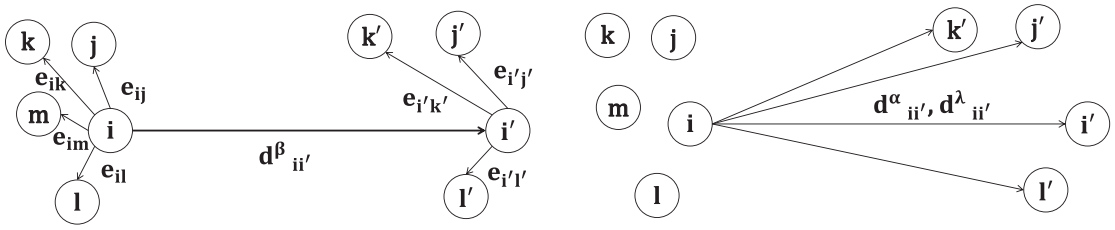

Fig. 4. Graphical representations of two different activities. Comparisons between the two may be made using edge (as shown on the left) and node (as shown on the right) information. Notations d^{α}, d^{λ} and d^{β} in this illustration represent a similarity measure between nodes based on their shape similarity, temporal similarity of occurrence and spatial similarity of occurrence. We define these measures in Section 3. The notation 'e' represents edges between nodes. The graphs we construct in our work are complete graphs. Not all edges have been shown in this illustration.

Table 1
A comparison of the classes and balance in number of instances per class between the GaTech dataset and UCF Football Dataset.

	Left run	Middle run	Right run	Rollout pass	Short pass	Deep pass	Option pass
GaTech Dataset							
# of instances	23	19	11	5	7	8	5
UCF Football Dataset							
# of instances	13	12	9	5	14	17	10

4.1. Dataset details

The most commonly used dataset for evaluation of football play recognition has been the GaTech dataset consisting of 7 classes [19], which is not publicly available, and no other standard datasets of offensive football plays exist. We therefore collected our own 7 class dataset which consists of broadcast footage obtained from 3 NCAA games, which will be shared with the community.

Each play is manually segmented starting from the hike to the quarter back and ending when the ball carrier becomes stationary. The total number of clips is 80. The number of instances of each class is shown in Table 1. Manual annotations for play direction (i.e. direction of offensive play) and homographies that warp the first frame of each clip to a field model are available. Sports video registration to a field model has been treated in detail in [6] and is beyond the scope of this work.

We first show the data available to us after complete processing of a football play. Fig. 5 shows an example representation for each particular play type and the corresponding play diagram. It can be observed that our representation bears similarity with play diagrams commonly used for coaching and/or planning of football plays. Fig. 6 shows a few examples of each play type. As can be

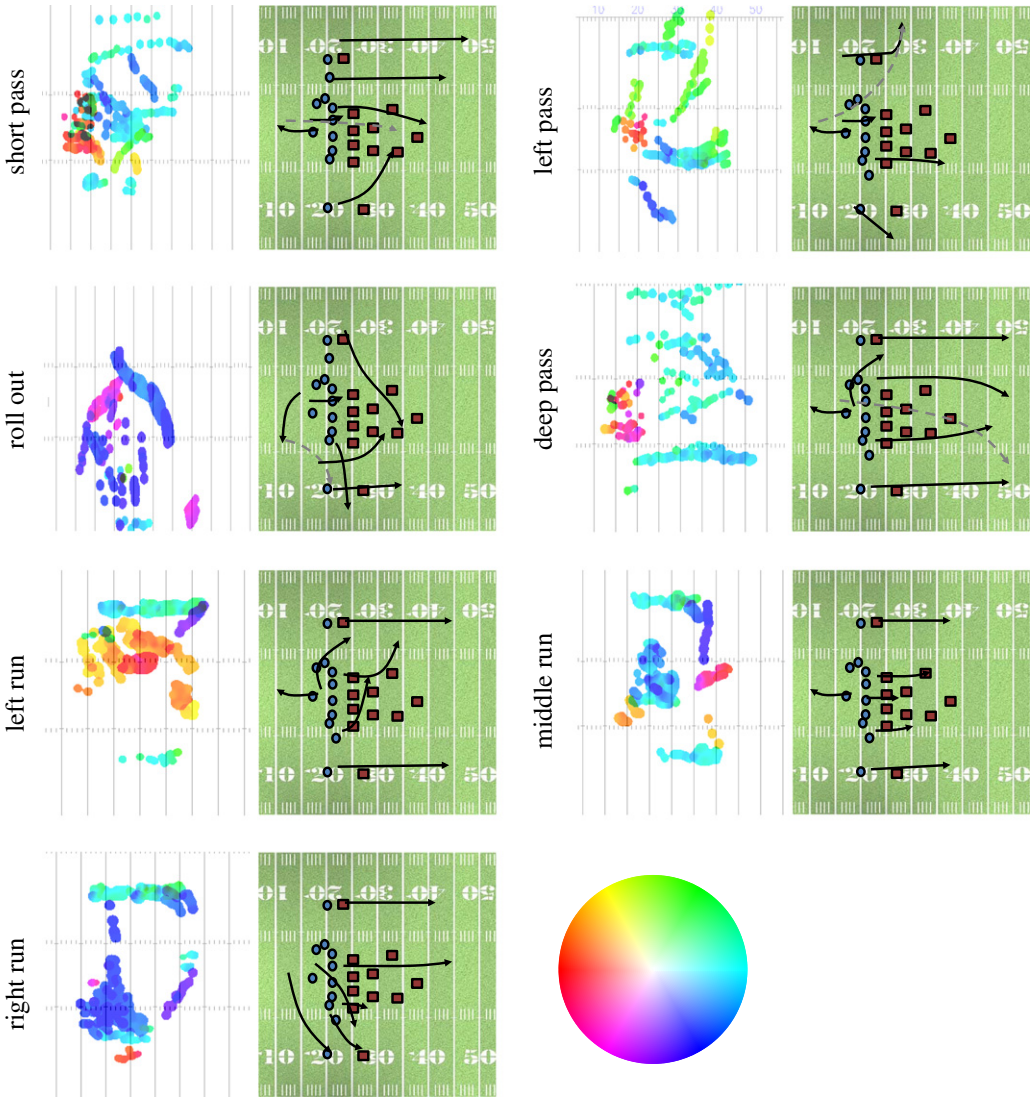

Fig. 5. One example of each type of play along with an associated coach's play diagram.

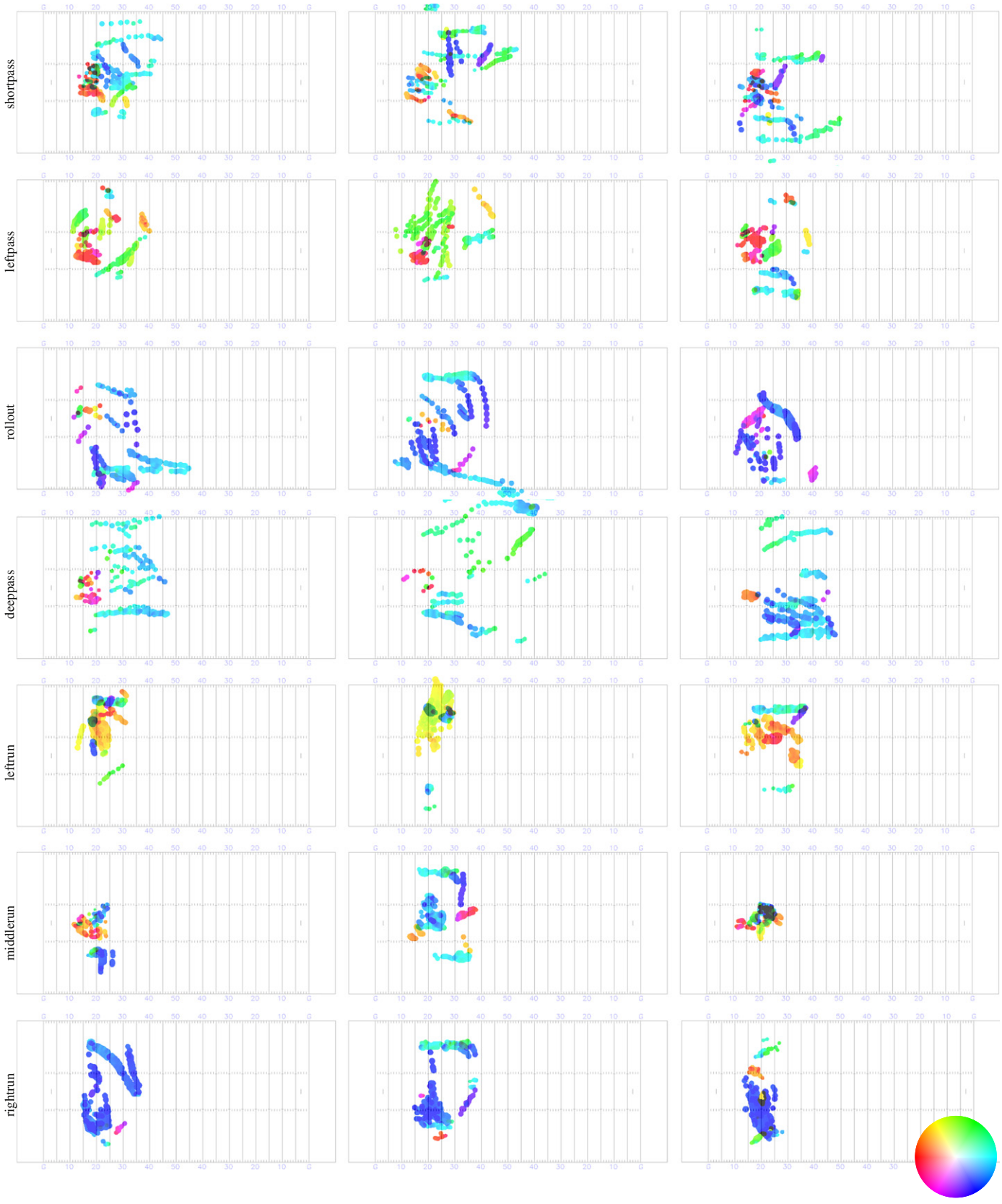

Fig. 6. Three examples each for the seven classes in the UCF dataset used in our experiments. Each example is shown as a collection of motion patterns, or individual agent behaviors observed in the video.

observed, play types have large intraclass variance and play examples have a lot of clutter that may be irrelevant to the prediction of the play.

Observe that the classes in the UCF Football Dataset are similar to the classes in the GaTech dataset with only one difference owing to the data available to us. Our dataset is more balanced

Table 2
Number of agent behaviors by Play type for $Z_f = 10$.

Play type	Left run	Middle run	Right run	Roll out pass	Short pass	Deep pass	Left pass	All running plays	All passing plays
Mean	30.76	29.08	37.33	37.8	35.57	32.58	38.2	31.91	34.97
Standard deviation	6.61	8.67	6.42	12.15	6.81	6.42	5.59	7.89	6.61

as compared to the GaTech dataset, therefore results on our dataset are a better estimate of the accuracy of a method under test. The most relevant approach to ours is by [30]. In fact it is the only approach that shows classification results on a comparable football dataset consisting of an equal number of classes without using manual tracks. That work however lacks implementation details as well as statistics related to tracking performance, therefore we are unable to replicate their results. However given the similarity of our dataset with the GaTech dataset as we have argued above, comparison of our results with theirs is justified.

Since the videos include frequent panning and zooming, we first stabilize the video to remove camera motion so that the optical flow computed strictly corresponds to object motion. We then transform the global reference to a football field reference for better visualization and inter-video alignment. Optical flow as well as individual agent behaviors V (motion patterns) are computed from stabilized video and then warped to the field model. We divide each video into a number of equal length video clips. The length of clips across videos however, can be different.

4.2. Experiment statistics

To get an idea of the diversity in video lengths, we note that the average length of a play video is 170.49 frames, with a standard deviation of 49.61. Notice that we do not require all videos to be of the same length. The number of agent behaviors obtained from various play classes is given in Table 2. As we can see the number of agent behaviors obtained varies by play type. Running plays have a fewer number of agent behaviors on average compared to passing plays as running plays usually end sooner whereas the activity in passing plays may be more spread out spatially and continue for a longer period of time.

4.3. Experimental setup

To test our method, we divide each video clip of a play into smaller temporal segments. We find the agent behaviors in each of these smaller segments and construct the graphical representations of all play instances in our dataset. We then divide the dataset into training

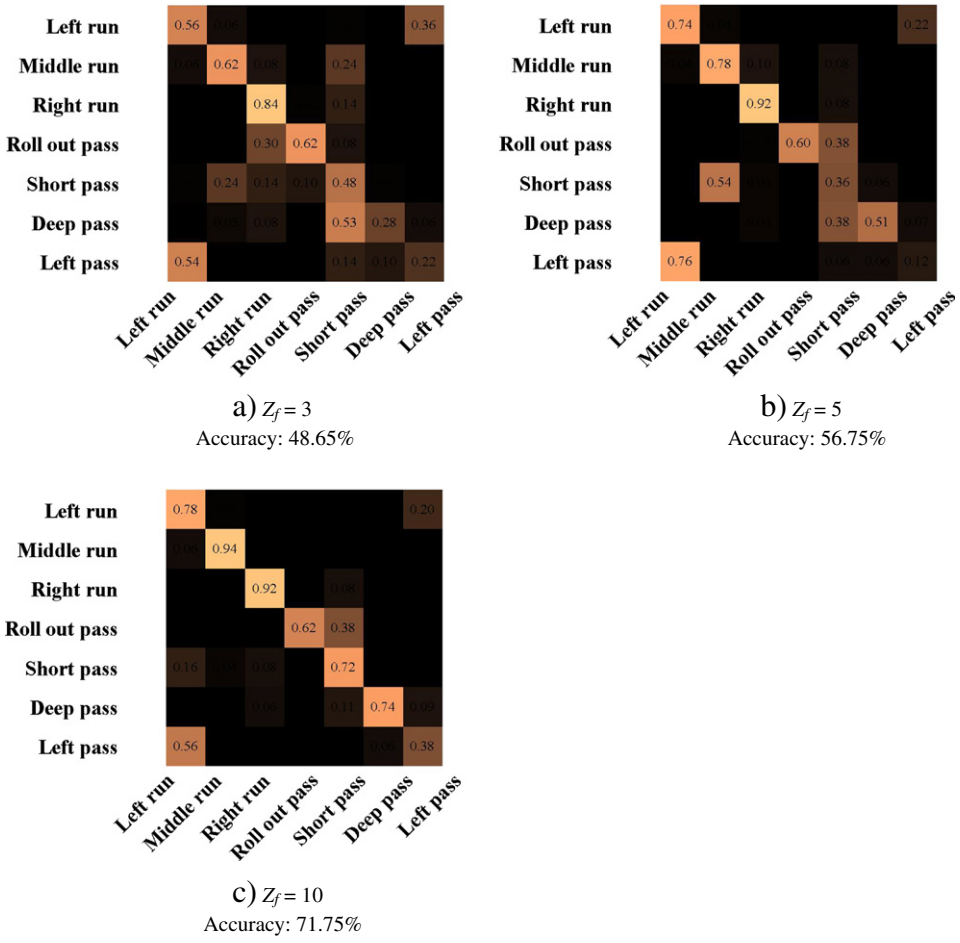

Fig. 7. Confusion tables for classification over 7 football play classes. The results in (a), (b) and (c) are generated with 3, 5 and 10 temporal segments respectively.

Table 3
Classification accuracy with respect to the weighting parameters c_β and c_α.

c_β	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
c_α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Accuracy (%)	24.01	59.16	66.6	69.37	70.51	71.75	70.99	71.40	71.49	69.73	69.71

and testing groups and perform classification in a simple nearest neighbor framework. The similarity between two plays is computed by graph matching using the cues and cost metrics defined in Section 3.

4.4. Results

To perform the Nearest Neighbor classification, we divide our data into training and testing groups in a 90–10 ratio and perform 200 runs for each experiment. We test our framework using three different numbers of temporal segments. For values of $Z_f = 3$, 5 and 10, we get classification accuracies of 48.65%, 56.75% and 71.75% respectively. The confusion matrices for these experiments are shown in Fig. 7. As can be observed, the results improve gradually as we increase the number of segments which shows that analysis of play events over time provides vital discrimination between classes. Note that the average number of frames per play in our dataset is 170 and as we keep increasing the number of temporal segments, the motion in each segment will approach frame by frame optical flow. In our experiments we do not use more than 10 segments as higher temporal resolution results in very small motion patterns and the matching scores become unreliable. All subsequent experiments are conducted with Z_f set to 10 unless specified otherwise.

In addition to the cost metric defined in Section. 3.5 we also explore other combinations of the individual cues and the effect of different weights in the combinations. Table 3 shows the total classification accuracy as the weights c_β and c_α in Eq. (14) are varied. As we can see, the performance degrades as the effect of one cue significantly outweighs the effect of the other cue. The best experimental performance is observed when we give equal weight to both.

We also test a different cost metric, whereby we use the temporal similarity cue in a similar manner as the edge and vertex similarity cues. We define this cost metric as,

$$\mathcal{D}_{f,g} = \mathbf{L}_f^\top \mathbf{L}_g \left(c_\lambda \mathcal{D}^\lambda + c_\alpha \mathcal{D}^\alpha + c_\beta \mathcal{D}^\beta \right), \tag{16}$$

The results for different combinations of the weighting parameters c_λ, c_β and c_α are shown in Table 4. We can see that the results are significantly worse, with this new cost metric especially when temporal similarity is given a higher weight. The performance is close to the results from the original cost metric when we do not use temporal similarity cue at all. The reason for an improved performance using the original cost metric is that it minimizes the similarity scores of patterns that may be similar but at temporally opposite extremes with respect to the duration of the two plays being matched. In using the alternate cost metric that we have tested, the temporal similarity cue imposes a smaller penalty on matching two motion patterns that may be temporally far apart and hence have a high similarity score in the resultant matrix $\mathcal{D}_{f,g}$ which results in a poor solution for $\mathcal{S}_{f,g}$ in Eq. (15).

Next, we test our framework using subsets of the cues mentioned in Section 3, so as to quantify their influence. The results for these tests are shown in Table 5. The corresponding confusion matrices are shown in Fig. 8. In Table 6 we show the classification performance when we use data from certain temporal segments only. As expected, performance increases as we increase the number of temporal segments used to build the graph of each play. Plots of classification accuracy for individual play types are shown in Fig. 9. The confusion matrices for these experiments are shown in Fig. 10. We observe that while accuracy for play types 'Deep Pass' and 'Left Pass' remains very low in the first few time segments, it rises sharply towards the end which suggests that the agent behaviors in these plays are discriminative towards the end of the play activity. Since activity towards the end of a play is usually near the tackle region, it can also be argued that agent behaviors near the tackle region provide discrimination between play types. A similar trend is observed in the 'Right run' play type. On the other hand the accuracies for the other play types either fluctuate around a mean value or grow slowly with time. It should be noted that although classification accuracies for running plays and passing plays have been plotted in separate figures, the values were obtained using all play types in the dataset.

Finally, we present a comparison of our results with other methods on football play recognition in Table 7. Li et al. [20] and Intille & Bobick [10] report a higher classification accuracy than ours. However it is very important to note that both these methods use manually generated player identities and tracks which makes both these approaches impractical in real scenarios. In addition Li et

Table 4
Classification accuracy with respect to the weighting parameters c_λ, c_β and c_α defined in Eq. (16).

c_α	0	0.2	0.4	0.6	0.8	1
c_λ c_β	1	0.8	0.6	0.4	0.2	0
0	0.2401	0.6184	0.6620	0.6754	0.6708	0.6724
0.2	0.1418	0.1417	0.1451	0.1582	0.1739	0.1724
0.4	0.1419	0.1401	0.1434	0.1427	0.1413	0.1429
0.6	0.1411	0.1396	0.1407	0.1419	0.1431	0.1417
0.8	0.1409	0.1411	0.1399	0.1447	0.1391	0.1421
1	0.1414	0.1399	0.1413	0.1392	0.1425	0.1417

Table 5
Quantitative results of proposed approach when using a subset of the cues mentioned in Section 3. These results correspond to $Z_f = 10$.

All but temporal similarity	Agent behavior and spatial layout similarity	Agent behavior and spatial saliency	Agent behavior and temporal similarity	All but spatial layout similarity	All
65.75%	32.0%	66.25%	41.0%	70.25%	71.75

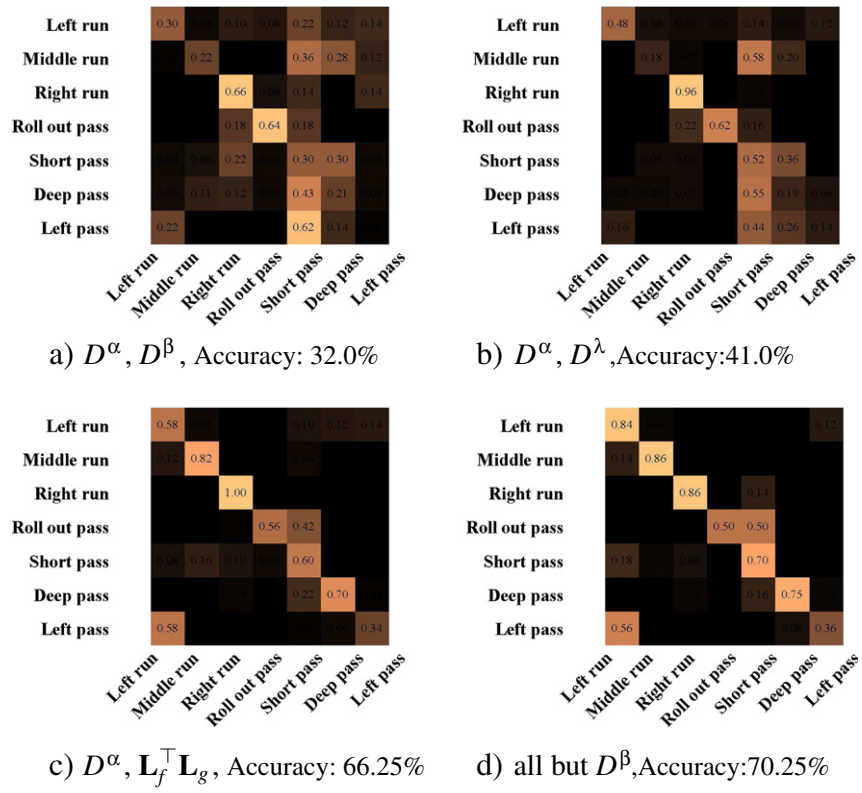

Fig. 8. Confusion tables for classification using subsets of the available cues.

Table 6
Classification accuracy with respect to the number of temporal segments (out of a total of ten) used.

Segments used	1	2	3	4	5	6	7	8	9	10
Accuracy (%)	27.08	29.4	31.5	38.0	49.5	54.25	57.25	57.5	61.75	71.75

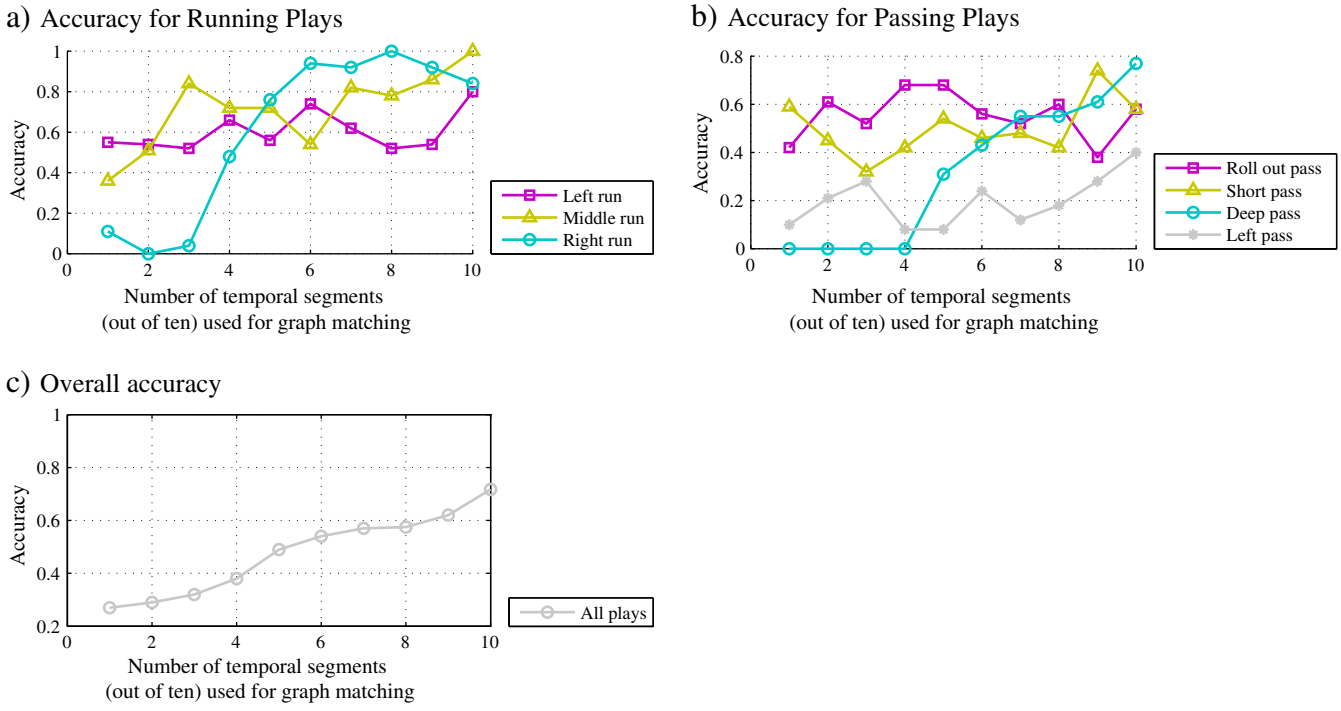

Fig. 9. Plots of classification accuracies of individual play types as well as for all play types with respect to the number of temporal segments used out of a total of ten.

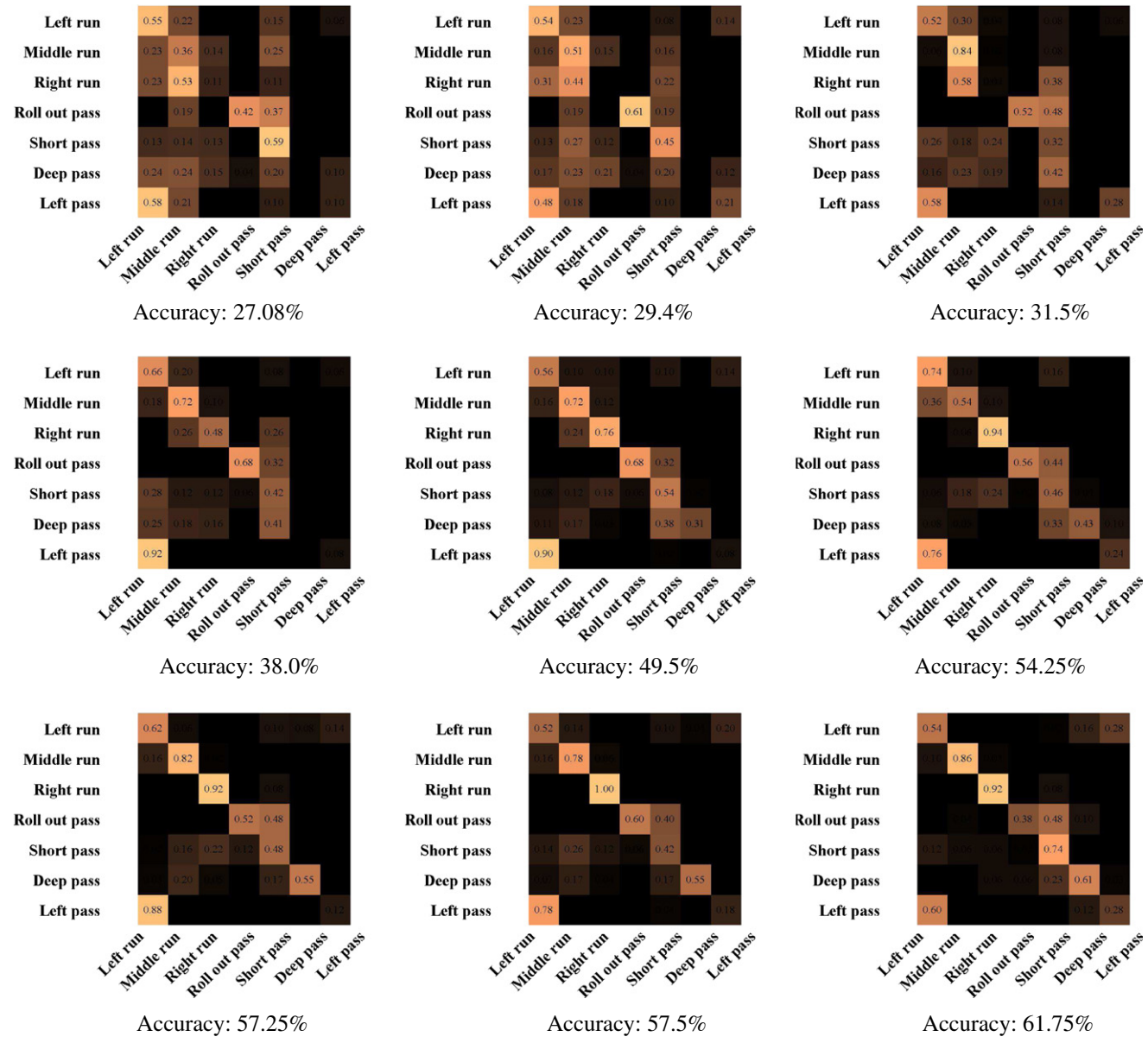

Fig. 10. Confusion matrices showing classification performance when using a fraction of the total temporally segmented play data. The nine confusion matrices starting from top-left in row-wise fashion correspond to the results generated with the firstone to nine consecutive temporal segments.

al. [20] uses a dataset with fewer classes than ours. Intille and Bobick [10] use more classes but their dataset consists of only 25 instances of football plays with 14 different classes. The results they have reported therefore, serve only as a proof of concept of their proposed method and should not be compared with other methods as they are statistically unreliable. Li & Chellappa [19] report good results however their dataset is again less challenging than ours as it consists of a fewer number of classes. In addition, their method suffers from the drawback that it relies on tracking of players in the challenging sports environment whereas our method does not require tracks. Only Swears and Hoogs [30] perform experiments without annotated tracks on an equal number of classes and a comparable dataset, even though it

Table 7
A bird's eye view of the assumptions, experimental conditions, and results of related approaches for football play recognition. Note that the average accuracy in general reduces as impractical assumptions and constraints are relaxed.

Method	Assumptions				Number of classes	Accuracy
	Performs tracking	Assumes tracks	Player roles	Team ids		
Li & Chellapa [19]	Yes	No	No	No	3	70%
Li et al. [20]	Yes	Yes	Yes	Yes	5	87.9%
Swears & Hoogs [30]	Yes	No	No	No	7	56%
Intillie & Bobick [10]	Yes	Yes	Yes	Yes	14	84% (21/25)
Proposed framework	No	No	No	No	7	71.75%

requires high quality player trajectories using a reasonable tracker. Our proposed approach shows a considerable improvement over this method.

5. Conclusion

To summarize the contributions of our framework, we have proposed a generalizable, probabilistic, graph-theoretic technique for recognizing multi-agent activities in videos. In contrast to existing approaches, we have completely avoided any tracking of objects, which is often infeasible in practical scenarios, and instead rely on low level, noisy optical flow to discover agent behaviors in a completely automatic, unsupervised fashion. We do not attempt to discover or leverage information such as number of agents, their roles or articulation of body parts etc. Results on a comprehensive seven class dataset are reported and the effect of proposed cues is quantified to demonstrate the feasibility and superiority of our approach.

References

[1] J. Allen, G. Ferguson, Actions and events in interval temporal logic, J. Log. Comput. 4 (5) (1994) 531–579.
[2] M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, G. Doretto, Joint recognition of complex events and track matching, CVPR, 2006.
[3] T. Drummond, R. Cipolla, Application of Lie algebras to visual servoing, IJCV 37 (1) (2000) 21–41.
[4] A. Hakeem, M. Shah, Multiple agent event detection and representation in videos, AAAI, 2005.
[5] B. Hall, Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, Springer, 2003.
[6] R. Hess, A. Fern, Improved Video Registration Using Non-distinctive Local Image Features, , 2007.
[7] S. Hongeng, R. Nevatia, Multi-agent event recognition, ICCV, 2001.
[8] S. Hongeng, R. Nevatia, F. Bremond, Video-based event recognition: activity representation and probabilistic recognition methods, CVIU 96 (2) (2004) 129–162.
[9] C. Huang, H. Shih, C. Chao, Semantic analysis of soccer video using dynamic Bayesian network, IEEE Trans. Multimedia 8 (4) (2006) 749–760.
[10] S. Intille, A. Bobick, Recognizing planned, multi-person action, CVIU 81 (2001) 414–445.
[11] Y. Ivanov, A. Bobick, Recognition of Visual Activities and Interactions by Stochastic Parsing, PAMI, 2000.
[12] B. Jian, B. Vemuri, Robust Point Set Registration Using Gaussian Mixture Models, PAMI, 2011.
[13] S. Khokhar, I. Saleemi, M. Shah, Similarity invariant classification of events by KL divergence minimization, ICCV, 2011.
[14] D. Kuettel, M. Breitenstein, L. Van Gool, V. Ferrari, What's going on? Discovering spatio-temporal dependencies in dynamic scenes, CVPR, 2010.
[15] H. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logist. Quart. 2 (1–2) (1955) 83–97.
[16] M. Lazarescu, S. Venkatesh, Using camera motion to identify types of American football plays, ICME, 2003.
[17] J. Lee, Introduction to Smooth Manifolds, Springer, 2002.
[18] R. Li, R. Chellappa, Group motion segmentation using a spatio-temporal driving force model, CVPR, 2010.
[19] R. Li, R. Chellappa, Recognizing offensive strategies from football videos, ICIP, 2010.
[20] R. Li, R. Chellappa, S. Zhou, Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition, CVPR, 2009.
[21] D. Lin, E. Grimson, J. Fisher, Learning visual flows: a Lie algebraic approach, CVPR, 2009.
[22] D. Lin, E. Grimson, J. Fisher, Modeling and estimating persistent motion with geometric flows, CVPR, 2010.
[23] C. Liu, May. Beyond pixels: exploring new representations and applications for motion analysis. Doctoral Thesis, Massachusetts Institute of Technology, 2009.
[24] T. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, CVIU (2006).
[25] V. Morariu, L. Davis, Multi-agent event recognition in structured scenarios, CVPR, 2011.
[26] J. Munkres, Algorithms for the assignment and transportation problems, J. Soc. Ind. Appl. Math. 5 (1) (1957) 32–38.
[27] I. Saleemi, L. Hartung, M. Shah, Scene understanding by statistical modeling of motion patterns, CVPR, 2010.
[28] C. Stauffer, W. Grimson, Learning Patterns of Activity Using Real Time Tracking, PAMI, 2000.
[29] E. Swears, A. Hoogs, Learning and recognizing American football plays, Snowbird Learning Workshop, 2009.
[30] E. Swears, A. Hoogs, Learning and recognizing complex multi-agent activities with applications to American football plays, Workshop on the Applications of Computer Vision, 2012, pp. 409–416.
[31] T. Wada, T. Matsuyama, Multiobject Behavior Recognition by Event Driven Selective Attention Method, PAMI, 2000.
[32] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, 2000.
[33] R. Zass, A. Shashua, Probabilistic graph and hypergraph matching, CVPR, 2008.