Group Level Activity Recognition in Crowded Environments across Multiple Cameras

Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, Ting Yu
GE Global Research
1 Research Circle, Niskayuna, NY, USA
{changm,krahnsto,limser,yut}@research.ge.com

Abstract

Environments such as schools, public parks and prisons and others that contain a large number of people are typically characterized by frequent and complex social interactions. In order to identify activities and behaviors in such environments, it is necessary to understand the interactions that take place at a group level. To this end, this paper addresses the problem of detecting and predicting suspicious and in particular aggressive behaviors between groups of individuals such as gangs in prison yards. The work builds on a mature multi-camera multi-target person tracking system that operates in real-time and has the ability to handle crowded conditions. We consider two approaches for grouping individuals: (i) agglomerative clustering *favored by the computer vision community, as well as* (ii) decisive clustering *based on the concept of modularity, which is favored by the social network analysis community. We show the utility of such grouping analysis towards the detection of group activities of interest. The presented algorithm is integrated with a system operating in real-time to successfully detect highly realistic aggressive behaviors enacted by correctional officers in a simulated prison environment. We present results from these enactments that demonstrate the efficacy of our approach.*

1. Introduction

The capability to automatically detect suspicious, disorderly or criminal activities from video sequences is highly desirable in domains such as prisons, schools, public places, sport venues and other public gatherings. Even more appealing than the detection is the early *prediction* of events that allows action to be taken before an event unfolds or escalates. We are particularly interested in domains and activities relevant to law-enforcement, and are addressing here the problem of detecting behaviors in environments where many close interactions occur at a group (or rather gang) level. Hence we seek to address the problem of detecting and reasoning about the spatio-temporal evolution of group structures so as to understand group-level activities of the crowd.

In computer vision, typical grouping strategies [10, 6, 16] rely on a bottom-up, *agglomerative* clustering [4, Ch.10.9] of individuals in order to find the group structure. Such grouping schemes could be *hierarchical* based on previously established clusters, using some distance metric reflecting the spatial-temporal features of the tracked targets. Hierarchical clustering approaches require a threshold (stopping criterion) to determine the final grouping. This is appropriate in environments where observed person-to-person distances follow standard social norms (*i.e.*, *proxemics* [8]) but not in environments where rapid changes in interaction distances occur [7, 11]. In order to compensate for the latter, we consider a second approach based on an eigen analysis of the graph adjacency matrix, which is motivated by its success in the social network analysis community. We propose a method based on the top-down concept of the graph *modularity* measure [11], which maximizes the difference between the connections within a group of individuals and the expected number (and strengths) of such connections. The grouping is performed by dividing the graph along connections that are not necessarily weak, but rather *weaker than expected*. This is crucial in achieving an *adaptive* grouping to segment groups across a variety of configurations, which is essential in this work.

In this paper, we present algorithms for recognizing several group-level activities that are of particular interest to the law-enforcement community. These algorithms range from recognizing low-level activities such as group formation, group dispersion, group loitering, to more advance activities such as group flanking and aggression/agitation. The presented algorithms are integrated in a comprehensive real-time surveillance system that performs multi-camera, multi-target tracking in challenging environments. The overall framework has been evaluated and tested live in an abandoned former prison with professional correctional officers enacting typical inmate behaviors. The system was able to

successfully detect a variety of group level activities, even successfully *predicting* the occurrence of (simulated) aggressive behaviors (gang on gang fights) before the actual event unfolded.

The rest of this paper is organized as follows. We discuss related work in Section 2. Section 3 briefly outlines our tracking system as well as the domain we are addressing in this paper. Section 4 describes the two strategies (bottom-up, top-down) for determining group structures. Section 5 describes the utilization of these group structures for recognizing group activities. In Section 6, we report test results from our system detecting in real-time group events in simulated law-enforcement environments. Section 7 concludes this paper.

2. Related Work

Fundamental to the success of any algorithms for recognizing group activities is the ability to track individuals (or group of individuals) under crowded conditions. There are numerous works that address the tracking problem both at the individual [5, 22] and group level [10]. In fact, whether or not a tracked blob belongs to an individual or a group could be ambiguous due to heavy occlusion. To this end, Grimson *et al.* [2] detect and track multiple objects as moving blobs and disambiguate fragmentation/over-segmentation by building an inference graph; from which they reason about the entire tracks of the objects based on spatial connectedness and motion coherence.

Given a set of detected tracks of individuals, one can group these tracks into cohesive entities. Ge *et al.* [6] identify small group structure of a crowd in a bottom-up fashion by iteratively merging sub-groups with the strongest inter-group closeness, utilizing a measure based on the symmetric Hausdorff distance. The clustering is hierarchical and is similar to the construction of a minimum-spanning tree (MST) from the individuals. The use of the Hausdorff distance requires continuous recomputation of group-to-group distance measures, which can be an expensive operation for large multi-camera surveillance sites.

As opposed to grouping individual tracks, there are also algorithms that identify grouping without necessarily identifying individuals in the groups. Lau *et al.* [10] hypothesize over both the partition of tracks into groups and the association of detections into tracks, and pose the group modeling problem as a recursive multi-hypothesis model selection problem. Groups are formed using *single linkage clustering*, which also is a variant of the MST algorithm. The assignment of observed cluster to a group is estimated using the minimum average Hausdorff distance. The merging of two groups is justified using a Mahalanobis distance between closest contour points. Hard thresholds on the blob size are used to identify whether a blob is a person or a group of people.

Robust detection and tracking of groups allows for the recognition of group activities and behaviors. Saxena *et al.* [16] model crowd events by defining case-specific scenarios and detect abnormalities such as falling, fighting, and emergence of new crowd flow. The event is trigger by imposing a hard threshold on several measures including crowd density, principal directions, number of individual motion vectors in a crowd.

3. System and Site Description

In this section we will provide a brief outline of our tracking system as well as the type of environment we are addressing in this paper.

3.1. Video Tracking System

A key factor for successful group analysis and subsequent recognition of group activities is the efficacy of the underlying video tracking system. The tracking system must perform reasonably well even under heavy occlusions, since groups often form in crowded conditions.

The tracking system that we utilize [9, 18, 22, 21] comprises of multiple calibrated static cameras tracking cooperatively in a synchronized fashion. For each view, based on the calibration, the image dimensions and positions of a given person at all possible 3D locations in the scene are estimated. These image locations are precomputed, and foreground pixels detected during online tracking are used to vote for these precomputed image locations to form a set of (foreground) detections [9]. This effectively leverages the calibration information to significantly reduce false positives arising from occlusions and crowdedness.

The set of detections for each view are then projected onto the ground plane in 3D in order to further disambiguate any confusion due to occlusions and crowdedness. These projections are consumed by a centralized tracking system that either (1) associates detections with existing tracks based on spatial proximity or (2) initiates new tracks. The states of tracks are estimated by a standard Kalman filter, performed in the world reference ground plane. The system is designed to maintain tracks across camera boundaries in order to perform site-wide tracking. Through calibrated camera views, all tracking information can be visualized in a top-down view of the whole surveillance site , which is particularly useful for security operations.

3.2. Domain

The system presented here is aimed at detecting suspicious and disorderly behaviors. For data collection and testing purposes the system was deployed in an abandoned prison yard in West Virginia, USA. Several correction officers volunteered to enact domain relevant behaviors such as

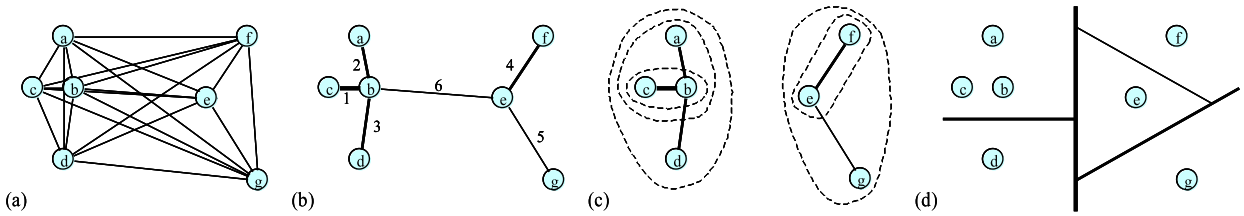

Figure 1. **Hierarchical *agglomerative* and *divisive* clustering.** (a) Determining the grouping of several individuals (a to g) is combinational in nature; here a complete graph is depicted. (b) Agglomerative clustering following Kruskal's algorithm in constructing the MST of the individuals. Edge weight reflects the distance between individuals. A "hierarchy" of groups corresponding to the disjoint forest sets in the MST is depicted in (c). (d) Divisive clustering formulates the grouping as a recursive cutting problem, where at each step the optimal cut between two subgroups is determined.

agitated arguments, fights, contraband exchange and many others. As many activities of interest for correctional settings are related to gang activities, many of the enacted scenarios simulated the presence of multiple gangs. The test system utilized a total of 4 standard CCTV cameras with three cameras used for tracking, one camera for automatic PTZ targeting (which will not be discussed in this work). An optional fifth camera was used for thermal imaging. The system performed live processing and reported events of interest in real-time to the operator.

4. Group Analysis

Given a set of tracked individuals, the first step towards group activity recognition is to cluster individuals into cohesive groups. This step plays a critical role in accurately detecting group-level events and recognizing group activities later on. Cluster analysis is well-studied in pattern classification and serves as a common technique in many fields. We focus on *hierarchical* clustering [4, Ch.10.9], which is simple in concept where successive clusters are found using previously established clusters. The clustering efficacy can be adaptively refined in an recursive fashion.

Hierarchical clustering can be divided into two main categories: agglomerative and divisive [4], Figure 1. *Agglomerative* clustering operates bottom-up, starting with each individual as a separate cluster and merging them into larger clusters. *Divisive* clustering operates top-down, beginning with the whole set and dividing it into smaller clusters. We investigate both approaches in the context of monitoring the (social) group structures of tracked individuals as follows.

A major component in group clustering is how the distance measure between individuals is defined. In agglomerative clustering, the distance function is often a fixed measure between two individuals such as the commonly used Euclidean (2-norm), Manhattan (1-norm), or maximum norm metric. It can be a variable measure depending on the current clustering configuration *e.g.* Hausdorff or Mahalanobis distance. In divisive clustering, finding the best division is often casted as finding the best cut in a graph network, where the distance is treated as edge weights and graph-theoretic methods can be directly applied.

4.1. Hierarchical Agglomerative Clustering

The agglomerative nature in clustering suggests a simple and intuitive way to form groups from individuals. We consider here the spatial-temporal dissimilarity between tracks of individuals as distance measure. First, a pair of elements with minimum distance are grouped together; then the second closest pair (which could be the newly formed group or a third) is merged; this process is repeated until a stopping criteria (distance threshold θ_m) is reached. Such clustering is bottom-up, local, greedy, and hierarchical, and is essentially constructing a minimum spanning tree (MST) of the individuals based on Kruskal's algorithm [3, Ch.24]. As Figure 1(b-c) illustrates, the intermediate groups and the hierarchy of subgroups correspond exactly to the disjoint forest sets generated by Kruskal's algorithm. The MST can be computed efficiently in $O(E \ln V)$, where V is the number of individuals and $E = O(V^2)$ is the number of edges of a complete graph of V nodes. where the edges of the complete graph is the upper diagonal matrix of M.

Result of the MST clustering, *i.e.,* the disjoint forest set provides a naive hierarchical group representation, Figure 1(c). In this representation, an individual can be assigned to many groups in the hierarchy. Group attributes such as geometric center, size (variance), and number of individuals can be computed at different grouping scale by tuning the threshold θ_m.

A major limitation of agglomerative clustering is that weaker connectivity is never considered in the clustering process. Figure 2 depicts an example, where five individuals a, ... e in a ring are clustered following the MST edges with weights 1, 2, 3, 4, respectively. Edge $\overline{ae}$, the clos-

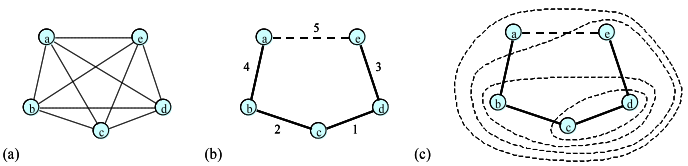

Figure 2. **Limitation of agglomerative clustering following the MST scheme.** (a) complete graph of five individuals; (b,c) result of the MST clustering is not optimal, since edge $\overline{ae}$, the closest path between a and e, is not part of the MST.

est path between a and e is never considered in the MST grouping process.

4.2. Hierarchical Divisive Clustering Using Modularity Cut

An alternative approach towards group analysis is to go top-down, to recursively divide the individuals into subgroups in a way that individuals with strong connections are placed in the same group. From a graph-theoretic perspective, the problem is to divide the complete graph containing all individuals as nodes into subgraphs in a way that maximizes within-group connections and minimizes between-group connections. A closer look at the problem should reveal the applicability of several well-known spectral clustering techniques [14, 17, 19, 20]. Most of these techniques approach the problem by looking for divisions that minimize the connections between subgraphs, or in other words, divisions that minimize the *cut size* [19].

In this paper, we apply a developed technique in social network analysis [21] to the problem of group structure formulation. We propose that instead of using cut size as the criterion, we adopt an approach originally proposed by Newman [12, 11] in the domain of social network study. Newman argued that using cut size as the division criterion is counter-intuitive to the concept of social group and that one instead needs to maximize the *modularity measure* [12, 11, 13], which expresses the difference between the actual and expected connections of individuals within each social group. Inherently, since individuals group together because of common social characteristics, such a modularity measure appropriately captures group analysis from a social perspective.

Consequently, two individuals, i and j, are strongly connected only if their connection B_{ij} is stronger than what would be *expected* between any pair of individuals, that is,

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}, \tag{1}$$

where A_{ij} is the connection strength between i and j, k_i and k_j are the total connection strengths of i and j (*i.e.*, $k_i = \sum_j A_{ij}$), and $m = \frac{1}{2}\sum_{ij} A_{ij}$ is the total strength of all connections in the complete graph. The term $\frac{k_i k_j}{2m}$ represents the expected edge strength, so that the further an edge (A_{ij}) deviates from expectation, the stronger the connection. From Eq. 1, the *modularity measure* Q, can be derived as

$$Q = \frac{1}{2m} \sum_{\substack{i,j \in \\ \text{same group}}} B_{ij} = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}, \tag{2}$$

where $\mathbf{s}$ is a labeling vector whose element s_i corresponds to an individual (node) in the complete graph. $s_i = +1$ if node i is assigned to the first group and $s_i = -1$ if node i is assigned to the second. $\mathbf{B}$ is the modularity matrix whose elements are B_{ij}. Thus, each time we divide a graph into two subgraphs, as opposed to "simply" minimizing cut size, we maximize modularity Q using $\mathbf{B}$.

Determining $\mathbf{s}$ that maximizes Q is shown to be NP-hard [11]. However, one can closely approximate the optimal solution by deriving the eigen decomposition $\mathbf{B} = \sum_i \beta_i \mathbf{u}_i \mathbf{u}_i{}^T$ with eigenvalues β_i and eigenvectors $\mathbf{u}_i$, and assigning s_i to $+1$ if the corresponding element in the maximum eigenvector is positive, and -1 otherwise. This has been shown in [11] to work well in practice.

The strategy for dividing a group into two subgraphs can be applied recursively to divide a group into an arbitrary number of hierarchical subgroups. To do so, we first define a $n \times c$ binary matrix $\mathbf{S}$, where n is the number of nodes in the complete graph and c is the number of groups. We begin with $c = 1$, i.e., there is only one group (the entire graph). As c increases, we recursively divide the graph into multiple groups. The $(i,j)^{th}$ element of $\mathbf{S}$ is 1 if node i belongs to j, and 0 otherwise. The modularity can be equivalently measured as

$$Q = \mathbf{Tr}(\mathbf{S}^T \mathbf{B} \mathbf{S}), \tag{3}$$

where $\mathbf{Tr}$ represents the trace operator. Based on Eq. 3, the strategy for dividing into multiple groups is as follow. Each time we obtain a new group, we generate a new community structure matrix $\mathbf{S}'$ with an additional column corresponding to the new group. Denoting the modularity for $\mathbf{S}'$ as Q' and the largest Q in the recursion so far as Q_{max}, the contribution, ΔQ, to the modularity measure is simply

$$\Delta Q = Q' - Q_{max}, \tag{4}$$

such that if $\Delta Q \leq 0$, the new group is "discarded" and the stopping criterion met.

In this top-down, cut-based group clustering scheme, group attributes such as center, size, members of a group are explicit. Given arbitrary configuration of individuals under tracking, the modularity cut stops when no better cut can be found. There is no parameter required in applying the modularity cut, which is very suitable to monitor rapid changes of group-level and individual interactions in this work.

The grouping of targets is performed on a per-frame basis. We explicitly keep track and maintain the temporal history of all groups and their members. Our system is thus capable of reasoning over the history of 'split-and-merge' of groups over time to reinforce the putative group behaviors. We will compare the performance of both agglomerative and divisive clustering for group-level event recognition in Section 6.

5. Group Activity Recognition

As described in section 3.2, the aim of this paper is to detect, recognize and even predict group behaviors. Based on data observations and feedback from domain experts, the

Event category	Events
Group detection	Formation, Dispersion, Distinct Groups
Motion pattern	Loitering, Fast Moving, Approaching, Following
Behavior event	Flanking, Agitation, Aggression

Table 1. **List of detectable group-level events.**

events and activities addressed by our system include low-level ones such as (i) group formation, (ii) group dispersion, and (iii) loitering. We also propose to detect semantically more advanced activities including (iv) approaching, (v) flanking, and (vi) aggression/agitation within groups, see Table 1. In the following we will describe a subset of the above events in more detail.

The event of *loitering* is detected by analyzing the standard deviation of the group location across a time window ($T_{\text{loi}} = 10$ sec). The group location for a given time is given by average ground location of all members in the group. If both components of the standard deviations are below a given threshold ($\tau_\sigma = 0.5$ m), a loitering event is generated. Based on *loitering* a second related event is defined, namely the *distinct groups* detection. Distinct groups are defined as pairs of loitering groups that maintain a stable membership set for a period of time ($T_{\text{dis}} = 2.0$ sec) and are within a certain reach of each other ($d_{\text{dis}} = 10.0$ m). The presence of multiple distinct groups indicates an increase in overall intra-group cohesion, which in turn raises the possibility of inter-group conflict. See Figure 3.

The event *group formation* is detected by counting the ancestors of a group within a certain time window ($T_{\text{gf}} = 3$ sec). A group has to form from at least three ancestors. And the ancestor groups must have existed for a minimal amount of time ($T_{\text{exist}} = 2$ sec). No further constraints need to be imposed on spatio-temporal relationships. The event of *group dispersion* is similar.

The event of *group flanking*, or flanking maneuver (groups surrounding another group prior to an attack) is aimed at detecting a certain spatio-temporal configurations that is exhibited by groups before they engage in aggressive behaviors (see Figure 4). Data seems to indicate that an aggressive and dominating (in terms of strength and numbers) group tends to "surround" the victim group or individual or at least spatially spread out before the event. Flanking is detected as follows.

We denote with $G = \{G_i, i = 0, \ldots, N_g - 1\}$ the set of all groups and with $G_i = \{T_{ij}, j = 0, \ldots, N_t^i\}$ the set of all tracked individuals in group G_i. We denote with $T = \{T_{ij}\}$ the set of all individuals. Furthermore, let $\mathbf{X_i}$ and $\mathbf{X_{ij}}$ be the locations of G_i and T_{ij} respectively. We now consider all triplets (G_i, T_{jk}, T_{lm}) and consider group G_i to be flanked if the following conditions are met:

- T_{jk} and T_{lm} are either in same group (i.e., $j = l$) of size $N_f := N_j$ or are direct descendants of the same group (of size N_f) and this group is not an direct relative of G_i.
- Group size N_f is at least 2.
- The angle between $\mathbf{X_{ijk}} = \mathbf{X_{jk}} - \mathbf{X_i}$ and $\mathbf{X_{ilm}} = \mathbf{X_{lm}} - \mathbf{X_i}$ exceeds a minimal angle θ_{fl}.
- The distance $d_{jklm} = ||\mathbf{X_{jk}} - \mathbf{X_{lm}}||$ between T_{jk} and T_{lm} must exceeds $d_{ijklm} = (||\mathbf{X_{ijk}}|| + ||\mathbf{X_{ilm}}||)/\mathbf{2}$.

The detection of *aggression/agitation* is different than the group-level events introduced so far, in that it does not operate on top of the tracking system but rather operates independently to detect image regions that contain agitated or aggressive behaviors. Only the observation of the spatio-temporal movements of individuals and groups is not sufficient to determine if agitated behavior is being exhibited. Rather, we perform sparse feature tracking using in the *foreground* of the scene and classify aggression/non-aggression based on features extracted from these tracks. We utilize the FAST feature detector developed by Rosten and Drummond [15] for low-level point detection. To obtain trajectories for the detected points, we developed a data association-based point feature tracker that utilizes a fast greedy approach to perform detection to track association. After the tracking step is done, every trajectory is analyzed with regards to a range of motion attributes and the attributes are accumulated in local "decision blocks" (of size 16×16). The per-block features are then classified according to a learned agitation model. We utilize a Support Vector Machine trained on a small number of example sequences to obtain an optimal classification. Figure 5 shows a set of examples where aggression was detected in scenarios that were enacted by correctional officers. It should be pointed out that unlike the tracking system, the agitation detection operates on each camera view independently. See figures 5 and 6 for examples of successfully detected events in the prison dataset presented in this work as well as the BEHAVE dataset [1].

6. Experiments and Results

We are presenting first experimental results on detecting behaviors in challenging multi-camera surveillance environments with a focus of detecting (i) the presence of gangs, (ii) the prediction of a possible fight, and finally (iii) the detection of agitated motion patterns that are indicative of a fight.

Figure 7 shows a sequence lasting about 1 minute where two smaller groups (a gang approaching from different angles) is approaching and then attacking two individuals. The correct sequence of events was recognized. The event evolves very quickly and there is only a gap of about 1.5 seconds between the detection of the flanking maneuver and the onset of the fight. Figure 8 shows a similar scenario

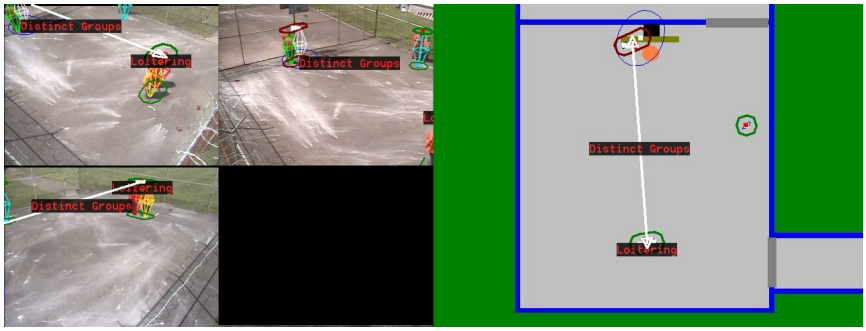

Figure 3. **Detection of Distinct and Loitering Groups.** Distinct groups are detected. (Left) Three views from synchronized cameras (one view masked out for privacy reasons). (Right) A fused planar view from the top, where distinct groups are detected and a loitering group is highlighted. Note the advantage of using a multi-view camera tracking system. Even though one of the distinct groups is outside each of the three camera views, the system still successfully detect such events. The shown activities have been *enacted* by law enforcement and corrections personnel.

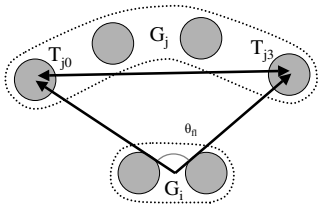

Figure 4. **Flanking Detection.** An event where one group is surrounding another group. Left: Example of a real-life flanking event in three synchronized views and one zoom-in view. Middle: A planar top view. Right: Schematic model of the flanking event. The shown activities have been *enacted* by law enforcement and corrections personnel.

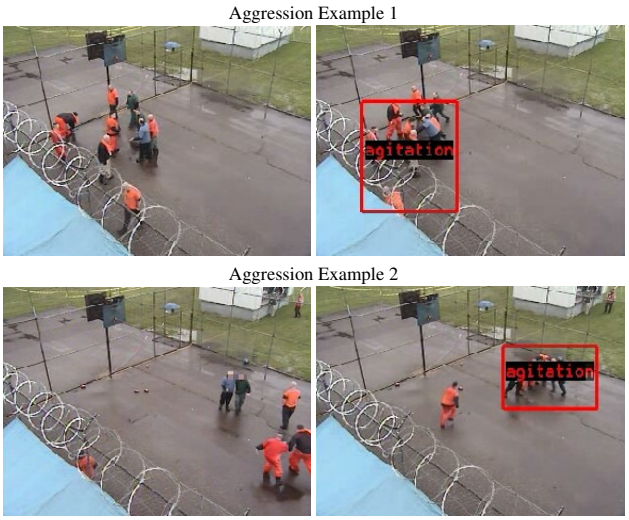

Figure 5. **Aggression Detection.** Examples of regions that were classified to contain agitated or aggressive behavior. Approximately 2 seconds elapse between the frames on the left and the frames on the right. The shown activities have been *enacted* by law enforcement and corrections personnel.

where a single group is approaching and then attacking another. The system again detected the correct sequence of events (only flanking and fighting shown) and predicts the onset of a fight three seconds before the actual fight breaks

Figure 6. **Aggression Detection.** Example aggression events detected in the BEHAVE dataset [1].

out. It should be noted that the same result was obtained with the system capturing *and processing* the scenario live on-site during the enactment of the scenario.

Figure 9 shows a more complex scenario, lasting 3 minutes in which two gangs engage in an argument and after some discussion in separate corners of the recreation yard one gang decided to attack the other. The system again managed to detect the key components of this event and predicts the onset of the fight 1.5 seconds before it begins.

Figure 10 shows a comparison of the bottom-up MST grouping approach with the top-down modularity cut scheme. We found that the proposed modularity cut is superior in separating groups during close interactions, an essential ingredient in analyzing small-scale changes in group structures. This encourages a further investigation into the

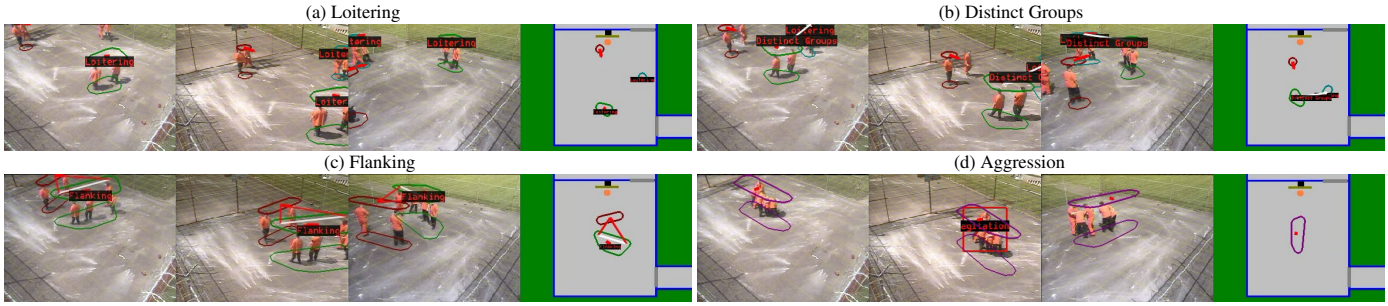

Figure 7. **Two Groups Attacking Another.** The images from top to bottom show the detection of (a) loitering, (b) distinct groups, (c) flanking, and (d) aggression. The shown activities have been *enacted* by law enforcement and corrections personnel.

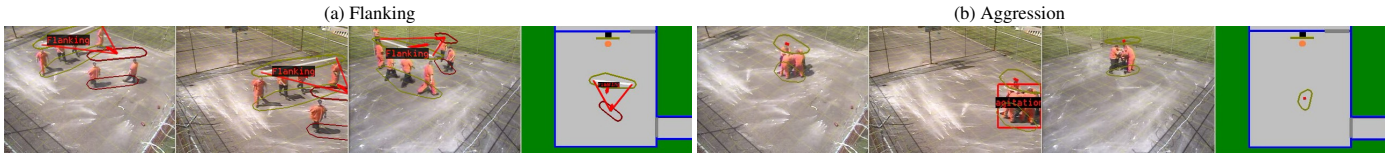

Figure 8. **Groups Attacking Another.** The images from top to bottom show (a) flanking and (b) aggression detection. The shown activities have been *enacted* by law enforcement and corrections personnel.

use of our method.

The system presented here is able to perform the presented functions live and in real-time on a single quad-core workstation. In addition to video capture, processing and display it performed PTZ camera targeting and encoded all video to disk. The frame rate of the system typically varied between 22 Hz and 10 Hz, where most CPU cycles were consumed by the foreground-background segmentation and the feature tracker for the agitation detection.

7. Conclusions

This work aims at addressing the challenging problem of detecting suspicious and disorderly behaviors in complex environments where frequent social interactions occur. To tackle this challenge we utilize a sophisticated multi-camera multi-target tracking system that is able to track individuals even under crowded conditions. To establish an understanding of behaviors we perform a group level analysis of tracks of individuals. This allows the system to reason about events at a group level. In this particular work we presented a solution to detecting a variety of low level events of interest and in particular showed how the system is able to both predict as well as detect the onset of fights between groups of individuals. Future work will provide a more thorough quantitative analysis of the presented work and will investigate a probabilistic formulation of grouping and scenario-specific event detection.

Acknowledgement. The data collections described in this work show activities *enacted* by law enforcement and corrections officers. This project was supported by grants #2007-RG-CX-K015 and #2009-SQ-B9-K013 awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

References

[1] Behave dataset. *http://homepages.inf.ed.ac.uk/rbf/BEHAVE/*. 5, 6

[2] B. Bose, X. Wang, and E. Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *IEEE CVPR*, 2007. 2

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001. 3

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., 2nd ed., 2001. 1, 3

[5] A. Ess, B. Leibe, K. Schindler, and L. van Gool. Robust multiperson tracking from a mobile platform. *IEEE Trans. PAMI*, 31(10):1831–1846, 2009. 2

[6] W. Ge, R. T. Collins, and B. Ruback. Automatically detecting the small group structure of a crowd. In *WACV*, pages 1–8, 2009. 1, 2

[7] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci USA*, 99(12):7821–7826, June 2002. 1

[8] E. T. Hall. *The Hidden Dimension*. Anchor, 1966. 1

[9] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, and R. Collins. Multi-view detection and tracking of travelers and luggage in mass transit environments. In *PETS*, 2006. 2

[10] B. Lau, K. O. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, 2009. 1, 2

[11] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006. 1, 4

[12] M. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci*, 103(23):8577–8582, 2006. 4

[13] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 2004. 4

Figure 9. **Large Group Attack.** Detection of (a) loitering, (b) distinct groups, (c) flanking, and (d) aggression in a large complex interaction between two gangs. The shown activities have been *enacted* by law enforcement and corrections personnel.

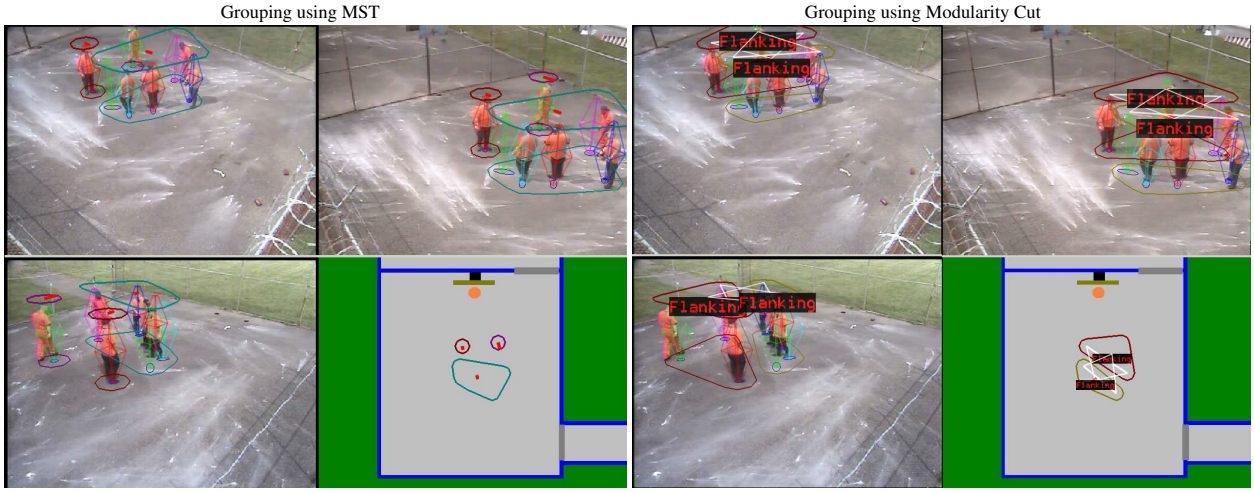

Figure 10. **Flanking - Comparison of Results.** This figure illustrates that in certain scenarios, the use of divisive clustering (modularity cut) is more advantageous than agglomerative clustering (MST), due to the ability of adaptive clustering. In this case, the MST clustering does not trigger a flanking event, while the modularity cut clustering correctly identifies the grouping structure and the event. The shown activities have been *enacted* by law enforcement and corrections personnel.

[14] A. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Info. Proc. Systems 14*, pages 849–856. MIT Press, 2001. 4

[15] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE ICCV*, pages 1508–1515, 2005. 5

[16] S. Saxena, F. Brémond, M. Thonnat, and R. Ma. Crowd behavior recognition for video surveillance. In *ACIVS*, pages 970–981, Berlin, Heidelberg, 2008. Springer-Verlag. 1, 2

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, August 2000. 4

[18] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu. Unified crowd segmentation. In *ECCV*, 2008. 2

[19] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007. 4

[20] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE ICCV*, pages 313–319, 2003. 4

[21] T. Yu, S. Lim, K. Patwardhan, and N. Krahnstoever. Monitoring, recognizing and discovering social networks. In *Proc. IEEE CVPR*, 2009. 2, 4

[22] T. Yu, Y. Wu, N. Krahnstoever, and P. Tu. Distributed data association and filtering for multiple target tracking. 2