Information Sciences 447 (2018) 83–103

Contents lists available at ScienceDirect

Information Sciences

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

A multi-temporal framework for high-level activity analysis: Violent event detection in visual surveillance

Donghui Song[a,b], Chansu Kim[a,c], Sung-Kee Park[a,b,*]

[a] Center for Robotics Research and Convergence Research Center for Diagnosis, Treatment and Care System of Dementia, Korea Institute of Science and Technology, Seoul 02792, Republic of Korea
[b] HCI and Robotics, Korea University of Science and Technology, Daejeon 34113, Republic of Korea
[c] Department of Electrical and Computer Engineering, Korea University, Seoul 02841, Republic of Korea

ARTICLE INFO

Article history:
Received 20 April 2017
Revised 14 December 2017
Accepted 28 February 2018
Available online 3 March 2018

Keywords:
Computer vision
Multi-temporal framework
High-level activity analysis
Violent event detection
Late fusion
Visual surveillance

ABSTRACT

This paper presents a novel framework for high-level activity analysis based on late fusion using multi-independent temporal perception layers. The method allows us to handle temporal diversity of high-level activities. The framework consists of multi-temporal analysis, multi-temporal perception layers, and late fusion. We build two types of perception layers based on situation graph trees (SGT) and support vector machines (SVMs). The results obtained from the multi-temporal perception layers are fused into an activity score through a step of late fusion. To verify this approach, we apply the framework to violent events detection in visual surveillance and experiments are conducted by using three datasets: BEHAVE, NUS–HGA and some videos from YouTube that show real situations. We also compare the proposed framework with existing single-temporal frameworks. The experiments produced results with accuracy of 0.783 (SGT-based, BEHAVE), 0.702 (SVM-based, BEHAVE), 0.872 (SGT-based, NUS–HGA), and 0.699 (SGT-based, YouTube), thereby showing that using our multi-temporal approach has advantages over single-temporal methods.

1. Introduction

Human activity analysis from video surveillance has attracted considerable attention in recent decades because this technology is able to increase the intelligence of surveillance systems by recognizing human activities and autonomously providing alarms in abnormal situations.

Human activity can be classified as low- and high-level activity based on its complexity and temporal scales [38]. Low-level activity analysis is a well-studied problem; therefore, there are some practical and commercial applications of video analytics that can handle low activity levels (e.g., enter, exit, appear, and loiter) for surveillance systems. In the case of high-level activity analysis such as detecting violent events (e.g., group fighting), it remains an unsolved problem; hence, practical applications do not yet exist because of the complexity and diversity of activity appearance.

In previous work on high-level activity analysis, researchers mainly focused on the following factors: the design of highly discriminative and robust features with learning methods [5,8,11–13,25,28,30,34,36,45], or a hierarchical/semantic model of human knowledge about activities with statistical or contextual methods [2,4,7,15,16,18,19,21,29,32]. Although these studies

* Corresponding author at: Center for Robotics Research and Convergence Research Center for Diagnosis, Treatment and Care System of Dementia, Korea Institute of Science and Technology, Seoul 02792, Republic of Korea. Tel.: +82 2 958 5626; fax: +82 2 958 5629.
E-mail addresses: donghui@kist.re.kr, donghui@ust.ac.kr (D. Song), csk@kist.re.kr (C. Kim), skee@kist.re.kr (S.-K. Park).

https://doi.org/10.1016/j.ins.2018.02.065
0020-0255/

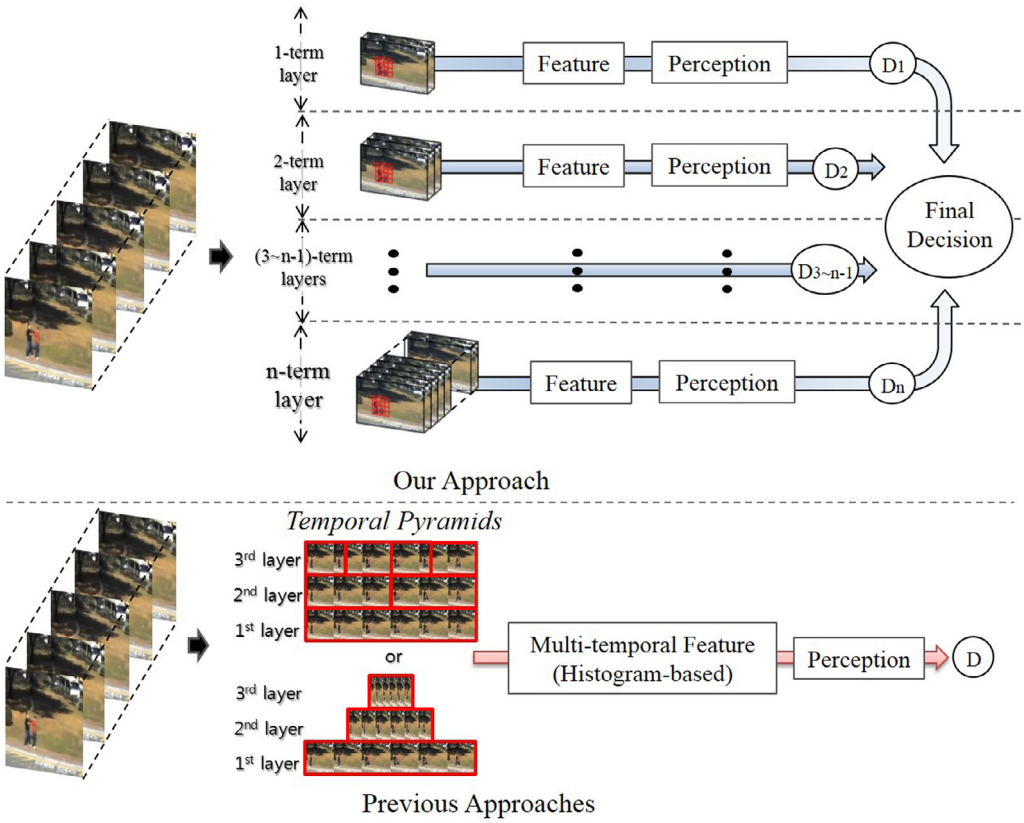

Fig. 1. Schematic comparison of our approach with previous approaches [26, 33, 46]. Our approach: late fusion with multi-independent temporal perception layers. Previous approaches (examples of three layers): early fusion with a perception model (red boxes indicate sequences are used to extract features at each layer). The proposed framework consists of multi-independent perception layers, each with a different time scale. Each decision (Di) is made by a perception model which controls i-term temporal duration. Lastly, decisions are fused into a final decision through late fusion.

overcame its complexity, most of them disregarded temporal diversity by using a certain/fixed time scale for analysis. In addition, their performance is affected by the temporal scale since their target activities cannot be defined by a certain time duration. This condition was presented in [1], where the problem of defining an appropriate temporal scale relative to temporal diversity continues to persist.

Some researchers tried to deal with this temporal diversity by aggregating multi-temporal features into a histogram-based feature vector, known as a temporal pyramid. This temporal pyramid, which is based on early (i.e., feature-level) fusion, can be built in two distinct ways as shown in Fig. 1: an approach is to reduce time scales of the first layer by half and the other method is to compress the entire sequence of the first layer into the half size of the original length as the level of pyramid increases. These works successfully proved that the temporal pyramid was useful for finding similar activities or video categories [26,33,46]. However, this early fusion is less discriminative than late (i.e. decision-level) fusion in representing the features' information of each layer [24]. This is because the temporal diversity disappears when early fusion combines multiple features into one feature vector. In contrast, late fusion is a way to use multiple perception models and each model is designed to accurately represent each feature; therefore, late fusion can be an efficient approach to high-level activity analysis in terms of using multiple time series to cope with the temporal diversity.

In this paper, we propose a novel framework of high-level activity analysis based on late fusion using multi-independent temporal perception layers. For this approach, various time scales are used as they were similarly used in the previous studies; but unlike before, those are being used independently and analyzed separately through multi-independent temporal perception layers as illustrated in Fig. 1. As an example case of high-level activity, it is often difficult to detect group fighting because their activities consist of various types and movements between people that have different time scales (e.g., punch and kick in a short period of time, wrestle and tussle in a long period of time). Our proposed method is able to detect these characteristics since each perception layer makes each decision within its time period (Ds in Fig. 1), whereas the previous temporal pyramids using one fused histogram-based feature vector may miss these characteristics.

The proposed multi-temporal framework consists of three parts: Multi-temporal Analysis (MtA), Multi-temporal Perception Layers (MtPLs), and late fusion. The MtA generates multiple temporal feature descriptors and the MtPLs, having multiple layers for different time scales but using the same perception method in each layer, analyze activity as an "activity score" for

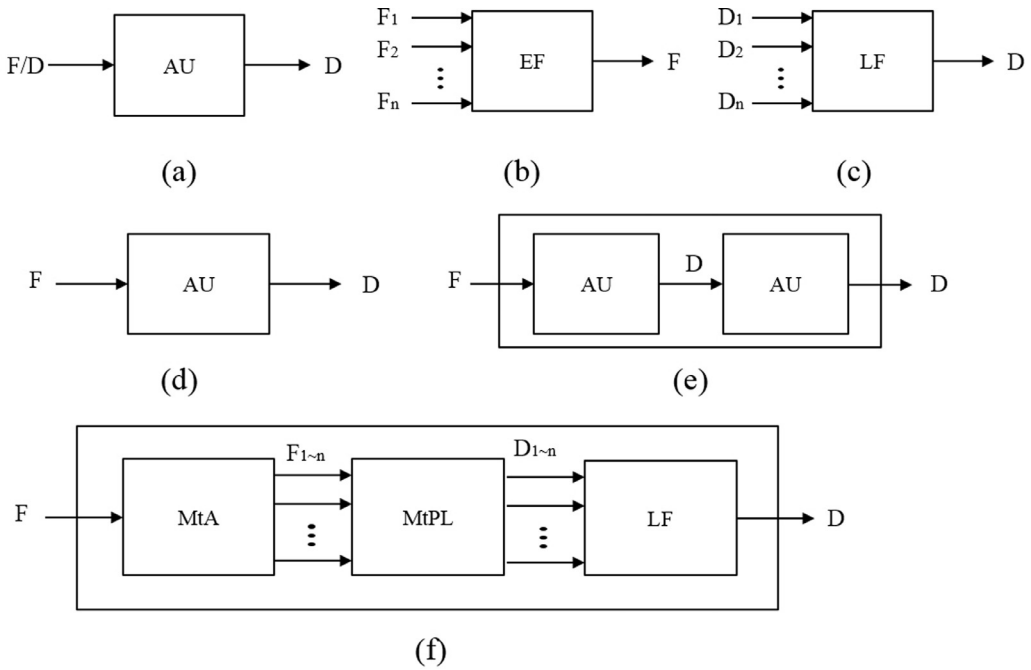

Fig. 2. Diagrams of human activity analysis and fusion methods: (a) analysis unit. (b) early fusion unit at feature level. (c) late fusion unit at decision level. (d) single-layered approach for activity analysis. (e) hierarchical approach. (f) proposed multi-temporal approach. F indicates feature and D is decision.

each target object (in this case, a group of humans). For the perception model of MtPLs, we built both semantic and learning methods: situation graph trees (SGT) and support vector machines (SVMs). All these scores from multiple layers are fused into one score through the late fusion step. We verify its efficiency to analyze high-level activity by applying the proposed framework to both multiple and monocular camera surveillance scenarios to detect violent events (i.e., group fighting).

The remainder of this paper is structured as follows: In Section 2, we introduce related work on human activity analysis, multiple temporal studies in video analysis, and fusion strategy. In Section 3, we propose our framework for high-level activity analysis. In Section 4, we introduce an SGT-based and SVM-based multi-temporal framework to detect violent events in multi-camera surveillance. Our experiments to test our framework and the results are described in Section 5. Finally, we offer our conclusions in Section 6.

2. Related work

2.1. Human activity analysis

Aggarwal and Ryoo [1] classified human activity analysis into two categories: single-layered and hierarchical approaches. The single-layered approach uses a single analysis unit (Fig. 2(d)) and covers the simple and short-term activities (low-level activity) because it analyzes human activities directly based on certain sequences of images. Therefore, these methods focus on the design of highly discriminative and robust features such as bag-of-words (BoW) [35] of spatio-temporal features. The BoW representation of spatio-temporal features, such as space-time interest points (STIP) [25] with a classifier, is widely used, and has shown satisfactory performance for activity recognition in the video domain. Burghouts and Schutte [8] adopt a pipeline consisting of STIP features, BoW, and an SVM classifier and proposed a spatio-temporal layout of human activity. Dollar et al. [13] developed a general framework for detecting and characterizing activity from video based on cuboids of spatio-temporal features. Niebles et al. [30] presented an unsupervised learning method by representing a video as a collection of spatio-temporal words. They used latent topic models such as the probabilistic Latent Semantic Analysis (pLSA) model and Latent Dirichlet Allocation (LDA). Thi et al. [36] applied Hierarchical Bayesian Feature Selection (HBFS) for spatio-temporal features, SVM for action classification, and Dynamic Conditional Random Fields (DCRF) and Structured SVM (SSVM) for action localization. Chen et al. [11] used SVM with spatio-temporal fisher vector coding. Ni et al. [5] recognized human group activity with three types of localized causalities: self-, par-, and group-causality as a way of BoW and tested the effectiveness of the proposed feature by using SVMs and Nearest Neighbor (NN). Recently, instead of using handcrafted features, approaches which are automatically learn spatio-temporal features are studied. Liu et al. [28] used genetic programming (GP) to evolve spatio-temporal feature descriptors that fuse color and motion (i.e., optical flow) information. Wu et al. [40] developed Deep Dynamic Neural Networks (DDNN) for multimodal gesture recognition. To learn spatio-temporal representations, they used deep learning methods such as a Gaussian–Bernouilli Deep Belief Network (DBN) and a 3D Con-

volutional Neural Network (3DCNN). The DBN was applied to handle skeletal dynamics and the 3DCNN was intended to manage and fuse depth and RGB data.

However, there were also other single-layered approaches except those spatio-temporal features-based methods. Yacoob and Black [42] proposed a framework to model and recognize people's activities over time, such as four types of walking and movements of the mouth, by using Principal Component Analysis (PCA), linear transformations, and by Eigen matching observations to exemplars. Wilson and Bobick [39] developed a method for gesture recognition using parametric hidden Markov models. Vaswani et al. [37] proposed a method to detect abnormalities rather than actual human activities based on statistical shape analysis of the location of a moving object using polygonal shapes. Yamato et al. [43] adopted a hidden Markov model (HMM) to represent and recognize human activities while HMMs have been widely used for speech recognition. Oliver et al. [31] presented a coupled HMM for interaction level activity between two people.

These single-layered methods are most effective when a target sequence can be characterized from training sequential patterns. In addition, image sequences are represented as robust features (e.g., STIP-based BoW features), which are successful in analyzing human activity, based on a sliding windows technique. Because of these characteristics, it is difficult for single-layered approaches to recognize activity with temporal diversity (e.g., group fighting), especially, in the real-time streaming video domain (i.e., surveillance). The categorization of target video is easier for these methods.

Some researchers have focused on hierarchical approaches: statistical and semantic methods. Statistical approaches use multiple layers of state-based models (usually two layers) such as HMM and dynamic Bayesian networks (DBNs). The lower layer makes decisions for atomic activities from the sequence of a feature vector and the higher layer uses a sequence of atomic activities as input for the analysis of higher activity. For example, Duong et al. [15] introduced the S-HSMM, an extension of the hidden semi-Markov model (HSMM), to process the recognition of human activities in daily life, and proposed a scheme to detect abnormalities without using training data. Nguyen et al. [29] proposed an application of the shared-structure HHMM for representing and recognizing indoor activities. Park and Aggarwal [32] applied a hierarchical Bayesian network (BN) to recognize two-person interactions, such as hugging, pointing, punching, pushing, and kicking, based on the poses of multiple body parts. Atrey et al. [4] applied a hierarchical probabilistic assimilation method to detect atomic and compound events for multimedia surveillance systems. For semantic and hierarchical approaches, Elhamod and Levine [16] introduced a framework to detect potentially suspicious behavior in public transport areas by using rule-based models. Bremond et al. [7] proposed a framework based on AND/OR trees with finite state automations for recognizing fighting, blocking vandalism, overcrowding, and fraudulent activities. Ghanem et al. [19] used PN for surveillance in a parking lot. Albanese et al. [2] proposed the concept of probabilistic Petri nets, which is an extension of PN. Haag et al. [21] used Fuzzy Metric Temporal Logic (FMTL) and situation trees for the recognition of traffic situations. Fernandez et al. [18] defined SGT as an extension of situation trees, used FMTL, and proposed top-down event modeling and bottom-up event inference frameworks.

These hierarchical methods are reasonable to recognize scenario-based events using simple data concerning objects (i.e., actors), such as trajectories, locations, speeds, and variations in object size. Hence, it appears that the hierarchical approach is more suitable than a single-layered one for surveillance systems [9]; however, determining the minimum time period for a specific atomic event is also critical for this approach [4] and detecting violent events, such as small group fights, is difficult because of the characteristics of high-level activity: high complexity and long temporal scale.

2.2. Multiple temporal extents in video analysis

Both Pirsiavash et al. [33] and Zelnik-Manor and Irani [46] used multiple temporal features known as a *temporal pyramid*. This *temporal pyramid* segmented the entire video sequence along different temporal scales. The former research applied it to detect activities in first-person camera views and the latter used it for event-based indexing into video. Laptev et al. [26] employed spatio-temporal features and generalized spatial pyramids to the spatio-temporal domain. They evaluated the performance of different spatio-temporal grids and verified that combining those multiple grids (two or three) showed improvement over the one best grid through action classification experiments. Typically, these studies about pyramids of the temporal and spatio-temporal domains succeeded in taking advantage of the analysis of multiple temporal scales. However, those pyramids were all about creating robust features for matching a query to a target or training a classifier. In other words, the above-mentioned studies belong to the single-layered approaches of activity analysis and have same disadvantages because of that. In addition, this approach also introduces some weaknesses of early fusion, which are explained in the next section. Based on these observations, we try to analyze human activity in the multiple temporal domain by addressing problems found in previous work.

2.3. Early and late fusion

Generally, there are two types of fusion: early and late fusion (Fig. 2(b) and (c)). Early fusion is carried out at a feature level at which one fused vector is a concatenation of different features, whereas late fusion approaches fuse various classification scores into one score at a decision level after multimodal classification results are obtained [14].

For image and video analysis, fusion approaches are usually used to combine characteristics of different modalities. For example, Cheng et al. [12] used multiple visual features: motion and appearance, and fused those features into a normalized histogram of visual word occurrences for recognizing human group activities. They performed the state-of-the-art accuracy

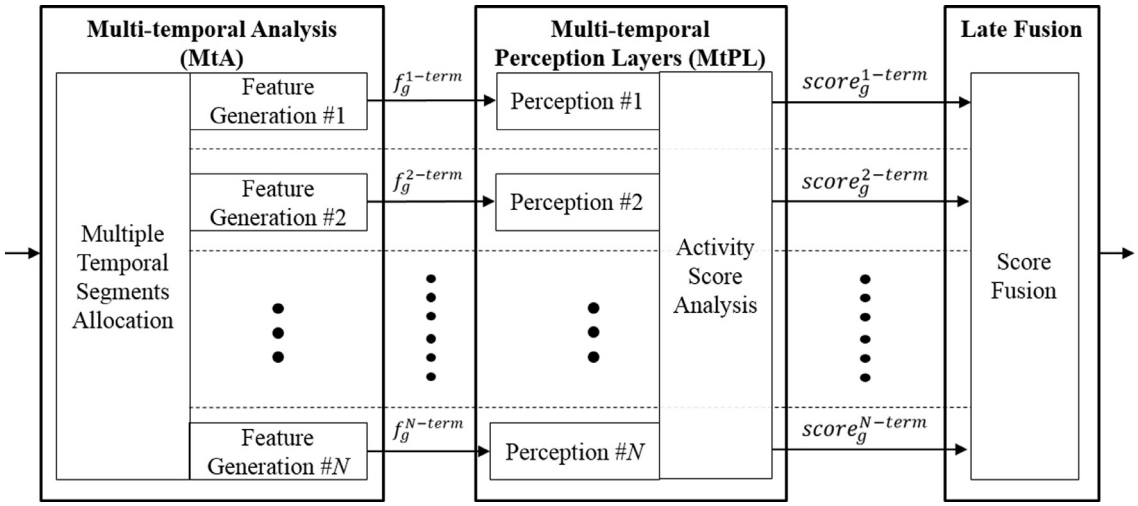

Fig. 3. Multi-temporal framework consisting of three stages: MtA, MtPL, and late fusion with multiple (N) layers based on different temporal scales.

in the BEHAVE followed by [27]. Yu et al. [45] proposed a novel binary local representation for RGB-D data fusion and action recognition. This local descriptor, namely Local Flux Feature (LFF), was concatenated RGB and depth data separately. Then the LFFs were fused into a Hamming space via a structure-preserving projection. Shao et al. [34] presented a method to fuse and encode different feature representations from multiple views for action recognition via Kernelized Multiview Projection (KMP). KMP was able to find a discriminate subspace to fuse all views into a feature vector. Lan et al. [24] used both fusion approaches, in the name of double fusion, for multimedia event detection. They extracted and fused visual, audio, and text features in an early fusion process and also combined the results of multiple classifiers for late fusion. Dong et al. [14] evaluated the performance of early and late fusion methods for video semantics indexing. Their features consisted of global and local features: color, texture, and edge descriptors for global features and various types of scale-invariant feature transform (SIFT) for the local features. SVMs were used as classifiers. Ye et al. [44] proposed a robust late fusion method with rank minimization and conducted experiments on various visual classification tasks. Because these studies considered and extracted different types of visual features, they required much more time to generate features of each kind, as the number of types increased. Moreover, they also encountered some challenges such as which features are useful or how to fuse them. Unlike these studies, we use only one type of feature, i.e., the most suitable to the perception method. This fact may solve the challenging issues in a simple way. However, it not only provides us with a simple solution but also gives us efficiency because this feature becomes multiple features, which have different temporal characteristics, through the multi-temporal analysis of our framework.

According to [3], the advantages and disadvantages of each fusion strategy are highlighted as follows: for early fusion, it can utilize a correlation among multiple features from various modalities at the feature level; therefore, it needs only one perception phase on the combined feature vector. However, it has a disadvantage in terms of presenting the time synchronization between the multimodal features and also these features should be in the same format to enable them to be combined before fusion. Moreover, the increase in the number of modalities leads to a problem to learn the cross-correlation among the features. The late fusion strategy has some advantages compared to early fusion. The implementation of late fusion is easier because the decisions usually have one representation, whereas the features of multimodal (e.g., video and audio) may have different representations. It also allows us to apply appropriate methods for analyzing each modality that provide more flexibility than early fusion. In addition, this decision level fusion offers scalability (i.e., graceful upgrading or degradation) in terms of modalities used in the fusion process, which is a challenge of feature level fusion. Nevertheless, late fusion also has drawbacks. For instance, this approach may fail to utilize feature level correlation between modalities and also the learning process for multimodal perception could be time consuming because these classifiers receive input from different features.

As mentioned above, we use late fusion for high-level activity analysis (i.e., violent event detection in surveillance systems) by absorbing the advantages of the decision-level fusion method, but at the same time, by overcoming the disadvantages of the fusion approach and challenges of the high-level activity analysis based on the analysis of multiple temporal extents.

3. Multi-temporal framework

The multi-temporal framework consists of three stages: MtA, MtPL, and late fusion, as shown in Fig. 3. This framework has N layers and each layer has different time scales t^{i-term}. The core of this method is using multiple time scales and ana-

lyzing them separately until each perception result is obtained. This shows that our approach has the following advantages: it is acceptable to any existing models and capable of improving the performance of these models.

3.1. Multi-temporal analysis

In this process, image sets that have time scales t_i per each i-term layer are allocated and feature vectors f_g^{i-term} are collected over the multi-temporal scales from those image sets. By using multi-temporal analysis, our framework can obtain not only short-term feature vectors, but also longer-term data. It forms the basis for the benefits of using multi-temporal frameworks simply but efficiently because it is not a fusion method for multiple features or multiple models that uses different input data, but one for sharing the same design of features and models for multi-temporal space.

3.2. Multi-temporal perception layers

In this step, predictions are made by multiple perception layers from each feature vector per temporal domain (i.e. f_g^{i-term}). Those perception models are identical to each other; therefore, a method is needed to perform activity score analysis to calculate a score of a target activity from a prediction. In summary, multiple scores ($score_g^{1-term}, \ldots, score_g^{N-term}$) of the target activity are generated from each feature vector per temporal domain (i.e. f_g^{i-term}) through MtPL, which can accept any existing perception method (e.g., a semantic or learning method).

3.3. Late fusion

Late fusion involves score fusion, which is a rule-based fusion method. The rule-based methods have various fusion strategies (e.g., linear-weighted fusion, majority voting, and custom-defined rules). These are used for face detection, object tracking, speech recognition, image and video retrieval, and person identification. In the case of violent events detection, we use a linear-weighted fusion method, which is widely used by researchers. This method performs well when the weights of each modality can be properly determined [3]. Our modalities of the multi-temporal framework are equal to each other; therefore, the fusion method is suitable to the framework and the weights are identically same for each score. Details of the implementation are described in the next section.

4. Violent event detection by using multi-temporal framework for multi-camera surveillance

As we mention above, technology for visual surveillance, which includes low- and high-level activity analysis, has become more crucial. In this section, we intend to detect violent events by using the multi-temporal framework and apply two types of perception models (i.e., SGT and SVMs) to the framework as perception model. The SGT approach uses optical-flow-based features, whereas the SVMs use STIP-based BoW features. Through MtPL, the scores of the target activity (in this case, a small group fight) are generated, and these scores fused into the "activity score" by late fusion. Data captured by multiple cameras is processed by carrying out data refinement based on look-up tables, which indicate the groups that are common to multiple camera views. The look-up tables are generated by matching group locations (Algorithm 1); these locations are warped from one camera view to others by using prior geometrical knowledge regarding each camera (see Fig. 11). Our approach for violent event detection in multi-camera surveillance is illustrated in Fig. 4.

4.1. Situation graph trees-based MtPL

4.1.1. Feature representation

Xie et al. [41] defined five kinds of human group features for violent events by using motion vectors. This is because raw motion vectors alone cannot satisfactorily distinguish between a violent and a non-violent event. However, one of the key features in their proposal, named the "coincidence indicator," was unable to distinguish between fighting and walking behavior. We use their feature vectors but re-define some of the features.

Our human group features f_g, for SGT-based MtPL, consist of five-dimensional visual features f_v and two types of time features (t_{group}, $t_{sit.}$). The visual features f_v can be calculated by a motion vector based on dense optical flow [17]:

$$f_v = [M, D_m, V, D_d, S],\ f_g = [f_v, t_{group}, t_{sit.}] \tag{1}$$

- **Movement** M: The movement of the group is the mean value of the magnitudes of the motion vectors. The motion vector of the i-th block is (V_{x_i}, V_{y_i}):

$$M = \frac{1}{N_{blocks}} \sum_{i=1}^{N_{blocks}} \sqrt{{V_{x_i}}^2 + {V_{y_i}}^2} \tag{2}$$

where N_{blocks} is the number of blocks, and the size of one block is $3 \times 3 = 9$ (the magnitude of the x-axis $\times$ the magnitude of the y-axis) and it is shown in Fig. 5.

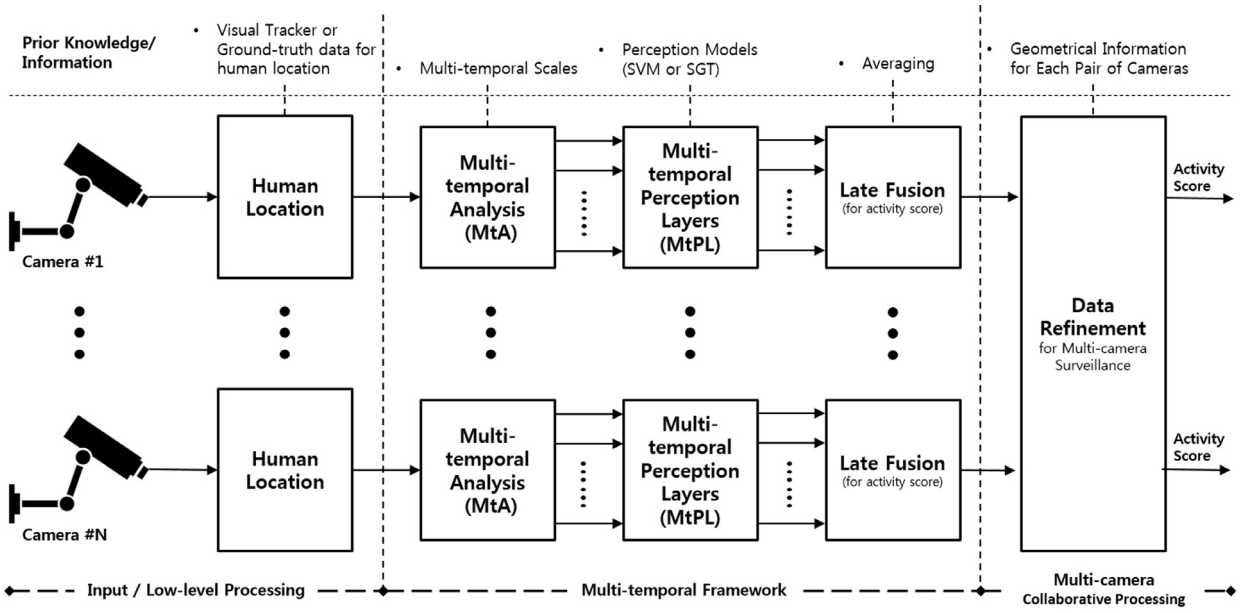

Fig. 4. Our approach to detect violent events for multi-camera surveillance systems: input/low-level processing for cameras and human locations, the multi-temporal framework, and data refinement. The requisite prior knowledge or information is also described for each module.

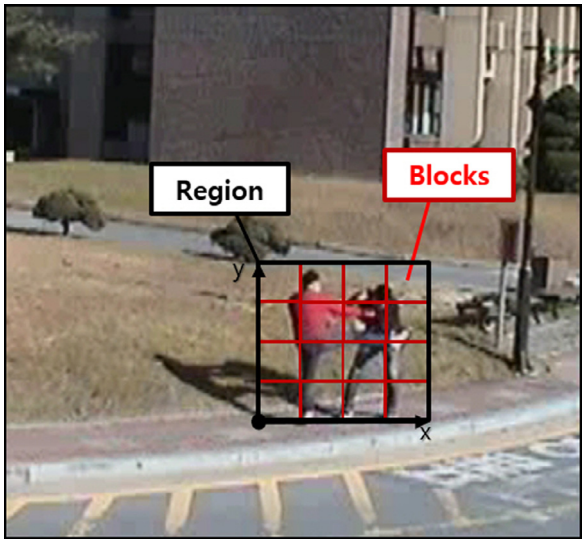

Fig. 5. Example of a human group region and its blocks to generate human group features from motion vectors based on optical flows.

- **Difference of movement** D_M: The disorder in group movement can be calculated from the ratio of the mean value of the motion vectors $\bar{M}$ to movement M:

$$\overline{V_x} = \frac{1}{N_{blocks}} \sum_{i=1}^{N_{blocks}} V_{x_i},\ \overline{V_y} = \frac{1}{N_{blocks}} \sum_{i=1}^{N_{blocks}} V_{y_i} \tag{3}$$

$$\bar{M} = \sqrt{\overline{V_x}^2 + \overline{V_y}^2} \tag{4}$$

$$D_M = \frac{\bar{M}}{M} \tag{5}$$

This value considers the magnitudes of the motion vectors and their directions. The range of values is from 0 to 1: when disorder is high, it tends toward 0.

- **Violence** V: This value uses the coincidence indicator f_c, which characterizes a phenomenon whereby relative motion among people often occurs in violent scenes [41]. We use the coincidence indicator f_c but divide it by the difference of movement D_M:

$$f_c = e^{-\lambda\sqrt{\overline{V_x}^2+\overline{V_y}^2}} \tag{6}$$

$$V = \frac{f_c}{D_M} \tag{7}$$

where λ is a coefficient set to 0.014 according to [41]. The value of violence V can be used to discriminate between fighting and walking behavior, which the coincidence indicator f_c cannot differentiate.

- **Difference of direction** D_d: We use the entropy value of the direction of the motion vectors:

$$D_d = \sum_{i=0}^{N_{angle}-1} p_i \log p_i \tag{8}$$

where, N_{angle} represents a resolution of direction, for which we use every one degree; therefore, it is set to 360. The possibility p_i represents the possibility that the motion vector points in the i-th direction [41].

- **Speed** S: The speed at which the center of the group moves from its previous location (x_{t-1}, y_{t-1}) to the present location (x_t, y_t):

$$S = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{9}$$

- **Time for group presence** t_{group}: It indicates the time that a group has been in existence since it first appeared.
- **Time for situation maintenance** $t_{sit.}$: This value represents the duration for which a group has been in the same situation, which is a node of the SGT model, as before. When a situtation of the group (i.e. a result of SGT) is different from the pervious, this time value resets to one.

For multi-temporal analysis, only visual features f_v of human group features f_g are averaged per temporal scale, and become multi-temporal visual features f_v^{i-term}:

$$f_v^{i-term} = \begin{cases} \frac{1}{t^{i-term}} \sum_{j=0}^{t^{i-term}} f_{v,t-j}, & \text{if } t^{i-term} \le t_{group} \\ \frac{1}{t_{group}} \sum_{j=0}^{t_{group}} f_{v,t-j}, & \text{otherwise} \end{cases} \tag{10}$$

where i represents varying temporal analyses and t^{i-term} represents a temporal scale for the i-term. Finally, the multi-temporal human group feature f_g^{i-term} is expressed by (11):

$$f_g^{i-term} = \left[f_v^{i-term}, t_{group}, t_{sit.} \right] \tag{11}$$

4.1.2. Perception layers

The SGT-based perception layer consists of spatio-temporal inference, contextual reasoning, and activity score analysis. The inference and reasoning are based on the work in [18,21]; however, a formula for the score analysis is defined by our work. A process of the SGT-based perception layer is illustrated in Fig. 6.

For spatio-temporal inference, FMTL is used to generate atomic predicates (symbolic data s_g), which are necessary for contextual reasoning, from quantitative data (human group feature f_g). FMTL is an extended fuzzy logic that accommodates temporal space, and its outputs (i.e., atomic predicates) are fuzzy values: *None, Low, Normal, and High*. Our FMTL rules were built by an expert for each of the human group features f_g; these rules were sheared for each temporal layer.

SGT is used as a contextual reasoning engine for scenario-dependent event recognition which generates a discrete result $event_g$ from atomic predicates s_g. Models of SGT define possible events in which agents can participate and consists of situation nodes, situation graphs, predication edges, and specialization edges. Fig. 7 shows an SGT model, which we built for group events as well as for detecting violent events (i.e., FIGHT). The details of SGT that exemplify its mechanisms to contextualize: events are hierarchically nested from general to specific by means of specialization edges forming a tree, and sequentially connected by prediction edges producing graphs within the tree [18]. That is, predictions (including self-predictions) and specializations occur when a given set of atomic predicates satisfies the demands of the model (FMTL predicts) whereby nodes are linked by these edges. Furthermore, the start/end-situation means that this situation node can represent a start or end situation within the situation graph. Actions represent what the system should do when a node is there, it is usually used to notify users of alterations. For more details about FMTL and SGT, we recommend to see [18,21].

To analyze activity score, we define a formula that uses distances within the nodes of SGT:

$$p(node_A | node_B) = \left(1 - \frac{distance_{A:B}}{max.distance_B}\right)^2 \tag{12}$$

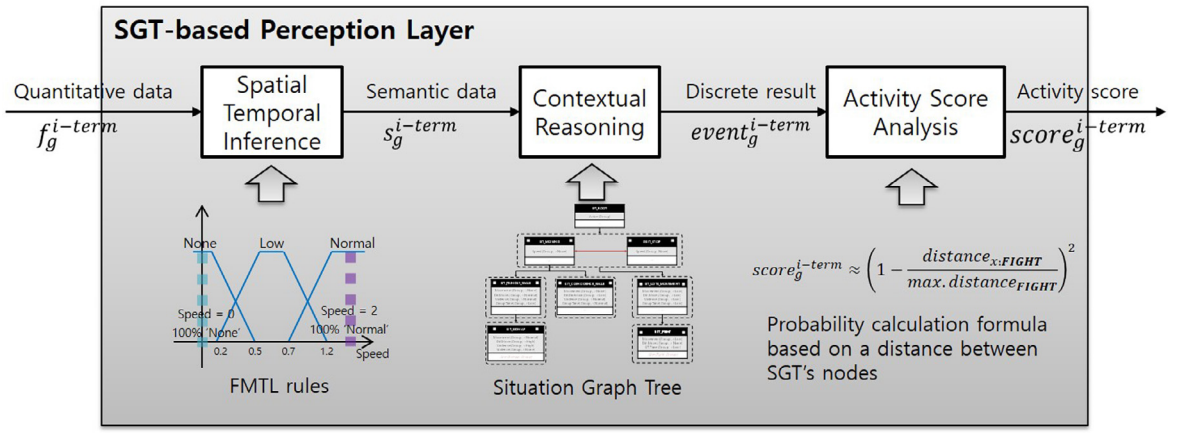

Fig. 6. Process of the SGT-based Perception Layer from human group features (quantitative data) to an activity score of a target event (in this case, a violent event/activity), consisting of: spatio-temporal inference by FMTL rules, SGT-based contextual reasoning, and activity score analysis using a probability calculation formula based on the distance between SGT nodes.

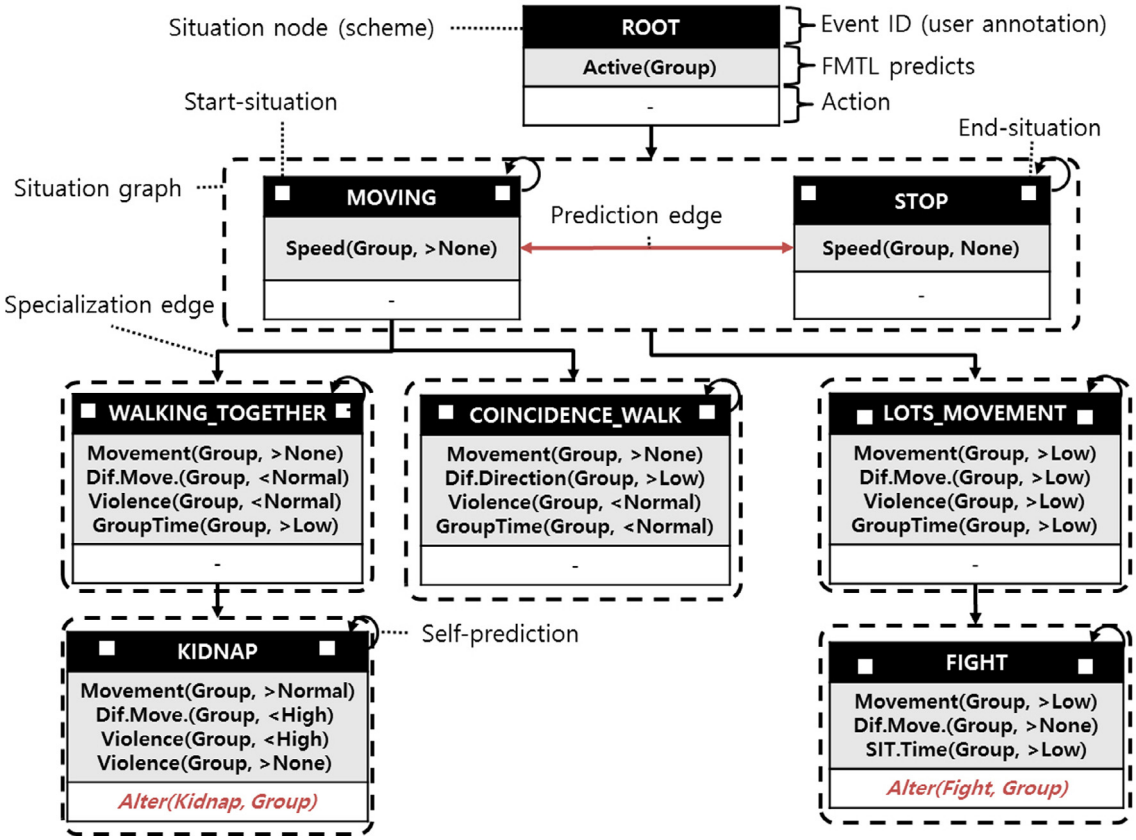

Fig. 7. SGT model for group events and its details (In FMTL predicts which are atomic predicates s_g of spatial temporal inference from human group feature f_g, some expressions are displayed in a short because of the size of situation node: Dif.Move is difference of movement, Dif.Direction is difference of direction, GroupTime is time for group presence, and SIT.Time is time for situation maintenance).

where each A and B represents an event ID of SGT models such as MOVING, STOP, FIGHT and so on (see Fig. 7), $distance_{A:B}$ is the distance between node A and B in SGT (e.g., the distance from FIGHT to STOP is 2), and $max.distance_B$ means the distance between node B and the farthest node from that node B (e.g., when B is STOP, the distance is 2; when B is FIGHT, it becomes 4).

Fig. 8. Detection of Space-time Interest Points (STIP). Left: an example of BEHAVE datasets during a violent event. Right: result of STIP detection of the left image (The points in cyan are STIP).

For target activity scores per temporal layer $score_g^{i-term}$ (in this case, the FIGHT event), the probability of being the target node (i.e. $node_{FIGHT}$) given by a node (i.e. $node_x$) which is the discrete result $event_g^{i-term}$ is described from (13) to (16):

$$p\left(node_{target=FIGHT}\middle|event_g^{i-term}=node_x\right)=p(node_{FIGHT}|node_x) \tag{13}$$

From the Bayesian rule, (13) can be (14):

$$p(node_{FIGHT}|node_x)=\frac{p(node_x|node_{FIGHT})p(node_{FIGHT})}{p(node_x)} \tag{14}$$

where $p(node_{FIGHT})$ is a prior probability that is $\frac{1}{Number\ of\ situation\ nodes}$ and $p(node_x)$ is a normalization constant which can be calculated through $\sum_i p(node_x|node_i)p(node_i)$. From (12), it becomes the below (15):

$$\frac{p(node_x|node_{FIGHT})p(node_{FIGHT})}{p(node_x)}=\frac{p(node_{FIGHT})}{p(node_x)}p(node_x|node_{FIGHT})=\mu\left(1-\frac{distance_{x:FIGHT}}{max.distance_{FIGHT}}\right)^2 \tag{15}$$

where μ is $\frac{p(node_{FIGHT})}{p(node_x)}$ which can be a constant value relative to the normalization constant p($node_x$). Finally, when the i-term SGT-based perception layer generates a result $event_g^{i-term}$ which is $node_x$, the score for violent event $score_g^{i-term}$ can be stated by (16):

$$score_g^{i-term}\approx\left(1-\frac{distance_{x:FIGHT}}{max.distance_{FIGHT}}\right)^2 \tag{16}$$

4.2. Support vector machines-based MtPL

4.2.1. Feature representation

The human group feature for SVM-based MtPL is based on interest points in the spatio-temporal domain unlike SGT-based MtPL, which is based on region. Because STIP features have been widely used, and work well with classifiers, we use the STIP in [25], which employs an extended Harris corner detector for STIP detection (see Fig. 8). For a feature descriptor, we use the Histogram of Gradients (HOG) [8,20].

For multi-temporal analysis, we use the same number of blocks for each STIP for multi-temporal layers: $3\times3\times2=18$ blocks (spatial × temporal spaces); however, the size of blocks varies in the temporal domain for each layer. The size of a block is $9\times9\times(\frac{t_i}{2}+1)$ (the number of x pixels × the number of y pixels × the number of frames). The number of dimensions of the HOG descriptor are equal to $3\times3\times2\times4=72$ (the number of spatial bins × temporal bins × gradient direction bins).

4.2.2. Perception layers

The SVM-based perception layer consists of BoW representation for STIP features, classification using a pre-trained SVM, and activity score analysis for a discrete event result (see Fig. 9). In addition, the BoW is motivated by the well-known bag-of-words representation of text documents, processes video frames as “documents,” and uses occurrence histogram features to represent a set of local features by using a “codebook,” which is generated by a large set of local features during a training phase [22]. The K-means algorithm is used for codebook quantization.

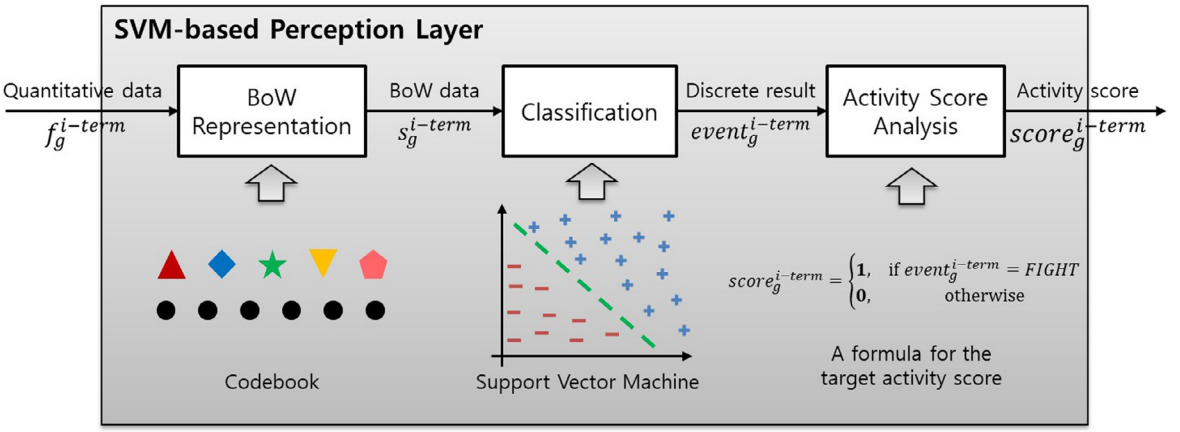

Fig. 9. Process of the SVM-based perception layer from human group features (quantitative data based on STIPs) to an activity score of a target event (in this case, a violent event/activity) consisting of BoW representation, SVM-based classification, and activity score analysis.

Table 1
Behave dataset description and comparison of ground truth between data from the dataset provider and our data.

Multi-camera surveillance of BEHAVE		Total length	Synchronization time for each pair	Ground truth data description (length (fight event))	
				[6]	Ours
Seq.#1	margaret#1	52:01	0:00	15:54(8)	35:56(10)
	taku#1	56:59	4:20	0(0)	48:34(10)
Seq.#2	margaret#2	19:26	0:00	0(0)	12:18(5)
	taku#2	21:22	0:40	0(0)	17:03(5)

Table 2
SVM-based multi-temporal framework performances based on various codebook sizes.

Codebook size	*Recall*	*Specificity*	*Precision*	*Accuracy*	*F1-score*
2	0.302	0.702	0.503	0.502	0.378
5	0.512	0.728	0.653	0.620	0.574
30	0.603*	0.690	0.660	0.646*	0.630
50	0.749**	0.477	0.589	0.613	0.660**
100	0.355	0.871	0.734	0.613	0.479
150	0.572	0.827*	0.768*	0.700**	0.656*
200	0.272	0.982**	0.939**	0.627	0.422

** and * denote the best and the second best performance.

SVM classifies events of human groups from BoW features, which are based on STIP features. The decision function of SVM is described as follows:

$$f(x) = \sum_i y_i a_i k(x, x_i) + b \tag{14}$$

where x_i is the ith training sample, x is the test sample, y_i is the label of an event class, a_i is the weight of the training sample x_i, b is a learned threshold parameter, and $k(x, x_i)$ is a kernel function. For the implementation, we used LibSVM [10] and set the kernel function $k(x, x_i)$ to a radial basis function (RBF).

To analyze the activity score from the SVM classification result $event_g^{i-term}$, we used the simple formula shown in Fig. 9 because it was difficult to apply semantic meaning or distance calculation to event labels. On the contrary, SGT accommodates definitions of distances between nodes. Our target event is violence (i.e., FIGHT in BEHAVE, NUS-HGA and YouTube); therefore, when a classification result $event_g^{i-term}$ is FIGHT, the score for violence $score_g^{i-term}$ becomes 1; otherwise, it is 0.

4.3. Late fusion

To combine activity scores $score_g^{i-term}$ obtained from MtPL, we carried out a simple fusion method that averages multiple scores and combines them with the most recent fusion score $Score_{g,t-1}$:

$$Score_{g,t} = \begin{cases} Score_{g,t-1} \frac{\sum_i score_g^{i-term}}{N_{layers}}, & \text{if } Score_{g,t-1} \text{ exists} \\ \frac{\sum_i score_g^{i-term}}{N_{layers}}, & \text{otherwise} \end{cases} \tag{15}$$

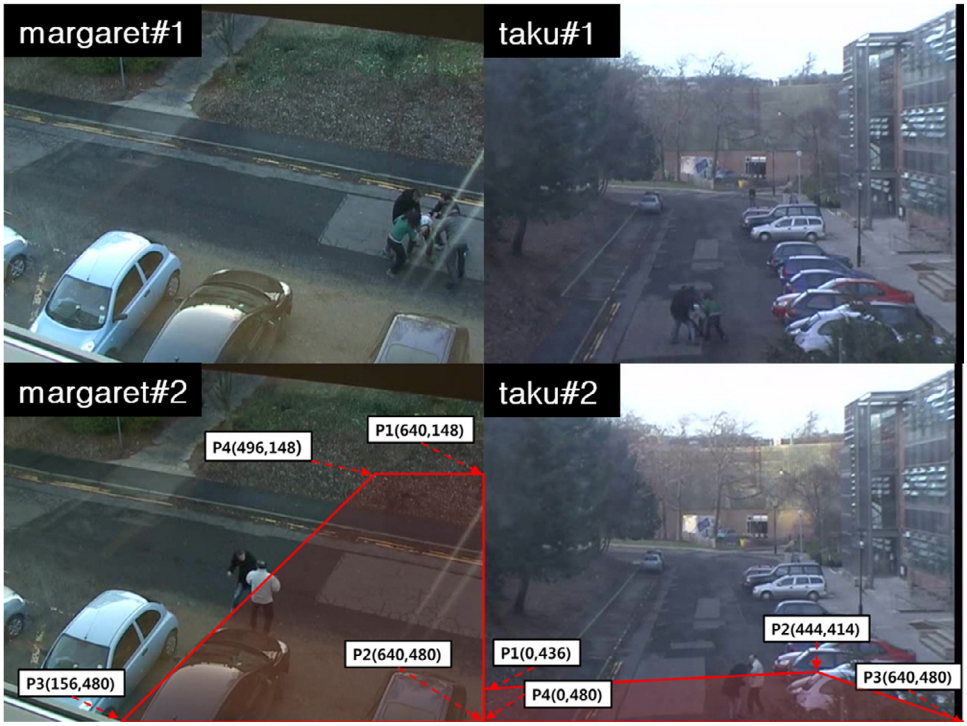

Fig. 10. Sample images of BEHAVE dataset (the name of each dataset is given at the top-left of each image). These images are synchronized for each row (margaret#1 to taku#1, margaret#2 to taku#2). The images in the second row represent geometrical information for regions of overlapping FOV.

Fig. 11. Warping result from margaret to taku by using $H_{margaret,\ taku}$. Left: an original sample image of margaret. Right: a warped image of the left.

where $score_{g,t}^{i-term}$ is the target activity score of group g, and N_{layers} is the number of multi-temporal layers.

4.4. Data refinement for multi-camera surveillance

For multi-camera surveillance with an overlapping field of view (FOV) among multiple cameras, the activity scores of such groups that can be observed from other views can be updated by using the activity scores of the view-based activity scores of these other groups.

We updated the activity scores by using look-up tables (LUTs) that represent the same group in multi-camera views. These LUTs are first generated by using prior geometrical information of each pair of cameras (i.e., the transformation matrix set H). The algorithm for LUT generation is stated in Algorithm 1. All pairs of multiple cameras are used to find groups that are common across camera views (lines 1–8 in Algorithm 1). When these groups are found, an element of the LUT

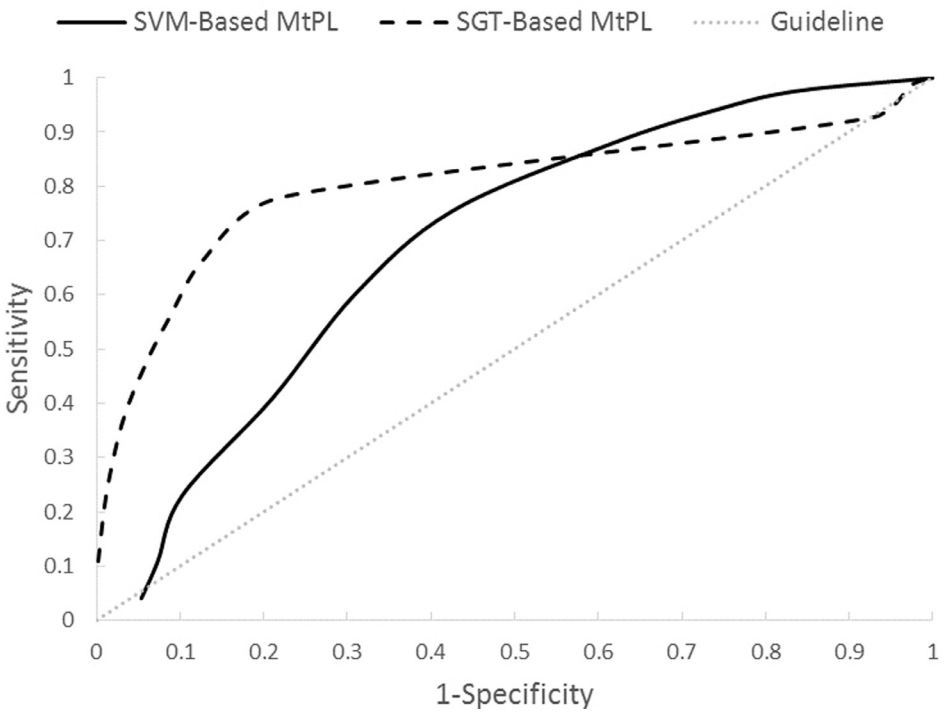

Fig. 12. ROC curves of SVM and SGT-based MtPL by using BEHAVE dataset. Experiments are conducted using various thresholds (0.0, 0.1, ..., 1.0) for activity scores. The best thresholds have resulted in 0.3 and 0.5 for the SGT- and SVM-based MtPL. The size of codebook for the SVM is 30.

becomes 1; otherwise, it is 0 (lines 5–8 in Algorithm 1). Following the comparison, the updated LUTs are returned (line 9 in Algorithm 1).

Finally, we use LUTs to update the activity score of the given group by averaging all scores of the group appearing in different views. This process of updating activity scores is shown in Algorithm 2. *SCORE* is a set of activity scores from multi-camera views (e.g., when there are three groups in the FOV of Camera #1, $SCORE_1 = \{Score_{1,t}, \ldots, Score_{3,t}\}$, $SCORE_1(3) = Score_{3,t}$).

5. Experiments and results

We conducted experiments to test the performance of the proposed framework for both multiple and monocular camera surveillance. For multi-camera surveillance, the BEHAVE dataset [6] was used as a benchmarking dataset because it offers multiple sequences captured by multiple cameras with an overlapping FOV for surveillance. The NUS–HGA dataset and some

Algorithm 1 Look-Up table generation representing the same group in multiple camera views.

$c \leftarrow$ number of multiple camera
$i, j \leftarrow$ index of camera
$k \leftarrow$ number of groups in ith view
$l \leftarrow$ number of groups in jth view
$R_i = \{r_{i,1}, \ldots, r_{i,k}\} \leftarrow$ group region set in ith view, where $R_i(a)$ is $r_{i,a}$
$warping(R_i, H_{i,j}) \leftarrow$ image warping from i- to j-th view for region R_i using $H_{i,j}$
INPUT: H: transformation matrix set $\{H_{1,2}, \ldots, H_{i,j}\}$, R: group region set $\{R_1, \ldots, R_c\}$
OUTPUT: LUT: Look-up tables $\{LUT_{1,2}, \ldots, LUT_{i,j}\}$
1: **for** all pairs of ith view and jth view **do**
2: **for** all kth group of ith view **do**
3: $tR_i(k) \leftarrow warping(R_i(k), H_{i,j})$
4: **for** all lth group of jth view **do**
5: **if** $tR_i(k) \cap R_j(l) \neq \emptyset$ **then**
6: $LUT_{i,j}(k, l) \leftarrow 1$
7: **else**
8: $LUT_{i,j}(k, l) \leftarrow 0$
9: **return** LUT

Algorithm 2 Activity score update for overlapping view.

$c, i, j, k, l \leftarrow$ same notations as the Algorithm 1
$SCORE_i = \{Score_{1,t}, \ldots, Score_{k,t}\} \leftarrow$ activity score set in ith view, where $SCORE_i(a)$ is $Score_{a,t}$
$n_i = \{n_{i,1}, \ldots, n_{i,k}\} \leftarrow$ a set of counting number for the updating, where $n_i(a)$ is $n_{i,a}$
INPUT: LUT: look-up tables $\{LUT_{1,2}, \ldots, LUT_{i,j}\}$, $SCORE$: activity score set $\{SCORE_1, \ldots, SCORE_c\}$
GOAL: updating $SCORE$ by using LUT
1: **for** all $SCORE$ **do**
2: $tempScore_c \leftarrow SCORE_c$
3: $n_c \leftarrow 1$
4: **for** all LUT **do**
5: **if** $LUT_{i,j}(k, l) == 1$ **then**
6: $tempScore_i(k) += SCORE_j(l)$
7: $n_i(k) += 1$
8: **for** all $SCORE$ **do**
9: $SCORE_c \leftarrow tempScore_c / n_c$

Table 3
Performance comparison of single-temporal and multi-temporal domains by using SGT- and SVM-based framework.

Method	Temporal domain		*Recall*	*Specificity*	*Precision*	*Accuracy*	*F1-score*
SGT	Single-temporal (Baseline)	0.2 s	0.713	0.848	0.824*	0.780	0.764
		1 s	0.620	0.871*	0.828**	0.746	0.709
		2 s	0.592	0.873**	0.823	0.732	0.689
		Average (*stdev.*)	0.642(0.052)	**0.864**(0.011)	**0.825(0.002)**	0.753(0.020)	0.721(0.032)
	Multi-temporal	0.2, 1 s	0.723*	0.843	0.822	0.783*	0.769*
		0.2, 2 s	0.717	0.831	0.809	0.774	0.761
		0.2, 1, 2 s	0.733**	0.833	0.814	0.783**	0.772**
		Average (*stdev.*)	**0.725(0.006)**	0.836**(0.005)**	0.815(0.004)	**0.780(0.004)**	**0.767(0.005)**
SVM	Single-temporal (Baseline)	0.2 s	0.598*	0.737	0.694	0.667	0.642
		1 s	0.468	0.840*	0.745	0.654	0.575
		2 s	0.584	0.731	0.685	0.658	0.631
		Average (*stdev.*)	*0.550(0.050)*	*0.770(0.043)*	*0.708(0.023)*	*0.660(0.025)*	*0.616(0.029)*
	Multi-temporal	0.2, 1 s	0.561	0.842**	0.780**	0.702*	0.653
		0.2, 2 s	0.638**	0.767	0.732	0.702**	0.682**
		0.2, 1, 2 s	0.572	0.827	0.768*	0.700	0.656*
		Average (*stdev.*)	**0.591(0.029)**	**0.812(0.028)**	**0.760(0.018)**	**0.701(0.011)**	**0.664(0.017)**

** and * denote the best and the second best performance for each MtPL in the temporal domain; results in bold indicate better performance between single- and multi-temporal in terms of average and standard deviation(stdev.) per each method.

online videos (YouTube), which captured real situations of group fights, were used as datasets of monocular camera surveillance. All the experiments on these datasets were carried out by using the same models and parameters except for the tests that were conducted to compare performances when using different parameters, and also experiments for efficiency of time parameters were proceeded to find influences of the parameters.

By using the BEHAVE dataset, two types of multi-temporal frameworks (SGT or SVM) were evaluated by comparison with ground truth activity data on a frame-by-frame basis. Both types of single-temporal frameworks of each multiple temporal term were evaluated in the same manner. A comparison of the results from frameworks that used either singular or multiple temporal domains made it possible to highlight the advantages of the proposed frameworks. Some experiments were additionally conducted to find appropriate parameters, such as the number of codebooks for the BoW representation and the threshold for decision-making. Finally, we compare the state-of-the-art methods to ours, analyze the experimental results, and discuss the proposed frameworks.

The entire datasets which we used have similar constraints such as the FOV with stationary camera and a ratio of human size is approximately 1:3 (width:height) which is almost 5% and 15% of image size. Our scopes of violent events are variety fight scenes from minimum two to maximum fifteen people include various movements and time durations between 2 and 48 s.

5.1. Behave dataset

5.1.1. Dataset description

Blunsden et al. built the BEHAVE dataset [6]. This dataset was captured at 25 frames per second (FPS) at a resolution of 640 × 480, and consists of four video clips: margaret#1, margaret#2, taku#1, and taku#2, as shown in Fig. 10. They contained the ground truth data regarding activity labels and locations of (some but not all) humans. However, these data were only for one of four clips (i.e., margaret#1) and accounted for approximately 16 of 52 minutes. It was difficult to use

the BEHAVE dataset for multi-camera surveillance by using this ground truth data, even though Blunsden et al. made an appropriate dataset for multi-camera surveillance.

Therefore, we manually formulated ground truth data for the activity labels and locations of all humans. Our data were produced not only for margaret#1, but also for the other datasets that were used: margaret#2, taku#1, and taku#2. Our ground truth data for activity labels consisted of Fight and non-Fight because we were only concerned with high-level activity in the dataset. Some information for multi-camera calibration and synchronization was also produced by our ground truth data. The geometrical information and synchronized sample images are shown as Fig. 10. The synchronization time, and the total duration of the BEHAVE dataset as well as the ground truth data, are listed in Table 1.

5.1.2. Performance evaluation

For data refinement in multi-camera surveillance by using BEHAVE, we used the geometrical information described in the second row of Fig. 10 to obtain a transformation matrix from margaret to taku $H_{margaret,\ taku}$, and a warping (from margaret to taku). The result can be seen in Fig. 11. To determine the thresholds (i.e., $threshold_{SGT}$ *and* $threshold_{SVM}$) for the activity score to make a binary decision regarding a violent event, we conducted experiments for both SGT and SVM-based MtPL involving a frame-by-frame comparison to find the receiver operating characteristic (ROC) curves from various activity scores. These experiments were performed on the entire BEHAVE dataset, and the results were averaged over all datasets (see Fig. 12). For frame-by-frame comparisons, we used the following evaluation methods:

$$predict_t = \begin{cases} TRUE, & max(SCORE_{view}) \geq threshold_{\mathrm{MtPL}} \\ FALSE, & \text{otherwise} \end{cases} \tag{16}$$

$$\begin{aligned} Sensitivity(= Recall) &= \frac{TP}{(TP+FN)}, \\ Specificity &= \frac{TN}{TN+FP}, \\ \mathrm{Pr}ecision &= \frac{TP}{TP+FP}, \\ Accuracy &= \frac{TP+TN}{TP+TN+FP+FN}, \\ F1-score &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \end{aligned} \tag{17}$$

where *view* represents the name of a given dataset or camera, MtPL represents one of two types of MtPL: SGT and SVM, *TP* stands for true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative. From the ROC curve, the thresholds for each framework were found (i.e., $threshold_{SGT} = 0.3$ and $threshold_{SVM} = 0.5$). The rest of the experiments were conducted using these thresholds and evaluation methods. Moreover, temporal scales for experiments were set to 0.2, 1.0, and 2.0 seconds.

To find proper a codebook size for SVM-based MtPL, we performed experiments by using various codebook sizes. These experiments were conducted based on 10-fold cross-validation using all BEHAVE datasets with the multi-temporal layers. The results of these experiments are shown in Table 2 for sensitivity, specificity, precision, accuracy and F1-score per codebook size. The best performance in terms of accuracy and F1-score was recorded with a size of 150 codebooks therefore we used 150 codebooks for the rests.

Experiments to verify the proposed multi-temporal framework were conducted by comparing single-temporal frameworks with multi-temporal frameworks. Ten-fold cross-validation is used to train and test SVMs. The results are listed in Table 3 and those examples are illustrated in Fig. 13. From these results, we have found that using the multiple temporal layers are better than the singles in terms of stability and performance.

Multi-temporal results, for both SGT- and SVM-based MtPL, show stability among the various temporal domains, whereas the single-temporal frameworks indicated instabilities: the standard deviations of multi-temporal results were smaller than those of the single-temporal results. Especially, recall that these numbers of the single-temporal results were the highest which is the worst. It shows that temporal scales can be affected to analyze high-level activity and are difficult to select. This difficulty becomes more severe when using a single time scale but can be handled by using the multi-intendant perception layers.

As an aspect of performance, the proposed multi-temporal framework produced highly accurate results in terms of sensitivity, accuracy, and the F1-score compared to the single-temporal frameworks. In this particular case of SVM-based MtPL, the multi-temporal results scored the best performances in all evolution methods.

To validate data refinement for multi-camera surveillance (Algorithms 1 and 2), experiments were performed using two types of multi-temporal frameworks that had double or triple layers with and without data refinement. These experiments are also based on 10-fold cross-validation for SVMs. Table 4 presents the results of these experiments for each BEHAVE

Fig. 13. Some experimental results of the multi-temporal framework for BEHAVE dataset: margaret#1(FR: first row and RB: red-dash box), margaret#2(FR and YB: yellow-dash box), taku#1(SR: second row and RB), and taku#2(SR and YB) (Thin blue rectangles: locations and regions of humans from ground truth data. Thick pseudo-colored rectangles: locations and regions of human groups; pseudo-colored bar on the left-top side of each image: activity score of the violent event involving each group shown in pseudo-color. Number to the bottom right of each image: number of frames of each dataset. It became red when the frame represented a violent event in ground truth data). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

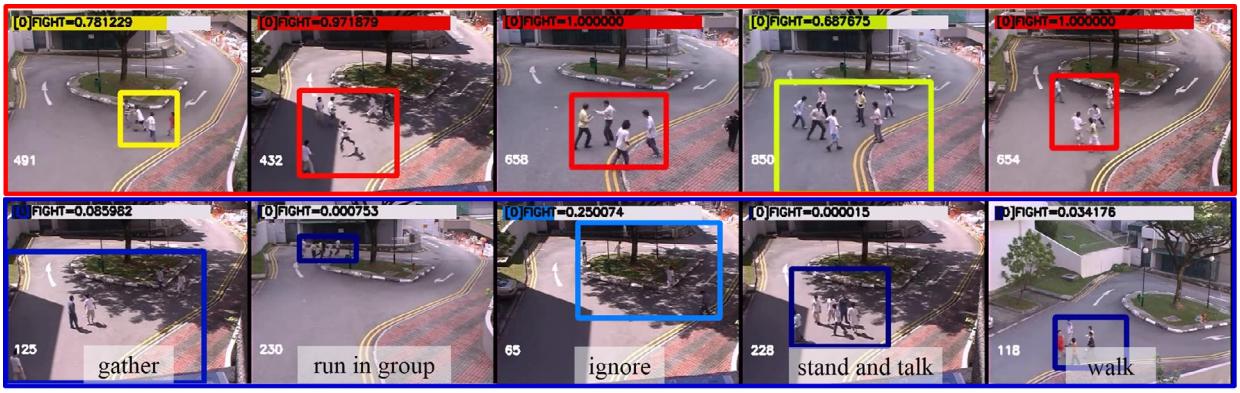

Fig. 14. Some experimental results of the multi-temporal framework for NUS–HGA dataset: images of the first row are a violent event and the second row are the other activities which are normal events (Thick pseudo-colored rectangles in each image: locations and regions of human groups; pseudo-colored bar on the left-top side of each image: activity score of the violent event involving each group shown in pseudo-color. Number to the bottom left of each image: number of frames of each dataset.).

Table 4
Performance evaluation results for data refinement for multi-camera surveillance (Algorithms 1 & 2) by using SGT-based and SVM-based multi-temporal frameworks (triple layers: 0.2, 1, 2 seconds) with and without data refinement.

MTF	Dataset	*Recall*		*Specificity*		*Precision*		*Accuracy*		*F1-score*	
		W/O DR	W DR	W/O DR	W DR	W/O DR	W DR	W/O DR	W DR	W/O DR	W DR
SGT	margaret#1	0.717	**0.739**	**0.837**	0.826	**0.815**	0.809	0.777	**0.782**	0.763	**0.773**
	margaret#2	0.440	**0.528**	**0.950**	0.938	0.810	**0.813**	0.813	**0.814**	0.814	**0.814**
	taku#1	**0.818**	0.815	0.808	**0.813**	**0.898**	0.895	0.695	**0.733**	0.591	**0.664**
	taku#2	**0.716**	**0.716**	0.862	**0.871**	0.839	**0.847**	0.789	**0.793**	0.772	**0.776**
	total	0.713	**0.733**	**0.835**	0.833	0.812	**0.814**	0.774	**0.783**	0.760	**0.772**
SVM	margaret#1	0.352	**0.440**	0.923	**0.926**	0.820	**0.856**	0.637	**0.683**	0.493	**0.581**
	margaret#2	0.298	**0.367**	0.946	**0.958**	0.675	**0.720**	0.674	**0.708**	0.674	**0.701**
	taku#1	0.673	**0.682**	0.675	**0.734**	0.846	**0.897**	0.622	**0.662**	0.441	**0.521**
	taku#2	0.729	**0.835**	0.782	**0.809**	0.770	**0.814**	0.756	**0.822**	0.749	**0.824**
	total	0.513	**0.572**	0.795	**0.827**	0.714	**0.768**	0.654	**0.700**	0.597	**0.656**

Results in bold indicate better performance between W/O DR and W DR (W DR: with data refinement, W/O DR: without data refinement).

Table 5
Performance comparison between state-of-the-art methods and the proposed methods: SVM- or SGT-based multi-temporal framework (MTF) for BEHAVE dataset (margaret#1).

Method		Temporal domain		*Recall*	*Specificity*	*Precision*	*Accuracy*	*F1-score*
[16]	Rule-based	Single	1 s	0.443	0.816	0.707	0.630	0.545
[12]	SVM (c:20)	Single	2 s				0.655	
Ours (MTF)	SVM (c:150)	Double	0.2, 1 s	0.429	0.920**	0.842**	0.674	0.568
		Double	0.2, 2 s	0.602	0.805	0.756	0.704	0.670
		Triple	0.2, 1, 2 s	0.496	0.904*	0.838*	0.700	0.623
	SGT	Double	0.2, 1 s	0.734	0.841	0.822	0.788**	0.775**
		Double	0.2, 2 s	0.735*	0.825	0.808	0.780	0.770
		Triple	0.2, 1, 2 s	0.739**	0.826	0.809	0.782*	0.773*

** and * denote the best and the second best performance; c represents the number of codebook.

dataset. Almost all overall results using data refinement were more accurate than those without. The differences in the results between datasets (i.e., views) tended to be larger for SVM than SGT; this result may indicate that semantic methods, such as SGT, are more suitable for surveillance applications than learning methods.

For the last experiment on the BEHAVE, we compared state-of-the-art methods [12,16], which use a rule-based model or SVM, with our multi-temporal frameworks: SVM with 150 codebooks (the best result in terms of F1-score) and SGT by using only the margaret#1 dataset. Table 5 presents a comparison between our results and the others. We used 3-fold cross-validation to train our SVMs to ensure that our learning environment is comparable to that of the state-of-the-art method [12].

The results indicate that the proposed frameworks, both SVMs and SGTs, were more accurate than the other methods. Overall, among our results, the SGT-based methods were more accurate than SVM-based MtPLs and the double temporal perception layers (time scales: 0.2 and 1.0 s), which are based on SGT models, show the best performance in terms of accuracy and F1-score.

Among all the experiments conducted by using the BEHAVE dataset, the proposed multi-temporal frameworks based on SGT models were determined to be the most suitable method for high-level activity. This is because sematic approaches can model human knowledge such as associations or hierarchical structures of activity; while machine learnings, which are highly related to features discriminative, have less capability of reflecting such knowledge. This is especially plain in visual surveillance for high-level activity analysis [1,16,18,22]. Hence other experiments, by using the other benchmarking dataset and some video collections from the web portraying real situations of group fights, were conducted to test and verify its efficiency by applying the same parameters and models that were used in the BEHAVE dataset experiments. Those other experiments are described in the following sections.

Table 6
Performance comparison for NUS-HGA dataset between state-of-the-art methods and the proposed methods include baseline.

Method		Temporal domain		*Recall*	*Specificity*	*Precision*	*Accuracy*	*F1-score*	*Test*
[5]	NN	Single	1 s	0.890	0.680	0.736	0.785	0.805	the leave-one-session-out strategy
	SVM	Single	1 s	0.890	0.890	0.890	0.890	0.890	
[12]	SVM (c: 20)	Single	2 s	0.930+	0.950+	0.949+	0.940+	0.939+	
Ours	SGT (Baseline)	Single	0.2 s	0.795	0.876	0.865	0.835	0.828	test all (no training process required)
		Single	1 s	0.768	0.838	0.826	0.803	0.796	
		Single	2 s	0.781	0.861	0.849	0.821	0.814	
		Average (stdev.)		0.781 (0.011)	0.859 (0.016)	0.847 (0.016)	0.812 (0.013)	0.813 (0.013)	
	SGT (MTF)	Double	0.2, 1 s	0.814	0.894	0.884	0.854	0.848	
		Double	0.2, 2 s	0.826*	0.906**	0.898**	0.866*	0.860*	
		Triple	0.2, 1, 2 s	0.841**	0.904*	0.897*	0.872**	0.868**	
		Average (stdev.)		**0.827 (0.011)**	**0.901 (0.005)**	**0.893 (0.008)**	**0.864 (0.008)**	**0.859 (0.008)**	

+ denotes the best overall performance. ** and * denote the best and the second best performance among ours; and results in bold indicate better performance between baseline and multi-temporal framework in terms of average and standard deviation (stdev.). c represents the number of codebook.

Table 7
YouTube collection dataset description.

	Total length of video (fight scenes)	Number of fight events (participants)
YouTube#1	3:29 (1:36)	6 (12–15)
YouTube#2	2:28 (0:55)	3 (10–12)
YouTube#3	2:12 (0:48)	1 (10–12)

Table 8
Performance comparison between baseline (single-temporal) and multi-temporal framework based on SGT for YouTube collection dataset.

Method	Temporal domain		*Recall*	*Specificity*	*Precision*	*Accuracy*	*F1-score*
SGT (Baseline)	Single	0.2 s	0.787	0.585*	0.662**	0.686	0.717
	Single	1 s	0.694	0.637**	0.657	0.665	0.673
	Single	2 s	0.760	0.552	0.628	0.656	0.688
	Average (stdev.)		0.747(0.048)	**0.591**(0.043)	0.649(0.018)	0.669(0.015)	0.693(0.022)
SGT (MTF)	Double	0.2, 1 s	0.811	0.578	0.662*	0.695*	0.728
	Double	0.2, 2 s	0.831**	0.567	0.660	0.699**	0.734**
	Triple	0.2, 1, 2 s	0.817*	0.567	0.657	0.692	0.727*
	Average (stdev.)		**0.812(0.008)**	0.571**(0.005)**	**0.660(0.003)**	**0.695(0.003)**	**0.730(0.004)**

** and * denote the best and the second best performance; and results in bold indicate better performance between baseline and multi-temporal framework in terms of average and standard deviation (stdev.)

5.2. NUS–HGA dataset

5.2.1. Dataset description

The NUS–HGA dataset was captured with 25 FPS at a resolution of 720 × 576, and collected in five different sessions. In each session, six categories of group activities were staged by 4–8 actors, including group fights and each session spans several minutes. These sessions were resampled as short video clips of about 8–16 seconds; therefore, the whole dataset includes 476 labeled video samples in total [5].

5.2.2. Performance evaluation

To test the proposed method, we resized the dataset into a resolution of 320 × 240 at 25 FPS and used a tracking method based on our previous work [23] for human location data. Experiments were conducted by using SGT-based MtPLs with the same experimental conditions used for BEHAVE and we compared the results with those obtained in other work [5,12]. The results are listed in Table 6 and sample images of the results are shown in Fig. 14.

Overall, SVM with multi-cue fusion [12] shows the best performance to analyze group fight activity. Their early fusion method on the feature level (motion and appearance fusion) well characterized the human activity on the dataset; however, they evaluated performances based on the leave-one-session-out strategy among 30 sessions in total. The proposed methods, even though it did not require training steps unlike the others, indicated reasonable results (accuracy = 0.872, F1-score = 0.868) and even show comparable results with the learning methods of the dataset provider [5]. Between multiple layers and singles, the multi-temporal results had higher performance and increased stability than the single-temporal results. This is similar to the BEHAVE experiments.

5.3. YouTube collection dataset

5.3.1. Dataset description

We collected three types of video including real situations of group fight from YouTube. All of these videos were recorded elsewhere by surveillance camera at different resolutions and 12–15 people were involved in the violent events. To match our experimental conditions, all videos were converted to a resolution of 320 × 240 at 25 FPS and each frame unit was also manually labeled as fight or normal activities. Details of these videos are listed in Table 7.

5.3.2. Performance evaluation

Performances were evaluated based on SGT-based MtPLs similar to the other experiments, which were carried out by using benchmarking datasets and the tracking method [42] was used to track humans. Experimental results are listed in Table 8 and sample images of the results and the event labels for each video are shown in Fig. 15.

In this dataset, the most successful method was double layers using temporal data in 0.2 and 1.0 seconds and the performance was 0.699 in terms of accuracy and 0.734 for the F1-score. Furthermore, we have confirmed that, like other experimental results, the results of this experiment also show the same trend: more accurate performance and stability on the multi-temporal results. However, in case of an evaluation of specificity, single-SGTs show higher results than the multiple-SGTs, which means there were many more false alarms in the multiple layers than the single layer.

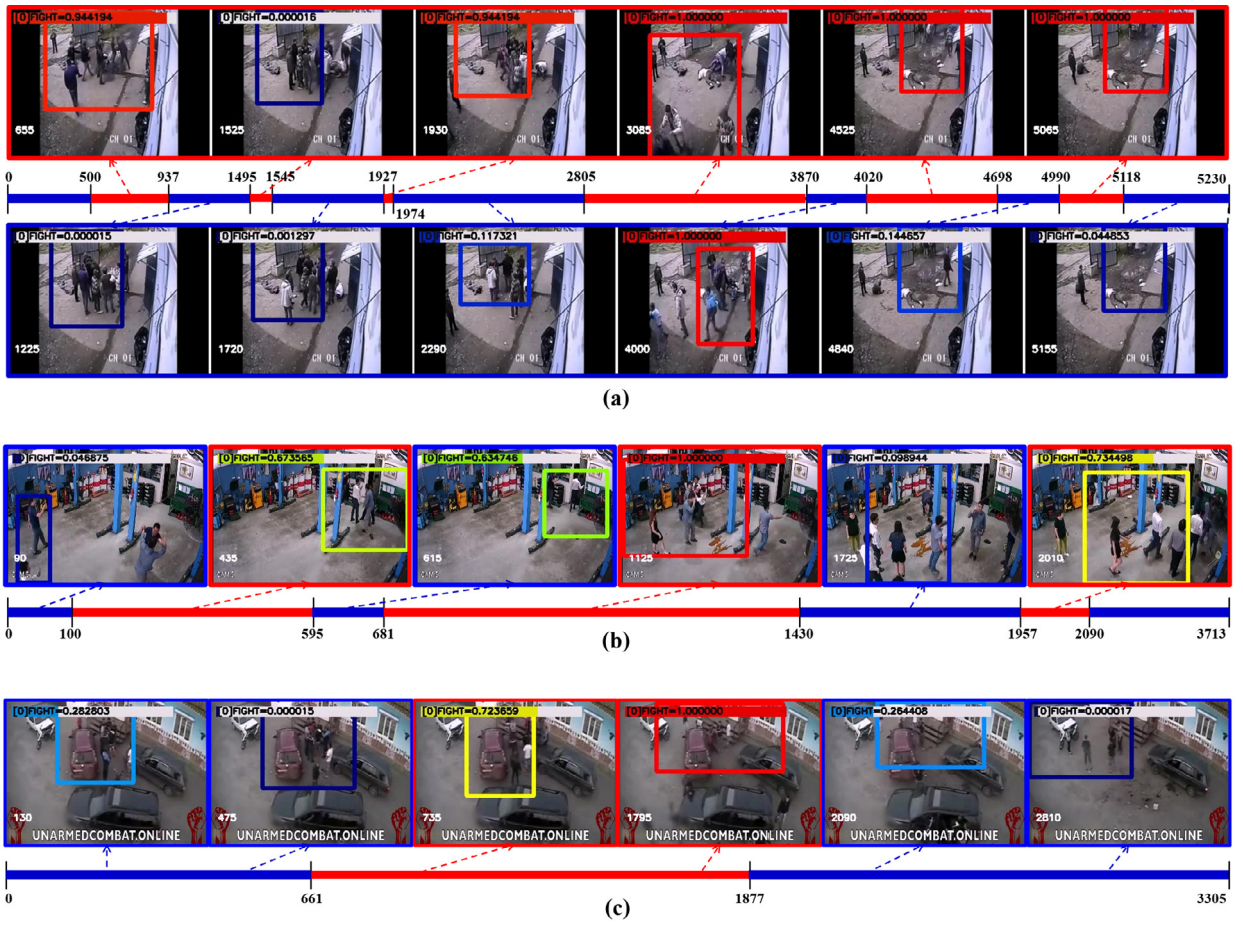

Fig. 15. Some experimental results of the multi-temporal framework for YouTube collection dataset: (a) YouTube#1, (b) YouTube#2, and (c) YouTube#3. A small bar in each example displays labeled events on a frame-by-frame basis. Red is a violent event section and blue is a normal event. (Examples inside on the red rectangle are the violent event and on blues are normal events. Thick pseudo-colored rectangles in each image: locations and regions of human groups; pseudo-colored bar on the left-top side of each image: activity score of the violent event involving each group shown in pseudo-color. Number to the bottom left of each image: number of frames of each dataset). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

The Fig. 15 shows that all violent events were detected by event unit except for one event, which was in YouTube#1 in the frames between 1495 and 1545, because it was an event among the people in which one person punched another (difficult to detect from a low degree of fighting). Compared to other results obtained with the same method in previous experiments, the performance in this experiment is the least accurate; therefore, this dataset appears to be the most difficult one. This is because it contains real situations of a group fight and each video was recorded in a different environment, unlike the others.

5.3.3. Experiments for efficiency of time parameters

Since the dataset collected from YouTube is the most realistic but difficult dataset to analyze, we conducted experiments on this dataset to figure out when time parameters are most efficient under different conditions: varying number of temporal layers and varying time scale for each layer. The tests were conducted with N layers and each layer has a different time scale in seconds t_i, where i represents varying temporal analyses: $N = 1, 2, \ldots, 20$; $t_i = i * \mu$; and $\mu = \{0.2, 0.4, 0.6, 1, 2\}$ (ex. when $N = 3$ and $\mu = 0.4$; $t_1 = 0.4s$, $t_2 = 0.8s$, and $t_3 = 1.2s$). The results from the experiment, in terms of F1-score, are illustrated in Fig. 16.

From the experimental results, we confirmed the following phenomenon: (1) in most cases, the experiment with three layers engendered optimal results; when the number of layers fewer and more than three were used the resulting graphs show positive and negative slope respectively (2) the larger the parameter of the time scale became, the less stable the result became; hence, at least for violent event detection in visual surveillance, the multi-temporal framework by using triple layers with short-, mid-, long-term time scales is recommendable.

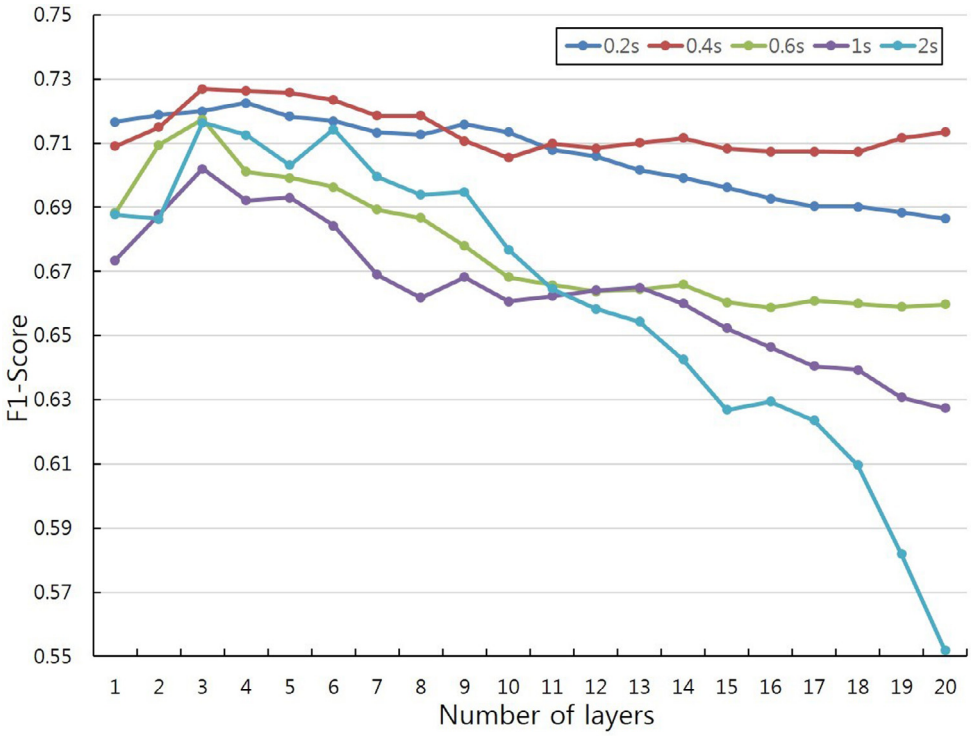

Fig. 16. Performance comparison of the multi-temporal framework per number of layers by using different incremental time scales for each experiment and the YouTube collection dataset is used for the evaluations.

6. Conclusion

In this paper, we proposed a novel framework for high-level activity analysis. The framework focused on overcoming the temporal diversity of high-level activity by using multi-independent perception layers with a different time scale for each layer. To verify our framework, both multiple and monocular camera surveillance scenarios were employed to detect violent events, and experiments based on the BEHAVE dataset were conducted by comparing single-temporal frameworks with multi-temporal frameworks based on a learning- or semantic modeling method. The experimental results showed that the proposed multi-temporal framework produced highly accurate results and the semantic-modeling method (SGT) was practical for visual surveillance. In addition, the usability of the SGT-based multi-temporal framework was verified by performing other experiments based on NUS–HGA and datasets collected from YouTube. The same parameters and models were used as in the BEHAVE test. The experimental results showed that the proposed method was not only comparable to state-of-the-art methods but also very practical for detecting violence in a surveillance environment. Lastly, we confirmed that for most cases the use of triple layers resulted in the optimal outcome with varying time parameters.

Acknowledgments

This work was supported by the Technology Innovation Program (No. 10060086, A robot intelligence software framework as an open and self-growing integration foundation of intelligence and knowledge for personal service robots) funded by the Ministry of Trade, Industry & Energy (MOTIE, Republic of Korea), and also supported by the National Research Council of Science & Technology (NST) grant funded by the Korea government(MSIT) (No. CRC-15-04-KIST).

References

[1] J.K. Aggarwal, M.S. Ryoo, Human activity analysis: a review, ACM Comput. Surv. 43 (2011) 1–43.
[2] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. Subrahmanian, P. Turaga, O. Udrea, A constrained probabilistic petri net framework for human activity detection in video, IEEE Trans. Multimedia 10 (2008) 982–996.
[3] P.K. Atrey, M.A. Hossain, A. El Saddik, M.S. Kankanhalli, Multimodal fusion for multimedia analysis: a survey, Multimedia Syst. 16 (2010) 345–379.
[4] P.K. Atrey, M.S. Kankanhalli, R. Jain, Information assimilation framework for event detection in multimedia surveillance systems, Multimedia Syst. 12 (2006) 239–253.
[5] B. Ni, S. Yan, A. Kassim, Recognizing human group activities with localized causalities, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1470–1477.
[6] S. Blunsden, R. Fisher, The BEHAVE video dataset: ground truthed video for multi-person behavior classification, Ann. BMVA 4 (2010) 1–12.
[7] F. Brémond, M. Thonnat, M. Zúniga, Video-understanding framework for automatic behavior recognition, Behav. Res. Methods 38 (2006) 416–426.

[8] G.J. Burghouts, K. Schutte, Spatio-temporal layout of human actions for improved bag-of-words action detection, Pattern Recognit. Lett. 34 (2013) 1861–1869.
[9] G.J. Burghouts, P. van Slingerland, R.J.M. ten Hove, R.J.M. den Hollander, K. Schutte, Complex threat detection: learning vs. rules, using a hierarchy of features, in: 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2014, pp. 375–380.
[10] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (2011) 27.
[11] Q. Chen, Y. Cai, L. Brown, A. Datta, Q. Fan, R. Feris, S. Yan, A. Hauptmann, S. Pankanti, Spatio-temporal fisher vector coding for surveillance event detection, in: Proceedings of the 21st ACM International Conference on Multimedia, ACM, 2013, pp. 589–592.
[12] Z.W. Cheng, L. Qin, Q.M. Huang, S.C. Yan, Q. Tian, Recognizing human group action by layered model with multiple cues, Neurocomputing 136 (2014) 124–135.
[13] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.
[14] Y. Dong, S. Gao, K. Tao, J. Liu, H. Wang, Performance evaluation of early and late fusion methods for generic semantics indexing, Pattern Anal. Appl. 17 (2014) 37–50.
[15] T.V. Duong, H.H. Bui, D.Q. Phung, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model, in: CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 831, 2005, pp. 838–845.
[16] M. Elhamod, M.D. Levine, Automated real-time detection of potentially suspicious behavior in public transport areas, IEEE Trans. Intell. Transp. Syst. 14 (2013) 688–699.
[17] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: Image Analysis, Springer, 2003, pp. 363–370.
[18] C. Fernández, P. Baiget, F.X. Roca, J. Gonzàlez, Determining the best suited semantic events for cognitive surveillance, Expert Syst. Appl. 38 (2011) 4068–4079.
[19] N. Ghanem, D. DeMenthon, D. Doermann, L. Davis, Representation and recognition of events in surveillance video using petri nets, CVPRW'04. Conference on Computer Vision and Pattern Recognition Workshop, IEEE, 2004, pp. 112–112.
[20] M. Golparvar-Fard, A. Heydarian, J.C. Niebles, Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers, Adv. Eng. Inf. 27 (2013) 652–663.
[21] M. Haag, H.-H. Nagel, Incremental recognition of traffic situations from video image sequences, Image Vision Comput. 18 (2000) 137–153.
[22] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, M. Shah, High-level event recognition in unconstrained videos, Int. J. Multimedia Inf. Retr. 2 (2012) 73–101.
[23] C. Kim, S.K. Park, Modified particle filtering using foreground separation and confidence for object tracking, in: 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2015, pp. 1–6.
[24] Z. Lan, L. Bao, S.-I. Yu, W. Liu, A.G. Hauptmann, Multimedia classification and event detection using double fusion, Multimedia Tools Appl. 71 (2014) 333–347.
[25] I. Laptev, On space-time interest points, Int. J. Comput. Vision 64 (2005) 107–123.
[26] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: CVPR 2008. IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[27] C.H. Lim, E. Vats, C.S. Chan, Fuzzy human motion analysis: a review, Pattern Recognit. 48 (2015) 1773–1796.
[28] L. Liu, L. Shao, X. Li, K. Lu, Learning spatio-temporal representations for action recognition: a genetic programming approach, IEEE Trans. Cybern. 46 (2016) 158–170.
[29] N.T. Nguyen, D.Q. Phung, S. Venkatesh, H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model, in: CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 952, 2005, pp. 955–960.
[30] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Int. J. Comput. Vision 79 (2008) 299–318.
[31] N.M. Oliver, B. Rosario, A.P. Pentland, A Bayesian computer vision system for modeling human interactions, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 831–843.
[32] S. Park, J.K. Aggarwal, A hierarchical Bayesian network for event recognition of human actions and interactions, Multimedia Syst. 10 (2004) 164–179.
[33] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: (CVPR), 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2847–2854.
[34] L. Shao, L. Liu, M. Yu, Kernelized multiview projection for robust action recognition, Int. J. Comput. Vision 118 (2016) 115–129.
[35] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings. Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 1470–1477.
[36] T.H. Thi, L. Cheng, J. Zhang, L. Wang, S. Satoh, Structured learning of local features for human action classification and localization, Image Vision Comput. 30 (2012) 1–14.
[37] N. Vaswani, A.R. Chowdhury, R. Chellappa, Activity recognition using the dynamics of the configuration of interacting objects, in: Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 632, 2003, pp. II-633-640.
[38] S. Vishwakarma, A. Agrawal, A survey on activity recognition and behavior understanding in video surveillance, Visual Comput. 29 (2013) 983–1009.
[39] A.D. Wilson, A.F. Bobick, Parametric hidden Markov models for gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 884–900.
[40] D. Wu, L. Pigou, P.-J. Kindermans, N.D.-H. Le, L. Shao, J. Dambre, J.-M. Odobez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2016) 1583–1597.
[41] J.B. Xie, T. Liu, W. Yan, P.Q. Li, Z.W. Zhuang, A fast and robust algorithm for fighting behavior detection based on motion Vectors, KSII Trans. Internet Inf. Syst. 5 (2011) 2191–2203.
[42] Y. Yacoob, M.J. Black, Parameterized modeling and recognition of activities, in: Sixth International Conference on Computer Vision, 1998, pp. 120–127.
[43] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden markov model, in: Proceedings CVPR'92., IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1992, pp. 379–385.
[44] G. Ye, D. Liu, I.-H. Jhuo, S.-F. Chang, Robust late fusion with rank minimization, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3021–3028.
[45] M. Yu, L. Liu, L. Shao, Structure-preserving binary representations for RGB-D action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2016) 1651–1664.
[46] L. Zelnik-Manor, M. Irani, Event-based analysis of video, in: CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 122, IEEE, 2001, II-123–II-130.