Crowd Activity Recognition using Optical Flow Orientation Distribution

Jinpyung Kim[1], Gyujin Jang[1], Gyujin Kim[1] and Moon-Hyun Kim[1]
[1] 1Artificial Intelligence Lab, College of Information and Communication Engineering
Sungkyunkwan University
440-746, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do, Republic of Korea
[e-mail: payon, gjjang, sperospera, mhkim@skku.edu]
*Corresponding author: Moon-Hyun Kim

Received January 21, 2015; revised May 18, 2015; accepted June 15, 2015; published August 31, 2015

Abstract

In the field of computer vision, visual surveillance systems have recently become an important research topic. Growth in this area is being driven by both the increase in the availability of inexpensive computing devices and image sensors as well as the general inefficiency of manual surveillance and monitoring. In particular, the ultimate goal for many visual surveillance systems is to provide automatic activity recognition for events at a given site. A higher level of understanding of these activities requires certain lower-level computer vision tasks to be performed. So in this paper, we propose an intelligent activity recognition model that uses a structure learning method and a classification method. The structure learning method is provided as a K2-learning algorithm that generates Bayesian networks of causal relationships between sensors for a given activity. The statistical characteristics of the sensor values and the topological characteristics of the generated graphs are learned for each activity, and then a neural network is designed to classify the current activity according to the features extracted from the multiple sensor values that have been collected. Finally, the proposed method is implemented and tested by using PETS2013 benchmark data.

Keywords: Structure learning, crowd behavior recognition, histogram of orientation optical-flow, multi-layer perceptron.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2014R1A2A1A11053902).

http://dx.doi.org/10.3837/tiis.2015.08.011 **ISSN : 1976-7277**

1. Introduction

A very important task for many visual recognition systems is to analyze the activities performed by an object within the frame. Activity recognition systems have inspired novel user interfaces and new applications for smart environments, including surveillance, emergency response, and military missions. In addition, a challenging problem for research in machine learning is to provide automatic recognition of object activity from data collected from imaging sensors [1].

The problem of learning patterns of human activity from sequences of images arises in many different areas where computer science is applied, including intelligent environments, surveillance, and assistive technology for the disabled. In particular, video surveillance has become more and more important due to the increase in need for security and related applications [2]. Video surveillance of dynamic scenes has a potentially wide range of applications, such as to assist security guards for communities, understand the behavior of crowds, provide traffic surveillance in cities and expressways, detect military targets, etc. [3]. One of the most important research topics in computer vision in particular is the video surveillance of dynamic scenes that contain crowds and objects [4][5]. A central topic of this area is the automatic analysis and recognition of crowd activity in video sequences [5][6]. The recognition of crowd activity can be described as a combination of two tasks: feature extraction and modeling classes of activities.

Cermeno et al. [7] proposed a method that extracts global features from an image, and they trained a two-class classifier using a feature vector for event recognition. Wang et al. proposed a method to detect abnormal events based on histograms of the orientation of the optical flow descriptor and a one-class SVM classifier [8]. Technically speaking, activity recognition can be divided into two tasks: (1) activity information extraction and (2) activity pattern modeling. The activity information represents attributes of movement (velocity, orientation, location) in the data while the activity patterns are representations of events that occur frequently.

In this study, we propose an intelligent activity recognition model from an image sequence of data. This paper provides the following contributions in image sequence-based activity recognition studies. First, the model proposes a representation method for crowd activity in an image sequence by using a histogram of optical flow. Second, a new machine learning method for activity recognition is also shown. The proposed learning method transforms a histogram of the optical flow to a graph, specifically a Bayesian network. In this network, each node corresponds to a histogram bin. The model extracts common structural features from the graphs generated for each activity, and these structural features are reflected in the structure of the neural network [8] as groups of input nodes. In addition, a numerical feature that represents the statistical properties of each histogram bin are used as input values for each input node.

2. Activity Recognition

In this paper, we present a three-stage method to recognize object activities. This method consists of a description stage, a structure learning stage, and a classification stage. During the

representation stage, we compute the optical flow and construct a histogram for the orientation of the optical-flow (HOOF)[9] in order to describe the movement of the crowd in an image sequence. The HOOF for the *i-th* frame is denoted as a vector $\Theta_i=[o_1,o_2,\ldots,o_9]$ where o_k denotes the value of the *k-th* histogram bin. These vectors are collected for the moving time window so that a set of vectors $\Theta_i^T=\{\Theta_{1-T+1},\ldots,\Theta_i\}$ can be formulated for T consecutive frames in a time window. In the structure learning stage, we generate a Bayesian network from Θ_i^T by using a k2-algorithm. The node in the Bayesian network is a component of the HOOF vector. The Bayesian network generalizes causal relationships between the nodes. We name this network as the context network. For each activity class A_{class}, a set of context networks, CN_{class} is constructed by collecting the Bayesian networks generated from frames with non-overlapping time windows. For each class, the common paths are extracted for the context networks. Each path P_i is defined as a structural feature of the class, and the structural features represent the topological characteristics of the context-networks. A structural feature P_i is implemented as a group of input nodes of the neural network, and a two-layer neural network is designed and is trained with a training image sequence for each class. During the training process, the structural features are extracted from the current context network while the numerical features for the nodes are included in the structural features that are to be applied to the input nodes of the neural network. The numerical features of the node represent the statistical characteristics of the nodes during a given time window.

For the input nodes of the non-existing structural features, 0's are applied in order to avoid training for the sample. After training the two-layer perceptron, the current activity is classified by deriving the context network from the current Θ_i^T. The paths of the context network are extracted and are compared to the stored structural features while the input nodes for the existing structural features are applied as numeric values for the nodes. The other input nodes are applied with 0 values, and the class is identified as an output node with a maximum output value. **Fig. 1** illustrates the proposed activity recognition method.

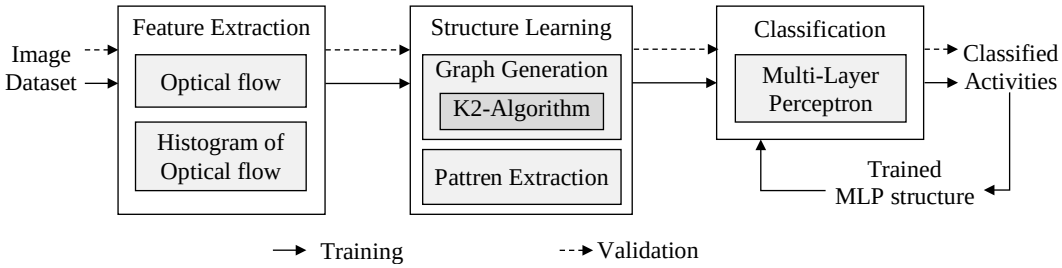


Fig. 1. The block diagram of proposed method for activity recognition.

2.1 Frame Description

2.1.1 Optical Flow

The optical flow is comprised of the apparent velocities of the pixels in an image sequence. Since the direction and the amplitude of movement are a representation of the activity, the optical flow is used as the scene description. B.Horn and B.Schunck [11] compute the optical flow by using a global smoothness constraint. The basic Horn-Schunck (HS) method is used to

compute the optical flow in this paper. The HS method minimizes the error function that combines two constraints. The first constraint is a brightness constancy constraint that assumes a constancy in the grey level of a point across the frames. The second constraint is a smoothness constraint that assumes a continuity in the velocities of the adjacent pixels[12]. Equation (1) shows the error function that is used.

$$E=\iint [(I_x u+I_y v+I_t)^2+\lambda(\left\| \bigtriangledown u^2 \right\| +\|\nabla v\|^2)]dxdy \quad (1)$$

I_x, I_y and It are the derivatives for the image intensity values along the x, y spatial axes and the time axis, u, v are the horizontal and vertical components of the optical-flow, and λ is a regularization constant. The optical-flow can be iteratively computed by using (2) and (3), where k denotes the iteration step, and $\bar{u}$ and $\bar{v}$ are weighted averages of u and v in the neighborhood of the pixel (x, y).

$$u(k+1)=\bar{u}(k)-\frac{I_x(I_x\bar{u}(k)+I_y\bar{v}(k)+I_t)}{\lambda^2+I_x^2+I_y^2} \quad (2)$$

$$v(k+1)=\bar{v}(k)-\frac{I_y(I_x\bar{u}(k)+I_y\bar{v}(k)+I_t)}{\lambda^2+I_x^2+I_y^2} \quad (3)$$

2.1.2 Histogram of Orientation Optical Flow

Fig. 2 shows a partition of an image with non-overlapping blocks, each block has the same size of $n \times n$ pixels, and each image frame is divided into m blocks, where m is the ($height_{image}/n$) × ($width_{image}/n$). For each block, the average of the optical flows for the block is computed to describe the local moving direction of the crowds in the block. For block k, the average of the optical-flow ($\overline{u_k}, \overline{v_k}$) is represented using the polar coordinates (r_k, θ_k). Where, f_{Bn} denotes a histogram of block B_n, and F_k denotes a set of histograms of frame k. The orientation bins are evenly spaced into 9 parts from 0° to 360°, as shown in **Fig. 2**, and the block k votes into one of the n orientation bins that include θ_k. The HOOF for frame i is denoted as a vector Θ_i= [o_1, o_2,…, o_9] that describes the distribution of the direction in which the crowd is moving in the entire frame. For T consecutive frames until the current i-th frame, $\Theta_i^T=\{\Theta_{i-T+1}, \ldots, \Theta_i\}$, each HOOF for all frames is collected in order to construct a HOOF sequence $\Theta_i=\{\Theta_{i-T+1},\ldots,x_i\}$. The HOOF sequence represents a change in the direction in which the crowd is moving during a given time window. The HOOF sequence is applied to the K2 algorithm in order to generate a context network. F_k denotes frame set of k blocks.

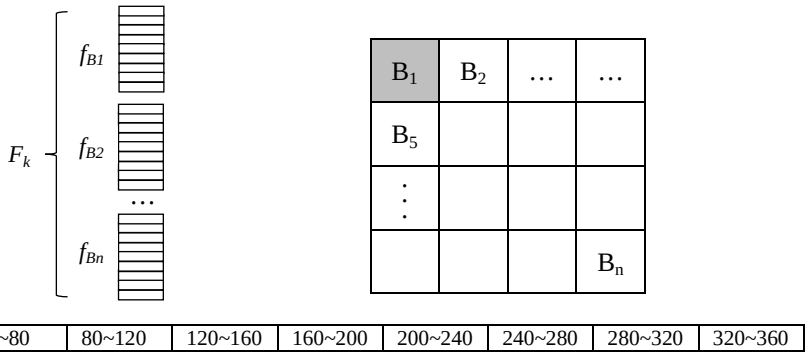


Fig. 2. 8×8 cells of Histogram of optical flow descriptor.

2.2 Structure Learning Stage

2.2.1 K2-Algorithm

A Bayesian Network (BN) is a graphical model that efficiently encodes the joint probability distribution for a set of variables [13]. The BN provides powerful knowledge representation and is a reasoning tool for conditions with uncertainty. A BN is a directed acyclic graph (DAG) that has a conditional probability distribution for each node, and the DAG structure of such networks contains nodes that represent the domain variables while the arcs between the nodes represent probabilistic dependencies [13]. We used a K2-algorithm to extract the structural relationship between histogram bin values. The K2-algorithm proposed by Cooper and Herskovits [14] is the most well-known Bayesian structure learning algorithm. The algorithm generates a Bayesian graph G with a joint probability and a Bayesian metric score. It is called the K2-metric and is the most well-known Bayesian network evaluation function. The K2-metric is expressed in (4).

$$\mathrm{P}\left(G,\theta_i^T\right)=\mathrm{P}(G)\prod_{\mathrm{i}=1}^{\mathrm{n}}\prod_{\mathrm{j}=1}^{\mathrm{q_i}}\frac{(r_i\text{-}1)!}{(N_{ij}+r_i\text{-}1)!}\prod_{\mathrm{k}=1}^{\mathrm{r_i}}N_{ijk}! \tag{4}$$

Maximizing $P(G, \Theta_i^T)$ searches for the most probable Bayesian network structure G given a database Θ_i^T. $P(G)$ is the structure prior probability that is constant for each G. In (4), r_i represents the number of possible values of the node o_i. And q_i is the list of all possible instantiations of the combination. We let π_i as set of parents of node o_i.

$$N_{ij}=\sum_{k=1}^{r_i}N_{ijk}! \tag{5}$$

N_{ijk} is the number of cases in Θ_i^T in which the attribute x_i is instantiated with its *k-th* value, and the parents of o_i in π_i are instantiated with the *j-th* instantiation in q_i. N_{ij} is the number of instances in the database in which the parents of o_i in π_i are instantiated with the *j-th* instantiation in q_i [14].

The K2-algorithm starts by assuming that a node has no parents, after which it incrementally adds the parent whose addition increases the probability of the resulting structure the most for every step. The K2-algorithm stops adding parents to the nodes when the addition of a single parent cannot increase the probability of the network for the given data [14][15]. The structure learning stage uses the K2-algorithm to obtain the graphs from the training data for each of the four classes. These learned graphs are named as context-graph G and are used to extract the distinctive path patterns that are used as input features for recognition of each class [16].

2.2.2 Pattern Extraction

The context-graph and extracted path patterns are generated as shown in **Fig. 3.** The node o_i in the context-graph is the value of the *i-th* histogram bin. In the graph, nodes o_4 and o_5 depend on the parent node o_2 which is implemented as an adjacency matrix in which an element represents the existence of the connection between the nodes.

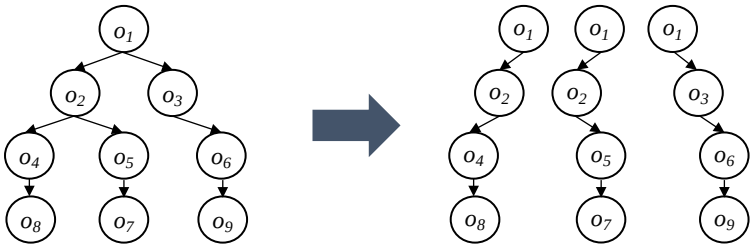


Fig. 3. Generated context-networks

The element of the adjacency matrix $A[i, j]=1$ if there is an edge between the *i-th* node and the *j-th* node, $A[i, j]=0$ otherwise. In a context-network $CN= (V, E)$ is the directed graph where V is a set of nodes and E is a set of edges. An edge $e=<o_s, o_e>\in E$, where o_s and o_e are the tail and head of edge e represents a causal relationship. That is, n_s affects the occurrence of n_e. Thus the structural features, which are topological characteristics, of the context-network that is generated reflect these causal relationships among the nodes in a current situation. The paths for the graphs generated in structure learning stage indicate the patterns that describe the specific relations between the nodes that characterize each class. The path patterns generated for the context-network are extracted, and these are used as structural features during situation recognition. Each path pattern is a path from the root to the leaf node of the context-network [16], and each path is represented as a sequence of nodes that are ordered from the root node to the leaf node. The paths from the root to the leaf nodes in the context-network are extracted, i.e. o_1-o_2-o_4-o_8 and o_1-o_2-o_5-o_7, o_1-o_3-o_6-o_9 where o_i is the *i-th* orientation bin node.

The context-graphs for each class $c\in C=\{Walk, Run, Evacuation, Merge\}$ are learned by using the training data for each class. A set of context-graphs for class $c\in C$ is named CN_c and is shown in (6), where $CN_i^c \in CN^c$ denotes the *i-th* generated context-network for class c and N_c is the number of generated context-networks for class c.

$$BM^c<\{BM_0^b+BM_1^b+BM_2^b+\cdots+BM_M^b\} \tag{6}$$

For each context network CN_i^c, which is the *i-th* context-network for class c, the path patterns are extracted where $K_{n,i}$ is the number of the extracted path patterns from CN_i^c. **Table 1** shows the set of path patterns for class c. Each column denotes a set of path patterns from a context network. *Walk class*, *run class*, *merge class*, and *evacuation class*, have their own path patterns, from the respective learned context-networks.

Table 1. Path Pattern from Context-Network

CN^C				
CN_1^C	CN_2^C	CN_3^C	…	$CN_{n^c}^C$
P_{11}^C	P_{21}^C	P_{31}^C	…	$P_{n^c1}^C$
P_{12}^C	P_{22}^C	$P_{3K_{n,3}}^C$		$P_{n^c2}^C$
P_{13}^C	$P_{2K_{n,2}}^C$	-		-
$P_{1K_{n,1}}^C$	-	-		$P_{N^cK_{n,n^c}}^C$

2.2.3 Classification stage

During the classification stage, we designed a 2-layer neural network for pattern classification by using the path patterns extracted as shown in **Fig. 4.** For each class $c\in C$, the conditional

probability $P(p|c)$ for each path $p \in P_S$ is computed during training. For each class, the path patterns with the highest conditional probabilities are selected and are used as input features for classification [17]. These selected path patterns are defined as structure features. For each distinctive selected structure feature $p_i=o_i^1\text{-}o_i^2\text{-}o_i^m$, a set of input nodes for the neural network $IP_i=\{o_i^1, o_i^2,\ldots,o_i^m\}$ is assigned. For every node in p_i, i.e. o_i^j, j=1,2,…,m is assigned as an input node for IP_i. **Table 2** shows the structural features as selected path patterns for each class. The input nodes in each selected path feature are grouped separately, as shown in **Fig. 4.** Note that a structural feature can appear in different classes repeatedly, i.e., IP_2 in the *run class* and *merge class*. *Null* denotes that no such path pattern exists in the context-network of that class. By using this organization for the input layer, we can reflect not only the topology of the generated context-network but also the numerical properties of the variables. A two-layer perceptron with 70 input nodes and 4 output nodes is used. Four output nodes, *Walk*, *Run*, *Merge* and *Evacuation*, correspond to a class. The activation function used for a node n_i is

$$f(\text{net}_i)=tansig(net_i)=2/(1+e^{-2net})-1 \tag{7}$$

$$mds_h < \sum_i V_{hi} W_i \tag{8}$$

where W_{ij} is the weight of the connection between n_i and n_j, which is a *j-th* node in the precedent layer. The output value for node nj is denoted as X_j. This function allows for a smooth transition between the low output and the high output of the neuron [17]. The weights of the connections are learned by using a back-propagation algorithm, and the learning algorithm minimizes the sum of squared error E between the network outputs a and the target outputs t, and E is defined as:

$$E=mse=\frac{1}{N}\sum_{i=1}^{N}(e_i)^2=\frac{1}{N}\sum_{i=1}^{N}(t_i-a_i)^2 \tag{9}$$

where t_i={0,1}, and a_i=[0,1] are the target values and the network output for the *i-th* output node, respectively. N is the number of output nodes, i.e. 4. Backpropagation is the most widely used algorithm for supervised learning with multi-layered feed-forward networks. The basic idea for backpropagation learning [18] is that there is a repeated application of the chain rule in order to compute the influence of each weight in the network with respect to the error function E:

$$\frac{\partial E}{\partial w_{ij}}=\frac{\partial E}{\partial a_i}\frac{\partial a_i}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \tag{10}$$

where w_{ij} is the weight of the connection from node n_j to node n_i, a_i is the output, and net_i is the input for node n_i. Once the partial derivative for each of the weights is known, the error function can be minimized by performing a simple gradient descent:

$$w_{ij}(t+1)=w_{ij}(t)\text{-}e\frac{\partial E}{\partial w_{ij}}(t) \tag{11}$$

During the training phase, the input nodes are partitioned into two sets S_1 and S_2

$$S_1 \cup S_2=I,\ \ S_1 \cap S_2=\theta \tag{12}$$

where I is set of input nodes. Set S_1 includes the nodes that belong to the structural features for the current target class, and set S_2 is a set of the remaining input nodes. The input nodes in S_1 are given as preprocessed current numeric values while the input nodes in S_2 are given as 0's.

Thus, it selectively trains weights connected to the input nodes for the structural features of the current target class. During the recognition phase after training, the paths are extracted from the context-network that is generated. If the input nodes in the structural features of the neural network are provided with current numeric values, the other nodes are given 0's. The current class is classified as

$$C_C = arg\ max_C\ v(O_C) \tag{13}$$

where O_C is output node for a class c, and $v(.)$ denotes the value of an output node.

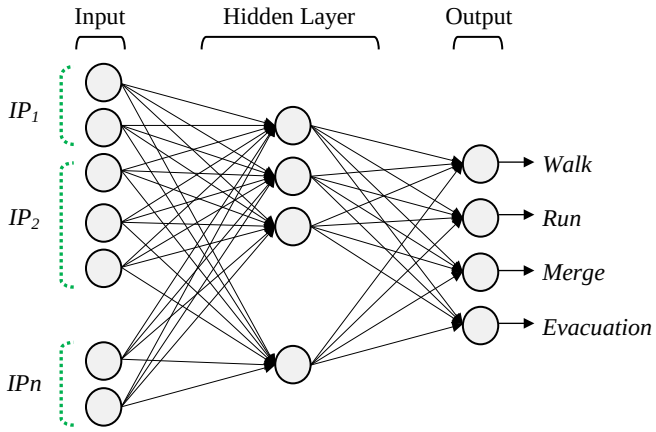


Fig. 4. Neural Network Architecture for Activity Recognition

Table 2. Implementation of structural features as clusters of input nodes

Path Feature / Class		*Walk*	*Run*	*Merge*	*Evacuation*
IP_1	1st Node	IP_1^W	*null*	*null*	IP_1^E
	2nd Node				
IP_2	1st Node	*null*	IP_2^R	IP_2^M	*null*
	2nd Node				
	3rd Node				
. . .	. . .	. . .	. . .	. . .	. . .
IP_n	1st Node	IP_n^W	*null*	IP_n^M	IP_n^E
	2nd Node				

3. Experiments and Result

We implemented the proposed method and measured its performance by using the PETS 2013 crowd activity dataset (S3) as experimental data.

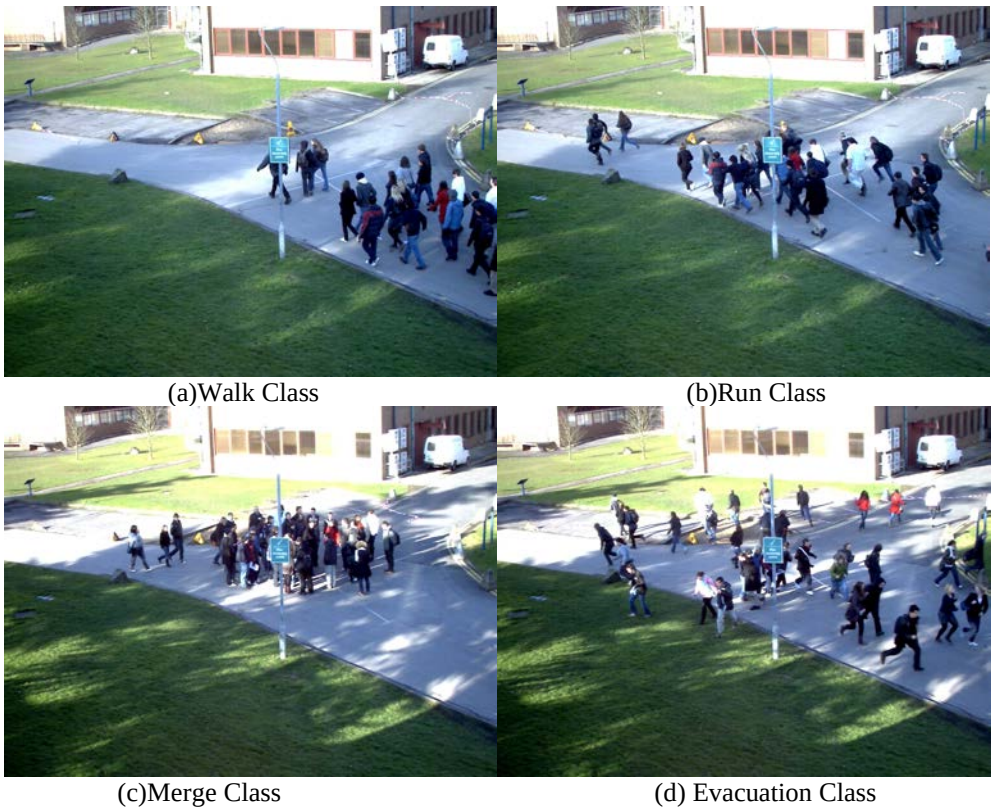
3.1 Crowd Activity Dataset

The crowd activity dataset contains different crowd activities, and the objective is to provide a probabilistic estimation at different times for each of the following events: *walking*, *running*, *evacuation*, and *merging*. Furthermore, we are interested in systems that can identify the start and end of the events as well as the transitions between them [19]. The image sequences depict

four crowd activities, and the image frames for four classes are shown in **Fig. 5**. 40 people act out the different crowd scenarios in the image sequences. In order to validate our approach, we tested the PETS DatasetS3, High Level, which contains four respective sequences with timestamps of 14:16 and 14:33. For each sequence, we use the videos recorded by camera1 (*view001*) and camera2 (*view002*). Moreover, we compared the performance of the proposed method against that of the MLP and Bayesian Network Classifier (BNC). **Table 5** shows the distribution of the class frames in the *view001* sequence and *view002* sequence. The number of image frames in the *walk class* and the *evacuation class* are almost equal, but the image frames in the *merge class* comprise the largest portion of the dataset. We can observe that the pedestrians have relatively similar sizes in the *view001* image sequence, and the pedestrians have different sizes in the *view002* image sequence due to characteristics of the perspective projection. Therefore, the optical flow vectors in *view001* have uniform magnitudes while the magnitudes of the optical flow vectors in *view002* are irregular.

Table 3. Distribution of classes in dataset

Class	Class Distribution	
	View001 & View002	
Walk	11%	14-16(0:37)
Run	21%	14-16(38:107)
Merge	56%	14-33(0 : 189)
Evacuation	12%	14-33(337:377)



(a)Walk Class (b)Run Class

(c)Merge Class (d) Evacuation Class

Fig. 5. Four crowd activity classes in view001

3.2 Feature Extraction

We proposed a method that analyzes the activity of the crowd through an optical flow and also accumulated the orientation of the representative optical flow for each image block. In **Fig. 6**, the feature extraction stage extracted its representative optical flow for the *run class* in *view001*.

Fig. 6. Four crowd activity classes in view001

Fig. 7 presents the accumulated orientation of the optical flow vectors for each class for the 9 orientation bins, which shows a different distribution depending on the class. For the *evacuation class*, the number of optical flow vectors in a bin of from 80° to 120° is the maximum. For the *run class*, a bin from 240° to 280° has the maximum number of flow vectors. The *walk class* has maximum number of optical flow vectors in the 160° to 200° bin.

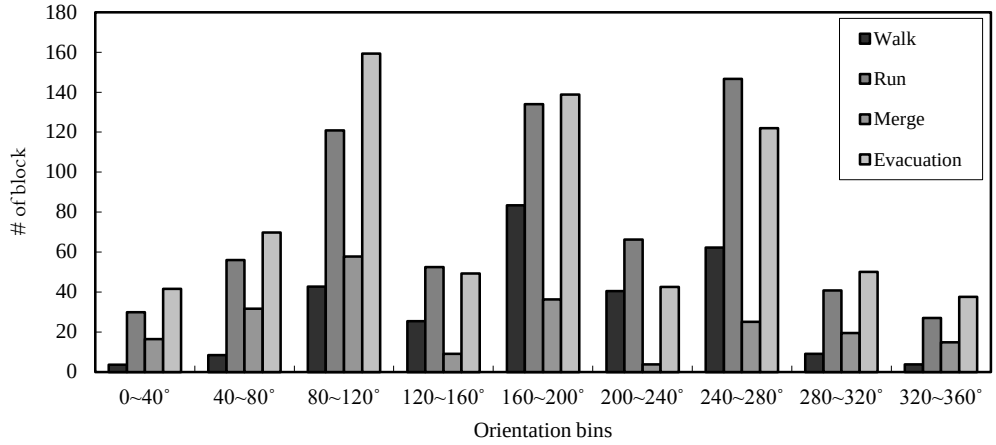


Fig. 7. Distibution of HOOF for four classes

3.3 Structure Learning

For the *i-th* image frame, a HOOF sequence $\theta_i^T = \{\theta_{i-T+1}, \ldots, \theta_i\}$, *T=10*, which is a collection of all HOOF for the previous frames, is constructed. The K2 algorithm is applied for this HOOF sequence in order to generate context networks for each class. From the context networks for each class, the path patterns with the highest conditional probabilities are extracted. The paths that most frequently appeared in the patterns for the four crowd activity

classes are shown in **Table 4**. It shows path patterns according to the number of occurrences during the training phase. The path o_1-o_2-o_3-o_7 for the *walk class* appears 12 times during training. This path also appears in the *merge class* with an occurrence number of 38. The path patterns represent the particular correlations among the variables that we have to significantly treat during the recognition phase. Thus these patterns become structural features for the input image sequence that is to be classified.

Table 4. Most frequently appeared path patterns for activity dataset

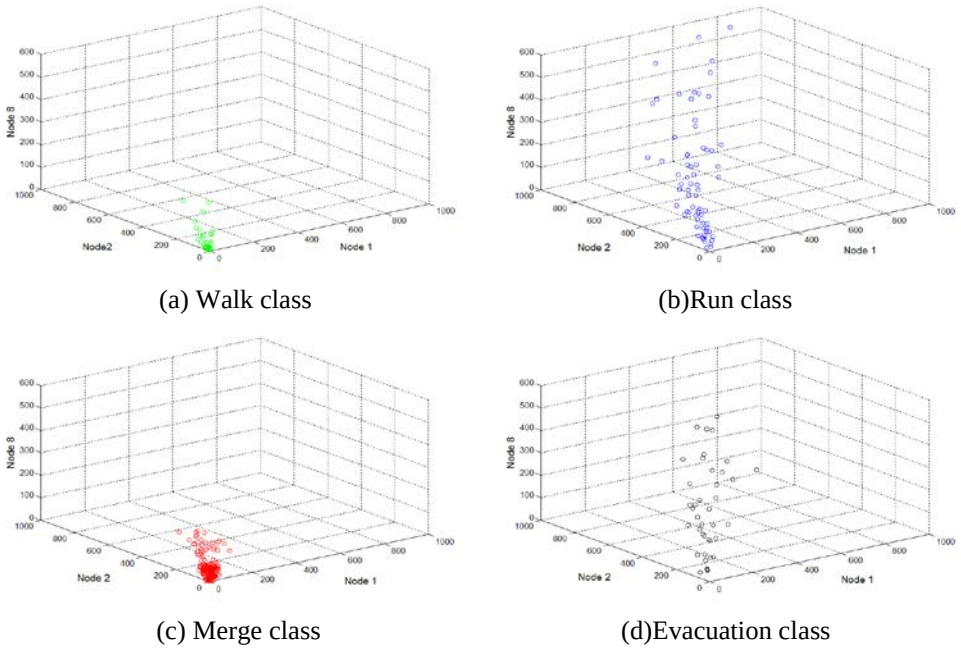
Class	Path Pattern						
Walk	**1-2-3-7**	**1-2-7**	**1-2-8**	**1-3-7**	**1-7**	**1-9**	**1-5-6**
	12	11	11	9	9	9	9
Run	**1-2-3-6**	**1-2-8**	**1-3-6**	**1-2-9**	**1-7**	**1-2-6**	**1-2-3-5-9**
	19	17	17	14	13	13	13
Merge	**1-2-3-4**	**1-2-8**	**1-2-3-7**	**1-8**	**1-2-5-9**	**1-2-3-8**	**1-2-3-9**
	44	40	38	36	36	32	32
Evacuation	**1-2-9**	**1-2-3-5**	**1-2-3-9**	**1-8**	**1-2-5**	**1-2-7**	**1-2-8**
	10	8	7	6	6	6	6

3.4 Classification

During the classification stage, structural features are used to classify the current activity by detecting the existence of these features in the current context network. However, common structural features are present in the different classes. For example, **Table 4** shows that the o_1-o_2-o_8 path pattern belongs to the structural feature sets of all classes. Since the existence of the structural features cannot be used to distinguish the classes, we used the second features, i.e., the numeric features. These features are the values of the HOOF variables. For the path pattern o_1-o_2-o_8, the conditional joint probability distribution of o_1, o_2, and o_8 for each class are shown in **Fig. 8**.

$$P(o_1,o_2,o_8|C), \quad C\in\{W,R,M,E\} \tag{14}$$

Fig. 8 shows the size of the optical flow about the 9 orientation bins of each class, namely the directional velocity. Each class represents a quite different distribution for the optical flow of the crowd behavior in the image sequence. In **Fig. 8(a)** and **Fig. 8(c)**, the *walk class* and the *merge class* mostly have narrow distribution for their values because *walk* occurs in one direction or all directions in each image sequence. In contrast, in **Fig. 8(b)** and **Fig. 8(d)**, the *run class* and *evacuation class* mostly have wider distributions for their values because *walk* occurs in one direction or all directions in each image sequence. In **Fig. 8(a)** and **Fig. 8(b)**, the *walk class* and *run class* have a skew in value distribution due to the movement in one direction in the image sequence. On the other hand, in **Fig. 8(c)** and **Fig. 8(d)**, the *merge class* and the *evacuation class* have a skew distribution of the value according to the movement in one direction for the image sequence. **Table 5** shows the mean of each variable, $E[o_i]$, as well as the correlation coefficient between the variables $\rho_{o_i o_j}$ in the o_1-o_2-o_8 path pattern. As show in Table 5, distribution of path pattern for each class is non-symetric distribution because, the context-network is directed network. This indicates a large mean value for o_1, $E[o_1]$ of the *run* and *evacuation class*. Strong positive correlations exists between the variables.



(a) Walk class (b)Run class

(c) Merge class (d)Evacuation class

Fig. 8. Distibution of HOOF for four classes

In particular, the o_1-o_2 in the evacuation class has the strongest relationship, 0.91, since pedestrians are spread in every direction.

Table 5. Value of each node in path patterns

Class		Average $E[o_i]$	Correlation coefficient $\rho_{o_i o_j}$		
			o_1	o_2	o_3
Walk	o_1	21.62	1.00	0	0
	o_2	180.01	0.73	1.00	0
	o_8	33.9	0.82	0.81	1.00
Run	o_1	209.03	1.00	0	0
	o_2	356.54	0.92	1.00	0
	o_8	190.13	0.89	0.88	1.00
Merge	o_1	48.44	1.00	0	0
	o_2	68.35	0.78	1.00	0
	o_8	37.20	0.73	0.82	1.00
Evacuation	o_1	223.21	1.00	0	0
	o_2	304.09	0.91	1.00	0
	o_8	198.41	0.85	0.89	1.00

Table 6 presents the variable values for path pattern o_1-o_2-o_4-o_5-o_6 in the context networks of the *walk class*. Their values are almost similar to those of every context network of the same class.

Table 6. Value of each node in path patterns

1-2-4-5-6 Pattern						
	G_3^N	G_6^N	G_8^N	G_9^N	G_{10}^N	G_{13}^N
o_1	0	0	0	0	0	0
o_2	0.22	0	0.22	0	0.22	0
o_4	-10	-10.2	-5.11	-8.89	-7.11	-10.9
o_5	-2.89	-3.56	-0.89	-2.89	-1.78	-3.78
o_6	0	0	0	0	0	0

3.5 Evaluation

We compare the performance of the proposed method against that of conventional MLP and a Bayesian classifier. Generally, conventional MLP exhibits better result as the number of hidden nodes becomes larger. However, the conventional MLP shows its best performance when a hidden layer consist of 20 hidden nodes. The used input nodes for the MLP and Bayesian classifiers are variables, $o_1,..,o_9$. The used input values for the variables are average during the time window, as shown in (15).

$$HOOF_i^{avg} = \frac{1}{10}\sum_{k=1}^{10} HOOF_i\,[k],\ i=1,2,\ldots,9 \tag{15}$$

The best performance for the proposed method can be achieved when it uses 25 hidden nodes and 99.4% of the whole image sequence as a training sequence. The performance is measured in terms of the precision in (16) and is compared with the performance of other methods in **Table 7**.

$$Accuracy = \frac{number\ of\ true\ positives + true\ negatives}{total\ number\ of\ image\ frames\ of\ class} \tag{16}$$

The proposed method exhibits the best performance for the *view001* sequence while also exhibiting the worst performance for the evacuation class in the *view002* sequence. Since this class takes the smallest part in the training sequence, this class is confused with other classes in *view002* due to poor training. The conventional Bayesian classifier showed the worst performance when compared to the proposed method and the conventional MLP. We also present the method proposed by Wang [9] to compare its performance by using the HOOF as features and the SVM as a classifier. They performed experiments by several classes(walking toward all the directions, walking toward the same direction, walking toward the same direction, crowd formation and evacuation, local dispersion) except run class. So, we compared their classes with our corresponding classes. We compute the performance as a weighted average of the precision of the four classes using (17).

$$Precision = \sum_{class} \frac{number\ of\ true\ positive\ image\ frames\ of\ class_i}{total\ number\ of\ positive\ image\ frames\ of\ class_i} \tag{17}$$

The average precision is plotted in **Fig. 9** and is compared to the average precision of a conventional MLP. It shows better performance both for the *view001* and *view002* sequences.

Table 7. Accuracy evaluation of crowd activity recognition

Class		Walk	Run	Merge	Evacuation
HOOF + SVM [9]		93.24%	-	97.88%	90.0%
Proposed Method	View001	100%	100%	100%	100%
	View002	92%	96%	100%	18%
MLP	View001	91.4%	95.6%	100%	85.4%
	View002	75%	87%	98.9%	100%
Bayesian Classifier	View001	74%	62%	75%	31%

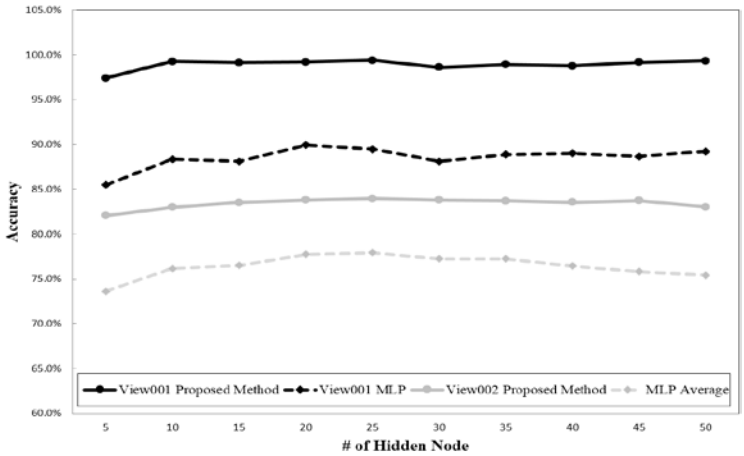


Fig. 9. Accuracy evaluation of crowd activity recognition

4. Conclusion

In this paper, we have proposed a crowd behavior recognition method that can be applied to a variety of complex activity environments. The proposed method consists of a feature extraction stage, a structure learning stage, and a classification stage. We propose a systematic structure learning approach that can automatically learn the appropriate Context-Network. The neural networks with structural features(pattern) designed and presented as part of this method can achieve an improved classification performance. Specifically, the structure learning stage is implemented in three steps: an input temporal vector formulation step, a subsequent context-network generation step with a K2-algorithm and a path pattern extraction step from the context-network. Our automatically learned situation recognition model outperformed the Multi-Layer Perceptron and Bayesian classifier. These results demonstrate that the proposed approach is feasible and provides sufficient recognition accuracy for multiple sensor signals. In the future, the study will expand path features in order to recognize complicated activity, such as loitering. Also, path features that are automatically generated from context-networks should be investigated. In addition, the proposed method does not consider real time execution yet. However, we are investigating efficient implementation of real time crowd activity recognition system.

References

[1] Hu, Weiming, et al., "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on 34.3, 334-352 2004. Article (CrossRef Link)

[2] Saxena, Shobhit, et al., "Crowd behavior recognition for video surveillance," *Advanced Concepts for Intelligent Vision Systems*, Springer Berlin Heidelberg, 2008. Article (CrossRef Link)

[3] Hu, Weiming, et al., "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on 34.3, 334-352, 2004. Article (CrossRef Link)

[4] Ke, Shian-Ru, et al., "A review on video-based human activity recognition," *Computers 2.2* 88-131 , 2013. Article (CrossRef Link)

[5] Candamo, Joshua, et al. "Understanding transit scenes: A survey on human behavior-recognition algorithms," *Intelligent Transportation Systems*, IEEE Transactions on 11.1, 206-224 ,2010. Article (CrossRef Link)

[6] Viola, Paul, Michael J. Jones, and Daniel Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. of Computer Vision*, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003. Article (CrossRef Link)

[7] Tae-Ki An, Moon-Hyun Kim, "Context-Aware Video Surveillance System," *Journal of Electrical Engineering & Technology*, Vol 7, No 1, 115-123, 2012. Article (CrossRef Link)

[8] Cermeno, Eduardo, Silvana Mallor, and J. A. Siguenza, "Learning crowd behavior for event recognition," *Performance Evaluation of Tracking and Surveillance (PETS)*, in *Proc. of 2013 IEEE International Workshop on. IEEE*, 2013. Article (CrossRef Link)

[9] Wang, Tian, and Hichem Snoussi, "Histograms of optical flow orientation for abnormal events detection," *Performance Evaluation of Tracking and Surveillance (PETS)*, in *Proc. of 2013 IEEE International Workshop on. IEEE*, 2013. Article (CrossRef Link)

[10] Yang, Jhun-Ying, Jeen-Shing Wang, and Yen-Ping Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern recognition letters* 29.16, 2213-2220, 2008. Article (CrossRef Link)

[11] Horn, Berthold K., and Brian G. Schunck, "Determining optical flow." in *Proc. of 1981 Technical Symposium East. International Society for Optics and Photonics*, 1981. Article (CrossRef Link)

[12] Fujisawa, Shizuka, et al., "Pedestrian Counting in Video Sequences based on Optical Flow Clustering," *International Journal of Image Processing 7.1* , 1-16 , 2013. Article (CrossRef Link)

[13] Heckerman, David, Dan Geiger, and David M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Machine learning 20.3, 197-243, 1995. Article (CrossRef Link)

[14] Cooper, Gregory F., and Edward Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine learning 9.4*, 309-347, 1992. Article (CrossRef Link)

[15] Lerner, Boaz, and Roy Malka*, "Investigation of the K2 algorithm in learning Bayesian network classifiers," Applied Artificial Intelligence 25.1 , 74-96, 2011 Article (CrossRef Link)

[16] Kim, Jin-Pyung, et al., "Multi-Sensor Signal based Situation Recognition with Bayesian Networks," *JOURNAL OF ELECTRICAL ENGINEERING & TECHNOLOGY 9.3*, 1051-1059, 2014. Article (CrossRef Link)

[17] Kim, Gyujin, Taeki An, and Moon-Hyun Kim, "Estimation of Crowd Density in Public Areas Based on Neural Network," *KSII Transactions on Internet and Information Systems (TIIS) 6.9* , 2170-2190, 2012. Article (CrossRef Link)

[18] Jain, Anil K., Robert P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: A review," *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 22.1, 4-37, 2000. Article (CrossRef Link)

[19] Choudhury, Tanzeem, et al., "The mobile sensing platform: An embedded activity recognition system," *Pervasive Computing*, IEEE 7.2, 32-41, 2008. Article (CrossRef Link)

[20] K. Y. Eom, J. Y. Jung, and Moon-Hyun Kim, "A heuristic search-based motion correspondence algorithm using fuzzy clustering," *International Journal of Control, Automation and Systems*, vol.

10, no. 3, pp. 594-602, 2012. Article (CrossRef Link)

Jin-Pyung Kim received his M.S. degree and Ph.D. degree in College of Information and Communication Engineering from Sungkyunkwan University, Suwon, Korea, in 2006 and 2014. He is currently a post-doctor in Sungkyunkwan University, Suwon, Korea. His research interests include structure learning, pattern recognition, and computer vision.

Gyu-Jin Jang received his M.S. degree in College of Information and Communication Engineering from Sungkyunkwan University, Suwon, Korea, in 2011. He is currently a Ph.D. candidate in Sungkyunkwan University, Suwon, Korea. His research interests include artificial intelligence, computer vision, and pattern recognition.

Gyu-Jin Kim received his M.S. degree in College of Information and Communication Engineering from Sungkyunkwan University, Suwon, Korea, in 2011. He is currently a Ph.D. candidate in Sungkyunkwan University, Suwon, Korea. His research interests include computer vision, pattern recognition, and artificial intelligence.

Moon-Hyun Kim received the B.S. degree in Electronic Engineering from Seoul National University in 1978, the M.S. degree in Electrical Engineering from KAIST, Korea, in 1980, and the Ph.D. degree in Computer Engineering from the University of Southern California in 1988. He joined the College of Information and Communication Engineering, Sungkyunkwan University, Seoul, Korea in 1988, where he is currently a Professor. In 1995, he was a Visiting Scientist at the IBM Almaden Research Center, San Jose, California. In 1997, he was a Visiting Professor at the Signal Processing Laboratory of Princeton University, Princeton, New Jersey. His research interests include artificial intelligence, pattern recognition, and machine learning.