J Intell Inf Syst (2014) 43:521–535
DOI 10.1007/s10844-013-0253-8

Employing flowgraphs for forward route reconstruction in video surveillance system

Andrzej Czyzewski · Karol Lisowski

Received: 15 September 2012 / Accepted: 21 May 2013
Published online: 21 June 2014

Abstract Pawlak's flowgraphs were utilized as a base idea and knowledge container for prediction and decision making algorithms applied to experimental video surveillance system. The system is used for tracking people inside buildings in order to obtain information about their appearance and movement. The fields of view of the cameras did not overlap. Therefore, when an object was moving through unsupervised areas, prediction was needed to identify the same object in the adjacent camera. Moreover, while being unobserved, the object was represented by measurements of probability of appearing in the subsequent camera's field of view. Those probability values were obtained on the basis of knowledge from the past events contained in the flowgraph. Backward Route Reconstruction methods generated paths of objects providing input for prediction of next steps in the path. This prediction was named as Forward Route Reconstruction. The output of the prediction algorithm is a tree of probabilities of future object movement. Methods for creating a flowgraph from the paths of objects and for obtaining probability values for movement prediction are presented in this paper together with some experimental results and discussion.

Keywords Route reconstruction · Video surveillance system · Movement prediction · Flowgraph

A. Czyzewski (✉) · K. Lisowski
Multimedia Systems Department, Gdansk University of Technology,
Narutowicza 11/12, 80-952 Gdansk, Poland
e-mail: ac@pg.gda.pl

K. Lisowski
e-mail: lisowski@sound.eti.pg.gda.pl

1 Introduction

In a video surveillance system it is not possible to observe all areas in each part of a building. Most often the observed places are: doors, entrances, exits, corridors and staircases. In these kinds of places, people are usually in motion. Accordingly, they appear, pass by, and leave the FOV (Field Of Vision) of a given camera. Spatio-temporal dependencies between cameras can be described with a directed (mostly bidirectional) weighted graph, the so-called topology graph. The decision which has to be made concerns determining if two visible objects seen by two different cameras represent the same physical object. Such decisions can be made based on following premises:

- Two appearances of the same physical object should have significant resemblance. The resemblance can be measured by comparing visual features of the objects present in the view of the camera.
- Certain paths of the object's movement are more probable than others. This information is encoded in the topology graph. Therefore, the probability of the object appearing in a certain area can be numerically expressed.
- Moreover, time dependencies are contained in the topology graph as well. The time period between the object observations is also taken into consideration.
- The last premise is the behaviour of the object. It is frequently observed that certain objects follow typical and repeatable paths. The knowledge of their typical behaviour should be gathered and stored in the system to support decision making.

The first three clues can be obtained instantly when two observations are available. However, in order to acquire knowledge of the behaviour of objects many more observations are necessary. Moreover, the behaviour can change over time, therefore a reaction to these fluctuations is desired. The gathered knowledge should be stored in a form which makes it easily accessible. The idea of Pawlak's flowgraph (hereinafter referred to as FG) complies with these conditions. Prediction of the next steps of the object's movement is expressed in the form of a probability tree. Every time an object is observed in the supervised area, its path is added to the set of observed paths. Subsequently, required changes are applied to the FG, incorporating the newly gathered knowledge. In order to keep the FG up to date, the paths stored in the system only cover a specified period of time (e.g., one month).

The remaining part of paper is organized as follows. In Section 2, the related work on using FGs as a tool for decision making is discussed. Methods for adapting FGs to changing trends in the analyzed environment are also presented. Additionally, other methods for prediction of object movement are mentioned. Section 3 describes the details of employing FGs for route reconstruction. We introduce, how the knowledge is gathered from past observations to construct the FG. Next, the process of refreshing the parameters of the FG is explained. Finally, the algorithm of movement prediction is mentioned. In Section 4, we use the proposed methods with the example data in order to clearly present to the reader the details of the engineered method. Section 5 describes computation complexity issues related to FGs. Section 6 contains the results and discussion of using the proposed algorithm with real data from a video surveillance system. Finally, Section 7 presents our conclusions and plans for future development of the method are discussed.

2 Related work

The foundations of the flowgraph idea were provided by Greco et al. (2002) and Pawlak (2005b). Many extensions to the original FG concept have been made. Simplicity of the FG and its applicability to problems of decision making and prediction are significant advantages. Determining strength, certainty and coverage values for paths in the FG has also been described in the literature (Pawlak 2005b). In related publications it has been shown how the FG can be used to create a decision tree (Pawlak 2006b). Moreover, the idea of FGs is strongly connected with Rough Sets and the Bayes' theorem (Pawlak 2003). Clear rules that control the FG allow for quick understanding of its principles, as well as for adaptation of this idea in numerous fields, including data mining (Pawlak 2005a), music metadata processing (Czyzewski and Kostek 2004), and conflict analysis and negotiations (Pawlak 2005c, 2006a). The FG is even applicable to making decision about deregulation of aviation market. Such solutions have been introduced, for instance, by the government of Taiwan (Ou Yang et al. 2008).

In order to predict the movement of an object between non-overlapping cameras, methods such as particle filters, joint probabilistic data association, Markov chains, Bayesian networks and other statistical algorithms can be used. Lev-Tov and Moses (2010) assign objects to two types of locations: unobserved (hidden) or visible to the camera. The state of the given object is composed of a vector of appearance descriptors and locations (hidden or visible) which the object is currently observed in. In order to decide in which state the object is, a particle filter is applied. Cheng et al. (2006) used joint probabilistic data association of visual features for matching of tracked objects. Javed et al. (2003) proposed a method for fusion of many types of clues for matching the same object in multiple non-overlapped cameras. A framework based on the Bayes Theorem was applied for this purpose. Kim et al. (2009) proposed an approach to the problem of tracking people utilizing a method based on Markov Chain Monte Carlo (MCMC). In this method probabilities of transitions between cameras are given a-priori. Moreover, Leung et al. (2008) proposed a probabilistic method based on a probability density function (p.d.f.) with entrance and exit events treated as random events. Additionally, determination of the probabilities of the object's possible routes is achieved with a dispersion function which can be interpreted as the conditional probability of a given exit event when a particular entrance event has occurred.

Movement of people through the supervised area can be considered as a flow of objects. Thus, such a phenomenon can be described and analyzed with the FG and methods employed for this task are described in the following subsections.

3 Proposed methods

In a video surveillance system in which the algorithms for making paths of the objects are implemented, methods for Forward Route Reconstruction can be applied as well. If the surveillance system has been operating for some time, certain statistics of the object movement can be gathered. Collected input data for Forward Route Reconstruction are the set of paths of observed objects from the past, as well as new, recently finished paths. The outcome encompasses the probability of object next path, provided it remains in the supervised area.

3.1 Building the flowgraph

In order to use Pawlak's flowgraph for prediction, some assumptions need to be made. A path is constructed from observations of an object in subsequent cameras. An observation is expressed by the identifier of a camera and the index determining which step of the path this observation denotes. Thus, these two parameters, as a pair, can be considered as an attribute for the FG. Furthermore, a class of attributes for the FG can be defined as a subset of all attributes which have the same index of step. It means that each consecutive vertical layer of vertices in the FG corresponds to the subsequent step in the path.

According to work of Greco et al. (2002), an information system $S = (U, A)$ can be described on the basis of a set of paths. Thus, U is a set of objects that enter and exit the area observed by the surveillance system and A is a set of the abovementioned attributes. Transforming the set of paths to FG is done using the following formulae. In (1), $\sigma(x_i, y_j)$ defines the rate of objects passing from camera x in step i of the path to camera y in step j of the path. The number of objects passing from x_i to y_j is denoted as $\varphi(x_i, y_j)$, and the total number of paths in the database is denoted as $\varphi(G)$.

$$\sigma\left(x_i, y_j\right) = \frac{\varphi\left(x_i, y_j\right)}{\varphi\left(G\right)} \tag{1}$$

Similarly, in (2), $\varphi(x_i)$ is the number of objects that passed the node x_i.

$$\sigma\left(x_i\right) = \frac{\varphi\left(x_i\right)}{\varphi\left(G\right)} \tag{2}$$

The next two formulas ((3) and (4)) are used for probability estimation. The names of these parameters, that is certainty (cer) and coverage (cov), are taken from terminology related to FGs.

$$cer\left(x_i, y_j\right) = \frac{\sigma\left(x_i, y_j\right)}{\sigma\left(x_i\right)} \tag{3}$$

The value of $cer(x_i, y_j)$ estimates the probability that the object which left camera x in step i of the path will appear in camera y in step j. Equation (4) is used to calculate the coverage factor $cov(x_i, y_j)$ for node y_j, so that it determines estimation of probability that an object which appears in camera y in step j of the path was seen before in camera x in step i of the path.

$$cov\left(x_i, y_j\right) = \frac{\sigma\left(x_i, y_j\right)}{\sigma\left(y_j\right)} \tag{4}$$

In (1), (3) and (4) j is equal to $i + 1$, thus it is related to the next step on the object path (the following layer in the FG).

The idea of FG needs some modification and adjustment in order to be used in Forward Route Reconstruction. The flow of objects through the video surveillance system can be described by a properly prepared FG. In such a system, the cameras which observe the entrances and exits (e.g., doors, gates) correspond to the first and last step of the path. The path of the object comprises several steps. Each step is associated to the object's appearance in one of the cameras. The first step is easily obtained when the object is first observed in the observed area. Subsequent steps are derived from following observations. The last step, which denotes the object's disappearance, is appended to the path after it has been observed that the object has left the supervised area. The FG encompasses all the observed paths. The nodes of the graph relate to steps on the object's path. The FG is arranged in layers, whose number corresponds to the order of the particular nodes in the observed paths. Therefore, the first layer of the FG represents the entrance (input) nodes, whereas the exit (output) nodes can be in all layers but the first one. Additionally, the last layer of the FG contains only one exit node. In other words, since the lengths of paths can vary, exit nodes can appear in layers $l \geq 2$. An example FG is presented in Fig. 1.

The example in Fig. 1 presents the FG for the monitoring system composed of two cameras (x and y) which can be both entered and left and transitions between them are possible in both directions. The longest considered path has the length of two steps. Additional output steps are marked as nodes $o2$ and $o3$.

3.2 Prediction of the object movement based on the flowgraph

On the basis of collected statistical data about the paths and employing (1)–(4), the FG is constructed and appropriate weights are assigned to the edges. The next step is using the prepared FG to predict the movement of an object located within the supervised area. The result of Forward Route Reconstruction is a probability tree describing probable next steps of objects. The root of the probability tree (denoted x_{root}) is the node in FG which represents current step of the path of theobject which remains in the supervised area. The certainty values are used as probability measurements. In case of prediction more than one step ahead from the root, the probability needs to be estimated as a product of probabilities of subsequent steps.

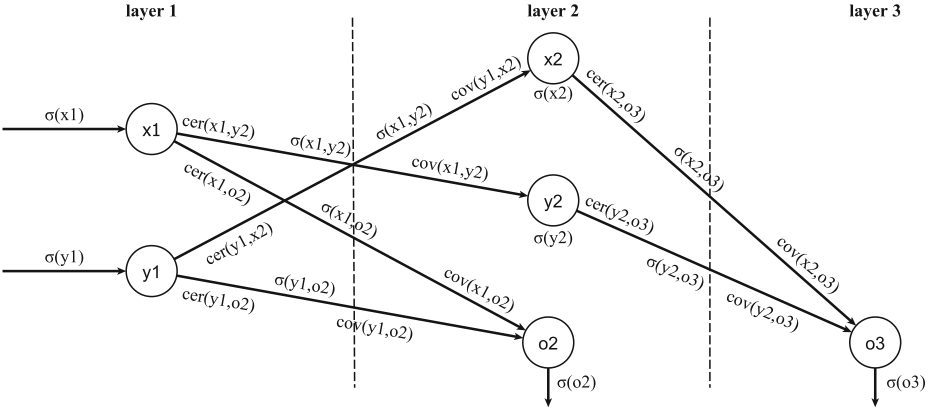


Fig. 1 The simple example of a modified FG employed in Forward Route Reconstruction

In (5) it is shown how the probability of the path from the node x_{root} to z_{end} is calculated (Pawlak 2005b):

$$cer\,[x_{\mathrm{root}}, \ldots, z_{\mathrm{end}}] = \sum_{i=\mathrm{root}}^{i=\mathrm{end}} cer\,(x_i, y_{i+1}) \tag{5}$$

The x_{root} position is changing as the object moves in the graph. Thus, the first node of the previous tree becomes a new root. Such prediction can be made as long as the object is present in the supervised area.

3.3 Updating the flowgraph

When the object enters the supervised area, it is assigned a copy of the FG. This copy is derived from the most up-to-date information. It is then used for movement prediction. The object's copy of the FG remains unchanged as long as the object is present in the system. When the object leaves the area, its path is used to update the general FG.

Updating the FG is based on adding a new path to the set of observed paths. The updated set of paths is an input for refreshing the FG by using (1)–(4). Another method cleans up the set of paths in order to remove paths which are obsolete. This method is triggered periodically in order to keep the database of paths up to date and to enable adaptation of the FG to the changing behaviour of objects. Adjusting this adaptation period influences the rate of 'forgetting' information from the past.

4 Usage example of flowgraph

The prototype of an example FG for Forward Route Reconstruction is a topological structure of the considered video surveillance system described with the topology graph. In our assumed situation, there will be four cameras and two of them have entrance and exit in their FOV(nodes x and y). A detailed topology is shown in Fig. 2.

The next stage is to obtain statistical data. An example set of paths is presented in Table 1. Paths can have various lengths and the number of layers in the FG depends on paths lengths. The paths in Table 1 are used to construct the FG shown in Fig. 3. The number of layers equals the number of steps in the longest path plus one, because of adding exit (output) nodes. Analysis of the FG structure allows to reconstruct the

Fig. 2 An example topology graph

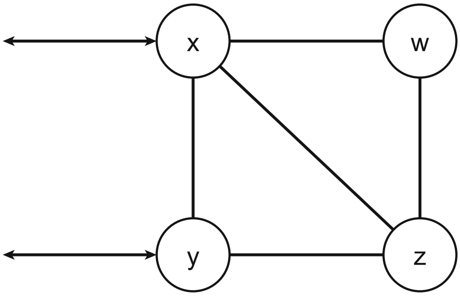


Table 1 An example set of paths

Index	Path	Index	Path
1	$x \rightarrow w \rightarrow z \rightarrow y$	9	$y \rightarrow x \rightarrow w \rightarrow z \rightarrow x$
2	$x \rightarrow y$	10	$\mathrm{x} \rightarrow w \rightarrow z \rightarrow x \rightarrow y$
3	$x \rightarrow w \rightarrow z \rightarrow x$	11	x
4	$y \rightarrow z \rightarrow w \rightarrow x$	12	y
5	$y \rightarrow x \rightarrow w \rightarrow z \rightarrow y$	13	x
6	$y \rightarrow z \rightarrow x$	14	$x \rightarrow z \rightarrow y$
7	$y \rightarrow z \rightarrow w \rightarrow x$	15	$y \rightarrow z \rightarrow x$
8	$x \rightarrow w \rightarrow z \rightarrow y$	16	$x \rightarrow z \rightarrow y$

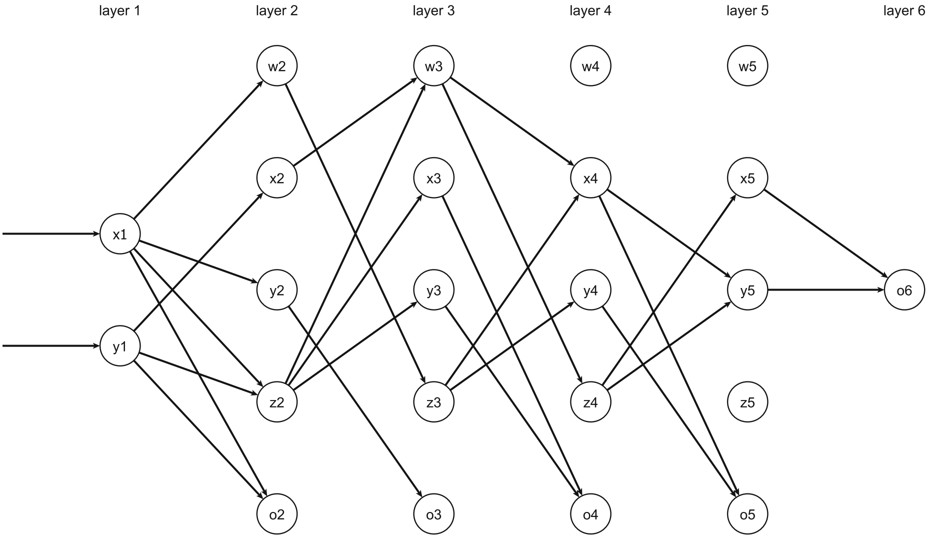


Fig. 3 The flowgraph constructed on the basis of a set of paths

Table 2 Flow measures of objects passing through given nodes

Node	$\varphi(x_i)$	$\sigma(x_i)$	Node	$\varphi(x_i)$	$\sigma(x_i)$
$x1$	9	$9/16 = 0.5625$	$z3$	4	$4/16 = 0.2500$
$y1$	7	$7/16 = 0.4375$	$o3$	1	$1/16 = 0.0625$
$w2$	4	$4/16 = 0.2500$	$x4$	4	$4/16 = 0.2500$
$x2$	2	$2/16 = 0.1250$	$y4$	2	$2/16 = 0.1250$
$y2$	1	$1/16 = 0.0625$	$z4$	2	$2/16 = 0.1250$
$z2$	6	$6/16 = 0.3750$	$o4$	4	$4/16 = 0.2500$
$o2$	3	$3/16 = 0.1875$	$x5$	1	$1/16 = 0.0625$
$w3$	4	$4/16 = 0.2500$	$y5$	2	$2/16 = 0.1250$
$x3$	2	$2/16 = 0.1250$	$o5$	5	$5/16 = 0.3125$
$y3$	2	$2/16 = 0.1250$	$o6$	3	$3/16 = 0.1875$

Table 3 Flow and probability measures for edges in the flowgraph

Edge($x_i \rightarrow y_j$)	$\varphi(x_i, y_j)$	$\sigma(x_i, y_j)$	$cer(x_i, y_j)$	$cov(x_i, y_j)$
$x1 \rightarrow w2$	4	0.2500	0.4444	1.0000
$x1 \rightarrow y2$	1	0.0625	0.1111	1.0000
$x1 \rightarrow z2$	2	0.1250	0.2222	0.3333
$x1 \rightarrow o2$	2	0.1250	0.2222	0.6667
$y1 \rightarrow x2$	2	0.1250	0.2857	1.0000
$y1 \rightarrow z2$	4	0.2500	0.5714	0.6667
$y1 \rightarrow o2$	1	0.0625	0.1429	0.3333
$w2 \rightarrow z3$	4	0.2500	1.0000	1.0000
$x2 \rightarrow w3$	2	0.1250	1.0000	0.5000
$y2 \rightarrow o3$	1	0.0625	1.0000	1.0000
$z2 \rightarrow w3$	2	0.1250	0.3333	0.5000
$z2 \rightarrow x3$	2	0.1250	0.3333	1.0000
$z2 \rightarrow y3$	2	0.1250	0.3333	1.0000
$w3 \rightarrow x4$	2	0.1250	0.5000	0.5000
$w3 \rightarrow z4$	2	0.1250	0.5000	1.0000
$x3 \rightarrow o4$	2	0.1250	1.0000	0.5000
$y3 \rightarrow o4$	2	0.1250	1.0000	0.5000
$z3 \rightarrow x4$	2	0.1250	0.5000	0.5000
$z3 \rightarrow y4$	2	0.1250	0.5000	1.0000
$x4 \rightarrow o5$	3	0.1875	0.7500	0.6000
$x4 \rightarrow y5$	1	0.0625	0.2500	0.5000
$y4 \rightarrow o5$	2	0.1250	1.0000	0.4000
$z4 \rightarrow y5$	1	0.0625	0.5000	0.5000
$z4 \rightarrow x5$	1	0.0625	0.5000	1.0000
$x5 \rightarrow o6$	1	0.0625	1.0000	0.3333
$y5 \rightarrow o6$	2	0.1250	1.0000	0.6667

topology graph on the basis of possible transitions between the cameras contained in the FG. In the graph presented in Fig. 3, some nodes are neither sources nor sinks, i.e., they are not connected to other nodes (e.g., $w4$, $w5$, $z5$). In this case camera was

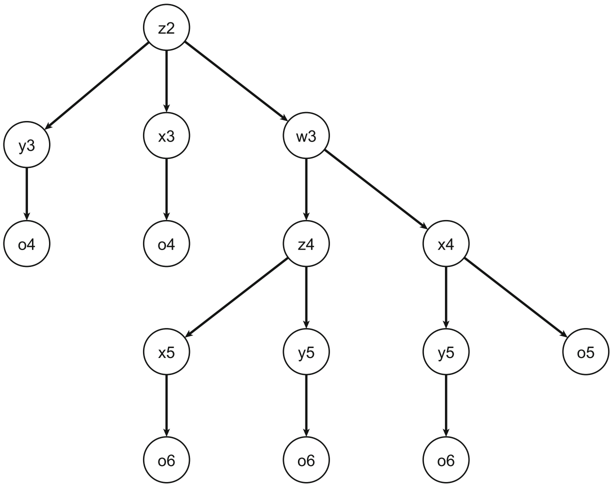


Fig. 4 Probability tree for the object O

Table 4 Probability measures for the tree with the root in node $z2$

Path $[x_{\text{root}}, \ldots, z_{\text{end}}]$	$cer\,[x_{\text{root}}, \ldots, z_{\text{end}}]$
$[z2, y3, o4]$	$0.3333 \cdot 1.0000 = 0.3333$
$[z2, x3, o4]$	$0.3333 \cdot 1.0000 = 0.3333$
$[z2, w3, x4, o5]$	$0.3333 \cdot 0.5000 \cdot 0.7500 = 0.1250$
$[z2, w3, x4, y5, o6]$	$0.3333 \cdot 0.5000 \cdot 0.2500 \cdot 1.0000 = 0.0417$
$[z2, w3, z4, x5, o6]$	$0.3333 \cdot 0.5000 \cdot 0.5000 \cdot 1.0000 = 0.0833$
$[z2, w3, z4, y5, o6]$	$0.3333 \cdot 0.5000 \cdot 0.5000 \cdot 1.0000 = 0.0833$

never a 4-th or 5-th step of any path. These nodes can be omitted and ignored in the process of creating the probability trees.

Equations (1)–(4) allow us to calculate the parameters for each edge in the FG. First, the number of transitions through each edge and through each node in the FG must be obtained. The measure of object flows through particular nodes is calculated according to (2) and presented in Table 2. Equation (1) determines the flows through the given edges. Therefore, a probability estimate of a particular transition can be obtained with (3) and (4). The listing of these probability measures is presented in Table 3.

The FG prepared in such way can be used to predict the movement of an object O which is located within the observed area. The current path of the object O is, for example, $x \rightarrow z \rightarrow ?$. Thus, in the FG, this object is localized in the node $z2$. The node $z2$ becomes the root of the nascent probability tree. The probability tree can be considered a subgraph of the FG, as it is shown in Fig. 4. In case of prediction of only one future step in the path of the object O, probability measures can be taken directly from the FG as values of $cer(z2, x3)$, $cer(z2, y3)$ and $cer(z2, w3)$. If more than one-step prediction is needed, (5) can be used to obtain the probability tree with root in $z2$. In the example, the query $[z2, w3, x4]$ results in $cer[z2, w3, x4] = 0.3333 \cdot 0.5000 = 0.1667$. Example queries, which are rooted in the node $z2$, and responses to them are presented in Table 4.

In this example, when the object O exits the supervised area, its path, i.e. $x \rightarrow z \rightarrow w \rightarrow z \rightarrow y$, is added to the set of paths and the FG is refreshed and rebuilt. The new version of the FG is built on the updated set of paths. If the path of the object O is the same as the one present in the set of paths, there are no changes in the structure of the FG, but the values $\varphi(x_i, y_j)$, $\varphi(x_i)$, $\varphi(G)$, $\sigma(x_i, y_j)$, $cer(x_i, y_j)$, and $cov(x_i, y_j)$ are updated. The new FG parameters are presented in Tables 5 and 6. The values

Table 5 Updated flow measures of objects passing through given nodes

Node	$\varphi(x_i)$	$\sigma(xi)$	Node	$\varphi(x_i)$	$\sigma(x_i)$
$x1$	10	$10/17 = 0.5882$	$z3$	4	$4/17 = 0.2353$
$y1$	7	$7/17 = 0.4118$	$o3$	1	$1/17 = 0.0588$
$w2$	4	$4/17 = 0.2353$	$x4$	4	$4/17 = 0.2353$
$x2$	2	$2/17 = 0.1176$	$y4$	2	$2/17 = 0.1176$
$y2$	1	$1/17 = 0.0588$	$z4$	3	$2/17 = 0.1765$
$z2$	7	$7/17 = 0.4118$	$o4$	4	$4/17 = 0.2353$
$o2$	3	$3/17 = 0.1765$	$x5$	1	$1/17 = 0.0588$
$w3$	5	$5/17 = 0.2941$	$y5$	3	$3/17 = 0.1765$
$x3$	2	$2/17 = 0.1176$	$o5$	5	$5/17 = 0.2941$
$y3$	2	$2/17 = 0.1176$	$o6$	4	$4/17 = 0.2353$

Table 6 Updated flow and probability measures for edges in the flowgraph

Edge($xi \to yj$)	$\varphi(x_i, y_j)$	$\sigma(x_i, y_j)$	$cer(x_i, y_j)$	$cov(x_i, y_j)$
$x1 \to w2$	4	0.2353	0.4000	1.0000
$x1 \to y2$	1	0.0588	0.1000	1.0000
$x1 \to z2$	3	0.1176	0.3000	0.4286
$x1 \to o2$	2	0.1176	0.2000	0.6667
$y1 \to x2$	2	0.1176	0.2857	1.0000
$y1 \to z2$	4	0.2353	0.5714	0.5714
$y1 \to o2$	1	0.0588	0.1429	0.3333
$w2 \to z3$	4	0.2353	1.0000	1.0000
$x2 \to w3$	2	0.1176	1.0000	0.4000
$y2 \to o3$	1	0.0588	1.0000	1.0000
$z2 \to w3$	3	0.1765	0.4286	0.6000
$z2 \to x3$	2	0.1176	0.1176	1.0000
$z2 \to y3$	2	0.1176	0.1176	1.0000
$w3 \to x4$	2	0.1176	0.4000	0.5000
$w3 \to z4$	3	0.1765	0.6000	1.0000
$x3 \to o4$	2	0.1176	1.0000	0.5000
$y3 \to o4$	2	0.1176	1.0000	0.5000
$z3 \to x4$	2	0.1765	0.5000	0.5000
$z3 \to y4$	2	0.1176	0.5000	1.0000
$x4 \to o5$	3	0.1765	0.7500	0.6000
$x4 \to y5$	1	0.1176	0.2500	0.3333
$y4 \to o5$	2	0.1176	1.0000	0.4000
$z4 \to y5$	2	0.1176	0.6667	0.6667
$z4 \to x5$	1	0.0588	0.3333	1.0000
$x5 \to o6$	1	0.0588	1.0000	0.2500
$y5 \to o6$	3	0.1765	1.0000	0.7500

which must be updated because of adding a new path to the set of paths, are shaded. If another object enters the supervised area, it will get the newest updated version of the FG.

5 Computational complexity

In order to determine computational complexity some assumptions must be made. These assumptions are related to a practical use of the flowgraph idea. A computer representation of the graph has an impact on the space complexity and the time complexity. In general, a graph can be stored in the computer memory as an adjacency matrix or as an adjacency list. Moreover, the matrix adjacency structure is more suitable for a dense graph that is a graph in which the number of edges is bigger than the squared number of vertices. The structure of the adjacency list is recommended for sparse graphs (Kubale 1998).

The following assumptions were made in order to determine the computational complexity: the number of cameras in system (corresponding to the number of vertices in a single layer) C is constant, the number of steps in the longest path (corresponding to the number of layers in the FG) L is equal to C. From the practical point of view the number of cameras in a deployed surveillance system changes very rarely. Additionally, we assume that all paths are as long as possible (the length is equal to L).

Thus, the number of vertices in the theoretically worse case can be estimated as $V = L \cdot (C + 1) \approx C2$. Therefore, the number of edges in the FG is strongly related to the number of edges in the topology graph. In the case, when a topology graph is a clique and the FG contains all possible transitions between consecutive layers, the maximum number of edges can be determined as $E = C \cdot [1 + (L - 1) \cdot (C + 1)]) \approx C3$. Going one step further the FG can be considered as an L-partite graph. In this case, an application of the representation of adjacency list is more suitable for the FG.

The space complexity (amount of needed memory) of the FG represented as an adjacency list is equal to $O(V + E)$. Following the assumptions that C and L are constant, the space complexity is $O(1)$.

Application of associative containers to store vertices and edges entails a constant time of access or adding a vertex and $O(\log(E/V))$ time complexity while accessing or adding an edge. The time complexity of the operation of adding a new sample (path) to the FG is $O(L \cdot \log(E/V))$ but according to the assumptions made, the time complexity is $O(1)$. After adding a path to the FG (changing values of $\varphi()$) an updating process (changing values of $\sigma()$, $cer()$, and $cov()$) occurs. This process has the time complexity $O(C^2 \cdot \log(E/V))$. If we assume that C and L are constant, then the time complexity of updating process is $O(1)$.

6 Results of experiments

The topology graph, on which the possible inter-camera transitions are presented, is shown in Fig. 5. For the experiment, four cameras were selected. Each camera was considered an entrance-exit camera. The views from these cameras are presented in Fig. 6.

In the presented setup, analysis of 6 hours of the recorded video was performed and 925 paths of objects were obtained. The analysis was made by a person but automated techniques application is also possible, provided that tracking methods are adequately efficient. Otherwise an error propagation would occur. The 925 paths are summarized in the FG (Fig. 7).

The values of strength and certainty for each edge in the FG are contained in Table 7. Values of $cer(x_i, y_j)$ are used to create probability trees using the methods presented in previous sections. Probabilities of particular paths, which can be predicted from the probability tree with the root node placed in node $A1$ are presented in Table 8.

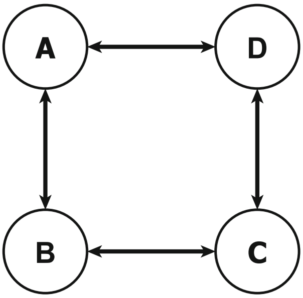
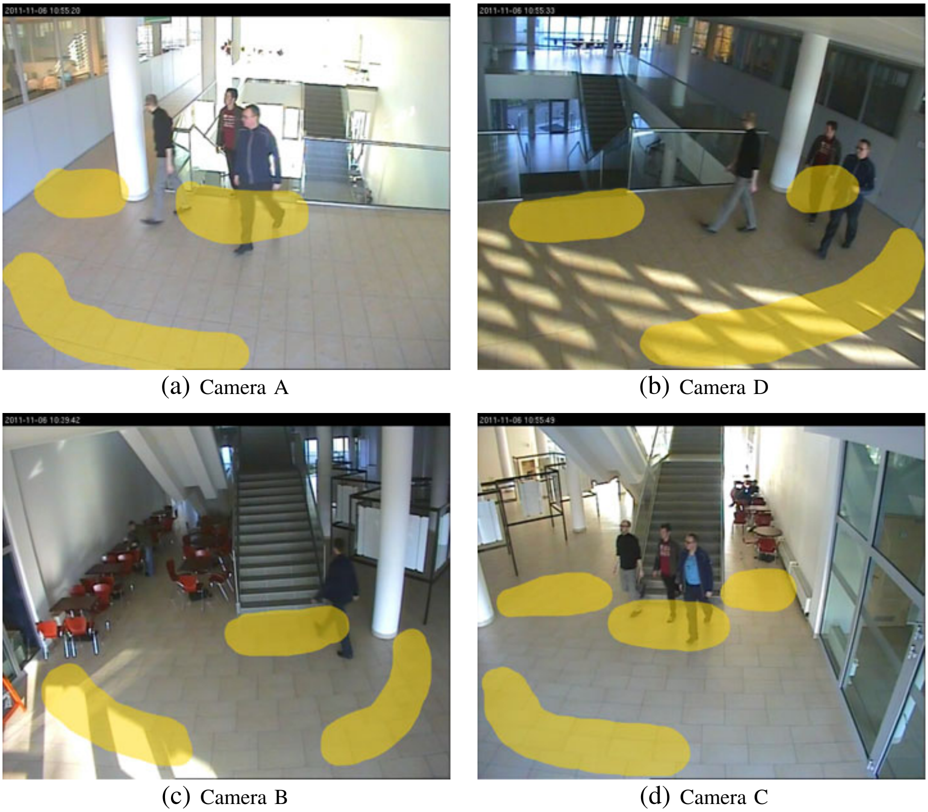


Fig. 5 The topology graph of the set of cameras in the experiment



(a) Camera A

(b) Camera D

(c) Camera B

(d) Camera C

Fig. 6 Screenshots of cameras fields of view used in the experiment, areas where people go in and go out from FOVs of the cameras are marked with *yellow color*

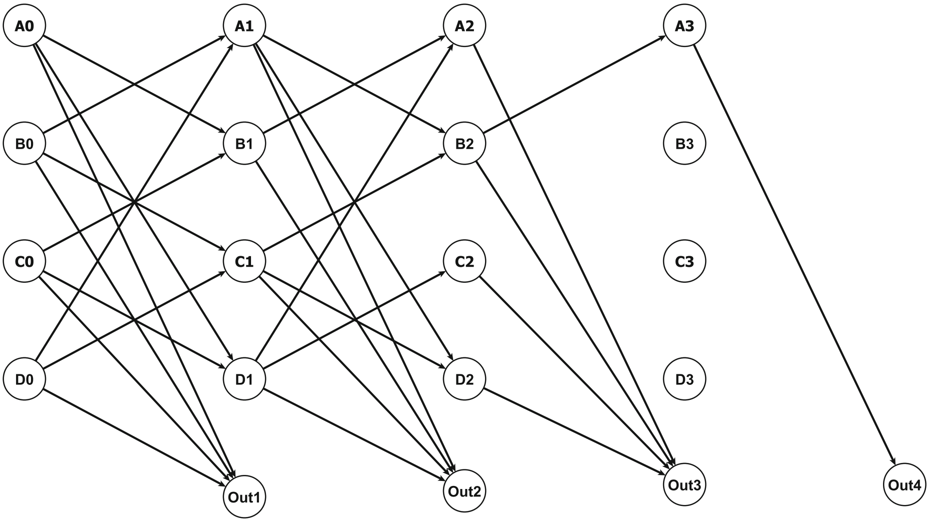


Fig. 7 The flowgraph presenting the statistical structure of people behaviour

Table 7 Probability measurements calculated for the collected set of paths

Edge($x_i \to y_j$)	$\sigma(x_i, y_j)$	$cer(x_i, y_j)$	Edge($x_i \to y_j$)	$\sigma(x_i, y_j)$	$cer(x_i, y_j)$
$A0 \to B1$	0.1524	0.4764	$B1 \to A2$	0.0022	0.0135
$A0 \to D1$	0.1157	0.3615	$B1 \to Out2$	0.1578	0.9865
$A0 \to Out1$	0.0519	0.1622	$C1 \to B2$	0.0130	0.1212
$B0 \to A1$	0.1611	0.6712	$C1 \to D2$	0.0032	0.0303
$B0 \to C1$	0.0076	0.0315	$C1 \to Out2$	0.0908	0.8484
$B0 \to Out1$	0.0714	0.2973	$D1 \to A2$	0.0497	0.2190
$C0 \to B1$	0.0076	0.0398	$D1 \to C2$	0.0162	0.0714
$C0 \to D1$	0.1114	0.5852	$D1 \to Out2$	0.1611	0.7095
$C0 \to Out1$	0.0714	0.3750	$A2 \to Out3$	0.0519	1.0000
$D0 \to A1$	0.1027	0.4113	$B2 \to A3$	0.0032	0.1579
$D0 \to C1$	0.0995	0.3983	$B2 \to Out3$	0.0173	0.8421
$D0 \to Out1$	0.0476	0.1905	$C2 \to Out3$	0.0162	1.0000
$A1 \to B2$	0.0076	0.0287	$D2 \to Out3$	0.0205	1.0000
$A1 \to D2$	0.0173	0.0656	$A3 \to Out4$	0.0032	1.0000
$A1 \to Out2$	0.2389	0.9057			

In order to validate the knowledge contained in the prepared FG a set of the new paths gathered in the same place at different times were added to the previously created database of paths (for this purpose 101 new paths were obtained). Average absolute deviation is used as a measure of the new FG conformity with the old FG. Calculation of this measure is performed according to (6).

$$D = \frac{\sum\limits_{\text{Edges}} \left| cer_1\left(x_i, y_j\right) - cer_0\left(x_i, y_j\right) \right|}{N} \tag{6}$$

where D determines how much the new FG differs from the previous one, $cer_1(x_i, y_j)$ and $cer_0(x_i, y_j)$ are certainty parameters assigned to the edge $x_i \to y_j$ in the new and old FG respectively and N is the number of edges in the new FG. In the case when a new edge is added, which was not present in the old FG, we assume that $cer_0(x_i, y_j) = 0$. The values of $cer(x_i, y_j)$ are normalized to 1, thus the value of D describes the degree of changes in the FG. When D is equal to 0 it is understood that no changes were made to the FG. Calculation of measure of conformity for the extended set of paths yields $D = 0.0158$. This value equals the average change in certainty of edges in the FG.

Table 8 Forward Route Reconstruction results—probabilities of future path of the object which is currently in the second step of its path and is observed by the Camera A

Path $[x_{\text{root}}, \ldots, z_{\text{end}}]$	$cer[x_{\text{root}}, \ldots, z_{\text{end}}]$
$[A1, B2]$	0.0287
$[A1, B2, A3]$	0.0045
$[A1, B2, A3, Out4]$	0.0045
$[A1, B2, Out3]$	0.0242
$[A1, D2]$	0.0656
$[A1, D2, Out3]$	0.0656
$[A1, Out2]$	0.9057

7 Conclusions

Pawlak's flowgraphs can be successfully applied to video analysis in a surveillance system which observes areas inside buildings. The useful feature of the FG is automatic creation and updating of rules, as well as clear representation of the object route statistics. Moreover, probability measurements can be used to predict next steps of the object's movement on the basis of a hitherto traversed path and information contained in the FG. The next advantage is possibility of using finished paths for updating the FG. Moreover, prediction of future steps in the path of an object can be made for various numbers of steps ahead of the current location.

An extension of the presented algorithm can be a more sophisticated method for updating the FG that would be based on assessment of changes in general behaviour of objects observed by a surveillance system, e.g., based on amount of information contained in FG. Additionally, representation of time-dependencies between cameras can also be included in the FG.

Acknowledgements This work is associated with the ADDPRIV ("Automatic Data relevancy Discrimination for a PRIVacy-sensitive video surveillance") Project No. 261653, being a part of the European Seventh Framework Program (FP7).

References

Cheng, Y.S., Huang, C.M., Fu, L.C. (2006). Multiple people visual tracking in a multi-camera system for cluttered environments. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 675–680). doi:10.1109/IROS.2006.282554.

Czyzewski, A., & Kostek, B. (2004). Musical metadata retrieval with flow graphs. In S. Tsumoto, R. Slowinski, J. Komorowski, J. Grzymala-Busse (Eds.), *Rough sets and current trends in computing. Lecture notes in computer science* (Vol. 3066, pp. 691–698). Berlin: Springer. doi:10.1007/978-3-540-25929-9_87.

Greco, S., Pawlak, Z., Slowinski, R. (2002). Generalized decision algorithms, rough inference rules, and flow graphs. In J. Alpigini, J. Peters, A. Skowron, N. Zhong (Eds.), *Rough sets and current trends in computing. Lecture notes in computer science* (Vol. 2475, pp. 93–104). Berlin: Springer. doi:10.1007/3-540-45813-1_12.

Javed, O., Rasheed, Z., Shafique, K., Shah, M. (2003). Tracking across multiple cameras with disjoint views. In *Proceedings. Ninth IEEE international conference on computer vision* (Vol. 2, pp. 952–957). doi:10.1109/ICCV.2003.1238451.

Kim, H., Romberg, J., Wolf, W. (2009). Multi-camera tracking on a graph using Markov chain Monte Carlo. In *Third ACM/IEEE international conference on distributed smart cameras. ICDSC 2009* (pp. 1–8). doi:10.1109/ICDSC.2009.5289352.

Kubale, M. (1998). *Introduction to computational complexity and algorithmic graph coloring*. Gdansk: Gdanskie Towarzystwo Naukowe.

Leung, V., Orwell, J., Velastin, S. (2008). Performance evaluation of re-acquisition methods for public transport surveillance. In *10th international conference on control, automation, robotics and vision. ICARCV 2008* (pp. 705–712). doi:10.1109/ICARCV.2008.4795604.

Lev-Tov, A., & Moses, Y (2010). Path recovery of a disappearing target in a large network of cameras. In *Proceedings of the fourth ACM/IEEE international conference on distributed smart cameras. ICDSC '10* (pp. 57–64). New York: ACM. doi:10.1145/1865987.1865997.

Ou Yang, Y.P., Shieh, H.M., Tzeng, G.H., Yen, L., Chan, C.C. (2008). Business aviation decision-making using rough sets. In C.C. Chan, J. Grzymala-Busse, W. Ziarko (Eds.), *Rough sets and* *current trends in computing. Lecture notes in computer science* (Vol. 5306, pp. 329–338). Berlin: Springer. doi:10.1007/978-3-540-88425-5_34.

Pawlak, Z. (2003). Rough sets, Bayes' theorem and flow graphs. In L.F.B. Bouchon-Meunier & R. Yager (Eds.), *Intelligent systems for information processing from representation to applications*. Amsterdam: Elsevier.

Pawlak, Z. (2005a). Flow graphs and data mining. In J.F. Peters, & A. Skowron (Eds.), *Transactions on rough sets III. Lecture notes in computer science* (Vol. 3400, pp. 1–36). Berlin: Springer. doi:10.1007/11427834_1.

Pawlak, Z. (2005b). Rough sets and flow graphs. In D. Slezak, G. Wang, M. Szczuka, I. Duntsch, Y. Yao (Eds.), *Rough sets, fuzzy sets, data mining, and granular computing. Lecture notes in computer science* (Vol. 3641, pp. 1–11). Berlin: Springer. doi:10.1007/11548669_1.

Pawlak, Z. (2005c). Some remarks on conflict analysis. *European Journal of Operational Research, 166*, 649.

Pawlak, Z. (2006a). Conflicts and negotations. In G. Wang, J.F. Peters, A. Skowron, Y. Yao (Eds.), *Rough sets and knowledge technology* (pp. 12–27). doi:10.1007/11795131_2.

Pawlak, Z. (2006b). Decision trees and flow graphs. In S. Greco, Y. Hata, S. Hirano, M. Inuiguchi, S. Miyamoto, H. Nguyen, R. Slowinski (Eds.), *Rough sets and current trends in computing. Lecture notes in computer science* (Vol. 4259, pp. 1–11). Berlin: Springer. doi:10.1007/11908029_1.